

Programming

Programming

- Remote Control
- Built-in VBA Programming
- TDR Remote Control (Option TDR)
- Command Reference
- TDR Command Reference (Option TDR)

Remote Control

Remote Control

- Overview
- Setting up Analyzer
- Performing Calibration
- Making Measurement
- Reading-Writing Measurement Data
- Analyzing Data
- Limit Test
- Saving and Recalling
- Communication with External Devices
- Status Reporting System
- Working with Automatic Test System
- Sample Programs

Other topics about Programming

Overview

Overview

- Types of remote control system
- GPIB remote control system
- LAN remote control system
- USB Remote Control System
- Sending SCPI command messages
- LXI

Types of remote control system

Depending on the system controller and the interface, you can configure 4 types of remote control system as shown in the table below.

System controller	Interface	Overview
External controller (external computer such as PC and workstation)	GPIB (talker/listener mode)	System to control the E5071C and other devices connected via GPIB from the external controller. For more information, refer to GPIB remote control system.
	LAN	System to control the E5071C and other devices connected via LAN from the external controller. For more information, refer to LAN remote control system.
	USB	System to control the E5071C and other devices connected via USB from the external controller. For more information, refer to USB Remote Control System.
E5071C		System to control the E5071C itself using built-in E5071C VBA.
	GPIB (system controller mode)	System to control the E5071C itself and external devices connected via GPIB using built-in E5071C VBA.

Other topics about Overview

GPIB remote control system

- [About GPIB](#)
- [System Configuration](#)

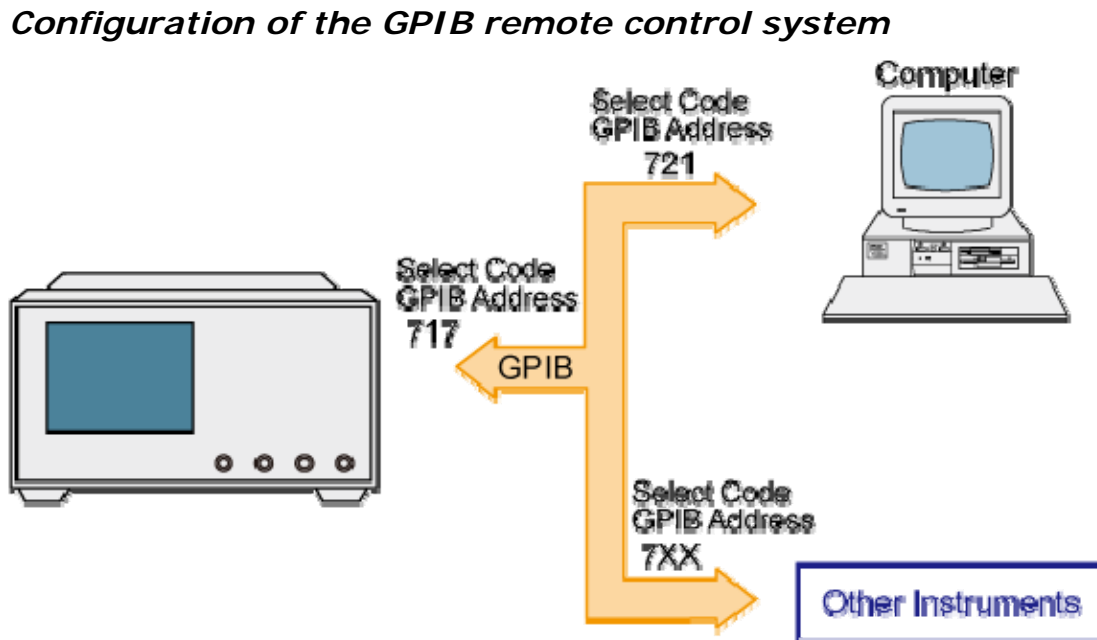
Other topics about Overview

About GPIB

GPIB (General Purpose Interface Bus) is an interface standard for connecting computers and peripherals, which supports the following international standards: IEEE 488.1, IEC-625, IEEE 488.2, and JIS-C1901. The GPIB interface allows you to control the Agilent E5071C from an external computer. The computer sends commands and instructions to the E5071C and receives data sent from the E5071C via GPIB.

System Configuration

Use GPIB cables to connect between the E5071C, the external controller (computer), and peripherals. The following figure shows the overview of the system configuration of the GPIB remote control system.



e5071c337

NOTE

While the E5071C is turned off, the SRQ status of the E5071C is active. To prevent an incorrect operation on the SRQ of the GPIB remote control system, disconnect the E5071C from the system when the E5071C is turned off.

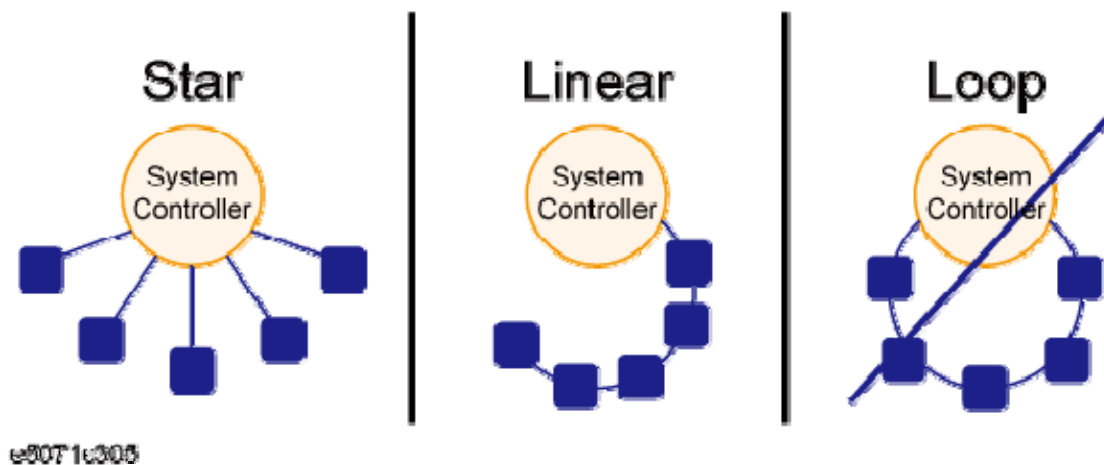
E5071C

Required Equipment

- E5071C
- External controller (PC or workstation that can be connected to LAN and Agilent I/O Library is installed into)
- Other devices (other instruments and/or peripherals that serve your purpose)
- GPIB cables

Scale of system you can construct

- You can connect up to 15 devices in a single GPIB system.
- The length of cables to connect between devices must be 4 m or less. The total length of connecting cables in a single GPIB system must be $2 \text{ m} \times \text{the number of connected devices (including the controller)}$ or less. You cannot construct the system in which the total cable length exceeds 20 m.
- The number of connectors connected to an individual device must be 4 or less. If you connect 5 or more connectors, excessive force is applied to the connector part, which may result in failure.
- You can choose the device connection topology from star, linear, and combined. Loop connection is not supported.



Device selector

The device selector is a unique value assigned to each device that is used by the controller to select the control target (to send/receive messages) among devices connected on the GPIB remote control system.

The device selector consists of a select code (usually, 7) and a GPIB address. For example, when the select code is 7 and the GPIB address is

17, the device selector is 717. The select code must be set for each system. The GPIB address must be set to a unique value for each device, which is used to identify devices on the same system. In the description and sample programs in this manual, it is assumed that the device selector is set to 717.

Setting the GPIB address of E5071C

To set the GPIB address for talker/listener mode, See `Setting_talker_listener_GPIB_address_of_E5071C`.

LAN remote control system

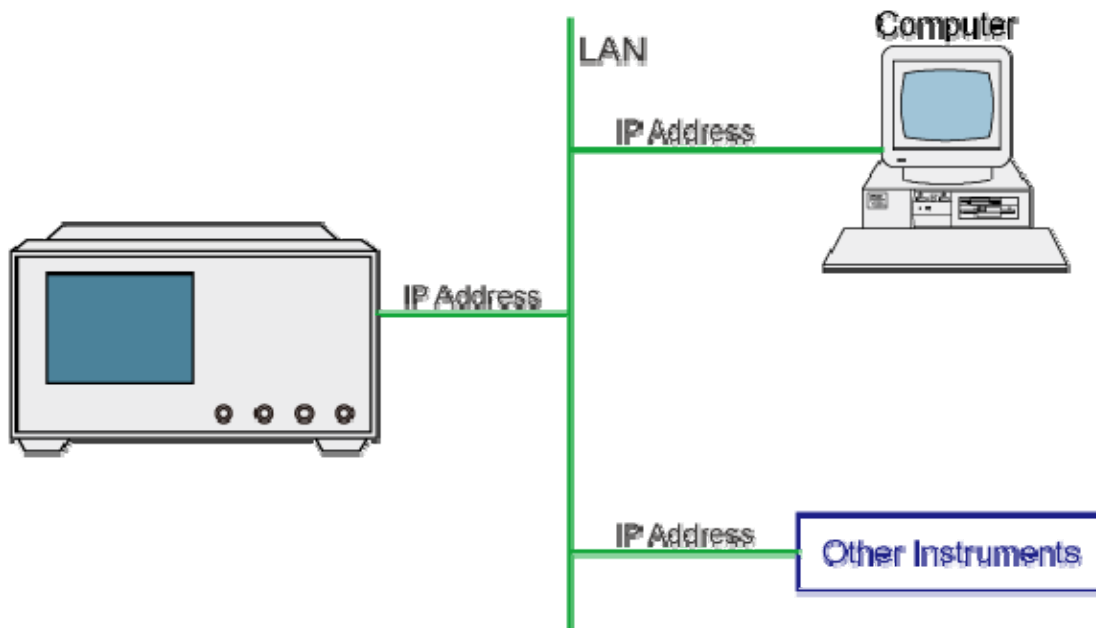
- [Overview](#)
- [System Configuration](#)
- [Required Equipment](#)
- [Control over SICL-LAN Server](#)
- [Control using C or Visual Basic](#)
- [Control using Agilent VEE](#)
- [Control with Telnet Server](#)

Other topics about Overview**Overview**

The LAN (Local Area Network) remote control system provides two methods: controlling the E5071C using the SICL-LAN server and controlling the E5071C using the telnet server.

System Configuration

Use a LAN cable to connect between the E5071C and the external controller (computer). The following figure shows the overview of the system configuration of the LAN remote control system.

Configuration of the LAN remote control system

e5071c338

Required Equipment

- E5071C
- External controller (PC or workstation that can be connected to LAN)
- Other devices (other instruments and/or peripherals that serve your purpose)
- LAN cables

Control over SICL-LAN Server

In the control system using the SICL-LAN server, communication between the external controller (client) and the E5071C (server) is performed using the SICL-LAN protocol. Communication is performed using SICL (Standard Instrument Control Library). You can control the E5071C by programming using SICL or VISA with the C language in the UNIX environment, or Visual C++, Visual Basic, or VEE in the Windows environment.

Preparing the E5071C

To communicate with the external controller, follow these steps to turn on the SICL-LAN server of the E5071C in advance.

1. Turn on the SICL-LAN server of the E5071C.

System > Misc Setup > Network Setup > SICL-LAN Server [ON]

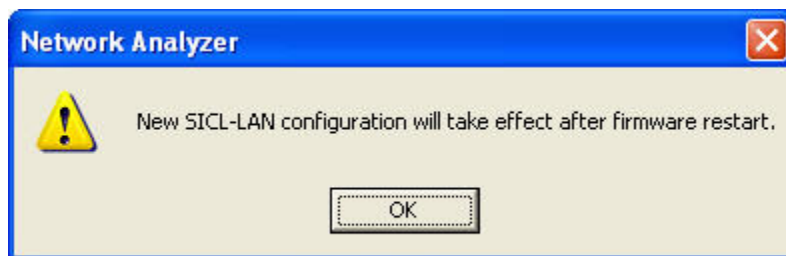
NOTE

When the SICL-LAN server is turned ON for the first time, the windows firewall setting dialog box appears. To use the SICL-LAN server, select **Unblock** and click **OK**.

2. Set the GPIB address of the E5071C for control with the SICL-LAN server. "XX" represents an address number.

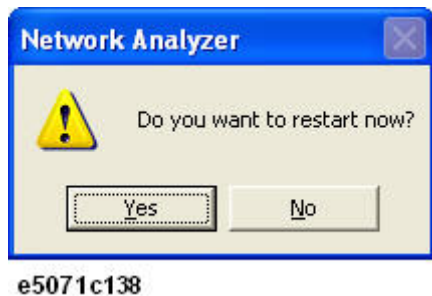
System > Misc Setup > Network Setup > SICL-LAN Address [XX]

3. By default, the SICL-LAN Address does not changes until the firmware of E5071C is restarted.



e5071c137

4. On pressing any key, a message appears for restarting the firmware. Click **Yes** to restart the firmware.



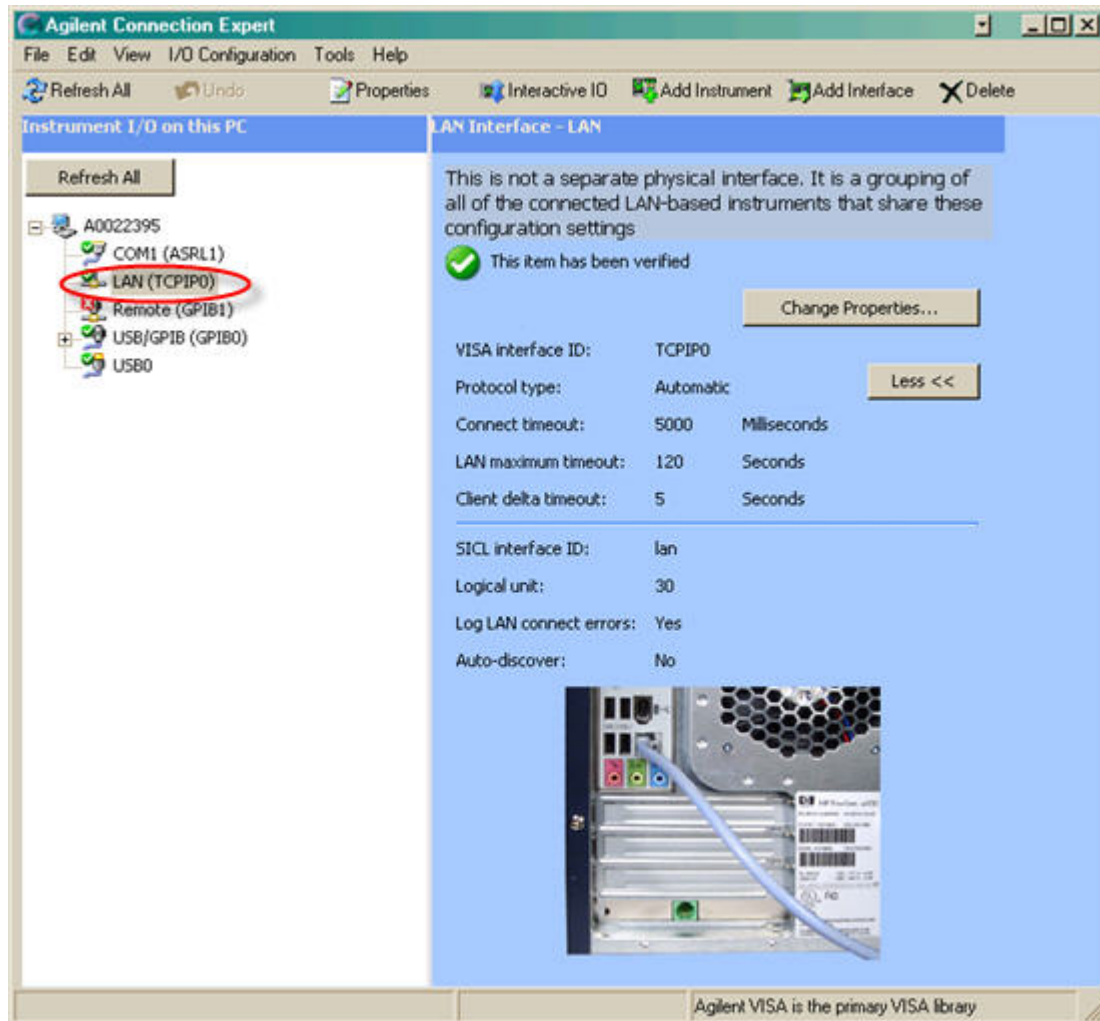
Preparing the external controller

In order to establish communication to the E5071C using the TCP/IP protocol, you need to set the I/O interface of the external controller in advance. This section shows the setting procedure when using the external controller in the Windows environment.

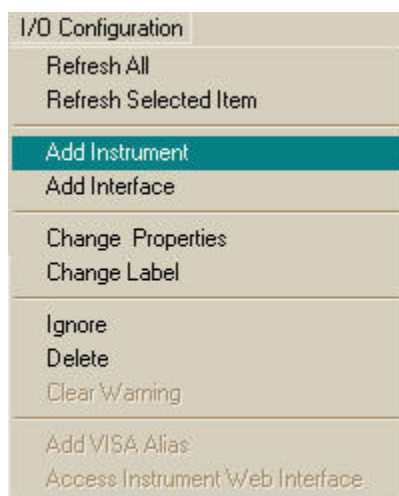
NOTE

You must install the Agilent I/O Libraries on your PC in advance. Use Agilent I/O Libraries Suite 14.2 or later.

1. From your PC's Start menu, click **Program > Agilent I/O Libraries Suite > Agilent Connection Expert** to open the Agilent Connection Expert setting screen.
2. In the Agilent Connection Expert setting screen, select **LAN(TCPIP0)** in the **Instrument I/O on this PC** frame, and then click **I/O Configuration > Add Instrument**.

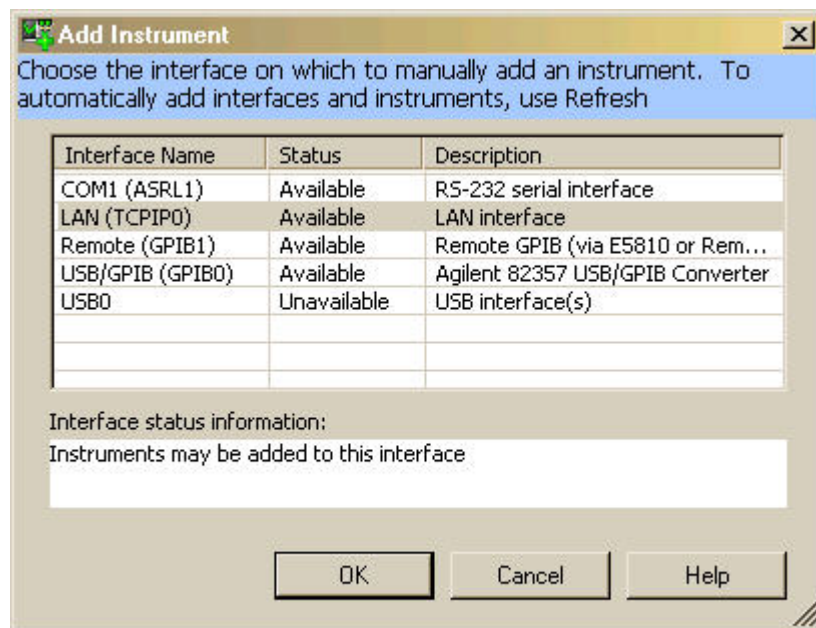


e5071c139



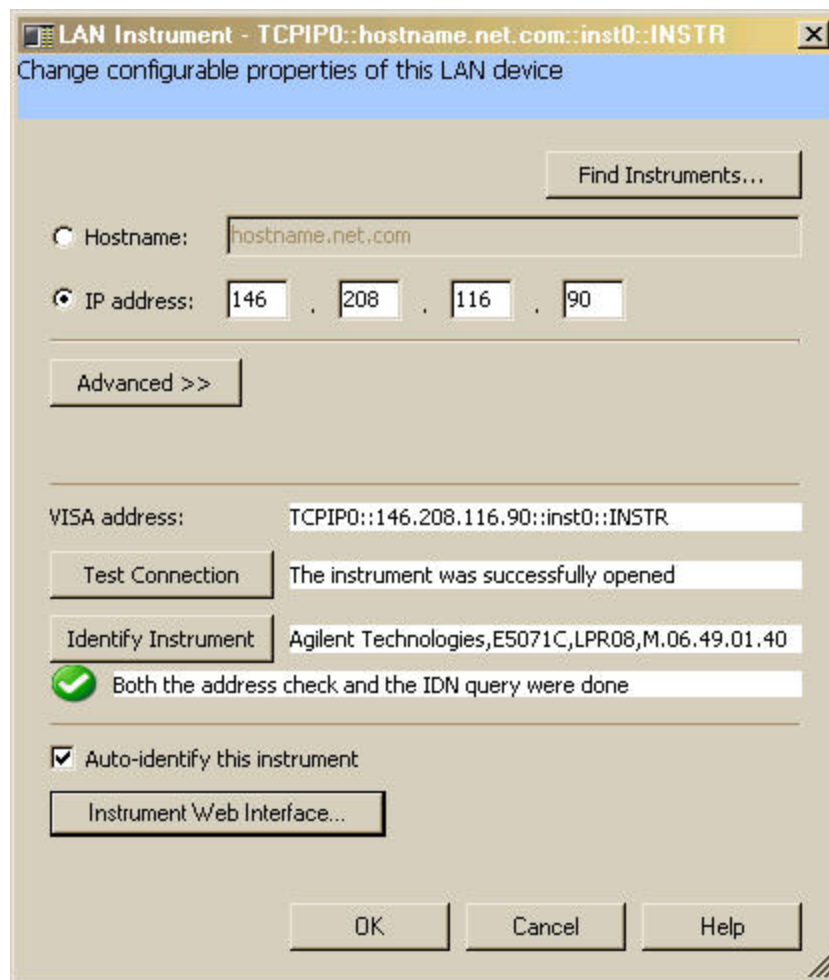
e5071c140

3. In the **Add Instrument** screen, select **LAN** (if it is not selected), and then click **OK**.



e5071c141

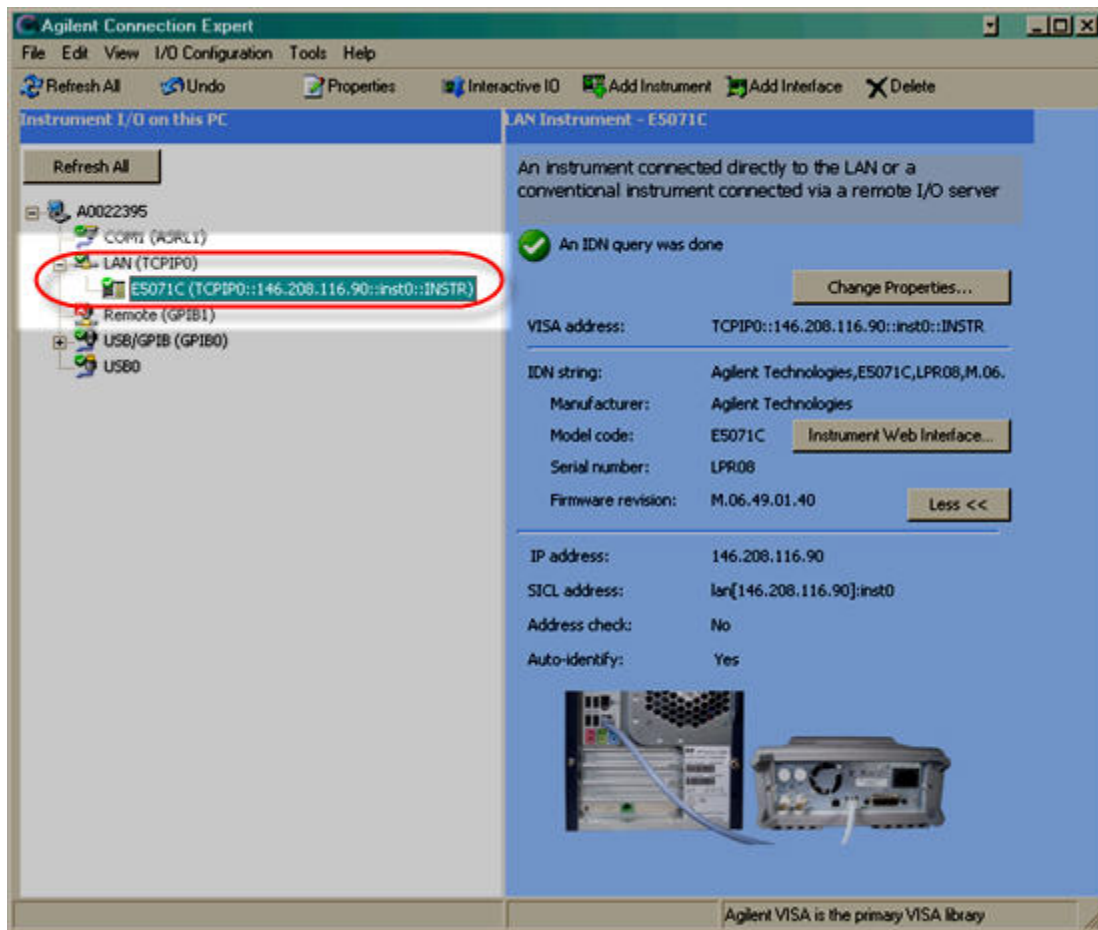
4. In the LAN Instrument Properties screen, set up the IP address of the E5071C and click **OK**. You can change settings as necessary. For details, refer to the Agilent I/O Libraries Suite documentation.



e5071c142

5. In the Agilent Connection Expert screen, check that the E5071C has been added under **LAN(TCPIP0)** in the **Instrument I/O on this PC** frame.

E5071C



e5071c143

Control using C or Visual Basic

You can control the E5071C by programming using SICL with the C language in the UNIX environment, or Visual C++ or Visual Basic in the Windows environment.

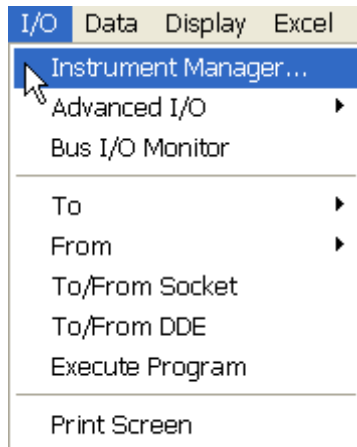
Control using Agilent VEE

Agilent VEE allows you to control the E5071C via the I/O interface. The following example shows how to control the E5071C that is set as follows: the address of the SICL-LAN server is 17 and the IP address is 146.208.116.90.

NOTE

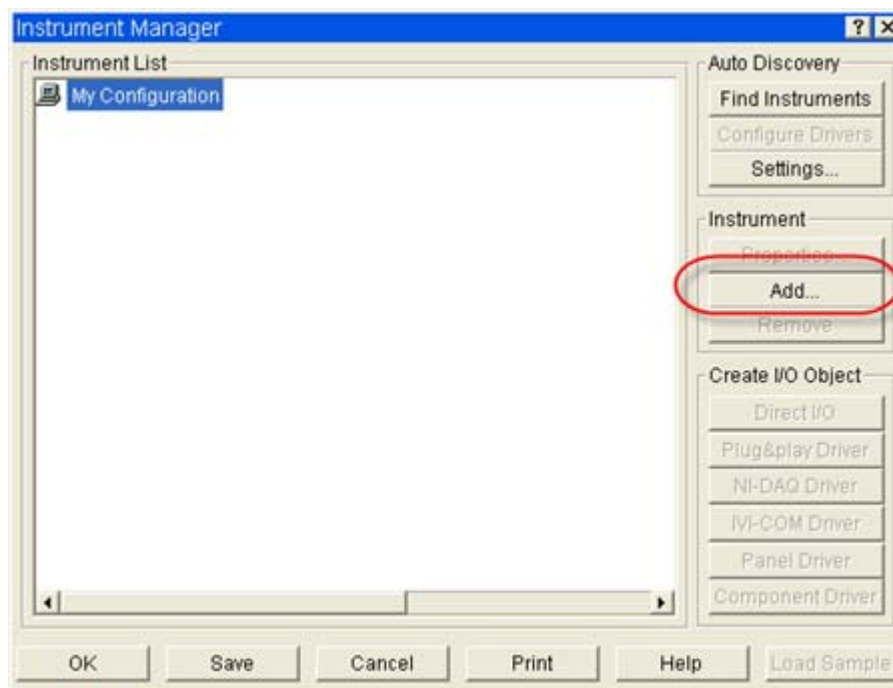
When using Agilent VEE for PC, use Agilent VEE Pro 7.5 for Windows or later.

1. On the Agilent VEE's **I/O** menu, click **Instrument Manager**



e5071c146

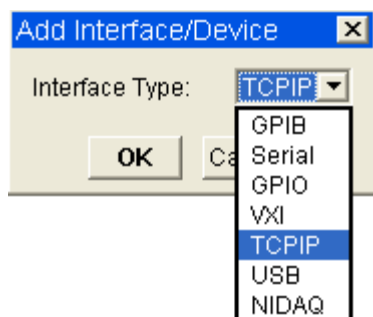
2. In **Instrument Manager**, click **Add...**



e5071c147

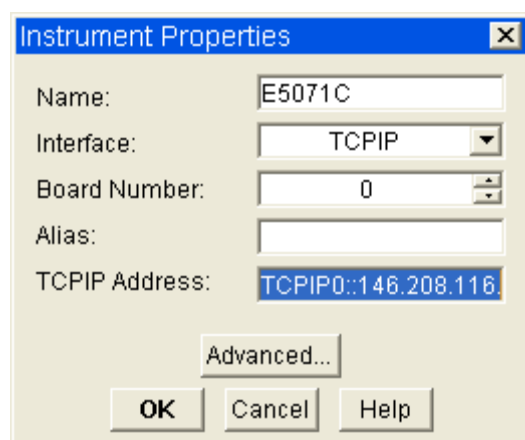
3. A new windows appears for the selection of Interface Type. Select **TCPIP** and click **OK**.

E5071C



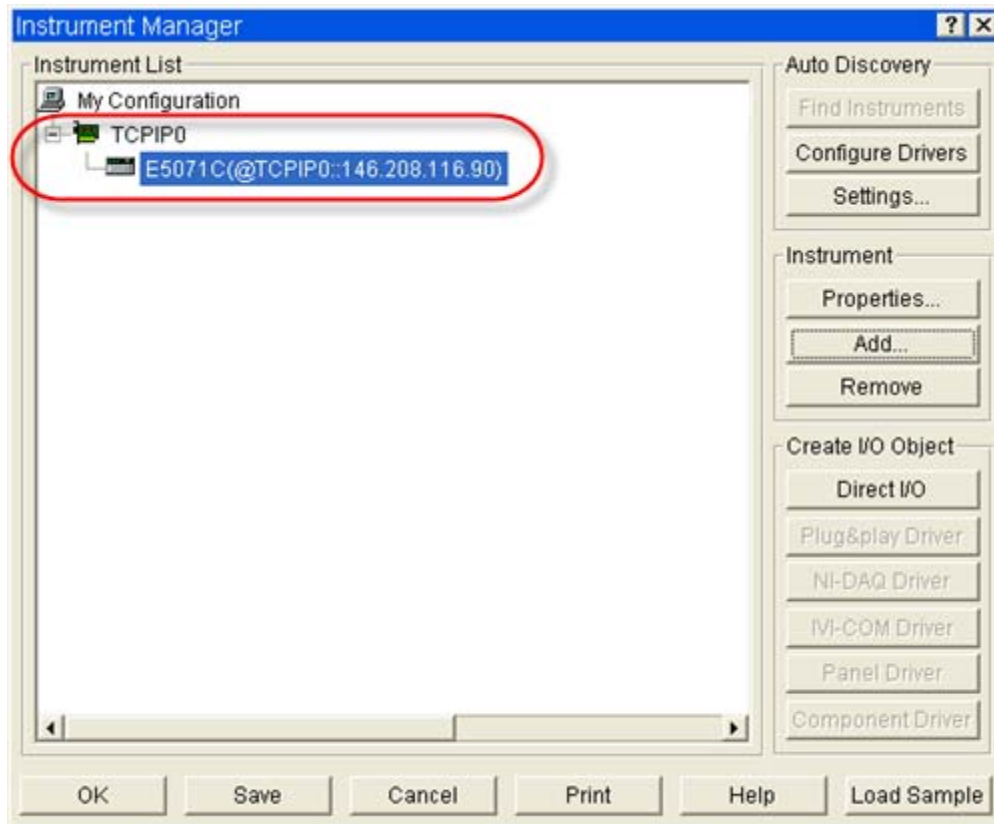
e5071c148

4. In **Instrument Properties**, type any name for the Instrument in Name (for example: ENA or E5071C), and add TCPIP0::<IP Address> in the **TCIP Address**, where <IP Address> is the IP address for E5071C. For example, if the IP address for E5071C is 146.208.116.90, then the value for **TCPIP Address** would be TCPIP0::146.208.116.90. Click **OK** after entering all the parameters.



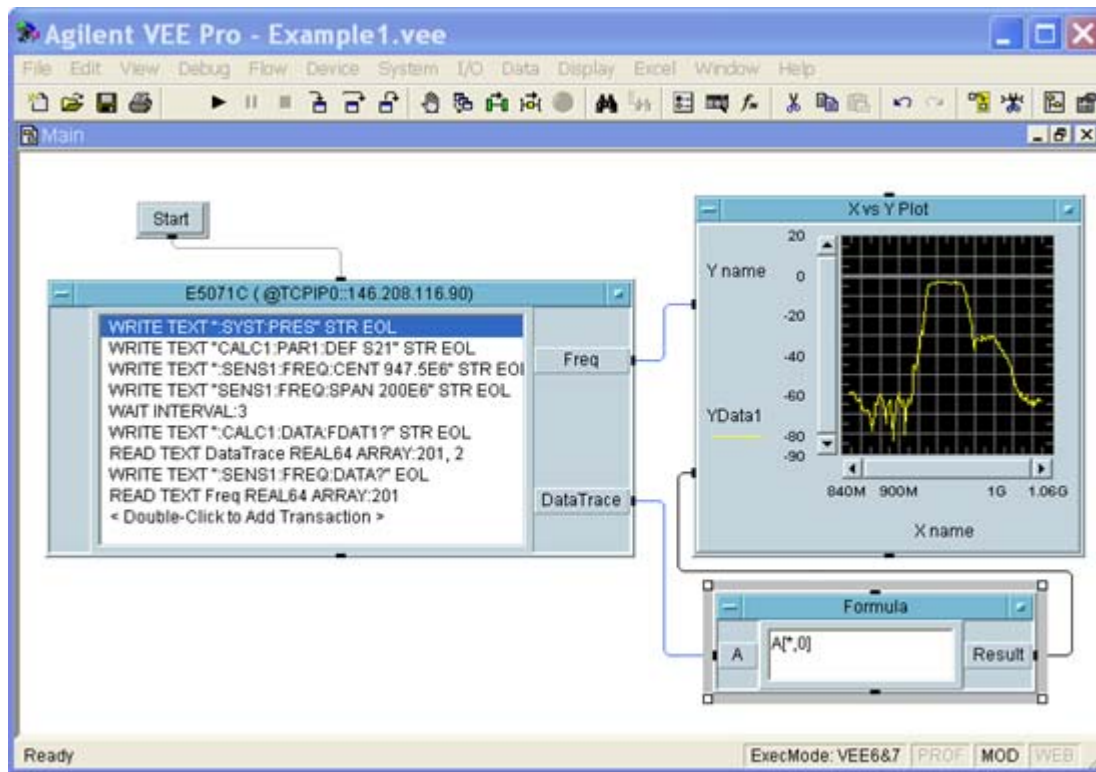
e5071c149

5. The Instrument manager displays the connection with E5071C.



e5071c150

The following figure shows an example of control using the I/O interface that has been set in the above procedure.



e5071c151

Control with Telnet Server

NOTE

Port 23 is replaced by port 5024 in revision A.11.0x and above. There is no change in port 5025.

In the control system over telnet server, communications are performed through connection between the sockets provided by the processes of the external controller and the E5071C to establish a network path between them.

A socket is an endpoint for network connection; port 23 and port 5025 are provided for the sockets for the E5071C. Port 23 is provided for conversational control using telnet (user interface program for the TELNET protocol) and port 5025 for control from a program.

NOTE

To use telnet, port 23 and 5025 should be opened through Windows firewall.

CAUTION

By opening port 23 and 5025, the E5071C can be controlled remotely using telnet. It is recommended to close port 23 and 5025 after usage from the security prospective.

Preparing the E5071C

To communicate with the external controller, follow these steps to turn on the telnet server of the E5071C in advance.

System > Misc Setup > Network Setup > Telnet Server [ON]

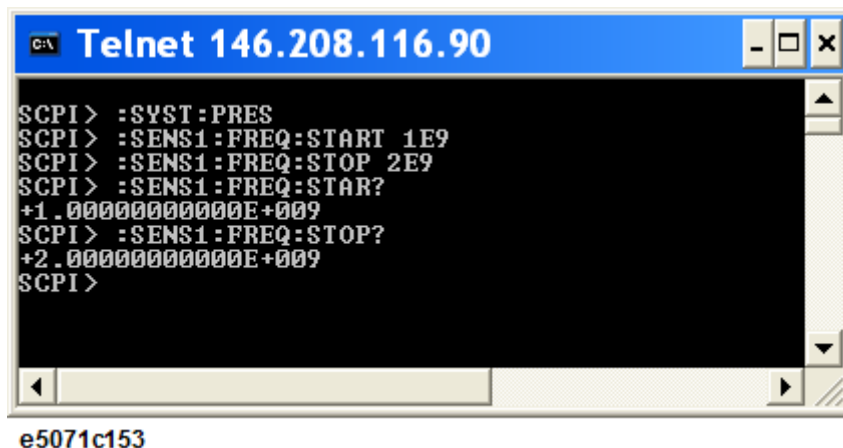
Conversational control using telnet (using port 23 for revision A.10.0x and below , and port 5024 for revision A.11.0x and above)

You can use telnet to perform conversational control by sending SCPI commands to the E5071C on a message-by-message basis. For telnet, the socket of port 23 is used for communications.

In this example, in order to show you the control procedure using telnet, you control the E5071C (IP address: 146.208.116.90 and host name: e5071c) from the external controller in the Windows environment.

1. Open the MS-DOS command prompt screen.
2. At the MS-DOS prompt, type telnet 146.208.116.90 or telnet e5071c and press the return key.
3. The telnet screen opens.
4. Type a command and press the return key; it is sent to the E5071C and executed. If you enter a command that queries some data, the query response is displayed below the line you have entered the command.
5. The following figure shows the screen after using the `:SYST:PRES` command to reset, the `:SENS{1-36}:FREQ:STAR` command and `:SENS{1-36}:FREQ:STOP` command commands to set the sweep start value and stop value to 1 GHz and 2 GHz respectively, and checking the settings.

Example of control using telnet



```

C:\> Telnet 146.208.116.90
SCPI> :SYST:PRES
SCPI> :SENS1:FREQ:START 1E9
SCPI> :SENS1:FREQ:STOP 2E9
SCPI> :SENS1:FREQ:STAR?
+1.000000000000E+009
SCPI> :SENS1:FREQ:STOP?
+2.000000000000E+009
SCPI>
e5071c153

```

6. Press] while holding down Ctrl in the telnet screen to break the connection to the E5071C. The telnet prompt appears. At the telnet prompt, type quit and press the Enter key. The connection to the E5071C breaks and telnet finishes.)

Control from a program (using port 5025)

E5071C

When controlling the E5071C from a program on the external controller, use the socket of port 5025 for connection.

NOTE

Some functions such as service requests that are available in the GPIB remote control system are not available in control over telnet server.

Control using C or Visual Basic

You can control the E5071C by socket programming using the C language in the UNIX environment, or Visual C++ or Visual Basic in the Windows environment.

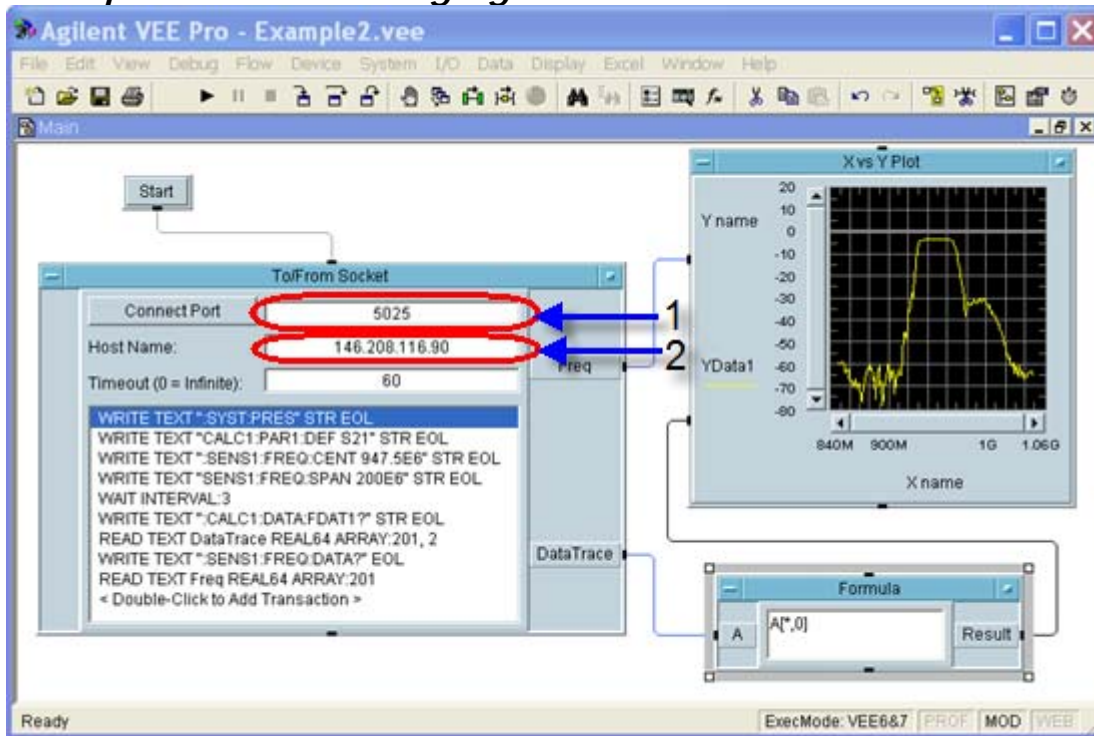
For socket programming, the library for network connection on the TCP/IP protocol is required. For the UNIX environment, BSD (Berkeley Software Distribution) Sockets API is available; for the Windows environment, WinSock (WinSock1.1 and WinSock2.0) created by porting BSD Sockets to Windows and expanding it is available.

For more information on the control method, see a sample program for control using WinSock described in "Controlling Using Telnet Server".

Control using Agilent VEE

Agilent VEE allows you to control the E5071C through the connection to the socket of port 5025 using To/From Socket. The following figure shows an example (when the IP address of the E5071C is 146.208.116.90). Enter 5025 in **Host Name** to specify the port for connection (1 in the following figure) and enter the IP address or host name of the E5071C in the field to specify the host name (2 in the following figure).

Example of control using Agilent VEE



e5071c152

USB Remote Control System

- [Overview](#)
- [System Configuration](#)

Other topics about Overview

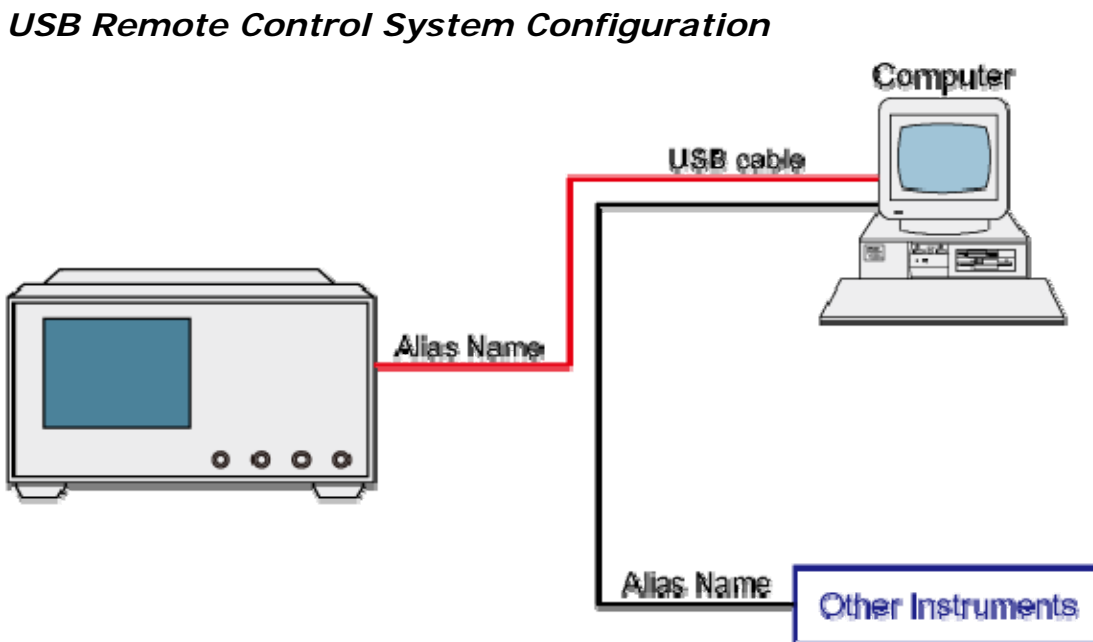
Overview

The USB (Universal Serial Bus) remote control system provides device control via USB, which is equivalent to control via GPIB. Connection is made through an interface in compliance with USBTMC-USB488 and USB 2.0.

System Configuration

The USB remote control system controls instruments that use the name "alias." There is no such address for GPIB connections.

Use a USB cable to connect the E5071C to an external controller (personal computer). The following figure shows an overview of the system configuration for the USB remote control system.



e5071c339



Required Equipment

- E5071C
- External controller (PC with USB host port (type A)).

- Other USB compatible devices (instruments and/or peripherals for specific purposes).
- USB cable connecting E5071C and external controller (with type A/4-prong male or type B/4-prong male connectors depending on device used).

USB Port Types

There are two standard types of USB ports. The external controller (PC) must be connected via the USB host port (type A), while the E5071C and other USB compatible devices must be connected via the USB interface port (type B).

Port Type	Description
	Type A: USB host port
	Type B: USB (USBTMC) interface port

Preparing E5071C

You do not have to configure any softkey or command of the E5071C in order to control the E5071C from an external controller. Simply connect a USB cable to the USB interface port.

Preparing External Controller

In order to establish communication with the E5071C via USB, you must set up the I/O interface of the external controller in advance. The USB can identify devices automatically, so once you connect a USB cable to a target device, a dialog box will appear for USB device registration.

NOTE

The E5071C will be identified as new device if its serial number has been changed.

NOTE

You must install the Agilent I/O Libraries on your PC in advance. Use Agilent I/O Libraries Suite 14.2 or later.

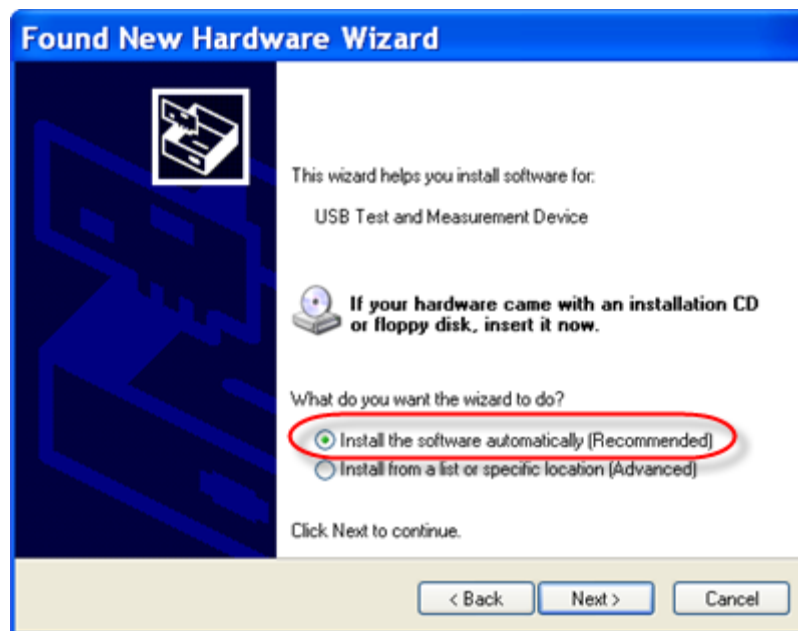
1. Setting E5071C when USB Cable Is Connected

1. When new device is connected via USB cable, the following dialog box will appear automatically. Select **No, not this time**, and then click **Next**.



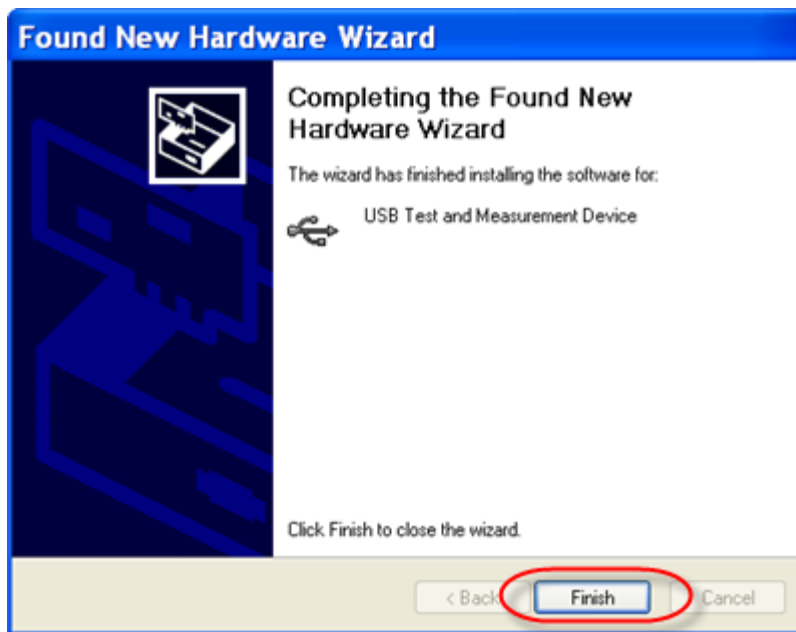
e5071c155

2. Select **Install the software automatically (Recommended)**, and then click **Next**.



e5071c156

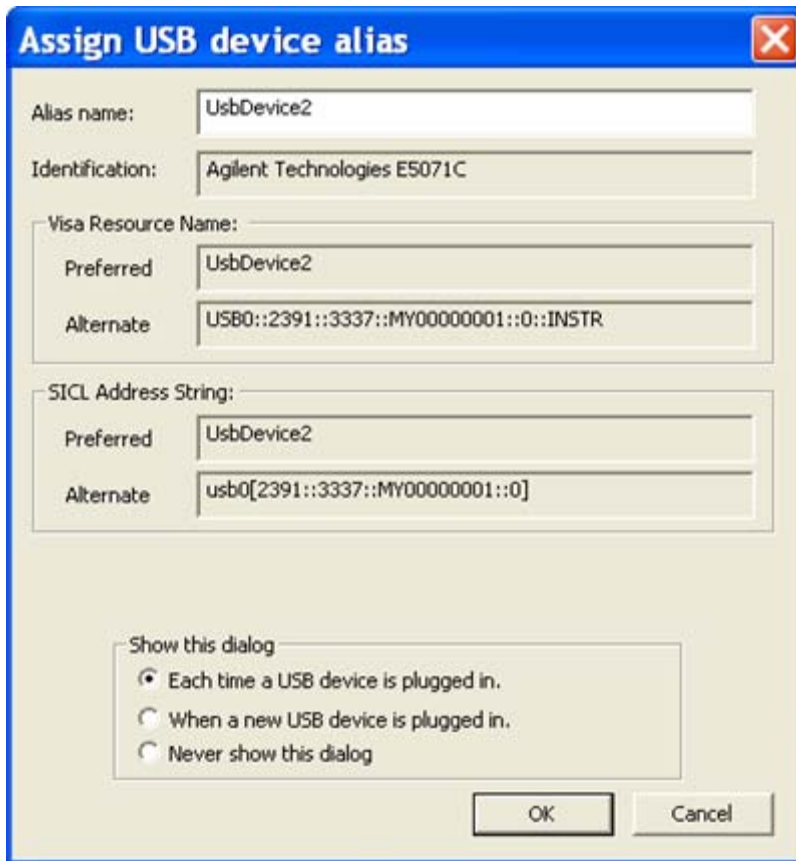
3. The drivers for E5071B are automatically installed and the completion screen appears. Click **Finish** to complete the process.



e5071c157

2. Registering Alias

Just after finishing the setting, another screen appears that can be used to change the Alias for the connected E5071C.



e5071c158

NOTE

For alias, use the ASCII format less than 127 digits. Alias is upper/lower case insensitive.

NOTE

If **Never show this dialog** is selected in **Show this dialog** frame, the dialog box does not appear even if a new device is connected.

NOTE

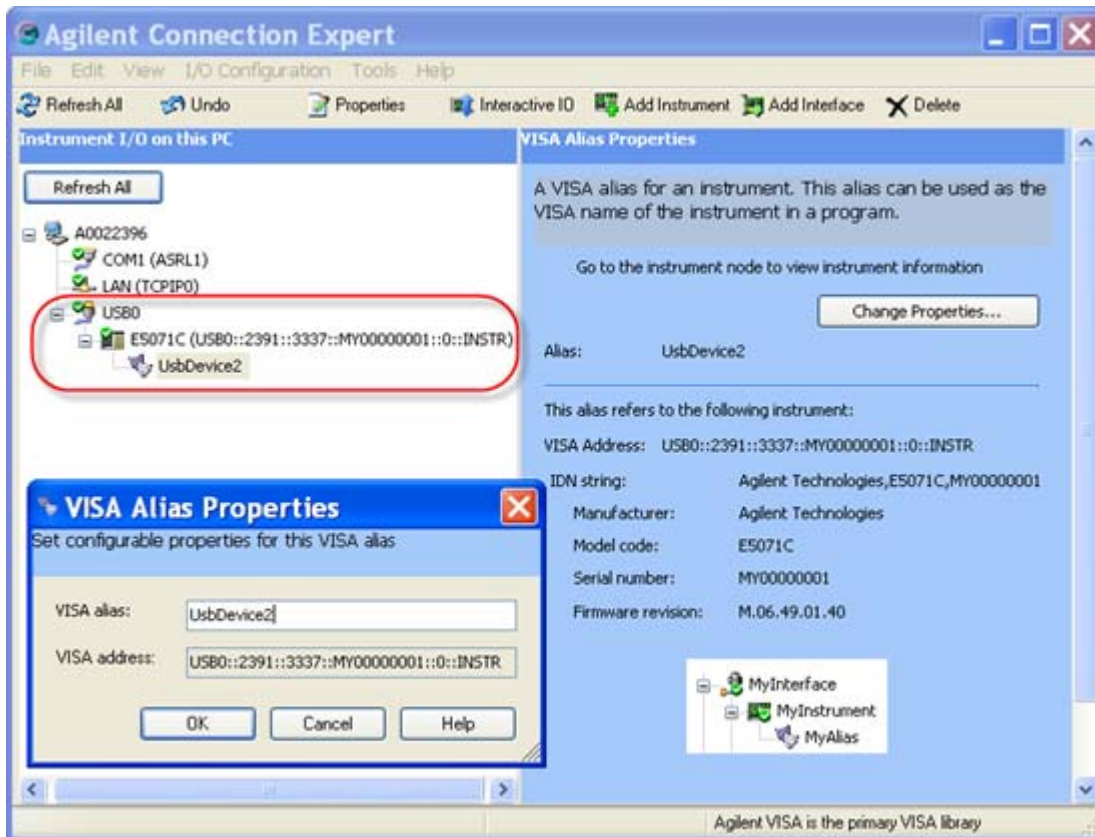
Once new device is identified, the "New Hardware Search Wizard" will start. Follow the instruction to implement the processing.

3. Changing Alias on Setting Screen

The following are steps using the Agilent I/O Libraries Suite 14.2.

1. From the Start menu of your PC, click **Programs > Agilent IO Libraries Suite > Agilent Connection Expert** to open the Config setting screen.
2. In the Config setting screen, select the alias names from **USB0** onward in the **Instrument I/O on this PC** frame, and then use the **Change Properties** from **I/O Configuration** on the menu bar.

Changing Alias



e5071c145

Control using C or Visual Basic

You can control the E5071C by programming using Visual C++ or Visual Basic in the Windows environment as well as SIDL/VISA. For further information on controlling the E5071C, see the manual of SIDL or VISA. For Agilent I/O Libraries, use Agilent I/O Libraries Suite 14.2 or later.

You may use alias in the programming using SIDL/VISA.

The following example shows an OPEN command to control the E5071C to which alias is given as ENA_USBIF.

SIDL	id = iopen("ENA_USBIF")
VISA	viOpen(...,"ENA_USBIF",...)

NOTE

For further details of the programming using SIDL/VISA, see the SIDL Users Guide or the VISA Users Guide.

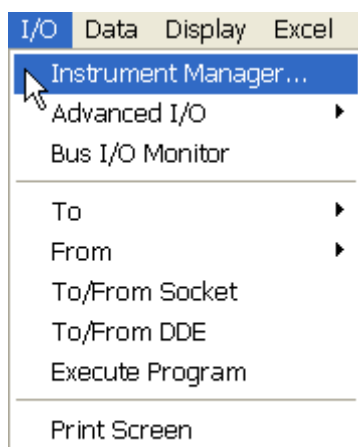
Control using Agilent VEE

Agilent VEE allows you to control the E5071C via the direct I/O interface. The following example shows how to control the E5071C to which alias is given as ENA_USBIF.

NOTE

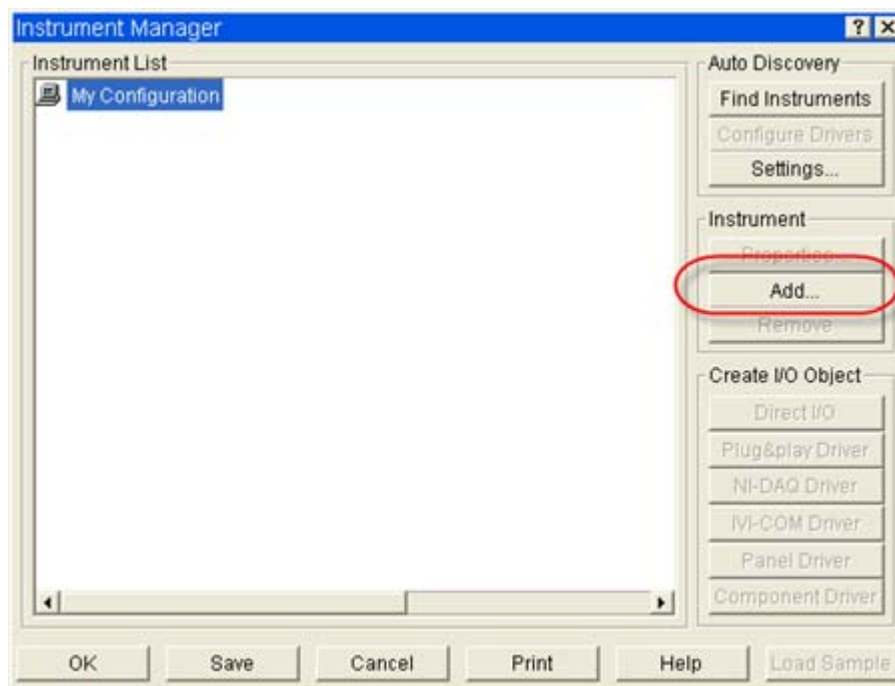
When using Agilent VEE for PC, use Agilent VEE Pro 7 for Windows or later version.

1. On the Agilent VEE's **I/O** menu, click **Instrument Manager**.



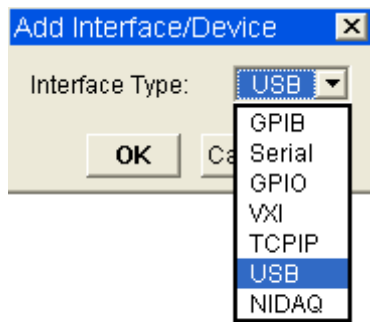
e5071c146

2. In **Instrument Manager**, click **Add...**



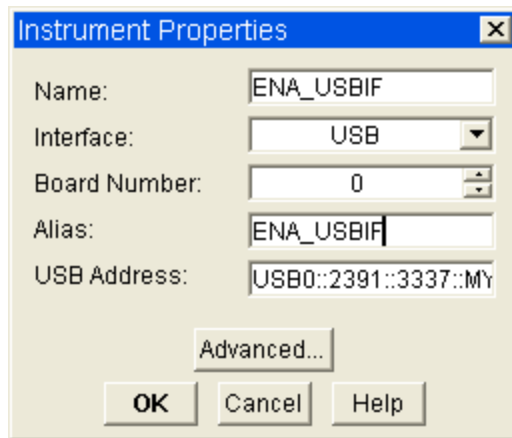
e5071c147

3. A new windows appears for the selection of Interface Type. Select **USB** and click **OK**.



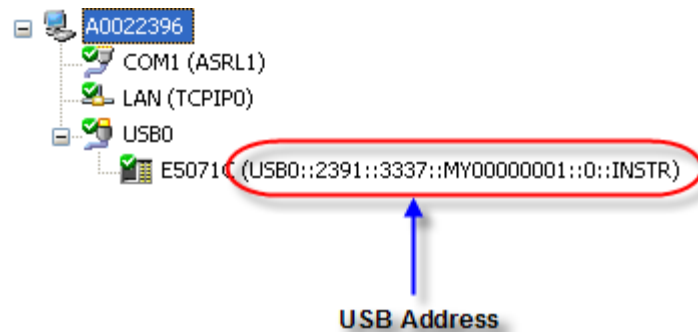
e5071c154

4. In **Instrument Properties**, type any name for the Instrument in Name (for example: ENA_USBIF or E5071C_USB), and add USB Address in the **USB Address**. Click **OK** after entering all the parameters.



e5071c181

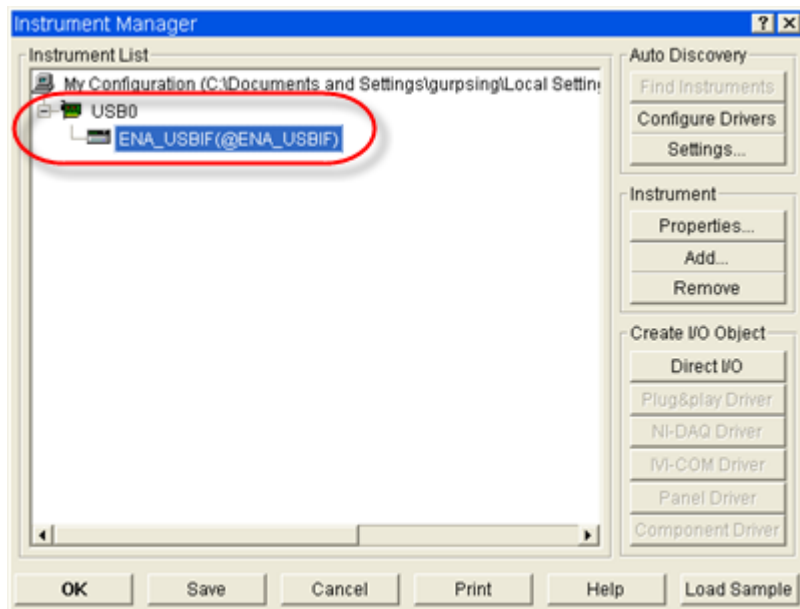
NOTE The USB address can be retrieved from Agilent Connection Expert.



e5071c159

E5071C

5. The E5071C successfully appears in the **Instrument Manager**.



e5071c182

Sending SCPI command messages

- [Type and Structure of Commands](#)
- [Grammar of Messages](#)
- [Remote Mode](#)

Other topics about Overview

Type and Structure of Commands

The SCPI commands available for the E5071C are classified into 2 groups as follows.

E5071C commands

Commands specific to the E5071C. They cover all measurement functions that the E5071C has and some general-purpose functions. The commands in this group are arranged in a hierarchical structure called the command tree. Each command consists of character strings (mnemonics) indicating each hierarchical level and colon (:) separators between hierarchical levels.

IEEE common commands

Commands to cover general-purpose functions defined in IEEE488.2 that are available commonly to instruments that support this standard. The commands in this group have an asterisk (*) at the beginning. For the commands in this group, there is no hierarchical structure.

Concepts of the command tree

The commands at the top of the command tree are called "root command" or simply "root." To access lower level commands in the tree, you need to specify a specific path like a directory path in the DOS file system. After power-on or reset, the current path is set to the root. Special characters in messages change the path setting as described below.

Message terminator

A message terminator such as the <new line> character sets the current path to the root.

Colon (:)

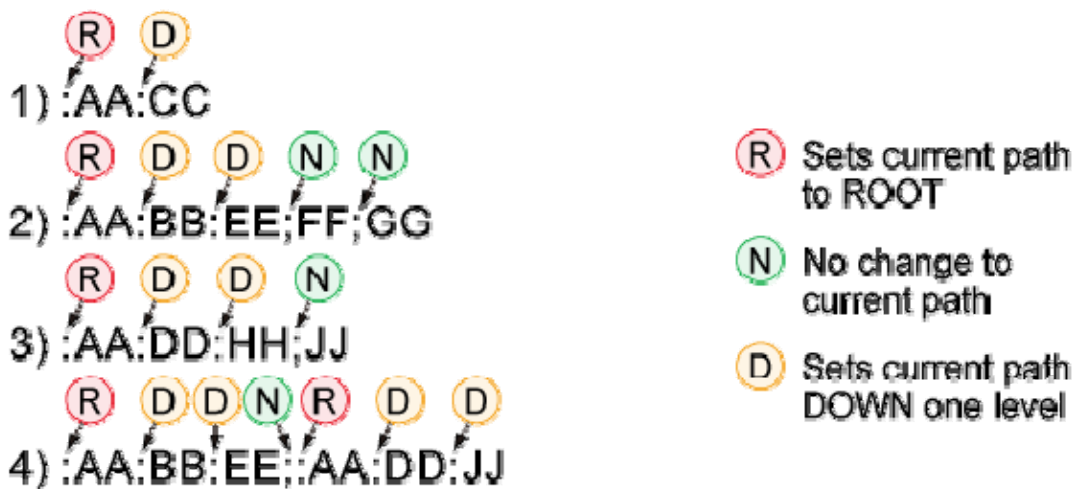
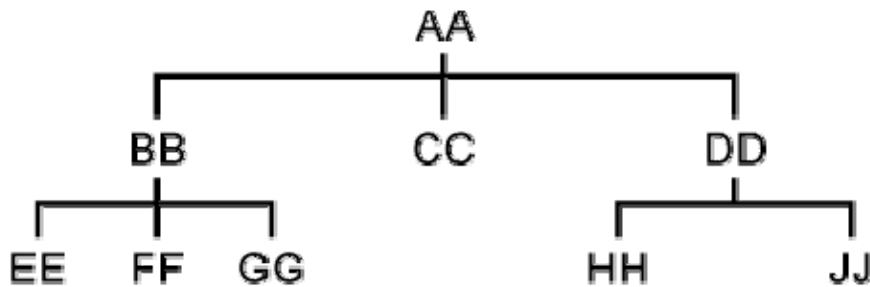
A colon between 2 command mnemonics lowers the level of the current path in the command tree. A colon used as the first character of a command specifies the command mnemonic that follows as the root-level command.

Semicolon (;)

A semicolon does not change the current path and separates 2 commands in the same message.

The following figure shows an example of how to use colons and semicolons to efficiently access commands in the command tree.

Using colons and semicolons



e5071c350

Grammar of Messages

This section describes the grammar to send program messages via GPIB. Program messages are messages that the user sends to the instrument from the external controller to control the instrument. A program message consists of 1 or more commands and their necessary parameters.

Upper/lower case sensitivity

Upper/lower case insensitive.

Program message terminator

A program message must be terminated with one of the 3 program message terminators: <new line>, <^END>, or <new line><^END>. <^END> indicates that EOI on the GPIB interface becomes active at the instant when the immediately previous data byte is sent. For example, the

OUTPUT command of HTBasic automatically sends the message terminator after the last data byte.

Parameters

A space (ASCII code: 32) is required between a command and its first parameter. When sending several parameters in a single command, separate each parameter with a comma (,).

Message including several commands

When sending 2 or more commands in a single message, separate each command with a semicolon (;). The following example shows how to send the ***CLS** command and the **:STAT:PRES** command in a single message using HTBasic.

OUTPUT 717;"*CLS;:STAT:PRES"

Remote Mode

The E5071C does not provide remote mode. Therefore, even if you send a GPIB command, it never enters into remote mode automatically. There is no local key to release remote mode.

If you need to prevent misoperation during remote control due to entry from the front panel or mouse, lock the input devices using the following commands.

- **:SYST:KLOC:KBD**
- **:SYST:KLOC:MOUS**

LXI

The E5071C is LXI-C compliant from firmware revision A.08.00 onwards.

About LXI

LXI (LAN eXtensions for Instrumentation) is the LAN-based successor to GPIB and combines the advantages of Ethernet with the simplicity and familiarity of GPIB. The key features of LXI are as follows:

- The speed, simplicity, worldwide reach, low cost, ongoing enhancement and backward compatibility of LAN.
- Quick, easy configuration through the intuitive web interface built into compliant instruments.
- Simplified programming and greater software reuse through IVI drivers.
- The ability to create hybrid systems that include LXI, GPIB, VXI, PXI, CANbus, etc.
- Enhanced system performance and event handling via hardware- and LAN-based triggering modes.
- Synchronization of local and remote instruments through the IEEE 1588 precision time protocol.
 - For more information on LXI, please visit www.lxistandard.org

Checking LXI Compliance

The E5071C having LXI compliance will show a LXI logo at the start up



screen of the firmware. If this logo does not appear at the start up screen of the firmware, it means that the system is not LXI-C compliant.

Files Required for LXI with HDD revision CN70x

The HDD revision CN70x does not come with installed Dot Net architecture which is a requirement for LXI. User having HDD revision CN70x need to install the required components for LXI on the E5071C, and then update the firmware to A.08.00 or later to use LXI-C functionality.

1. Update the E5071C with the latest Firmware revision if your firmware revision is A.07.0x.
2. LXI works only on Dot Net platform. To install dotnet framework and other software required for LXI, download the file ENALXISetup from www.agilent.com/find/ena_support, and then execute it to install all the required components for LXI on the E5071C.

Setting up Analyzer

Setting up Analyzer

- Selecting the Active Channel/Trace
- Configuring Measurement Conditions
- Configuring Display Settings
- Saving and Loading the Settings

Selecting the Active Channel/Trace

You can configure the E5071C by using various commands. Some commands require you to specify and work with a particular channel or trace, while other commands do not have this restriction.

Those commands that do not require you to specify a particular channel or trace apply to the currently active channels and traces. Therefore, before issuing such a command, you must make the appropriate channels and traces active.

To make a channel active, use the following command:

:DISP:WIND{1-160}:ACT

NOTE

Only the currently displayed channels can be active channels. Therefore, you must display the desired channels by using the **:DISP:SPL** command before making them active.

To make a trace active, use the following command:

:CALC{1-160}:PAR{1-16}:SEL

NOTE

Only the currently displayed traces can be active traces. Therefore, you must display the desired traces by using the **:CALC{1-160}:PAR:COUN** command before making them active.

If you are using E5071C revision A.9.60 and above, you can select a trace by the trace name, provided it has been defined earlier. To define a trace name, use the following command:

:CALC{1-160}:PAR{1-16}:TNAME:DATA

NOTE

If the trace name is not defined but used, the following error occurs:

'51, Specified trace does not exist'

NOTE

If the defined trace name already exist, the following error occurs:

'63, Duplicate trace name'

If you are using E5071C revision A.9.60 and above, you also have an option to select the trace directly by using **TRAC{1-16}** for all **SElected** commands. As such, you do not need to make a trace active before assigning a command to it.

For example, to activate marker 3 in trace 2, channel 1:

NOTE

In E5071C revision A.9.60 and above:

:CALC1:TRAC2:MARK3:ACT

NOTE

In E5071C revision A.9.5x and below:

[:CALC1:PAR2:SEL](#)

[:CALC1:MARK3:ACT](#)

Other topics about Setting up Analyzer

Configuring Measurement Conditions

- [Setting the Number of Traces](#)
- [Selecting Measurement Parameters](#)
- [Setting Sweep Condition \(Stimulus\)](#)
- [Configuring Averaging Settings](#)
- [Setting the System Z0](#)

Other topics about Setting up Analyzer

Setting the Number of Traces

When you set the number of traces, that setting determines the upper limit trace number; for example, if the setting is 3, traces 1 through 3 will be displayed. To set the number of traces, use the following command:

:CALC{1-36}:PAR:COUN

NOTE

Only the currently displayed traces can be active traces. Therefore, you must set the number of traces appropriately before making them active.

Selecting Measurement Parameters

To select the measurement parameter (S parameter) for each trace, use the following command:

:CALC{1-36}:PAR{1-36}:DEF

When you use the Balance-Unbalance Conversion feature, you can select the mixed mode S parameter as well. For more information, refer to Analysis Using the Fixture Simulator.

Setting Sweep Condition (Stimulus)

How you can set the sweep condition depends on the sweep type. You can choose between the following four sweep types:

- Linear sweep
- Log sweep
- Segment sweep
- Power sweep

To select one of the above sweep types, use the following command:

:SENS{1-36}:SWE:TYPE

To select the sweep mode (stepped/swept), use the following command:

:SENS{1-36}:SWE:GEN

Turning On/Off stimulus signal output

To turn on/off the stimulus signal output, use the following commands. For example, if the power output is automatically turned off due to the power trip feature, remove the cause of the over-input and turn on the stimulus signal output by executing the following command. You cannot perform measurement until you turn on the stimulus signal output.

:OUTP

Configuring linear/log sweep settings

To set the sweep range, use the following commands:

Type	Command
Start value	:SENS{1-36}:FREQ:STAR
Stop value	:SENS{1-36}:FREQ:STOP
Center value	:SENS{1-36}:FREQ:CEN
Span value	:SENS{1-36}:FREQ:SPAN

To set the number of measurement points, use the following command:

:SENS{1-36}:SWE:POIN

To set the sweep time, use the following commands:

Type	Command
Sweep time	:SENS{1-36}:SWE:TIME
Turning on/off auto setting	:SENS{1-36}:SWE:TIME:AUTO

To set the sweep delay time, use the following command:

:SENS{1-36}:SWE:DEL

To set the IF bandwidth, use the one of the following commands (both provide the same function):

:SENS{1-36}:BAND

:SENS{1-36}:BWID

Setting power level

To set the power level, use the following command:

:SOUR{1-36}:POW

To select whether to output the same power level (the set value for port 1) or a different power level for each port, use the following command:

:SOUR{1-36}:POW:PORT:COUP

:SOUR{1-36}:POW:PORT{1-4}

To set the correction of power-level attenuation so that it's proportional to the frequency (power slope feature), use the following command:

:SOUR{1-36}:POW:SLOP:STAT

:SOUR{1-36}:POW:SLOP

NOTE

If you turn on the power slope feature, the sweep mode is changed to the step mode.

Configuring segment sweep settings

When you opt to use segment sweep, you can set all items (in the segment sweep table) by using a single command:

:SENS{1-36}:SEGM:DATA

Alternatively, you can configure the segment sweep settings based on the data contained in a CSV file by issuing the following command:

:MMEM:LOAD:SEGM

Also, you can save the contents of the current segment sweep table to a file by issuing the following command:

:MMEM:STOR:SEGM

For more information on how to save and load the segment sweep table, refer to Saving and recalling the segment sweep table.

Configuring power sweep settings

To set the sweep range, use the following commands:

Type	Command
Start value	:SOUR{1-36}:POW:STAR
Stop value	:SOUR{1-36}:POW:STOP
Center value	:SOUR{1-36}:POW:CENT
Span value	:SOUR{1-36}:POW:SPAN

To set the fixed frequency (CW frequency), use the following command:

:SENS{1-36}:FREQ

To set the number of points, the sweep time, the sweep delay time, and the IF bandwidth, use the same commands as for the linear/log sweep.

Configuring Averaging Settings

To configure the averaging settings, use the following commands:

Type	Command
On/off	<code>:SENS{1-36}:AVER</code>
Averaging factor	<code>:SENS{1-36}:AVER:COUN</code>
Clear (Restart)	<code>:SENS{1-36}:AVER:CLE</code>

For averaging, normally, the instrument must be triggered according to the number of averaging; however, when the averaging trigger is turned on, sweeps for the number of averaging can be executed by a single trigger. For details on the averaging trigger, refer to Averaging Trigger Function.

Setting the System Z0

NOTE

This function is available with the Firmware revision 3.01 or greater.

To set the system characteristic impedance (Z_0), use the following command:

`:SENS:CORR:IMP`

Configuring Display Settings

- [Setting the Layout of Windows and Graphs](#)
- [Configuring Trace Display Settings](#)
- [Setting Display Color](#)

Other topics about Setting up Analyzer

Setting the Layout of Windows and Graphs

You can split the E5071C's LCD screen into multiple windows that display channel-specific result information, and the window layout can be selected from a number of variations. In addition, you can place on screen a segment sweep table or echo window, which you can use to display messages from your custom program.

Selecting the window layout (Channel Display Mode)

One window displays the results for a single channel. You cannot have a single window display the results from more than one channel. This means that setting the window layout determines the number of channels displayed on screen.

To select one of the 19 different window layouts shown in the figure below, use the following command:

:DISP:SPL

Selecting the graph layout (Trace Display Mode)

You can place a number of trace graphs in each window by selecting one of the pre-defined graph layouts. The number of graphs differs depending on your selected graph layout. If the number of graphs is equal to or larger than the number of traces (set by the **:CALC{1-36}:PAR:COUN** command), each graph always displays one trace. On the other hand, if the number of graphs is smaller than the number of traces, some of the graphs display two or more traces. Graph 1 is populated with trace 1, graph 2 with trace 2, and so on. Traces whose numbers exceed the last graph's number will populate graph 1, graph 2, and so on.

To select one of the 19 different graph layouts shown in the figure below, use the following command:

:DISP:WIND{1-36}:SPL

Maximizing a window or a trace graph

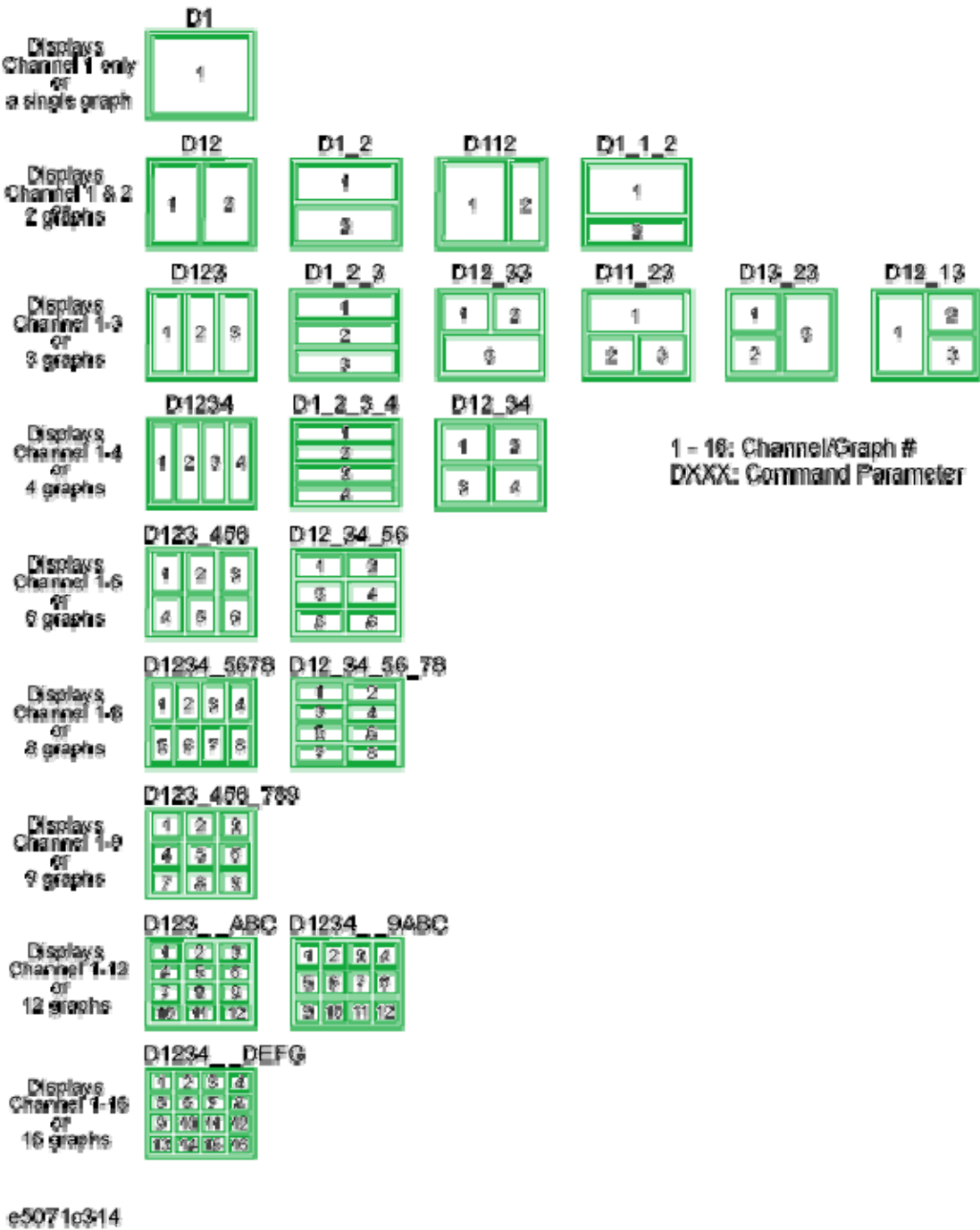
When you have multiple windows displayed, you can maximize the active channel window so that it covers the entire screen area. To maximize a window, use the following command:

`:DISP:MAX`

Similarly, when you have multiple traces displayed, you can maximize the active trace so that it extends throughout the entire window. To maximize a trace, use the following command:

`:DISP:WIND{1-36}:MAX`

Window/graph layouts and command parameters



Showing/hiding a table or echo window

You can display the following items at the bottom of the LCD screen:

- Segment sweep table
- Limit table
- Marker list table

- Echo window (a window that displays messages from a custom program)
- Loss compensation table
- Power sensor's calibration factor table

To show or hide each of the above items, use the following command:

:DISP:TABL

You cannot have two or more of the above items displayed at a time. The screen displays only the selected item by using the following command:

:DISP:TABL:TYPE

Showing/hiding softkey labels

You can show or hide the softkey labels placed alongside the right-hand edge of the LCD screen. To show or hide the softkey labels, use the following command:

:DISP:SKEY

Configuring Trace Display Settings

Selecting which traces to display

Each trace has two different representations: data and memory traces. You can show or hide the data and memory traces independently of each other. To show or hide the data or memory traces, use the following commands:

Type	Command
Data trace	:DISP:WIND{1-36}:TRAC{1-36}:STAT
Memory trace	:DISP:WIND{1-36}:TRAC{1-36}:ANN:MARK:POS:X

To copy the data trace to the memory trace, use the following command:

:CALC{1-36}:MATH:MEM

Configuring cross-trace math operations

You can perform math operations between the data and memory traces and have the results displayed as the data trace. To perform cross-trace math operations, use the following command:

:CALC{1-36}:MATH:FUNC

Configuring smoothing settings

To turn on/off smoothing, use the following command:

:CALC{1-36}:SMO

The smoothing aperture is expressed as a percentage with respect to the sweep range. To set the smoothing aperture, use the following command:

:CALC{1-36}:SMO:APER

Selecting the data format

You can select the following data formats:

- Rectangular display formats
 - Log magnitude format
 - Phase format
 - Group delay format
 - Linear magnitude format
 - SWR format
 - Real format
 - Imaginary format
 - Expanded phase format
 - Positive phase format
- Smith chart format
- Polar format

To select the measurement parameter data format, use the following command:

:CALC{1-36}:FORM

Configuring the display scale

Depending on the measurement parameter data format, you can configure the display scale in one of the following two ways:

Rectangular display formats:

When you use one of rectangular display formats (Logarithmic magnitude/Phase/ Group delay/Linear magnitude/SWR/Real/Imaginary/Expanded phase/Positive phase), you can configure the display scale by setting the following four items:

Type	Command
Number of divisions	:DISP:WIND{1-36}:Y:DIV

Scale per division	<code>:DISP:WIND{1-36}:TRAC{1-36}:Y:PDIV</code>
Reference graticule line	<code>:DISP:WIND{1-36}:TRAC{1-36}:Y:RPOS</code>
Reference graticule line value	<code>:DISP:WIND{1-36}:TRAC{1-36}:Y:RLEV</code>

NOTE

The number of divisions is a channel-wide setting (shared among all traces), while the remaining three settings are trace-specific.

You can show or hide graticule label (the label on the left-hand side of the graticule lines) by issuing the following command:

`:DISP:WIND{1-36}:LAB`

Smith chart/Polar formats:

When you are using one of Smith chart/Polar formats, you can only set the full scale value (the outermost circle's value) using the following command:

`:DISP:WIND{1-36}:TRAC{1-36}:Y:PDIV`

Auto Scale

You can use Auto Scale to automatically set the display scale. This feature works by automatically adjusting the reference division line value and the scale value per division when you are using one of the rectangular display formats or the full scale value when you are using one of Smith chart/Polar formats.

To perform Auto Scale, use the following command:

`:DISP:WIND{1-36}:TRAC{1-36}:Y:AUTO`

Displaying a message in the echo window

You can display a message in the echo window by issuing the following command:

`:DISP:ECHO`

You can clear any message displayed in the echo window by issuing the following command:

`:DISP:ECHO:CLE`

E5071C

Turning On/Off display update

To turn on/off the update of the LCD screen, use the following command:

:DISP:ENAB

Showing/hiding frequencies

To show or hide frequencies on the LCD screen, use the following command:

:DISP:ANN:FREQ

Showing or hiding the title

To show or hide the title, use the following command:

:DISP:WIND{1-36}:TITL

To define the title string that appears in the title display area, use the following command:

:DISP:WIND{1-36}:TITL:DATA

Configuring date/time display

To show or hide the current date and time on the right-hand side of the instrument status bar, use the following command:

:DISP:CLOC

To set the date and time, use the following command:

:SYST:DATE

:SYST:UPR

Turning On/Off the LCD backlight

To turn on/off the LCD backlight, use the following command (note that turning off the backlight makes the screen unreadable):

:SYST:BACK

Setting Display Color

Selecting Display Mode

You can select the one of two LCD display modes: normal display (black background) or inverted display (white background).

To select the display mode, use the following command:

:DISP:IMAG

Setting display color for each item

To set the display colors, use the following commands:

Data trace	:DISP:COL{1-2}:TRAC{1-36}:DATA
Memory trace	:DISP:COL{1-2}:TRAC{1-36}:MEM
Graph	:DISP:COL{1-2}:GRAT{1-2}
Limit test	:DISP:COL{1-2}:LIM{1-2}
Background	:DISP:COL{1-2}:BACK

Resetting display colors to factory state

You can reset the display colors in normal display and inverted display to the preset factory state.

To reset the display colors, use the following command:

:DISP:COL{1-2}:RES

Saving and Loading the Settings

You can save the settings for measurement conditions and screen display to a file along with other instrument settings, and these settings can later be loaded from the file.

Once you have saved the measurement condition and screen display settings to a file, you can later load them whenever necessary; therefore, you can quickly modify the settings loaded from a file to create new settings without having to issue many commands.

To save the current settings to a file, use the following command:

:MMEM:STOR

To load the settings from a file, use the following command:

:MMEM:LOAD

Other topics about Setting up Analyzer

Performing Calibration

Performing Calibration

- Calibration
- Power Calibration
- Receiver Calibration
- Scalar-Mixer Calibration
- Partial overwrite

Calibration

- [Overview](#)
- [Performing Calibration](#)
- [Defining Calibration Kits](#)
- [Standard Definitions](#)
- [Reading/Writing Calibration Coefficient Alone](#)
- [Clearing Calibration Data and Calibration Coefficients](#)

Other topics about Performing Calibration

Overview

You need to execute calibration to eliminate error elements related to measurement, thus allowing you to perform accurate measurement.

Performing Calibration (Obtaining calibration coefficients)

Selecting a Calibration Kit

To select a calibration kit, use the following command:

:SENS{1-160}:CORR:COLL:CKIT

Selecting a Calibration Type

The calibration coefficients are calculated based on the selected calibration type. Therefore, before you can calculate the calibration coefficients, you must select the appropriate calibration type by using one of the following commands.

Calibration type		Command
Response	OPEN	:SENS{1-160}:CORR:COLL:METH:OPEN
	SHORT	:SENS{1-160}:CORR:COLL:METH:SHOR
	THRU	:SENS{1-160}:CORR:COLL:METH:THRU
Enhanced Response		:SENS{1-160}:CORR:COLL:METH:ERES
1-Port		:SENS{1-160}:CORR:COLL:METH:SOLT1
Full 2-Port		:SENS{1-160}:CORR:COLL:METH:SOLT2
Full 3-Port		:SENS{1-160}:CORR:COLL:METH:SOLT3
Full 4-Port		:SENS{1-160}:CORR:COLL:METH:SOLT4
2-Port TRL		:SENS{1-160}:CORR:COLL:METH:TRL2

3-Port TRL	:SENS{1-160}:CORR:COLL:METH:TRL3
4-Port TRL	:SENS{1-160}:CORR:COLL:METH:TRL4

NOTE

To calculate the calibration coefficients for the simplified full 3-port and simplified full 4-port calibrations, select the full 3-port and full-4 port commands, respectively. To calculate the calibration coefficient for the simplified 3-port TRL calibration and the simplified 4-port TRL calibration, select the 3-port TRL and the 4-port TRL commands, respectively.

To check the currently selected calibration type, use the following command:

:SENS{1-160}:CORR:COLL:METH:TYPE?

Setting the trigger source for calibration

To set the trigger source for calibration, use the following command. By setting the trigger source for calibration to "System," setting the trigger source for measurement to "External," and turning on the point trigger function, it becomes possible to use "calibration for each measurement point using the external trigger."

:SENS{1-160}:CORR:TRIG:FREE

NOTE

The trigger source for calibration does not function for the calibrations of E-Cal, power, receiver, and scalar mixer.

Measuring Calibration Data

To measure the calibration data, use one of the following commands:

Calibration data items	Command
OPEN	:SENS{1-160}:CORR:COLL:OPEN
SHORT	:SENS{1-160}:CORR:COLL:SHOR
LOAD	:SENS{1-160}:CORR:COLL:LOAD
THRU	:SENS{1-160}:CORR:COLL:THRU
Isolation	:SENS{1-160}:CORR:COLL:ISOL
TRL Thru	:SENS{1-160}:CORR:COLL:TRLT
TRL Reflection	:SENS{1-160}:CORR:COLL:TRLR
TRL Line/Match	:SENS{1-160}:CORR:COLL:TRLL

NOTE

You cannot run more than one of the commands listed above at a time; if you issue another command before the currently running command completes successfully, the current command will be aborted. Therefore, when you write a program that issues multiple calibration commands in series, you should use the *OPC? command or some other means to ensure that no command is executed before the preceding command completes itself.

As shown in the table below, the data required to calculate the calibration coefficients differ depending on the selected calibration type.

Calibration type (Selected ports are enclosed in parentheses)		Data				
		OPEN	SHORT	LOAD	THRU	Isolation
Response	OPEN (a)	a	Not required	[a]	Not required	Not required
	SHORT (a)	Not required	a	[a]	Not required	Not required
	THRU (a-b)	Not required	Not required	Not required	a-b	[a-b]
Enhanced Response (a-b)		b	b	b	a-b	[a-b]
1-Port (a)		a	a	a	Not required	Not required
Full 2-Port (a-b)		a, b	a, b	a, b	a-b, b-a	[a-b], [b-a]
Full 3-Port (a-b-c)		a, b, c	a, b, c	a, b, c	a-b, b-a a-c, c-a b-c, c-b	[a-b], [b-a] [a-c], [c-a] [b-c], [c-b]
Simplified Full 3-Port						
(1-2-3)		1, 2, 3	1, 2, 3	1, 2, 3	1-2, 2-1 1-3, 3-1 [2-3], [3-2]	[1-2], [2-1] [1-3], [3-1] [2-3], [3-2]

(1-2-4)	1, 2, 4	1, 2, 4	1, 2, 4	1-2, 2-1 1-4, 4-1 [2-4], [4-2]	[1-2], [2-1] [1-4], [4-1] [2-4], [4-2]
(1-3-4)	1, 3, 4	1, 3, 4	1, 3, 4	1-3, 3-1 [1-4], [4-1] 3-4, 4-3	[1-3], [3-1] [1-4], [4-1] [3-4], [4-3]
(2-3-4)	2, 3, 4	2, 3, 4	2, 3, 4	2-3, 3-2 [2-4], [4-2] 3-4, 4-3	[2-3], [3-2] [2-4], [4-2] [3-4], [4-3]
Full 4-Port	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1-2, 2-1 1-3, 3-1 1-4, 4-1 2-3, 3-2 2-4, 4-2 3-4, 4-3	[1-2], [2-1] [1-3], [3-1] [1-4], [4-1] [2-3], [3-2] [2-4], [4-2] [3-4], [4-3]
Simplified Full 4-Port	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1-2, 2-1 1-3, 3-1 [1-4], [4-1] [2-3], [3-2] [2-4], [4-2] 3-4, 4-3	[1-2], [2-1] [1-3], [3-1] [1-4], [4-1] [2-3], [3-2] [2-4], [4-2] [2-4], [4-2] [3-4], [4-3]

In the data section in the table, the letter m (for example, 1, a) represents the measurement data at port m; m-n (for example, 1-2, a-b) represents the measurement data between response port m and stimulus port n. You can omit data enclosed in brackets.

Calculating Calibration Coefficients

To calculate the calibration coefficients, use one of the following commands:

Calibration type	Command
Response, 1/2/3/4 port	<code>:SENS{1-160}:CORR:COLL:SAVE</code>
Simplified full 3/4 port Simplified 3/4-port TRL	<code>:SENS{1-160}:CORR:COLL:SIMP:SAVE</code>

Before issuing the above commands, you must measure all required calibration data items according to your selected [calibration type](#). Calculating the calibration coefficients clears all calibration data regardless of whether they are used for the calculation. The calibration type selection is also cleared, which results in a state where no calibration type is selected.

Simplified full 3-/4-port calibration

NOTE

This function is available with Firmware revision 3.50 or greater. Note that you can execute this function from the front panel only for Firmware revision A.06.50 or greater.

The simplified full 3-/4-port calibration acquires the calibration coefficients while omitting a part of the thru measurement.

Notes on the simplified full 3-/4-port calibration

Compared to the normal full 3-/4-port calibration, the simplified full 3-/4-port calibration is more sensitive to the error that may arise when acquiring calibration data. This because the calibration coefficients are calculated without a part of the thru measurement data. Therefore, the following should be considered when measuring data for the simplified full 3-/4-port calibration.

- The standard used for measurement must match its definition value.
 - Use a standard that provides good repeatability (stability).
 - Do not omit the length of the thru when defining the standard.
 - When using a user-created standard, verify the definition value.
 - For the N connector, remember that it has two different types: male and female.
- Assure high reliability and repeatability for measurement.

- Reduce the difference in external environment (such as temperature difference) between the time when measuring calibration data and when measuring actual data.
- Set the power level of the stimulus signal sufficiently small so that compression does not occur.
- Narrow the IF bandwidth.
- Increase the averaging factor.
- Use a cable that has robust amplitude/phase characteristics against bending.
- Use high-precision connectors.

Simplified 3/4-port TRL calibration

- This function is available with Firmware revision A.06.50 or greater.
- The simplified 3/4-port TRL calibration calculates the calibration coefficient by skipping part of the thru/line measurement (or line/match measurement) that is necessary for normal 3/4-port TRL calibration.
- You need to give consideration to the same conditions for the simplified 3/4-port TRL calibration as those for the simplified full 3/4-port calibration listed above.

Turning ON/OFF Error Correction

To turn ON/OFF error correction, use the following command:

:SENS{1-160}:CORR:STAT

Also, once you have calculated the calibration coefficient using the **:SENS{1-160}:CORR:COLL:SAVE** or **:SENS{1-160}:CORR:COLL:SIMP:SAVE** command, error correction is automatically turned on.

Using ECal

An ECal (Electronic Calibration) module allows you to perform 1-/2-/3-/4-port calibration and response (THRU) calibration without having to replace the standard device.

ECal works by using the calibration kit data contained in the ECal module instead of the calibration kit data selected for the E5071C. This means that you do not have to define or select a calibration kit when using ECal.

NOTE

When two or more ECal modules are connected through the USB port, the system uses the calibration kit data of the first ECal module.

To perform ECal, use one of the following commands:

Calibration type	Command
1-Port Calibration	<code>:SENS{1-160}:CORR:COLL:ECAL :SOLT1</code>
Full 2-Port Calibration	<code>:SENS{1-160}:CORR:COLL:ECAL :SOLT2</code>
Full 3-Port Calibration	<code>:SENS{1-160}:CORR:COLL:ECAL :SOLT3</code>
Full 4-Port Calibration	<code>:SENS{1-160}:CORR:COLL:ECAL :SOLT4</code>
Enhanced Response Calibration	<code>:SENS{1-160}:CORR:COLL:ECAL :ERES</code>
Response Calibration (THRU)	<code>:SENS{1-160}:CORR:COLL:ECAL :THRU</code>

Simply issuing one of the above commands completes all of the tasks necessary for error correction, including measuring the calibration data, calculating the calibration coefficients, and running the error correction feature.

NOTE

Once you have initiated ECal, you cannot cancel the operation.

NOTE

No command entered following the initiation of ECal will be processed until ECal completes successfully. Accordingly, if you issue a command that queries some data, the system will not respond to the query until ECal is complete.

The below command was intended to turn ON/OFF the isolation measurement for performing ECal. However, as the isolation performance of ENA is better than ECal, this command no longer works. ENA ignores this command.

`:SENS{1-160}:CORR:COLL:ECAL:ISOL`

NOTE

This command takes no action and only exists to maintain backward compatibility.

To select the ECal characteristic for a user-characterized ECal, use the following command:

`:SENS{1-160}:CORR:COLL:ECAL:UCH`

ECal Auto-detect Function

The ECal module can automatically detect which port of the ECal module is connected to the E5071C test port. Turn off the auto-detect function to specify a port manually.

To turn OFF the auto-detect function, use the following command.

`:SENS:CORR:COLL:ECAL:ORI`

To turn OFF the auto-detect function and set a port manually, use the following command.

:SENS:CORR:COLL:ECAL:PATH

Checking the Applied Calibration Type

When you turn on error correction, you can check the calibration type actually applied to each trace. To check the calibration type, use the following command:

:SENS{1-160}:CORR:TYPE{1-160}?

The above command reads out the same parameter (SOLT3) for both the full 3-port and simplified full 3-port calibrations, and thus they cannot be discriminated. For the same reason, the following calibrations cannot be discriminated: full 4-port and simplified full 4-port, 3-port TRL and simplified 3-port TRL, and 4-port TRL and simplified 4-port TRL.

Defining Calibration kits

Selecting a Calibration Kit

To select a calibration kit, use the following command:

:SENS{1-160}:CORR:COLL:CKIT

Setting the Calibration Kit Name

To set the name of a calibration kit, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:LAB

Standard Definitions

Selecting a Standard Type

To select a standard type, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:TYPE

Setting the Standard Name

To set the standard name, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:LAB

Setting the Standard Value

To set the standard value, use one of the following commands:

Item	Command
C0	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:C0

C1	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C1
C2	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C2
C3	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C3
L0	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L0
L1	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L1
L2	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L2
L3	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L3
Offset Delay	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :DEL
Offset Loss	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :LOSS
Offset Z0	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :Z0
Arbitrary Impedance	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}: ARB
Start Frequency	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :FMIN
Stop Frequency	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}: FMAX
Media Type	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :CHAR
Length Type	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :LTYP

As shown in the table below, you need to set different items depending on the standard type.

Stand ard Types	C O t o C 3	L O t o L 3	Off set Del ay	Off set Los s	Off set Z0	Arbitra ry Imped ance	Min. Frequ ency	Max. Frequ ency	Conne ctor Type
OPEN	*		*	*	*		*	*	*
SHOR T		*	*	*	*		*	*	*

LOAD			*	*	*		*	*	*
THRU			*	*	*		*	*	*
Arbitra ry Imped ance			*	*	*	*	*	*	*

You need to set the items identified by * marks in the table above.

Saving/Recalling the Definition File

To save the definition of the selected calibration kit to a file, use the following command.

:MMEM:STOR:CKIT{1-20}

To recall the definition of the calibration kit from a file and set, use the following command.

:MMEM:LOAD:CKIT{1-20}

Defining a Subclass of the Standard

To set the standard type that varies with the frequency range, use the following command to specify the subclass.

:SENS{1-160}:CORR:COLL:SUBC

To select the subclass, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD

To set the start frequency of a specified subclass, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:FMIN

To set the stop frequency of a specified subclass, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:FMAX

Defining a Standard Class Assignment

To select the standard to be applied to the OPEN measurement for each port, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:OPEN

To select the standard to be applied to the SHORT measurement for each port, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:SHOR

To select the standard to be applied to the LOAD measurement for each port, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:LOAD

To select the standard to be applied to the THRU measurement between each pair of ports, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:THRU

To select the standard to be applied to the THRU measurement for the TRL calibration between each pair of ports, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD:TRLT

To select the standard to be applied to the Reflection measurement for the TRL calibration between each pair of ports, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD:TRLR

To select the standard to be applied to the Line/Match measurement for the TRL calibration between each pair of ports, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD:TRLL

Setting the Standard Media Type

To select the media type of the standard to be used, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:CHAR

Saving and loading calibration coefficients

You can save calibration coefficients to a file along with other instrument settings and then later load them from the file.

By default, the system does not save calibration coefficients when it saves instrument settings. Therefore, to save calibration coefficients, you must explicitly configure the system to save them by issuing the following command:

:MMEM:STOR:STYP

To save calibration coefficients to a file, use the following command:

:MMEM:STOR

To load calibration coefficients from a file, use the following command:

:MMEM:LOAD

For more information on how to save and load calibration coefficients, refer to Saving and recalling instrument status

Reading/Writing Calibration Coefficient Alone

The calibration coefficient alone can be read from and written to the E5071C by using the following command:

:SENS{1-160}:CORR:COEF

To write a positive calibration coefficient, use one of the following commands to declare the calibration type:

:SENS{1-160}:CORR:COEF:METH:ERES

:SENS{1-160}:CORR:COEF:METH:OPEN

:SENS{1-160}:CORR:COEF:METH:SHOR

:SENS{1-160}:CORR:COEF:METH:SOLT1

:SENS{1-160}:CORR:COEF:METH:SOLT2

:SENS{1-160}:CORR:COEF:METH:SOLT3

:SENS{1-160}:CORR:COEF:METH:SOLT4

:SENS{1-160}:CORR:COEF:METH:THRU

To validate the written calibration coefficient, use the following command:

:SENS{1-160}:CORR:COEF:SAVE

About Calibration Types and Coefficients

The following table shows the required calibration coefficients for each calibration type.

Calibration Type	Calibration Coefficient					
	ES	ER	ED	EL	ET	EX
Response calibration (OPEN)		*	*			
Response calibration (SHORT)		*	*			
Response calibration (THRU)					*	*
Enhanced response calibration	*	*	*		*	*
1-port calibration	*	*	*			
Full 2-port calibration	*	*	*	*	*	*
Full 3-port calibration	*	*	*	*	*	*
Full 4-port calibration	*	*	*	*	*	*
2-Port TRL calibration	*	*	*	*	*	
3-Port TRL calibration	*	*	*	*	*	

4-Port TRL calibration	*	*	*	*	*	
------------------------	---	---	---	---	---	--

NOTE

If either an invalid calibration coefficient is specified for the writing command or a nonexistent calibration coefficient is specified for its reading command, the following error will occur: 23, Specified error term does not exist

Procedures for Writing Calibration Coefficient

You must follow the steps below to write the calibration coefficient.

1. Declare the calibration type to write.

Execute :SENS{1-160}:CORR:COEF:METH:xxxx command

2. Write any calibration coefficient.

Execute :SENS{1-160}:CORR:COEF command as needed for the written calibration coefficients

3. Validate the calibration coefficients.

Execute :SENS{1-160}:CORR:COEF:SAVE command

NOTE

Do not execute any other command while writing the calibration coefficients. This may cause the system to function incorrectly.

NOTE

If the fixture simulator is turned on, the calibration coefficient writing will not function correctly. Make sure to turn off the fixture simulator before execution.

Clearing Calibration Data and Calibration Coefficients**Clearing Calibration Data**

When the frequency offset function has been disabled

You can use the following command to clear the measurement values of calibration data executed with :SENS{1-160}:CORR:COLL:OPEN command, etc.

:SENS{1-160}:CORR:COLL:CLE

When the frequency offset function has been enabled

You can use the following command to clear the measurement values of calibration data executed with :SENS{1-160}:CORR:OFFS:COLL:OPEN command, etc.

:SENS{1-160}:CORR:OFFS:COLL:CLE

These clear functions make the temporarily settings during the calibration, such as trace number and measurement parameters, recover to the original state.

Clearing Calibration Coefficients

When the frequency offset function has been disabled

You can use the following command to clear the calibration coefficients used.

:SENS{1-160}:CORR:CLE

When the frequency offset function has been enabled

You can use the following command to clear the calibration coefficients used.

:SENS{1-160}:CORR:OFFS:CLE

This command does not clear calibration coefficients related to normal calibration.

Power Calibration

- [Overview](#)
- [Preparation for Power Calibration](#)
- [Loss Compensation](#)
- [Turning ON/OFF Power-level Error Correction](#)

Other topics about Performing Calibration

Overview

The E5071C lets you perform power calibration by using a power meter to output a more accurate power level.

Preparation for Power Calibration

Connecting Power Meter and Power Sensor

The E5071C performs power calibration by controlling the power meter via GPIB. Therefore, you need to connect the E5071C and the power meter by using the USB/GPIB interface.

Selecting the Power Meter

Use the following command to select the type of GPIB or USB enabled power meter:

:SYST:COMM:PSEnSor

Setting GPIB Address of Power Meter

Use the following command to set the GPIB address of the power meter to the E5071C.

:SYST:COMM:GPIB:PMET:ADDR

Setting Power Sensor Calibration Factor Table

Before performing power calibration, set the reference calibration factor (the calibration factor at 50 MHz) and the calibration factor for each frequency depending on the power sensor you use.

NOTE

Depending on the combination of power meter and power sensor that you use, the power sensor calibration factor table may be set automatically by the power meter. In this case, do not set the calibration factor table with the E5071C.

To set the reference calibration factor of the power sensor, use the following commands:

Channel A	<code>:SOUR:POW:PORT:CORR:COLL:ASEN:RCF</code>
Channel B	<code>:SOUR:POW:PORT:CORR:COLL:BSEN:RCF</code>

To set the calibration factor table of the power sensor for each frequency, use the following commands:

Channel A	<code>:SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA</code>
Channel B	<code>:SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA</code>

Alternatively, you can configure the power sensor calibration factor table based on the data contained in a CSV file by issuing the following commands:

Channel A	<code>:MMEM:LOAD:ASCF</code>
Channel B	<code>:MMEM:LOAD:BSCF</code>

Also, you can save the contents of the current power sensor calibration factor table to a file by issuing the following commands:

Channel A	<code>:MMEM:STOR:ASCF</code>
Channel B	<code>:MMEM:STOR:BSCF</code>

For more information on saving/recalling the power sensor calibration factor table, refer to Saving/recalling Power Sensor Calibration Factor Table.

Loss Compensation

You can perform loss compensation by setting in advance the power loss due to the difference in connections (cables, adapters, etc.) between the when the power calibration data are measured and when the actual DUTs are measured.

Setting Loss Compensation Table

To set the loss compensation table for each frequency, use the following command:

`:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS:DATA`

Alternatively, you can configure the loss compensation table based on the data contained in a CSV file by issuing the following command:

```
:MMEM:LOAD:PLOS{1-4}
```

Also, you can save the contents of the current loss compensation table to a file by issuing the following command:

```
:MMEM:STOR:PLOS{1-4}
```

Turning ON/OFF Loss Compensation

To turn on or off the loss compensation, use the following command:

```
:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS
```

Measuring power calibration data

NOTE

Before measuring power calibration data, you need to execute zero adjustment and calibration of the power sensor. For information on how to perform these procedures, refer to the operation manual of the power meter you are using.

NOTE

When using the power sensor calibration factor table of the E5071C, set the reference calibration factor of the power meter to 100% when calibrating the power sensor.

To set the number of power calibration data measurements at one measurement point (averaging factor), use the following command:

```
:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:AVER
```

To set a tolerance for power calibration, use the following command:

```
:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:NTOL
```

NOTE

When a tolerance for power calibration is set, if the measured value does not fall within the tolerance, even after measurement is performed during power calibration the number of times specified by the averaging factor, an error message is displayed and the power calibration is aborted. In this case, the power correction is not turned on.

To measure the power calibration data, use the following command. When the measurement is complete, the power calibration is automatically turned on.

```
:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL
```

NOTE

If you issue another command during the measurement of the power calibration data by the above command, the measurement may be aborted. Therefore, when you write a program that issues multiple calibration commands in series, you should use the *OPC or some other means to

ensure that no command is executed before the preceding command completes itself.

Turning ON/OFF Power-level Error Correction

To turn on or off the power-level error correction, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR

When the power calibration data measurement initiated with the **:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL** command is complete, the power calibration is automatically turned on.

Reading/writing power calibration data array

The power calibration data array contains data used to perform error correction for the power level at each measurement point (values obtained by subtracting the value actually measured with the power meter from the set power level value when measuring power calibration data at each measurement point).

The number of power calibration data arrays that are assigned to individual ports of individual channels can be up to 64 (16x4). To read/write one of the power calibration data arrays, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR:DATA

Receiver Calibration

- [Overview](#)
- [Measurement of Receiver Calibration Data & Calculation of Calibration Coefficient](#)
- [Turning ON/OFF Receiver Error Correction](#)

Other topics about Performing Calibration

Overview

The E5071C has a receiver calibration capability to calibrate the gain for each receiver in an absolute measurement.

Measurement of Receiver Calibration Data & Calculation of Calibration Coefficient

NOTE

Before starting a measurement of receiver calibration data, you must connect a THRU between the source port at which power calibration was applied and the receiver port on which you want to implement receiver calibration.

Use the following command for the measurement of receiver calibration data and calculation of calibration coefficient. The value following REC signifies a receiver port number, and the value given as an argument is a source port number. Specifying the same port to both the receiver port and source port will cause an error.

`:SENS{1-160}:CORR:REC{1-4}:COLL:ACQ`

NOTE

From Firmware rev 9.2, you can calibrate the receiver and transmitter port independently.

Use the following command to calibrate the receiver port:

`:SENS{1-160}:CORR:REC{1-4}:COLL:RCH:ACQ`

Use the following command to calibrate the transmitter port:

`:SENS{1-160}:CORR:REC{1-4}:COLL:TCH:ACQ`

Once the measurement is completed, calculation of the calibration coefficient takes place automatically, turning on receiver error correction automatically.

NOTE

The power calibration information on both the receiver port and source port is used to calculate calibration coefficients. The accuracy of receiver calibration will increase if power calibration is implemented for both the receiver port and the source port before starting receiver calibration. For information on power calibration, refer to Power Calibration.

NOTE

Verify the frequency of each port before starting a frequency offset sweep. Since this command does not change

the frequency setting, you cannot expect a correct measurement result unless the frequency is the same for both the receiver port and the source port.

NOTE

When using power calibration in combination with receiver calibration, you must implement it so that it sufficiently covers the frequencies of both ports or implement it twice, before and after the receiver calibration.

Turning ON/OFF Receiver Error Correction

To turn on or off the receiver error correction, use the following command:

:SENS{1-36}:CORR:REC{1-4}

NOTE

Once the measurement of receiver calibration data is completed with the **:SENS{1-36}:CORR:REC{1-4}:COLL:ACQ** command, receiver error correction is automatically turned on.

Scalar-Mixer Calibration

- [Overview](#)
- [Measurement of Scalar-Mixer Calibration Data](#)

Other topics about Performing Calibration

Overview

The E5071C's frequency offset function allows you to measure any device that has different input and output frequencies. To correct such measurements, you must determine the transmission frequency characteristics at different frequencies.

Scalar-mixer calibration is a method to calculate the transmission frequency characteristics at different frequencies by using a power meter.

Measurement of Scalar-Mixer Calibration Data

Setting Frequency Offset Function

You can use the following command to enable the frequency offset function before starting a measurement of scalar-mixer calibration data.

[:SENS{1-36}:OFFS](#)

NOTE

To use scalar-mixer calibration, the frequency offset function must be enabled.

Selecting Calibration Port and Type

You can use the following commands to set the port on which a measurement of scalar-mixer calibration data is implemented and the type of calibration.

- **[:SENS{1-36}:CORR:OFFS:COLL:METH:SMIX2](#)**
- **[:SENS{1-36}:CORR:OFFS:COLL:METH:SOLT1](#)**

"SMIX2" will set a scalar-mixer calibration for 2-port forward, reverse and both directions. "SOLT1" will set 1-port calibration.

NOTE

The two ports specified by "SMIX2" must be different from each other.

The setting of forward, reverse and both directions is not dependent on the order of the two specified ports but determined by specifying an appropriate command for calibration data measurement, such as

[:SENS{1-36}:CORR:OFFS:COLL:OPEN](#)

Measuring the Calibration Data

You can use the following command to measure the scalar-mixer calibration data.

- `:SENS{1-36}:CORR:OFFS:COLL:OPEN`
- `:SENS{1-36}:CORR:OFFS:COLL:SHOR`
- `:SENS{1-36}:CORR:OFFS:COLL:LOAD`
- `:SENS{1-36}:CORR:OFFS:COLL:THRU`

Measuring Power

You can use the following command to measure power.

`:SENS{1-36}:CORR:OFFS:COLL:PMET`

NOTE

Power measurement is required for scalar-mixer calibration, but it is not required for 1-port calibration. The setting for power measurement is the same as for the normal power calibration. For information on power calibration, refer to Power Calibration.

Calculation of scalar-mixer calibration coefficient

If the needed measurement is completed for the calibration port and type you have selected, use the following command to start calculation of calibration coefficients.

`:SENS{1-36}:CORR:OFFS:COLL:SAVE`

NOTE

If the necessary measurement is not completed, an error will occur. Once the calculation of calibration coefficients is completed, the measurement values are cleared despite whether they were used and error correction is turned on automatically.

Turning ON/OFF scalar-mixer error correction

To turn on or off scalar-mixer error correction, use the following command. This command is commonly used for the normal calibration; scalar-mixer calibration data are enabled while the frequency offset is turned on, and the normal calibration data are enabled while it is turned off.

`:SENS{1-36}:CORR:STAT`

NOTE

Once the scalar-mixer calibration coefficient has been calculated, scalar-mixer correction is automatically turned on.

Implementing scalar-mixer calibration with ECal

An ECal (Electronic Calibration) module allows you to perform scalar-mixer calibration and 1-port calibration without needing to replace the standard device.

To perform ECal, use one of the following commands:

Calibration type	Command
1-Port Calibration	:SENS{1-36}:CORR:OFFS:COLL:ECAL:SOLT1
Scalar-Mixer Calibration	:SENS{1-36}:CORR:OFFS:COLL:ECAL:SMIX2

Simply issuing one of the above commands completes some of the tasks necessary for error correction, including measuring the calibration data, calculating the calibration coefficients, and running the error correction feature. To implement scalar-mixer calibration, you must measure the power in advance. For 1-port calibration, you do not need to do this.

NOTE

Once ECal has started, you cannot interrupt the operation.

NOTE

Any command received after ECal has started will not be executed until ECal is completed. In other words, even if you issue a command with a Query, you cannot get the Query response until ECal is finished.

Partial Overwrite

- [Overview](#)
- Executing_calculation_of_calibration_coefficients_using_partial_overwrite

Other topics about Performing Calibration

Overview

The E5071C has the following calibration coefficients for full N-port calibration: Er, Es, Ed (reflection), Et (transmission), and Ex (isolation). The partial overwrite function is used to measure some of these calibration coefficients after completion of the initial calibration and then to overwrite them.

The conditions under which the calibration coefficients can be calculated by the partial overwrite are as follows:

- Calibration is completed once and valid (status other than C? or C!)
- One or more measurements for re-calculation are performed.
 - The isolation calibration coefficient, Ex, cannot be returned to the initial value, 0, once it is calculated.

If calculation of the calibration coefficients is attempted without the measurements required to execute the partial overwrite, an error message (20: Additional Standard Needed) is displayed.

Executing calculation of calibration coefficients using partial overwrite

To calculate the calibration coefficients using partial overwrite, use the following command:

:SENS{1-36}:CORR:COLL:PART:SAVE

NOTE

Before you can calculate the calibration coefficients with the partial overwrite, you must select the appropriate calibration type in the same way used for normal calibration. If calculation of the calibration coefficients is attempted without selecting the calibration type, an error message (28: Invalid Calibration Method) is displayed.

Making Measurement

Making Measurement

- Trigger System
- Starting a Measurement Cycle (triggering the instrument)
- Waiting for the End of Measurement
- Point Trigger Function
- Averaging Trigger Function

Trigger System

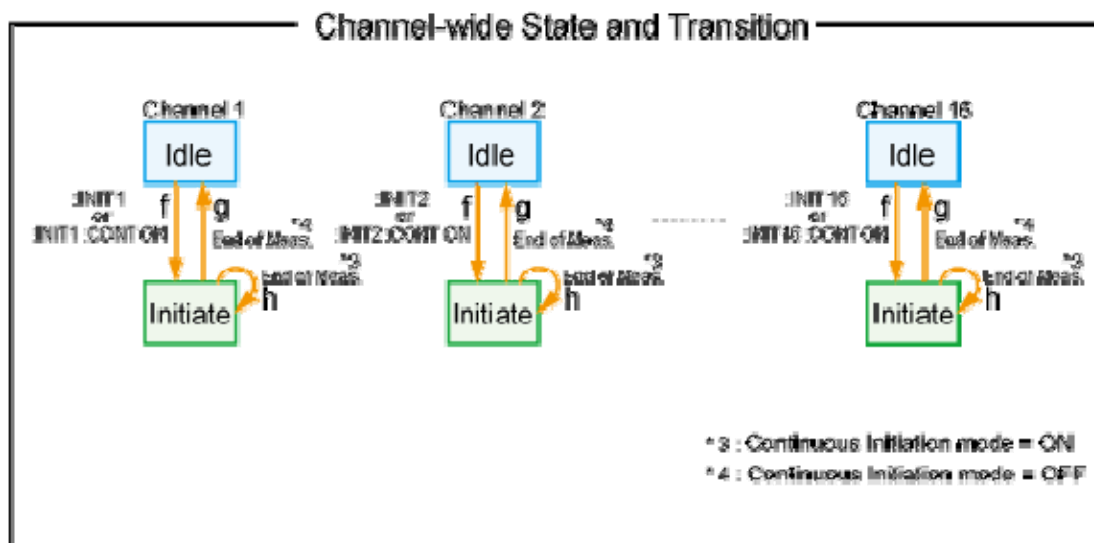
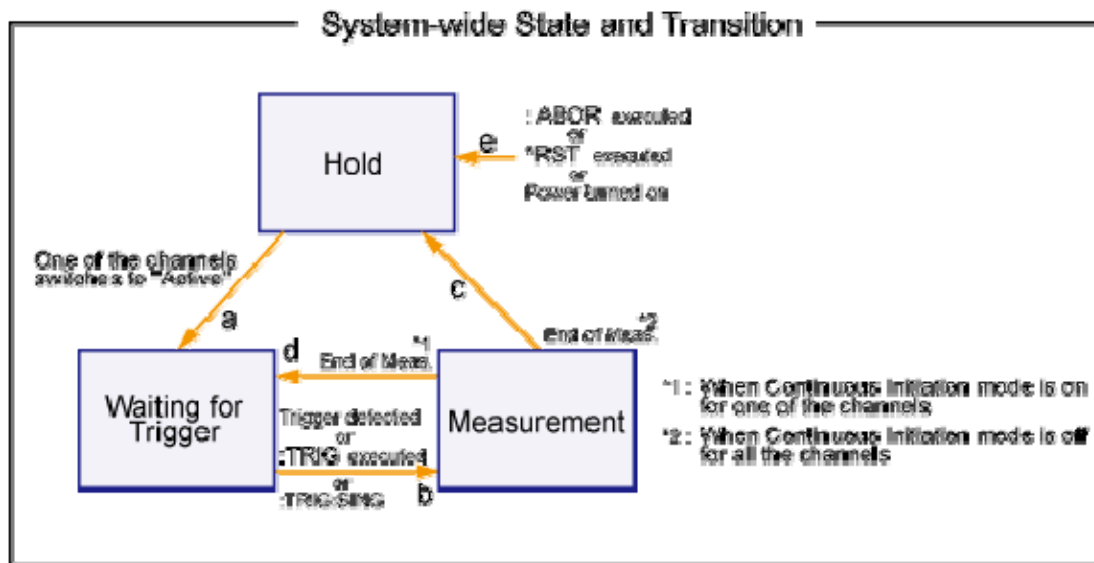
- [Overview](#)
- [System-Wide States and Transitions](#)
- [Channel-wide States and Transitions](#)

Other topics about Making Measurement

Overview

The trigger system is responsible for such tasks as detecting the start of a measurement cycle (triggering) and enabling/disabling measurement on each channel. As shown the following figure, the trigger system has two types of states: system-wide and channel-wide. The system-wide state can be "Hold", "Waiting for Trigger", or "Measurement", while the channel-wide state can be "Idle" or "Initiate".

Trigger system



e5071c312

The following subsections describe each state and explains how the trigger system switches among the states.

System-Wide States and Transitions

"Hold" State

The trigger system switches to "Hold" state when one of the following commands has been executed (arrow "e" in the above figure). Also, turning on the power to the instrument puts the trigger system into "Hold" state. When the power is turned on, however, continuous initiation mode is on for channel 1 and the trigger source is set to "Internal"; accordingly, the trigger system immediately switches to "Waiting for Trigger" state and subsequently repeats transitions between "Measurement" and "Waiting for Trigger" states.

- **:ABOR**
- ***RST**

When the trigger system is in "Hold" state and one of the channels switches to "Initiate" state (arrow "f" in the above figure), the trigger system switches to "Waiting for Trigger" state (arrow "a" in the above figure).

"Waiting for Trigger" State

When the trigger system is in "Waiting for Trigger" state and either the instrument is triggered (i.e., a trigger is detected) or one of the following commands is executed, the trigger system switches to "Measurement" state (arrow "b" in the above figure)

- **:TRIG**
- **:TRIG:SING**

As shown in the table below, the instrument is triggered differently depending on which trigger source is specified. To specify the trigger source, use the :TRIG:SOUR command.

Trigger Source	How instrument is triggered
Internal trigger	The instrument is automatically triggered within itself.
External trigger	The instrument is triggered when a trigger signal is input through the Ext Trig terminal or the handler interface
Bus trigger	The instrument is triggered when the *TRG command is issued.
Manual trigger	The instrument is triggered when you press Trigger > Trigger on the front panel.

"Measurement" State

In "Measurement" state, the instrument waits for the elapse of the sweep delay time (set by the :SENS{1-36}:SWE:DEL) and then starts a measurement cycle; this process is performed sequentially on each of those channels that were in "Initiate" state immediately before the transition to this state, in ascending order of channel number.

When the instrument has finished measuring all of the active channels, the trigger system behaves in one of the following ways depending on the setting of the continuous initiation mode.

If continuous initiation mode is off for all channels:

The trigger system switches to "Hold" state (arrow "c" in the above figure).
If continuous initiation mode is on for one of the channels:

The trigger system switches to "Waiting for Trigger" state (arrow "d" in the above figure).

Channel-wide States and Transitions

"Idle" State

A channel switches to "Initiate" state when one of the following commands has been executed (arrow "f" in the above figure).

- `:INIT{1-36}`
- `:INIT{1-36}:CONT("ON" specified)`

"Initiate" State

A channel in this state is measured just before the entire system switches to "Measurement" state.

When the instrument has finished measuring a channel, the channel behaves in one of the following ways depending on the setting of the continuous initiation mode (set by the `:INIT{1-36}:CONT`).

If continuous initiation mode is off:

The channel switches to "Idle" state (arrow "g" in the above figure).

If continuous initiation mode is on:

The channel remains in "Initiate" state (arrow "h" in the above figure).

Starting a Measurement Cycle (Triggering the Instrument)

- [Configuring the Instrument](#)
- [Starting Measurement on Demand](#)

Other topics about Making Measurement

Configuring the Instrument to Automatically Perform Continuous Measurement

1. Use the `:INIT{1-36}:CONT` command to turn on continuous initiation mode for the channels you want to measure and turn the mode off for any other channel.
2. Issue the `:TRIG:SOUR` command to set the trigger source to Internal trigger.

Starting Measurement on Demand

1. Use the `:INIT{1-36}:CONT` command to turn on continuous initiation mode for the channels you want to measure and turn the mode off for any other channel.
2. Issue the `:TRIG:SOUR` command to set the trigger source to "Bus Trigger".
3. Trigger the instrument whenever you want to perform measurement. An external controller can trigger the instrument by using one of the following three commands:

Command	Can *OPC? command be used to wait for end of sweep?	Applicable trigger source
<code>*TRG</code>	No	Bus trigger only
<code>:TRIG</code>		External trigger Bus trigger Manual trigger
<code>:TRIG:SING</code>	Yes	

4. Repeat step 3 to start the next measurement cycle.

Waiting for the End of Measurement

- [Using the Status Register](#)
- [Using :TRIG:SING Command](#)
- Using Wait Time

Other topics about Making Measurement

Using the Status Register

The status of the E5071C can be detected through the status registers. This section explains how to detect the end of measurement by using the status registers.

Measurement status is reported by the operation status condition register. An SRQ (service request) is useful when creating a program that uses the information reported by this register to detect the end of measurement.

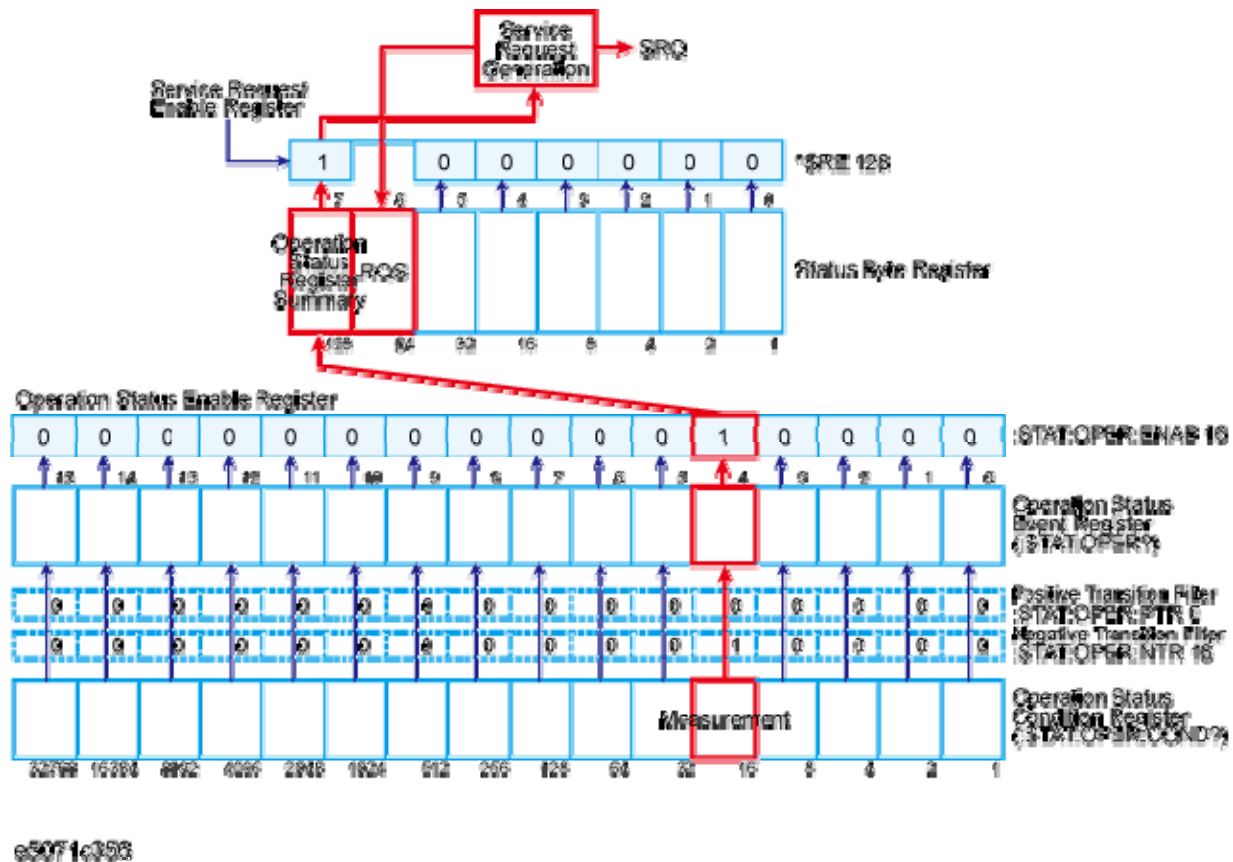
To detect the end of measurement via an SRQ, use one of the following commands:

- [*SRE](#)
- [:STAT:OPER:ENAB](#)
- [:STAT:OPER:PTR](#)
- [:STAT:OPER:NTR](#)

Follow these steps to utilize an SRQ:

1. Configure the E5071C so that it generates an SRQ when the operation status condition register's bit 4 (a bit that is set to 1 during measurement) is changed from 1 to 0.
2. Trigger the instrument to start a measurement cycle.
3. When an SRQ is generated, the program interrupts the measurement cycle.

SRQ generation sequence (at end of measurement)



Sample Program

See the Waiting for Trigger (SRQ).

Using :TRIG:SING Command

When you trigger the instrument by issuing the **:TRIG:SING** command, you can use the ***OPC** command to wait until the measurement cycle is completed.

Sample Program

See the Waiting for Trigger (OPC?).

Using Wait Time

Before creating your program, actually measure the time between the start and end of the measurement cycle. Then code your program so that the controller waits for the actually measured time by using the appropriate command (for example, the WAIT command for HTBasic). This is a straightforward method, but care must be taken: an incorrect wait time could result in an unexpected error.

Point Trigger Function

- [Point Trigger](#)
- [Switching Mode](#)

Other topics about Making Measurement

Point Trigger

Point trigger is a method used to measure a single point each time the instrument is triggered. You can perform this type of measurement by turning ON/OFF the point trigger function.

Switching Mode

You can use the following command to toggle between the point trigger's ON/OFF status.

:TRIG:POIN

NOTE

If the trigger source is internal, the point trigger function does not work even if it is specified to ON.

Low-latency external trigger

When the trigger source is an external trigger and the point trigger function is on, the low-latency external trigger function is available. The low-latency external trigger function is used to set the external trigger delay time each time a trigger for a measurement point is given.

The setting of the low-latency external trigger function is valid for all channels.

Setting the low-latency external trigger

To toggle ON/OFF the low-latency external trigger function, use the following command:

:TRIG:EXT:LLAT

If the trigger source is not an external trigger, the low-latency external trigger function does not work even if it is specified to ON, nor does it work when the point trigger function is off.

To set the external trigger delay time, use the following command:

:TRIG:EXT:DEL

Averaging Trigger Function

- [Averaging Trigger](#)
- [Toggling Mode](#)

Other topics about Making Measurement

Averaging Trigger

The averaging trigger function is used to execute the number of sweeps specified as the averaging factor with a single trigger when the sweep averaging function is used. When a trigger is given, the sweep count is cleared, the sweep is executed the number of times specified by the averaging count, and then the instrument goes into Hold status.

The setting of the averaging trigger function is valid for all channels. Note that you can set the sweep averaging function for each channel.

Toggling Mode

To toggle ON/OFF the averaging trigger function, use the following command:

:TRIG:AVER

NOTE

When the point trigger function is on, its setting has priority and the setting of the averaging trigger is ignored. More specifically, the required number of triggers is: number of measurement points in a single sweep x averaging factor.

NOTE

When the averaging function is off for the active channel, the averaging trigger function does not work for the channel. When a trigger is given, the sweep is executed only once, and then the instrument goes into Hold status.

NOTE

OPC? command can be used to detect end of measurement. To learn more, see SCPI.IEEE4882.OPC and Waiting for Trigger (OPC?).

Reading-Writing Measurement Data

Reading-Writing Measurement Data

- Data Transfer Format
- Internal Data Processing
- Retrieving Measurement Results
- Entering Data into a Trace

Data Transfer Format

- [Overview](#)
- [ASCII Transfer Format](#)
- [Integer Format](#)
- [Floating-Point Number Format](#)
- [Binary Transfer Format](#)

Other topics about Reading-Writing Measurement Data

Overview

When you transfer data using the one of the following commands, you can choose among ASCII transfer format, IEEE 64-bit floating point binary transfer format and IEEE 32-bit floating point binary transfer format.

NOTE

The instrument always uses the ASCII transfer format when you transfer data without using any of the following commands:

- [:CALC{1-160}:BLIM:REP?](#)
- [:CALC{1-160}:DATA:FDAT](#)
- [:CALC{1-160}:DATA:FMEM](#)
- [:CALC{1-160}:DATA:SDAT](#)
- [:CALC{1-160}:DATA:SMEM](#)
- [:CALC{1-160}:FUNC:DATA?](#)
- [:CALC{1-160}:LIM:DATA](#)
- [:CALC{1-160}:LIM:REP?](#)
- [:CALC{1-160}:LIM:REP:ALL?](#)
- [:CALC{1-160}:RLIM:DATA](#)
- [:CALC{1-160}:RLIM:REP?](#)
- [:SENS{1-160}:CORR:COEF?](#)
- [:SENS{1-160}:FREQ:DATA?](#)
- [:SENS{1-160}:SEGM:DATA](#)
- [:SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA](#)
- [:SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA](#)
- [:SOUR{1-160}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS](#)
- [:SOUR{1-160}:POW:PORT{1-4}:CORR:DATA](#)

To set the data transfer format, use the following command:

[:FORM:DATA](#)

NOTE

Executing the :SYST:PRES or *RST does not affect the current setting of the data transfer format.

ASCII Transfer Format

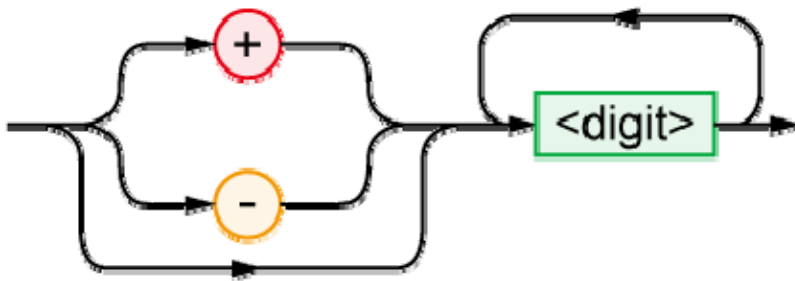
When you select the ASCII transfer format as the data transfer format, numbers are transferred as ASCII bytes, each of which corresponds to one of the formats shown below. Note that numbers are separated from one another with a comma (,) in accordance with the IEEE 488.2 specification.

NOTE

Numeric data strings vary in length. Keep this in mind when you extract some data from retrieved numeric data strings in your program.

Integer Format

The figure below shows this format. Numbers are expressed as integers. For example, 201 is expressed as "+201" or "201."

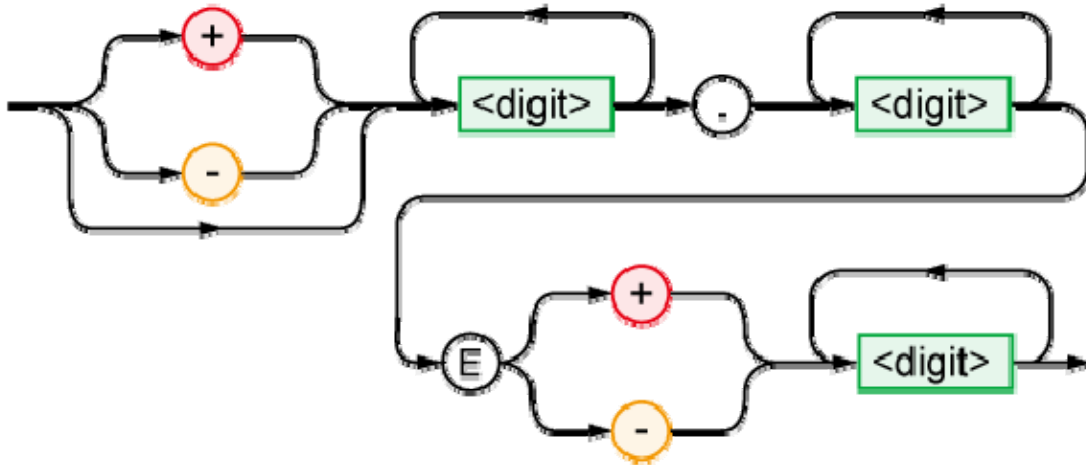
Integer format

e5071c351

Floating-Point Number Format

The figure below shows this format. Numbers are expressed with floating points. For example, 1000 is expressed as "+1.00000000000E+003."

Floating-point number format



e5071c352

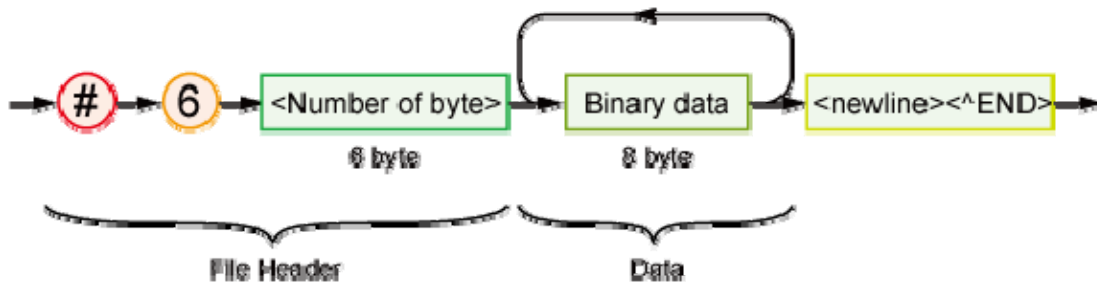
Binary Transfer Format

You can select the binary transfer format from the IEEE 64-bit floating point format or the IEEE 32-bit floating point format depending on the controller you use.

IEEE 64-bit floating point format

When you select the IEEE 64-bit floating point binary transfer format as the data transfer format, numbers are transferred in the format shown in the figure below.

Binary transfer format

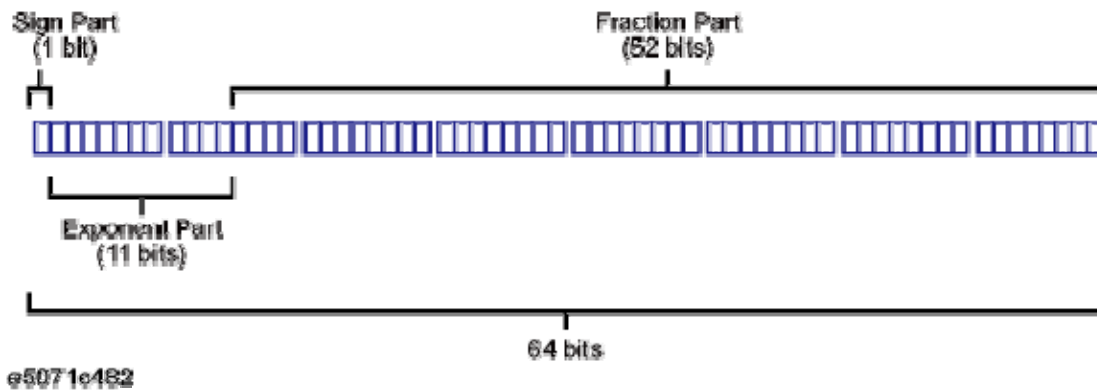


e5071c480

This data transfer format uses a header that consists of a sharp character (#), a number of 6 (which indicates the byte size of the <number of bytes transferred> part), and the <number of bytes transferred> part in this order. The header is followed by the binary data (each number consists of 8 bytes and the total is the byte size indicated by <number of bytes transferred>) and the message terminator <new line>^END.

The binary data is expressed in the IEEE 754 64-bit floating-point number format shown in the figure below.

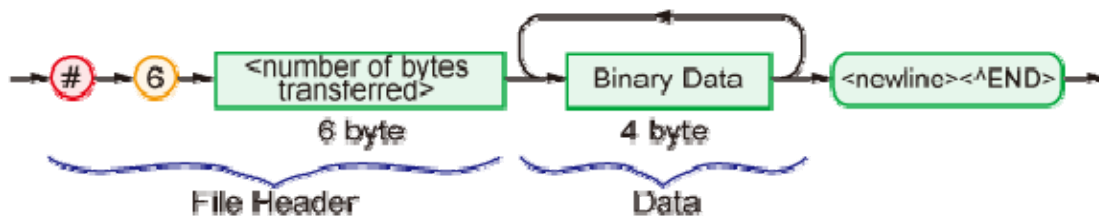
64-bit floating point format



IEEE 32-bit floating point format

When you select the IEEE 32-bit floating point binary transfer format as the data transfer format, numbers are transferred in the format shown in the figure below.

IEEE 32-bit floating point binary transfer format

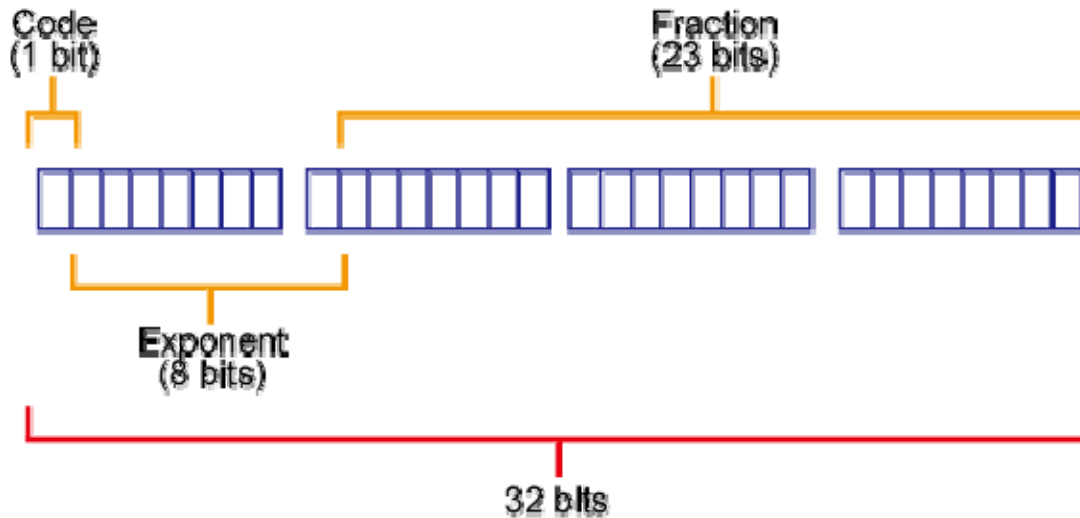


e5071e439

This data transfer format uses a header that consists of a sharp character (#), a number of 6 (which indicates the byte size of the <number of bytes transferred> part), and the <number of bytes transferred> part in this order. The header is followed by the binary data (each number consists of 4 bytes and the total is the byte size indicated by <number of bytes transferred>) and the message terminator <new line>^END.

The binary data is expressed in the IEEE 754 32-bit floating-point number format shown in the figure below.

32-bit floating point data



e5071c440

Byte order

When you opt to perform binary transfer, you can configure the instrument to transfer the bytes of the data in one of the following two byte orders:

NORMAL

Transfer begins with the byte that contains the MSB (Most Significant Bit); that is, the leftmost byte in 64 bit floating point format and 32 bit floating point data.

SWAPped

Transfer begins with the byte that contains the LSB (Least Significant Bit); that is, the rightmost byte in 64 bit floating point format and 32 bit floating point data.

To set the byte order, use the following command:

:FORM:BORD

NOTE

Executing the :SYST:PRES or *RST does not affect the current setting of the byte order.

Internal Data Processing

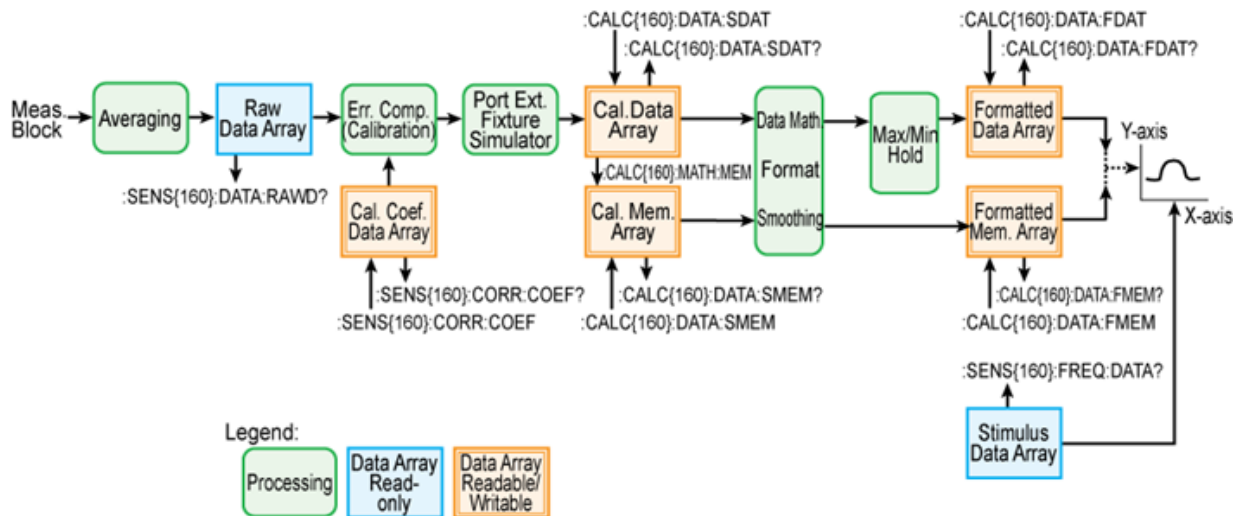
- [Data Flow](#)
- [Internal Data Arrays](#)

Other topics about Reading-Writing Measurement Data

Data Flow

The following figure provides an overview of the E5071C's internal data processing flow.

E5071C's data processing flow



(*1):CALC{1-160}:DATA:CORR? allows you to get the other S-parameter than the selected.

e5071c614

Internal Data Arrays

Raw data arrays

Raw data array contains uncorrected data. The data is obtained by performing IF range correction, ratio calculation, port characteristics correction and sweep averaging. This data will be used in error correction.

To read one of the raw data arrays, use the following command:

:SENS{1-160}:DATA:RAWD

Calibration Coefficient Data Arrays

A calibration coefficient data array contains the calibration coefficients calculated based on the results of measurement performed with standard devices.

Commands are available for reading or writing calibration coefficient data arrays. To read or write, first use the following command:

:SENS{1-160}:CORR:COEF**NOTE**

If any calibration coefficient is interpolated, the interpolated calibration coefficient data array will be retrieved. Once a calibration coefficient data array has been written, execute the command **:SENS{1-160}:CORR:COEF:SAVE** to validate it

Corrected data arrays

A corrected data array contains the corrected data obtained by performing error correction, port extension compensation (calibration), Fixture Simulator operations on the raw measured data of S-Parameter specified for each trace of each channel. Each data element is stored as a complex number (Re/Im).

To read/write one of the corrected data arrays, use the following command:

:CALC{1-160}:DATA:SDAT

Corrected data arrays (S-Parameter)

You can get the S-Parameter measurement result which the analyzer has.

For example, when you measure 2 full ports calibration, the analyzer measures not only the S-Parameter you selected but also four S-Parameters at the measurement. In this case, you can get the other S-Parameter.

To read one of the corrected S-Parameter data arrays, use the following command:

:SENS{1-160}:DATA:CORR

Corrected memory arrays

When the **:CALC{1-160}:MATH:MEM** command is executed on a particular corrected data array, its copy is stored into the corrected memory array corresponding to that corrected data array.

To read/write one of the corrected data arrays, use the following command:

:CALC{1-160}:DATA:SMEM

Formatted data array

A formatted data array contains the formatted data (values to be displayed) obtained by performing data math operations, measurement parameter conversion, and smoothing on a particular corrected data array.

Regardless of the data format, it contains two data elements per measurement point as shown in the following table:

Data format	Data element (primary value)	Data element (secondary value)
log magnitude	log magnitude	Always 0
Phase	Phase	Always 0
Group delay	Group delay	Always 0
Smith chart (Lin)	Liner magnitude	Phase
Smith chart (Log)	log magnitude	Phase
Smith chart (Re/Im)	Real part of a complex number	Imaginary part of a complex number
Smith chart (R+jX)	Resistance	Reactance
Smith chart (G+jB)	Conductance	Susceptance
Polar (Lin)	Liner magnitude	Phase
Polar (Log)	log magnitude	Phase
Polar (Re/Im)	Real part of a complex number	Imaginary part of a complex number
Liner magnitude	Liner magnitude	Always 0
SWR	SWR	Always 0
Real number	Real part of a complex number	Always 0
Imaginary number	Imaginary part of a complex number	Always 0
Expanded phase	Expanded phase	Always 0

To read/write one of the formatted data arrays, use the following command:

:CALC{1-160}:DATA:FDAT

Formatted memory arrays

A formatted memory array contains the formatted data (values to be displayed) obtained by performing data math operations, measurement parameter conversion, and smoothing on a particular corrected memory array.

To read/write one of the formatted memory arrays, use the following command:

:CALC{1-160}:DATA:FMEM

Stimulus data arrays

A stimulus data array contains the stimulus values for all measurement points.

The instrument retains 160 stimulus data arrays at maximum, each of which is associated with one of the 160 channels. Stimulus data arrays are read-only. To retrieve one of the stimulus data arrays, use the following command:

:SENS{1-160}:FREQ:DATA?

Retrieving Measurement Results

- [Overview](#)
- [Retrieving Internal Data Arrays](#)
- Sample Program

Other topics about Reading-Writing Measurement Data

Overview

Internal data arrays allows you to retrieve all measurement results throughout a particular trace. Alternatively, markers allow you to retrieve measurement results at your specified points. For information on how to retrieve marker values, refer to Retrieving measurement results at marker positions.

Retrieving Internal Data Arrays

You can chose between the ASCII and binary data transfer formats when you retrieve internal data arrays. For more information, please refer to Data Transfer Format.

Sample Program

See Reading Data in Ascii Format and Reading Data in Binary Format.

Entering Data into a Trace

- [Overview](#)
- [Sample Program](#)

Other topics about Reading-Writing Measurement Data

Overview

You can change the data/memory trace on the LCD by writing the new data into the Formatted data array/Formatted memory arrays.

When you write data into formatted data/memory array, you can choose either the ASCII or binary transfer format (see Data Transfer Format).

Using the ASCII Transfer Format to Write Formatted Data Arrays (write_a.htb) and Using the Binary Transfer Format to Write Formatted Data Arrays (write_b.htb) show sample programs that demonstrate how to write data into formatted data arrays. The sample program in Using the ASCII Transfer Format to Write Formatted Data Arrays (write_a.htb) uses the ASCII transfer format while the sample in Using the Binary Transfer Format to Write Formatted Data Arrays (write_b.htb) uses the binary transfer format. You can find the source files of these programs, named write_a.htb and write_b.htb, on the sample program disk.

Each of the sample programs holds the sweep on channel 1, retrieves the data from a specified file (a file saved measurement data using the [:MMEM:STOR:FDAT](#) command), and populates trace 1 for channel 1 with the retrieved data.

Sample Program

See Writing Data in Ascii Format and Writing Data in Binary Format.

Analyzing Data

Analyzing Data

- Retrieving Measurement Results at Specified Points
- Searching for Positions That Match Specified Criteria
- Bandwidth Search
- Notch Search
- Statistical Analysis
- Analysis Using the Fixture Simulator
- Analysis in Time Domain (time domain function)
- Analyzing impedance

Retrieving Measurement Results at Specified Points

- [Overview](#)
- [Showing/Hiding Markers](#)
- [Turning ON/OFF Reference Marker Mode](#)
- [Setting and Retrieving Stimulus Value at Marker Positions](#)
- [Retrieving Measurement Results at Marker Positions](#)

Other topics about Analyzing Data

Overview

Markers allow you to retrieve measurement results at specified points. You can use up to nine markers for each trace, and you can move them to any point on the trace. In addition to the regular markers, you can use a reference marker.

Showing/Hiding Markers

To show or hide markers, including the reference marker, use the following command:

:CALC{1-36}:MARK{1-10}

NOTE

You can move markers or retrieve the data at a marker even when the markers are hidden.

NOTE

The display of the reference marker is turned on or off when you turn On or Off Reference Marker mode.

Turning ON/OFF Reference Marker Mode

Turning on Reference Marker mode provides relative marker values with respect to the reference marker (by subtracting the value at the reference marker from the value at a particular marker).

To turn On or Off Reference Marker mode, use the following command:

:CALC{1-36}:MARK:REF

Setting and Retrieving Stimulus Value at Marker Positions

To set (or change along the frequency axis) the stimulus value at a particular marker or the reference marker or to retrieve the current stimulus value, use the following command:

:CALC{1-36}:MARK{1-10}:X

When Reference Marker mode is on, the stimulus value at a regular marker is a relative stimulus value obtained by subtracting the stimulus value at the reference marker from the actual stimulus value at that particular marker.

Retrieving Measurement Results at Marker Positions

To retrieve the measurement results (response values) at a particular marker or the reference marker, use the following command:

`:CALC{1-36}:MARK{1-10}:Y?`

When Reference Marker mode is on, the response value at a regular marker is a relative value obtained by subtracting the response value at the reference marker from the actual response value at that particular marker.

Searching for Positions Matching Specified Criteria

- [Overview](#)
- [Using Marker Search](#)
- [Analysis Commands](#)
- [Sample Program](#)

Other topics about Analyzing Data

Overview

You can search for a position that matches specified criteria by using the Marker Search feature or analysis commands.

Using Marker Search

NOTE

Marker Search is available whether the markers are shown or hidden.

Setting the Search Range

You can use either the entire sweep range or a user-defined range for the marker search range by using the following command:

`:CALC{1-36}:MARK:FUNC:DOM`

When you opt to use a user-defined range, use the following commands to set the range:

Description	Command
Start value (lower limit value)	<code>:CALC{1-36}:MARK:FUNC:DOM :STAR</code>
Stop value (upper limit value)	<code>:CALC{1-36}:MARK:FUNC:DOM :STOP</code>

You can also select whether to specify the marker search range independently for each trace by using the following command.

`:CALC{1-36}:MARK:FUNC:DOM:COUP`

Selecting a Search Type

Marker Search allows you to choose from the following eight search types:

- Maximum value
- Minimum value
- Peak (3 types)
 - Maximum peak (for a positive peak), minimum peak (for a negative peak)
 - Peak nearest to the marker position on its left-hand side

- Peak nearest to the marker position on its right-hand side
- Target (3 types)
 - Peak nearest to the marker position
 - Target nearest to the marker position on its left-hand side
 - Target nearest to the marker position on its right-hand side

To select a search type, use the following command:

`:CALC{1-36}:MARK{1-10}:FUNC:TYPE`

Defining a Peak

You can define a peak by specifying the lower limit for the peak excursion value and polarity (positive or negative peak).

To define a peak, use the following commands:

Lower limit for the peak excursion value	<code>:CALC{1-36}:MARK{1-10}:FUNC:PEXC</code>
Polarity	<code>:CALC{1-36}:MARK{1-10}:FUNC:PPOL</code>

Defining a Target

You can define a target by specifying the target value (response value) and transitional direction (positive or negative value change).

To define a target, use the following commands:

Target value	<code>:CALC{1-36}:MARK{1-10}:FUNC:TARG</code>
Transitional direction	<code>:CALC{1-36}:MARK{1-10}:FUNC:TTR</code>

Performing Marker Search

To perform Marker Search, use the following command:

`:CALC{1-36}:MARK{1-10}:FUNC:EXEC`

To turn On or Off the Search Tracking feature, which performs Marker Search every time the trace is updated, use the following command:

`:CALC{1-36}:MARK{1-10}:FUNC:TRAC`

Retrieving Search Results

Performing Marker Search moves the marker to the points that matches the search criteria, so you can obtain the search results by retrieving the marker value. For information on how to retrieve marker values, refer to

Setting (changing) and retrieving stimulus value at marker positions and
Retrieving measurement results at marker positions.

Analysis Commands

You can use the analysis Commands to perform search and analysis.

Setting the Search (Analysis) Range

You can use either the entire sweep range or a user-defined range as the search (analysis) range by using the following command:

:CALC{1-36}:FUNC:DOM

When you opt to use a user-defined range, use the following commands to set the range:

Start value (lower limit value)	:CALC{1-36}:FUNC:DOM:STAR
Stop value (upper limit value)	:CALC{1-36}:FUNC:DOM:STOP

You can also select whether to specify the marker search (analysis) range independently for each trace by using the following command:

:CALC{1-36}:FUNC:DOM:COUP

Selecting the Search (Analysis) Type

The analysis commands allows you to choose from the following five search types:

- Maximum value
- Minimum value
- Maximum peak (for a positive peak), minimum peak (for a negative peak)
- All peaks
- All targets

In addition, you can choose from the following three analysis types:

- Difference between the maximum and minimum values
- Standard deviation
- Average

To select the search (analysis) type, use the following command:

:CALC{1-36}:FUNC:TYPE

Defining a Peak

You can define a peak by specifying the lower limit for the peak excursion value and polarity (positive or negative peak).

To define a peak, use the following commands:

Lower limit for the peak excursion value	:CALC{1-36}:FUNC:PEXC
Polarity	:CALC{1-36}:FUNC:PPOL

Defining a Target

You can define a target by specifying the target value (response value) and transitional direction (positive or negative value change).

To define a target, use the following commands:

Target value	:CALC{1-36}:FUNC:TARG
Transitional direction	:CALC{1-36}:FUNC:TTR

Performing Search (Analysis)

To perform search (analysis), use the following command:

:CALC{1-36}:FUNC:EXEC

Retrieving Search (Analysis) Results

To retrieve search (analysis) results, use the following command:

:CALC{1-36}:FUNC:DATA?

The number of data items contained in search (analysis) results differ depending on the search (analysis) type and the number of points found by the search operation. To retrieve the number of data items, use the following command:

:CALC{1-36}:FUNC:POIN?

Sample Program

See Peak Search.

Bandwidth Search

- [Overview](#)
- [Setting Bandwidth Definition Value](#)
- [Retrieving Bandwidth Search Results](#)
- [Sample Program](#)

Other topics about Analyzing Data

Overview

The E5071C has a feature called Bandwidth Search. This feature searches for the cutoff points on both right- and left-hand sides of the marker position on the trace.

- Bandwidth (BW= high-low)
- Center frequency (cent= {high+low}/2))
- Q value (Q=cent/BW)
- Loss (response value at the marker position)

Where high is the right-hand cutoff point frequency; low is the left-hand cutoff point frequency.

Setting Bandwidth Definition Value

Bandwidth Search finds a point whose response value is different, by the amount defined as the bandwidth definition value, from the response value at the marker position, and identifies that point as the cutoff point.

To set the bandwidth definition value, use the following command:

```
:CALC{1-36}:MARK{1-10}:BWID:THR
```

Retrieving Bandwidth Search Results

Once you have moved the marker to the appropriate position using Marker Search or some other feature, you can retrieve the results of Bandwidth Search by using the following command:

```
:CALC{1-36}:MARK{1-10}:BWID:DATA?
```

Also, you can use the following command to control whether to display the results of Bandwidth Search on the LCD:

```
:CALC{1-36}:MARK:BWID
```

You can retrieve the results of Bandwidth Search even after you have turned off the display of markers or the results of Bandwidth Search.

Sample Program

See Bandwidth Search.

Notch Search

- Overview
- Setting the notch definition value
- Displaying the notch search result
- Reading out the notch search result

Other topics about Analyzing Data

Overview

The notch search function is used to obtain the bandwidth, center frequency, cutoff points (high-frequency side and low-frequency side), Q and insertion loss of a trace based on the active marker position. The notch search function starts at the left side of the active marker position, and ends when points that meet the condition are found.

- Bandwidth ($BW = \text{high} - \text{low}$)
- Center frequency ($\text{cent} = (\text{high} + \text{low})/2$)
- Q value ($Q = \text{cent}/BW$)
- Loss (response value at marker position)

Where, high is the right-hand cutoff point frequency, and low is the left-hand cutoff point frequency.

NOTE

For more information on notch search, see
Determining the bandwidth of a trace (Notch Search)

Setting the notch definition value

The notch search function finds a point whose response value is different, by the amount defined as the notch definition value, than the response value at the marker position, and identifies that point as the cutoff point.

To set the notch definition value, use the following command:

```
:CALC{Ch}:MARK{1-10}:NOTC:THR
```

Displaying the notch search result

The following command is used to control whether to display the notch search result on the LCD:

```
:CALC{Ch}:MARK:NOTC
```

Reading out the notch search result

Once the marker is moved to an appropriate position using the marker search function or some other function, it is able to retrieve the notch search result using the following command:

```
:CALC{Ch}:MARK{1-10}:NOTC:DATA
```

NOTE

It is able to retrieve the notch search result regardless of whether the marker display and the notch search result display is on/off.

Statistical Analysis

The E5071C provides a mechanism that analyzes trace statistics. You can analyze the following statistics:

- Average
- Standard deviation
- Difference between the maximum and minimum values

To retrieve the results of statistical analysis, use the following command:

:CALC{1-36}:MST:DATA?

Also, you can use the following command to control whether to display the results of statistical analysis on the LCD:

:CALC{1-36}:MST

NOTE

You can retrieve the results of statistical analysis even after you have turned off the display showing these results.

Alternatively, you can use the analysis commands to analyze the trace statistics. When you use the analysis commands, you can analyze the trace data in a specific range as well as the trace data throughout the entire sweep range. For information on how to use the analysis commands, refer to Analysis Commands.

Other topics about Analyzing Data

Analysis Using Fixture Simulator

- [Overview](#)
- [Matching Circuit Embedding](#)
- [Port Impedance Conversion](#)
- [Network De-embedding](#)
- [4-Port Network Embedding/De-embedding](#)
- [Balance-Unbalance Conversion \(only 3-port/4-port models\)](#)
- [Differential Matching Circuit Embedding \(only 3-port/4-port models\)](#)
- [Differential Port Impedance Conversion \(only 3-port/4-port models\)](#)
- [Sample Program](#)

Other topics about Analyzing Data

Overview

The Fixture Simulator provides the following functions:

- Matching Circuit Embedding
- Port Impedance Conversion
- Network De-embedding
- 4-Port Network Embedding/De-embedding
- Balance-Unbalance Conversion (only 3-port/4-port models)
- Differential Matching Circuit Embedding (only 3-port/4-port models)
- Differential Port Impedance Conversion (only 3-port/4-port models)

Before you can use any of the features listed above, you must turn on the desired feature individually and issue the following command to turn on the Fixture Simulator:

:CALC{1-36}:FSIM:STAT
[Matching Circuit Embedding](#)

The Matching Circuit feature simulates the characteristics that the DUT would exhibit when connected to a matching circuit.

On/Off

To turn on or off the Matching Circuit, use the following command:

:CALC{1-36}:FSIM:SEND:PMC:STAT

You can only turn on or off Matching Circuit for all the ports, not for each port individually. However, any port whose circuit type is set to "None" behaves as if this feature were permanently off.

Configuring the Matching Circuit Settings

To select a circuit type, use the following command:

`:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}`

To set the circuit constant, use the following commands:

Circuit constant	Command
C	<code>:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:C</code>
G	<code>:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:G</code>
L	<code>:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:L</code>
R	<code>:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:R</code>

When you want to use a user file to define the circuit type, specify the file by using the following command:

`:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:USER:FILE`

Port Impedance Conversion

The Port Impedance Conversion feature converts the measurement results with a port impedance of 50 ohm to the characteristics of a different port impedance.

On/Off

To turn on or off Port Impedance Conversion, use the following command:

`:CALC{1-36}:FSIM:SEND:ZCON:STAT`

You can only turn on or off Port Impedance Conversion for all of the ports, not for each port individually. However, any port with ZO set to 50 ohm behaves as if this feature were permanently off.

Setting the ZO Value

To set the target port impedance, use the following command:

`:CALC{1-36}:FSIM:SEND:ZCON:PORT{1-4}:ZO`

`:CALC{1-36}:FSIM:SEND:ZCON:PORT{1-4}:REAL`

`:CALC{1-36}:FSIM:SEND:ZCON:PORT{1-4}:IMAG`

Network De-embedding

The Network De-embedding feature eliminates any network that can cause error between the calibration level and the DUT.

On/Off

To turn on or off Network De-embedding, use the following command:

`:CALC{1-36}:FSIM:SEND:DEEM:STAT`

You can only turn on or off Network De-embedding for all of the ports, not for each port individually. However, any port whose Network De-

embedding type is set to "None" behaves as if this feature were permanently off.

Selecting a Type

To select a Network De-embedding type, use the following command:

```
:CALC{1-36}:FSIM:SEND:DEEM:PORT{1-4}
```

Specifying the File

To specify the file that defines the criteria for Network De-embedding, use the following command:

```
:CALC{1-36}:FSIM:SEND:DEEM:PORT{1-4}:USER:FIL
```

4-Port Network Embedding/De-embedding

The 4-port Network Embedding/De-embedding feature is provided to embed or de-embed a network (defined in a 4-port touchstone file) between the calibration surface and the DUT.

On/Off

To turn on or off the 4-port network embedding/de-embedding feature, use the following command:

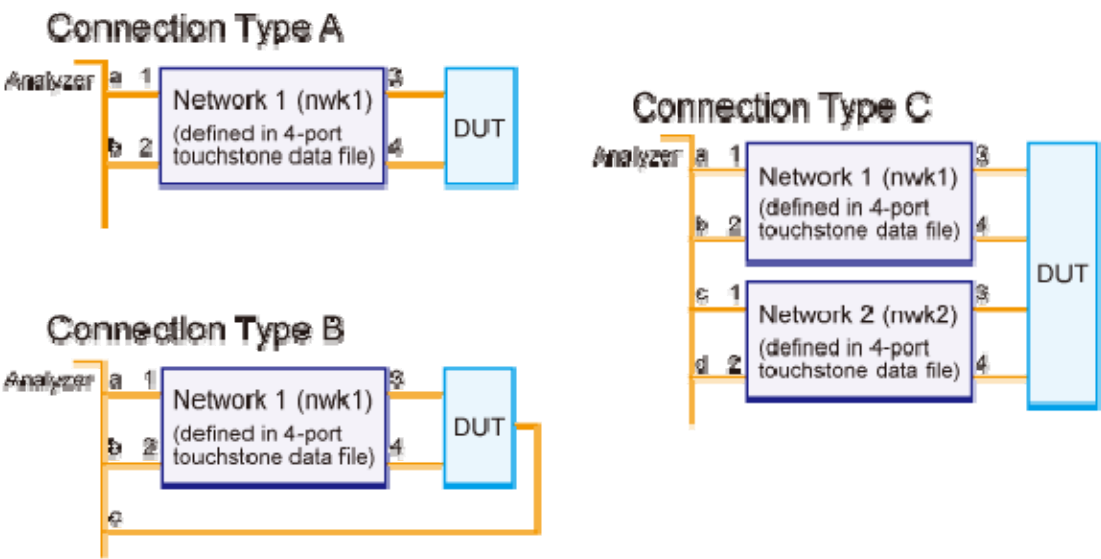
```
:CALC{1-36}:FSIM:EMB:STAT
```

Setting Topology (connection method between analyzer and DUT)

To select a connection type, use the following command:

```
:CALC{1-36}:FSIM:EMB:TYPE
```

Connection type



E5071C313

To assign the ports (ports a through d in Connection_type), use the appropriate command that matches your selected connection type, as identified in the following table:

Connection type	Command
A	<code>:CALC{1-36}:FSIM:EMB:TOP:A:PORT</code>
B	<code>:CALC{1-36}:FSIM:EMB:TOP:B:PORT</code>
C	<code>:CALC{1-36}:FSIM:EMB:TOP:C:PORT</code>

Selecting Processing Type (embedding/de-embedding)

To select a network processing type, use the following command:

`:CALC{1-36}:FSIM:EMB:NETW{1-2}:TYPE`

Specifying File

To select a file that defines the criteria for network embedding/de-embedding, use the following command:

`:CALC{1-36}:FSIM:EMB:NETW{1-2}:FIL`
Balance-Unbalance Conversion

The Balance-Unbalance Conversion feature converts the measurement results obtained in an unbalanced state to the characteristics of a balanced state. You can select mixed mode S parameter, balance and CMRR as the measurement parameter when you turn on Balance-Unbalance Conversion.

On/Off

You can turn on or off Balance-Unbalance Conversion for each trace individually. To turn on or off Balance-Unbalance Conversion, use the following command:

```
:CALC{1-36}:FSIM:BAL:PAR{1-36}:STAT
```

Setting the Topology

To select a balance device type, use the following command:

```
:CALC{1-36}:FSIM:BAL:DEV
```

Balance device type

Balance Device Type:Unbalance-Balance (SBALanced)



Balance Device Type:Balance-Balance (BBALanced)



Balance Device Type:Unbalance-Unbalance-Balance (SSBALanced)



950716302

To assign the ports (ports a through d in Balance_device_type), use the command that matches your selected device type, as identified in the following table:

Device Type	Command
Unbalance-balance (SBALanced)	:CALC{1-36}:FSIM:BAL:TOP :SBAL
Balance-balance (BBALanced)	:CALC{1-36}:FSIM:BAL:TOP :BBAL
Unbalance-unbalance-balance (SSBALanced)	:CALC{1-36}:FSIM:BAL:TOP :SSB

Selecting the Measurement Parameter

To select the measurement parameter, use the command that matches your selected device type, as identified in the following table:

Device Type	Command
Unbalance-balance	<code>:CALC{1-36}:FSIM:BAL:PAR{1-36} :SBAL</code>
Balance-balance	<code>:CALC{1-36}:FSIM:BAL:PAR{1-36} :BBAL</code>
Unbalance-unbalance-balance	<code>:CALC{1-36}:FSIM:BAL:PAR{1-36} :SSB</code>

Differential Matching Circuit Embedding

The Differential Matching Circuit Embedding feature simulates the characteristics the DUT would exhibit if a balance-unbalance converted differential port were connected to a matching circuit after being subjected to balance-unbalance conversion.

On/Off

To turn on or off Differential Matching Circuit Embedding, use the following command:

`:CALC{1-36}:FSIM:BAL:DMC:STAT`

You can only turn on or off Differential Matching Circuit Embedding for all of the ports, not for each balance port individually. However, any balance port whose circuit type is set to "None" behaves as if this feature were permanently off.

Configuring the Matching Circuit Settings

To select a circuit type, use the following command:

`:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}`

To set the circuit constant, use the following commands:

Circuit constant	Command
C	<code>:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:C</code>
G	<code>:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:G</code>
L	<code>:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:L</code>
R	<code>:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:R</code>

When you want to use a user file to define the circuit type, specify the file using the following command:

`:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:USER:FIL`

Differential Port Impedance Conversion

The Differential Port Impedance Conversion feature converts the measurement results for a balance-unbalance converted differential port to the characteristics of a different port impedance.

On/Off

To turn on or off Differential Port Impedance Conversion, use the following command:

`:CALC{1-36}:FSIM:BAL:DZC:STAT`

You can only turn on or off Differential Port Impedance Conversion for all of the balance ports, not for each port individually.

Setting the Z0 Value

To set the target differential port impedance, use the following command:

`:CALC{1-36}:FSIM:BAL:DZC:BPOR{1-2}:Z0`

`:CALC{1-36}:FSIM:BAL:DZC:BPOR{1-2}:REAL`

`:CALC{1-36}:FSIM:BAL:DZC:BPOR{1-2}:IMAG`

Sample Program

See Fixture Simulator.

Analysis in Time Domain (time domain function)

- [Overview](#)
- [Transforming Measurement Data to Time Domain](#)
- Sample Program

Other topics about Analyzing Data

Overview

The time domain function provides the following functions:

- Transforming measurement data to data in the time domain (Transformation Function)
- Deleting unnecessary measurement data in the time domain (Gating Function)

Transforming Measurement Data to Time Domain

By using the Transformation Function, you can convert the results measured in the frequency domain to data in the time domain and analyze it.

On/Off

To turn on or off the transformation function, use the following command:

:CALC{1-36}:TRAN:TIME:STAT

Selecting Transformation Type

To select the transformation type (band pass/low pass), use the following command:

:CALC{1-36}:TRAN:TIME

To select the stimulus type (impulse/step) when the transformation type is low pass, use the following command:

:CALC{1-36}:TRAN:TIME:STIM

When the transformation type is low pass, you need to execute the following command because each measurement point must be a multiple of the start frequency.

:CALC{1-36}:TRAN:TIME:LPFR

Setting Window Shape

To set the window shape, use one of the following items.

Item	Command
β	:CALC{1-

	36}:TRAN:TIME:KBES
Impulse width	:CALC{1-36}:TRAN:TIME:IMP :WIDT
Rise time of step signal	:CALC{1-36}:TRAN:TIME:STEP :RTIM

The above three items are dependent on each other. When the value of one of them is changed, the values of the other two are automatically changed to corresponding values.

Unlike manual operation, you cannot set the window shape by selecting the window type (maximum/normal/minimum). However, you can set the same shape as each window type by setting β as follows:

	Maximum	Normal	Minimum
Value of β	13	6	0

Setting Display Range

To set the display range after time domain transformation, use the following commands:

Description	Command
Start value	:CALC{1-36}:TRAN:TIME:STAR
Stop value	:CALC{1-36}:TRAN:TIME:STOP
Center value	:CALC{1-36}:TRAN:TIME:CENT
Span value	:CALC{1-36}:TRAN:TIME:SPAN

Deleting unnecessary measurement data in the time domain

You can use the Gating Function to delete unnecessary time domain data.

On/Off

To turn on or off the gating function, use the following command:

:CALC{1-36}:FILT:TIME:STAT

Selecting Gate Type

To select the gate type, use the following command:

:CALC{1-36}:FILT:TIME

Setting Gate Shape

To select the gate shape, use the following command:

[:CALC{1-36}:FILT:TIME:SHAP](#)

Setting Gate Range

To set the gate range, use the following commands:

Description	Command
Start value	:CALC{1-36}:FILT:TIME:STAR
Stop value	:CALC{1-36}:FILT:TIME:STOP
Center value	:CALC{1-36}:FILT:TIME:CENT
Span value	:CALC{1-36}:FILT:TIME:SPAN

[Sample Program](#)

See the Time Domain.

Analyzing impedance

By turning on the parameter conversion function, you can convert the measurement result of the S parameter to the following parameters.

- Equivalent impedance in reflection measurement
- Equivalent impedance in transmission measurement
- Equivalent admittance in reflection measurement
- Equivalent admittance in transmission measurement
- Inverse of S-parameter ($1/S$)

To turn On/Off the parameter conversion function, use the following command:

`:CALC{1-36}:CONV`

To select the parameter to which you want to convert the measurement result, use the following command:

`:CALC{1-36}:CONV:FUNC`

Other topics about Analyzing Data

E5071C

Limit Test

Limit Test

- Performing a Limit Test
- Obtaining Test Results

Performing a Limit Test

- [Configuring Limit Lines](#)
- [Showing/Hiding Limit Lines](#)
- [Turning Limit Test Feature ON/OFF](#)
- [Showing/Hiding "Fail"](#)

Other topics about Limit Test

Configuring Limit Lines

The Limit Test feature of the E5071C allows you to create up to 100 upper/lower limit lines on each trace and then determine the pass/fail status of the measured data.

When you manually configure limit lines, you must add each segment (limit line) to the limit table and define various conditions that apply to the specific segment. On the other hand, when you use an external controller to configure limit lines, you can use the following command to define all of the segment conditions (all limit lines) in the active table trace at once.

:CALC{1-36}:LIM:DATA

Alternatively, you can configure limit lines based on the data contained in a CSV file by issuing the following command:

:MMEM:LOAD:LIM

Also, you can save the contents of the current limit table to a file by issuing the following command:

:MMEM:STOR:LIM

Showing/Hiding Limit Lines

To turn ON or OFF limit lines, use the following command:

:CALC{1-36}:LIM:DISP

Even when limit lines are hidden, the system performs a limit test if the Limit Test feature is on.

Turning Limit Test Feature ON/OFF

To turn ON or OFF the Limit Test feature, use the following command:

:CALC{1-36}:LIM

Showing/Hiding "Fail"

To turn ON or OFF the "Fail" indicator that is displayed at the center of the window when the test result for the channel is "fail," use the following command:

:DISP:FSIG

Obtaining Test Results

- [Overview](#)
- [Test Results at each Measurement Point](#)
- [Test Results for each Trace](#)
- [Test Results for each Channel](#)
- [Overall Test Results](#)

Other topics about Limit Test

Overview

You can obtain test results by issuing a result retrieval command or through the status register.

Test Results at each Measurement Point

Using commands that retrieve test results

You can obtain the test results at each measurement point by retrieving the stimulus value at failed measurement points. To retrieve failed measurement points, use the following command:

Stimulus value	:CALC{1-36}:LIM:REP?
Number of measurement points	:CALC{1-36}:LIM:REP:POIN?

Using the status register

You cannot use the status register to obtain the test results at each measurement point.

Test Results for each Trace

Using commands that retrieve test results

You can retrieve the test result for each trace (i.e., the trace-wide result that combines the results for all measurement points in a particular trace) by issuing the following command:

[:CALC{1-36}:LIM:FAIL?](#)

Using the status register

The condition register and event register under the questionable limit channel {1-16} status register provide 14 bits that correspond to traces 1 to 14 and contain the test results (0: Pass, 1: Fail) for the respective traces; for example, you can obtain the test result for trace 1 from bit 1 and that for trace 14 from bit 14. The condition register and event register

under the questionable limit channel {1-16} extra status register provide two bits that correspond to traces 15 to 16 and contain the test results (0: Pass, 1: Fail) for the respective traces; for example, you can obtain the test result for trace 15 from bit 1 and that for trace 16 from bit 2.

Every bit of the condition register is set to 0 when a measurement cycle is started. Upon completion of measurement, those bits that correspond to failed traces are set to 1.

If the corresponding bit of the positive transition filter is set to 1 (preset value), each bit of the event register is set to 1 when the corresponding bit of the condition register changes from 0 to 1 (indicating that the corresponding trace failed the test).

To retrieve the registers, use the following commands:

Questionable limit channel { 1-16 } status register	
Condition register	:STAT:QUES:LIM:CHAN{1-16}:COND?
Event register	:STAT:QUES:LIM:CHAN{1-16}?
Questionable limit channel { 1-16 } extra status register	
Condition register	:STAT:QUES:LIM:CHAN{1-16}:ECH:COND?
Event register	:STAT:QUES:LIM:CHAN{1-16}:ECH?

NOTE

Although up to 36 channel is available on E5071C, the registers for limit test and Ripple/Bandwidth limit tests are for up to 16 channel.

Test Results for each Channel**Using commands that retrieve test results**

No command is available that allows you to directly retrieve the test result for each channel (i.e., the channel-wide result that combines the results for all traces in a particular channel).

Using the status register

The questionable limit status event register provides 14 bits that correspond to channels 1 to 14 and contain the test results (0: Pass, 1: Fail) for the respective channels; for example, you can obtain the test result for channel 1 from bit 1 and that for channel 14 from bit 14. The questionable limit extra status event register provides nine bits that correspond to channels 1 to 2 and contain the test results (0: Pass, 1: Fail)

for the respective channels; for example, you can obtain the test result for channel 15 from bit 1 and that for channel 16 from bit 2.

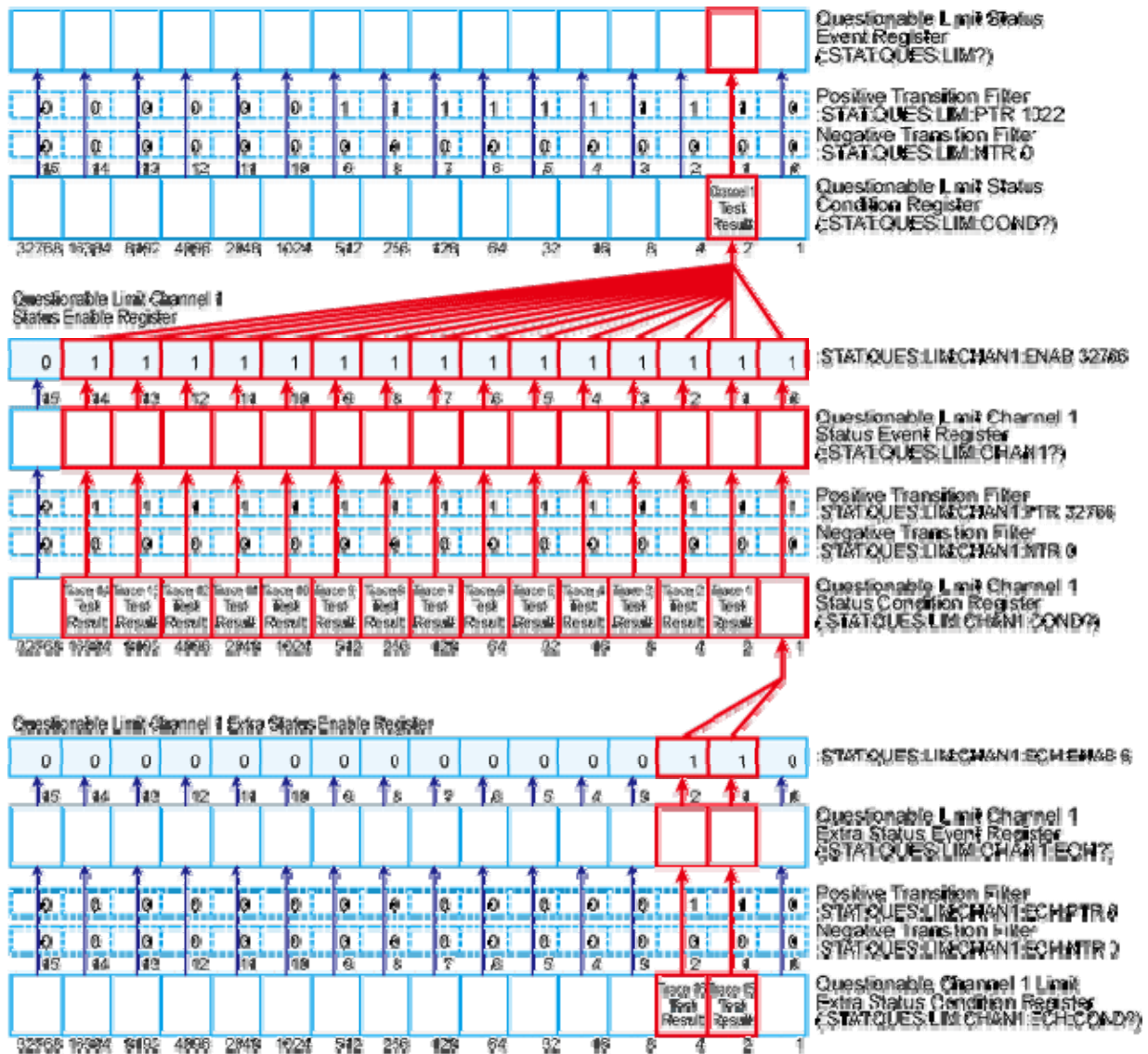
Every bit of the condition register is set to 0 after the event registers are cleared by the *CLS. Upon completion of measurement, if the channel-wide test result that combines the results for all traces in a channel is "fail," the corresponding bit of the condition register is set to 1.

If the corresponding bit of the positive transition filter is set to 1 (preset value), every bit of the event register is set to 1 when the corresponding bit of the condition register changes from 0 to 1.

To retrieve the registers, use the following commands:

Questionable limit status register	
Condition register	:STAT:QUES:LIM:COND?
Event register	:STAT:QUES:LIM?
Questionable limit extra status register	
Condition register	:STAT:QUES:LIM:ELIM:COND?
Event register	:STAT:QUES:LIM:ELIM?

Obtaining test results for a channel (channel 1 in this example) using the status register



e5071c43f

Overall Test Results

Using commands that retrieve test results

No command is available that allows you to directly retrieve the overall test results that combine the test results for all channels.

Using the status register

The condition register and event register under the questionable status event register each provides bit 10, from which you can obtain the overall test result (0: Pass, 1: Fail).

E5071C

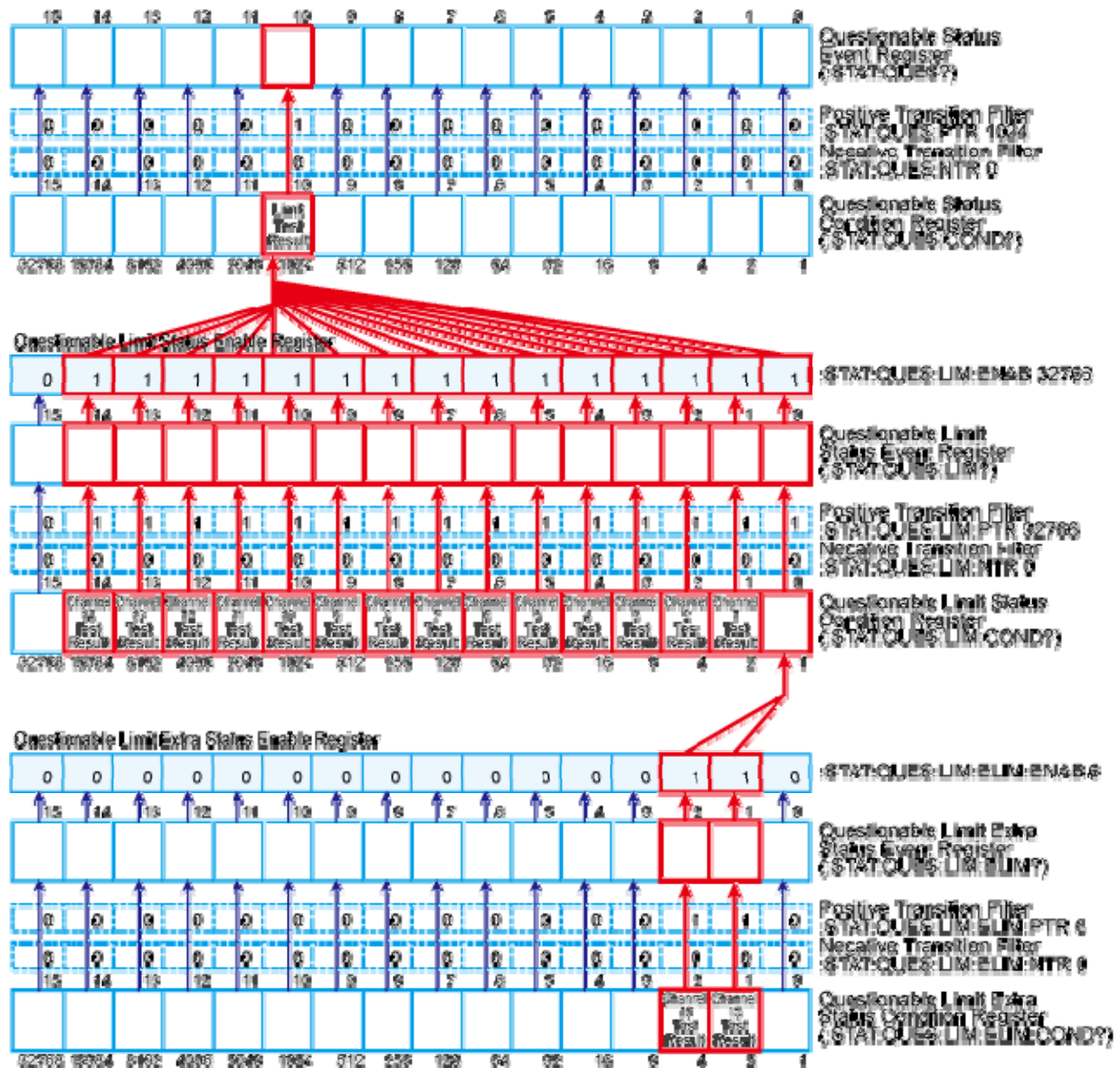
The condition register's bit 10 is set to 0 after the event registers are cleared by the *CLS. Upon completion of measurement, this bit is set to 1 if the overall test result that combines the results for all channels is "fail."

If the positive transition filter's bit 10 is set to 1 (preset value), the event register's bit 10 is set to 1 when the condition register's bit 10 changes from 0 to 1.

To retrieve the condition register and event register under the questionable status event register, use the following commands:

Condition register	:STAT:QUES:COND?
Event register	:STAT:QUES?

Obtaining overall test results using the status register



e5071c438

E5071C

Saving and Recalling

Saving and Recalling

- Saving and Recalling File
- Managing Files

Saving and Recalling File

- [Specifying File](#)
- [Saving and Recalling Instrument Status](#)
- [Saving Measurement Data](#)
- [Saving Measurement Data in Touchstone Format](#)
- [Saving Images](#)
- [Saving and Recalling Segment Sweep Table](#)
- [Saving and Recalling Limit Table](#)
- [Saving/Recalling Power Sensor Calibration Factor Table](#)
- [Saving/Recalling Loss Compensation Table](#)
- [Saving/Loading \(Importing\) a VBA Program](#)

Other topics about Saving and Recalling

Specifying File

When running a command for saving, recalling, and managing files, use a filename with extension to specify a particular file. Specify "D:" in the beginning of the file name, when specifying a file on the user area of hard disk. Also, when specifying a file name with directory, use "/" (slash) or "\" (backslash) as a delimiter.

Saving and Recalling Instrument Status

You can save the instrument state using one of the following 2 methods:

- Saving the entire instrument state into a file
- Saving the state for each channel into registers A to D (volatile memory)

Selecting content to be saved

When saving the instrument status into a file or register, the content to be saved can be selected among the following 4 options:

- Instrument status only
- Instrument status and calibration coefficient array.
- Instrument status, corrected data/memory array (measurement data)
- Instrument status, calibration coefficient array, and corrected data/memory array (measurement data)

To select a content to be saved, use the following command:

:MMEM:STOR:STYP

E5071C

Selecting Content to be Saved

To select whether to save the setting of all channels/traces or that of the displayed channels/traces, use the following command:

:MMEM:STOR:SALL

Saving and recalling entire instrument status

To save the entire instrument status into a file, use the following command:

:MMEM:STOR

Recalling a file saved with the above command can reproduce the status when it was saved. To recall the settings from a file, use the following command:

:MMEM:LOAD

Auto recall

The file saved with the name autorec.sta or A:autorec.sta will be automatically recalled when the E5071C is powered ON.

Saving the state for each channel into a register

For the active channel, when you want to save the instrument state specific to that channel into only one of the registers A to D, use the following command:

:MMEM:STOR:CHAN

Recalling an instrument state saved in a register can reproduce it as the state of the active channel. To recall a register, use the following command:

:MMEM:LOAD:CHAN

NOTE

It is possible to recall a file from a different channel where it was saved.

The contents in the registers are lost when you turn OFF the power. You can delete (clear) the contents of all registers using the following command.

:MMEM:STOR:CHAN:CLE

Saving Measurement Data

Measurement data (in a formatted data array) can be saved to a file in CSV (Comma Separated Value) format.

To save measurement data in a file, use the following command:

:MMEM:STOR:FDAT

Executing the above command will save the measurement data of the active trace. Note that the data saved using the above command cannot be recalled from the E5071C.

Saving Measurement Data in Touchstone Format

Measurement data for the active channel can be saved to a file in touchstone format.

To determine a file type in touchstone file format and specify a port, use one of the following commands according to the number of ports used:

- `:MMEM:STOR:SNP:TYPE:S1P`
- `:MMEM:STOR:SNP:TYPE:S2P`
- `:MMEM:STOR:SNP:TYPE:S3P`
- `:MMEM:STOR:SNP:TYPE:S4P`

To set a data type for files saved in touchstone format, use the following command:

`:MMEM:STOR:SNP:FORM`

To save measurement data in touchstone format, use the following command:

`:MMEM:STOR:SNP`

- Only trace data of frequency sweep can be saved in touchstone format files. Trace data of frequency offset measurement and power sweep measurement cannot be saved.

Saving Images

Images displayed on the LCD screen can be saved to a file in the bitmap (.bmp) or portable network graphics (.png) format.

To save the screen image to a file, use the following command:

`:MMEM:STOR:IMAG`

Executing the above command will save the screen image when the command is invoked.

- This gives different screen image results from those obtained by pressing the [Capture] key on the front panel.

Saving and Recalling Segment Sweep Table

Segment sweep table can be saved in the file with CSV (Comma Separated Value) format.

To save segment sweep table on a file, use the following command:

`:MMEM:STOR:SEGM`

Executing the above command will save the segment sweep table for the active channel.

Recalling the file saved using the above command can reproduce the segment sweep table on the active channel.

To recall the settings from a file, use the following command:

:MMEM:LOAD:SEGM

- It is possible to recall a file from a different channel where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving and Recalling Limit Table

Limit table can be saved in the file with CSV (Comma Separated Value) format. To save limit table on a file, use the following command:

:MMEM:STOR:LIM

Executing the above command will save the limit table for the active trace of the active channel.

Recalling the file saved using the above command can reproduce the limit table on the active trace of the active channel. To recall the settings from a file, use the following command:

:MMEM:LOAD:LIM

- It is possible to recall a file from a different channel or trace where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Recalling Power Sensor Calibration Factor Table

The power sensor calibration table can be saved in the file with CSV (Comma Separated Value) format. To save the power sensor calibration factor table that is set specifically to the E5071C into a file, use the following commands:

- **:MMEM:STOR:ASCF**
- **:MMEM:STOR:BSCF**

Recalling a file saved with the above command can reproduce the power sensor calibration factor table when it was saved. To recall the settings from a file, use the following command:

- **:MMEM:LOAD:ASCF**
- **:MMEM:LOAD:BSCF**
 - Recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Recalling Loss Compensation Table

The loss compensation table can be saved in the file with CSV (Comma Separated Value) format. To save the loss compensation table that is set for each port into a file, use the following command:

:MMEM:STOR:PLOS{1-4}

Executing the above command will save the loss compensation table for the active channel.

Recalling the file saved using the above command can reproduce the loss compensation table on the active channel. To recall the settings from a file, use the following command:

:MMEM:LOAD:PLOS{1-4}

- It is possible to recall a file from a different channel or port where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Loading (Importing) a VBA Program

Saving

Only the VBA project file can be saved using command.

To save the VBA project that is opened on the VBA editor on the file, use the following command.

:MMEM:STOR:PROG

Loading (importing)

To load the VBA project to the VBA editor, or to import the module/form file, use the following command.

:MMEM:LOAD:PROG

Executing above command will load/import the file according to its extension as follows:

Extension	File type
vba	VBA Project
bas	Standard module
frm	User Forms
cls	Class Modules

Sample program

See the Saving Files.

Managing Files

- [Various Commands](#)
- [Sample Program](#)

Other topics about Saving and Recalling

Various Commands

Creating directory (folder)

To create a directory (folder), use the following command:

:MMEM:MDIR

Deleting file (directory)

To delete a file or a directory, use the following command:

:MMEM:DEL

Copying file

To copy a file, use the following command:

:MMEM:COPY

Transferring files

File transfer from the external controller to the E5071C can be possible by reading data from a file on the controller and then writing them to the file on the E5071C.

:MMEM:TRAN

Also, file transfer from the E5071C to the external controller can be possible by reading data from a file on the E5071C using the commands as query and then writing them to the file on the controller.

Retrieving data from storage

To retrieve information for the storage that is built in the E5071C (usage, property of file located in a specified directory), use the following command;

:MMEM:CAT?

Sample Program

See the Transferring Files.

Communication with External Devices using I/O Port

Communication with External Devices (Handler I/O)

- Handler I/O Port Overview
- I/O Signal Pin Layout and Description
- Inputting/Outputting Data
- Preset states at power-on
- Timing Chart
- Electrical Characteristics

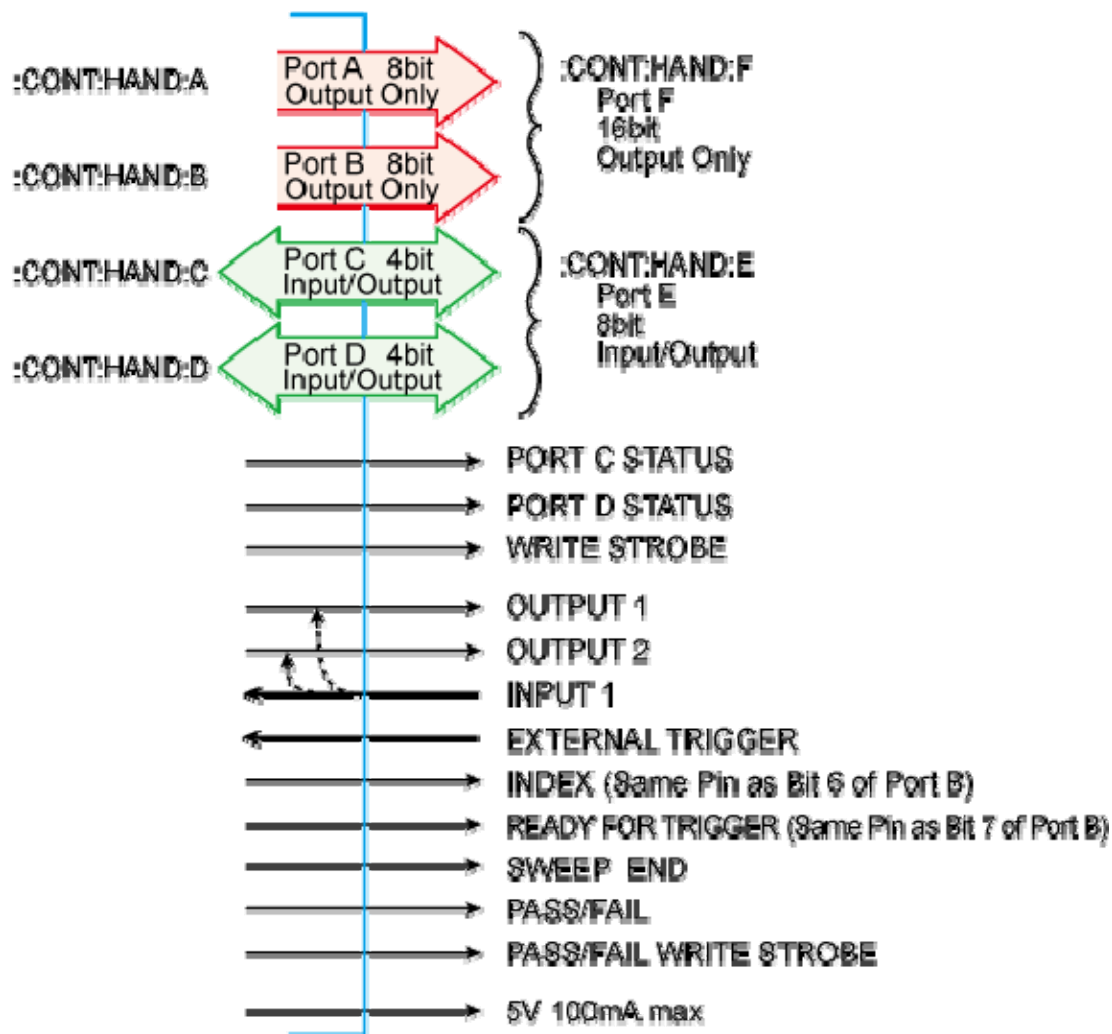
Handler I/O Port Overview

The E5071C handler I/O port provides four independent parallel ports for data I/O associated with several control signal lines and the power line. All signals operate in TTL logic.

The data I/O ports are configured with 2 pairs of 8 bit output port and 2 pairs of 4 bit bi-directional port. Also those ports can cooperate to provide a maximum 16-bit-width output port or a maximum 8-bit-width input port.

The I/O signals operate on the negative logic basis, which can be altered. The control signal lines consist of various control output data, including completion of measurement or control signal for handshaking.

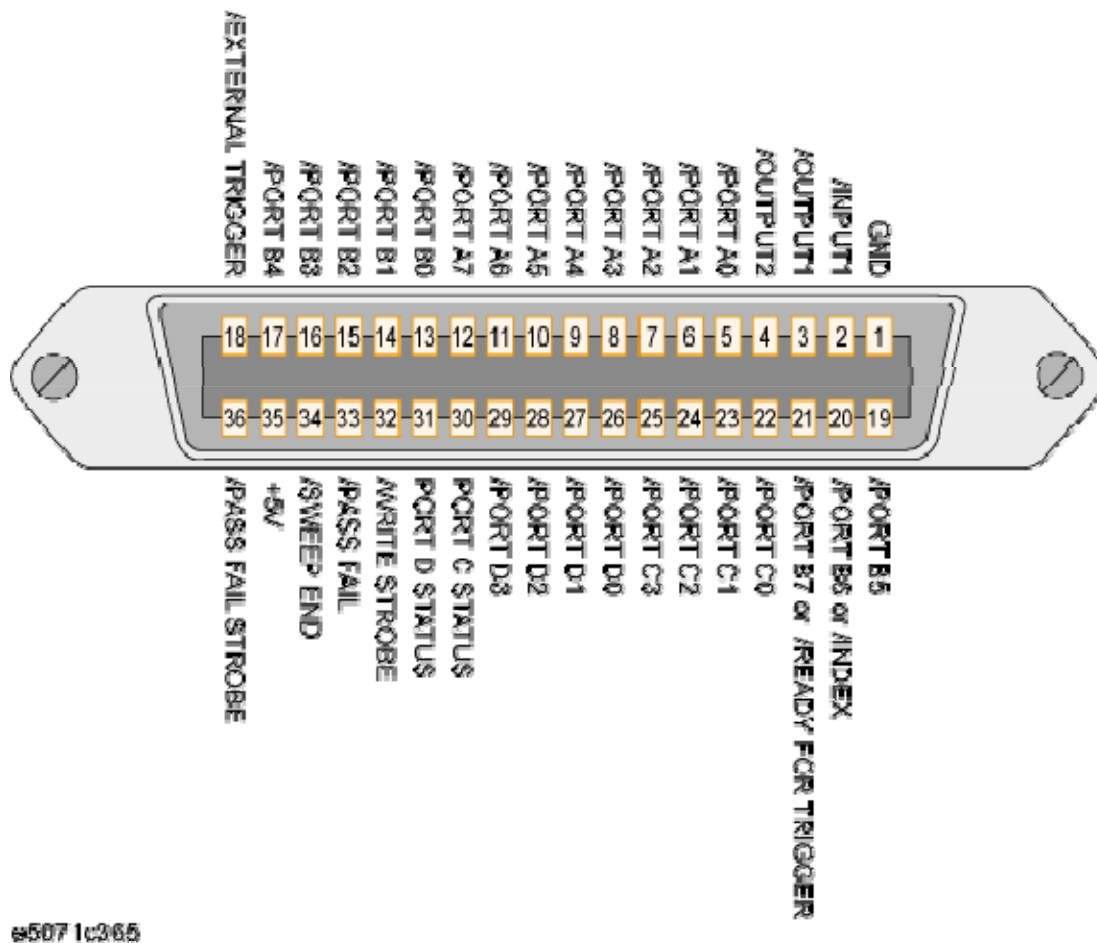
I/O ports and control signal lines



©5071c384

I/O Signal Pin Layout and Description

The layout of the I/O signal pins on the handler interface connector and its description are shown below.



A slash (/) symbol preceding signal names means that they are negative logic (active low).

Pin number	Signal name	Input/Output	Description
1	GND	N/A	Ground.
2	/INPUT1	Input	When this port receives a negative pulse, /OUTPUT1 and /OUTPUT2 are changed to the Low level.
3	/OUTPUT1	Output	Changes to the Low level when /INPUT1 receives a negative pulse. A command can be available for altering the

			Low/High level logic.
4	/OUTPUT2	Output	Changes to the Low level when /INPUT1 receives a negative pulse. A command can be available for altering the Low/High level logic.
5	/PORT A0	Output	Bit 0 of port A (8 bit parallel output port)
6	/PORT A1	Output	Bit 1 of port A.
7	/PORT A2	Output	Bit 2 of port A.
8	/PORT A3	Output	Bit 3 of port A.
9	/PORT A4	Output	Bit 4 of port A.
10	/PORT A5	Output	Bit 5 of port A.
11	/PORT A6	Output	Bit 6 of port A.
12	/PORT A7	Output	Bit 7 of port A.
13	/PORT B0	Output	Bit 0 of port B (8 bit parallel output port)
14	/PORT B1	Output	Bit 1 of port B.
15	/PORT B2	Output	Bit 2 of port B.
16	/PORT B3	Output	Bit 3 of port B.
17	/PORT B4	Output	Bit 4 of port B.
18	/EXTERNAL TRIGGER	Input	An external trigger signal. When the trigger source is set to the "External," this port generates a trigger in response to the trailing edge of a negative pulse.
19	/PORT B5	Output	Bit 5 of port B.
20	/PORT B6	Output	Bit 6 of port B.
	/INDEX		Indicates that analog measurement is complete. The /INDEX signal changes to the Low level when analog measurement (all sweeps of all channels) is complete. When the handler receives the signal, it assumes that it is ready to connect the next DUT. However, no measurement data are available until data calculation is completed. When the point trigger function is on, it

			goes to the High level before starting measurement of the first measurement point and returns to the Low level after completing measurement of all measurement points.
21	/PORT B7	Output	Bit 7 of port B.
	/READY FOR TRIGGER		<p>Indicates that the instrument is ready for triggering. This signal is changed to the Low level when the instrument is ready for receiving a trigger signal.</p> <p>With the point trigger function on, it goes to the Low level when the instrument is ready to accept the trigger signal for the first point and goes to the High level when the trigger signal for the first point is received. When measurement of all measurement points is completed and the instrument is ready to receive the trigger signal for the first point of the next sweep, this signal goes to the Low level again.</p>
22	/PORT C0	Input/Output	Bit 0 of port C (4 bit parallel I/O port)
23	/PORT C1	Input/Output	Bit 1 of port C.
24	/PORT C2	Input/Output	Bit 2 of port C.
25	/PORT C3	Input/Output	Bit 3 of port C.
26	/PORT D0	Input/Output	Bit 0 of port D (4 bit parallel I/O port)
27	/PORT D1	Input/Output	Bit 1 of port D.
28	/PORT D2	Input/Output	Bit 2 of port D.
29	/PORT D3	Input/Output	Bit 3 of port D.
30	PORT C STATUS	Output	Port C status signal. This signal is changed to the High level when the port C is configured to output port. It is changed to the Low level when the port is configured to input port.
31	PORT D STATUS	Output	Port D status signal. This signal is changed to the High level when the port D is configured to output port. It is changed to the Low level when the port is configured to input port.

32	/WRITE STROBE	Output	A output port write strobe signal. When data is present (that is, output level changes) on any of the output ports, this signal provides a negative pulse.
33	/PASS FAIL	Output	Each limit test's results signal. This signal changes to the High level when limit test, bandwidth test, or ripple test results return FAIL. It changes to the Low level when all limit test results return PASS.
34	/SWEEP END	Output	A sweep completion signal. When measurement (all sweeps of all channels) and data calculation are completed, this signal provides a negative pulse.
35	+5V	Output	Provides +5V DC power supply for external instruments.
36	/PASS FAIL STROBE	Output	Each limit test's results write a strobe signal. When limit test result is present on /PASS FAIL, this signal provides a negative pulse.

Other topics about Communication with External Devices


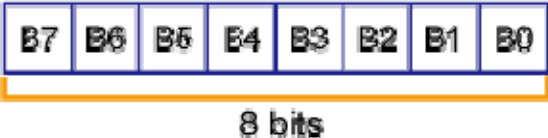



Inputting/Outputting Data

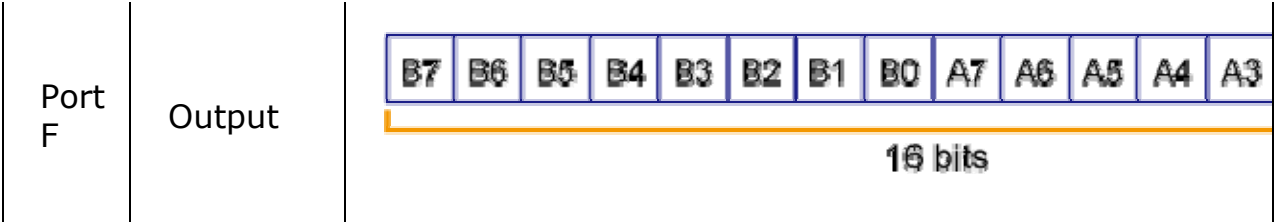
- [Overview](#)
- [Specifying Signal Direction of Port](#)
- [Reading Data Input from Port](#)
- [Data Output to Port](#)
- Sample Program

Other topics about Communication with External Devices

Overview

The E5071C handler I/O port provides the ports for data I/O shown below.

Port Name	Usage	Data Structure
Port A	Output	 <p>A7 A6 A5 A4 A3 A2 A1 A0 8 bits</p>
Port B	Output	 <p>B7 B6 B5 B4 B3 B2 B1 B0 8 bits</p>
Port C	Input/Output	 <p>C3 C2 C1 C0 4 bits</p>
Port D	Input/Output	 <p>D3 D2 D1 D0 4 bits</p>
Port E	Input/Output	 <p>D3 D2 D1 D0 C3 C2 C1 C0 8 bits</p>



Specifying Signal Direction of Port

Signal direction (input/output) can be changed for the ports C, D, and E as shown in I/O ports and control signal lines. Thus, before the ports are used, the directions should be determined according to their usage.

To specify the signal direction for the ports C and D, use the following command. Direction for the port E depends on the setting for the ports C and D.

Port Name	Command
Port C	:CONT:HAND:C:MODE
Port D	:CONT:HAND:D:MODE

Reading Data Input into Port

When the ports C, D, or E is configured to input ports, binary data represented with High(0)/Low(1) of each bit of the port will be read as decimal data.

To retrieve the data, use the following command as query:

Port Name	Command
Port C	:CONT:HAND:C
Port D	:CONT:HAND:D
Port E	:CONT:HAND:E

Data Output to Port

To the ports A through F (the ports C, D, and E should be configured to output ports), binary data (decimal data when output data is specified with a command) represented with High(0)/Low(1) of each bit of the port can be output.

To output data, use the following command:

Port Name	Command
-----------	---------

Port A	:CONT:HAND:A
Port B	:CONT:HAND:B
Port C	:CONT:HAND:C
Port D	:CONT:HAND:D
Port E	:CONT:HAND:E
Port F	:CONT:HAND:F

- The bit 6 of the data output by :CONT:HAND:B (the bit 14 of the data output by :CONT:HAND:F) is ignored when outputting the /INDEX signal is turned on.
- The bit 7 of the data output by :CONT:HAND:B (the bit 15 of the data output by :CONT:HAND:F command) is ignored when outputting the /READY FOR TRIGGER signal is turned on.

Sample Program

See Handler Interface.

Preset states at power-on

The handler I/O port is set at power-on as follows (not affected at reset)

Description	Status
Port A	High (All Bits)
Port B	High (All Bits)
Port C	Input
Port D	Input
Port C STATUS	Low
Port D STATUS	Low
/OUTPUT1	High
/OUTPUT2	High
/SWEEP END	High
/PASS FAIL	High

Other topics about Communication with External Devices

Timing Chart

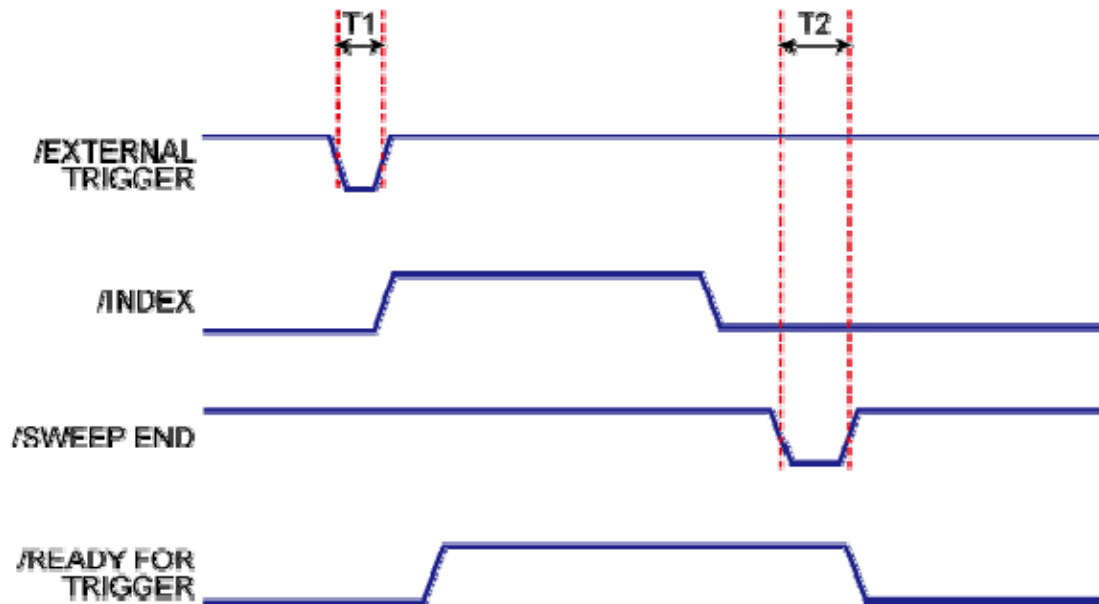
- [Overview](#)
- [Timing Chart of I/O Port Signal](#)
- [Timing Chart of Data Output and Write Strobe Signal](#)
- [Timing Chart of Limit Test Result Output and Write Strobe Signal](#)
- [Timing Chart of /INPUT1 and /OUTPUT1, /OUTPUT2](#)

Other topics about Communication with External Devices

Overview

This section shows the typical timing chart of I/O port Signal.

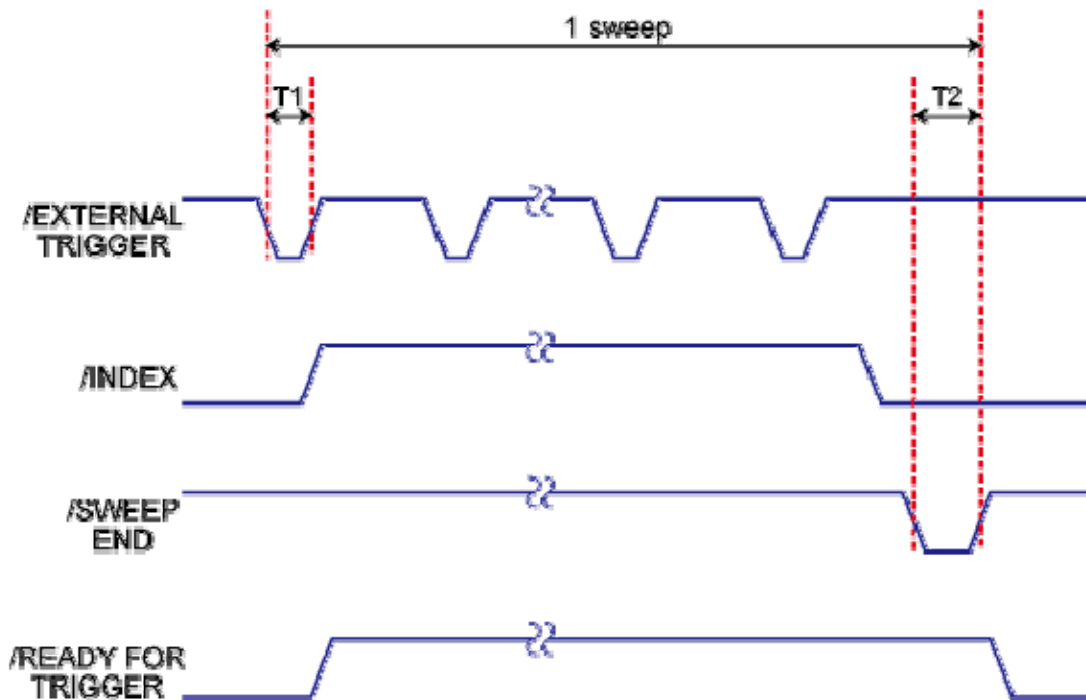
Timing chart of I/O Port Signal (Point trigger function: off)



e5071c368

		Minimum value	Typical Value	Maximum value
T1	Pulse width of /EXTERNAL TRIGGER	1 μ s	-	-
T2	Pulse width of /SWEEP END	10 μ s	12 μ s	-

Timing Chart of I/O Port Signal (Point trigger function: on, Low-latency external trigger mode: on)



e5071c309

When both the point trigger function and the low-latency external trigger mode are on, the /EXTERNAL TRIGGER signal must be inputted for each measurement point during a single sweep. The /INDEX signal goes to the High level before starting measurement of the first measurement point and returns to the Low level after the completing measurement of all measurement points.

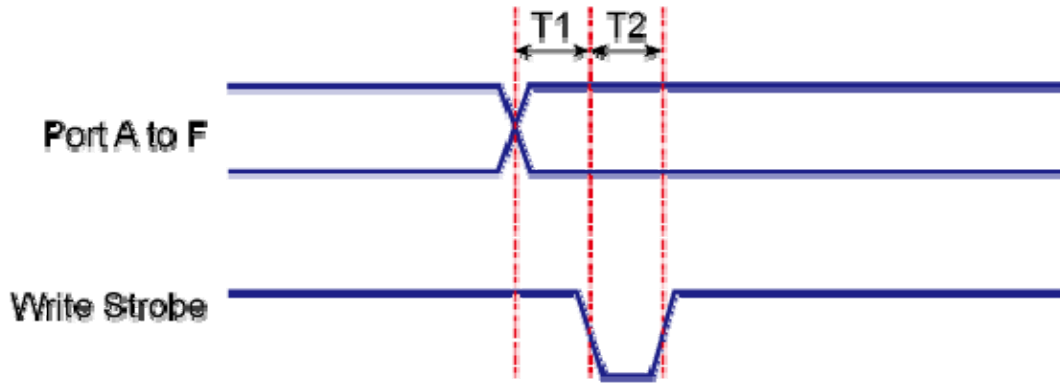
The /READY FOR TRIGGER signal goes to the Low level when the instrument is ready to accept the trigger signal for the first point and then goes to the High level when the trigger signal for the first point is received.

When measurement of all measurement points is completed and the instrument is ready to receive the trigger signal for the first point of the next sweep, this signal goes to the Low level again.

When the point trigger function is on and the low-latency external trigger mode is off, the /READY FOR TRIGGER signal goes High each time a trigger signal is received and goes Low when measurement of each measurement point is completed and the instrument is ready to accept a trigger for the next measurement point.

The times of T1 and T2 are the same as those when the point trigger function is off. For more information, see Timing chart of I/O Port Signal(Point trigger function:off).

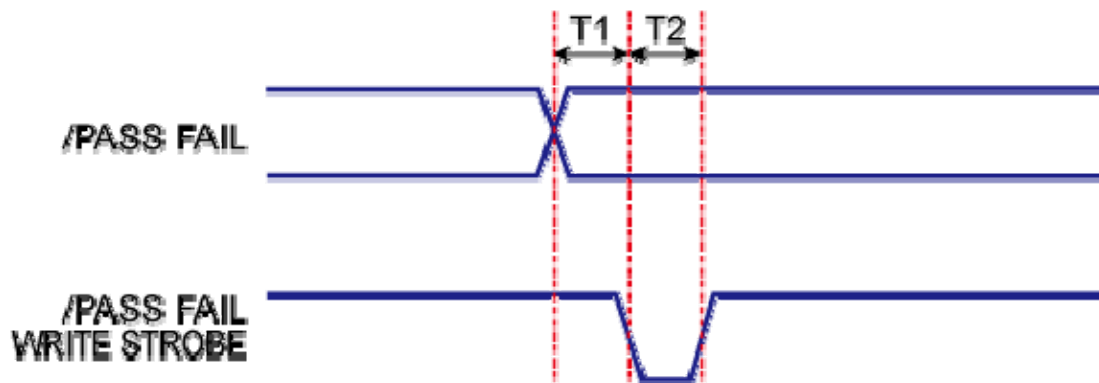
Timing Chart of Data Output and Write Strobe Signal



e50710366

T1	Response time of write strobe signal	1 μ s
T2	Pulse width of write strobe signal	1 μ s

Timing Chart of Limit Test Result Output and Write Strobe Signal

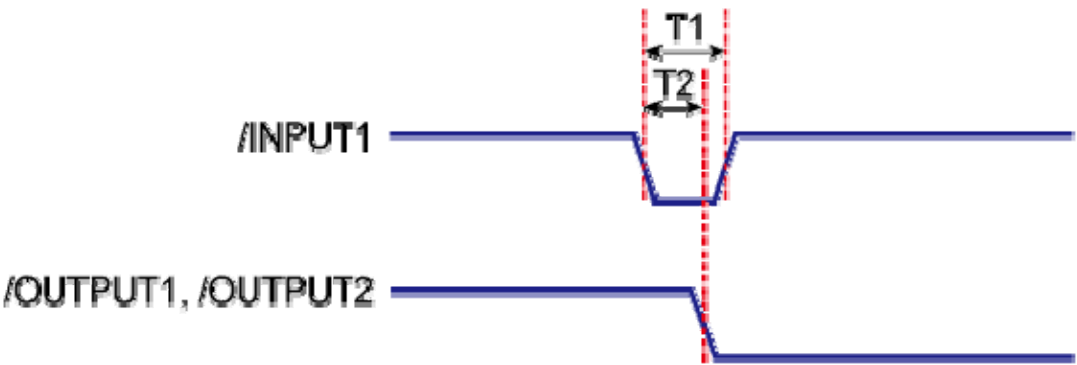


e50710367

T1	Response time of /PASS FAIL write strobe	1 μ s
T2	Pulse width of /PASS FAIL write strobe	1 μ s

- When the average trigger function is activated, the fail and write strobe signals are output at the time that the average test result shows "failed" on a certain channel.

Timing Chart of /INPUT1 and /OUTPUT1, /OUTPUT2



e5071c368

		Minimum value	Maximum value
T1	Pulse width of /INPUT1	1 μ s	-
T2	Response time of /OUTPUT1, /OUTPUT2	0.4 μ s	0.6 μ s

Electrical Characteristics

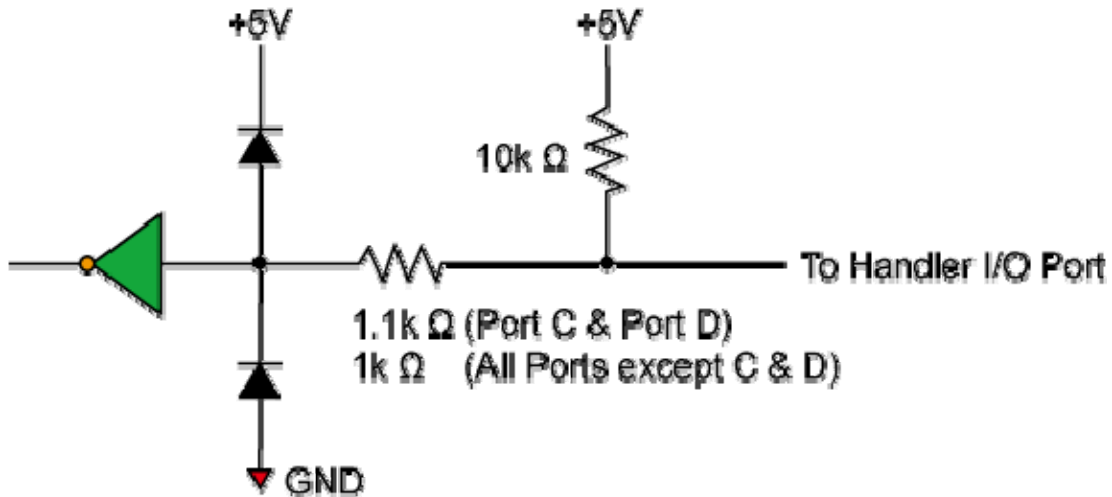
- [Input Signal](#)
- [Output Signal](#)
- [Power Supply \(+5 V\)](#)

Other topics about Communication with External Devices

Input Signal

All input signals are TTL compatible.

Maximum rate input voltage		-0.5 V to 5.5 V
Input voltage	High level	2.0 V to 5.0 V
	Low level	0 V to 0.5 V



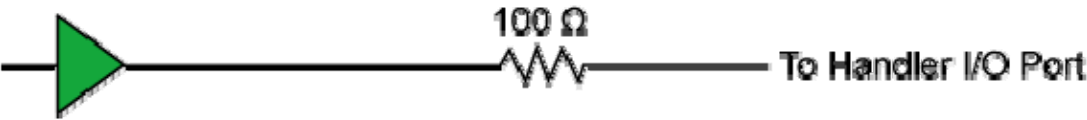
e5071c370

Output Signal

All output signals are TTL compatible.

Maximum rate output current		-10 mA to 10 mA
Output current	High level	-5 mA
	Low level	3 mA

Output voltage	High level	2.0 V to 3.3 V (when output current is from -5 mA to 0 mA) 3.20 V (when output current is -1 mA) 2.75 V (when output current is -5 mA)
	Low level	0 V to 0.8 V (when output current is from 0 mA to 3 mA) 0.25 V (when output current is 1 mA) 0.55 V (when output current is 3 mA)



e5071c371

Power Supply (+5 V)

The following table shows electrical characteristics of +5 V power supply for external instruments.

Output voltage	4.5 V to 5.5 V
Maximum output current	100 mA

Status Reporting System

Status Reporting System

- General Status Register Model
- Using the Status Reporting System
- Status Register Structure

General Status Register Model

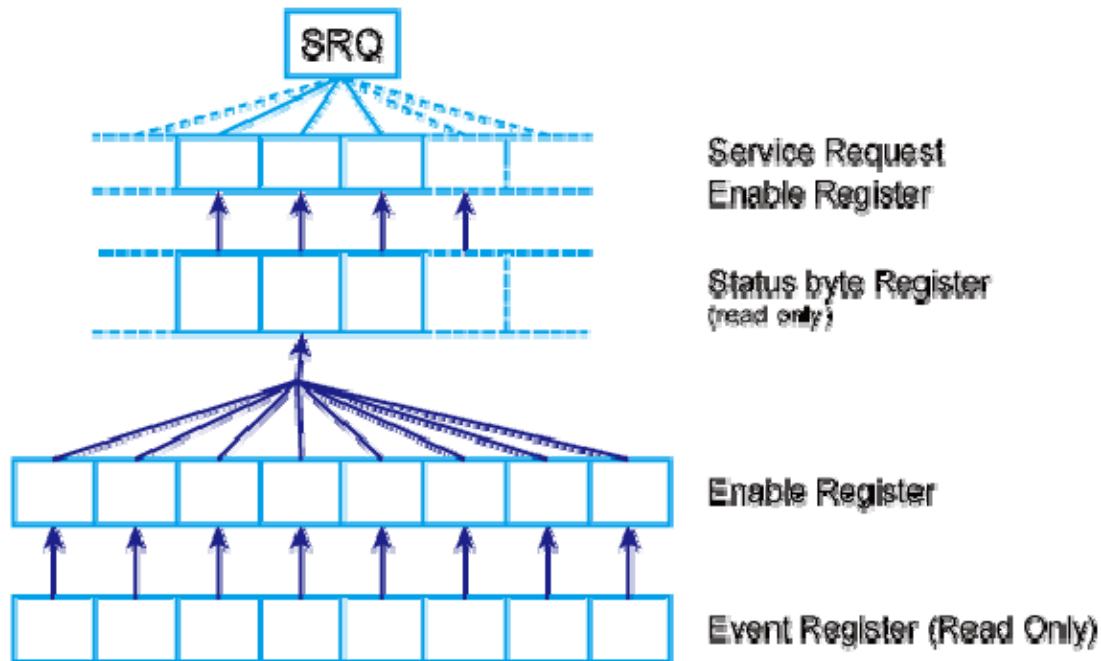
- [Overview](#)
- [Event Register](#)
- [Enable Register](#)
- [Status Byte Register](#)
- [Condition Register and Transition Filter](#)

Other topics about Status Reporting System

Overview

The Agilent E5071C has a status reporting system to report the condition of the instrument.

General status register model



e5071c479

The status reporting system has a hierarchical structure as shown in the figure above. When the instrument satisfies a particular condition, the corresponding bit of the event register is set to 1. Therefore, you can check the instrument status by reading the event register.

When the event register bit is set to "1" and a corresponding enable register bit (a bit marked with an arrow in General status register model) is also "1," the summary bit of the status byte register is set to "1." You can read the status byte register by using the serial poll.

If the bit of the service request enable register is "1," a service request (SRQ) is generated by the positive transition of the corresponding status byte register bit. By generating SRQ, you can notify the controller that the E5071C is requesting service. In other words, interruption by SRQ can be programmed. For more information on using SRQ, see Using the status register or Using the status reporting system.

Event Register

Reflects the corresponding condition of the E5071C (e.g., occurrence of an event) as a bit status. These bits continuously monitor changes in the E5071C's state and change the bit status when the condition (e.g., change bit status to "1" if a specific event occurs) for each bit is met. You cannot change the bit status by issuing a SCPI command.

Enable Register

Setting the enable register allows you to specify event register bits that can set "1" to the summary bit of the status byte register when an event occurs. The register bits work as mask bits; setting "1" to an enable register will enable a corresponding bit in the event register.

For example, when you want to set "1" as the summary bit in the status byte register by a specific register condition, set the corresponding enable register to "1."

Status Byte Register

If the enabled event register is set to "1," a corresponding bit of the status byte register is also set to "1." This register also indicates the output queue and SRQ status.

The value of the status byte register can be read by using the ***STB?** command or serial poll (SPOLL statement in HTBasic) from the controller.

Reading the status byte register by using the ***STB?** command does not affect the contents of the status byte register. However, reading it with the SPOLL statement of HTBasic will clear the RQS bit in the status byte register.

Also, setting the service request enable register using the ***SRE** command can generate a service request synchronously with the status byte register.

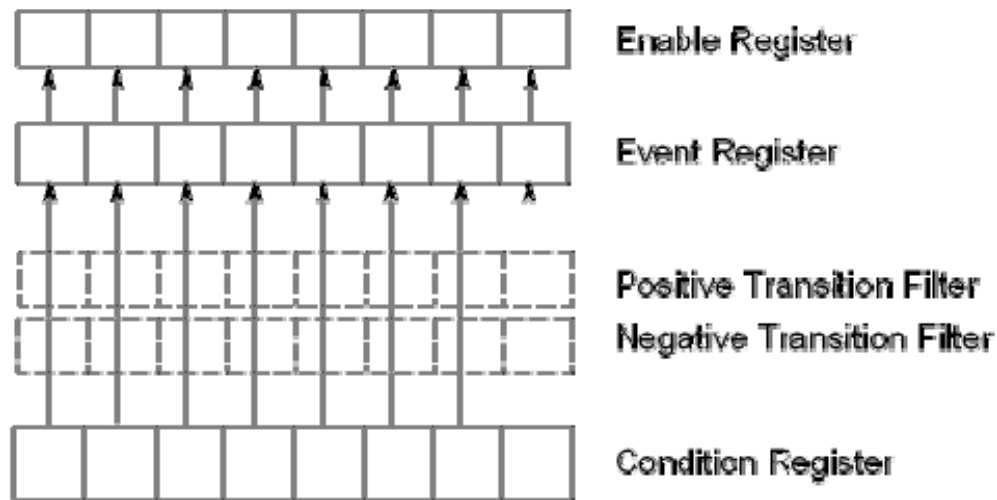
Condition Register and Transition Filter

When the status register has a transition filter, there is a lower register called a condition register under the event register. The transition filter is between the event register and the condition register.

The transition filter enables you to select a positive and/or negative transition of the condition register bit in order to set a bit in the corresponding event register. For example, using the negative transition filter to set bit 3 to "1" causes bit 3 of the event register to be set to "1"

when bit 3 of the condition register makes a negative transition, that is, changes from 1 to 0.

Transition filter and condition register



1294apr022

In the E5071C, the following registers provide a condition register and transition filter:

- Operation status register
- Questionable status register
- Questionable limit status register
- Questionable limit extra status register
- Questionable limit channel {1-16} status register
- Questionable limit channel {1-16} extra status register
- Questionable bandwidth limit status register
- Questionable bandwidth limit extra status register
- Questionable bandwidth limit channel {1-16} status register
- Questionable bandwidth limit channel {1-16} extra status register
- Questionable ripple limit status register
- Questionable ripple limit extra status register
- Questionable ripple limit channel {1-16} status register
- Questionable ripple limit channel {1-16} extra status register

Using the Status Reporting System

You can manage the status report system using the following commands in any combination:

- *CLS
- *SRE
- *STB?
- *ESE
- *ESR?
- :STAT:PRES
- :STAT:OPER:ENAB
- :STAT:OPER:COND?
- :STAT:OPER?
- :STAT:OPER:PTR
- :STAT:OPER:NTR
- :STAT:QUES:ENAB
- :STAT:QUES:COND?
- :STAT:QUES?
- :STAT:QUES:PTR
- :STAT:QUES:NTR
- :STAT:QUES:LIM:ENAB
- :STAT:QUES:LIM:COND?
- :STAT:QUES:LIM?
- :STAT:QUES:LIM:PTR
- :STAT:QUES:LIM:NTR
- :STAT:QUES:LIM:ELIM:ENAB
- :STAT:QUES:LIM:ELIM:COND?
- :STAT:QUES:LIM:ELIM?
- :STAT:QUES:LIM:ELIM:PTR
- :STAT:QUES:LIM:ELIM:NTR
- :STAT:QUES:LIM:CHAN{1-16}:ENAB
- :STAT:QUES:LIM:CHAN{1-16}:COND?
- :STAT:QUES:LIM:CHAN{1-16}?
- :STAT:QUES:LIM:CHAN{1-16}:PTR
- :STAT:QUES:LIM:CHAN{1-16}:NTR

- :STAT:QUES:LIM:CHAN{1-16}:ECH:ENAB
- :STAT:QUES:LIM:CHAN{1-16}:ECH:COND?
- :STAT:QUES:LIM:CHAN{1-16}:ECH?
- :STAT:QUES:LIM:CHAN{1-16}:ECH:PTR
- :STAT:QUES:LIM:CHAN{1-16}:ECH:NTR
- :STAT:QUES:BLIM:ENAB
- :STAT:QUES:BLIM:COND?
- :STAT:QUES:BLIM?
- :STAT:QUES:BLIM:PTR
- :STAT:QUES:BLIM:NTR
- :STAT:QUES:BLIM:ELIM:ENAB
- :STAT:QUES:BLIM:ELIM:COND?
- :STAT:QUES:BLIM:ELIM?
- :STAT:QUES:BLIM:ELIM:PTR
- :STAT:QUES:BLIM:ELIM:NTR
- :STAT:QUES:BLIM:CHAN{1-16}:ENAB
- :STAT:QUES:BLIM:CHAN{1-16}:COND?
- :STAT:QUES:BLIM:CHAN{1-16}?
- :STAT:QUES:BLIM:CHAN{1-16}:PTR
- :STAT:QUES:BLIM:CHAN{1-16}:NTR
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:ENAB
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:COND?
- :STAT:QUES:BLIM:CHAN{1-16}:ECH?
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:PTR
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:NTR
- :STAT:QUES:RLIM:ENAB
- :STAT:QUES:RLIM:COND?
- :STAT:QUES:RLIM?
- :STAT:QUES:RLIM:PTR
- :STAT:QUES:RLIM:NTR
- :STAT:QUES:RLIM:ELIM:ENAB
- :STAT:QUES:RLIM:ELIM:COND?
- :STAT:QUES:RLIM:ELIM?
- :STAT:QUES:RLIM:ELIM:PTR

- :STAT:QUES:RLIM:ELIM:NTR
- :STAT:QUES:RLIM:CHAN{1-16}:ENAB
- :STAT:QUES:RLIM:CHAN{1-16}:COND?
- :STAT:QUES:RLIM:CHAN{1-16}?
- :STAT:QUES:RLIM:CHAN{1-16}:PTR
- :STAT:QUES:RLIM:CHAN{1-16}:NTR
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:ENAB
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:COND?
- :STAT:QUES:RLIM:CHAN{1-16}:ECH?
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:PTR
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:NTR

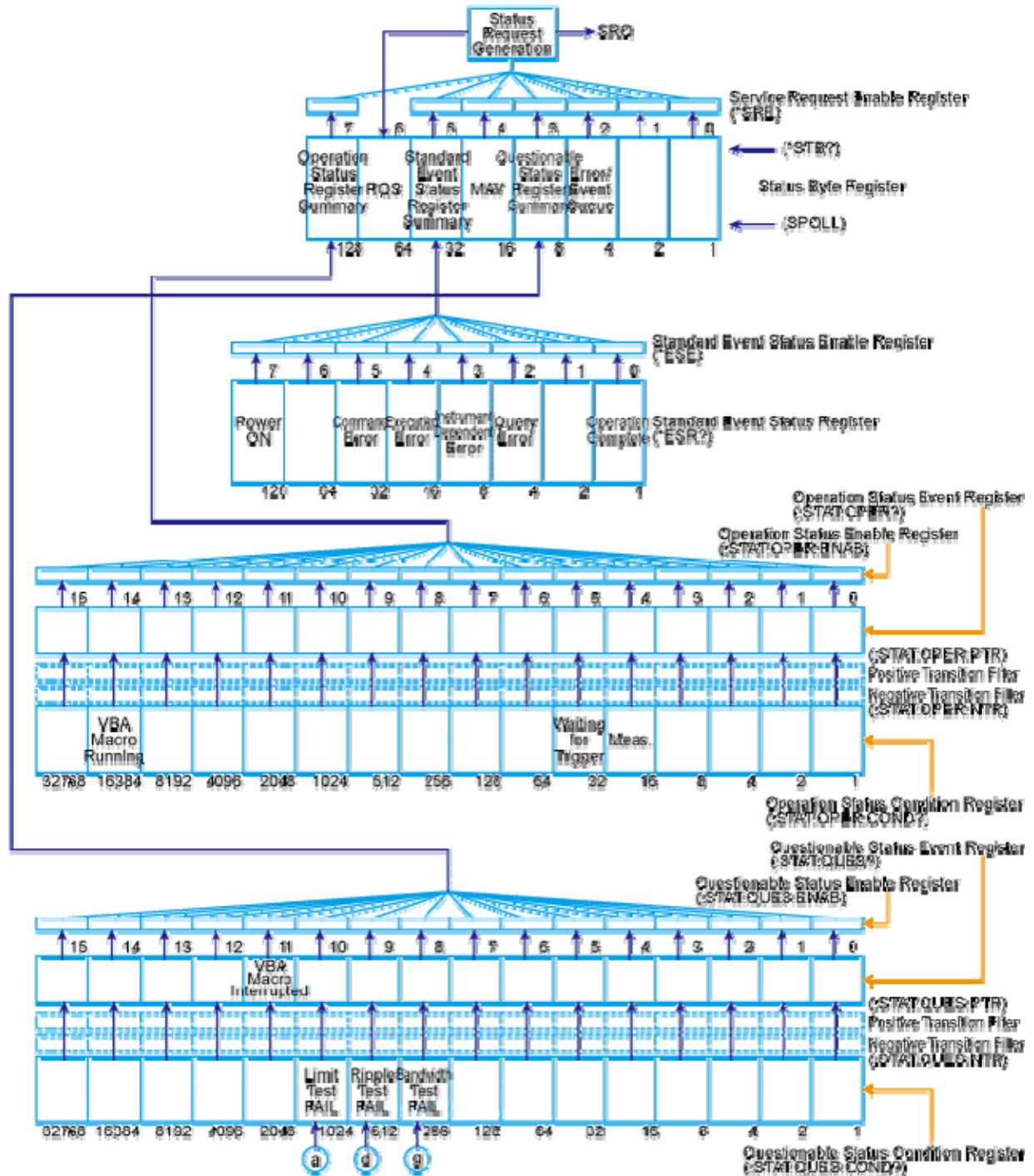
Other topics about Status Reporting System

Status Register Structure

Status Register Structure

- Status Register
- Status Register for Limit Test (Channel)
- Status Register for Limit Test (Trace) (1 of 2)
- Status Register for Limit Test (Trace) (2 of 2)
- Status Register for Bandwidth Limit (Channel)
- Status Register for Bandwidth Limit (Trace) (1 of 2)
- Status Register for Bandwidth Limit (Trace) (2 of 2)
- Status Register for Ripple Limit (Channel)
- Status Register for Ripple Limit (Trace) (1 of 2)
- Status Register for Ripple Limit (Trace) (2 of 2)
 - Although up to 36 channel is available on E5071C, the registers for limit test and Ripple/Bandwidth limit tests are for up to 16 channel.

Status Register



e5071e220

Status Bit Definitions of Status Byte Register

Bit	Name	Description
-----	------	-------------

Position		
0, 1	Not used	Always 0
2	Error/Event Queue	Set to "1" if the error/event queue contains data; reset to "0" when all the data has been retrieved.
3	Questionable Status Register Summary	Set to "1" when one of the enabled bits in the questionable status register is set to "1."
4	MAV (Message Available)	Set to "1" when the output queue contains data; reset to "0" when all the data has been retrieved.
5	Standard Event Status Register Summary	Set to "1" when one of the enabled bits in the standard event status register is set to "1."
6	RQS	Set to "1" when any of the status byte register bits enabled by the service request enable register is set to "1"; reset to "0" when all the data has been retrieved through serial polling.
7	Operation Status Register Summary	Set to "1" when one of the enabled bits in the operational status register is set to "1."

Issuing the *CLS command will clear all bits from the status byte register.

Status Bit Definitions of Standard Event Status Register

Bit Position	Name	Description
0	Operation Complete	Set to "1" upon completion of all operations done by commands that precede the *OPC command.
1	Not used	Always 0
2	Query Error	<ol style="list-style-type: none"> Set to "1" when the E5071C receives a data output request but there is no data to output. Set to "1" when the data of the E5071C's output queue has been cleared because of a new message received before the completion of data output.

3	Instrument Dependent Error	Set to "1" when an error has occurred and the error is not a command, query, or execution error.
4	Execution Error	<ol style="list-style-type: none"> 1. Set to "1" when any parameter in an SCPI command exceeds its input range or is inconsistent with the E5071C's capabilities. 2. Set to "1" when an SCPI command cannot be properly executed due to some condition of the E5071C.
5	Command Error	<ol style="list-style-type: none"> 1. Set to "1" when an IEEE 488.2 syntax error occurs (a command sent to the E5071C does not follow the IEEE 488.2 syntax). Possible violations include the command parameter violating the E5071C listening formats or being unacceptable. 2. Set to "1" when a semantic error occurs. Possible causes include a command containing misspellings being sent to the E5071C or an IEEE 488.2 command not supported by the E5071C being sent. 3. Set to "1" when GET (Group Execution Trigger) is input while a program message is being received.
6	Not used	Always 0
7	Power ON	Set to "1" when the E5071C is powered ON, or when the firmware is restarted.

Issuing the ***CLS** command will clear all bits from the standard event status register.

Status Bit Definitions of the Operation Status Condition Register

Bit Position	Name	Description
0 - 3	Not used	Always 0
4	Measurement	Set to "1" during measurement
5	Waiting for Trigger	Set to "1" while the instrument is waiting for a trigger.
6 - 13	Not used	Always 0

14	VBA Macro Running	Set to "1" while a VBA macro is running.
15	Not used	Always 0

Issuing the *CLS command will clear all bits from the operation status event register.

Status Bit Definitions of the Questionable Status Condition Register

Bit Position	Name	Description
0 - 7	Not used	Always 0
8	Bandwidth Test Fail (Questionable bandwidth limit status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit status event register is set to "1."
9	Ripple Test Fail (Questionable ripple limit status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit status event register is set to "1."
10	Limit Test Fail (Questionable limit status register summary)	Set to "1" while one of the enabled bits in the questionable limit status event register is set to "1."
11 - 15	Not used	Always 0

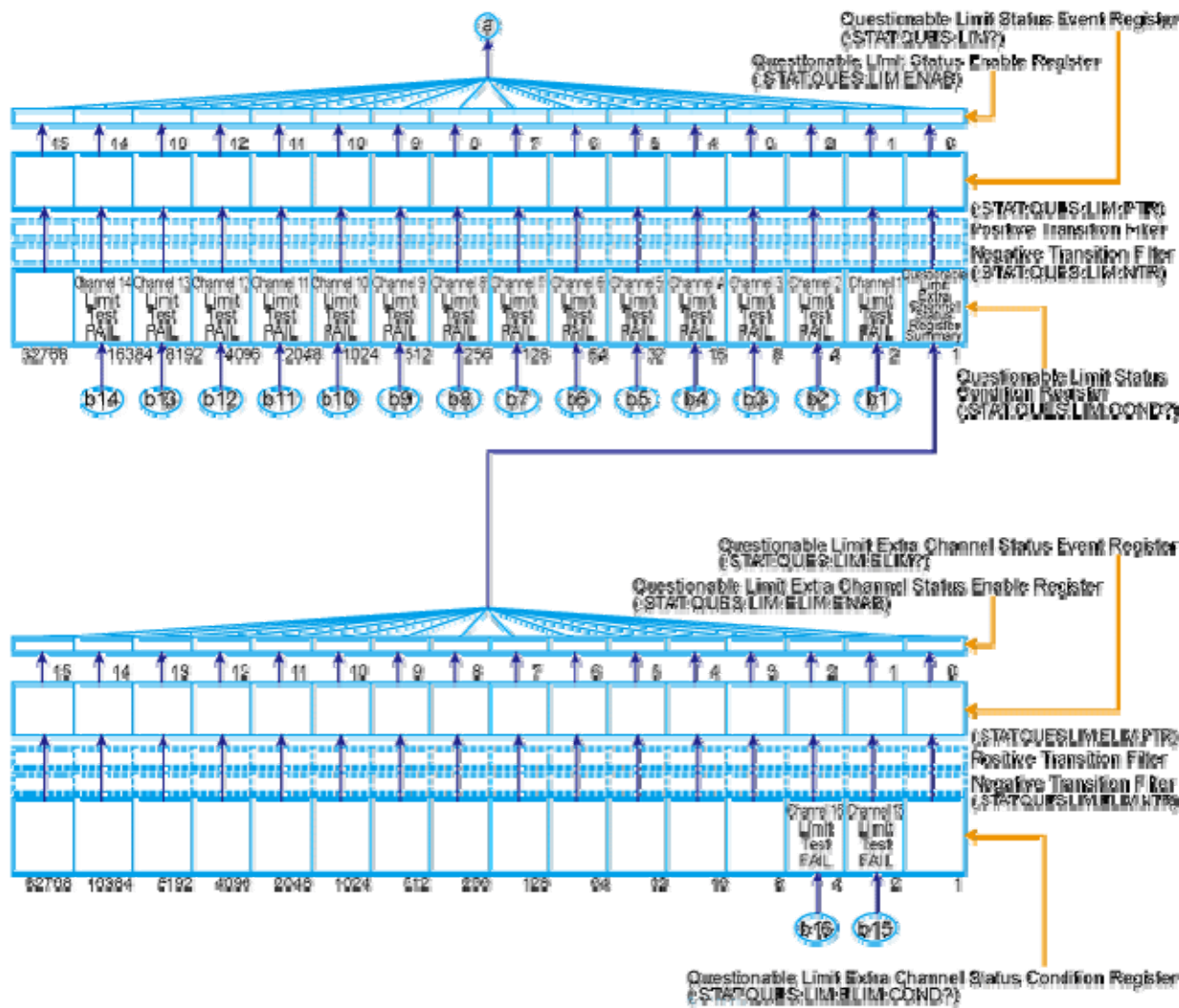
Status Bit Definitions of the Questionable Status Event Register

Bit Position	Name	Description
0 - 7	Not used	Always 0
8	Bandwidth Test Fail (Questionable bandwidth limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
9	Ripple Test Fail (Questionable ripple limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
10	Limit Test Fail	Set to "1" when a transition of the condition register

	(Questionable limit status register summary)	occurs if the transition filters are set as valid values.
11	VBA Macro Interrupted	<p>Set to "1" when a VBA macro is interrupted by one of the following reasons.</p> <p>Occurrence of an execution error</p> <p>Executing "End" statement in the VBA Macro</p> <p>Executing :PROG:STAT STOP</p> <p>Operating CTRL + Break using the keyboard</p> <p>Operating Macro Break or Macro Setup > Stop using the front panel</p>
12 - 15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable status event register.

Status Register for Limit Test (channel)



e5071c221

Status Bit Definitions of the Questionable Limit Status Condition Register

Bit Position	Name	Description
0	Channel 15, 16 Limit Test summary (questionable limit extra status register summary)	Set to "1" while one of the enabled bits in the questionable limit extra status event register is set to "1."
1	Channel 1 Limit Test Fail (questionable limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 1 status event register is set to "1."

2	Channel 2 Limit Test Fail (questionable limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 2 status event register is set to "1."
3	Channel 3 Limit Test Fail (questionable limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 3 status event register is set to "1."
4	Channel 4 Limit Test Fail (questionable limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 4 status event register is set to "1."
5	Channel 5 Limit Test Fail (questionable limit channel 5 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 5 status event register is set to "1."
6	Channel 6 Limit Test Fail (questionable limit channel 6 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 6 status event register is set to "1."
7	Channel 7 Limit Test Fail (questionable limit channel 7 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 7 status event register is set to "1."
8	Channel 8 Limit Test Fail (questionable limit channel 8 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 8 status event register is set to "1."
9	Channel 9 Limit Test Fail (questionable limit channel 9 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 9 status event register is set to "1."
10	Channel 10 Limit Test Fail (questionable limit channel 10 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 10 status event register is set to "1."
11	Channel 11 Limit Test Fail (questionable limit channel 11 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 11 status event register is set to "1."
12	Channel 12 Limit Test Fail (questionable limit channel 12 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 12 status event register is set to "1."
13	Channel 13 Limit Test Fail (questionable limit channel 13 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 13 status event register is set to "1."
14	Channel 14 Limit Test Fail (questionable limit channel 14 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 14 status event register is set to "1."
15	Not used	Always 0

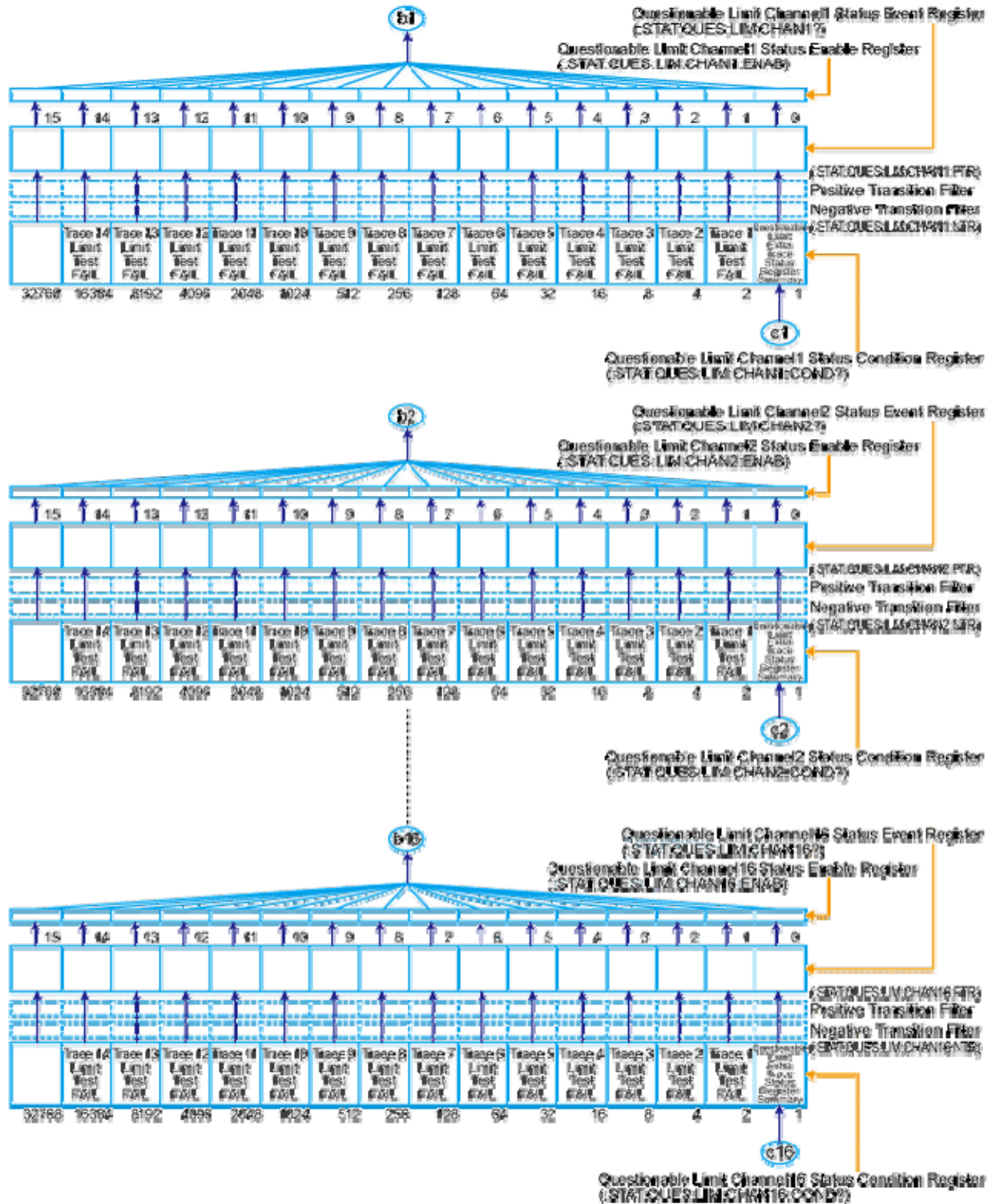
Issuing the ***CLS** command will clear all bits from the questionable limit status event register.

Status Bit Definitions of the Questionable Limit Extra Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 15 Limit Test Fail (questionable limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 15 status event register is set to "1."
2	Channel 16 Limit Test Fail (questionable limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 16 status event register is set to "1."
3 - 15	Not used	Always 0

Issuing the ***CLS** command will clear all bits from the questionable limit extra status event register.

Status Register for Limit Test (Trace) (1 of 2)



e5071c222

***Status Bit Definitions of the Questionable Limit Channel {1-16}
Status Condition Register***

Bit Position	Name	Description
0	Trace 15, 16 Limit Test summary (questionable limit channel {1-16} extra status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel {1-16} extra status event register is set to "1."
1	Trace 1 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 1.
2	Trace 2 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 2.
3	Trace 3 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 3.
4	Trace 4 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 4.
5	Trace 5 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 5.
6	Trace 6 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 6.
7	Trace 7 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 7.

8	Trace 8 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 8.
9	Trace 9 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 9.
10	Trace 10 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 10.
11	Trace 11 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 11.
12	Trace 12 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 12.
13	Trace 13 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 13.
14	Trace 14 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 14.
15	Not used	Always 0

Issuing the ***CLS** command will clear all the bits in the questionable limit channel {1-16} status event register.

***Status Bit Definitions of the Questionable Limit Channel {1-16}
Extra Status Condition Register***

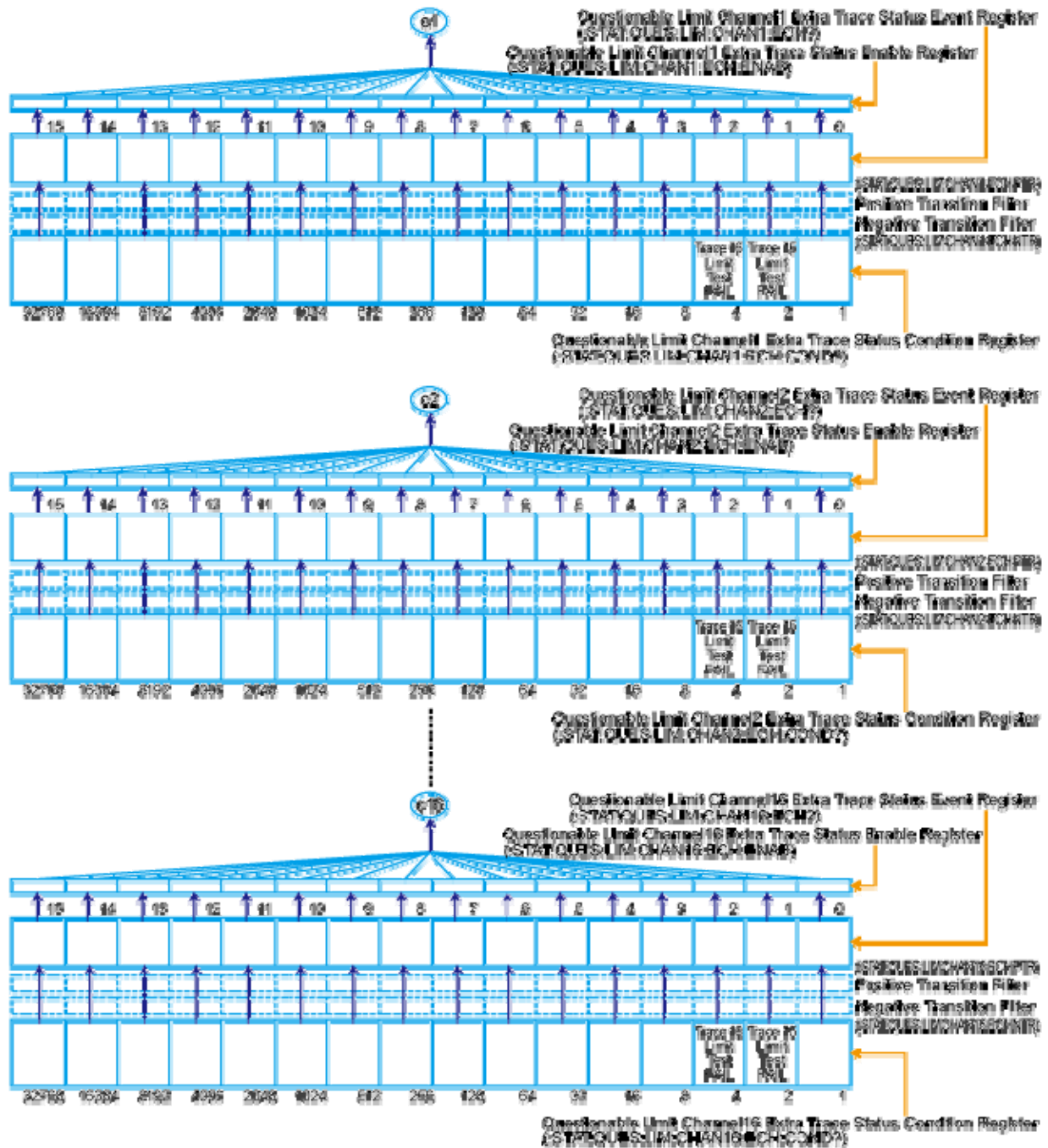
Bit Position	Name	Description
0	Not used	Always 0
1	Trace 15 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 15.
2	Trace 16 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 16.
3 - 15	Not used	Always 0

Issuing the ***CLS** command will clear all the bits in the questionable limit channel {1-16} extra status event register.

NOTE

Although up to 36 channel is available on E5071C, the register for limit test is for up to 16 channel.

Status Register for Limit Test (Trace) (2 of 2)



65071e223

Status Bit Definitions of the Questionable Limit Extra Status Condition Register

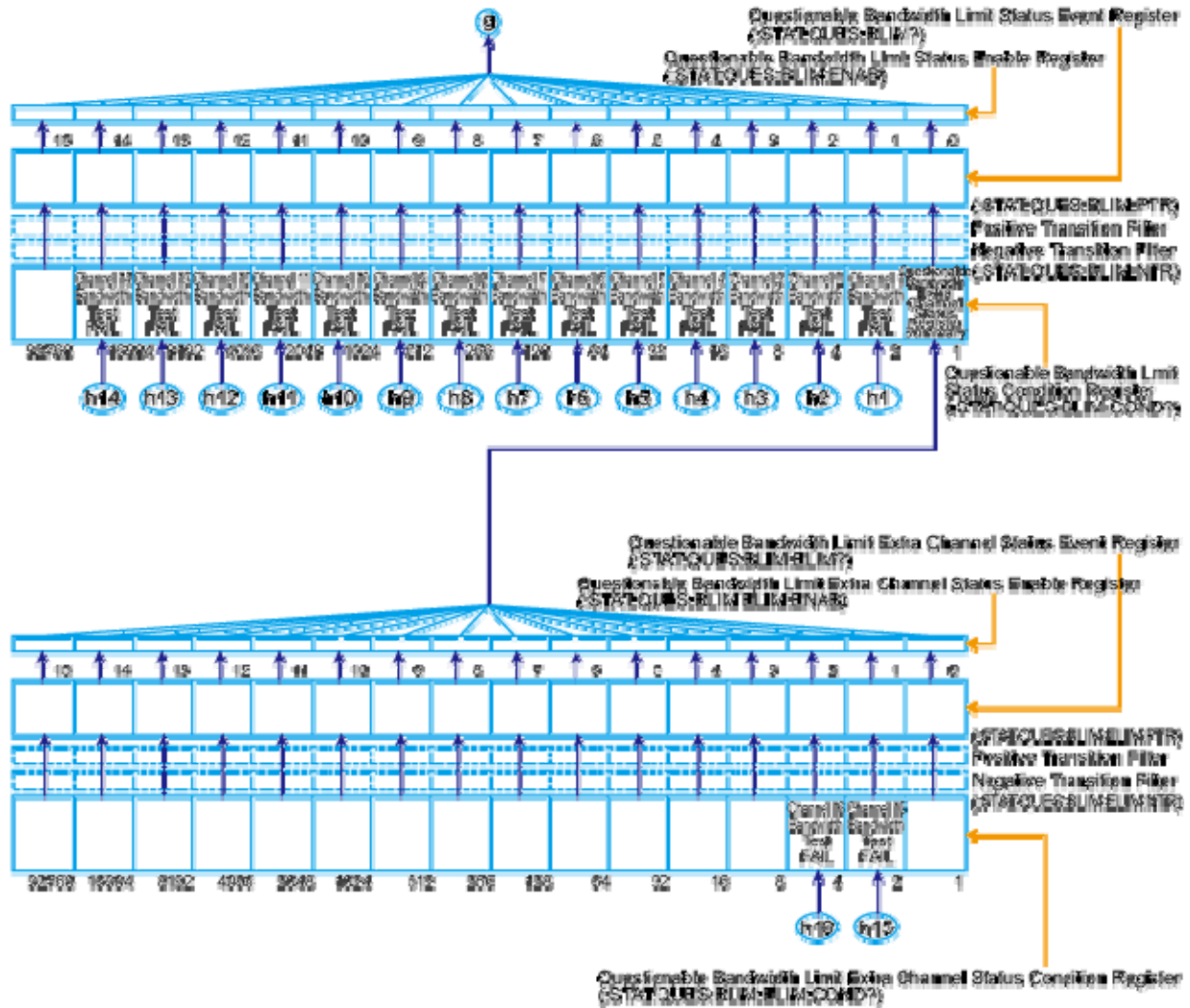
Bit Position	Name	Description
--------------	------	-------------

E5071C

0	Not used	Always 0
1	Channel 15 Limit Test Fail (questionable limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 15 status event register is set to "1."
2	Channel 16 Limit Test Fail (questionable limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 16 status event register is set to "1."
3 - 15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable limit extra status event register.

Status Register for Bandwidth Limit (Channel)



e5071c228

Status Bit Definitions of the Questionable Bandwidth Limit Channel {1-16} Status Condition Register

Bit Position	Name	Description
0	Trace 15, 16 Bandwidth Test summary (questionable bandwidth limit channel {1-16} extra status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel {1-16} extra status event register is set to "1."
1	Trace 1 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as

		the bandwidth test result for trace 1.
2	Trace 2 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 2.
3	Trace 3 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 3.
4	Trace 4 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 4.
5	Trace 5 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 5.
6	Trace 6 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 6.
7	Trace 7 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 7.
8	Trace 8 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 8.
9	Trace 9 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 9.
10	Trace 10 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement

		cycle finishes and returns "fail" as the bandwidth test result for trace 10.
11	Trace 11 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 11.
12	Trace 12 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 12.
13	Trace 13 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 13.
14	Trace 14 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 14.
15	Not used	Always 0

Issuing the ***CLS** command will clear all the bits in the questionable bandwidth limit channel {1-16} status event register.

Status Bit Definitions of the Questionable Bandwidth Limit Channel {1-16} Extra Status Condition Register

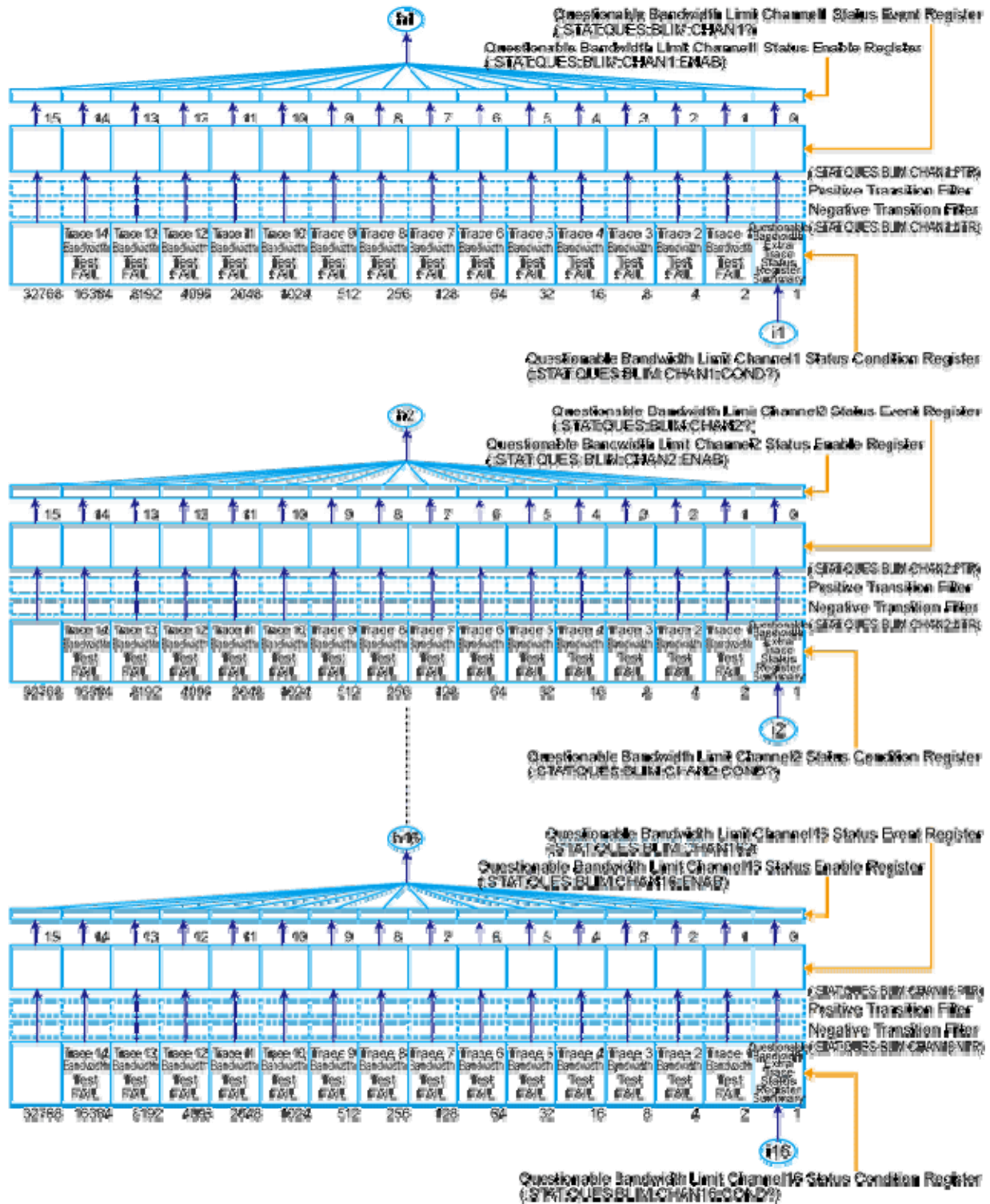
Bit Position	Name	Description
0	Not used	Always 0
1	Trace 15 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 15.
2	Trace 16 Bandwidth Test	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace

E5071C

	Fail	16.
3 - 15	Not used	Always 0

Issuing the [*CLS](#) command will clear all the bits in the questionable bandwidth limit channel {1-16} extra status event register.

Status Register for Bandwidth Limit (Trace) 1



e5071c227

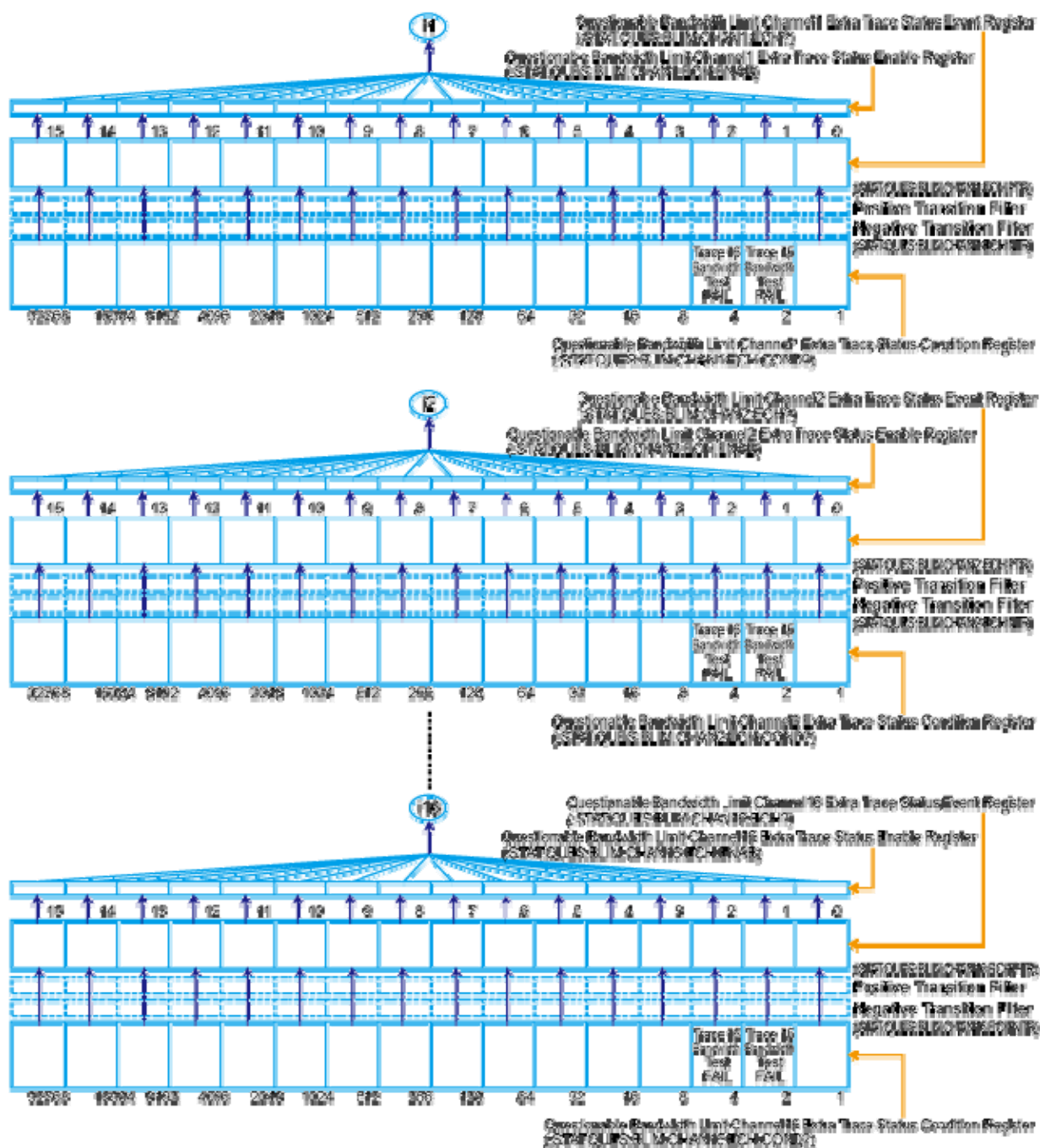
Status Bit Definitions of the Questionable Bandwidth Limit Status Condition Register

Bit Position	Name	Description
0	Channel 15, 16 Bandwidth test summary (questionable bandwidth limit extra status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit extra status event register is set to "1."
1	Channel 1 Bandwidth Test Fail (questionable bandwidth limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 1 status event register is set to "1."
2	Channel 2 Bandwidth Test Fail (questionable bandwidth limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 2 status event register is set to "1."
3	Channel 3 Bandwidth Test Fail (questionable bandwidth limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 3 status event register is set to "1."
4	Channel 4 Bandwidth Test Fail (questionable bandwidth limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 4 status event register is set to "1."
5	Channel 5 Bandwidth Test Fail (questionable bandwidth limit channel 5 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 5 status event register is set to "1."
6	Channel 6 Bandwidth Test Fail (questionable bandwidth limit channel 6 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 6 status event register is set to "1."
7	Channel 7 Bandwidth Test Fail (questionable bandwidth limit channel 7 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 7 status event register is set to "1."
8	Channel 8 Bandwidth Test Fail (questionable bandwidth limit channel 8 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 8 status event register is set to "1."
9	Channel 9 Bandwidth Test Fail (questionable bandwidth limit channel 9 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 9 status event register is set to "1."

10	Channel 10 Bandwidth Test Fail (questionable bandwidth limit channel 10 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 10 status event register is set to "1."
11	Channel 11 Bandwidth Test Fail (questionable bandwidth limit channel 11 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 11 status event register is set to "1."
12	Channel 12 Bandwidth Test Fail (questionable bandwidth limit channel 12 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 12 status event register is set to "1."
13	Channel 13 Bandwidth Test Fail (questionable bandwidth limit channel 13 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 13 status event register is set to "1."
14	Channel 14 Bandwidth Test Fail (questionable bandwidth limit channel 14 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 14 status event register is set to "1."
15	Not used	Always 0

Issuing the ***CLS** command will clear all bits from the questionable bandwidth limit status event register.

Status Register for Bandwidth Limit (Trace) 2



03/16/2020

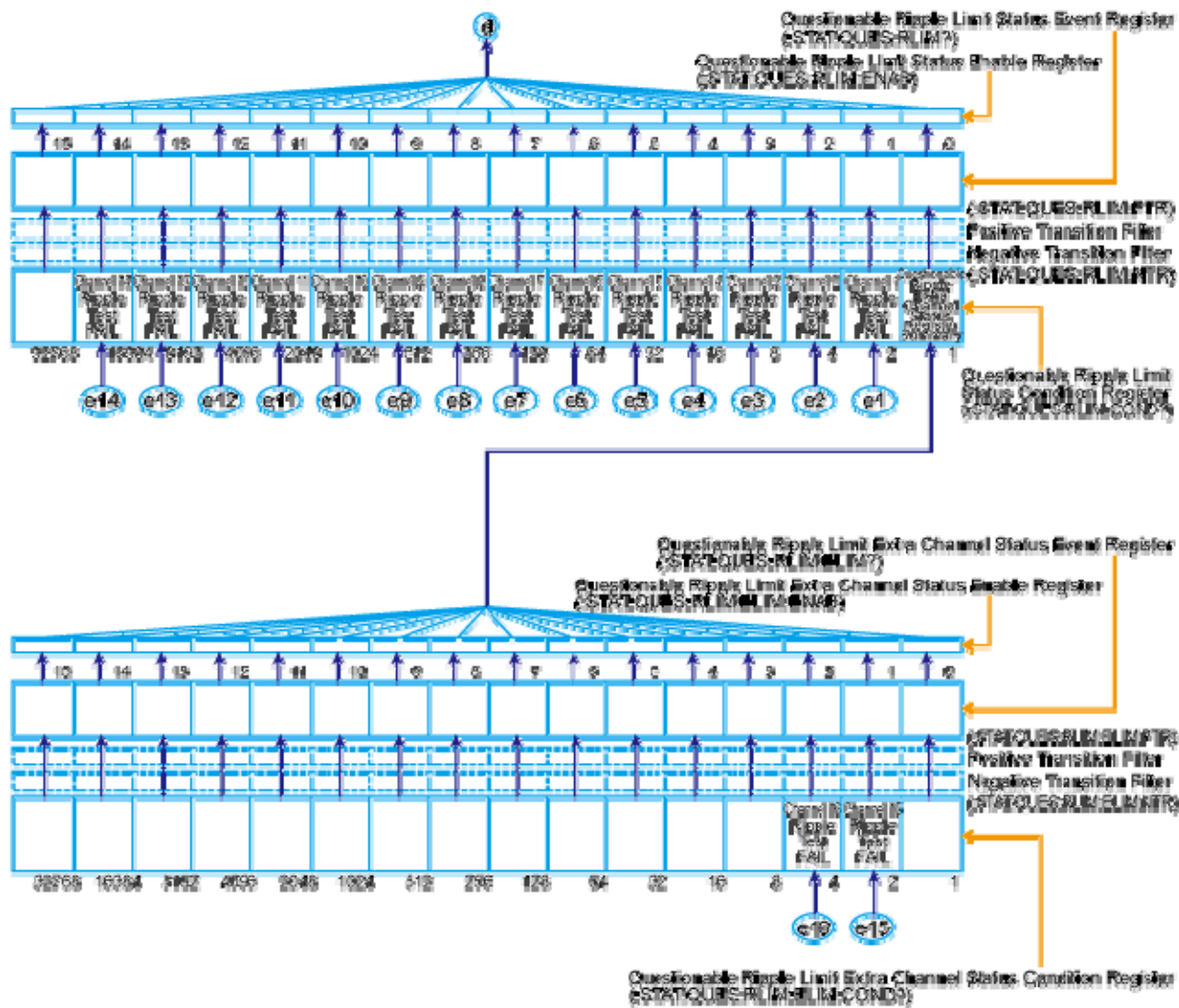
Status Bit Definitions of the Questionable Bandwidth Limit Extra Status Condition Register

Bit Position	Name	Description
--------------	------	-------------

0	Not used	Always 0
1	Channel 15 Bandwidth Test Fail (questionable bandwidth limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 15 status event register is set to "1."
2	Channel 16 Bandwidth Test Fail (questionable bandwidth limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 16 status event register is set to "1."
3 - 15	Not used	Always 0

Issuing the [*CLS](#) command will clear all bits from the questionable bandwidth limit extra status event register.

Status Register for Ripple Limit (Channel)



e5071c224

Status Bit Definitions of the Questionable Ripple Limit Status Condition Register

Bit Position	Name	Description
0	Channel 15, 16 Ripple Test summary (questionable ripple limit extra status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit extra status event register is set to "1."
1	Channel 1 Ripple Test Fail (questionable ripple limit channel 1 status register)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 1 status event register is set

	summary)	to "1."
2	Channel 2 Ripple Test Fail (questionable ripple limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 2 status event register is set to "1."
3	Channel 3 Ripple Test Fail (questionable ripple limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 3 status event register is set to "1."
4	Channel 4 Ripple Test Fail (questionable ripple limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 4 status event register is set to "1."
5	Channel 5 Ripple Test Fail (questionable ripple limit channel 5 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 5 status event register is set to "1."
6	Channel 6 Ripple Test Fail (questionable ripple limit channel 6 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 6 status event register is set to "1."
7	Channel 7 Ripple Test Fail (questionable ripple limit channel 7 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 7 status event register is set to "1."
8	Channel 8 Ripple Test Fail (questionable ripple limit channel 8 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 8 status event register is set to "1."
9	Channel 9 Ripple Test Fail (questionable ripple limit channel 9 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 9 status event register is set to "1."
10	Channel 10 Ripple Test Fail (questionable ripple limit channel 10 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 10 status event register is set to "1."
11	Channel 11 Ripple Test Fail (questionable ripple limit channel 11 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 11 status event register is set to "1."
12	Channel 12 Ripple Test Fail	Set to "1" while one of the enabled

	(questionable ripple limit channel 12 status register summary)	bits in the questionable ripple limit channel 12 status event register is set to "1."
13	Channel 13 Ripple Test Fail (questionable ripple limit channel 13 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 13 status event register is set to "1."
14	Channel 14 Ripple Test Fail (questionable ripple limit channel 14 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 14 status event register is set to "1."
15	Not used	Always 0

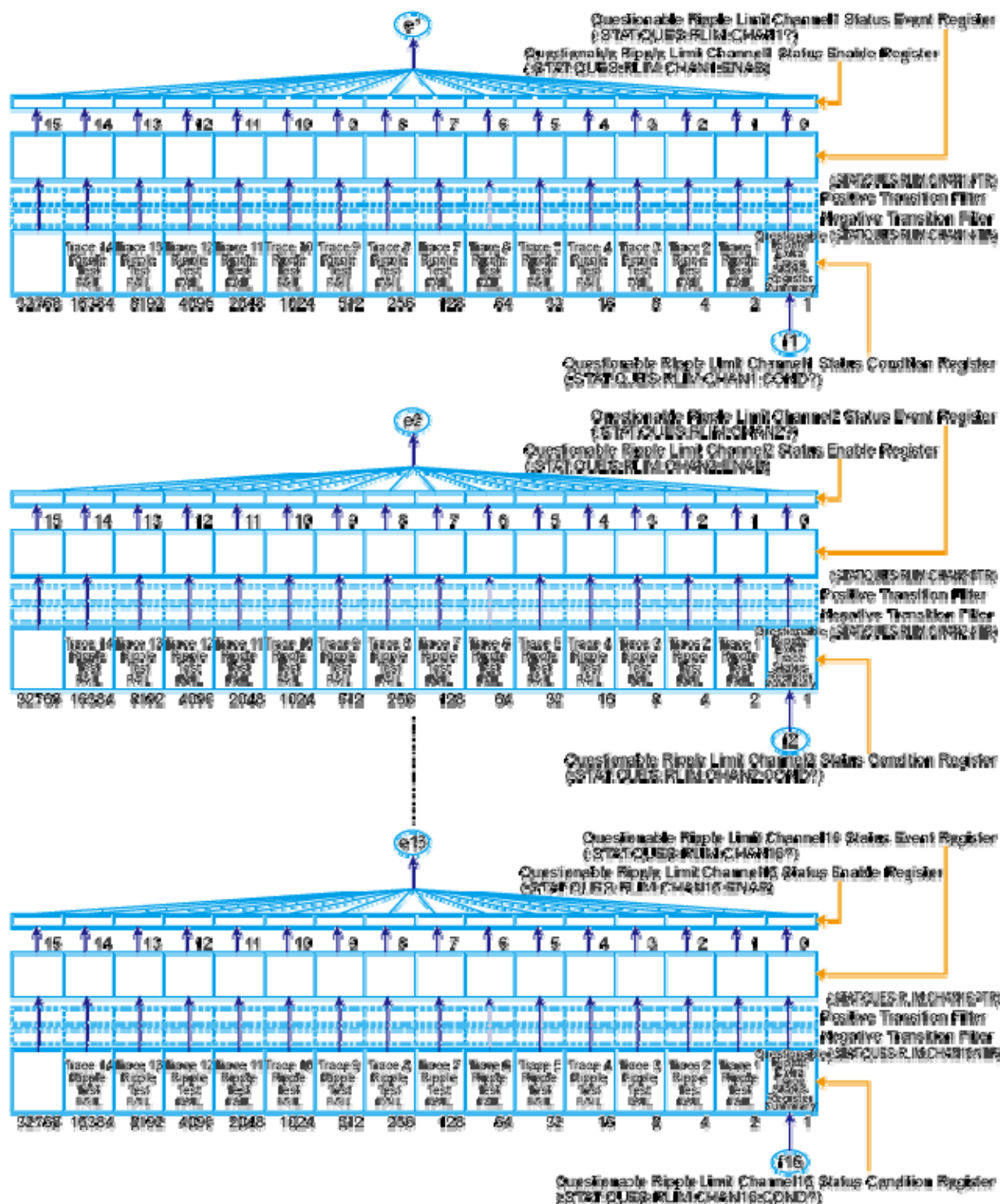
Issuing the ***CLS** command will clear all bits from the questionable ripple limit status event register.

Status Bit Definitions of the Questionable Ripple Limit Extra Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 15 Ripple Test Fail (questionable ripple limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 15 status event register is set to "1."
2	Channel 16 Ripple Test Fail (questionable ripple limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 16 status event register is set to "1."
3-15	Not used	Always 0

Issuing the ***CLS** command will clear all bits from the questionable ripple limit extra status event register.

Status Register for Ripple Limit (Trace) (1 of 2)



e5071e024

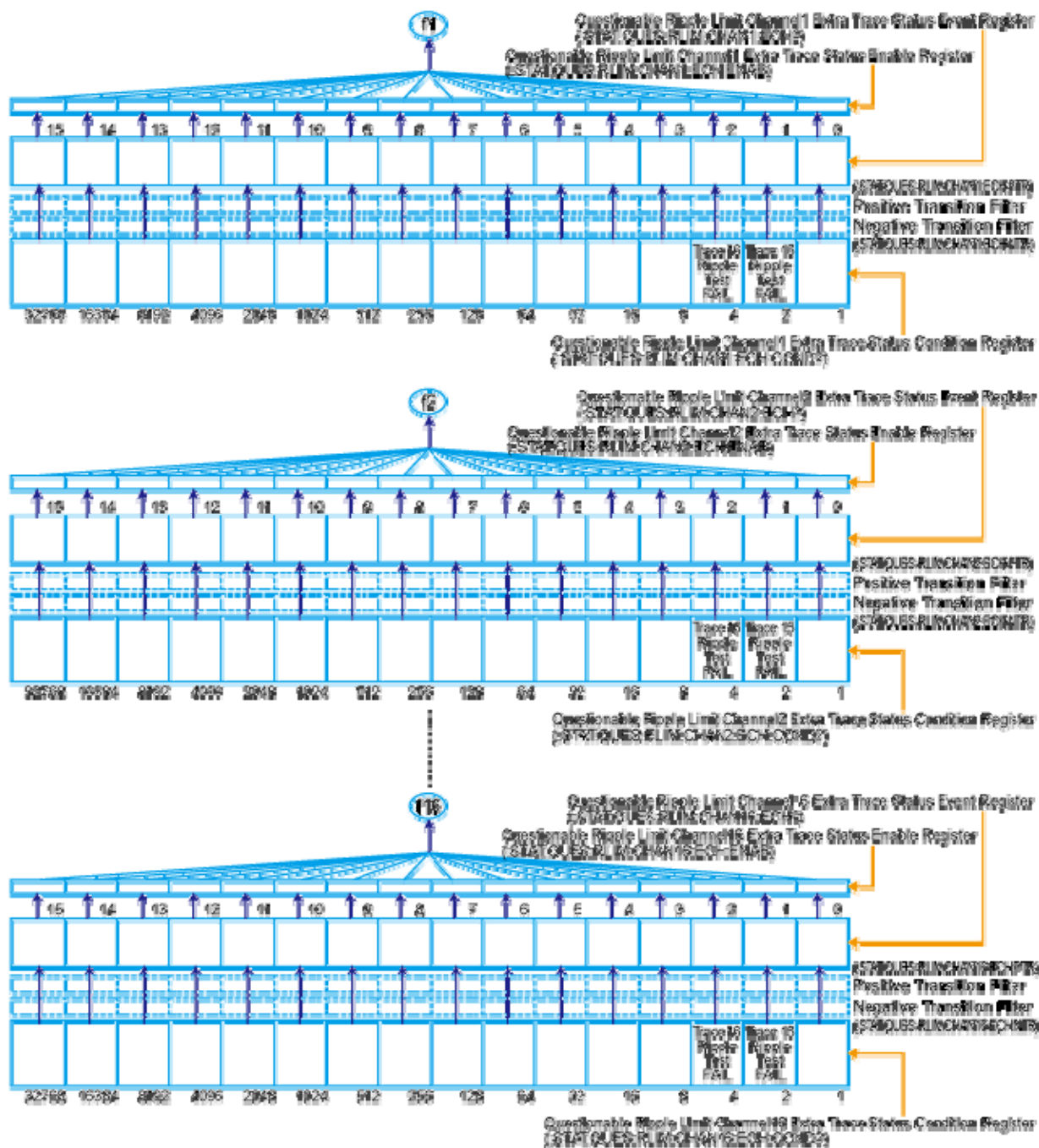
Status Bit Definitions of the Questionable Ripple Limit Channel {1-16} Status Condition Register

Bit Position	Name	Description
0	Trace 15, 16 Ripple Test summary (questionable ripple limit channel {1-16} extra status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel {1-16} extra status event register is set to "1."
1	Trace 1 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 1.
2	Trace 2 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 2.
3	Trace 3 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 3.
4	Trace 4 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 4.
5	Trace 5 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 5.
6	Trace 6 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 6.
7	Trace 7 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 7.
8	Trace 8 Ripple Test Fail	Set to "0" when a measurement cycle begins;

		set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 8.
9	Trace 9 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 9.
10	Trace 10 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 10.
11	Trace 11 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 11.
12	Trace 12 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 12.
13	Trace 13 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 13.
14	Trace 14 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 14.
15	Not used	Always 0

Issuing the ***CLS** command will clear all the bits in the questionable ripple limit channel {1-16} status event register.

Status Register for Ripple Limit (Trace) (2 of 2)



e5071c225

Status Bit Definitions of the Questionable Ripple Limit Channel { 1-16 } Extra Status Condition Register

Bit Position	Name	Description
--------------	------	-------------

0	Not used	Always 0
1	Trace 15 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 15.
2	Trace 16 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 16.
3-15	Not used	Always 0

Issuing the ***CLS** command will clear all the bits in the questionable ripple limit channel {1-16} extra status event register.

Working with Automatic Test System

Working with Automatic Test System

- Preventing Erroneous Key Operation on the Front Panel (key lock feature)
- Improving Command Processing Speed
- Detecting Occurrence of an Error

Preventing Erroneous Key Operation on the Front Panel (key lock feature)

When no operation is required from the front panel controls, the mouse, or the keyboard, disabling these input devices can prevent any erroneous operation that might be caused by accidentally touching the devices.

To turn on or off Key Locking, use the following commands:

Command	Description
<code>:SYST:KLOC:KBD</code>	Locking the front panel controls and the keyboard
<code>:SYST:KLOC:MOUS</code>	Locking the mouse and the touch screen

Improving Command Processing Speed

SCPI commands should be processed quickly to improve throughput when such commands are frequently executed (for example, reading out traces for each measurement).

With the E5071C, the processing time for SCPI commands can be improved by decreasing the refresh rate of the LCD display.

Measurement results (trace) do not need to be updated

When the measurement trace does not need to be updated, turn off the updating of the LCD display. This improves the processing speed of SCPI commands and eliminates the time used for updating the screen.

To turn off the updating of the LCD display, use the following command:

:DISP:ENAB

Measurement results (trace) need to be updated

When the measurement trace needs to be updated, the processing speed of SCPI commands can still be improved by controlling the update timing of the LCD display:

1. Execute all SCPI commands that are required before measurement, including commands setting conditions.
2. Turn OFF the update of the LCD display.
3. Perform the measurement.
4. Execute the commands for reading out measurement result or analyzing the result. Note that reading out the result in binary format will accelerate data transfer.
5. Execute the following command to update the LCD display once
:DISP:UPD
6. Return to Step 3.

Sample program

See Control LCD Update Timing.

Detecting Occurrence of an Error

- [Using Status Reporting System](#)
- [Using Error Queue](#)
- [Sample Program](#)

Other topics about Working with Automatic Test System

Using Status Reporting System

The status of the E5071C can be detected through the status registers. This section describes how to detect the end of measurement by using the status registers.

The occurrence of an error will be present in the standard event status register. An SRQ (service request) is useful when you create a program that uses the information reported by this register to detect the occurrence of an error.

To detect the end of sweep via an SRQ, use one of the following commands:

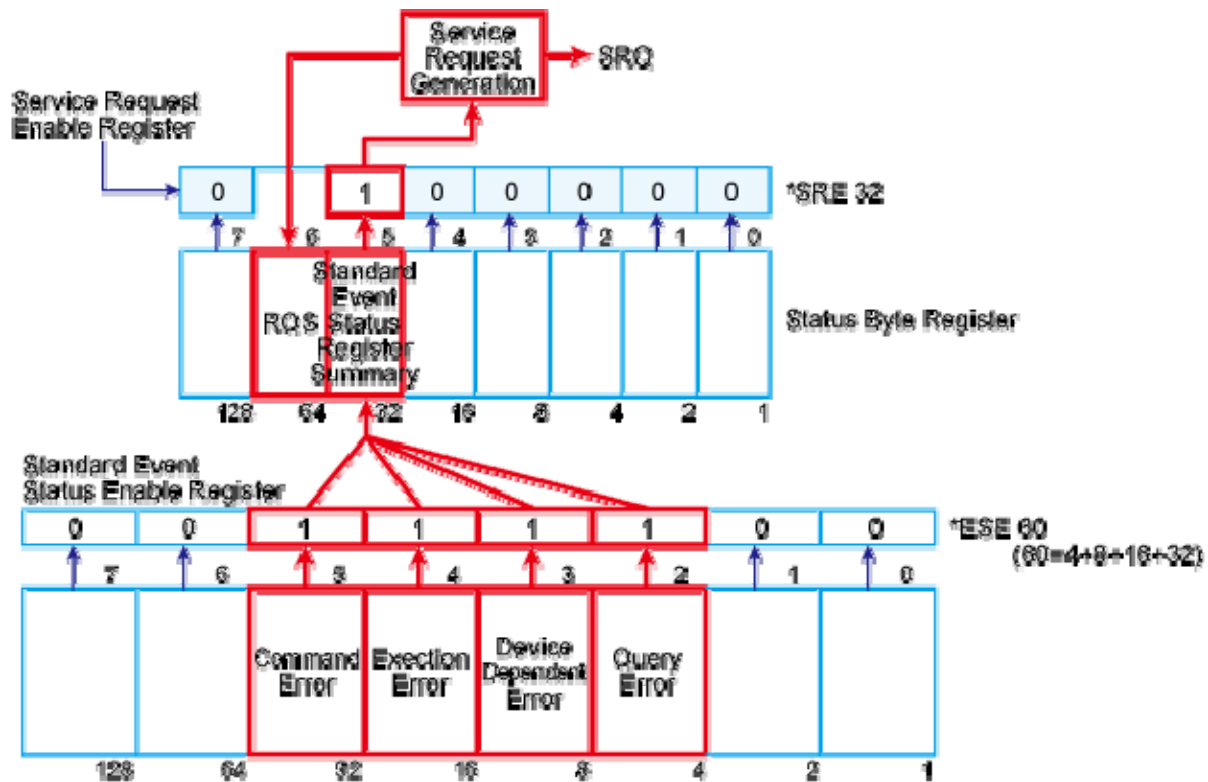
*SRE

*ESE

Follow these steps:

1. Set the E5071C so that it generates an SRQ when any of the error occurrence bits is set to 1 in the standard event status register.
2. When an SRQ is generated, the program interrupts the measurement cycle.

SRQ generation sequence (when an error occurs)



65071c257

Using Error Queue

An error queue holds the number for the error and the error message. Reading the error queue allows the user to verify the error that has occurred. To retrieve the content of an error queue, use the following command:

:SYST:ERR?

The error queue can be used in the following ways:

1. It is used as a branch for error handling. When an error queue is retrieved, it returns 0 as the error number and "No error" as the error message if no error is detected. This can be used for detecting of an error and for branching the flow of a program. This is also useful when you wish to handle a specific error(s). Note that this method will not allow the user to perform any processing during the occurrence of an error.
2. When an error is detected using SRQ, the error queue is used to examine the error. Refer to the sample program in this section.

Sample Program

See Error Detection (SRQ).

Sample Programs

Sample Programs

This section shows sample programs with the SCPI commands which can be executed from the external controller. See Application Programs under VBA Programming about the sample programs for built-in VBA.

- Analyzer Setup
- Calibration
- ECal
- Power Calibration
- Reading/Writing Error Coefficient
- Waiting for Trigger (OPC?)
- Waiting for Trigger (SRQ)
- Error Detection (SRQ)
- Reading Data in Ascii Format
- Reading Data in Binary Format
- Writing Data in Ascii Format
- Writing Data in Binary Format
- Peak Search
- Bandwidth Search
- Limit Test
- Saving Files
- Transferring Files
- Fixture Simulator
- Time Domain
- Control Using SICL-LAN Server
- Controlling Using Telnet Server
- Handler Interface
- Controlling E5091A

These sample program files can be downloaded from http://www.agilent.com/find/ena_support.

Analyzer Setup

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The program listed in this section is a sample program that demonstrates how to configure measurement conditions.

The sample program puts the instrument into the preset state, configures it as shown in table below, and saves the settings to a file named **sample.sta**.

See Setting up Analyzer for this programming.

Target settings

Item		Setting
Window Layout		Channel 1 in the upper window (2/3 of the screen height) and channel 2 in the lower window (1/3 of the screen height)
Channel 1	Sweep type	Segment
	Sweep range	See Segment table.
	Number of measurement points	
	IF bandwidth	
	Power	
	Number of traces	4

	Graph Layout		Four graphs at upper left, upper right, lower left, and lower right.
	Trace 1	Measurement parameter	S11
		Data format	Smith chart (Lin)
		Full-scale value	2
	Trace 2	Measurement parameter	S21
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 3	Measurement parameter	S12
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 4	Measurement parameter	S22
		Data format	Smith chart (Lin)
		Full-scale value	2
Channel 2	Sweep type		Linear
	Sweep range	Center value	1.9 GHz
		Span value	500 MHz

	Number of measurement points		101
	IF bandwidth		70 kHz
	Power		0 dBm
	Number of traces		4
	Graph Layout		Two graphs at left and right
	Trace 1	Measurement parameter	S21
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 2	Measurement parameter	S22
		Data format	Smith chart (Lin)
		Full-scale value	2

Segment table for channel 1

Segment Number	Start value	Stop value	Number of measurement points	IF bandwidth	Power
1	1.7 GHz	1.9 GHz	21	50 kHz	0 dBm
2	1.9 GHz	2 GHz	101	10 kHz	-10 dBm
3	2 GHz	2.2 GHz	21	50 kHz	0 dBm

[Sample Program in Excel VBA](#)

```
Sub Setup()
```

```
,
```

```
Dim defrm As Long
```

```
Dim vi As Long
```

Const TimeOutTime = 20000

```
Dim Allocate1 As String, Allocate2 As String, File As String
Dim Para1(4) As String, Para2(2) As String
Dim Fmt1(4) As String, Fmt2(2) As String
Dim Star1(3) As String, Stop1(3) As String
Dim IfBw1(3) As Double, IfBw2 As Double
Dim Power1(3) As Double, Power2 As Double
Dim Cent2 As Double, Span2 As Double
Dim RefLev1(4) As Double, RefLev2(2) As Double, Scale1(4) As Double, Scale2(2) As Double
Dim Segm As Integer, Nop1(3) As Integer, Nop2 As Integer
Dim NumOfTr1 As Integer, NumOfTr2 As Integer
Dim RefPos1(4) As Integer, RefPos2(2) As Integer
Dim SendData As String
```

```
Segm = 3 ' Number of Segment Ch.1 : 3
Star1(1) = "1.7E9" ' Start Frequency Ch.1 Segm.1: 1.7 GHz
Star1(2) = "1.9E9" ' Segm.2: 1.9 GHz
Star1(3) = "2E9" ' Segm.3: 2 GHz
Stop1(1) = "1.9E9" ' Stop Frequency Ch.1 Segm.1: 1.9 GHz
Stop1(2) = "2E9" ' Segm.2: 2 GHz
Stop1(3) = "2.2E9" ' Segm.3: 2.2 GHz
Cent2 = 1900000000# ' Center Frequency Ch.2 : 1.9 GHz
Span2 = 500000000# ' Span Ch.2 : 500 MHz
Nop1(1) = 21 ' Number of points Segm.1: 21
Nop1(2) = 101 ' Segm.2: 101
Nop1(3) = 21 ' Segm.3: 21
Nop2 = 101 ' Ch.2 : 101
IfBw1(1) = 50000# ' IFBW Ch.1 Segm.1: 50 kHz
IfBw1(2) = 10000# ' Segm.2: 10 kHz
IfBw1(3) = 50000# ' Segm.3: 50 kHz
IfBw2 = 70000# ' Ch.2 : 70 kHz
Power1(1) = 0 ' Power Ch.1 Segm.1: 0 dBm
Power1(2) = -10 ' Segm.2: -10 dBm
Power1(3) = 0 ' Segm.3: 0 dBm
Power2 = 0 ' Ch.2 : 0 dBm
NumOfTr1 = 4 ' Number of Trace Ch.1 : 4
```

E5071C

```

NumOfTr2 = 2      ' Ch.2 : 2
Allocate1 = "D12_34" ' Allocate Traces Ch.1 : D12_34
Allocate2 = "D12"   ' Ch.2 : D12
Para1(1) = "S11"    ' Measurement Ch.1 Trace1: S11
Para1(2) = "S21"    ' Parameter Trace2: S21
Para1(3) = "S12"    ' Trace3: S12
Para1(4) = "S22"    ' Trace4: S22
Para2(1) = "S31"    ' Ch.2 Trace1: S31
Para2(2) = "S33"    ' Trace2: S33
Fmt1(1) = "SLIN"    ' Data Format Ch.1 Trace1: Smith(Lin/Phase)
Fmt1(2) = "MLOG"    ' Trace2: Log Mag
Fmt1(3) = "MLOG"    ' Trace3: Log Mag
Fmt1(4) = "SLIN"    ' Trace4: Smith(Lin/Phase)
Fmt2(1) = "MLOG"    ' Ch.2 Trace1: Log Mag
Fmt2(2) = "SLIN"    ' Trace2: Smith(Lin/Phase)
RefPos1(1) = 9      ' Reference Ch.1 Trace2: 9
RefPos1(2) = 9      ' Position Trace3: 9
RefPos2(1) = 9      ' Ch.2 Trace1: 9
RefLev1(1) = 0      ' Reference Level Ch.1 Trace2: 0 dBm
RefLev1(2) = 0      ' Trace3: 0 dBm
RefLev2(1) = 0      ' Ch.2 Trace1: 0 dBm
Scale1(1) = 2       ' Scale Ch.1 Trace1: 2
Scale1(2) = 10      ' Trace2: 10 dBm
Scale1(3) = 10      ' Trace3: 10 dBm
Scale1(4) = 2       ' Trace4: 2
Scale2(1) = 10      ' Ch.2 Trace1: 10 dBm
Scale2(2) = 2       ' Trace2: 2
StaFileName = "sample.sta" ' Save File Name : sample.sta
'
' Assigns a GPIB address to the I/O pass.
Call viOpenDefaultRM(defrm)
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) ' Set time out
'

Call viVPrintf(vi, ":SYST:PRES" + vbLf, 0)
Call viVPrintf(vi, ":DISP:SPL D1_1_2" + vbLf, 0) 'Allocate Channel
Call viVPrintf(vi, ":INIT1:CONT ON" + vbLf, 0) 'Turn on Continuous Activation mode for channel 1

```



```

Call viVPrintf(vi, ":INIT2:CONT ON" + vbLf, 0) 'Turn on Continuous Activation mode for channel 2
' =====
' Setup Channel 1
' =====
Call viVPrintf(vi, ":SENS1:SWE:TYPE SEGM" + vbLf, 0) 'Sets channel 1 sweep type to segment
'
' Create the data string for Segment Table
SendData = "5,0,1,1,0,0," & Str(Segm)
For i = 1 To Segm
    SendData = SendData + "," & Star1(i) & "," + Stop1(i) & "," & CStr(Nop1(i)) & "," & CStr(IfBw1(i)) &
"," & CStr(Power1(i))
Next i
Call viVPrintf(vi, ":SENS1:SEGM:DATA " + SendData + vbLf, 0)
'

Call viVPrintf(vi, ":CALC1:PAR:COUN " & CStr(NumOfTr1) & vbLf, 0) 'Set number of traces
Call viVPrintf(vi, ":DISP:WIND1:SPL " & Allocate1 & vbLf, 0) 'Set graph layout

For i = 1 To NumOfTr1
    Call viVPrintf(vi, ":CALC1:PAR" & CStr(i) & ":DEF " & Para1(i) & vbLf, 0) 'Set measurement
parameter
    Call viVPrintf(vi, ":CALC1:PAR" & CStr(i) & ":SEL" & vbLf, 0) ' Make trace active
    Call viVPrintf(vi, ":CALC1:FORM " & Fmt1(i) & vbLf, 0) ' Set data format
    '

    Select Case Fmt1(i)
        Case "SLIN", "SLOG", "SCOM", "SMIT", "SADM", "PLIN", "PLOG", "POL"
            ' If data format is neither Smith chart nor polar, sets reference division line number and scale per
division
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:PDIV " + CStr(Scale1(i)) + vbLf, 0)
        Case Else
            ' If data format is Smith chart or polar, set full-scale value
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:RPOS " & CStr(RefPos1(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:RLEV " & CStr(RefLev1(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:PDIV " & CStr(Scale1(i)) & vbLf, 0)
        End Select
    Next i
' =====
' Setup Channel 2
' =====

```

E5071C

```
Call viVPrintf(vi, ":SENS2:SWE:TYPE LIN " + vbLf, 0) ' Set sweep type to linear
Call viVPrintf(vi, ":SENS2:FREQ:CENT " + CStr(Cent2) + vbLf, 0) ' Set center frequency
Call viVPrintf(vi, ":SENS2:FREQ:SPAN " + CStr(Span2) + vbLf, 0) ' Set span frequency
Call viVPrintf(vi, ":SENS2:SWE:POIN " + CStr(Nop2) + vbLf, 0) ' Set number of points
Call viVPrintf(vi, ":SENS2:BAND " + CStr(IfBw2) + vbLf, 0) ' Set IFBW
Call viVPrintf(vi, ":SOUR2:POW " + CStr(Power2) + vbLf, 0) ' Set power level
Call viVPrintf(vi, ":CALC2:PAR:COUN " & CStr(NumOfTr2) & vbLf, 0) ' Set number of traces
Call viVPrintf(vi, ":DISP:WIND2:SPL " & Allocate2 & vbLf, 0) 'Set graph layout
```

```
For i = 1 To NumOfTr2
```

```
    Call viVPrintf(vi, ":CALC2:PAR" & CStr(i) & ":DEF " & Para2(i) & vbLf, 0) 'Set measurement parameter
```

```
    Call viVPrintf(vi, ":CALC2:PAR" & CStr(i) & ":SEL" & vbLf, 0) ' Make trace active
```

```
    Call viVPrintf(vi, ":CALC2:FORM " & Fmt2(i) & vbLf, 0) ' Set data format
```

```
    '
```

```
    Select Case Fmt2(i)
```

```
        Case "SLIN", "SLOG", "SCOM", "SMIT", "SADM", "PLIN", "PLOG", "POL"
```

```
        ' If data format is neither Smith chart nor polar, sets reference division line number and scale per division
```

```
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:PDIV " + CStr(Scale2(i)) + vbLf, 0)
```

```
        Case Else
```

```
        ' If data format is Smith chart or polar, set full-scale value
```

```
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:RPOS " & CStr(RefPos2(i)) & vbLf, 0)
```

```
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:RLEV " & CStr(RefLev2(i)) & vbLf, 0)
```

```
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:PDIV " & CStr(Scale2(i)) & vbLf, 0)
```

```
        End Select
```

```
    Next i
```

```
    '
```

```
    Call viVPrintf(vi, ":MMEM:STOR """" & StaFileName & """" & vbLf, 0) ' Save ENA settings to file
```

```
    ' Close IO
```

```
    Call viClose(vi)
```

```
    Call viClose(defrm)
```

```
End Sub
```

Sample Program in HT Basic (setup.htb)

```
10 DIM Allocate1$(9),Allocate2$(9),File$(20)
```

```
20 DIM Para1$(1:4)(9),Para2$(1:2)(9),Fmt1$(1:4)(9),Fmt2$(1:2)(9)
```

```
30 REAL Star1(1:3),Stop1(1:3),Pow1(1:3),Cent2,Span2,Pow2
```

```

40 REAL Ref_rev1(1:4),Ref_rev2(1:2),Scale1(1:4),Scale2(1:2)
50 INTEGER Segm,Nop1(1:3),Nop2,Num_of_tr1,Num_of_tr2
60 INTEGER Ref_pos1(1:4),Ref_pos2(1:2),I
70 ASSIGN @Agte507x TO 717
80 !
90 Segm=3 ! Number of Segment Ch.1 : 3
100 Star1(1)=1.7E+9 ! Start Frequency Ch.1 Segm.1: 1.7 GHz
110 Star1(2)=1.9E+9 ! Segm.2: 1.9 GHz
120 Star1(3)=2.E+9 ! Segm.3: 2 GHz
130 Stop1(1)=1.9E+9 ! Stop Frequency Ch.1 Segm.1: 1.9 GHz
140 Stop1(2)=2.E+9 ! Segm.2: 2 GHz
150 Stop1(3)=2.2E+9 ! Segm.3: 2.2 GHz
160 Cent2=1.9E+9 ! Center Frequency Ch.2 : 1.9 GHz
170 Span2=5.00E+8 ! Span Ch.2 : 500 MHz
180 Nop1(1)=21 ! Number Ch.1 Segm.1: 21
190 Nop1(2)=101 ! of Points Segm.2: 101
200 Nop1(3)=21 ! Segm.3: 21
210 Nop2=101 ! Ch.2 : 101
220 If_bw1(1)=5.0E+4 ! IF Bandwidth Ch.1 Segm.1: 50 kHz
230 If_bw1(2)=1.0E+4 ! Segm.2: 10 kHz
240 If_bw1(3)=5.0E+4 ! Segm.3: 50 kHz
250 If_bw2=7.0E+4 ! Ch.2 : 70 kHz
260 Pow1(1)=0 ! Power Ch.1 Segm.1: 0 dBm
270 Pow1(2)=-10 ! Segm.2: -10 dBm
280 Pow1(3)=0 ! Segm.3: 0 dBm
290 Pow2=0 ! Ch.2 : 0 dBm
300 Num_of_tr1=4 ! Number Ch.1 : 4
310 Num_of_tr2=2 ! of Traces Ch.2 : 2
320 Allocate1$="D12_34" ! Allocate Traces Ch.1 : D12_34
330 Allocate2$="D12" ! Ch.2 : D12
340 Para1$(1)="S11" ! Measurement Ch.1 Trace1: S11
350 Para1$(2)="S21" ! Parameter Trace2: S21
360 Para1$(3)="S12" ! Trace3: S12
370 Para1$(4)="S22" ! Trace4: S22
380 Para2$(1)="S31" ! Ch.2 Trace1: S31
390 Para2$(2)="S33" ! Trace2: S33
400 Fmt1$(1)="SLIN" ! Data Format Ch.1 Trace1: Smith(Lin/Phase)

```

E5071C

```
410 Fmt1$(2)="MLOG" ! Trace2: Log Mag
420 Fmt1$(3)="MLOG" ! Trace3: Log Mag
430 Fmt1$(4)="SLIN" ! Trace4: Smith(Lin/Phase)
440 Fmt2$(1)="MLOG" ! Ch.2 Trace1: Log Mag
450 Fmt2$(2)="SLIN" ! Trace2: Smith(Lin/Phase)
460 Ref_pos1(2)=9 ! Reference Ch.1 Trace2: 9
470 Ref_pos1(3)=9 ! Position Trace3: 9
480 Ref_pos2(1)=9 ! Ch.2 Trace1: 9
490 Ref_lev1(2)=0 ! Reference Level Ch.1 Trace2: 0 dBm
500 Ref_lev1(3)=0 ! Trace3: 0 dBm
510 Ref_lev2(1)=0 ! Ch.2 Trace1: 0 dBm
520 Scale1(1)=2 ! Scale Ch.1 Trace1: 2
530 Scale1(2)=10 ! Trace2: 10 dBm
540 Scale1(3)=10 ! Trace3: 10 dBm
550 Scale1(4)=2 ! Trace4: 2
560 Scale2(1)=10 ! Ch.2 Trace1: 10 dBm
570 Scale2(2)=2 ! Trace2: 2
580 File$="sample.sta" ! Save File Name : sample.sta
590 !
600 OUTPUT @Agte507x;".SYST:PRES"
610 !
620 OUTPUT @Agte507x;".DISP:SPL D1_1_2"
630 OUTPUT @Agte507x;".INIT1:CONT ON"
640 OUTPUT @Agte507x;".INIT2:CONT ON"
650 !
660 ! Channel 1
670 !
680 OUTPUT @Agte507x;".SENS1:SWE:TYPE SEGM"
690 OUTPUT @Agte507x;".SENS1:SEGM:DATA 5,0,1,1,0,0;";Segm;";";
700 FOR I=1 TO Segm-1
710 OUTPUT @Agte507x;Star1(I);";";Stop1(I);";";Nop1(I);";";If_bw1 (I);";";Pow1(I);";";
720 NEXT I
730 OUTPUT @Agte507x;Star1(Segm);";";Stop1(Segm);";";Nop1(Segm);";";
;If_bw1(Segm);";";Pow(Segm)
740 !
750 OUTPUT @Agte507x;".CALC1:PAR:COUN ";Num_of_tr1
760 OUTPUT @Agte507x;".DISP:WIND1:SPL "&Allocate1$
```

```

770 FOR I=1 TO Num_of_tr1
780 OUTPUT @Agte507x;":CALC1:PAR"&VAL$(I)&":DEF "&Para1$(I)
790 OUTPUT @Agte507x;":CALC1:PAR"&VAL$(I)&":SEL"
800 OUTPUT @Agte507x;":CALC1:FORM "&Fmt1$(I)
810 SELECT Fmt1$(I)
820 CASE "SLIN","SLOG","SCOM","SMIT","SADM","PLIN","PLOG","POL"
830 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:PDIV "; Scale1(I)
840 CASE ELSE
850 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:RPOS "; Ref_pos1(I)
860 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:RLEV "; Ref_rev1(I)
870 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:PDIV "; Scale1(I)
880 END SELECT
890 NEXT I
900 !
910 ! Channel 2
920 !
930 OUTPUT @Agte507x;":SENS2:SWE:TYPE LIN"
940 OUTPUT @Agte507x;":SENS2:FREQ:CENT ";Cent2
950 OUTPUT @Agte507x;":SENS2:FREQ:SPAN ";Span2
960 OUTPUT @Agte507x;":SENS2:SWE:POIN ";Nop2
970 OUTPUT @Agte507x;":SENS2:BAND ";lf_bw2
980 OUTPUT @Agte507x;":SOUR2:POW ";Pow2
990 !
1000 OUTPUT @Agte507x;":CALC2:PAR:COUN ";Num_of_tr2
1010 OUTPUT @Agte507x;":DISP:WIND2:SPL "&Allocate2$
1020 FOR I=1 TO Num_of_tr2
1030 OUTPUT @Agte507x;":CALC2:PAR"&VAL$(I)&":DEF "&Para2$(I)
1040 OUTPUT @Agte507x;":CALC2:PAR"&VAL$(I)&":SEL"
1050 OUTPUT @Agte507x;":CALC2:FORM "&Fmt2$(I)
1060 SELECT Fmt2$(I)
1070 CASE "SLIN","SLOG","SCOM","SMIT","SADM","PLIN","PLOG","POL"
1080 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:PDIV "; Scale2(I)
1090 CASE ELSE
1100 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:RPOS "; Ref_pos2(I)
1110 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:RLEV "; Ref_rev2(I)
1120 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:PDIV "; Scale2(I)
1130 END SELECT

```

E5071C

1140 NEXT I

1150 !

1160 OUTPUT @Agte507x;":MMEM:STOR ""&File\$&""

1170 END

Calibration

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program performs calibration with the specified calibration type.

See Calibration for this programming.

Sample Program in Excel VBA

Sub Cal_Click()

Dim defrm As Long 'Session to Default Resource Manager

Dim vi As Long 'Session to instrument

Dim Ch As String

Dim CalKit As Integer

Dim Port(4) As String

Const TimeOutTime = 40000 'timeout time.

Const Cal85032F = 4 'cal kit number.

Ch = Cells(5, 5) 'Select channel

Port(1) = Cells(3, 6) 'Sets the select port 1.

Port(2) = Cells(3, 7) 'Sets the select port 2.

Port(3) = Cells(3, 8) 'Sets the select port 3.

Port(4) = Cells(3, 9) 'Sets the select port 4.

CalKit = Cal85032F 'Sets cal kit (85032F)

Call viOpenDefaultRM(defrm) 'Initializes the VISA system.

Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi) 'Opens the session to the specified instrument.

Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) 'The state of an attribute for the specified session.

Call viVPrintf(vi, "**RST" & vbLf, 0) 'Presets the setting state of the ENA.

Call viVPrintf(vi, "**CLS" & vbLf, 0) 'Clears the all status register.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:CKIT " & CalKit & vbLf, 0) 'Select the calibration kit

E5071C

Select Case Cells(3, 5)

Case "Response (Open)" 'Perform response calibration (OPEN).

Call Cal_Resp(vi, Ch, "OPEN", Port(1))

Case "Response (Short)" 'Perform response calibration (SHORT).

Call Cal_Resp(vi, Ch, "Short", Port(1))

Case "Response (Thru)" 'Perform response calibration (Thru).

Call Cal_RespThru(vi, Ch, "Thru", Port(1), Port(2))

Case "Full 1 Port" 'Perform 1-port calibration.

Call Cal_Slot(vi, Ch, 1, Port)

Case "Full 2 Port" 'Perform full 2-port calibration.

Call Cal_Slot(vi, Ch, 2, Port)

Case "Full 3 Port" 'Perform full 3-port calibration.

Call Cal_Slot(vi, Ch, 3, Port)

Case "Full 4 Port" 'Perform full 4-port calibration.

Call Cal_Slot(vi, Ch, 4, Port)

End Select

Call viClose(vi) 'Closes the resource manager session.

Call viClose(defrm) 'Breaks the communication and terminates the VISA system.

End

'End

End Sub

Sub Cal_Resp(vi As Long, Ch As String, CalType As String, Port As String)

Dim Dummy As Variant 'Variant to receive the result

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:" & CalType & " " & Port & vbLf, 0) 'Sets the calibration type.

MsgBox ("Set " & CalType & " to Port " & Port & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:" & CalType & " " & Port & vbLf, 0) 'Measurement the calibration data.

Call viVQueryf(vi, "**OPC?" & vbLf, "%t", Dummy) 'Reads the *OPC? result.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0) 'Calculating the calibration coefficients.

Call ErrorCheck(vi) 'Checking the error.

End Sub

Sub Cal_RespThru(vi As Long, Ch As String, CalType As String, Port1 As String, Port2 As String)

Dim Dummy As Variant 'Variant to receive the result.

If Port1 <> Port2 Then

Call viVPrintf(vi, "SENS" & Ch & "CORR:COLL:METH:" & CalType & " " & Port1 & "," & Port2 & vbCrLf, 0) 'Sets the calibration type

MsgBox ("Set " & CalType & " to Port " & Port1 & "&" & Port2 & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, "SENS" & Ch & "CORR:COLL:" & CalType & " " & Port1 & "," & Port2 & vbCrLf, 0) 'Measurement the calibration data.

Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.

Call viVPrintf(vi, "SENS" & Ch & "CORR:COLL:SAVE" & vbCrLf, 0) 'Calculating the calibration coefficients.

Call ErrorCheck(vi) 'Checking the error.

Else

MsgBox ("Thru calibration select port error!") 'Displaying the error message when selected same ports.

Exit Sub

End If

End Sub

Sub Cal_Slot(vi As Long, Ch As String, NumPort As String, Port() As String)

Dim Dummy

Dim i As Integer, j As Integer

Select Case NumPort

Case 1

Call viVPrintf(vi, "SENS" & Ch & "CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & vbCrLf, 0) 'Set the 1-port calibration type.

Case 2

Call viVPrintf(vi, "SENS" & Ch & "CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & vbCrLf, 0) 'Set the full 2-port calibration type.

Case 3

E5071C

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & "," & Port(3) & vbCrLf, 0) 'Set the full 3-port calibration type.

Case 4

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT4 1,2,3,4" & vbCrLf, 0) 'Set the full 4-port calibration type.

End Select

'Reflection

For i = 1 To NumPort

MsgBox ("Set Open to Port " & Port(i) & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:OPEN " & Port(i) & vbCrLf, 0) 'Measurement the OPEN calibration.

Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.

MsgBox ("Set Short to Port " & Port(i) & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SHORT " & Port(i) & vbCrLf, 0) 'Measurement the SHORT calibration.

Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.

MsgBox ("Set Load to Port " & Port(i) & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:LOAD " & Port(i) & vbCrLf, 0) 'Measurement the LOAD calibration.

Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.

Next i

'Transmission

For i = 1 To NumPort - 1

For j = i + 1 To NumPort

MsgBox ("Set Thru to Port " & Port(i) & "&" & Port(j) & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(i) & "," & Port(j) & vbCrLf, 0) 'Measurement the THRU calibration.

Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC result.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(j) & "," & Port(i) & vbCrLf, 0) 'Measurement the THRU calibration.

Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC result.

Next j

Next i

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbCrLf, 0) 'Calculating the calibration coefficients.

Call ErrorCheck(vi) 'Checking the error.

```

End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err) 'Reads error message.
    ErrNo = Split(err, ",") 'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly) 'Display the message box.
    End If
End Sub

```

Sample Program in HT Basic (cal.htb)

```

10 DIM File$(20),Ch$(9),Inp_char$(9)
20 INTEGER Cal_kit,Cal_type,Port(1:4)
30 !
40 ASSIGN @Agte507x TO 717
50 File$="Ex_4_1.sta"
60 Ch$="1"
70 !
80 Select_cal_kit(@Agte507x,Ch$)
90 !
100 CLEAR SCREEN
110 ON ERROR GOTO Type_select
120 Type_select: !
130 PRINT "## Calibration Type Selection ##"
140 PRINT " 1: Response (Open)"
150 PRINT " 2: Response (Short)"
160 PRINT " 3: Response (Thru)"
170 PRINT " 4: Full 1 Port"
180 PRINT " 5: Full 2 Port"
190 PRINT " 6: Full 3 Port"
200 PRINT " 7: Full 4 Port"
210 PRINT ""
220 PRINT "Input 1 to 7"
230 INPUT "Input number? (1 to 7)",Inp_char$
240 Cal_type=IVAL(Inp_char$,10)

```

E5071C

```
250 IF Cal_type<1 OR Cal_type>7 THEN Type_select
260 OFF ERROR
270 !
280 SELECT Cal_type
290 CASE 1
300 Select_port(1,Port(*))
310 Cal_resp(@Agte507x,Ch$,"OPEN",Port(1))
320 CASE 2
330 Select_port(1,Port(*))
340 Cal_resp(@Agte507x,Ch$,"SHOR",Port(1))
350 CASE 3
360 Select_port(2,Port(*))
370 Cal_resp_thru(@Agte507x,Ch$,Port(1),Port(2))
380 CASE 4
390 Select_port(1,Port(*))
400 Cal_solt(@Agte507x,Ch$,1,Port(*))
410 CASE 5
420 Select_port(2,Port(*))
430 Cal_solt(@Agte507x,Ch$,2,Port(*))
440 CASE 6
450 Select_port(3,Port(*))
460 Cal_solt(@Agte507x,Ch$,3,Port(*))
470 CASE 7
480 Select_port(4,Port(*))
490 Cal_solt(@Agte507x,Ch$,4,Port(*))
500 END SELECT
510 !
520 OUTPUT @Agte507x;":MMEM:STOR:STYP CST"
530 OUTPUT @Agte507x;":MMEM:STOR ""&File$&""
540 END
550 !======
560 ! Calibration Kit Selection Function
570 !======
580 SUB Select_cal_kit(@Agte507x,Ch$)
590 DIM Cal_kit_lbl$(1:10)[20],Inp_char$(9)
600 INTEGER Cal_kit,I
610 CLEAR SCREEN
```

```

620 !
630 FOR I=1 TO 10
640 OUTPUT @Agte507x;".SENS1:CORR:COLL:CKIT ";I
650 OUTPUT @Agte507x;".SENS1:CORR:COLL:CKIT:LAB?"
660 ENTER @Agte507x;Cal_kit_lbl$(I)
670 NEXT I
680 ON ERROR GOTO Kit_select
690 Kit_select: !
700 PRINT "## Calibration Kit Selection ##"
710 FOR I=1 TO 10
720 PRINT USING "X,2D,A,X,20A";I,".",Cal_kit_lbl$(I)
730 NEXT I
740 PRINT ""
750 PRINT "Input 1 to 10"
760 INPUT "Input number? (1 to 10)",Inp_char$
770 Cal_kit=IVAL(Inp_char$,10)
780 IF Cal_kit<1 OR Cal_kit>10 THEN Kit_select
790 OFF ERROR
800 !
810 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:CKIT ";Cal_kit
820 SUBEND
830 !=====
840 ! Port Selection Function
850 !=====
860 SUB Select_port(INTEGER Num_of_ports,INTEGER Port(*))
870 DIM Inp_char$(9)
880 !
890 CLEAR SCREEN
900 IF Num_of_ports=4 THEN
910 Port(1)=1
920 Port(2)=2
930 Port(3)=3
940 Port(4)=4
950 ELSE
960 PRINT "## Test Ports Selection ##"
970 ON ERROR GOTO Port_select
980 FOR I=1 TO Num_of_ports

```

E5071C

```
990 PRINT "Port("&VAL$(I)&").";
1000 Port_select=!
1010 INPUT "Number?",Inp_char$
1020 Port(I)=IVAL(Inp_char$,10)
1030 IF Port(I)<1 OR Port(I)>4 THEN Port_select
1040 FOR J=1 TO I-1
1050 IF Port(I)=Port(J) THEN Port_select
1060 NEXT J
1070 PRINT Port(I)
1080 NEXT I
1090 OFF ERROR
1100 END IF
1110 SUBEND
1120 !======
1130 ! Response (Open/Short) Calibration Function
1140 !======
1150 SUB Cal_resp(@Agte507x,Ch$,Type$,INTEGER Port)
1160 DIM Buff$[9]
1170 !
1180 PRINT "## Response ("&Type$&) Calibration ##"
1190 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:METH:"&Type$&" ";Port
1200 PRINT "Set "&Type$&" to Port "&VAL$(Port)&". Then push [Enter] key."
1210 INPUT "",Buff$
1220 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:"&Type$&" ";Port
1230 OUTPUT @Agte507x;"*OPC?"
1240 ENTER @Agte507x;Buff$
1250 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:SAVE"
1260 PRINT "Done"
1270 SUBEND
1280 !======
1290 ! Response (Thru) Calibration Function
1300 !======
1310 SUB Cal_resp_thru(@Agte507x,Ch$,INTEGER Port1,Port2)
1320 DIM Buff$[9]
1330 !
1340 PRINT "## Response (Thru) Calibration ##"
1350 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:METH:THRU ";Port1;","; Port2
```

```

1360 PRINT "Set THRU between Port "&VAL$(Port1)&" and Port "&VAL$(Port2)&". Then push [Enter]
key."
1370 INPUT "",Buff$
1380 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:THRU ";Port1;";Port2
1390 OUTPUT @Agte507x;"*OPC?"
1400 ENTER @Agte507x;Buff$
1410 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:SAVE"
1420 PRINT "Done"
1430 SUBEND
1440 !======
1450 ! Full n Port Calibration Function
1460 !======
1470 SUB Cal_solt(@Agte507x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
1480 DIM Buff$[9]
1490 INTEGER I,J
1500 !
1510 PRINT "## Full "&VAL$(Num_of_ports)&" Port Calibration ##"
1520 !
1530 ! Calibration Type Selection
1540 !
1550 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:METH:SOLT"&VAL$(Num_of_ports)&" ";
1560 FOR I=1 TO Num_of_ports-1
1570 OUTPUT @Agte507x;Port(I);";";
1580 NEXT I
1590 OUTPUT @Agte507x;Port(Num_of_ports)
1600 !
1610 ! Reflection Measurement
1620 !
1630 FOR I=1 TO Num_of_ports
1640 PRINT "Set OPEN to Port "&VAL$(Port(I))&". Then push [Enter] key."
1650 INPUT "",Buff$
1660 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:OPEN ";Port(I)
1670 OUTPUT @Agte507x;"*OPC?"
1680 ENTER @Agte507x;Buff$
1690 PRINT "Set SHORT to Port "&VAL$(Port(I))&". Then push [Enter] key."
1700 INPUT "",Buff$
1710 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:SHOR ";Port(I)

```

E5071C

```
1720 OUTPUT @Agte507x;"*OPC?"
1730 ENTER @Agte507x;Buff$
1740 PRINT "Set LOAD to Port "&VAL$(Port(I))&". Then push [Enter] key."
1750 INPUT "",Buff$
1760 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:LOAD ";Port(I)
1770 OUTPUT @Agte507x;"*OPC?"
1780 ENTER @Agte507x;Buff$
1790 NEXT I
1800 !
1810 ! Transmission Measurement
1820 !
1830 FOR I=1 TO Num_of_ports-1
1840 FOR J=I+1 TO Num_of_ports
1850 PRINT "Set THRU between Port "&VAL$(Port(I))&" and Port "& VAL$(Port(J))&". Then push [Enter]
key."
1860 INPUT "",Buff$
1870 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(I);"," ;Port(J)
1880 OUTPUT @Agte507x;"*OPC?"
1890 ENTER @Agte507x;Buff$
1900 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(J);"," ;Port(I)
1910 OUTPUT @Agte507x;"*OPC?"
1920 ENTER @Agte507x;Buff$
1930 NEXT J
1940 NEXT I
1950 !
1960 ! Done
1970 !
1980 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SAVE"
1990 PRINT "Done"
2000 SUBEND
```


ECal

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program performs 1-port or 2-port calibration using ECal.
See Calibration for this programming.

Sample Program in Excel VBA

```
Sub ECal_Click()
    Dim defrm As Long      'Session to Default Resource Manager
    Dim vi As Long         'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(4) As String
    Const TimeOutTime = 40000 'timeout time.

    Ch = Cells(5, 5)      'Select channel
    Port(1) = Cells(3, 6)  'Sets the select port 1.
    Port(2) = Cells(3, 7)  'Sets the select port 2.
    Port(3) = Cells(3, 8)  'Sets the select port 3.
    Port(4) = Cells(3, 9)  'Sets the select port 4.

    Call viOpenDefaultRM(defrm) 'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi) 'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) 'The state of an attribute for the
    specified session.

    Call viVPrintf(vi, "**RST" & vbLf, 0) 'Presets the setting state of the ENA.
    Call viVPrintf(vi, "**CLS" & vbLf, 0) 'Clears the all status register.

    Select Case Cells(3, 5)
        Case "1 Port"
            Call ECal(vi, Ch, 1, Port) 'Perform 1-port calibration.
        Case "2 Port"
            Call ECal(vi, Ch, 2, Port) 'Perform full 2-port calibration.
        Case "3 Port"
```

E5071C

```
    Call ECal(vi, Ch, 3, Port) 'Perform full 3-port calibration.
Case "4 Port"
    Call ECal(vi, Ch, 4, Port) 'Perform full 4-port calibration.
End Select
Call viClose(vi) 'Closes the resource manager session.
Call viClose(defrm) 'Breaks the communication and terminates the VISA system.
End
End Sub
Sub ECal(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy As Variant
    Dim i As Integer, j As Integer

    Select Case NumPort
        Case 1
            MsgBox ("Connect Port " & Port(1) & ". then click [OK] button") 'Display the message box.
            Call viVPrintf(vi, "SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & vbCrLf,
0) 'Execute the 1-port calibration.
        Case 2
            MsgBox ("Connect Port " & Port(1) & " and Port " & Port(2) & ". then click [OK] button") 'Display
the message box.
            Call viVPrintf(vi, "SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & vbCrLf, 0) 'Execute the full 2-port calibration.
        Case 3
            MsgBox ("Connect Port " & Port(1) & "," & Port(2) & " and Port " & Port(3) & ". then click [OK]
button") 'Display the message box.
            Call viVPrintf(vi, "SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & "," & Port(3) & vbCrLf, 0) 'Execute the full 3-port calibration.
        Case 4
            MsgBox ("Connect Port 1, 2, 3 and 4. then click [OK] button") 'Display the message box.
            Call viVPrintf(vi, "SENS" & Ch & ":CORR:COLL:ECAL:SOLT4 1,2,3,4" & vbCrLf, 0) 'Execute the full
4-port calibration.
    End Select

    Call ErrorCheck(vi) 'Checking the error.

End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response
```

```

Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err) 'Reads error message.
ErrNo = Split(err, ",") 'Gets the error code.
If Val(ErrNo(0)) <> 0 Then
    Response = MsgBox(CStr(ErrNo(1)), vbOKOnly) 'Display the message box.
End If
End Sub

```

Sample Program in HT Basic (ecal.htb)

```

10 DIM File$(20),Ch$(9),Inp_char$(9)
20 INTEGER Cal_kit,Cal_type,Port(1:4)
30 !
40 ASSIGN @Agte507x TO 717
50 File$="Ex_4_2.sta"
60 Ch$="1"
70 !
80 CLEAR SCREEN
90 ON ERROR GOTO Type_select
100 Type_select: !
110 PRINT "## Calibration Type Selection ##"
120 PRINT " 1: Full 1 Port"
130 PRINT " 2: Full 2 Port"
140 PRINT " 3: Full 3 Port"
150 PRINT " 4: Full 4 Port"
160 PRINT ""
170 PRINT "Input 1 to 4"
180 INPUT "Input number? (1 to 4)",Inp_char$
190 Cal_type=IVAL(Inp_char$,10)
200 IF Cal_type<1 OR Cal_type>4 THEN Type_select
210 OFF ERROR
220 !
230 Select_port(Cal_type,Port(*))
240 Ecal(@Agte507x,Ch$,Cal_type,Port(*))
250 !
260 OUTPUT @Agte507x;":MMEM:STOR:STYP CST"
270 OUTPUT @Agte507x;":MMEM:STOR """"&File$&""""
280 END
290 !======
300 ! Port Selection Function

```

E5071C

```
310 !=====
320 SUB Select_port(INTEGER Num_of_ports,INTEGER Port(*))
330 DIM Inp_char$(9)
340 !
350 CLEAR SCREEN
360 IF Num_of_ports=4 THEN
370 Port(1)=1
380 Port(2)=2
390 Port(3)=3
400 Port(4)=4
410 ELSE
420 PRINT "## Test Ports Selection ##"
430 ON ERROR GOTO Port_select
440 FOR I=1 TO Num_of_ports
450 PRINT "Port("&VAL$(I)&"):";
460 Port_select: !
470 INPUT "Number?",Inp_char$
480 Port(I)=IVAL(Inp_char$,10)
490 IF Port(I)<1 OR Port(I)>4 THEN Port_select
500 FOR J=1 TO I-1
510 IF Port(I)=Port(J) THEN Port_select
520 NEXT J
530 PRINT Port(I)
540 NEXT I
550 OFF ERROR
560 END IF
570 SUBEND
580 !=====
590 ! Electronic Calibration Function
600 !=====
610 SUB Ecal(@Agte507x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
620 DIM Buff$(9),Err_msg$(100)
630 INTEGER Err_no,Port1
640 !
650 PRINT "## Full "&VAL$(Num_of_ports)&" Port ECal ##"
660 !
670 OUTPUT @Agte507x;"*CLS"
```

```

680 SELECT Num_of_ports
690 CASE 1
700 PRINT "Connect Port "&VAL$(Port(1))&" to ECal Module."
710 PRINT "Then push [Enter] key."
720 INPUT "",Buff$
730 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:ECAL:SOLT1 ";Port(1)
740 CASE 2
750 PRINT "Connect Port "&VAL$(Port(1));
760 PRINT " and Port "&VAL$(Port(2))&" to ECal Module."
770 PRINT "Then push [Enter] key."
780 INPUT "",Buff$
790 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:ECAL:SOLT2 ";Port(1); ";Port(2)
800 CASE 3
810 PRINT "Connect Port "&VAL$(Port(1));
820 PRINT ", Port "&VAL$(Port(2));
830 PRINT " and Port "&VAL$(Port(3))&" to ECal Module."
840 PRINT "Then push [Enter] key."
850 INPUT "",Buff$
860 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:ECAL:SOLT3 ";Port(1); ";Port(2); ";Port(3)
870 CASE 4
880 PRINT "Connect Port 1, Port 2, Port 3 and Port 4 to to ECal Mod ule."
890 PRINT "Then push [Enter] key."
900 INPUT "",Buff$
910 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COLL:ECAL:SOLT4 1,2,3,4"
920 END SELECT
930 PRINT "Executing ..."
940 OUTPUT @Agte507x;".SYST:ERR?"
950 ENTER @Agte507x;Err_no,Err_msg$
960 IF Err_no<>0 THEN
970 PRINT "Error occurred!!"
980 PRINT " No: ";Err_no,"Description: "&Err_msg$
990 PRINT "ECAL INTERRUPT!!"
1000 ELSE
1010 PRINT "Done"
1020 END IF
1030 SUBEND

```

Power Calibration

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

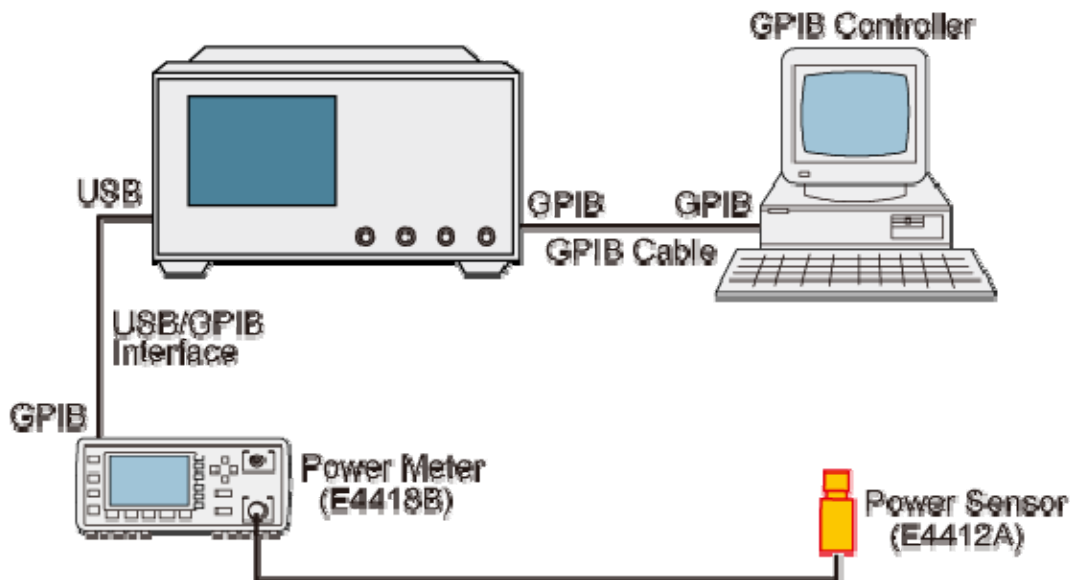
Other topics about Sample Programs

Overview

This program that demonstrates how to perform power calibration.

This program, as shown in the figure below, is run by making connections between the E5071C and the power meter (E4418B) through the USB/GPIB interface and between the E5071C and the external controller through the GPIB cable. Then the program executes the power calibration of the E5071C by using the power sensor (E4412A). The obtained power calibration data array is saved into a file.

Connecting E5071C, power meter, and external controller



e5071c334

See Power Calibration for this programming.

Sample Program in Excel VBA

```
Sub pow_cal_Click()
```

```
'
```

```
Dim defrm As Long
```

```
Dim vi As Long
```

```
Dim SwpType As String, StartPower As String, StopPower As String, CwFreq As String
```

```

Dim Nop As Long, NumOfAve As String, Limit As Double, CorrData() As Double
Dim Result As String * 10000, OpcRes As String * 2, Res As Variant
Dim i As Long, Stat As VbMsgBoxResult
Dim err As String * 50, ErrNo As Variant
Const TimeOutTime = 50000 ' TimeOut time should be greater than the measurement time.
' Assign a GPIB address to the I/O pass.
Call viOpenDefaultRM(defrm)
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
'

SwpType = "POW"      ' Sweep type          : POWER
Nop = 21             ' Number of points    : 41
StartPower = "-20"   ' Start Power       : -20 dBm
StopPower = "-10"    ' Stop Power        : -10 dBm
CwFreq = "1E9"       ' CW frequency      : 1 GHz
NumOfAve = "4"       ' Number of averaging : 4
Limit = 10           ' Limit for corrected data : 10 dBm
'

Call viVPrintf(vi, ":SYST:PRES" + vbLf, 0) ' Presetting the analyzer
Call viVPrintf(vi, ":SYST:COMM:GPIB:PMET:ADDR 13" + vbLf, 0) ' Setting GPIB address of the power
meter to ENA
'

' Setting measurement conditions
Call viVPrintf(vi, ":SENS1:SWE:TYPE " & SwpType & vbLf, 0)
Call viVPrintf(vi, ":SENS1:SWE:POIN " & CStr(Nop) & vbLf, 0)
Call viVPrintf(vi, ":SOUR1:POW:STAR " & StartPower & vbLf, 0)
Call viVPrintf(vi, ":SOUR1:POW:STOP " & StopPower & vbLf, 0)
Call viVPrintf(vi, ":SENS1:FREQ " & CwFreq & vbLf, 0)

Stat = MsgBox("Do you perform zeroing and calibrating the power sensor?", vbYesNo)

If Stat = vbYes Then
    MsgBox "Zero and calibrate the power sensor by using the power meter, then press [OK] key.",
vbOKOnly
End If

MeasStart:
'

```

E5071C

```
' Connecting the power sensor A to the port-1 of ENA
Call viVPrintf(vi, "**CLS" + vbLf, 0)
Stat = MsgBox("Set the power sensor connected to the port 1 in the ENA, then press [OK] key.",
vbOKOnly)
'
' Performing power calibration measurement
Call viVPrintf(vi, ":SOUR1:POW:PORT1:CORR:COLL:AVER " & NumOfAve & vbLf, 0)
Call viVPrintf(vi, ":SOUR1:POW:PORT1:CORR:COLL:ASEN" + vbLf, 0)
Call viVPrintf(vi, "**OPC?" + vbLf, 0)
Call viVScanf(vi, "%t", OpcRes)
'
' Error handling at power meter measurement
Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
ErrNo = Split(err, ",")
'
If Val(ErrNo(0)) = 0 Then
    ReDim CorrData(Nop)

    Call viVPrintf(vi, ":FORM:DATA ASC" + vbLf, 0)
    Call viVPrintf(vi, ":SOUR1:POW:PORT1:CORR:DATA?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Res = Split(Result, ",")

    If fnLim(vi, Nop, Limit, Res) Then
        MsgBox "Power meter calibration measurement is complete.", vbOKOnly
        For i = 1 To Nop
            Cells(i + 5, 2) = i
            Cells(i + 5, 3) = Res(i - 1)
        Next i
    Else
        GoTo ReCalibration
    End If
Else
    MsgBox "Error", vbOKOnly
    GoTo ReCalibration
End If
ProgEnd:
```



```

Call viClose(vi)
Call viClose(defrm)
Exit Sub
ReCalibration:
    Stat = MsgBox("Do you perform the power meter calibration measurement again?", vbYesNo)
    If Stat = vbYes Then
        GoTo MeasStart
    Else
        GoTo ProgEnd
    End If
End Sub
Function fnLim(vi As Long, Nop As Long, Limit As Double, Res As Variant) As Boolean
    For i = 1 To Nop
        If Abs(Res(i - 1)) > Limit Then
            Call viVPrintf(vi, "SOUR1:POW:PORT1:CORR OFF" + vbLf, 0)
            MsgBox "The corrected data is out of limit!", vbOKOnly
            fnLim = False
            Exit Function
        End If
    Next i
    fnLim = True
End Function

```

Sample Program in HT Basic (pow_cal.htb)

```

10 DIM Swp_type$(11),Inp_char$(9),Buff$(9),Err_mes$(50),File$(20)
20 DIM Corr_data(1:1601)
30 REAL Start_p,Stop_p,Cw_freq,Limit
40 INTEGER Nop,Pow_rang,Num_avg,Err_no,Verifier,Data_size,I
50 CLEAR SCREEN
60 !
70 ASSIGN @Agte507x TO 717
80 !
90 Swp_type$="POW" !Sweep type : POWER
100 Nop=41 !Number of points : 41
110 Pow_rang=0 !Power Range :-20 to +12 dBm
120 Start_p=-2.0E+1 !Start Power :-20 dBm
130 Stop_p=-1.0E+1 !Stop Power :-10 dBm
140 Cw_freq=1.0E+9 !CW frequency : 1 GHz

```

E5071C

```
150 Num_avg=4 !Number of averaging : 4
160 Limit=10 !limit for corrected data : 10 dBm
170 !
180 ! Presetting the analyzer
190 !
200 OUTPUT @Agte507x;":SYST:PRES"
210 !
220 ! Setting GPIB address of the power meter to E5071C
230 !
240 OUTPUT @Agte507x;":SYST:COMM:GPIB:PMET:ADDR 13"
250 !
260 ! Setting measurement conditions
270 !
280 OUTPUT @Agte507x;":SENS1:SWE:TYPE "&Swp_type$
290 OUTPUT @Agte507x;":SENS1:SWE:POIN ";Nop
300 OUTPUT @Agte507x;":SOUR1:POW:ATT ";Pow_rang
310 OUTPUT @Agte507x;":SOUR1:POW:STAR ";Start_p
320 OUTPUT @Agte507x;":SOUR1:POW:STOP ";Stop_p
330 OUTPUT @Agte507x;":SENS1:FREQ ";Cw_freq
340 !
350 PRINT "Do you perform zeroing and calibrating the power sensor?"
360 PRINT
370 INPUT "[Y/N]",Inp_char$
380 IF UPC$(Inp_char$)="Y" THEN
390 PRINT "Zero and calibrate the power sensor by using the power meter, then press [Enter] key."
400 PRINT
410 INPUT "",Inp_char$
420 END IF
430 !
440 Meas_start: !
450 !
460 ! Connecting the power sensor to the port 1 in the ENA
470 !
480 OUTPUT @Agte507x;"*CLS"
490 PRINT "Set the power sensor connected to the port 1 in the ENA, then press [Enter] key."
500 PRINT
510 INPUT "",Inp_char$
```

```

520 !
530 ! Performing power calibration measurement
540 !
550 OUTPUT @Agte507x;":SOUR1:POW:PORT1:CORR:COLL:AVER ";Num_avg
560 OUTPUT @Agte507x;":SOUR1:POW:PORT1:CORR:COLL:ASEN"
570 OUTPUT @Agte507x;""*OPC?"
580 ENTER @Agte507x;Buff$
590 !
600 ! Error handling at power meter measurement
610 !
620 OUTPUT @Agte507x;":SYST:ERR?"
630 ENTER @Agte507x;Err_no,Err_mes$
640 !
650 IF Err_no=0 THEN
660 REDIM Corr_data(1:Nop)
670 OUTPUT @Agte507x;":FORM:DATA ASC"
680 OUTPUT @Agte507x;":SOUR1:POW:PORT1:CORR:DATA?"
690 ENTER @Agte507x;Corr_data(*)
700 Verifier=FNLim(@Agte507x,Nop,Limit,Corr_data(*))
710 IF Verifier=-1 THEN
720 PRINT "Do you perform the power meter calibration measurement again?"
730 PRINT
740 INPUT "[Y/N]",Inp_char$
750 IF UPC$(Inp_char$)="Y" THEN GOTO Meas_start
760 IF UPC$(Inp_char$)<>"Y" THEN GOTO Prog_stop
770 END IF
780 PRINT "Power meter calibration measurement is complete."
790 PRINT
800 ELSE
810 PRINT "Error: "&Err_mes$
820 PRINT
830 PRINT "Do you perform the power meter calibration measurement again?"
840 PRINT
850 INPUT "[Y/N]",Inp_char$
860 IF UPC$(Inp_char$)="Y" THEN GOTO Meas_start
870 IF UPC$(Inp_char$)<>"Y" THEN GOTO Prog_stop
880 END IF

```

E5071C

```
890 !
900 ! Installing the corrected data to a file
910 !
920 File$="CORR_DATA"
930 Data_size=Nop*8
940 ON ERROR GOTO Skip_purge
950 PURGE File$
960 Skip_purge: OFF ERROR
970 PRINT "The file installing power correction data: "&File$
980 PRINT
990 CREATE File$,Data_size
1000 ASSIGN @File TO File$;FORMAT ON
1010 FOR I=1 TO Nop
1020 OUTPUT @File USING "3D,3X,MD.4DE";I,Corr_data(I)
1030 NEXT I
1040 ASSIGN @File TO *
1050 !
1060 PRINT "Installing the corrected data to the file is DONE."
1070 !
1080 GOTO Prog_end
1090 !
1100 Prog_stop: !
1110 PRINT "Program Interruption"
1120 !
1130 Prog_end: !
1140 !
1150 END
1160 !
1170 ! Limit Test Function for the Corrected Data
1180 !
1190 DEF FNLim(@Agte507x,INTEGER Nop,REAL Limit,REAL Corr_data(*))
1200 INTEGER I
1210 !
1220 FOR I=1 TO Nop
1230 IF ABS(Corr_data(I))>Limit THEN
1240 OUTPUT @Agte507x;"SOUR1:POW:PORT1:CORR OFF"
1250 PRINT "The corrected data is out of limit!"
```

```
1260 PRINT
1270 RETURN -1
1280 GOTO Fn_exit
1290 END IF
1300 NEXT I
1310 !
1320 RETURN 0
1330 !
1340 Fn_exit: !
1350 !
1360 FNEND
```

Reading/Writing Error Coefficient

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program reads/writes the error coefficient.

This program will set measurement conditions and perform full 2-port calibration, preset the E5071C with the read error coefficient to be written, and then again read the error coefficient.

- The error coefficient read from the E5071C will be displayed in a graph.

Sample Program in Excel VBA

```
Sub Err_Term_Click()
    Dim defrm As Long      'Session to Default Resource Manager
    Dim vi As Long         'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(2) As String
    Dim Result As String * 10
    Dim tNop As Long
    Dim Respons As String
    Dim Stimulus As String
    Dim ErrTerm As String

    Const TimeOutTime = 40000 'timeout time.
    Const Cal85032F = 4       'cal kit number

    Ch = Cells(2, 6)         'Select channel
    Port(1) = Cells(4, 6)    'Sets the select port 1.
    Port(2) = Cells(5, 6)    'Sets the select port 2.
    Respons = Cells(6, 6)    'Sets the respons port.
    Stimulus = Cells(7, 6)   'Sets the stimulus port.
    ErrTerm = Cells(8, 6)    'Sets the error term.

    CalKit = Cal85032F       'Set cal kit (85032F)
```

```

Call viOpenDefaultRM(defrm) 'Initializes the VISA system.
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi) 'Opens the session to the specified instrument.
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) 'The state of an attribute for the
specified session.

Call viVPrintf(vi, "**RST" & vbLf, 0) 'Presets the setting state of the ENA.
Call viVPrintf(vi, "**CLS" & vbLf, 0) 'Clears the all status register.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:CKIT " & CalKit & vbLf, 0) 'Select the calibration kit.

Call Set_sgm_tbl(vi, Ch) 'Configures the segment table.

Select Case Cells(3, 6) 'Sets the read/write.
    Case "Read"
        Call Cal_Slot(vi, Ch, 2, Port) 'Full 2-Port Calibration.
    Case "Write"
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF:METH:SOLT2 1,2" & vbLf, 0) 'Sets the
calibration type to the full 2-port calibration.
    End Select

    Call viVPrintf(vi, ":SENS" & Ch & ":SEGM:SWE:POIN?" & vbLf, 0) 'Reads out the total number of the
measurement points of all segments.
    Call viVScanf(vi, "%t", Result)

    Call Exec_Error_Term(vi, Ch, Val(Result), ErrTerm, Respons, Stimulus) 'Reads the error coefficient.

    Call viClose(vi) 'Closes the resource manager session.
    Call viClose(defrm) 'Breaks the communication and terminates the VISA system.

End
End Sub
Sub Exec_Error_Term(vi As Long, Ch As String, Nop As Long, ErrTerm As String, Respons As String,
Stimulus As String)

    Dim Error_Term_Data As Variant
    Dim Freq_Data As Variant
    Dim i As Integer, j As Integer
    Dim SelMode As String

```

E5071C

```
Dim Result As String * 10000
Dim RealData As Double
Dim ImagData As Double
Dim FreqData As Double

ReDim Error_Term_Data(Nop * 2) As String      'Defines the stock variables for the error coefficient
as needed for NOP.
ReDim Freq_Data(Nop) As String                'Defines the stock variables for the frequency values.

SelMode = Cells(3, 6) 'Reads the read/write mode.

Select Case SelMode
    Case "Read"                                'Reads the error coefficient from the ena.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF?" & ErrTerm & "," & Respons & "," & Stimulus &
vbLf, 0) 'Read the calibration coefficient data.
        Call viVScanf(vi, "%t", Result)
        Error_Term_Data = Split(Result, ",") 'Splits the read data by comma.

        Freq_Data = Make_Freq(vi, Nop)        'Calculates the frequency values.

        For i = 0 To Nop - 1
            RealData = CDbI(Error_Term_Data(i * 2)) 'Reads the real data from error coefficient items.
            ImagData = CDbI(Error_Term_Data(i * 2 + 1)) 'Reads the imag data from error coefficient
items.
            FreqData = CDbI(Freq_Data(i + 1))      'Reads the frequency values.
            Cells(10 + i, 2) = RealData            'Displays the real data to the excel sheet.
            Cells(10 + i, 3) = ImagData            'Displays the imag data to the excel sheet.
            Cells(10 + i, 1) = FreqData            'Displays the frequency values to the excel sheet.
        Next i

        Call Data_Plot(vi, Nop, ErrTerm)          'Displays the graph to the excel sheet.

    Case "Write" 'Write the error coefficient to the ena.
        Error_Term_Data = ErrTerm & "," & Respons & "," & Stimulus 'Sets the command parameter.
        For i = 0 To Nop - 1
            RealData = Cells(10 + i, 2)            'Retrieves the real data from the excel sheet.
            ImagData = Cells(10 + i, 3)            'Retrieves the imag data from the excel sheet.
```



```
Error_Term_Data = Error_Term_Data & "," & RealData & "," & ImagData    'Sets the
command parameter.
```

```
Next i
```

```
Call viVPrintf(vi, ".SENS" & Ch & ":CORR:COEF " & Error_Term_Data & vbLf, 0) 'Write the
calibration coefficient data.
```

```
Call viVPrintf(vi, ".SENS" & Ch & ":CORR:COEF:SAVE" & vbLf, 0)    'Calculates the
calibration coefficients.
```

```
End Select
```

```
End Sub
```

```
Function Make_Freq(vi As Long, tPoint As Long) As Variant
```

```
Dim start_freq As Double
```

```
Dim stop_freq As Double
```

```
Dim Nop As Integer
```

```
Dim fStep As Double
```

```
Dim fPoint As Double
```

```
Dim freq_array() As Variant
```

```
Dim MeasPoint As Integer
```

```
Const SegmentCnt = 2    'number of segment table.
```

```
ReDim freq_array(tPoint) As Variant
```

```
MeasPoint = 1
```

```
For j = 1 To SegmentCnt
```

```
start_freq = Cells(3 + j - 1, 9)
```

```
'Sets the start frequency of segment table.
```

```
stop_freq = Cells(3 + j - 1, 10)
```

```
'Sets the stop frequency of segment table.
```

```
Nop = Cells(3 + j - 1, 13)
```

```
'Sets the nop of segment table.
```

```
fStep = (stop_freq - start_freq) / (Nop - 1)
```

```
'Calculate the frequency step.
```

```
fPoint = start_freq
```

```
'Sets the frequency start point.
```

```
For i = 1 To Nop
```

```
freq_array(MeasPoint) = fPoint
```

```
'Sets the frequency value.
```

```
fPoint = fPoint + fStep
```

```
'Calculate the frequency points.
```

```
MeasPoint = MeasPoint + 1
```

```
'Add to measurement points.
```

```
Next i
```

E5071C

Next j

Make_Freq = freq_arr

'Sets the frequency data array.

End Function

Sub Data_Plot(vi As Long, Nop As Long, ErrTerm As String)

Range("B10:C" & Nop + 9 & "").Select 'Select the error coefficient.

Charts.Add

ActiveChart.ChartType = xlLineStacked 'Sets the chart type.

ActiveChart.SetSourceData Source:=Sheets("Error Term").Range("A9:C" & Nop + 9 & "") 'Sets the error coefficient and displays the graph.

ActiveChart.Location Where:=xlLocationAsObject, Name:="Error Term"

ActiveChart.Axes(xlCategory).Select 'Select the formatting of X-axis.

With Selection

.TickLabelPosition = xlLow 'Displays the frequency values to low area.

End With

With ActiveChart

.HasTitle = True

.ChartTitle.Characters.Text = "Error Term " & ErrTerm 'Display the title.

End With

End Sub

Sub Set_sgm_tbl(vi As Long, Ch As String)

Dim Star1(2) As Double, Stop1(2) As Double, Pow1(2) As Double, If_bw1(2) As Double

Dim Segm As Integer, Nop1(2) As Integer, Num_of_tr1 As Integer

Dim i As Integer

Segm = 2

Star1(1) = Cells(3, 9) 'Sets the start frequency of segment 1 table.

Stop1(1) = Cells(3, 10) 'Sets the stop frequency of segment 1 table.

Pow1(1) = Cells(3, 11) 'Sets the power of segment 1 table.

If_bw1(1) = Cells(3, 12) 'Sets the ifbw of segment 1 table.

Nop1(1) = Cells(3, 13) 'Sets the nop of segment 1 table.

Star1(2) = Cells(4, 9) 'Sets the start frequency of segment 2 table.

Stop1(2) = Cells(4, 10) 'Sets the stop frequency of segment 2 table.

Pow1(2) = Cells(4, 11) 'Sets the power of segment 2 table.

If_bw1(2) = Cells(4, 12) 'Sets the ifbw of segment 2 table.

Nop1(2) = Cells(4, 13) 'Sets the nop of segment 2 table.

Call viVPrintf(vi, ":SENS" & Ch & ":SWE:TYPE SEGM" & vbLf, 0) 'Sets sweep type to segment.

Call viVPrintf(vi, ":SENS" & Ch & ":SEGM:DATA 5,0,1,1,0,0," & Segm & ",", 0) 'Sets the header of segment table.

Call viVPrintf(vi, Star1(1) & "," & Stop1(1) & "," & Nop1(1) & "," & If_bw1(1) & "," & Pow1(1) & ",", 0)
'Sets the 1st parameter.

Call viVPrintf(vi, Star1(2) & "," & Stop1(2) & "," & Nop1(2) & "," & If_bw1(2) & "," & Pow1(2) & vbLf, 0)
'Sets the 2nd parameter.

Call ErrorCheck(vi) 'Checking the error.

End Sub

Sub Cal_Slot(vi As Long, Ch As String, NumPort As String, Port() As String)

Dim Dummy

Dim i As Integer, j As Integer

Select Case NumPort

Case 1

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & vbLf, 0) 'Set the 1-port calibration type.

Case 2

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & vbLf, 0) 'Set the full 2-port calibration type.

Case 3

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & "," & Port(3) & vbLf, 0) 'Set the full 3-port calibration type.

Case 4

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT4 1,2,3,4" & vbLf, 0) 'Set the full 4-port calibration type.

End Select

'Reflection

For i = 1 To NumPort

MsgBox ("Set Open to Port " & Port(i) & ". then click [OK] button") 'Display the message box.

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:OPEN " & Port(i) & vbLf, 0) 'Measurement the OPEN calibration.

Call viVQueryf(vi, "**OPC?" & vbLf, "%t", Dummy) 'Reads the *OPC? result.

E5071C

```
MsgBox ("Set Short to Port " & Port(i) & ". then click [OK] button") 'Display the message box.
Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SHORT " & Port(i) & vbCrLf, 0) 'Measurement the
SHORT calibration.
Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.

MsgBox ("Set Load to Port " & Port(i) & ". then click [OK] button") 'Display the message box.
Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:LOAD " & Port(i) & vbCrLf, 0) 'Measurement the
LOAD calibration.
Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.
Next i
'Transmission
For i = 1 To NumPort - 1
    For j = i + 1 To NumPort
        MsgBox ("Set Thru to Port " & Port(i) & "&" & Port(j) & ". then click [OK] button") 'Display the
        message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(i) & "," & Port(j) & vbCrLf, 0)
        'Measurement the THRU calibration.
        Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(j) & "," & Port(i) & vbCrLf, 0)
        'Measurement the THRU calibration.
        Call viVQueryf(vi, "**OPC?" & vbCrLf, "%t", Dummy) 'Reads the *OPC? result.
    Next j
Next i

Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbCrLf, 0) 'Calculating the calibration
coefficients.

Call ErrorCheck(vi) 'Checking the error.

End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbCrLf, "%t", err) 'Reads error message.
    ErrNo = Split(err, ",") 'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly) 'Display the message box.
    End If
```

End Sub

Sample Program in HT Basic (ErrTerm.hrb)

```

2000 Main:
2010 INTEGER Agte507x,li,Nop
2020 INTEGER Respons,Stimulas
2030 INTEGER Port(1:2)
2040 REAL Stok(12,1:5000)
2050 REAL Stok2(12,1:5000)
2060 REAL Stok3(12,1:5000)
2070 DIM Ch$[10],Wk$[128]
2080 !
2090 ! PC's Monitor Clear
2100 CLEAR SCREEN
2110 GINIT
2130 !
2140 ! Set ENA++'s Addr
2150 Agte507x=717
2160 !
2170 Ch$="1"
2180 !
2190 ! Set ENA++'s I/O Path
2200 ASSIGN @Agte507x TO Agte507x
2210 !
2220 ON TIMEOUT SC(@Agte507x),15 RECOVER Tout
2230 !
2240 ! Set Start Port and End Port
2250 Port(1)=1
2260 Port(2)=2
2270 !
2280 ! Setup Segment Table
2290 CALL Set_sgm_tbl(@Agte507x)
2300 !
2310 ! Select Cal Kit
2320 CALL Select_cal_kit(@Agte507x,Ch$)
2330 !
2340 ! Execute Full-2Port Calibration
2350 CALL Cal_solt(@Agte507x,Ch$,2,Port(*))

```

E5071C

2360 !

2370 ! Get All Segment's Points

2380 CALL Get_nop(@Agte507x,Nop,Ch\$)

2390 !

2400 REDIM Stok(12,1:Nop*2)

2410 REDIM Stok2(12,1:Nop*2)

2420 REDIM Stok3(12,1:Nop*2)

2430 !

2440 CALL Exec_error_term(@Agte507x,"READ","ES",Ch\$,1,Nop,1,1,Stok(*))

2450 CALL Exec_error_term(@Agte507x,"READ","ES",Ch\$,2,Nop,2,2,Stok(*))

2460 CALL Exec_error_term(@Agte507x,"READ","ER",Ch\$,3,Nop,1,1,Stok(*))

2470 CALL Exec_error_term(@Agte507x,"READ","ER",Ch\$,4,Nop,2,2,Stok(*))

2480 CALL Exec_error_term(@Agte507x,"READ","ED",Ch\$,5,Nop,1,1,Stok(*))

2490 CALL Exec_error_term(@Agte507x,"READ","ED",Ch\$,6,Nop,2,2,Stok(*))

2500 !

2510 CALL Exec_error_term(@Agte507x,"READ","EL",Ch\$,7,Nop,1,2,Stok(*))

2520 CALL Exec_error_term(@Agte507x,"READ","EL",Ch\$,8,Nop,2,1,Stok(*))

2530 CALL Exec_error_term(@Agte507x,"READ","ET",Ch\$,9,Nop,1,2,Stok(*))

2540 CALL Exec_error_term(@Agte507x,"READ","ET",Ch\$,10,Nop,2,1,Stok(*))

2550 !

2560 CLEAR SCREEN

2570 PRINT "Push [Preset] - OK of ENA. Then push [Enter] key."

2580 INPUT "",Wk\$

2590 !

2600 CALL Set_sgm_tbl(@Agte507x)

2610 !

2620 OUTPUT @Agte507x;":SENS"&Ch\$&":CORR:COEF:METH:SOLT2 ";Port(1);";Port(2)

2630 !

2640 CALL Exec_error_term(@Agte507x,"WRITE","ES",Ch\$,1,Nop,1,1,Stok(*))

2650 CALL Exec_error_term(@Agte507x,"WRITE","ES",Ch\$,2,Nop,2,2,Stok(*))

2660 CALL Exec_error_term(@Agte507x,"WRITE","ER",Ch\$,3,Nop,1,1,Stok(*))

2670 CALL Exec_error_term(@Agte507x,"WRITE","ER",Ch\$,4,Nop,2,2,Stok(*))

2680 CALL Exec_error_term(@Agte507x,"WRITE","ED",Ch\$,5,Nop,1,1,Stok(*))

2690 CALL Exec_error_term(@Agte507x,"WRITE","ED",Ch\$,6,Nop,2,2,Stok(*))

2700 !

2710 CALL Exec_error_term(@Agte507x,"WRITE","EL",Ch\$,7,Nop,1,2,Stok(*))

2720 CALL Exec_error_term(@Agte507x,"WRITE","EL",Ch\$,8,Nop,2,1,Stok(*))

```

2730 CALL Exec_error_term(@Agte507x,"WRITE","ET",Ch$,9,Nop,1,2,Stok(*))
2740 CALL Exec_error_term(@Agte507x,"WRITE","ET",Ch$,10,Nop,2,1,Stok(*))
2750 !
2760 OUTPUT @Agte507x;".SENS"&Ch$&".CORR:COEF:SAVE"
2770 !
2780 CALL Exec_error_term(@Agte507x,"READ","ES",Ch$,1,Nop,1,1,Stok2(*))
2790 CALL Exec_error_term(@Agte507x,"READ","ES",Ch$,2,Nop,2,2,Stok2(*))
2800 CALL Exec_error_term(@Agte507x,"READ","ER",Ch$,3,Nop,1,1,Stok2(*))
2810 CALL Exec_error_term(@Agte507x,"READ","ER",Ch$,4,Nop,2,2,Stok2(*))
2820 CALL Exec_error_term(@Agte507x,"READ","ED",Ch$,5,Nop,1,1,Stok2(*))
2830 CALL Exec_error_term(@Agte507x,"READ","ED",Ch$,6,Nop,2,2,Stok2(*))
2840 !
2850 CALL Exec_error_term(@Agte507x,"READ","EL",Ch$,7,Nop,1,2,Stok2(*))
2860 CALL Exec_error_term(@Agte507x,"READ","EL",Ch$,8,Nop,2,1,Stok2(*))
2870 CALL Exec_error_term(@Agte507x,"READ","ET",Ch$,9,Nop,1,2,Stok2(*))
2880 CALL Exec_error_term(@Agte507x,"READ","ET",Ch$,10,Nop,2,1,Stok2(*))
2890 !
2900 ASSIGN @Agte507x TO *
2910 !
2920 DISP CHR$(139)&" Done ..."&CHR$(136)
2930 STOP
2940 !
2950 Tout: OFF TIMEOUT SC(@Agte507x)
2960 !
2970 ASSIGN @Agte507x TO *
2980 !
2990 PRINT CHR$(137)&" ENA Timeout ..."&CHR$(136)
3000 END
3010!
3020 Set_sgm_tbl: SUB Set_sgm_tbl(@Agte507x)
3030 REAL Star1(1:2),Stop1(1:2),Pow1(1:2)
3040 INTEGER Segm,Nop1(1:2),Num_of_tr1
3050 INTEGER I
3060 !
3070 CLEAR SCREEN
3080 DISP CHR$(138)&" Wait ..."&CHR$(136)
3090 !

```

E5071C

```
3100 Segm=2 ! Number of Segment Ch.1 : 2
3110 Star1(1)=3.E+6 ! Start Frequency Ch.1 Segm.1: 3.0 MHz
3120 Star1(2)=5.0E+7 ! Segm.2: 50.0 MHz
3130 Stop1(1)=1.0E+7 ! Stop Frequency Ch.1 Segm.1: 10.0 MHz
3140 Stop1(2)=8.E+9 ! Segm.2: 8.0 GHz
3150 Nop1(1)=2 ! Number Ch.1 Segm.1: 2
3160 Nop1(2)=170 ! of Points Segm.2: 170
3170 If_bw1(1)=7.0E+4 ! IF Bandwidth Ch.1 Segm.1: 70 kHz
3180 If_bw1(2)=7.0E+4 ! Segm.2: 70 kHz
3190 Pow1(1)=0 ! Power Ch.1 Segm.1: 0 dBm
3200 Pow1(2)=0 ! Segm.2: 0 dBm
3210 !
3220 OUTPUT @Agte507x;":SYST:PRES"
3230 !
3240 WAIT 5
3250 !
3260 ! Channel 1
3270 !
3280 OUTPUT @Agte507x;":SENS1:SWE:TYPE SEGM"
3290 OUTPUT @Agte507x;":SENS1:SEGM:DATA 5,0,1,1,0,0,";Segm;",";
3300 FOR I=1 TO Segm-1
3310 OUTPUT @Agte507x;Star1(I);",";Stop1(I);",";Nop1(I);",";If_bw1(I);",";Pow1(I);",";
3320 NEXT I
3330 OUTPUT
@Agte507x;Star1(Segm);",";Stop1(Segm);",";Nop1(Segm);",";If_bw1(Segm);",";Pow1(Segm)
3340 !
3350 OUTPUT @Agte507x;":CALC1:PAR:COUN ";Num_of_tr1
3360 FOR I=1 TO Num_of_tr1
3370 OUTPUT @Agte507x;":CALC1:PAR"&VAL$(I)&":SEL"
3380 NEXT I
3390 SUBEND
3400!
3410 Select_cal_kit: SUB Select_cal_kit(@Agte507x,Ch$)
3420 !=====
3430 ! Calibration Kit Selection Function
3440 !=====
3450 !
```



```

3460 DIM Cal_kit_lbl$(1:10)[20],Inp_char$(9)
3470 DIM Msg$(80),Wk$(10)
3480 INTEGER Cal_kit,I,Noc
3490 !
3500 ! PC's Monitor Clear
3510 CLEAR SCREEN
3520 !
3530 ! Number of Cal Kid
3540 Noc=10
3550 !
3560 FOR I=1 TO Noc
3570 OUTPUT @Agte507x;".SENS1:CORR:COLL:CKIT ";I
3580 OUTPUT @Agte507x;".SENS1:CORR:COLL:CKIT:LAB?"
3590 ENTER @Agte507x;Cal_kit_lbl$(I)
3600 NEXT I
3610 ON ERROR GOTO Kit_select
3620 !
3630 PRINT "## Calibration Kit Selection ##"
3640 FOR I=1 TO Noc
3650 PRINT USING "X,2D,A,X,20A";I,".",Cal_kit_lbl$(I)
3660 NEXT I
3670 PRINT ""
3680 PRINT "Input 1 to "&VAL$(Noc)
3690 !
3700 Msg$="Input number? (1 to "&VAL$(Noc)&") "
3710 LOOP
3720 LOOP
3730 DISP Msg$;
3740 INPUT Inp_char$
3750 Cal_kit=IVAL(Inp_char$,10)
3760 EXIT IF 1<=Cal_kit AND Cal_kit<=Noc
3770 Kit_select:!
3780 BEEP
3790 END LOOP
3800 !
3810 Wk$=""
3820 PRINT TABXY(1,Cal_kit+1);

```

E5071C

```
3830 PRINT USING "X,B,2D,A,X,20A,B";139,Cal_kit,":",Cal_kit_lbl$(Cal_kit),136
3840 INPUT "Sure ? [Y/N]",Wk$
3850 EXIT IF (UPC$(Wk$)="Y")
3860 PRINT TABXY(1,Cal_kit+1);
3870 PRINT USING "X,2D,A,X,20A";Cal_kit,":",Cal_kit_lbl$(Cal_kit)
3880 BEEP
3890 BEEP
3900 END LOOP
3910 OFF ERROR
3920 !
3930 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:CKIT ";Cal_kit
3940 SUBEND
3950!
3960 Cal_solt: SUB Cal_solt(@Agte507x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
3970 !======
3980 ! Full n Port Calibration Function
3990 !======
4000 !
4010 DIM Buff$(9)
4020 INTEGER I,J
4030 !
4040 ! PC's Monitor Clear
4050 CLEAR SCREEN
4060 !
4070 PRINT "## Full "&VAL$(Num_of_ports)&" Port Calibration ##"
4080 !
4090 ! Calibration Type Selection
4100 !
4110 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:METH:SOLT"&VAL$(Num_of_ports)&" ";
4120 FOR I=1 TO Num_of_ports-1
4130 OUTPUT @Agte507x;Port(I);";";
4140 NEXT I
4150 OUTPUT @Agte507x;Port(Num_of_ports)
4160 !
4170 ! Reflection Measurement
4180 !
4190 FOR I=1 TO Num_of_ports
```

```

4200 PRINT "Set OPEN to Port "&VAL$(Port(I))&". Then push [Enter] key."
4210 INPUT "",Buff$
4220 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:OPEN ";Port(I)
4230 OUTPUT @Agte507x;"*OPC?"
4240 ENTER @Agte507x;Buff$
4250 PRINT "Set SHORT to Port "&VAL$(Port(I))&". Then push [Enter] key."
4260 INPUT "",Buff$
4270 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SHOR ";Port(I)
4280 OUTPUT @Agte507x;"*OPC?"
4290 ENTER @Agte507x;Buff$
4300 PRINT "Set LOAD to Port "&VAL$(Port(I))&". Then push [Enter] key."
4310 INPUT "",Buff$
4320 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:LOAD ";Port(I)
4330 OUTPUT @Agte507x;"*OPC?"
4340 ENTER @Agte507x;Buff$
4350 NEXT I
4360 !
4370 ! Transmission Measurement
4380 !
4390 FOR I=1 TO Num_of_ports-1
4400 FOR J=I+1 TO Num_of_ports
4410 PRINT "Set THRU between Port "&VAL$(Port(I))&" and Port "&VAL$(Port(J))&". Then push [Enter]
key."
4420 INPUT "",Buff$
4430 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(I);",";Port(J)
4440 OUTPUT @Agte507x;"*OPC?"
4450 ENTER @Agte507x;Buff$
4460 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(J);",";Port(I)
4470 OUTPUT @Agte507x;"*OPC?"
4480 ENTER @Agte507x;Buff$
4490 NEXT J
4500 NEXT I
4510 !
4520 ! Done
4530 !
4540 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SAVE"
4550 PRINT "Done"

```

E5071C

4560 SUBEND

4570!

4580 Get_nop: SUB Get_nop(@Agte507x,INTEGER Nop,Ch\$)

4590 ! Get All Segment's Points

4600 OUTPUT @Agte507x;":SENS"&Ch\$&":SEGM:SWE:POIN?"

4610 ENTER @Agte507x;Nop

4620 SUBEND

4630 Exec_error_term: SUB Exec_error_term(@Agte507x,Rw\$,Id\$,Ch\$,INTEGER
Idx,Nop,Respons,Stimulus,REAL Stok(*))

4640 INTEGER li

4650 REAL Error_term_data(1:5000)

4660 !

4670 DISP CHR\$(138)&" Wait ... "&CHR\$(136)

4680 !

4690 REDIM Error_term_data(1:Nop*2)

4700 !

4710 SELECT Rw\$

4720 CASE "WRITE"

4730 FOR li=1 TO Nop

4740 Error_term_data(2*li-1)=Stok(Idx,2*li-1)

4750 Error_term_data(2*li)=Stok(Idx,2*li)

4760 NEXT li

4770 !

4780 OUTPUT @Agte507x;":SENS"&Ch\$&":CORR:COEF
"&Id\$&",";Respons;",";Stimulus;",";Error_term_data(*)

4790 !

4800 CASE "READ"

4810 FOR li=1 TO Nop

4820 Error_term_data(2*li-1)=-999

4830 Error_term_data(2*li)=-999

4840 NEXT li

4850 !

4860 OUTPUT @Agte507x;":SENS"&Ch\$&":CORR:COEF? "&Id\$&",";Respons;",";Stimulus

4870 ENTER @Agte507x;Error_term_data(*)

4880 !

4890 CALL Data_plot(Id\$,Respons,Stimulus,Nop,Error_term_data(*))

4900 !

4910 FOR li=1 TO Nop

```

4920 Stok(Idx,2*li-1)=Error_term_data(2*li-1)
4930 Stok(Idx,2*li)=Error_term_data(2*li)
4940 NEXT li
4950 !
4960 END SELECT
4970 SUBEND
4980!
4990 Data_plot: SUB Data_plot(Error_term$,INTEGER Respons,Stimulus,Nop,REAL
Error_term_data(*))
5000 INTEGER li,Pen(1:2)
5010 REAL Y_minmax(1:2)
5020 DIM Wk$[20]
5030 !
5040 CLEAR SCREEN
5050 GINIT
5060 GCLEAR
5070 !
5080 Pen(1)=3
5090 Pen(2)=4
5100 !
5110 ! Get Min Value and Max Value from all data
5120 Y_minmax(1)=MIN(Error_term_data(*))
5130 Y_minmax(2)=MAX(Error_term_data(*))
5150 !
5160 IF (Y_minmax(1)=Y_minmax(2)) AND (Y_minmax(1)=0) THEN
5170 Y_minmax(1)=1
5180 Y_minmax(2)=-1
5190 ELSE
5200 IF (Y_minmax(1)=Y_minmax(2)) THEN
5210 Y_minmax(1)=Y_minmax(1)*.5
5220 Y_minmax(2)=Y_minmax(2)*1.5
5230 END IF
5240 END IF
5250 !
5260 VIEWPORT 25*RATIO,80*RATIO,40,90
5270 WINDOW 1,Nop,Y_minmax(1),Y_minmax(2)
5280 FRAME

```

E5071C

5290 !

5300 VIEWPORT 80*RATIO,100*RATIO,40,90

5310 WINDOW 0,2,0,2

5320 PEN Pen(1)

5330 CSIZE 2.5

5340 LONG 2

5350 MOVE .2,1.5

5360 DRAW .4,1.5

5370 MOVE .5,1.5

5380 PEN 1

5390 LABEL ":Real Value"

5400 !

5410 PEN Pen(2)

5420 MOVE .2,1

5430 DRAW .4,1

5440 MOVE .5,1

5450 PEN 1

5460 LABEL ":Image Value"

5470 !

5480 VIEWPORT 25*RATIO,80*RATIO,90,100

5490 WINDOW 0,2,0,2

5500 CSIZE 3

5510 LONG 5

5520 MOVE 1,1.2

5530 LABEL "Error Term:"&Error_term\$

5540 !

5550 MOVE 1,5

5560 LABEL "Respons Port:"&VAL\$(Respons)&" Stimulus Port:"&VAL\$(Stimulus)

5570 !

5580 VIEWPORT 0,25*RATIO,40,90

5590 WINDOW 0,2,0,2

5600 CLIP -10,10,-10,10

5610 LONG 8

5620 CSIZE 3

5630 !

5640 MOVE 1.9,0

5650 LABEL VAL\$(Y_minmax(1))

```

5660 MOVE 1,9,2
5670 LABEL VAL$(Y_minmax(2))
5680 !
5690 VIEWPORT 25*RATIO,80*RATIO,30,40
5700 WINDOW 0,2,0,2
5710 CLIP -10,10,-10,10
5720 LONG 5
5730 MOVE 0,1.5
5740 LABEL VAL$(1)
5750 MOVE 2,1.5
5760 LABEL VAL$(Nop)
5770 !
5780 VIEWPORT 25*RATIO,80*RATIO,40,90
5790 WINDOW 1,Nop,Y_minmax(1),Y_minmax(2)
5800 FOR li=2 TO Nop
5820 PEN Pen(1)
5830 MOVE li-1,Error_term_data(2*(li-1)-1)
5840 DRAW li,Error_term_data(2*li-1)
5860 !
5870 PEN Pen(2)
5880 MOVE li-1,Error_term_data(2*(li-1))
5890 DRAW li,Error_term_data(2*li)
5900 NEXT li
5910 !
5920 PEN 1
5930 BEEP
5940 INPUT "Cont:push [Enter] key",Wk$
5950 SUBEND
5960!

```

Waiting for Trigger (OPC?)

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use the **:TRIG:SING** command to wait until the measurement cycle is completed.

- This sample program correctly runs when the maximum number of channels/traces is set to 9 channels/9 traces.

The sample program uses the **:TRIG:SING** command to start a sweep (measurement) cycle, uses the ***OPC** command to wait until the measurement cycle is completed, then prints a message and exits.

See [Waiting for the End of Measurement](#) for this programming.

Sample Program in Excel VBA

```
Sub trg_sing_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim ContMode(9) As String
    Dim Result As String * 10
    Dim i As Integer
    Const TimeOutTime = 100000 ' TimeOut time should be greater than the measurement time.
    '
    ' Assign a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Store the settings of continuous initiation mode for eachchannel
    ' (on for channels 1 and 2; off for channels 3 through 9)
    ' into the array variable ContMode().
    ContMode(1) = "ON"
    ContMode(2) = "ON"
    For i = 3 To 9
        ContMode(i) = "OFF"
    Next i
```



```

'
' Turn on or off continuous initiation mode for each channel
' depending on the value of ContMode(*).
For i = 1 To 9
    Call viVPrintf(vi, ":INIT" & CStr(i) & ":CONT " & ContMode(i) & vbLf, 0)
Next i
' Set the trigger source to Bus Trigger.
Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)
'
' Trigger the instrument to start a sweep cycle.
Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)
'
' Execute the *OPC? command and wait until the command
' returns 1 (i.e., the measurement cycle is completed).
Call viVPrintf(vi, "**OPC?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
'
' Display a measurement completion message.
Stat = MsgBox("Measurement complete", vbOKOnly)
Call viClose(vi)
Call viClose(defrm)
End Sub

```

Sample Program in HT Basic (trg_sing.htb)

```

10 DIM Cont_mode$(1:9)[9],Buff$(9)
20 INTEGER I
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 Cont_mode$(1)="ON"
70 Cont_mode$(2)="ON"
80 Cont_mode$(3)="OFF"
90 Cont_mode$(4)="OFF"
100 Cont_mode$(5)="OFF"
110 Cont_mode$(6)="OFF"
120 Cont_mode$(7)="OFF"
130 Cont_mode$(8)="OFF"
140 Cont_mode$(9)="OFF"

```

E5071C

```
150 !
160 FOR I=1 TO 9
170 OUTPUT @Agte507x;":INIT"&VAL$(I)&":CONT "&Cont_mode$(I)
180 NEXT I
190 OUTPUT @Agte507x;":TRIG:SOUR BUS"
200 !
210 OUTPUT @Agte507x;":TRIG:SING"
220 OUTPUT @Agte507x;":*OPC?"
230 ENTER @Agte507x;Buff$
240 !
250 PRINT "Measurement complete"
260 END
```

Waiting for Trigger (SRQ)

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use an SRQ to detect the end of measurement.

- This sample program correctly runs when the maximum number of channels/traces is set to 9 channels/9 traces.

The sample program sets up the trigger system, configures the instrument to properly generate an SRQ, and then triggers the instrument. When the instrument has generated an SRQ that indicates the end of measurement, the program exits after printing a measurement completion message.

See Waiting for the End of Measurement for this programming.

Sample Program in Excel VBA

```
Sub srq_meas_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim ContMode(9) As String
    Dim Result As String * 10
    Dim i As Integer, StbStatus As Integer
    Const TimeOutTime = 100000 ' TimeOut time should be greater than the measurement time.
    '
    ' Assign a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Store the settings of continuous initiation mode for eachchannel
    ' (on for channels 1 and 2; off for channels 3 through 9)
    ' into the array variable ContMode().
    ContMode(1) = "ON"
    ContMode(2) = "ON"
    For i = 3 To 9
        ContMode(i) = "OFF"
```

E5071C

Next i

,

' Turn on or off continuous initiation mode for each channel

' depending on the value of ContMode(*).

For i = 1 To 9

 Call viVPrintf(vi, ":INIT" & CStr(i) & ":CONT " & ContMode(i) & vbCrLf, 0)

Next i

,

' Set the trigger source to Bus Trigger.

Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbCrLf, 0)

,

,

Call viVPrintf(vi, ":STAT:OPER:PTR 0" & vbCrLf, 0) 'Set 0 at all bits of Position Transition Filter

Call viVPrintf(vi, ":STAT:OPER:NTR 16" & vbCrLf, 0) 'Set 1 at bit 4 of Negative Transition Filter

Call viVPrintf(vi, ":STAT:OPER:ENAB 16" & vbCrLf, 0) 'Set 1 at bit 4 of Operation status enable

Call viVPrintf(vi, "**SRE 128" & vbCrLf, 0) 'Set 1 at bit 7 of Service Request Enable Register

Call viVPrintf(vi, "**CLS" & vbCrLf, 0) ' Clear Register.

,

Call viVPrintf(vi, "**TRG" & vbCrLf, 0) 'Make a trigger

' Wait until Status Byte Register became 192

Do

 Call viReadSTB(vi, StbStatus) ' Read Status Byte Register

 Range("B5").Value = StbStatus

Loop Until StbStatus = 192

,

' Display a measurement completion message.

Stat = MsgBox("Measurement complete", vbOKOnly)

' Close IO

Call viClose(vi)

Call viClose(defrm)

End Sub

Sample Program in HT Basic (srq_meas.htb)

10 DIM Cont_mode\$(1:9)[9],Buff\$(9)

20 INTEGER I

30 !

40 ASSIGN @Agte507x TO 717

50 !

```

60 Cont_mode$(1)="ON"
70 Cont_mode$(2)="ON"
80 Cont_mode$(3)="OFF"
90 Cont_mode$(4)="OFF"
100 Cont_mode$(5)="OFF"
110 Cont_mode$(6)="OFF"
120 Cont_mode$(7)="OFF"
130 Cont_mode$(8)="OFF"
140 Cont_mode$(9)="OFF"
150 !
160 FOR I=1 TO 9
170 OUTPUT @Agte507x;".INIT"&VAL$(I)&".CONT "&Cont_mode$(I)
180 NEXT I
190 OUTPUT @Agte507x;".TRIG:SOUR BUS"
200 !
210 OUTPUT @Agte507x;".STAT:OPER:PTR 0"
220 OUTPUT @Agte507x;".STAT:OPER:NTR 16"
230 OUTPUT @Agte507x;".STAT:OPER:ENAB 16"
240 OUTPUT @Agte507x;"*SRE 128"
250 OUTPUT @Agte507x;"*CLS"
260 OUTPUT @Agte507x;"*OPC?"
270 ENTER @Agte507x;Buff$
280 !
290 ON INTR 7 GOTO Meas_end
300 ENABLE INTR 7;2
310 OUTPUT @Agte507x;"*TRG"
320 PRINT "Waiting..."
330 Meas_wait: GOTO Meas_wait
340 Meas_end: OFF INTR 7
350 PRINT "Measurement Complete"
360 END

```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 140

E5071C

These lines store the settings of continuous initiation mode for each channel (on for channels 1 and 2; off for channels 3 through 9) into the array variable Cont_mode\$(*).

Lines 160 to 180

These lines turn on or off continuous initiation mode for each channel depending on the value of Cont_mode\$(*).

Line 190

Sets the trigger source to "Bus Trigger".

Lines 210 to 220

These lines configure the instrument so that operation status event register's bit 4 is set to 1 only when the operation status condition register's bit 4 is changed from 1 to 0 (negative transition).

Lines 230 to 240

These lines enable the operation status event register's bit 4 and status byte register's bit 7.

Lines 250 to 270

These lines clear the status byte register and operation status event register.

Lines 290 to 300

These lines set the branch target for an SRQ interrupt to enable SRQ interruptions.

Lines 310 to 320

These lines trigger the instrument and wait until the measurement cycle finishes.

Line 350

Displays a measurement completion message.

Error Detection (SRQ)

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use an SRQ to detect the occurrence of an error.

This program sets SRQs and then intentionally sends an invalid parameter to generate an error to be handled by this program. In the error handling part, this program examines the error, displays the error number and error message, and then displays the message indicating the suspension of the program. See Detecting Occurrence of an Error for this programming.

- The sequence interception by an error can not be performed on Excel VBA.

Sample Program in HT Basic (srq_err.bas)

```

10 DIM Buff$(9),Err_mes$(50)
20 INTEGER Err_no
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 OUTPUT @Agte507x;"*ESE 60"
70 OUTPUT @Agte507x;"*SRE 32"
80 OUTPUT @Agte507x;"*CLS"
90 OUTPUT @Agte507x;"*OPC?"
100 ENTER @Agte507x;Buff$
110 !
120 ON INTR 7 GOTO Err_proc
130 ENABLE INTR 7;2
140 OUTPUT @Agte507x;".CALC1:PAR:COUN 2"
150 PRINT "Trace 1 Meas.Para: S21"
160 PRINT "Trace 1 Format : Log Mag"
170 OUTPUT @Agte507x;".CALC1:PAR1:DEF S21"
180 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
190 OUTPUT @Agte507x;".CALC1:FORM MLOG"
200 PRINT "Trace 2 Meas.Para: S11"
210 PRINT "Trace 2 Format : Log Mag"
220 OUTPUT @Agte507x;".CALC1:PAR2:DEF S11"

```

E5071C

```
230 OUTPUT @Agte507x;";CALC1:PAR2:SEL"
240 OUTPUT @Agte507x;";CALC1:FORM LOG"
250 OUTPUT @Agte507x;"*OPC?"
260 ENTER @Agte507x;Buff$
270 GOTO Skip_err_proc
280 Err_proc: OFF INTR 7
290 OUTPUT @Agte507x;";SYST:ERR?"
300 ENTER @Agte507x;Err_no,Err_mes$
310 PRINT "Error occurred!!"
320 PRINT " No: ";Err_no,"Description: "&Err_mes$
330 PRINT "PROGRAM INTERRUPT!!"
340 GOTO Prog_end
350 Skip_err_proc: PRINT "PROGRAM DONE."
360 Prog_end: END
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 70

These lines enable bits 2, 3, 4 and 5 in the standard event status register and set bit 5 to 1 in the service request enable register.

Lines 80 to 100

These lines clear the status byte register, the standard event status register, and the error queue.

Lines 120 to 130

These lines set the branch target for an SRQ interrupt to enable SRQ interruptions.

Lines 140 to 260

These lines set the measurement parameters and their data formats for traces 1 and 2. An invalid parameter is given to the data format setting for trace 2, causing an error.

Lines 280 to 330

These lines define an error handler in the following way.

Lines 290 to 300: These lines retrieve the error number and error messages for the error from the error queue.

Lines 310 to 330 These lines display the message indicating the occurrence of the error, the error number, the error message, and the message showing that the program is suspended.

Line 350

Displays a closing message. Note that this message will not display unless this program is re-executed after setting a corrected parameter to the data format setting for trace 2.

Reading Data in ASCII Format

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to retrieve formatted data arrays in the ASCII transfer format.

This program holds the sweep on channel 1, then retrieves and displays the stimulus array for channel 1 and the formatted data array for trace 1.

See Retrieving Measurement Results for this programming.

Sample Program in Excel VBA

```
Sub read_asc_Click()
    '
    Dim defrm As Long
    Dim vi As Long
    Dim Result As String * 10000
    Dim Res As Variant
    Dim Res2 As Variant
    Dim Nop As Long
    Const TimeOutTime = 10000
    '
    ' Open the Analyzer
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Select Parameter 1
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" + vbLf, 0)
    Call viVPrintf(vi, ":INIT1:CONT OFF" + vbLf, 0)
    Call viVPrintf(vi, ":ABOR" + vbLf, 0)
    '
    ' Read out NOP Data in ASCII transfer format
    Call viVPrintf(vi, ":SENS1:SWE:POIN?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Nop = Val(Result)
    ReDim FMTData(Nop, 2)
```

```

ReDim Freq(Nop)
'
' Read out Measurement Data in ASCII transfer format
Call viVPrintf(vi, ":FORM:DATA ASC" + vbLf, 0)
Result = ""
Call viVPrintf(vi, ":CALC1:DATA:FDAT?" + vbLf, 0)
Call viVScanf(vi, "%t", Result)
Res = Split(Result, ",")
'
Range("A6:D1607").Clear 'Clear cells of Excel
'
' Write data in cells of Excel
j = 0
For i = 1 To Nop
    Cells(i + 5, 1) = i
    Cells(i + 5, 3) = Val(Res(j))
    Cells(i + 5, 4) = Val(Res(j + 1))
    j = j + 2
Next i
'
' Read out Measurement Frequency Data in ASCII transfer format
Result = ""
Call viVPrintf(vi, ":SENS1:FREQ:DATA?" + vbLf, 0)
Call viVScanf(vi, "%t", Result)
Res2 = Split(Result, ",")
'
' Write data in cells of Excel
For i = 1 To Nop
    Cells(i + 5, 2) = Val(Res2(i - 1))
Next i
'
' Close the Analyzer
Call viClose(vi)
Call viClose(defrm)

```

End Sub

Sample Program in HT Basic ([read_asc.htb](#))

10 REAL Fdata(1:1601,1:2),Freq(1:1601)

E5071C

```
20 DIM Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
80 OUTPUT @Agte507x;":INIT1:CONT OFF"
90 OUTPUT @Agte507x;":ABOR"
100 OUTPUT @Agte507x;":SENS1:SWE:POIN?"
110 ENTER @Agte507x;Nop
120 REDIM Fdata(1:Nop,1:2),Freq(1:Nop)
130 !
140 ! Reading out in ASCII transfer format
150 !
160 OUTPUT @Agte507x;":FORM:DATA ASC"
170 !
180 OUTPUT @Agte507x;":CALC1:DATA:FDAT?"
190 ENTER @Agte507x;Fdata(*)
200 OUTPUT @Agte507x;":SENS1:FREQ:DATA?"
210 ENTER @Agte507x;Freq(*)
220 !
230 ! Displaying
240 !
250 OUTPUT @Agte507x;":CALC1:FORM?"
260 ENTER @Agte507x;Fmt$
270 SELECT Fmt$
280 CASE "MLOG","PHAS","GDEL","MLIN","SWR","REAL","IMAG","UPH"
290 Img$="MD.4DE,2X,MD.6DE"
300 PRINT " Frequency Data"
310 FOR I=1 TO Nop
320 PRINT USING Img$;Freq(I),Fdata(I,1)
330 NEXT I
340 CASE ELSE
350 Img$="MD.4DE,2X,MD.6DE,2X,MD.6DE"
360 PRINT " Frequency Data1 Data2"
370 FOR I=1 TO Nop
380 PRINT USING Img$;Freq(I),Fdata(I,1),Fdata(I,2)
```

```
390 NEXT I
400 END SELECT
410 !
420 END
```

Reading Data in Binary Format

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to retrieve formatted data arrays in the Binary transfer format.

This program holds the sweep on channel 1, then retrieves and displays the stimulus array.

See Retrieving Measurement Results for this programming.

Sample Program in Excel VBA

```
Sub read_bin_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim Result As String * 10000
    Dim Res() As Double
    Dim Nop As Long
    Const TimeOutTime = 10000
    ' Open Analyzer
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" + vbLf, 0)
    Call viVPrintf(vi, ":INIT1:CONT OFF" + vbLf, 0)
    Call viVPrintf(vi, ":ABOR" + vbLf, 0)
    '
    ' Reading out Measurement Frequency Data in Binary transfer format
    Call viVPrintf(vi, ":FORM:DATA REAL" + vbLf, 0)
    Call viVPrintf(vi, ":CALC1:DATA:FDAT?" + vbLf, 0)
    Call Scpi_read_binary_double_array(vi, Res, Nop)
    '
    ' Write data in cells of Excel
    Range("A6:D1607").Clear
    For i = 0 To Nop - 1
        j = i Mod 2
```

```

        k = i \ 2
        Cells(k + 6, j + 3).Value = Res(i)
    Next i
    '
    ' Read out Measurement Frequency Data in Binary transfer format
    Call viVPrintf(vi, ":SENS1:FREQ:DATA?" + vbLf, 0)
    Call Scpi_read_binary_double_array(vi, Res, Nop)
    '
    ' Write data in cells of Excel
    For i = 0 To Nop - 1
        Cells(i + 6, 1) = i + 1
        Cells(i + 6, 2).Value = Res(i)
    Next i
    ' Close
    Call viClose(vi)
    Call viClose(defrm)
End Sub

'=====
' BinaryAry Read Subroutine
'=====

Sub Scpi_read_binary_double_array(vi As Long, data() As Double, Nop As Long)
    Dim dblArray(10000) As Double
    Dim paramsArray(3) As Long
    Dim err As Long
    Dim i As Long
    Dim lf_eoi As String * 1

    Nop = UBound(dblArray) - LBound(dblArray) + 1
    paramsArray(0) = VarPtr(Nop)
    paramsArray(1) = VarPtr(dblArray(0))
    err = viVScanf(vi, "%#Zb%1t", paramsArray(0))
    If err <> 0 Then MsgBox "Binary Error"

    ReDim data(Nop - 1)
    For i = 0 To Nop - 1
        data(i) = dblArray(i)
    Next

```

E5071C

End Sub

Sample Program in HT Basic (read_bin.htb)

```
10 REAL Fdata(1:1601,1:2),Freq(1:1601)
20 DIM Buff$(9),Img$(30)
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte507x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 !
80 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
90 OUTPUT @Agte507x;".INIT1:CONT OFF"
100 OUTPUT @Agte507x;".ABOR"
110 OUTPUT @Agte507x;".SENS1:SWE:POIN?"
120 ENTER @Agte507x;Nop
130 REDIM Fdata(1:Nop,1:2),Freq(1:Nop)
140 !
150 ! Reading out in binary transfer format
160 !
170 OUTPUT @Agte507x;".FORM:DATA REAL"
180 !
190 OUTPUT @Agte507x;".CALC1:DATA:FDAT?"
200 ENTER @Agte507x USING "#,8A";Buff$
210 ENTER @Binary;Fdata(*)
220 ENTER @Agte507x USING "#,1A";Buff$
230 OUTPUT @Agte507x;".SENS1:FREQ:DATA?"
240 ENTER @Agte507x USING "#,8A";Buff$
250 ENTER @Binary;Freq(*)
260 ENTER @Agte507x USING "#,1A";Buff$
270 !
280 ! Displaying
290 !
300 OUTPUT @Agte507x;".CALC1:FORM?"
310 ENTER @Agte507x;Fmt$
320 SELECT Fmt$
330 CASE "MLOG","PHAS","GDEL","MLIN","SWR","REAL","IMAG","UPH"
340 Img$="MD.4DE,2X,MD.6DE"
350 PRINT " Frequency Data"
```



```
360 FOR I=1 TO Nop
370 PRINT USING Img$;Freq(I),Fdata(I,1)
380 NEXT I
390 CASE ELSE
400 Img$="MD.4DE,2X,MD.6DE,2X,MD.6DE"
410 PRINT " Frequency Data1 Data2"
420 FOR I=1 TO Nop
430 PRINT USING Img$;Freq(I),Fdata(I,1),Fdata(I,2)
440 NEXT I
450 END SELECT
460 !
470 END
```

Writing Data in Ascii Format

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates to Write Formatted Data Arrays in Using the ASCII Transfer Format

See Entering Data into a Trace for this programming.

Sample Program in HT Basic ([write_a.htb](#))

```

10 REAL Freq,Fdata(1:1601,1:2)
20 DIM File$(300)
30 INTEGER Nop
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 CALL Inp_file_name(File$)
80 !
90 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
100 OUTPUT @Agte507x;".INIT1:CONT OFF"
110 OUTPUT @Agte507x;".ABOR"
120 !
130 OUTPUT @Agte507x;".SENS1:SWE:POIN?"
140 ENTER @Agte507x;Nop
150 REDIM Fdata(1:Nop,1:2)
160 !
170 ON ERROR GOTO File_error
180 ASSIGN @File TO File$
190 ENTER @File USING "K";Buff$
200 ENTER @File USING "K";Buff$
210 ENTER @File USING "K";Buff$
220 FOR I=1 TO Nop
230 ENTER @File USING "19D,2X,19D,2X,19D";Freq,Fdata(I,1),Fdata (I,2)
240 NEXT I
250 ASSIGN @File TO *
260 OFF ERROR
270 !
280 OUTPUT @Agte507x;".FORM:DATA ASC"

```

```

290 !
300 OUTPUT @Agte507x;":CALC1:DATA:FDAT ";Fdata(*)
310 !
320 GOTO Prog_end
330 !
340 File_error: OFF ERROR
350 PRINT "##### ERROR #####"
360 PRINT File$&" is NOT exist."
370 PRINT " or"
380 PRINT File$&" has UNSUITABLE data."
390 !
400 Prog_end: END
410 !=====
420 ! File Name Input Function
430 !=====
440 SUB Inp_file_name(Inp_name$)
450 DIM Inp_char$(9)
460 ON ERROR GOTO Inp_start
470 Inp_start: !
480 PRINT "Input File Name!"
490 INPUT "Name?",Inp_name$
500 PRINT "Input Name: "&Inp_name$
510 INPUT "OK? [Y/N]",Inp_char$
520 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
530 OFF ERROR
540 SUBEND

```

Description**Line 50**

Assigns a GPIB address to the I/O pass.

Line 70

Passes control to a subprogram named `Inp_file_name`, which lets the user input a file name, and then stores the returned file name into the `File$` variable. For more information on the `Inp_file_name` subprogram, refer to the description in *Using the Binary Transfer Format to write Formatted Data Arrays*.

Lines 90 to 110

These lines set channel 1's active trace to trace 1 and hold the sweep.

Lines 130 to 140

These lines retrieve the number of points in channel 1 and stores that number into the Nop variable.

Line 150

Resizes the Fdata array based on the value of the Nop variable (the number of points).

Line 170

This line points to the statement block to be executed if an error occurs in retrieving data from the file (for example, if no file matches File\$).

Lines 180 to 260

These lines retrieve the formatted data from the file identified by File\$, and store the data into the Fdata array.

Line 280

Sets the data transfer format to ASCII.

Line 300

Writes Fdata into the formatted data array for the active trace (trace 1) in channel 1.

Lines 340 to 380

This statement block is executed if an error occurs in retrieving data from the file.

Writing Data in Binary Format

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates to Write Formatted Data Arrays in Using the Binary Transfer Format

See Entering Data into a Trace for this programming.

Sample Program in HT Basic (write_b.htb)

```

10 REAL Freq,Fdata(1:1601,1:2)
20 DIM File$(300),Header$(10)
30 INTEGER Nop
40 !
50 ASSIGN @Agte507x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 CALL Inp_file_name(File$)
80 !
90 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
100 OUTPUT @Agte507x;".INIT1:CONT OFF"
110 OUTPUT @Agte507x;".ABOR"
120 !
130 OUTPUT @Agte507x;".SENS1:SWE:POIN?"
140 ENTER @Agte507x;Nop
150 REDIM Fdata(1:Nop,1:2)
160 !
170 ON ERROR GOTO File_error
180 ASSIGN @File TO File$
190 ENTER @File USING "K";Buff$
200 ENTER @File USING "K";Buff$
210 ENTER @File USING "K";Buff$
220 FOR I=1 TO Nop
230 ENTER @File USING "19D,2X,19D,2X,19D";Freq,Fdata(I,1),Fdata (I,2)
240 NEXT I
250 ASSIGN @File TO *
260 OFF ERROR
270 !
280 OUTPUT @Agte507x;".FORM:DATA REAL"

```

E5071C

```
290 Header$="#6"&IVAL$(8*2*Nop,10)
300 OUTPUT @Agte507x;":CALC1:DATA:FDAT ";Header$;
310 OUTPUT @Binary;Fdata(*),END
320 GOTO Prog_end
330 !
340 File_error: OFF ERROR
350 PRINT "##### ERROR #####"
360 PRINT File$&" is NOT exist."
370 PRINT " or"
380 PRINT File$&" has UNSUITABLE data."
390 !
400 Prog_end: END
410 !======
420 ! File Name Input Function
430 !======
440 SUB Inp_file_name(Inp_name$)
450 DIM Inp_char$(9)
460 ON ERROR GOTO Inp_start
470 Inp_start: !
480 PRINT "Input File Name!"
490 INPUT "Name?",Inp_name$
500 PRINT "Input Name: "&Inp_name$
510 INPUT "OK? [Y/N]",Inp_char$
520 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
530 OFF ERROR
540 SUBEND
```

Description

Lines 50 to 60

Assigns a GPIB address to the I/O pass.

Line 70

Passes control to a subprogram named Inp_file_name, which lets the user input a file name, and then stores the returned file name into the File\$ variable.

Lines 90 to 110

These lines set channel 1's active trace to trace 1 and hold the sweep.

Lines 130 to 140

These lines retrieve the number of points in channel 1 and stores that number into the Nop variable.

Line 150

Resizes the Fdata array based on the value of the Nop variable (the number of points).

Line 170

This line points to the statement block to be executed if an error occurs in retrieving data from the file (for example, if no file matches File\$).

Lines 180 to 260

These lines retrieve the formatted data from the file identified by File\$, and store the data into the Fdata array.

Line 280

Sets the data transfer format to binary.

Line 290

Creates the data header and stores it into the Header\$ variable.

Line 300

Sends the command that writes data into the formatted data array for the active trace (trace 1) in channel 1, following it with the data header (Header\$).

Line 310

Sends the data itself (Fdata), following it with a message terminator.

- Because binary data must be written without being formatted, the program uses an I/O path (@Binary) that is configured to support writing unformatted data.

Lines 340 to 380

This statement block is executed if an error occurs in retrieving data from the file.

The Inp_file_name subprogram in lines 440 to 540, which is used to enter a save filename, is described below.

Line 460

Allows the user to return to the entry start line and re-enter the data if an error (such as an invalid entry) occurs while entering the target file name.

Lines 480 to 490

These lines prompt the user to enter the target file name. The program does not continue till the user actually enters the file name.

Lines 500 to 510

E5071C

These lines display the entered file name and waits for a confirmation entry (y/n key).

Line 520

Returns to the entry start line if the key the user pressed in line 870 is not the y key.

Peak Search

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to search for peaks using the Marker Search feature and analysis commands.

This program works in two steps: first, it uses Marker Search to search for the maximum positive peak and displays the results; second, it uses analysis commands to search for all positive peaks and displays the results.

See Searching for Positions That Match Specified Criteria for this programming.

Sample Program in Excel VBA

```
Sub PeakSearch_Click()
    Dim defrm As Long
    Dim vi As Long
    Const TimeOutTime = 20000
    '

    Dim Buff As String, Img As String, Err_msg As String
    Dim Excursion As String, Freq As String * 20, Resp As Variant, PeakPoint As Variant
    Dim Poin As String * 5, Result As String * 1000, errmsg As String * 20

    Excursion = "0.5"
    ' Open Analyzer
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    Call viVPrintf(vi, "*CLS" & vbLf, 0)
    '

    ' Setup Analyzer
    Call viVPrintf(vi, ":SENS1:FREQ:CEN 947.5E6" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:FREQ:SPAN 200E6" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:DEF S11" & vbLf, 0)
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)
    '
End Sub
```

E5071C

```
Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:TYPE PEAK" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:PEXC " & Excursion & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:PPOL POS" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:EXEC" & vbLf, 0)
Call ErrorCheck(vi)
Call viVPrintf(vi, ":CALC1:MARK1:X?" & vbLf, 0)
Call viVScanf(vi, "%t", Freq)
,

Call viVPrintf(vi, ":CALC1:MARK1:Y?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
,

Resp = Split(Result, ",")
Cells(5, 5).Value = Val(Freq)
Cells(5, 6).Value = Val(Resp(0))
,

Call viVPrintf(vi, ":CALC1:FUNC:DOM OFF" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:TYPE APE" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:PEXC " & Excursion & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:PPOL NEG" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:EXEC" & vbLf, 0)
Call ErrorCheck(vi)
,

Call viVPrintf(vi, ":CALC1:FUNC:POIN?" & vbLf, 0)
Call viVScanf(vi, "%t", Poin)
Call viVPrintf(vi, ":CALC1:FUNC:DATA?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
PeakPoint = Split(Result, ",")
,

j = 1
  For i = 1 To Val(Poin) / 2
    Cells(6 + i, 5).Value = Val(PeakPoint(j))
    Cells(6 + i, 6).Value = Val(PeakPoint(j + 1))
    j = j + 2
  Next i
,

Call viClose(vi)
```

```
Call viClose(defrm)
```

```
,
```

```
End Sub
```

```
Sub ErrorCheck(vi)
```

```
Dim err As String * 50, ErrNo As Variant, Response As VbMsgBoxResult
```

```
Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
```

```
ErrNo = Split(err, ",")
```

```
If Val(ErrNo(0)) <> 0 Then
```

```
Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
```

```
End If
```

```
End Sub
```

Sample Program in HT Basic ([search.htb](#))

```
10 DIM Buff$(9),Img$(50),Err_msg$(100)
```

```
20 REAL Excursion,Freq,Resp,Result(1:100,1:2)
```

```
30 INTEGER Poin,Err_no
```

```
40 !
```

```
50 ASSIGN @Agte507x TO 717
```

```
60 Excursion=.5
```

```
70 !
```

```
80 OUTPUT @Agte507x;"*ESE 60"
```

```
90 OUTPUT @Agte507x;"*SRE 32"
```

```
100 OUTPUT @Agte507x;"*CLS"
```

```
110 OUTPUT @Agte507x;"*OPC?"
```

```
120 ENTER @Agte507x;Buff$
```

```
130 ON INTR 7 GOTO Err
```

```
140 ENABLE INTR 7;2
```

```
150 !
```

```
160 PRINT "Maximum Peak Search using Marker 1"
```

```
170 !
```

```
180 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
```

```
190 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:TYPE PEAK"
```

```
200 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:PEXC ";Excursion
```

```
210 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:PPOL POS"
```

```
220 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:EXEC"
```

```
230 OUTPUT @Agte507x;":CALC1:MARK1:X?"
```

```
240 ENTER @Agte507x;Freq
```

```
250 OUTPUT @Agte507x;":CALC1:MARK1:Y?"
```

E5071C

```
260 ENTER @Agte507x;Resp
270 Img$="8A,MD.4DE,2X,MD.6DE"
280 PRINT " Frequency Response"
290 PRINT USING Img$;"Peak: ",Freq,Resp
300 !
310 PRINT "All Peaks Search using Command"
320 !
330 OUTPUT @Agte507x;".CALC1:FUNC:DOM OFF"
340 OUTPUT @Agte507x;".CALC1:FUNC:TYPE APE"
350 OUTPUT @Agte507x;".CALC1:FUNC:PEXC ";Excursion
360 OUTPUT @Agte507x;".CALC1:FUNC:PPOL POS"
370 OUTPUT @Agte507x;".CALC1:FUNC:EXEC"
380 OUTPUT @Agte507x;".CALC1:FUNC:POIN?"
390 ENTER @Agte507x;Poin
400 REDIM Result(1:Poin,1:2)
410 OUTPUT @Agte507x;".CALC1:FUNC:DATA?"
420 ENTER @Agte507x;Result(*)
430 Img$="4A,2D,2A,MD.4DE,2X,MD.6DE"
440 PRINT " Frequency Response"
450 FOR I=1 TO Poin
460 PRINT USING Img$;"Peak",I," ",Result(I,2),Result(I,1)
470 NEXT I
480 GOTO No_err
490 Err: OFF INTR 7
500 OUTPUT @Agte507x;";SYST:ERR?"
510 ENTER @Agte507x;Err_no,Err_msg$
520 PRINT "Error occurred!!"
530 PRINT " No: ";Err_no,"Description: "&Err_msg$
540 No_err: OFF INTR 7
550 END
```

Limit Test

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to perform limit tests.

The sample program creates a limit table as shown in the following two tables, turns on the Limit Test feature, performs one cycle of measurement, and then displays the test results.

No.	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	847.5 MHz	905.0 MHz	-55.0 dBm	-55.0 dBm
2	MIN	935.0 MHz	960.0 MHz	-3.5 dBm	-3.5 dBm
3	MAX	935.0 MHz	960.0 MHz	0 dBm	0 dBm
4	MAX	980.0 MHz	1047.5 MHz	-25.0 dBm	-25.0 dBm

See Limit Test for this programming.

Sample Program in HT Basic ([lim_test.htb](#))

```

10 DIM Param1$(9),Param2$(9),Fmt1$(9),Fmt2$(9),Buff$(9)
20 REAL Cent,Span,Lim1(1:4,1:5),Lim2(1:3,1:5),Fail_data(1:1601)
30 INTEGER Num_of_seg1,Num_of_seg2,Segment,Column,Fail_point
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 Cent=9.475E+8
80 Span=2.00E+8
90 Param1$="S21"
100 Param2$="S11"
110 Fmt1$="MLOG"
120 Fmt2$="MLOG"
130 !
140 ! == Trace 1 Limit Line ==
150 Num_of_seg1=4 ! Number of segments: 4

```

E5071C

160 ! -- Segment 1 --
170 Lim1(1,1)=1 ! Type : Maximum
180 Lim1(1,2)=8.475E+8 ! Frequency Start: 847.5 MHz
190 Lim1(1,3)=9.050E+8 ! Stop : 905.0 MHz
200 Lim1(1,4)=-55 ! Response Start: -55 dBm
210 Lim1(1,5)=-55 ! Stop : -55 dBm
220 ! -- Segment 2 --
230 Lim1(2,1)=2 ! Type : Minimum
240 Lim1(2,2)=9.350E+8 ! Frequency Start: 935.0 MHz
250 Lim1(2,3)=9.600E+8 ! Stop : 960.0 MHz
260 Lim1(2,4)=-3.5 ! Response Start: -3.5 dBm
270 Lim1(2,5)=-3.5 ! Stop : -3.5 dBm
280 ! -- Segment 3 --
290 Lim1(3,1)=1 ! Type : Maximum
300 Lim1(3,2)=9.350E+8 ! Frequency Start: 935.0 MHz
310 Lim1(3,3)=9.600E+8 ! Stop : 960.0 MHz
320 Lim1(3,4)=0 ! Response Start: 0 dBm
330 Lim1(3,5)=0 ! Stop : 0 dBm
340 ! -- Segment 4 --
350 Lim1(4,1)=1 ! Type : Maximum
360 Lim1(4,2)=9.800E+8 ! Frequency Start: 980.0 MHz
370 Lim1(4,3)=1.0475E+9 ! Stop : 1047.5 MHz
380 Lim1(4,4)=-25 ! Response Start: -25 dBm
390 Lim1(4,5)=-25 ! Stop : -25 dBm
400 ! == Trace 2 Limit Line ==
410 Num_of_seg2=3 ! Number of segments: 3
420 ! -- Segment 1 --
430 Lim2(1,1)=1 ! Type : Maximum
440 Lim2(1,2)=8.475E+8 ! Frequency Start: 847.5 MHz
450 Lim2(1,3)=9.250E+8 ! Stop : 925.0 MHz
460 Lim2(1,4)=0 ! Response Start: 0 dBm
470 Lim2(1,5)=0 ! Stop : 0 dBm
480 ! -- Segment 2 --
490 Lim2(2,1)=1 ! Type : Maximum
500 Lim2(2,2)=9.350E+8 ! Frequency Start: 935.0 MHz
510 Lim2(2,3)=9.600E+8 ! Stop : 960.0 MHz
520 Lim2(2,4)=-9.5 ! Response Start: -9.5 dBm

```

530 Lim2(2,5)=-9.5 ! Stop : -9.5 dBm
540 ! -- Segment 3 --
550 Lim2(3,1)=1 ! Type : Maximum
560 Lim2(3,2)=9.700E+8 ! Frequency Start: 970.0 MHz
570 Lim2(3,3)=1.0475E+9 ! Stop : 1047.5 MHz
580 Lim2(3,4)=0 ! Response Start: 0 dBm
590 Lim2(3,5)=0 ! Stop : 0 dBm
600 !
610 OUTPUT @Agte507x;":SENS1:FREQ:CENT ";Cent
620 OUTPUT @Agte507x;":SENS1:FREQ:SPAN ";Span
630 OUTPUT @Agte507x;":CALC1:PAR1:COUN 2"
640 OUTPUT @Agte507x;":DISP:WIND1:SPL D1_2"
650 OUTPUT @Agte507x;":TRIG:SOUR BUS"
660 OUTPUT @Agte507x;":INIT1:CONT ON"
670 !
680 ! Trace 1
690 !
700 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
710 !
720 OUTPUT @Agte507x;":CALC1:PAR1:DEF "&Param1$
730 OUTPUT @Agte507x;":CALC1:FORM "&Fmt1$
740 !
750 OUTPUT @Agte507x;":CALC1:LIM:DATA ";Num_of_seg1;
760 FOR Segment=1 TO Num_of_seg1
770 FOR Column=1 TO 5
780 OUTPUT @Agte507x;";";Lim1(Segment,Column);
790 NEXT Column
800 NEXT Segment
810 OUTPUT @Agte507x;""
820 OUTPUT @Agte507x;":CALC1:LIM:DISP ON"
830 OUTPUT @Agte507x;":CALC1:LIM ON"
840 !
850 ! Trace 2
860 !
870 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
880 !
890 OUTPUT @Agte507x;":CALC1:PAR2:DEF "&Param2$

```

E5071C

```
900 OUTPUT @Agte507x;":CALC1:FORM "&Fmt2$
910 !
920 OUTPUT @Agte507x;":CALC1:LIM:DATA ";Num_of_seg2;
930 FOR Segment=1 TO Num_of_seg2
940 FOR Column=1 TO 5
950 OUTPUT @Agte507x;";";Lim2(Segment,Column);
960 NEXT Column
970 NEXT Segment
980 OUTPUT @Agte507x;""
990 OUTPUT @Agte507x;":CALC1:LIM:DISP ON"
1000 OUTPUT @Agte507x;":CALC1:LIM ON"
1010 !
1020 ! Setting status registers
1030 !
1040 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1:ENAB 6"
1050 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1:PTR 6"
1060 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1:NTR 0"
1070 OUTPUT @Agte507x;":STAT:QUES:LIM:PTR 2"
1080 OUTPUT @Agte507x;":STAT:QUES:LIM:NTR 0"
1090 OUTPUT @Agte507x;"*CLS"
1100 !
1110 OUTPUT @Agte507x;":TRIG:SING"
1120 OUTPUT @Agte507x;"*OPC?"
1130 ENTER @Agte507x;Buff$
1140 !
1150 ! Checking test results
1160 !
1170 OUTPUT @Agte507x;":STAT:QUES:LIM?"
1180 ENTER @Agte507x;Reg_val
1190 Ch1_judge=BIT(Reg_val,1)
1200 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1?"
1210 ENTER @Agte507x;Reg_val
1220 Tr1_judge=BIT(Reg_val,1)
1230 Tr2_judge=BIT(Reg_val,2)
1240 !
1250 ! Displaying test results
1260 !
```



```
1270 IF Ch1_judge=0 THEN
1280 PRINT "## PASS! ##"
1290 ELSE
1300 PRINT "## FAIL! ##"
1310 IF Tr1_judge=0 THEN
1320 PRINT " Trace1(S21): PASS"
1330 ELSE
1340 PRINT " Trace1(S21): FAIL"
1350 !
1360 ! Reading and displaying frequency at failed points
1370 !
1380 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
1390 OUTPUT @Agte507x;":CALC1:LIM:REP:POIN?"
1400 ENTER @Agte507x;Fail_point
1410 REDIM Fail_data(1:Fail_point)
1420 OUTPUT @Agte507x;":CALC1:LIM:REP?"
1430 ENTER @Agte507x;Fail_data(*)
1440 PRINT " Frequency:"
1450 FOR I=1 TO Fail_point
1460 PRINT USING "3X,MD.4DE";Fail_data(I)
1470 NEXT I
1480 END IF
1490 IF Tr2_judge=0 THEN
1500 PRINT " Trace2(S11): PASS"
1510 ELSE
1520 PRINT " Trace2(S11): FAIL"
1530 !
1540 ! Reading and displaying frequency at failed points
1550 !
1560 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
1570 OUTPUT @Agte507x;":CALC1:LIM:REP:POIN?"
1580 ENTER @Agte507x;Fail_point
1590 REDIM Fail_data(1:Fail_point)
1600 OUTPUT @Agte507x;":CALC1:LIM:REP?"
1610 ENTER @Agte507x;Fail_data(*)
1620 PRINT " Frequency:"
1630 FOR I=1 TO Fail_point
```

E5071C

```
1640 PRINT USING "3X,MD.4DE";Fail_data(I)
1650 NEXT I
1660 END IF
1670 END IF
1680 END
```

Description

Line 50

Assigns a GPIB address to the I/O pass.

Lines 70 to 120

These lines store the sweep center value, sweep span value, trace 1 measurement parameter, trace 2 measurement parameter, trace 1 data format, and trace 2 data format into the variables Cent, Span, Param1\$, Param2\$, Fmt1\$, and Fmt2\$, respectively.

Line 150

Stores the number of segments in trace 1 limit table into the Num_of_seg1 variable.

Lines 160 to 390

These lines store the settings in trace 1 limit table into the Lim1(*) variable.

Line 410

Stores the number of segments in trace 2 limit table into the Num_of_seg2 variable.

Lines 420 to 590

These lines store the settings in trace 2 limit table into the Lim2(*) variable.

Lines 610 to 620

These lines configure the sweep range for channel 1's sweep range using the center and span values contained in the Cent and Span values.

Lines 630 to 660

These lines configure channel 1 so that it contains 2 traces, displays graphs in two windows tiled horizontally (i.e., with the screen split into upper and lower halves), uses a bus trigger source, and works in continuous activation mode.

Line 700

Sets channel 1's active trace to trace 1.

Lines 720 to 730

These lines store trace 1's measurement parameter and data format into the variables Param1\$ and Fmt1\$, respectively.

Lines 750 to 810

These lines set up the limit table for trace 1.

Line 750: Sends the command that sets up a limit table along with the Num_of_seg1 variable that contains the number of segments.

Lines 770 to 790: Sends five data items (type, start point stimulus value, end point stimulus value, start point response value, and end point response value) for each segment.

Lines 820 to 830

These lines turn on the display of limit lines and the Limit Test feature for trace 1.

Line 870

Sets channel 1's active trace to trace 2.

Lines 890 to 900

These lines set trace 2's measurement parameter and data format to Param2\$ and Fmt2\$, respectively.

Lines 920 to 980

These lines set up the limit table for trace 2.

Lines 990 to 1000

These lines turn on the display of limit lines and the Limit Test feature for trace 2.

Lines 1040 to 1060

These lines set, under the questionable limit channel 1 status register, the enable register and positive transition filter to 6 (0000000000000110 in binary notation) while setting the negative transition filter to 0 so that the questionable limit status condition register's bit 1 is set to 1 when the test result that combines the results for trace 1 and trace 2 is "fail."

The sample program provides an example of explicitly configuring the register bits so that they reflect the test results that only cover trace 1 and trace 2. However, because the results for traces 3 to 9 will never be "fail" as long as the Limit Test feature is disabled for those traces, the register bits would reflect the test result that is limited to traces 1 and 2, even if the default setting is not changed.

Lines 1070 to 1080

These lines set transition filters so that the questionable limit status event register's bit 1 is set to 1 when the questionable limit status condition register's bit 1 changes from 0 to 1.

Line 1090

Clears the questionable limit status event register and questionable limit channel 1 status event register.

Lines 1110 to 1130

These lines trigger the instrument and wait until the sweep cycle is completed.

Lines 1170 to 1190

These lines retrieve the value of the questionable limit status event register and store the setting of bit 1 of the value into Ch1_judge.

Lines 1200 to 1230

These lines retrieve the value of the questionable limit channel 1 status event register and store the settings of bit 1 and bit 2 of the value into Tr1_judge and Tr2_judge, respectively.

Line 1280

Displays a message indicating that the DUT has passed the limit test if the test result for channel 1 is "Pass" (i.e., if Ch1_judge returns 0).

Lines 1300 to 1660

These lines are executed if the test result for channel 1 is "Fail" (i.e., if Ch1_judge returns 1).

Line 1300: Notifies the user that the limit test result is "Fail".

Line 1320: Displays a message indicating that trace 1 has passed the limit test if the test result for trace 1 is "Pass" (i.e., if Tr1_judge returns 0).

Lines 1340 to 1470: These lines are executed if the test result for trace 1 is "Fail" (i.e., if Tr1_judge returns 1). The lines notify the user that the test result for trace 1 is "Fail" and then retrieve and display the frequencies at the failed measurement points on trace 1.

Line 1340: Notifies the user that the limit test result for trace 1 is "Pass."

Line 1380: Sets channel 1's active trace to trace 2.

Lines 1390 to 1410: These lines retrieve the number of failed measurement points on trace 1 and, based on that number, resize the array that contains retrieved frequencies.

Lines 1420 to 1470: These lines retrieve and display the frequencies at the failed measurement points on trace 1.

Line 1500: Displays a message indicating that trace 2 has passed the limit test if the test result for trace 2 is "Pass" (i.e., if Tr2_judge returns 0).

Lines 1520 to 1650: If the test result for trace 2 is "Fail" (i.e., if Tr2_judge returns 1), these lines notify the user that trace 2 has failed to pass the limit test and then retrieve and display the frequencies at the failed measurement points on trace 2.

Bandwidth Search

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to perform Bandwidth Search.

The sample program moves the marker to the maximum value position and then retrieves and displays the results of Bandwidth Search.

See Bandwidth Search for this programming.

Sample Program in Excel VBA

```
Sub Bandwid_click()
    Dim vi As Long
    Dim defrm As Long
    Dim Threshold As Long
    Dim Result As String * 1000
    Dim Bdata As Variant
    Const TimeOutTime = 10000
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    Threshold = -3
    Call viVPrintf(vi, ":SENS1:FREQ:CENT 947.5E6" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:FREQ:SPAN 200E6" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:DEF S21" & vbLf, 0)
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:FUNC:TYPE MAX" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:FUNC:EXEC" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:BWID:THR " & CStr(Threshold) & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:BWID:DATA?" & vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Bdata = Split(Result, ",")
    Call ErrorCheck(vi)
    Cells(5, 2).Value = Val(Bdata(0))
    Cells(6, 2).Value = Val(Bdata(1))
End Sub
```

```
Cells(7, 2).Value = Val(Bdata(2))
Cells(8, 2).Value = Val(Trim(Bdata(3)))
Call viClose(vi)
Call viClose(defrm)
```

End Sub

Sub ErrorCheck(vi)

```
Dim err As String * 50, ErrNo As Variant, Response
Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
ErrNo = Split(err, ",")
If Val(ErrNo(0)) <> 0 Then
    Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
```

End If

End Sub

Sample Program in HT Basic (bandwid.htb)

```
10 DIM Buff$(9),Err_msg$(100)
20 REAL Threshold,Bwid,Cent,Q,Loss
30 INTEGER Err_no
40 !
50 ASSIGN @Agte507x TO 717
60 Threshold=-3
70 !
80 OUTPUT @Agte507x;"*ESE 60"
90 OUTPUT @Agte507x;"*SRE 32"
100 OUTPUT @Agte507x;"*CLS"
110 OUTPUT @Agte507x;"*OPC?"
120 ENTER @Agte507x;Buff$
130 ON INTR 7 GOTO Err
140 ENABLE INTR 7;2
150 !
160 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
170 OUTPUT @Agte507x;".CALC1:MARK1:FUNC:TYPE MAX"
180 OUTPUT @Agte507x;".CALC1:MARK1:FUNC:EXEC"
190 OUTPUT @Agte507x;".CALC1:MARK1:BWID:THR ";Threshold
200 OUTPUT @Agte507x;".CALC1:MARK1:BWID:DATA?"
210 WAIT .5
220 ENTER @Agte507x;Bwid,Cent,Q,Loss
```

E5071C

```
230 !
240 PRINT "## Bandwidth Search ##"
250 PRINT "Bandwidth : ",Bwid
260 PRINT "Center Frequency: ",Cent
270 PRINT "Q : ",Q
280 PRINT "Loss : ",Loss
290 !
300 GOTO No_err
310 Err: OFF INTR 7
320 OUTPUT @Agte507x;";:SYST:ERR?"
330 ENTER @Agte507x;Err_no,Err_msg$
340 PRINT "Error occurred!!"
350 PRINT " No: ";Err_no,"Description: "&Err_msg$
360 No_err: OFF INTR 7
370 END
```


Saving Files

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to save a file. This program saves selected content on a file with a specified name.

See Saving and Recalling File for this programming.

Sample Program in Excel VBA

```
Sub File_Save()
    ' Declare two string variables for file name and file type
    Dim File_Name As String
    Dim File_Type As String
    Dim defrm As Long
    Dim vi As Long
    Const TimeOutTime = 10000
    ' Check whether file name textbox is empty or not
    If TextBox1.Text <> "" Then
        File_Name = Trim(TextBox1.Text)
        File_Type = Trim(frmFileSave.ComboBox1.Value)
    ' Open connection to the E5071C
        Call viOpenDefaultRM(defrm)
        Call viOpen(defrm, "GPIB0::18::INSTR", 0, 0, vi)
        Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
        Select Case File_Type
            Case "1: State (State only)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP STAT" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """" & File_Name & ".sta"""" + vbLf, 0)
            Case "2: State (State & Cal)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP CST" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """" & File_Name & ".sta"""" + vbLf, 0)
            Case "3: State (State & Trace)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP DST" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """" & File_Name & ".sta"""" + vbLf, 0)
            Case "4: State (State & Cal & Trace)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP CDST" + vbLf, 0)
        End Select
    End If
End Sub
```

E5071C

```
    Call viVPrintf(vi, ".MMEM:STOR "" & File_Name & ".sta"" + vbLf, 0)
Case "5: State (Trace Data (CSV))"
    Call viVPrintf(vi, ".MMEM:STOR:FDAT "" & File_Name & ".csv"" + vbLf, 0)
Case "6: State (Screen)"
    Call viVPrintf(vi, ".MMEM:STOR:IMAG "" & File_Name & ".bmp"" + vbLf, 0)
Case Else
    msgbox "Error in code"
End Select
Call viClose(defrm)
Else
    msgbox "Please enter a filename"
End If
End Sub
```

Sample Program in HT Basic (file_sav.htb)

```
10 DIM File$(300),Inp_char$(30)
20 INTEGER Content
30 CLEAR SCREEN
40 ASSIGN @Agte507x TO 717
50 !
60 ON ERROR GOTO Content_select
70 Content_select: !
80 PRINT "## Save Content Selection ##"
90 PRINT "Select Content"
100 PRINT " 1: State (State only)"
110 PRINT " 2: State (State & Cal)"
120 PRINT " 3: State (State & Trace)"
130 PRINT " 4: State (State & Cal & Trace)"
140 PRINT " 5: Trace Data (CSV)"
150 PRINT " 6: Screen"
160 PRINT ""
170 PRINT "Input 1 to 6"
180 INPUT "Number?",Inp_char$
190 Content=IVAL(Inp_char$,10)
200 IF Content<1 OR Content>6 THEN Content_select
210 OFF ERROR
220 !
230 CALL Inp_file_name(File$)
```

```

240 !
250 SELECT Content
260 CASE 1
270 OUTPUT @Agte507x;":MMEM:STOR:STYP STAT"
280 OUTPUT @Agte507x;":MMEM:STOR """"&File$&".sta""""
290 CASE 2
300 OUTPUT @Agte507x;":MMEM:STOR:STYP CST"
310 OUTPUT @Agte507x;":MMEM:STOR """"&File$&".sta""""
320 CASE 3
330 OUTPUT @Agte507x;":MMEM:STOR:STYP DST"
340 OUTPUT @Agte507x;":MMEM:STOR """"&File$&".sta""""
350 CASE 4
360 OUTPUT @Agte507x;":MMEM:STOR:STYP CDST"
370 OUTPUT @Agte507x;":MMEM:STOR """"&File$&".sta""""
380 CASE 5
390 OUTPUT @Agte507x;":MMEM:STOR:FDAT """"&File$&".csv""""
400 CASE 6
410 OUTPUT @Agte507x;":MMEM:STOR:IMAG """"&File$&".bmp""""
420 END SELECT
430 !
440 END
450 !=====
460 ! File Name Input Function
470 !=====
480 SUB Inp_file_name(Inp_name$)
490 DIM Inp_char$(9)
500 ON ERROR GOTO Inp_start
510 Inp_start: !
520 PRINT "## File Name Input ##"
530 PRINT "Input Save File Name (without Extension)"
540 INPUT "Name?",Inp_name$
550 PRINT "Input Name: "&Inp_name$
560 INPUT "OK? [Y/N]",Inp_char$
570 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
580 OFF ERROR
590 SUBEND

```

Transferring Files

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates to transfer files between the external controller and the E5071C.

This program reads out data from a specified file on the external controller (or the E5071C), then write them to a specified file on the E5071C (or the external controller).

See Managing Files for this programming.

Sample Program in HT Basic (file_xfr.htb)

```

10 DIM Src_file$(50),Dst_file$(50),Src_size_char$(50),Inp_char$(30)
20 INTEGER Direction
30 ASSIGN @Agte507x TO 717
40 !
50 CLEAR SCREEN
60 ON ERROR GOTO Direct_select
70 Direct_select: !
80 PRINT "#### File Transfer ####"
90 PRINT " 1: E507x -> Controller"
100 PRINT " 2: Controller -> E507x"
110 PRINT ""
120 PRINT "Input 1 or 2"
130 INPUT "Number?",Inp_char$
140 Direction=IVAL(Inp_char$,10)
150 IF Direction<1 OR Direction>2 THEN Direct_select
160 OFF ERROR
170 !
180 PRINT ""
190 PRINT " Input source file name. ";
200 INPUT "Name?",Src_file$
210 PRINT ": "&Src_file$
220 !
230 IF Direction=2 THEN
240 PRINT " Input source file size. ";
250 INPUT "Size[Byte]?",Src_size_char$

```

```

260 PRINT ": "&Src_size_char$&"[Byte]"
270 END IF
280 !
290 PRINT " Input destination file name. ";
300 INPUT "Name?",Dst_file$
310 PRINT ": "&Dst_file$
320 PRINT ""
330 !
340 IF Direction=1 THEN
350 Copy_to_contr(@Agte507x,Src_file$,Dst_file$)
360 ELSE
370 Copy_to_e507x(@Agte507x,Src_file$,Src_size_char$,Dst_file$)
380 END IF
390 !
400 END
410 !=====
420 ! File Transfer Function (E507x -> Controller)
430 !=====
440 SUB Copy_to_contr(@Agte507x,Src_file$,Dst_file$)
450 DIM Img$(32),Src_size_char$(10),Buff$(9),Err_msg$(100)
460 INTEGER Max_bsize,Block_size,Err_no
470 REAL Src_size
480 !
490 ON ERROR GOTO Skip_purge
500 PURGE Dst_file$
510 Skip_purge: OFF ERROR
520 CREATE Dst_file$,1
530 ASSIGN @Dst_file TO Dst_file$
540 Max_bsize=24576 ! 24KByte
550 !
560 OUTPUT @Agte507x;"*ESE 60"
570 OUTPUT @Agte507x;"*SRE 32"
580 OUTPUT @Agte507x;"*CLS"
590 OUTPUT @Agte507x;"*OPC?"
600 ENTER @Agte507x;Buff$
610 !
620 ON INTR 7 GOTO Err

```

E5071C

```
630 ENABLE INTR 7;2
640 PRINT "Now Copying: "&Src_file$&"(@E507x) -> "&Dst_file$&"(@Contro ller)"
650 OUTPUT @Agte507x;":MMEM:TRAN? ""&Src_file$&"""
660 WAIT .1
670 ENTER @Agte507x USING "#,A";Buff$
680 ENTER @Agte507x USING "#,A";Digit$
690 Img$="#,"&Digit$&"A"
700 ENTER @Agte507x USING Img$;Src_size_char$
710 !
720 Src_size=VAL(Src_size_char$)
730 WHILE Src_size>0
740 IF Src_size>Max_bsize THEN
750 Block_size=Max_bsize
760 ELSE
770 Block_size=Src_size
780 END IF
790 !
800 ALLOCATE Dat$[Block_size]
810 Img$="#,"&VAL$(Block_size)&"A"
820 ENTER @Agte507x USING Img$;Dat$
830 OUTPUT @Dst_file USING Img$;Dat$
840 DEALLOCATE Dat$
850 !
860 Src_size=Src_size-Block_size
870 END WHILE
880 !
890 PRINT "Done"
900 ENTER @Agte507x USING "#,A";Buff$
910 ASSIGN @Dst_file TO *
920 !
930 GOTO Skip_error
940 Err: OFF INTR 7
950 OUTPUT @Agte507x;";:SYST:ERR?"
960 ENTER @Agte507x;Err_no,Err_msg$
970 PRINT "Error occurred!!"
980 PRINT " No:";Err_no,"Description: "&Err_msg$
990 Skip_error: OFF INTR 7
```

```

1000 SUBEND
1010 !======
1020 ! File Transfer Function (Controller -> E507x)
1030 !======
1040 SUB Copy_to_e507x(@Agte507x,Src_file$,Src_size_char$,Dst_file$)
1050 DIM Img$[32],Header$[10],Buff$[9],Err_msg$[100]
1060 INTEGER Max_bsize,Block_size,Err_no
1070 REAL Src_size
1080 !
1090 ON ERROR GOTO File_error
1100 ASSIGN @Src_file TO Src_file$
1110 OFF ERROR
1120 Max_bsize=24576 ! 24KByte
1130 !
1140 OUTPUT @Agte507x;"*CLS"
1150 OUTPUT @Agte507x;"*OPC?"
1160 ENTER @Agte507x;Buff$
1170 !
1180 PRINT "Now Copying: "&Src_file$&"(@Controller) -> "&Dst_file$&"(@ E507x)"
1190 Header$="#"&VAL$(LEN(Src_size_char$))&Src_size_char$
1200 OUTPUT @Agte507x;".MMEM:TRAN """"&Dst_file$&"""" "&Header$;
1210 !
1220 Src_size=VAL(Src_size_char$)
1230 WHILE Src_size>0
1240 IF Src_size>Max_bsize THEN
1250 Block_size=Max_bsize
1260 ELSE
1270 Block_size=Src_size
1280 END IF
1290 !
1300 ALLOCATE Dat$[Block_size]
1310 Img$="#,"&VAL$(Block_size)&"A"
1320 ENTER @Src_file USING Img$;Dat$
1330 OUTPUT @Agte507x USING Img$;Dat$
1340 DEALLOCATE Dat$
1350 !
1360 Src_size=Src_size-Block_size

```

E5071C

```
1370 END WHILE
1380 !
1390 OUTPUT @Agte507x;"",END
1400 ASSIGN @Src_file TO *
1410 !
1420 OUTPUT @Agte507x;"::SYST:ERR?"
1430 ENTER @Agte507x;Err_no,Err_msg$
1440 IF Err_no=0 THEN
1450 PRINT "Done"
1460 ELSE
1470 PRINT "Error occurred!!"
1480 PRINT " No: ";Err_no,"Description: "&Err_msg$
1490 END IF
1500 GOTO Skip_error
1510 File_error:OFF ERROR
1520 PRINT "File name NOT found!"
1530 Skip_error:!
1540 SUBEND
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 130

These line allow the user to return to the entry start line and re-enter the data if an error (such as an invalid entry) occurs while entering the number that indicates the transfer direction. Then, these line display the list of transfer directions and prompt the user to input a selected number.

Lines 80 to 130

These lines display the list of transfer directions, and prompt the user to choose one of the items by typing in the appropriate number.

Lines 140 to 150

Converts the entered value into an integer and stores it into the Direction variable. Returns to the entry start line if an invalid value is contained in Direction.

Lines 180 to 210

These lines obtain the name of the source file for copying from the user input, store it into the Src_file\$ variable, and display the value of Src_file\$.

Lines 180 to 210

These lines obtain the name of the source file for copying from the user input, store it into the Src_file\$ variable, and display the value of Src_file\$.

Lines 230 to 270

If Direction is equal to 2 (from the external controller to the E5071C), these lines obtain the size of the source file for copying, store it into the Src_size_char\$, and display the value of Src_size_char\$.

Lines 290 to 320

These lines obtain the name of the destination file for copying from the user input, store it into the Dst_file\$ variable, and display the value of Dst_file\$.

Line 350

If Direction is equal to 1 (from the E5071C to the external controller), these lines use the subprogram Copy_to_contr to transfer (copy) a file with the name Src_file\$ on the E5071C to a file with the name Dst_file\$ on the external controller.

Line 370

If Direction is equal to 2, these lines use the subprogram Copy_to_e507x to transfer (copy) a file with the name Src_file\$ on the external controller to a file with the name Dst_file\$ on the E5071C.

Copy_to_contr, a subprogram for transferring files from the E5071C to the external controller that appears in lines 440 to 1000, is described below.

Lines 490 to 520

If any file with the name File\$ already exists, these lines delete the file and newly create a file with the name File\$.

Line 530

Assigns a destination file for copying to the I/O pass.

Line 540

This line stores a maximum number of transferred data (in bytes) per one transfer, that is 24 KByte to meet the size limitation of string arrays in the HTBasic, into Max_bsize variable.

Lines 560 to 600

These lines configure the system to generate an SRQ when it cannot find a source file for copying due to an error.

Lines 620 to 630

These lines set the branch target for an SRQ interrupt to enable SRQ interrupts.

E5071C

Lines 640 to 650

These lines display a message showing that the transfer has started, and execute commands for reading data from a file on the E5071C.

Lines 670 to 680

These lines read the header symbol (#) in a block data, read number of digits (characters) indicating the size of data in bytes, then store it into Digit\$ variable.

Line 690

This line creates a format for reading characters in Digit\$.

Line 700

This line reads the data size in byte and stores it into Src_size_char\$ variable.

Line 720

This line converts Src_size_char\$ to a real number and stores it into Src_size variable.

Lines 730 to 870

These lines repeat the procedures below until Src_size reaches 0.

Lines 740 to 780: If Src_size is greater than Max_bsize, these lines assign the value of the Max_bsize to Block_size variable (transferred data in bytes). If Src_size is equal or less than Max_bsize, assign the value of Src_size to Block_size.

Line 800 This line defines Dat\$ string variable with the size as large as Block_size and reserves memory area.

Line 810 This line creates a format for reading characters as many as Block_size characters.

Line 820 This line reads data from the file on the E5071C, then stores them into Dat\$.

Line 830 This line writes the contents of Dat\$ to the file on the external controller.

Lines 840 to 860 These lines free the memory area for Dat\$ and subtract Block_size from Src_size.

Lines 890 to 900

These lines display a message showing the completion of transfer, then read a message terminator at the end of the data.

Lines 940 to 980

These lines define an error handler that retrieves and displays the number and message of an error that has occurred.

Copy_to_e507x, a subprogram for transferring files from the external controller to the E5071C that appears in lines 1040 to 1540, is described below.

Lines 1090 to 1110

Assigns a destination file for copying to the I/O pass.

Line 1120

This line stores a maximum number of transferred data (in bytes) per one transfer, that is 24 KByte, into Max_bsize variable.

Lines 1140 to 1160

Clears the error queue.

Line 1180

Displays a measurement start message.

Lines 1190 to 1200

These lines create the header part indicating that data will be sent as many as Src_size_char\$ bytes, then send the header part of the command and its parameters for writing the data to the file on the E5071C.

Line 1220

This line converts Src_size_char\$ to a real number and stores it into Src_size variable.

Lines 1230 to 1370

These lines repeat the procedures below until Src_size reaches 0.

Lines 1240 to 1280: If Src_size is greater than Max_bsize, these lines assign the value of the Max_bsize to Block_size variable (transferred data in bytes). If Src_size is equal or less than Max_bsize, assign the value of Src_size to Block_size.

Line 1300 This line defines Dat\$ string variable with the size as large as Block_size and reserves memory area.

Line 1310 This line creates a format for reading characters as many as Block_size characters.

Line 1320 This line reads data from the file on the external controller, then stores them into Dat\$.

Line 1330 This line writes the contents of Dat\$ to the file on the E5071C.

Lines 1340 to 1360 These lines free the memory area for Dat\$ and subtract Block_size from Src_size.

Line 1390

This line sends a message terminator at the end of data.

E5071C

Lines 1420 to 1430

These lines retrieve the error number and error message from the error queue, and then store them into the variables Err_no and Err_msg\$, respectively.

Lines 1440 to 1490

If Err_no is equal to 0 (no error occurred), these lines display the message indicating completion of transfer, and if Err_no is not equal to 0 (an error occurred), display Err_no along with Err_msg\$.

Lines 1510 to 1520

These lines handle the case with no source file for copying is found.

Fixture Simulator

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to use the Fixture Simulator.

This configures the Balance-Unbalance Conversion, Differential Matching Circuit Embedding, Port Impedance Conversion, and Differential Port Impedance Conversion features so that the instrument can correctly deal with an unbalance-balance (3-port) DUT.

See Analysis Using the Fixture Simulator for this programming.

Sample Program in Excel VBA

```
Sub Fixture_Click()
    Dim defrm As Long      'Session to Default Resource Manager
    Dim vi As Long         'Session to instrument
    Dim Para As String
    Dim Dev_Type As String
    Dim Dmc_c As Double
    Dim Dmc_g As Double
    Dim Dmc_l As Double
    Dim Dmc_r As Double
    Dim Z0_se As Double
    Dim Z0_diff As Double
    Dim Port(3) As Integer
    Dim Result As String

    Const TimeOutTime = 20000

    Dev_Type = "SBAL"      'Device Type : SE-Bal
    Port(1) = 1            'Port1(SE) : 1
    Port(2) = 2            'Port2(Bal) : 2,3
    Port(3) = 3            '
    Para = "SDS21"         'Meas. Parameter : Sds21
    Dmc_c = 0.000000000001 'Diff. C : 1 pF
    Dmc_g = 0.002          'Matching G : 2 mS
    Dmc_l = 0.000000003    'Circuit L : 30 nH
```

E5071C

Dmc_r = 0.004 ' R : 4 mohm
Z0_se = 100 'Z Conversion Z0 : 100 ohm
Z0_diff = 210 'Diff. Z Conv.Z0 : 210 ohm

Call viOpenDefaultRM(defrm) 'Initializes the VISA system.
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi) 'Opens the session to the specified instrument.
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) 'The state of an attribute for the specified session.

Call viVPrintf(vi, "**RST" & vbLf, 0) 'Presets the setting state of the ENA.
Call viVPrintf(vi, "**CLS" & vbLf, 0) 'Clears the all status register.

Call viVPrintf(vi, ":CALC1:FSIM:BAL:DEV " & Dev_Type & vbLf, 0) 'Sets the balance device type.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:TOP:" & Dev_Type & " " & Port(1) & "," & Port(2) & "," & Port(3) & vbLf, 0) 'Sets the port assignment.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:PAR1:" & Dev_Type & " " & Para & vbLf, 0) 'Sets the measurement parameter of port 1.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:PAR1:STAT ON" & vbLf, 0) 'Turns on balance-unbalance conversion.
Call ErrorCheck(vi) 'Checking the error.

Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1 PLPC" & vbLf, 0) 'Specifies the type of differential matching circuit.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C " & Dmc_c & vbLf, 0) 'Sets the differential matching circuit constants C.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G " & Dmc_g & vbLf, 0) 'Sets the differential matching circuit constants G.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:L " & Dmc_l & vbLf, 0) 'Sets the differential matching circuit constants L.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R " & Dmc_r & vbLf, 0) 'Sets the differential matching circuit constants R.
Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:STAT ON" & vbLf, 0) 'Turns of differential matching circuit embedding.
Call ErrorCheck(vi) 'Checking the error.

Call viVPrintf(vi, ":CALC1:FSIM:SEND:ZCON:PORT2:Z0 " & Z0_se & vbLf, 0) 'Sets the port impedance of port 2.
Call viVPrintf(vi, ":CALC1:FSIM:SEND:ZCON:PORT3:Z0 " & Z0_se & vbLf, 0) 'Sets the port impedance of port 3.
Call viVPrintf(vi, ":CALC1:FSIM:SEND:ZCON:STAT ON" & vbLf, 0) 'Turns on port impedance conversion.

Call ErrorCheck(vi) 'Checking the error.

Call viVPrintf(vi, ":CALC1:FSIM:BAL:DZC:BPOR1:Z0 " & Z0_diff & vbLf, 0) 'Sets the differential port impedance of balance port 1.

Call viVPrintf(vi, ":CALC1:FSIM:BAL:DZC:STAT ON" & vbLf, 0) 'Turns on differential port impedance conversion.

Call viVPrintf(vi, ":CALC1:FSIM:STAT ON" & vbLf, 0) 'Turns on the fixture simulator.

Call ErrorCheck(vi) 'Checking the error.

Call viClose(vi) 'Closes the resource manager session.

Call viClose(defrm) 'Breaks the communication and terminates the VISA system.

End

End Sub

Sub ErrorCheck(vi As Long)

Dim err As String * 50, ErrNo As Variant, Response

Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err) 'Reads error message.

ErrNo = Split(err, ",") 'Gets the error code.

If Val(ErrNo(0)) <> 0 Then

Response = MsgBox(CStr(ErrNo(1)), vbOKOnly) 'Display the message box.

End If

End Sub

Sample Program in HT Basic (fixture.htb)

10 DIM Dev_type\$(9),Para\$(9)

20 REAL Dmc_c,Dmc_g,Dmc_l,Dmc_r,Z0_se,Z0_diff

30 INTEGER Port(1:3)

40 !

50 ASSIGN @Agte507x TO 717

60 !

70 Dev_type\$="SBAL" ! Device Type : SE-Bal

80 Port(1)=1 ! Port1(SE) : 1

90 Port(2)=2 ! Port2(Bal) : 2,3

100 Port(3)=3 !

110 Para\$="SDS21" ! Meas. Parameter : Sds21

120 Dmc_c=1.E-12 ! Diff. C: 1 pF

130 Dmc_g=2.E-3 ! Matching G: 2 mS

E5071C

140 Dmc_l=3.E-8 ! Circuit L: 30 nH
150 Dmc_r=4.E-3 ! R: 4 mohm
160 Z0_se=100 ! Z Conversion Z0: 100 ohm
170 Z0_diff=210 ! Diff. Z Conv. Z0: 210 ohm
180 !
190 ! Balance-Unbalance Conversion Setting
200 !
210 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DEV "&Dev_type\$
220 OUTPUT @Agte507x;".CALC1:FSIM:BAL:TOP:"&Dev_type\$&" ";Port(*)
230 OUTPUT @Agte507x;".CALC1:FSIM:BAL:PAR1:"&Dev_type\$&" "&Para\$
240 OUTPUT @Agte507x;".CALC1:FSIM:BAL:PAR1:STAT ON"
250 !
260 ! Diff. Matching Circuit Setting
270 !
280 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DMC:BPOR1 PLPC"
290 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DMC:BPOR1:PAR:C ";Dmc_c
300 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DMC:BPOR1:PAR:G ";Dmc_g
310 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DMC:BPOR1:PAR:L ";Dmc_l
320 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DMC:BPOR1:PAR:R ";Dmc_r
330 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DMC:STAT ON"
340 !
350 ! Z Conversion Setting
360 !
370 OUTPUT @Agte507x;".CALC1:FSIM:SEND:ZCON:PORT2:Z0 ";Z0_se
380 OUTPUT @Agte507x;".CALC1:FSIM:SEND:ZCON:PORT3:Z0 ";Z0_se
390 OUTPUT @Agte507x;".CALC1:FSIM:SEND:ZCON:STAT ON"
400 !
410 ! Diff. Z Conversion Setting
420 !
430 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DZC:BPOR1:Z0 ";Z0_diff
440 OUTPUT @Agte507x;".CALC1:FSIM:BAL:DZC:STAT ON"
450 !
460 ! Fixture Simulator On/Off
470 !
480 OUTPUT @Agte507x;".CALC1:FSIM:STAT ON"
490 !
500 END

Time Domain

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use the transformation function of the time domain function.

This program executes calibration (ECal), performs measurement once, converts the results to data in the time domain, and displays this data.

See Analysis in Time Domain (time domain function) for this programming.

Sample Program in Excel VBA

Sub Time_Domain_Click()

Dim defrm As Long 'Session to Default Resource Manager

Dim vi As Long 'Session to instrument

Dim Para As String

Dim Tran_Type As String

Dim Stim_Type As String

Dim stop_freq As Double

Dim Win_Beta As Double

Dim Star_Time As Double

Dim Stop_Time As Double

Dim Result As String

Const TimeOutTime = 40000 'timeout time.

stop_freq = 3000000000# 'Stop Frequendy : 3 GHz

Nop = 201 'Nop : 201

Para = "S11" 'Meas. Parameter : S11

Tran_Type = "LPAS" 'Transform Type : Lowpass

Stim_Type = "IMP" 'Stimulus Type : Impulse

Win_Beta = 13 'Window Beta : 13 (Maximum Type)

Star_Time = 0 'Start time : 0 sec

Stop_Time = 0.00000001 'Stop time : 10 nsec

Call viOpenDefaultRM(defrm) 'Initializes the VISA system.

E5071C

Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi) specified instrument.	'Opens the session to the
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) for the specified session.	'The state of an attribute
Call viVPrintf(vi, ":SYST:PRES" & vbLf, 0)	'Presets the setting state of the ENA.
Call viVPrintf(vi, "*CLS" & vbLf, 0)	'Clears the all status register.
Call viVPrintf(vi, ":SENS1:FREQ:STOP " & stop_freq & vbLf, 0) frequency.	'Sets the sweep stop
Call viVPrintf(vi, ":SENS1:SWE:POIN " & Nop & vbLf, 0)	'Sets the number of points.
Call viVPrintf(vi, ":CALC1:TRAN:TIME:LPFR" & vbLf, 0)	'Sets a measurement point.
Call viVPrintf(vi, ":CALC1:PAR1:DEF " & Para & vbLf, 0) parameter.	'Sets the measurement
Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)	'Sets the trigger source to BUS.
Call ErrorCheck(vi)	'Checking the error.
MsgBox "Connect Port1 to Ecal Module. Then click OK button.", vbOKOnly box.	'Display the message
Call viVPrintf(vi, ":SENS1:CORR:COLL:ECAL:SOLT1 1" & vbLf, 0) calibration.	'Execute the 1-port
Call viVPrintf(vi, "*OPC?" & vbLf, 0)	'Sets the *OPC? command.
Call viVScanf(vi, "%t", Result)	'Reads the *OPC? result.
Call ErrorCheck(vi)	'Checking the error.
MsgBox "Set DUT. Then click OK button.", vbOKOnly	'Display the message box.
Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)	'Execute the trigger.
Call viVPrintf(vi, "*OPC?" & vbLf, 0)	'Sets the *OPC? command.
Call viVScanf(vi, "%t", Result)	'Reads the *OPC? result.
Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)	'Execute auto scale.
MsgBox "Click OK button. < Time Domain Transform >", vbOKOnly box.	'Display the message
Call viVPrintf(vi, ":CALC1:TRAN:TIME " & Tran_Type & vbLf, 0) type.	'Sets the transformation

Call viVPrintf(vi, ":CALC1:TRAN:TIME:STIM " & Stim_Type & vbLf, 0)	'Sets the stimulus type.
Call viVPrintf(vi, ":CALC1:TRAN:TIME:KBES " & Win_Beta & vbLf, 0) the window.	'Sets the beta value of
Call viVPrintf(vi, ":CALC1:TRAN:TIME:STAR " & Star_Time & vbLf, 0) the display range.	'Sets the start value of
Call viVPrintf(vi, ":CALC1:TRAN:TIME:STOP " & Stop_Time & vbLf, 0) the display range.	'Sets the end value of
Call viVPrintf(vi, ":CALC1:TRAN:TIME:STAT ON" & vbLf, 0) function.	'Turns on the time domain
Call ErrorCheck(vi)	'Checking the error.
Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)	'Sets the active trace.
Call viVPrintf(vi, ":CALC1:FORM REAL" & vbLf, 0)	'Sets the real data format.
Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)	'Execute auto scale.
Call ErrorCheck(vi)	'Checking the error.
Call viClose(vi)	'Closes the resource manager session.
Call viClose(defrm) the VISA system.	'Breaks the communication and terminates
End	'End
End Sub	

Sample Program in HT Basic

```

10 DIM Para$(9),Tran_type$(9),Stim_type$(9),Buff$(9),Inp_ch ar$(9)
20 REAL Stop_freq,Win_beta,Star_time,Stop_time
30 INTEGER Nop
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 Stop_freq=3.E+9 ! Stop Frequency : 3 GHz
80 Nop=201 ! Nop : 201
90 Para$="S11" ! Meas. Parameter : S11
100 !
110 Tran_type$="LPAS" ! Transform Type : Lowpass
120 Stim_type$="IMP" ! Stimulus Type : Impulse
130 Win_beta=13 ! Window Beta : 13 (Maximum Type)
140 Star_time=0 ! Start time : 0 s
150 Stop_time=1.E-8 ! Stop time : 10 ns
160 !

```

E5071C

```
170 OUTPUT @Agte507x;".SYST:PRES"
180 OUTPUT @Agte507x;".SENS1:FREQ:STOP ";Stop_freq
190 OUTPUT @Agte507x;".SENS1:SWE:POIN ";Nop
200 !
210 OUTPUT @Agte507x;".CALC1:TRAN:TIME:LPFR"
220 !
230 OUTPUT @Agte507x;".CALC1:PAR1:DEF "&Para$
240 OUTPUT @Agte507x;".TRIG:SOUR BUS"
250 !
260 ! 1 Port Full Calibration (ECal)
270 !
280 PRINT "Connect Port 1 to ECal Module. Then push [Enter] key."
290 INPUT "",Buff$
300 OUTPUT @Agte507x;".SENS1:CORR:COLL:ECAL:SOLT1 1"
310 OUTPUT @Agte507x;".SYST:ERR?"
320 ENTER @Agte507x;Buff$
330 !
340 ! Measurement
350 !
360 PRINT "Set DUT. Then Push [Enter] key."
370 INPUT "",Inp_char$
380 !
390 OUTPUT @Agte507x;".TRIG:SING"
400 OUTPUT @Agte507x;".*OPC?"
410 ENTER @Agte507x;Buff$
420 !
430 OUTPUT @Agte507x;".DISP:WIND1:TRAC1:Y:AUTO"
440 PRINT "Push [Enter] key. -> [Time Domain Transform]"
450 INPUT "",Inp_char$
460 !
470 ! Time Domain Transform
480 !
490 OUTPUT @Agte507x;".CALC1:TRAN:TIME "&Tran_type$
500 OUTPUT @Agte507x;".CALC1:TRAN:TIME:STIM "&Stim_type$
510 OUTPUT @Agte507x;".CALC1:TRAN:TIME:KBES ";Win_beta
520 OUTPUT @Agte507x;".CALC1:TRAN:TIME:STAR ";Star_time
530 OUTPUT @Agte507x;".CALC1:TRAN:TIME:STOP ";Stop_time
```

```
540 OUTPUT @Agte507x;":CALC1:TRAN:TIME:STAT ON"  
550 !  
560 OUTPUT @Agte507x;":CALC1:PAR1:SEL"  
570 OUTPUT @Agte507x;":CALC1:FORM REAL"  
580 OUTPUT @Agte507x;":DISP:WIND1:TRAC1:Y:AUTO"  
590 END
```

Control Using SICL-LAN Server

- [Overview](#)
- Sample Program in Excel VBA

Other topics about Sample Application Program

Overview

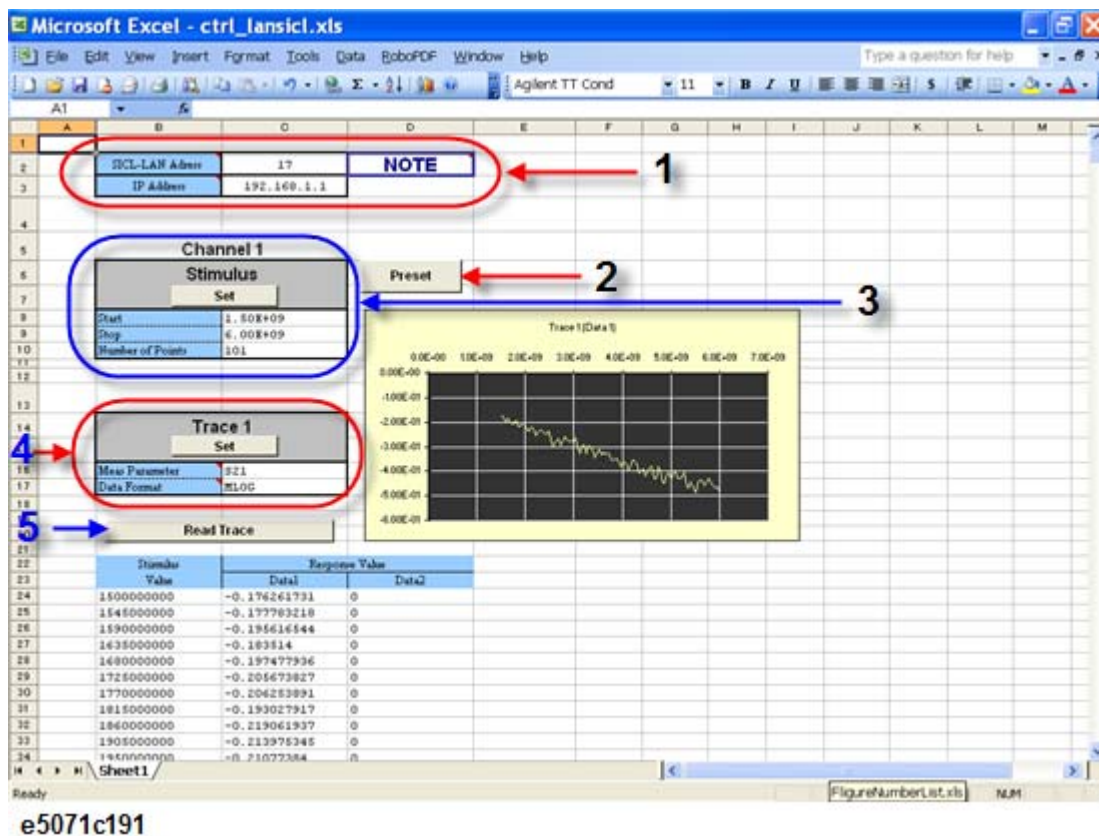
This section explains how to control the E5071C by using SICL in the Windows environment.

- To control the E5071C using the SICL-LAN server, you need to make the preparations described in Control over SICL-LAN server.

Sample Program in Excel VBA

Opening **ctrl_lansicl.xls** in Microsoft Excel displays a screen as shown in the figure below:

ctrl_lansicl.xls



For how to use each element in ctrl_lansicl.xls, refer to the following description.

1. In part 1, in the cell to the right of the SICL-LAN Address, enter the address of the E5071C for control with the SICL-LAN server. This

address is **XX**, which has been set with the command **System > Misc Setup > Network Setup > SICL-LAN Address [XX]**. Enter the IP address of the E5071C in the cell to the right of the IP Address. This VBA macro will not work properly without the correct values in these two cells.

2. Click **Preset** in part 2 to execute the presetting operation.
3. In part 3, the sweep range (start and stop points) and the number of measurement points for channel 1 are set. Click **Set** to execute the setting as shown in the setting table.
4. Part 4 sets the measurement parameters and data format for trace 1 in channel 1. Click **Set** to execute the setting as shown in the setting table.
5. Click **Read Trace** in part 5 retrieves the formatted data array of trace 1 in channel 1. The data is displayed in tabular and graphical format.

Description of Operation in VBA macro

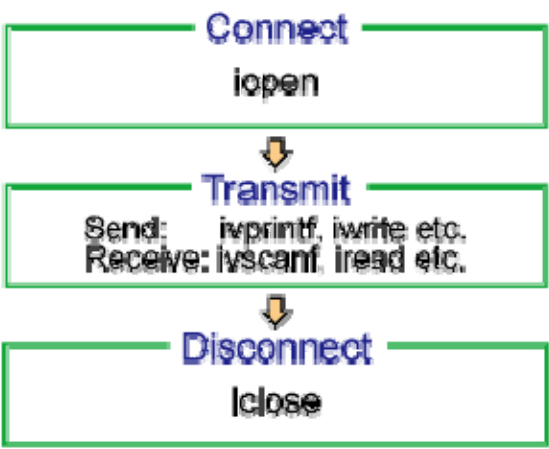
This section describes the operation of the VBA macro, focusing on the part related to control with SICL.

- In order to use SICL in your VBA macro, you must declare functions and define variables with a SICL definition file (for VB).
- In the VBA macro, `ctrl_sicllan.xls`, the standard module whose object name is "SICL," is the definition file.

The basic control flow with SICL is shown in Flow of control using SICL.

- In this sample program, the **ivprintf** function, the **ivscanf** function, and the **iread** function are used in its communication part; you can use other SICL functions as well. For details, refer to `sicl.hlp` (the online help of SICL).

Flow of control using SICL



E5071C441

- For more information on how to use each function of SICTL, refer to the SICTL manual.

The procedures of each step in Flow of control using SICTL are described below.

Connection

The procedure corresponding to connection is OpenSession (OpenSession). OpenSession establishes a connection to the E5071C with the **iopen** function of SICTL, using the SICTL-LAN Address and IP Address entered in part 1 in ctrl_lansicl.xls. The **iopen** function takes the address information of the E5071C you specify as its parameters.

Syntax

addr = iopen(*dev*)

Variable	addr
Description	Session information (output)
Data type	Integer type
Description	dev
	Address information of the instrument you specify (input)
	Character string type

Grammar

sicl-name [*ip-address*]:*interface*, *sicl-lan-address*

For example, if the parameter (*dev*) is "lan[192.168.0.1]:hpi9,17," connection is made to the address of **17** of the interface of **hpi9** with the E5071C whose IP address is **192.168.0.1** by using the external controller whose SICL interface name is **lan**.

OpenSession

```
Function OpenSession() As Integer
Dim ServAddr As String
Dim IpAddr As String
On Error GoTo ErrHandler
""Get Sicl-Lan Address
Sheets("Sheet1").Select
Range("C2").Select
ServAddr = ActiveCell.FormulaR1C1
""Get Ip Address
Sheets("Sheet1").Select
Range("C3").Select
IpAddr = ActiveCell.FormulaR1C1

OpenSession = iopen("lan[" & IpAddr & "]:hpi9," & ServAddr)
Call itimeout(OpenSession, 10000)
Exit Function
ErrHandler:
MsgBox "*** Error : " & Error$
Call sicleanup
End
End Function
```

Sending

The procedure corresponding to sending in communication is OutputSicLan. OutputSicLan uses the **ivprintf** function of SICL to send messages (SCPI commands). The **ivprintf** function takes the session information output from the **iopen** function and a program message as its parameters.

Syntax

E5071C

Status = ivprintf(*addr*,*mes*)

Variable	Status
Description	Return value of the function (output)
Data type	Integer type
	addr
Description	Session information (input)
Data type	Integer type
	mes
Description	Program message (input)
Data type	Character string type

OutputSicLan

```
Sub OutputSicLan(addr As Integer, message As String)
```

```
Dim Status As Integer
```

```
Dim actualcnt As Long
```

```
Dim length As Long
```

```
On Error GoTo ErrHandler
```

```
length = Len(message)
```

```
Status = ivprintf(addr, message & Chr$(10))
```

```
Exit Sub
```

```
ErrHandler:
```

```
MsgBox "**** Error : " & Error$
```

```
Call sicleanup
```

```
End
```

```
End Sub
```

Receiving

The procedure corresponding to receiving ASCII format messages in communication is EnterSicLan. EnterSicLan uses the **ivscanf** function of SICL to receive a message in ASCII format and store it into the output variable. The **ivscanf** function takes the session information output from the **iopen** function, the format for output, and the data to be output as its parameters.

Syntax

Status = ivscanf(*addr*,*fmt*,*ap*)

Variable	fmt
Description	Format for output (input)
Data type	Character string type
	ap
Description	Data to be output (output)
Data type	Character string type

For information on the variable (*Status*) and the variable (*addr*), refer to Variable.

In Visual Basic, variables must be declared as a fixed-length string when receiving string data using the **ivscanf** function.

EnterSicLan

```
Sub EnterSicLan(addr As Integer, Query As String)
```

```
Dim Status As Integer
```

```
Dim actualcnt As Long
```

```
Dim res As String * 256
```

E5071C

```
On Error GoTo ErrHandler
Status = ivscanf(addr, "%t", res)
Query = Trim(res)
Exit Sub
```

```
ErrHandler:
MsgBox "**** Error : " & Error$
Call siclcleanup
End
```

```
End Sub
```

The procedure corresponding to receiving array data in communication is EnterSicLanArrayReal64, which uses the **iread** function of SICL to receive array data in the IEEE 64-bit floating point binary transfer format and store it into the output variable. The **iread** function takes the session information output from the **iopen** function, the data to be output, the number of data bytes, the condition to finish reading data, and the number of data bytes actually read out as its parameters.

Syntax

```
Status = iread(addr,buf,bufsize,reason,actual)
```

Variable	buf
Description	Data to be output (output)
Data type	Character string type
	bufsize
Description	The number of data bytes (input)
Data type	Long integer type
	reason
Description	The condition to finish reading out data (input)
Data type	Integer type
	actual

Description	The number of data bytes actually read out (output)
Data type	Long integer type

For information on the variable (*Status*) and the variable (*addr*), refer to Variable.

Each functional part of EnterSicLanArrayReal64 is described below.

- (1) Retrieves the data header.
- (2) Stores the number of data bytes into the size variable in the header part.
- (3) Retrieves the formatted data array for trace 1 in channel 1 and stores it into the databuf variable.
- (4) Retrieves the message terminator at the end of the data.

EnterSicLanArrayReal64

Function EnterSicLanArrayReal64(addr As Integer, databuf() As Double)
As Long

```

Dim Status As Integer
Dim actualcnt As Long
Dim buf As String * 8
Dim size As Long
On Error GoTo ErrHandler
""Read header info of "#6NNNNNN"
Status = iread(addr, buf, 8, I_TERM_MAXCNT, actualcnt) '.....(1)
size = Val(Mid$(buf, 3, 6)) '.....(2)
""Read data
Status = iread(addr, databuf, size, I_TERM_MAXCNT, actualcnt) '.....(3)
""Read ending LF
Status = iread(addr, buf, 1, I_TERM_MAXCNT, actualcnt) '.....(4)
EnterSicLanArrayReal64 = size / 8
Exit Function

```

ErrHandler:

E5071C

```
MsgBox "**** Error : " & Error$  
Call sicleanup  
End  
End Function
```

Disconnection

The **iclose** function of SICL is used to disconnect communication. The **iclose** function takes the session information output from the **iopen** function as its parameter.

Syntax

Status = iclose(*addr*)

For information on the variable (*Status*) and the variable (*addr*), refer to Variable

Sample control

The E5071C can be controlled by executing the above procedures in order, following the control flow in Flow of control using SICL. This is demonstrated by the Preset procedure (a procedure that is executed when the Preset button is clicked) as described in Preset.

Preset

```
Sub Preset()  
  
"" Open Session  
E507x = OpenSession  
  
""Presetting the analyzer  
Call OutputSicLan(E507x, "":SYST:PRES")  
""Close Session  
Call iclose(E507x)  
  
End Sub
```

Control Using Telnet Server

- [Overview](#)
- Sample Program in Excel VBA

Other topics about Sample Programs

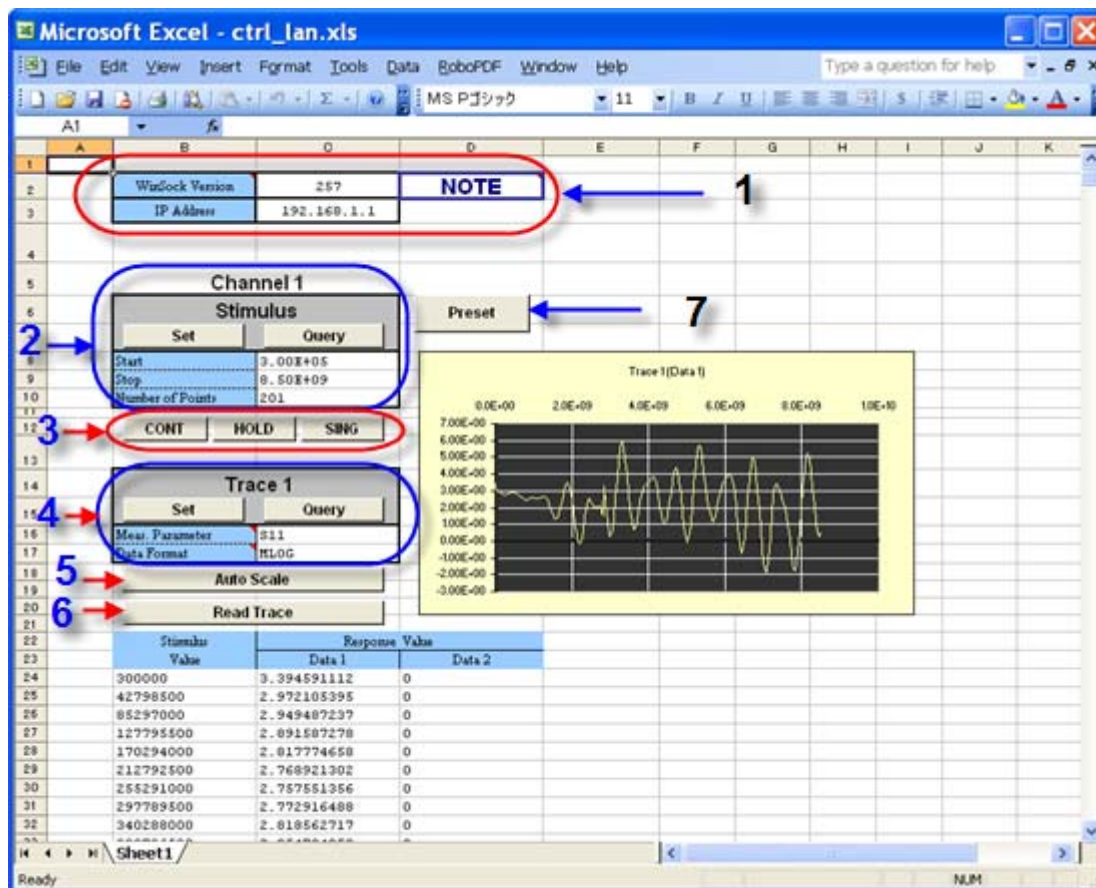
Overview

This section explains how to control the E5071C by using WinSock API in the Windows environment.

Sample Program in Excel VBA

Opening **ctrl_lan.xls** in Microsoft Excel displays the screen shown in the figure below.

ctrl_lan.xls



e5071c192

For how to use each element in ctrl_lan.xls, refer to the following description.

1. Enter the version number of WinSock API in the cell to the right side of "Winsock Version." The version number is obtained by multiplying 256 by the major version and then adding the minor version. For example, when the version of your Winsock API is 1.1, the version

number is obtained as follows: $256 \times 1 + 1 = 257$. Enter the IP address of the E5071C in the cell to the right side of "IP Address." This VBA macro will not work properly without the correct values in these two cells.

2. In part 2, the sweep range (start and stop points) and the number of measurement points are set. Click **Set** to execute the setting operation as specified with the setting table, while clicking the button labeled "Query" retrieves the current settings of the E5071C.
3. Part 3 is dedicated to setting the trigger mode.
4. Part 4 sets the measurement parameters and data format for trace 1 in channel 1. Click **Set** to execute the setting operation as specified with the setting table, while clicking the button labeled "Query" retrieves the current settings of the E5071C.
5. In part 5, click **Auto Scale** to execute auto scaling for trace 1 in channel 1.
6. Click **Read Trace** in part 6 to retrieve the formatted data of trace 1 in channel 1. The data is displayed in tabular and graphical formats.
7. Click **Preset** to execute the presetting operation.

Description of operation in VBA macro

This section describes the operation of the VBA macro, focusing on the part related to control with WinSock API.

In order to use WinSock API, you must declare functions and define variables with a definition file of WinSock API, as shown in Definition file of WinSock API.

Definition file of WinSock API

'This is the Winsock API definition file for Visual Basic

'Setup the variable type 'hostent' for the WSASStartup command

Type Hostent

h_name As Long

h_aliases As Long

h_addrtype As String * 2

h_length As String * 2

h_addr_list As Long

End Type

Public Const SZHOSTENT = 16

'Set the Internet address type to a long integer (32-bit)

Type in_addr

s_addr As Long

End Type

'A note to those familiar with the C header file for Winsock

'Visual Basic does not permit a user-defined variable type

'to be used as a return structure. In the case of the

'variable definition below, sin_addr must

'be declared as a long integer rather than the user-defined

'variable type of in_addr.

Type sockaddr_in

sin_family As Integer

sin_port As Integer

sin_addr As Long

sin_zero As String * 8

End Type

Public Const WSADESCRIPTION_LEN = 256

Public Const WSASYS_STATUS_LEN = 128

Public Const WSA_DescriptionSize = WSADESCRIPTION_LEN + 1

Public Const WSA_SysStatusSize = WSASYS_STATUS_LEN + 1

'Setup the structure for the information returned from

'the WSASStartup() function.

Type WSADATA

wVersion As Integer

wHighVersion As Integer

szDescription As String * WSA_DescriptionSize

szSystemStatus As String * WSA_SysStatusSize

iMaxSockets As Integer

iMaxUdpDg As Integer

lpVendorInfo As String * 200

End Type

'Define socket return codes

Public Const INVALID_SOCKET = &HFFFF

E5071C

Public Const SOCKET_ERROR = -1

'Define socket types

Public Const SOCK_STREAM = 1 'Stream socket

Public Const SOCK_DGRAM = 2 'Datagram socket

Public Const SOCK_RAW = 3 'Raw data socket

Public Const SOCK_RDM = 4 'Reliable Delivery socket

Public Const SOCK_SEQPACKET = 5 'Sequenced Packet socket

'Define address families

Public Const AF_UNSPEC = 0 'unspecified

Public Const AF_UNIX = 1 'local to host (pipes, portals)

Public Const AF_INET = 2 'internetwork: UDP, TCP, etc.

Public Const AF_IMPLINK = 3 'arpanet imp addresses

Public Const AF_PUP = 4 'pup protocols: e.g. BSP

Public Const AF_CHAOS = 5 'mit CHAOS protocols

Public Const AF_NS = 6 'XEROX NS protocols

Public Const AF_ISO = 7 'ISO protocols

Public Const AF_OSI = AF_ISO 'OSI is ISO

Public Const AF_ECMA = 8 'european computer manufacturers

Public Const AF_DATAKIT = 9 'datakit protocols

Public Const AF_CCITT = 10 'CCITT protocols, X.25 etc

Public Const AF_SNA = 11 'IBM SNA

Public Const AF_DECnet = 12 'DECnet

Public Const AF_DLI = 13 'Direct data link interface

Public Const AF_LAT = 14 'LAT

Public Const AF_HYLINK = 15 'NSC Hyperchannel

Public Const AF_APPLETALK = 16 'AppleTalk

Public Const AF_NETBIOS = 17 'NetBios-style addresses

Public Const AF_MAX = 18 'Maximum # of address families

'Setup sockaddr data type to store Internet addresses

Type sockaddr

sa_family As Integer

sa_data As String * 14

End Type

Public Const SADDRLEN = 16

'Declare Socket functions

```
Public Declare Function closesocket Lib "wsck32.dll" (ByVal s As Long) As Long
```

```
Public Declare Function connect Lib "wsck32.dll" (ByVal s As Long, addr As sockaddr_in, ByVal  
namelen As Long) As Long
```

```
Public Declare Function htons Lib "wsck32.dll" (ByVal hostshort As Long) As Integer
```

```
Public Declare Function inet_addr Lib "wsck32.dll" (ByVal cp As String) As Long
```

```
Public Declare Function recv Lib "wsck32.dll" (ByVal s As Long, ByVal buf As Any, ByVal buflen As  
Long, ByVal flags As Long) As Long
```

```
Public Declare Function recvB Lib "wsck32.dll" Alias "recv" (ByVal s As Long, buf As Any, ByVal buflen  
As Long, ByVal flags As Long) As Long
```

```
Public Declare Function send Lib "wsck32.dll" (ByVal s As Long, buf As Any, ByVal buflen As Long,  
ByVal flags As Long) As Long
```

```
Public Declare Function socket Lib "wsck32.dll" (ByVal af As Long, ByVal socktype As Long, ByVal  
protocol As Long) As Long
```

```
Public Declare Function WSASStartup Lib "wsck32.dll" (ByVal wVersionRequired As Long, lpWSAData  
As WSAData) As Long
```

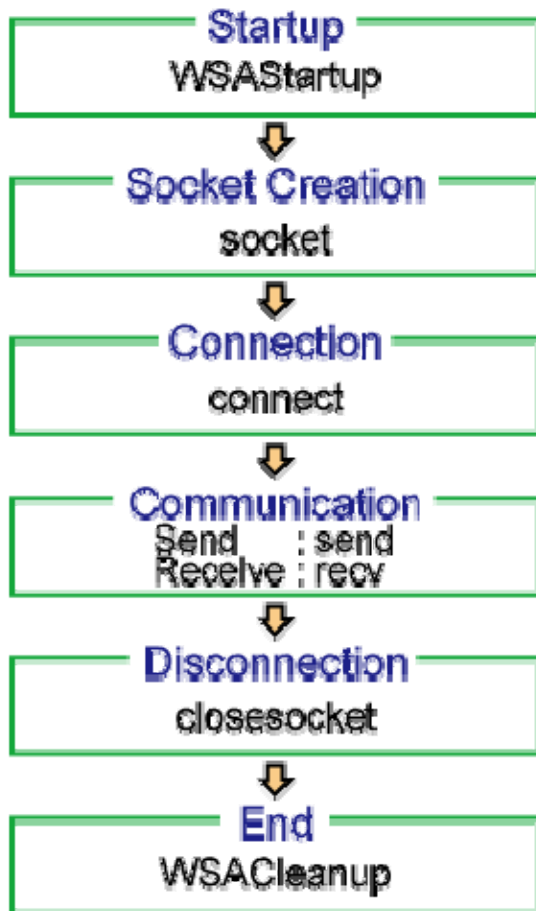
```
Public Declare Function WSACleanup Lib "wsck32.dll" () As Long
```

```
Public Declare Function WSAUnhookBlockingHook Lib "wsck32.dll" () As Long
```

```
Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (hpdDest As Any, hpdSource As  
Any, ByVal cbCopy As Long)
```

The basic control flow with WinSock API is shown in the figure below:

Control flow with WinSock API



e5071c372

The procedures of each step in Control flow with WinSock API are described below.

Startup

The procedure corresponding to Startup is StartIt. StartIt launches and initializes WinSock API with **WSAStartup**, whose version is shown in part 1 of ctrl_lan.xls. The function WSAStartup should always be used when initiating WinSock. This function takes the version number (input) and launching information (output) as its parameters.

StartIt

Sub StartIt()

Dim StartUpInfo As WSADATA

'Version 1.1 (1*256 + 1) = 257

```
'version 2.0 (2*256 + 0) = 512
'Get WinSock version
Sheets("Sheet1").Select
Range("C2").Select
version = ActiveCell.FormulaR1C1
'Initialize Winsock DLL
x = WSASStartup(version, StartUpInfo)

End Sub
```

Socket Creation and Connection

The procedure for Socket Creation and Connection is **OpenSocket**. **OpenSocket** makes a connection to an instrument associated with the IP address specified with the input parameter **Hostname**. It uses a socket of the port specified with the input parameter **PortNumber**. Each functional part of **OpenSocket** is described below.

In (1), the **inet_addr** function of WinSock API is used to convert an IP address delimited by "." to an Internet address.

In (2), a new socket is created with the **socket** function of WinSock API and its socket descriptor is obtained. If an error occurs, the control returns to the main program with a message. The socket function takes an address family (input), a socket type (input), and a protocol number (input) as its parameters.

In (3), the socket address is specified. Note that **htons**, which is used for specifying the port number, is a function of WinSock API. This function converts a 2-byte integer from the Windows byte order (little endian) to the network byte order (big endian).

In (4), a connection to the E5071C is made by using the **connect** function of WinSock API. If an error occurs, the control returns to the main program with a message. The connect function takes a socket descriptor (input), a socket address (input), and the size of the socket address (input) as its parameters.

OpenSocket

```
Function OpenSocket(ByVal Hostname As String, ByVal PortNumber As Integer) As Integer
Dim l_SocketAddress As sockaddr_in
Dim ipAddress As Long
ipAddress = inet_addr(Hostname) '.....(1)
```

E5071C

```
'Create a new socket
socketId = socket(AF_INET, SOCK_STREAM, 0) '
If socketId = SOCKET_ERROR Then '
MsgBox ("ERROR: socket = " + Str$(socketId)) '.....(2)
OpenSocket = COMMAND_ERROR '
Exit Function '
End If '

'Open a connection to a server

l_SocketAddress.sin_family = AF_INET '
l_SocketAddress.sin_port = htons(PortNumber) '.....(3)
l_SocketAddress.sin_addr = ipAddress '
l_SocketAddress.sin_zero = String$(8, 0) '

x = connect(socketId, l_SocketAddress, Len(l_SocketAddress)) '
If socketId = SOCKET_ERROR Then '
MsgBox ("ERROR: connect = " + Str$(x)) '..(4)
OpenSocket = COMMAND_ERROR '
Exit Function '
End If '

OpenSocket = socketId

End Function
```

Communication

The procedure corresponding to Communication is SendCommand. SendCommand transmits a message (SCPI command) specified with the input parameter "command" to the E5071C using the **send** function of WinSock API. The send function takes a socket descriptor (input), a message to be transmitted (input), message length (input) and a flag (input) as its parameters.

SendCommand

```
Function SendCommand(ByVal command As String) As Integer
```

```

Dim strSend As String
strSend = command + vbCrLf
count = send(socketId, ByVal strSend, Len(strSend), 0)
If count = SOCKET_ERROR Then
MsgBox ("ERROR: send = " + Str$(count))
SendCommand = COMMAND_ERROR
Exit Function
End If
SendCommand = NO_ERROR

```

```
End Function
```

The procedure corresponding to the Receiving part of communication is `RecvAscii` and other functions. `RecvAscii` receives a message in ASCII format and stores it in the `dataBuf` output parameter. Maximum length of the message is specified with the `maxLength` input parameter. Each functional part of `RecvAscii` is described below.

In (1), a message (a response to a query for a SCPI command) is received from the E5071C as a series of characters using the **recv** function of WinSock API. If an error occurs, the control returns to the main program with a message. The `recv` function takes a socket descriptor (input), a message to be received (input), message length (input) and a flag (input) as its parameters.

In (2), it is determined whether each received character is LF (ASCII code: 10). When it is LF, receiving is terminated by adding NULL (ASCII code: 0) to the end of the `dataBuf` string and the control returns to the main program.

In (3), the number of the last characters that were read out is added to the count value for checking the number of received characters, and the characters are appended to the end of the `dataBuf` string.

RecvAscii

```
Function RecvAscii(dataBuf As String, ByVal maxLength As Integer) As Integer
```

```

Dim c As String * 1
Dim length As Integer
dataBuf = ""
While length < maxLength
DoEvents
count = recv(socketId, c, 1, 0) '
If count < 1 Then '

```

E5071C

```
RecvAscii = RECV_ERROR '.....(1)
dataBuf = Chr$(0) '
Exit Function '
End If '
If c = Chr$(10) Then '
dataBuf = dataBuf + Chr$(0) '.....(2)
RecvAscii = NO_ERROR '
Exit Function '
End If '
length = length + count '.....(3)
dataBuf = dataBuf + c '
Wend
RecvAscii = RECV_ERROR
End Function
```

Disconnection

The procedure corresponding to Disconnection is CloseConnection. CloseConnection disconnects communication and removes a socket using the **closesocket** function of WinSock API. The closesocket function takes a socket descriptor (input) as its parameter.

C

```
Sub CloseConnection()
```

```
x = closesocket(socketId)
If x = SOCKET_ERROR Then
MsgBox ("ERROR: closesocket = " + Str$(x))
Exit Sub
End If
```

```
End Sub
```

```
End
```

The procedure corresponding to End is EndIt. EndIt disconnects WinSock API using the **WSACleanup** function of WinSock API. The function WSACleanup should always be used when terminating WinSock.

```
EndIt
```

```
Sub EndIt()
```



```
'Shutdown Winsock DLL
x = WSACleanup()
```

```
End Sub
```

Example of control

The E5071C can be controlled by executing the above procedures in order, following the control flow in Control flow with WinSock API. This is demonstrated by the procedure autoscale (a procedure that is executed when the Auto Scale button is clicked) as described in autoscale.

```
autoscale
```

```
Sub autoscale()
```

```
,
```

```
' auto scaling
```

```
,
```

```
Call StartIt
```

```
Call get_hostname
```

```
x = OpenSocket(Hostname$, ScpiPort)
```

```
x = SendCommand(":DISP:WIND1:TRAC1:Y:AUTO")
```

```
Call CloseConnection
```

```
Call EndIt
```

```
End Sub
```

When you execute more than one command by connecting and disconnecting a socket for every command, the sequence of execution may change.

Control LCD Update Timing

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program is provided in this section where the command processing time is improved by controlling the update timing of the LCD display.

- This sample program correctly runs when the maximum number of channels/traces is set to 9 channels/9 traces.

This program sets the necessary measurement conditions and then turns off the updating of the LCD display. Next, it performs measurement, reads out the result, and updates the screen once. This program repeats this measurement procedure ten times.

See Improving Command Processing Speed for this programming.

Sample Program in HT Basic ([cont_upd.htb](#))

```

10 REAL Trace1(1:201,1:2),Trace2(1:201,1:2)
20 DIM Buff$(9),Img$(30)
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte507x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 !
80 OUTPUT @Agte507x;".SENS1:SWE:TYPE LIN"
90 OUTPUT @Agte507x;".SENS1:FREQ:CEN 950E6"
100 OUTPUT @Agte507x;".SENS1:FREQ:SPAN 100E6"
110 OUTPUT @Agte507x;".SENS1:SWE:POIN 201"
120 OUTPUT @Agte507x;".TRIG:SOUR BUS"
130 OUTPUT @Agte507x;".INIT1:CONT ON"
140 FOR I=2 TO 9
150 OUTPUT @Agte507x;".INIT"&VAL$(I)&".CONT OFF"
160 NEXT I
170 !
180 OUTPUT @Agte507x;".DISP:SPL D1"
190 OUTPUT @Agte507x;".DISP:WIND1:SPL D1_2"
200 !
210 OUTPUT @Agte507x;".CALC1:PAR:COUN 2"

```

```

220 OUTPUT @Agte507x;".CALC1:PAR1:DEF S21"
230 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
240 OUTPUT @Agte507x;".CALC1:FORM MLOG"
250 OUTPUT @Agte507x;".CALC1:PAR2:DEF S11"
260 OUTPUT @Agte507x;".CALC1:PAR2:SEL"
270 OUTPUT @Agte507x;".CALC1:FORM MLOG"
280 !
290 OUTPUT @Agte507x;".DISP:ENAB OFF"
300 OUTPUT @Agte507x;".FORM:DATA REAL"
310 !
320 FOR I=1 TO 10
330 OUTPUT @Agte507x;".TRIG:SING"
340 OUTPUT @Agte507x;"*OPC?"
350 ENTER @Agte507x;Buff$
360 !
370 ! Read Trace Data
380 !
390 OUTPUT @Agte507x;".CALC1:PAR1:SEL"
400 OUTPUT @Agte507x;".CALC1:DATA:FDAT?"
410 ENTER @Agte507x USING "#,8A";Buff$
420 ENTER @Binary;Trace1(*)
430 ENTER @Agte507x USING "#,1A";Buff$
440 !
450 OUTPUT @Agte507x;".CALC1:PAR2:SEL"
460 OUTPUT @Agte507x;".CALC1:DATA:FDAT?"
470 ENTER @Agte507x USING "#,8A";Buff$
480 ENTER @Binary;Trace2(*)
490 ENTER @Agte507x USING "#,1A";Buff$
500 !
510 ! Update Display
520 !
530 OUTPUT @Agte507x;".DISP:UPD"
540 NEXT I
550 END

```

Description

Lines 50 to 60

E5071C

Assigns a GPIB address to the I/O pass.

Lines 80 to 110

These lines set the sweep type to linear sweep, the sweep center value to 950 MHz, the sweep span value to 100 MHz, and the number of measurement points to 201.

Lines 120 to 160

These lines set the trigger source to bus trigger, turn on Continuous Activation mode for channel 1, and turn the mode off for channels 2 through 9.

Lines 180 to 190

These lines display the window for channel 1 only and arrange two graphs tiled horizontally.

Lines 210 to 270

These lines set the number of traces for channel 1 to 2, the measurement parameter and its data format for trace 1 to S21 and Log Mag, respectively, and those for trace 2 to S11 and Log Mag, respectively.

Line 290

This line turns Off the updating of the LCD screen.

Line 300

This line sets the data transfer format to binary.

Lines 320 to 540

These lines repeat the following procedure ten times.

Lines 340 to 360: These lines trigger the instrument and wait until the measurement cycle finishes.

Lines 400 to 440: Reads out the formatted data array of trace 1 in channel 1.

Lines 460 to 500: Reads out the formatted data array of trace 2 in channel 1.

Line 540: This line updates the LCD screen once.

Handler Interface

- Overview
- Program Code

Other topics about Sample Programs

Overview

The sample program communicates with an external instrument through the handler I/O port.

This program outputs 5 (sets bit 2 and bit 0 to Low, and the other bits to High) to the port A of the handler I/O port, then waits until the bit 3 of the port C is set to Low.

See Inputting/Outputting Data for this programming.

Program Code

Excel VBA

```
Sub Handler_Click()
    Dim defrm As Long      'Session to Default Resource Manager.
    Dim vi As Long         'Session to instrument.
    Dim Out_Data As Integer 'Decimal value.
    Dim In_Data As Long
    Dim Bit_stat As Integer
    Dim Flag_bit As Integer
    Dim Out_Data_Bin As String
    Dim Ret As Long        'Return value.
    Dim i As Long
    Dim X As Long
    Const TimeOutTime = 40000 'timeout time.
    Out_Data_Bin = "00000101" 'Store the output data on the port A (binaly).
    Flag_bit = 3              'Bit location (bit 3).

    Call viOpenDefaultRM(defrm) 'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi) 'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) 'The state of an attribute for the
    specified session.

    Call viVPrintf(vi, "**RST" & vbLf, 0) 'Presets the setting state of the ENA.
    Call viVPrintf(vi, "**CLS" & vbLf, 0) 'Clears the all status register.
```

E5071C

Call viVPrintf(vi, ":CONT:HAND:C:MODE INP" & vbLf, 0) 'Configures the port C to input port.
Call viVPrintf(vi, ":CONT:HAND:IND:STAT ON" & vbLf, 0) 'Line enable /INDEX signal.
Call viVPrintf(vi, ":CONT:HAND:RTR:STAT ON" & vbLf, 0) 'Line enable /READY FOR TRIGGER signal.

For i = 1 To Len(Out_Data_Bin) 'Convert Out_Dara_Bin to a decimal value.

If Mid(Out_Data_Bin, Len(Out_Data_Bin) - i + 1, 1) = "1" Then

X = 2 ^ (i - 1)

Ret = Ret + X

End If

Next i

Out_Data = Ret 'Sets the decimal value.

Call viVPrintf(vi, ":CONT:HAND:A " & Ret & vbLf, 0) 'Sets to the port A.

Call viVPrintf(vi, ":CONT:HAND:C?" & vbLf, 0) 'Outputs data to output port C.

Call viVScanf(vi, "%t", In_Data) 'Reads data from the port C.

Call ErrorCheck(vi) 'Checking the error.

Call viClose(vi) 'Closes the resource manager session.

Call viClose(defrm) 'Breaks the communication and terminates the VISA system.

End

End Sub

Sub ErrorCheck(vi As Long)

Dim err As String * 50, ErrNo As Variant, Response

Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err) 'Reads error message.

ErrNo = Split(err, ",") 'Gets the error code.

If Val(ErrNo(0)) <> 0 Then

Response = MsgBox(CStr(ErrNo(1)), vbOKOnly) 'Display the message box.

End If

End Sub

HT Basic (handler.htb)

10 INTEGER Out_data, In_data, Bit_stat

20 DIM Out_data_bin\$(9)

```
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 Out_data_bin$="00000101"
70 Flag_bit=3
80 !
90 OUTPUT @Agte507x;":CONT:HAND:C:MODE INP"
100 OUTPUT @Agte507x;":CONT:HAND:IND:STAT ON"
110 OUTPUT @Agte507x;":CONT:HAND:RTR:STAT ON"
120 !
130 Out_data=IVAL(Out_data_bin$,2)
140 OUTPUT @Agte507x;":CONT:HAND:A ";Out_data
150 !
160 REPEAT
170 OUTPUT @Agte507x;":CONT:HAND:C?"
180 ENTER @Agte507x;In_data
190 Bit_stat=BIT(In_data,Flag_bit)
200 UNTIL Bit_stat=1
210 END
```

Controlling E5091A

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This program assigns Port 1 of the E5091A to A, Port 2 to T2, Port 3 to R2, and Port 4 to R2 and sets Line 1 and Line 3 of the control line to HIGH.

See Controlling E5091A by Programming for this programming.

Sample Program in Excel VBA

```

Sub e5091ctrl_Click()
    Dim defrm As Long      'Session to Default Resource Manager
    Dim vi As Long         'Session to instrument
    Dim Port1 As String
    Dim Port2 As String
    Dim Port3 As String
    Dim Port4 As String
    Dim Line1 As String
    Dim Line2 As String
    Dim Line3 As String
    Dim Line4 As String
    Dim Line5 As String
    Dim Line6 As String
    Dim Line7 As String
    Dim Line8 As String
    Dim Data_Bin As String
    Dim Data_Dec As Long
    Dim i As Integer
    Dim X As Long
    Dim Model As String

    Const TimeOutTime = 20000

    Model = "E5091_9"           'Model   E5091A-009

    Port1 = "A"                 'Port1 = A
    Port2 = "T2"                'Port2 = T2

```


Port3 = "R2"	'Port3 = R2
Port4 = "R2"	'Port4 = R2
Line1 = "1"	'Line1 = High
Line2 = "0"	'Line2 = Low
Line3 = "1"	'Line3 = High
Line4 = "0"	'Line4 = Low
Line5 = "0"	'Line5 = Low
Line6 = "0"	'Line6 = Low
Line7 = "0"	'Line7 = Low
Line8 = "0"	'Line8 = Low
Call viOpenDefaultRM(defrm)	'Initializes the VISA system.
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)	'Opens the session to the
specified instrument.	
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)	'The state of an attribute
for the specified session.	
Call viVPrintf(vi, "*RST" & vbLf, 0)	'Presets the setting state of the ENA.
Call viVPrintf(vi, "*CLS" & vbLf, 0)	'Clears the all status register.
Call viVPrintf(vi, ":SENS1:MULT1:NAME " & Model & vbLf, 0)	'Specifies the test set
name.	
Call viVPrintf(vi, ":SENS:MULT:PORT1 " & Port1 & vbLf, 0)	'Sets the port assigned to port 1
to A	
Call viVPrintf(vi, ":SENS:MULT:PORT2 " & Port2 & vbLf, 0)	'Sets the port assigned to port 2
to T2	
Call viVPrintf(vi, ":SENS:MULT:PORT3 " & Port3 & vbLf, 0)	'Sets the port assigned to port 3
to R2	
Call viVPrintf(vi, ":SENS:MULT:PORT4 " & Port4 & vbLf, 0)	'Sets the port assigned to port 4
to R2	
Call ErrorCheck(vi)	'Checking the error.
Data_Bin = Line8 & Line7 & Line6 & Line5 & Line4 & Line3 & Line2 & Line1	
For i = 1 To Len(Data_Bin)	'Creates a decimal setting value.
If Mid(Data_Bin, Len(Data_Bin) - i + 1, 1) = "1" Then	

E5071C

```

    X = 2 ^ (i - 1)
    Data_Dec = Data_Dec + X
End If
Next i
Call viVPrintf(vi, ":SENS:MULT:OUTP " & CStr(Data_Dec) & vbCrLf, 0)    'Sets the control line.

    Call viVPrintf(vi, ":SENS:MULT:DISP ON" & vbCrLf, 0)                'Turns on the E5091A property
display.
    Call viVPrintf(vi, ":SENS:MULT:STAT ON" & vbCrLf, 0)                'Turns on the control of the
E5091A.

    Call ErrorCheck(vi)                                                  'Checking the error.

    Call viClose(vi)                                                     'Closes the resource manager session.
    Call viClose(defrm)                                                  'Breaks the communication and terminates
the VISA system.

End                                                                    'End
End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbCrLf, "%t", err)                'Reads error message.
    ErrNo = Split(err, ",")                                              'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)                    'Display the message box.
    End If
End Sub
End Sub
```

Sample Program in HT Basic (e5091ctr_2.htb)

```

10 DIM Port1$(3),Port2$(3),Port3$(3),Port4$(3),Data_bin$(9)
20 DIM Line1$(3),Line2$(3),Line3$(3),Line4$(3)
30 DIM Line5$(3),Line6$(3),Line7$(3),Line8$(3)
40 DIM Model$(8)
50 INTEGER Data_dec
60 !
70 ASSIGN @Agte507x TO 717
80 !
```

```

90 Model$="E5091_9" ! Model:E5091A-009
100 !
110 Port1$="A" ! Port1: A
120 Port2$="T2" ! Port2: T2
130 Port3$="R2" ! Port3: R2
140 Port4$="R2" ! Port4: R2
150 !
160 Line1$="1" ! Line1: HIGH
170 Line2$="0" ! Line2: Low
180 Line3$="1" ! Line3: HIGH
190 Line4$="0" ! Line4: Low
200 Line5$="0" ! Line5: Low
210 Line6$="0" ! Line6: Low
220 Line7$="0" ! Line7: Low
230 Line8$="0" ! Line8: Low
240 !
250 OUTPUT @Agte507x;".SENS1:MULT1:NAME "&Model$
260 !
270 OUTPUT @Agte507x;".SENS1:MULT1:PORT1 "&Port1$
280 OUTPUT @Agte507x;".SENS1:MULT1:PORT2 "&Port2$
290 OUTPUT @Agte507x;".SENS1:MULT1:PORT3 "&Port3$
300 OUTPUT @Agte507x;".SENS1:MULT1:PORT4 "&Port4$
310 !
320 Data_bin$=Line8$&Line7$&Line6$&Line5$&Line4$&Line3$&Line2$&Line1$
330 Data_dec=IVAL(Data_bin$,2)
340 OUTPUT @Agte507x;".SENS1:MULT1:OUTP ";Data_dec
350 !
360 OUTPUT @Agte507x;".SENS:MULT1:DISP ON"
370 OUTPUT @Agte507x;".SENS:MULT1:STAT ON"
380 !
390 END

```

Built-in VBA Programming

Built-in VBA Programming

- Introduction to VBA Programming
- Operation Basics
- Controlling E5071C
- Controlling Peripherals
- Application Programs
- Complex Operation Library
- Waveform Analysis Library

Other topics about Programming

Introduction to VBA Programming

Introduction to VBA Programming

- Introduction of the E5071C Macro Function
- An Overview of a Control System Based on the Macro Function
- Overview of E5071C COM Object

Introduction of the E5071C Macro Function

- [Overview](#)
- [Macro Function](#)

Other topics about Introduction to VBA Programming

Overview

The E5071C has a built-in macro function that allows a single instruction to substitute for multiple instructions. You can have the E5071C automatically execute your own macro program that contains a series of VBA (Visual Basic for Application) statements. The macro function allows you to run a variety of applications; you can control not only the E5071C but also various peripherals from your own macro code.

The VBA is based on the VB (Visual Basic) programming language. Although the VBA is similar to the VB, they are not the same. The VBA is decreased some of the VB's features and added characteristic features for each application. The E5071C VBA is added features for controlling the E5071C. For details of difference between the VBA and the VB, refer to Microsoft official guides, and various books on VBA.

For information on the basic operating procedures for the E5071C's VBA, see Operation Basics. This manual is not meant to be an in-depth guide to VBA programming basics and the syntax of VBA functions and commands. Such in-depth information is covered in VBA Help, Microsoft official guides, and various books on VBA.

NOTE

The built-in VBA can not control the ENA option TDR application.

Macro Function

The macro function allows you to control the E5071C itself as well as various peripherals. You can do the following:

1. Automate repetitive tasks

You can use the E5071C's macro function to combine several processes into one. Automating repetitive tasks provides higher efficiency and eliminates human error. Once you have contained repetitive tasks in Sub procedures, you can later call the procedures from other programs, thus allowing effective reuse of programming assets.

2. Implement a user interface

The E5071C VBA supports user forms that simplify creating a visual user interface. User forms guide users through common tasks such as performing measurement and entering data, without requiring familiarity with the E5071C, thus minimizing the possibility of human error.

An Overview of a Control System Based on the Macro Function

- [Overview](#)
- [Implementing a Control System](#)
- [Required Equipment](#)
- [Control Methods](#)

Other topics about Introduction to VBA Programming

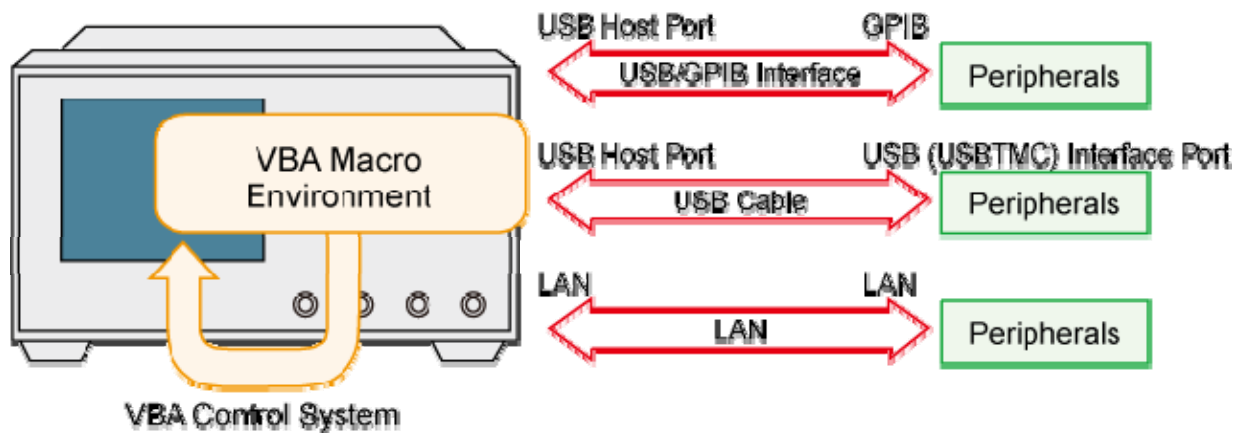
Overview

This section describes how you can use the E5071C's built-in VBA macro function to implement a system that controls the E5071C and peripherals, and what command sets are available for such purposes.

Implementing a Control System

Macro-based control systems are classified into two types: As shown in the following figure, a VBA control system controls the E5071C itself while a VBA remote control system controls peripherals. When you use the macro function to control peripherals, you must connect the E5071C with the peripherals through USB/GPIB interface, USB or LAN, and configure them to communicate over VISA (Virtual Instrument Software Architecture). For information on programming using the VISA library, refer to Programming with VISA.

Configuration example of control system using macro environment



e5071c478

Required Equipment

- E5071C
- Peripherals and/or other instruments that serve your purpose
- USB/GPIB interface, USB Cable, or LAN

Control Methods

E5071C

The command set you can use differs depending on whether you use the macro function to control the E5071C or a peripheral.

Controlling the E5071C

When you want to control the E5071C itself, you can create a program using COM objects within the E5071C VBA environment. COM objects that come with the E5071C include seven objects specific to the COM interface and COM objects that correspond to SCPI commands.

Controlling a Peripheral

When you want to control a peripheral, you can create a program using VISA library functions within the E5071C VBA environment.

For information on using the VISA library, see [Controlling Peripherals](#). For a complete description of VISA functions, refer to the VISA library's online help.

For information on the GPIB commands available with a particular peripheral, refer to the documentation that comes with the peripheral.

Overview of E5071C COM Object

- [Overview](#)
- [About COM Object](#)
- [Property](#)
- [Method](#)
- [Event](#)
- [Using COM Object to Control E5071C](#)
- [Major Control Difference between COM Object and SCPI Command](#)

Other topics about Introduction to VBA Programming

Overview

The E5071C VBA environment provides COM objects that support controlling the E5071C. This section provides an overview of COM objects as well as considerations for using the E5071C's COM objects.

The definitions and specifications of COM are beyond the scope of this guide. Such in-depth information is covered in a variety of books on COM.

About COM Object

When you control the E5071C through the macro function, you can use COM objects as components of your application. The functionality of the E5071C's COM objects is exposed through properties and methods.

Property

A property allows you to read or write a setting or attribute of an object. With the E5071C, you can use properties to set or read the settings of the E5071C.

You can find properties in the list of object types in COM Object Reference.

Method

A method allows you to manipulate an object in a particular way. With the E5071C, you can use methods to perform specific tasks.

You can find methods in the list of object types in COM Object Reference.

Event

An event means an operation from outside that the program can recognize such as clicking a mouse. The E5071C detects events that a specific softkey is pressed using the `UserMenu_OnPress(ByVal Key_id As Long)` procedure to execute the assigned procedure.

You can find events in the list of object types in COM Object Reference.

Using COM Object to Control E5071C

When you want to control the E5071C, you can use COM objects alone or in conjunction with SCPI commands and the Parse object. The latter method is a little slower than the former method because the Parse object

E5071C

is used to parse the messages of SCPI commands. For instructions on using the E5071C's VBA Editor to create a program that uses COM objects, refer to Operation Basics of the E5071C's VBA.

Major Control Difference between COM Object and SCPI Command

While the control using SCPI commands allows SRQ (Service Request) interrupts through the status reporting mechanism, the control using COM objects does not support SRQ interrupts. Instead of SRQ interrupts, you can use the [WaitOnSRQ](#) object to suspend the program until the E5071C is put into the desired state.

Operation Basics

Operation Basics

- Displaying Visual Basic Editor
- Closing Visual Basic Editor
- Switching to the E5071C Measurement Screen
- Making a Preparation Before Coding
- Coding a VBA Program
- Saving a VBA program
- Loading a VBA Program
- Running a VBA Program
- Stopping a VBA Program
- Errors and Debugging
- Printing Output Values in the Echo Window
- Uses Advanced Techniques
- Using VBA Online Help

Displaying Visual Basic Editor

- [Overview](#)
- [Initial Screen of Visual Basic Editor](#)

Other topics about Operation Basics

Overview

This section describes how to launch Visual Basic Editor.

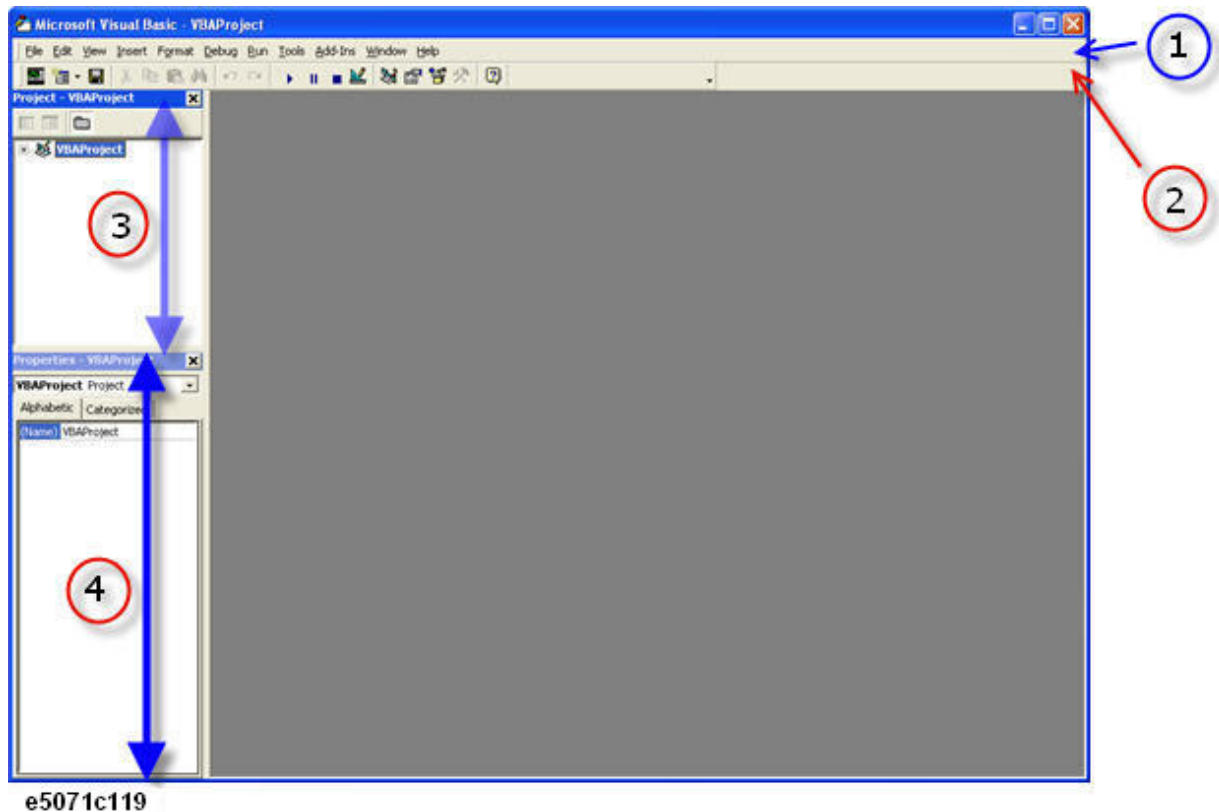
From the E5071C measurement screen, launch Visual Basic Editor using one of the following methods:

1. **Macro Setup > VBA Editor**
2. Press **Alt + F11** keys on the keyboard.

Initial Screen of Visual Basic Editor

When you launch Visual Basic Editor, it displays the initial screen, which contains a number of windows as shown in the following figure. The initial screen provides the following GUI elements:

Example of Visual Basic Editor initial screen



1. Menu bar
2. Toolbar
3. Project Explorer

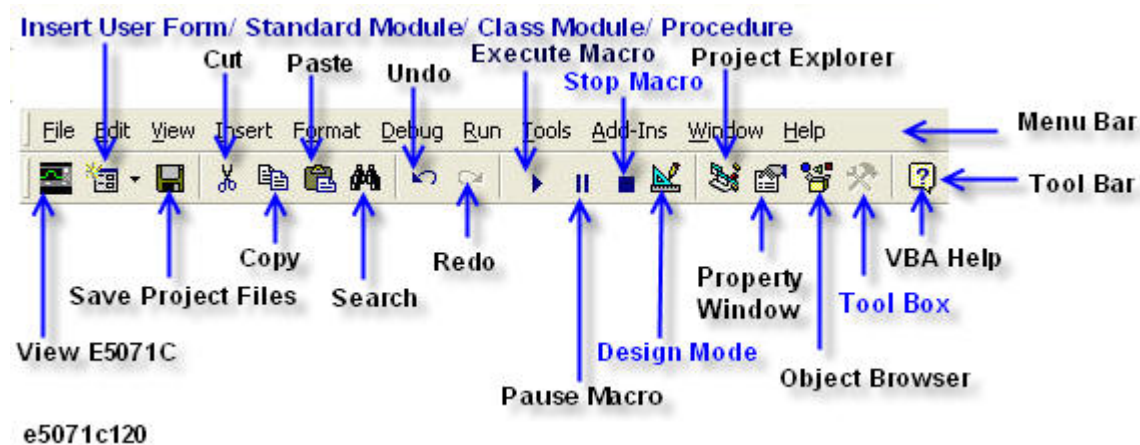
4. Property Window

Menu Bar

Clicking one of the menu labels brings up the corresponding menu. The menu bar can be used as the primary method to navigate through E5071C's VBA environment.

Toolbar


The toolbar provides access to commonly used commands via icon buttons; these commands are a subset of the commands accessible from the menu bar.



Project Explorer

Within the E5071C's VBA environment, you can develop your application as a project that consists of a number of files (modules). Project Explorer shows a list of all files (modules) that make up a project. The list also includes files (modules) created or loaded in Visual Basic Editor. For information on modules, refer to A Project and Three Types of Module.

To display the project explorer, do one of the following:

1. On the **View** menu, click **Project Explorer**.
2. Press **Ctrl + R** keys on the keyboard.
3. On the toolbar, click .


Property Window

A property window shows the settings (label, font, color, size, etc.) of a control (such as a command button or text box) placed on the user form. For information on user forms, refer to User Form.

You can also set properties by programming in the code window.

E5071C

To display the properties window, do one of the following:

1. On the **View** menu, click **Properties Window**.
2. Press **F4** key on the keyboard.
3. On the toolbar, click .

Closing Visual Basic Editor

This section describes how to quit Visual Basic Editor. Close the Visual Basic Editor using either one of the following methods:

- On Visual Basic Editor's **File** menu, click **Close and Return to E5071**.
- Within Visual Basic Editor, press **Alt + Q** keys on the keyboard.
- **Macro Setup > Close Editor** (E5071C measurement screen)
 - Whenever you launch Visual Basic Editor, it automatically displays the project files you were working with in the previous session. However, once you turn off the power to the E5071C, the project files kept in memory will be lost; therefore, it is strongly recommended to save your VBA programs before you turn off the power.

Other topics about Operation Basics

Switching to the E5071C Measurement Screen

You can switch to the E5071C measurement screen without closing Visual Basic Editor.

- On the **View** menu, click **E5071**.
- Press **Alt** + **F11** keys on the keyboard.
- On the toolbar, click "E5071C" icon.
- Press **Focus** key on the E5071C front panel.

Other topics about Operation Basics

Making a Preparation Before Coding

- [A Project and Three Type of Modules](#)
- [Displaying a Code Window](#)

Other topics about Operation Basics

A Project and Three Type of Modules

Project Explorer displays a list of files (modules) that are used in the E5071C VBA. This section describes a project composed of a number of files (modules) and three types of modules ("user form", "standard," and "class"). Each type of module serves its own purposes as described below.

Project

When you develop an application within the E5071C's VBA environment, you use a number of VBA program files (modules), and manage them as one project. The project is saved with the file extension ".vba".

User Form

A user form contains controls such as buttons and text boxes. You can code event-driven procedures that are invoked when a particular event occurs on a particular control, thereby creating a user interface. The user form is saved with the file extension ".frm".

Standard module

A standard module contains a collection of one or more procedures (subprograms enclosed between Sub and End Sub). One typical use of a standard module is to contain shared subroutines and globally called functions. The standard module is saved with the file extension ".bas".

Class Module

A class module contains both data and procedures and acts as one object. Once you have created a class module that serves as an object, you can create any number of instances of that object by naming each instance as an object variable. While each procedure must be unique in a standard module, you can have multiple instances of an object created through a class module. The class module is saved with the file extension ".cls".

Displaying a Code Window

The code windows appear on the Visual Basic Editor by inserting the modules in a project. You can do coding (programming) on this code windows practically.

The E5071C's VBA environment does not allow you to manage multiple projects. When the current project is existing in the Visual Basic Editor by

loading the saved project file, you can replace the current project with a new project by the following method from the E5071C measurement screen.

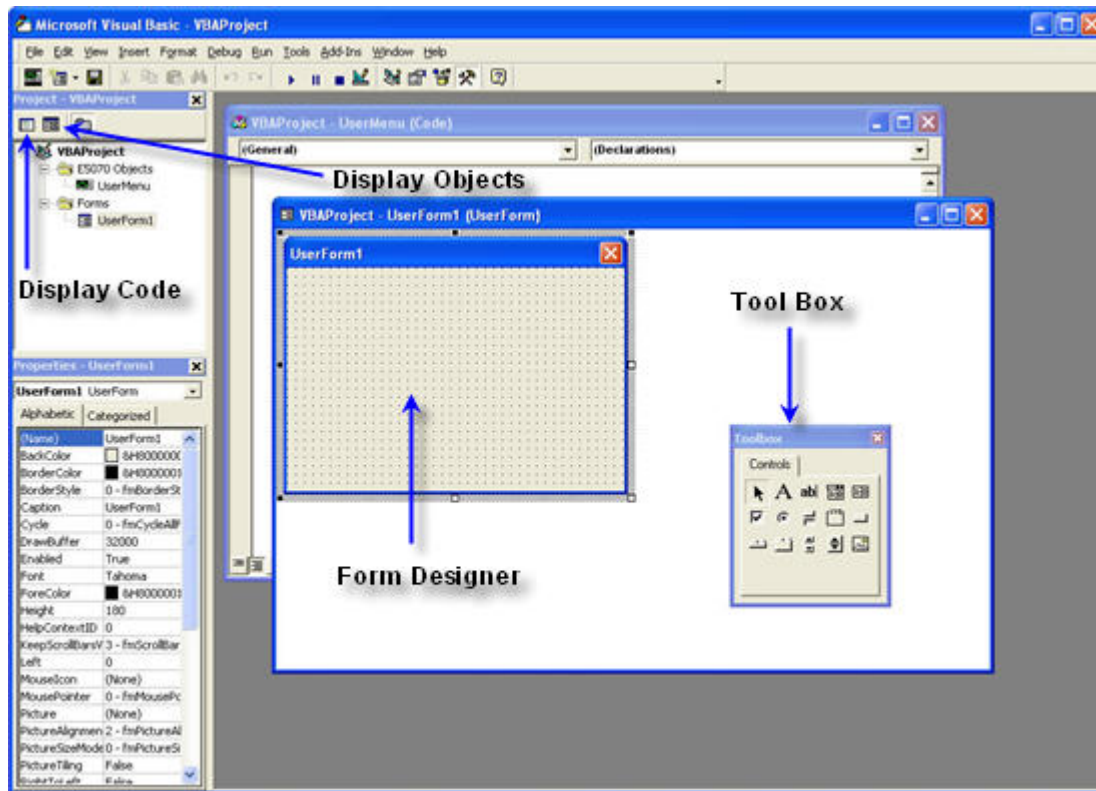
1. **Macro Setup > New Project**
2. When you replace the current project with a new project, the message whether or not the current project is saved may appear. If you want to save the project, click **Yes** button to display a dialog box for saving. For saving the project, see Saving a Project.

Inserting the User Form

Within Visual Basic Editor, do one of the following to add a user form to your project.

1. On the **Insert** menu, click **UserForm**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **UserForm**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert > UserForm**.
4. Adding a user form does not automatically open the code window for that user form. To open the code window, click the "Display Code" icon on Project Explorer in the following figure or double-click a control placed on the user form.

Adding a user form



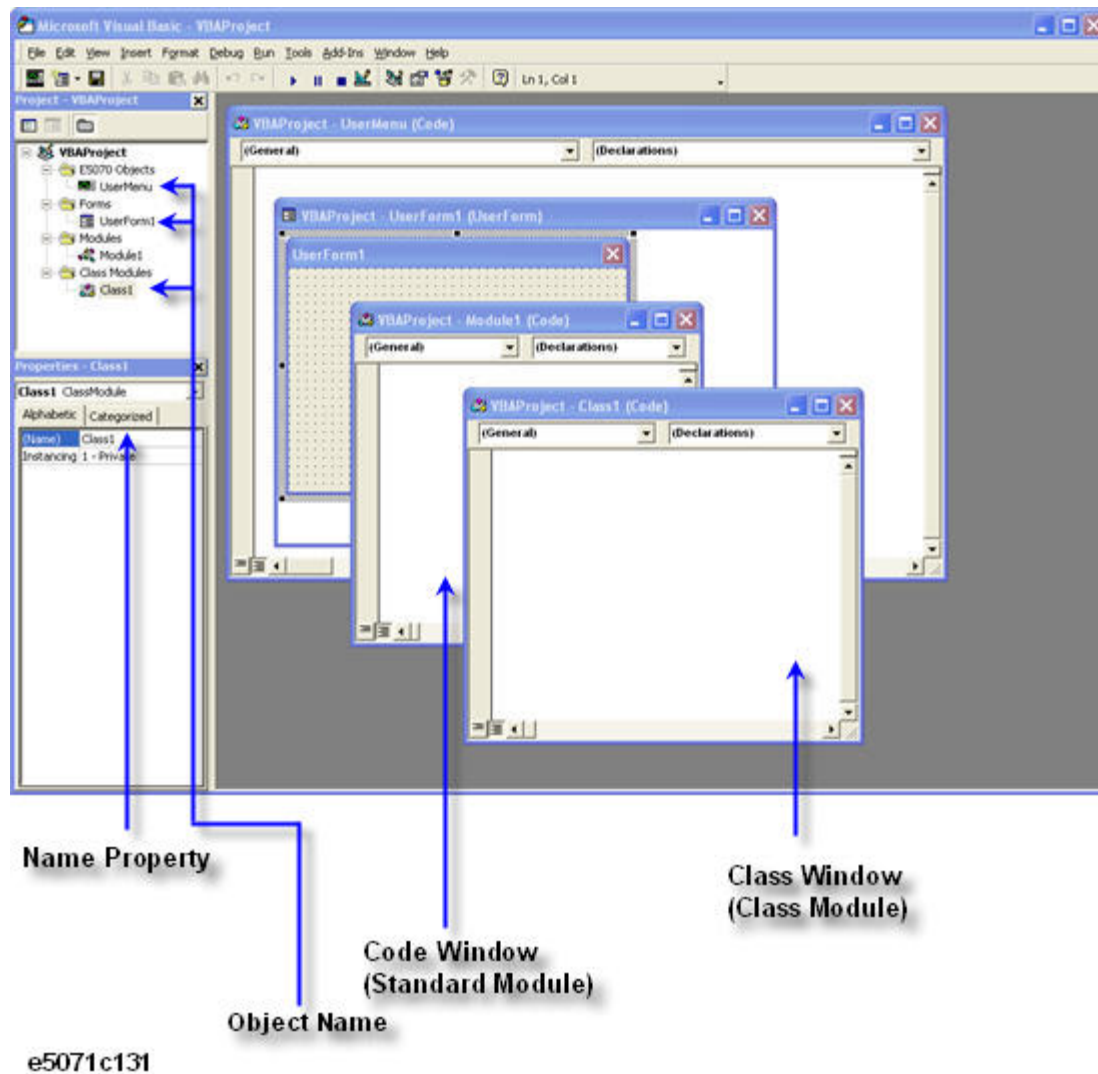
e5071c130

Inserting the Standard Module

Within Visual Basic Editor, do one of the following to add a standard module to your project.

1. On the **Insert** menu, click **Module**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **Module**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert > Module**.

Adding a standard module/class module



Inserting the Class Module

Within Visual Basic Editor, do one of the following to add a class module to your project.

1. On the **Insert** menu, click **ClassModule**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **ClassModule**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert > ClassModule**.

Deleting Modules

You can delete any unnecessary module from the project within Visual Basic Editor. The following procedure assumes that you want to delete a class module named "Class1".

1. In Project Explorer, click the "Class1" class module under the "Class Modules" icon to highlight it.
2. Delete the "Class1" class module using one of the following methods:
 - a. On the **File** menu, click **Remove Class1....**
 - b. Click the right mouse button, and click **Remove Class1....**
3. When you are prompted to confirm whether to export (save) "Class1", click **No**. Alternatively, you can click **Yes** if you want to save the module.

Coding a VBA Program

- [Overview](#)
- [User Interface Elements of a Code Window](#)
- [Creating a Simple VBA Program](#)
- [Auto-complete Feature](#)

Other topics about Operation Basics

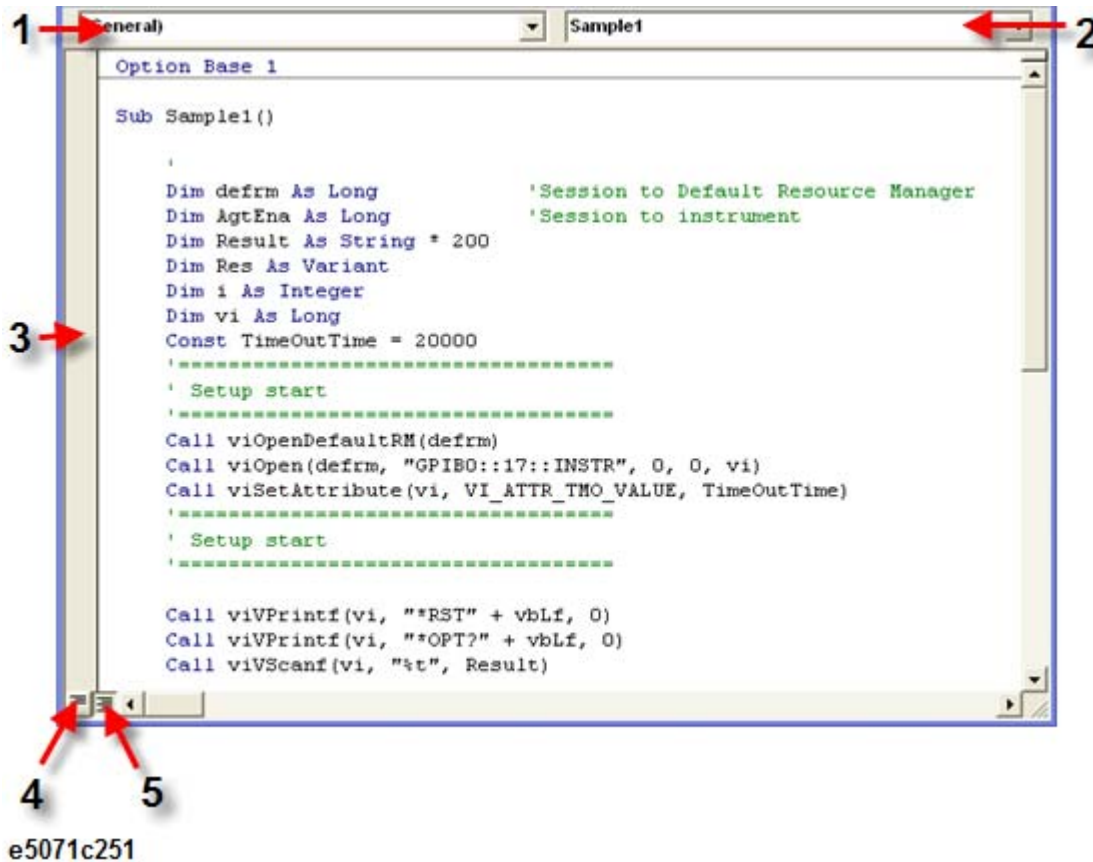
Overview

This section provides descriptive information on the user interface elements of a code window that lets you code a VBA program, and walks through a sample program (procedure) that finds the maximum value contained in an array so you can gain insight into how to create your own programs.

User Interface Elements of a Code Window

A code window is where you code a VBA program. When you are working with a user form, you can open the code window for that user form by double-clicking a control (such as a button or text box) placed on the form. Similarly, when you are working with a standard or class module, you can open the code window associated with that module by double-clicking the module's icon in Project Explorer.

Code window for a standard module



1. Object box

Provides a list of objects currently used within the code window.

2. Procedure box

Provides a list of procedures that reside within the code window. When you are working with a user form, this provides a list of events (actions such as click or double-click).

3. Margin indicator bar

Primarily intended for use when debugging a program.

4. Show Procedure button

Displays only the procedure at the cursor position.

5. Show Module button

Displays the entire program contained in the code window.

[Creating a Simple VBA Program](#)

This section walks through a sample program that finds the maximum value contained in an array while breaking down the code into a number of blocks and describing what they do. Line numbers are added for description purpose only, and do not appear in the actual program source code.

Sample program that finds the maximum value contained in an array

```
10| Option Explicit
20|
30| Sub Maximum()
40|
50| Dim q As Variant
60| Dim x(100) As Integer
70| Dim i As Integer, n As Integer
80| Dim Start As Integer, Last As Integer, Num As Integer
90| Dim Maximum As Integer
100|
110| ' Defining the array
120| q = Array(7, -2, 3, -20, 15, -6, 27, -12, 9, -5, 18, 23, _
130| 9, -16, 22, 0)
140|
150| Start = LBound(q)
160| Last = UBound(q)
170| Num = Last - Start + 1
180|
190| For i = Start To Last
200| x(i) = q(i)
210| Next i
220|
230| Maximum = x(Start)
240|
250| For n = Start + 1 To Last
260| If x(n) > Maximum Then Maximum = x(n)
270| Next n
280|
290| MsgBox Maximum
300|
310| End Sub
```


Let us break down the code into a number of blocks and see what they do.

Line 10

This instruction mandates explicit declaration of variables.

Lines 30 to 310

The code enclosed between Sub Maximum() and End Sub will be executed within the E5071C's macro environment. Thus enclosed code is called a procedure. In this example, "Maximum" is the procedure name.

Lines 50 to 90

These lines declare data types of variables using Dim statements. A statement is the minimum instruction unit based on the syntax. The sample program declares the variable "q" as Variant, and the variables "x(100)", "i", "n", "Start", "Last", "Num", and "Maximum" as Integer. For a complete list of statements and data types supported by VBA, see VBA Online Help.

Line 110

Any text preceded by a comment indicator (') is treated as a comment.

Lines 120 to 130

These lines use VBA's Array function to initialize the array. The q() array contains elements delimited with commas in the ascending order of index numbers (zero-based). A combination of a space and underscore () is used to continue the statement across two or more lines.

Line 150

Stores the starting index number of the q array into the Start variable.

Line 160

Stores the last index number of the q array into the Last variable.

Line 170

Stores the number of elements in the q array into the Num variable.

Lines 190 to 210 and Lines 250 to 270

The code within each For ...Next statement is iterated until the counter reaches the specific number.

Line 200

Stores the contents of the q array (Variant) into the x variable (Integer).

Line 230

Uses the first element of the x array as the tentative maximum value.

Line 260

Compares the tentative maximum value with each of elements that follow; if an element is larger than the tentative maximum value, then that element is used as the tentative maximum value.

Line 290

Uses a message box function to display the maximum value. For a complete list of functions supported by VBA, see VBA Online Help.

NOTE

The above sample program consists of a single procedure contained in a single module. However, when you deal with procedures and variables across multiple modules, you should be aware of the scope of variables and procedures.

Auto-complete Feature

When you use COM objects in Visual Basic Editor, the editor's auto-complete feature allows you to easily type in keywords without misspelling them.

The following procedure assumes that you are entering the SCPI.INITiate(Ch).CONTInuous object.

1. In a standard module, type **sub main** and press the **Enter** key. **End Sub** is automatically added.
2. Typing **scpi** followed by a dot (.) brings up a list of classes under the SCPI class.
3. Typing **in** automatically moves focus to **INITiate** in the list box.
4. Typing **(** brings up a list of indexes.
5. Typing **1).** brings up a list of classes under the INITiate class.
6. Typing **c** automatically moves focus to **CONTInuous** in the list box.
7. Typing **=** brings up a list box for setting a Boolean value (**True/False**).
8. Typing **t** automatically moves focus to **True**.
9. Pressing the **Enter** key completes the statement:
SCPI.INITiate(1).CONTInuous = True.

Saving a VBA program

- [Overview](#)
- [Saving a Project](#)
- [Saving a Module \(Exporting\)](#)

Other topics about Operation Basics

Overview

You can save VBA programs either as one complete project or on a module by module basis.

Saving a Project

When you opt to save your program as one complete project, you can have the files (modules) making up the project into a single package. A project is saved as a .vba file. You can save your program to a project file using one of the following two methods:

Saving a Project from Visual Basic Editor

1. Open the Save As dialog box by doing one of the following:
 - On the **File** menu, click **Save xxx.VBA**. "xxx" represents the file name.
 - On the toolbar, click "Save Project File" icon.
 - Press **Ctrl** + **S** keys on the keyboard.
2. The Save As dialog box appears. Specify the file name and location (drive or folder) and click **Save**.

E5071C Saving a Project from the E5071C Measurement Screen

1. Display the E5071C measurement screen following the instructions given in Switching to the E5071C Measurement Screen.
2. Open the Save As dialog box using the following key sequence:
 - **Macro Setup** > **Save Project**
3. The **Save As dialog box** appears. Specify the file name and location (drive or folder) and click Save.

Saving a Module (Exporting)

Alternatively, you can save each module (user form, standard, or class) of your VBA program individually. To save a module, you must use Visual Basic Editor. User forms are saved as .frm files, standard modules as .bas files, and class modules as .cls files.

1. In Project Explorer, click the file name that appears under the desired module icon to highlight it.

2. Open the Export File dialog box by doing one of the following:
 - On the **File** menu, click **Export File....**
 - Click the right mouse button, and click **Export File....**
 - Press **Ctrl** + **E** keys on the keyboard.
3. The **Export File dialog box** appears. Specify the file name and location (drive or folder) and click **Save**.

Loading a VBA Program

- [Overview](#)
- [Loading a Project](#)

Other topics about Operation Basics

Overview

Once you have saved a project or module file, you can load it later whenever necessary.

Loading a Project

You can load a saved project file either from the E5071C measurement screen or by specifying that the project file be automatically loaded when the power is turned on.

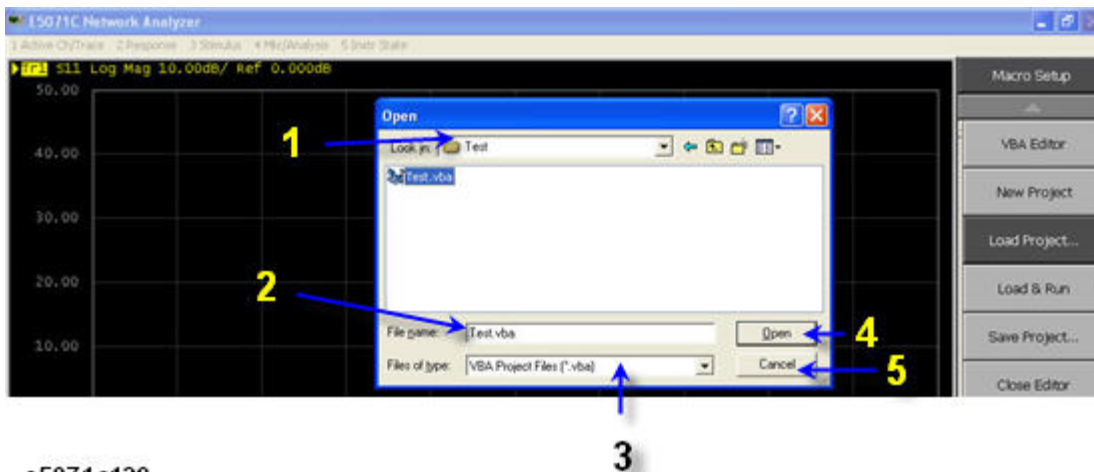
Loading a Project from the E5071C Measurement Screen

1. Press **Macro Setup** key, then click **Load Project**.

NOTE When the another project has already been loaded on the Visual Basic Editor, the message whether or not the current project is saved may appear. If you want to save the project, click **Yes** button to display a dialog box for saving. For saving the project, see Saving a Project.

2. The Open dialog box appears. Specify the file name and location (drive or folder) of the file you want to load and click **Open**.

Open dialog box



The Open dialog box has the following user interface elements:

1. **Look in:** Specify the location (drive or folder) where the project resides.

2. **File name:** Specify the file name of the project you want to load.
3. **Files of type:** Select the type of the file you want load. Normally, you should select **VBA Project Files (*.vba)**.
4. **Open:** Clicking this button loads the project.
5. **Cancel:** Clicking this button closes the Open dialog box and brings you back to the main screen.

Automatically Loading a Project at Power-On

Once you have saved a project file that satisfies the following conditions, the project will be automatically loaded whenever the power is turned ON.

Auto-loaded project	Conditions
Directory where the project resides.	A:\ or D:\
Project file name	autoload.vba

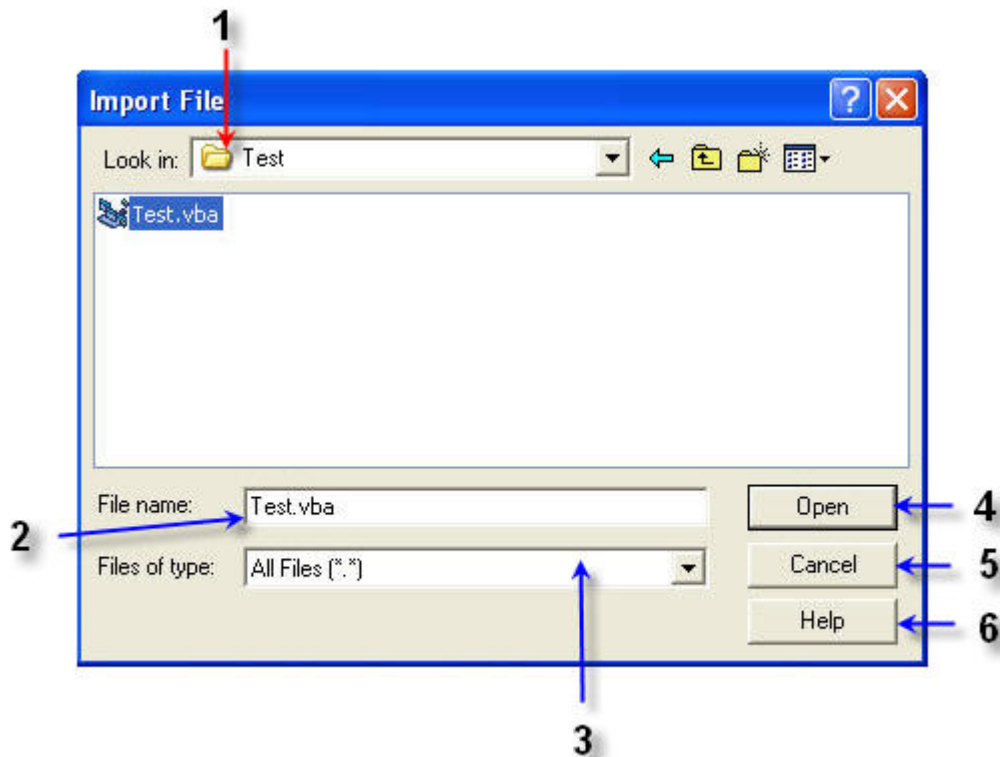
If there is the file named "autoload.vba" in both the A drive and the D drive, the file in the A drive is used.

Loading a Module (Importing)

To load a saved module into a project, you must use Visual Basic Editor.

1. In Project Explorer, click the file name that appears under the desired module icon to highlight it.
2. Open the Import File dialog box by doing one of the following:
 - On the **File** menu, click **Import File....**
 - In Project Explorer, right-click the "VBAProject" icon, and click **Import File....**
 - Press **Ctrl + M** keys on the keyboard.
3. The Import File dialog box appears. Specify the file name and location (drive or folder) of the file (module) you want to load and click **Open**.
4. The Import File dialog box has the following user interface elements:

Import File dialog box



e5071c128

The Import File dialog box has the following user interface elements:

1. **Look in:** Specify the location (drive or folder) where the module resides.
2. **File name:** Specify the file name of the module you want to load.
3. **Files of type:** Select the type of the file you want load. Normally, you should select **VB Files (*.frm,*.bas,*.cls)**.
4. **Open:** Clicking this button loads the module.
5. **Cancel:** Clicking this button closes the Import File dialog box and brings you back to the main screen.
6. **Help:** Clicking this button brings up VBA Online Help.

Running a VBA Program

- [Overview](#)
- [Running a Previously Loaded VBA Program](#)
- Running a Program from the _E5071C_Measurement_Screen
- [Loading and Executing Programs in Batch Process](#)

Other topics about Operation Basics

Overview

The E5071C provides 2 methods to execute a VBA program: executing a program that you previously loaded and loading and executing a program in a batch process. The execution status of the VBA program is indicated in the instrument status bar, as shown in the following figure. "Run" indicates that the program is running while "Stop" indicates that the program is stopped.

Instrument status bar indicating the status of the VBA program



Running a Previously Loaded VBA Program

Running a Program from Visual Basic Editor

The E5071C allows you to run a previous loaded VBA program using one of the four methods listed below.

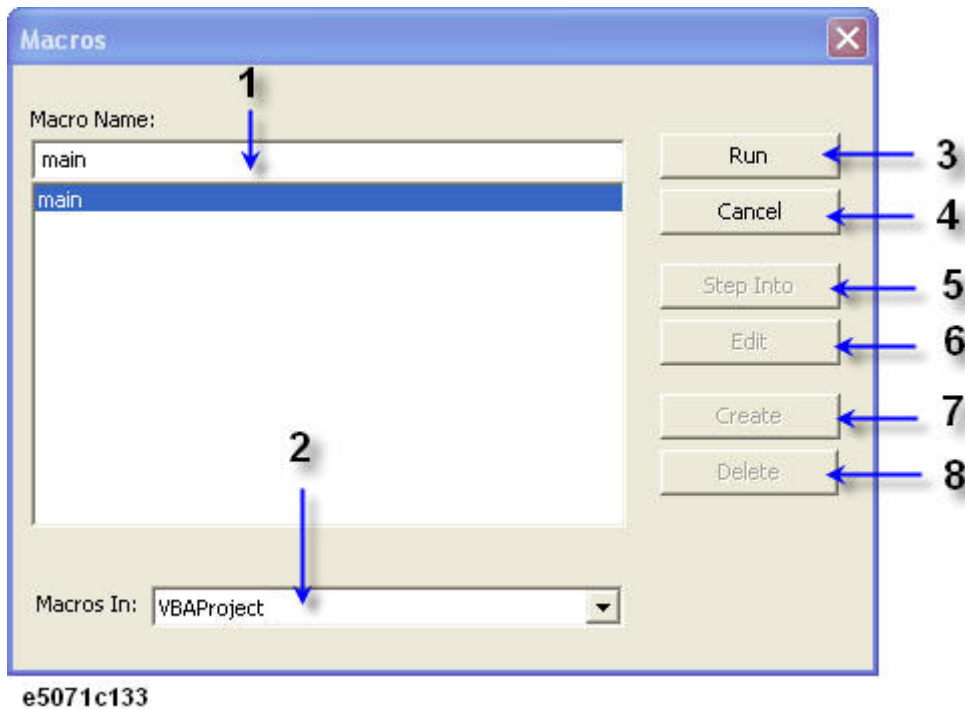
1. Open the Macros dialog box by doing either one of the following:
 - On the **Run** menu, click **Run Macro**.
 - On the **Tools** menu, click **Macros...**
 - On the toolbar, click "Run Macro" icon.
 - Press **F5** key on the keyboard.

NOTE

Doing the above steps with the cursor positioned within a procedure in the code window immediately runs the program without displaying the Macros dialog box.

1. In the Macros dialog box, select the VBA program (procedure name) you want to run, and click the **Run** button.

Macros dialog box



The Macros dialog box has the following user interface elements:

1. **Macro Name:** Select the VBA program (procedure name) you want to run from the list box so its name appears here.
2. **Macro In:** Specify the project that contains the VBA program you want to run. Normally, use the default.
3. **Run:** Clicking this button runs the selected VBA program (procedure).
4. **Cancel:** Clicking this button closes the Macros dialog box and brings you back to the main screen.
5. **Step Into:** Clicking this button brings up Visual Basic Editor and put it into step-in mode, where the selected VBA program is run step by step. This mode is primarily intended for use when debugging a VBA program. For more information on step-in mode, see Debug Toolbar.
6. **Edit:** Displays the code of the selected VBA program. You can use this for re-editing your code.
7. **Create:** This button is normally dimmed.
8. **Delete:** Clicking this button deletes the selected VBA program. Take care not to inadvertently delete your VBA program before saving it.

The Macros dialog provides access to subprograms (procedures enclosed between Sub and End Sub) created in a standard module.

Running a Program from the E5071C Measurement Screen

The E5071C allows you to run a program from E5071C screen using one of the four methods listed below.

1. Display the E5071C measurement screen following the instructions given in Switching to the E5071C Measurement Screen.
2. Run the VBA program (procedure) using the following key sequence:
 - **Macro Setup** > **Select Macro - Module xxx**
 where "**Module**" is the object name (Name property shown in the property window) and "**xxx**" is the procedure name.
 - Press the **Macro Run** key on the E5071C front panel. For a program to be run from the measurement screen, its procedure name must be "Main" (subprogram enclosed between Sub Main() and End Sub), and its object name (Name property as displayed in the property window) must be "Module1".

NOTE

When you are working with the E5071C measurement screen, the E5071C's macro environment only provides access to those VBA programs that are created as subprograms (enclosed between Sub and End Sub) in a standard module.

Loading and Executing Programs in Batch Process

This section describes how to load and execute a program (VBA project) in a batch process by pressing the softkey corresponding to the program name.

1. Save the VBA program (VBA project file) into the following folder.
D:\VBA

NOTE

This feature is available only for programs saved in **D:\VBA**. This feature is not available for programs saved in subfolders of **D:\VBA**.

NOTE

When copying a VBA program to D:\VBA from another folder, copy all the files necessary to execute the program to appropriate folders. When copying a factory-installed VBA program into D:\VBA, choose only its VBA project file.

2. Press **Macro Setup** key.
3. Click **Load & Run**.
4. Press the softkey corresponding to the VBA project file name of the program you want to execute. The pressed VBA project is loaded and the program whose procedure name is set to "Main" (subprogram enclosed between Sub Main() and End Sub) and whose object name (Name property as displayed in the property window) is set to "**Module1**" is executed.

NOTE

There is no limit to the number of VBA project files that can be saved in **D:\VBA**. However, the maximum number of programs that can be displayed as softkeys is 50.

- File names of the VBA projects saved in **D:\VBA** are displayed as softkeys in alphabetical order.
- The maximum number of characters that can be displayed in a softkey is 12. If a file name has 13 or more characters, "..." is added to the 12th character from the beginning of the program name and displayed. In this case a .vba extension is omitted.

Stopping a VBA Program

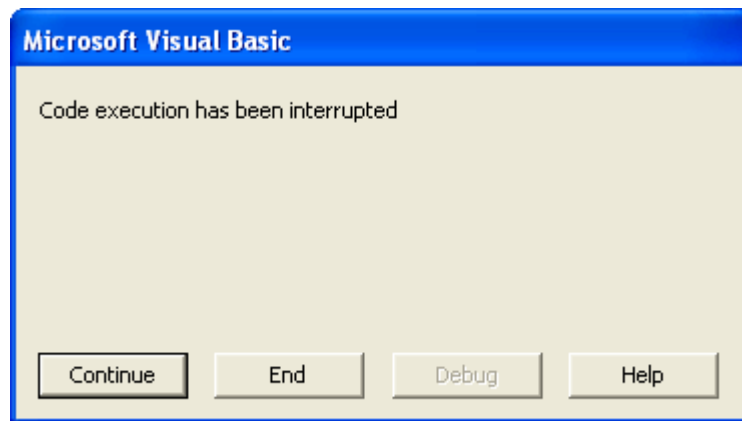
- [Stopping a Procedure](#)
- [Abruptly Terminating a VBA Program](#)

Other topics about Operation Basics

Stopping a Procedure

This section describes how to break a procedure during the execution of a VBA program.

1. To break the running VBA program, do one of the following:
 - On the **Run** menu, click **Break**.
 - On the toolbar, click "Break Macro" icon.
 - Press **Ctrl** + **Break** keys on the keyboard.
 - **Macro Setup** > **Stop** (E5071C measurement screen)
 - Press **Macro Break** key on the E5071C front panel.
2. A dialog box is displayed through forced interrupts, and the program is suspended.



e5071c250

Select one of the following:

- **Continue:** Resumes the execution of the program.
- **End:** Terminates the VBA program.
- **Debug:** Displays a run-time error.
- **Help:** Brings up VBA Online Help.

Abruptly Terminating a VBA Program

This section describes how to abruptly terminate a running procedure. When abruptly terminating the VBA program by the below methods, the "Program interrupted" message is shown in the instrument status bar on the bottom of the LCD display.

To terminate the running VBA program, do one of the following:

- On the **Run** menu, click **Reset**.
- On the toolbar, click "Reset Macro" icon.
- Insert an *End* statement into your code.

Errors and Debugging

- [Type of Errors](#)
- [Using a Debug Tool](#)

Other topics about Operation Basics

Type of Errors

Errors in VBA programs are classified into the following two types:

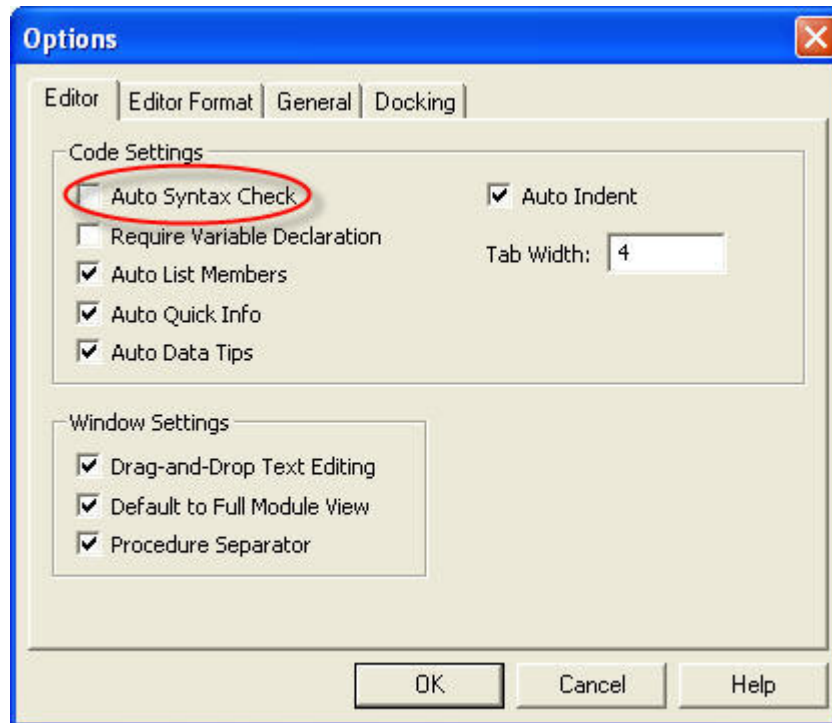
Syntax errors

A syntax error is generated when Visual Basic Editor detects an invalid statement that violates the Visual Basic syntax rules. For example, misspelled keywords generate syntax errors. An error dialog box appears that indicates the error message, and highlight the invalid statement in red. To get detailed information on the error, click the **HELP** button in the error dialog box to display the help topic on the error. You cannot run the macro until you correct the syntax error.

The E5071C VBA environment is by default configured to automatically check for syntax errors, but you can disable the auto syntax check feature using the following steps:

1. On the **Tools** menu, click **Options....**

2. On the **Editor** tab, clear the **Auto Syntax Check** check box.



e5071c121

3. Click the **OK** button.

Run-time Errors

A run-time error is generated when a VBA program attempts to execute an invalid statement at run time. When a run-time error is generated, the program is stopped at the invalid statement, and an error dialog box appears. You can terminate the program by clicking the END button in the error dialog box. Also, you can click the DEBUG button in the error dialog box to identify the statement that caused the error. In this case, the statement in question is highlighted in yellow.

NOTE

Some run-time errors occur under particular conditions, even though a program run without occurring the errors under normal conditions. For example, the "Target value not found" error that occurs when a program that analyzes the results using the Marker Bandwidth Search feature fail to perform bandwidth search because the marker is not in the appropriate position, the "Ecal module not in RF path" error that occurs when a program that performs calibrations using a ECal module fail to measure the calibration data because the ECal module is not appropriately connected to test ports, and so on.

Using a Debug Tool

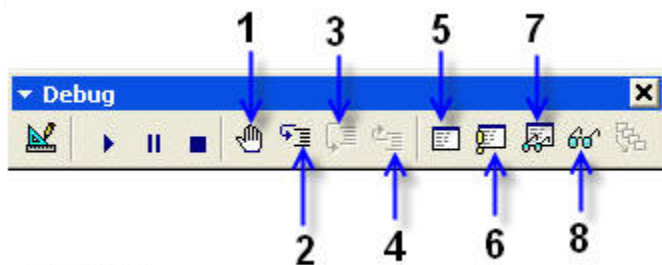
The E5071C's VBA environment provides a variety of debug tools that help you identify logical errors. Detailed information on using the debug tools is covered in VBA Online Help and books on VBA.

Debug Toolbar

The debug toolbar provides tool buttons that allow you to easily access various debug tools.

On the **View** menu, click **Toolbars > Debug**.

Debug toolbar



1. Set/clear break points (keyboard **F9**): Puts a break point at the cursor position or clears an existing break point.
2. Step-in (keyboard: **F8**): Runs the VBA program step by step. If the current program contains a call to another procedure, that procedure is also run step by step.
3. Step-over (keyboard: **Shift + F8**): Runs the VBA program step by step. If the current program contains a call to another procedure, that procedure is run as one line.
4. Step-out (keyboard: **Ctrl + Shift + F8**): Executes the remaining lines of the function where the execution point is currently placed.
5. Local window: Opens the local window that shows the current values of local variables.
6. Immediate window (keyboard: **Ctrl + G**): Opens the immediate window that evaluates entered values of variables or expressions.
7. Watch window: Opens the watch window that displays the current value of a specified expression.
8. Quick window (keyboard: **Shift + F9**): Displays the current value of a specified expression in a dialog box.

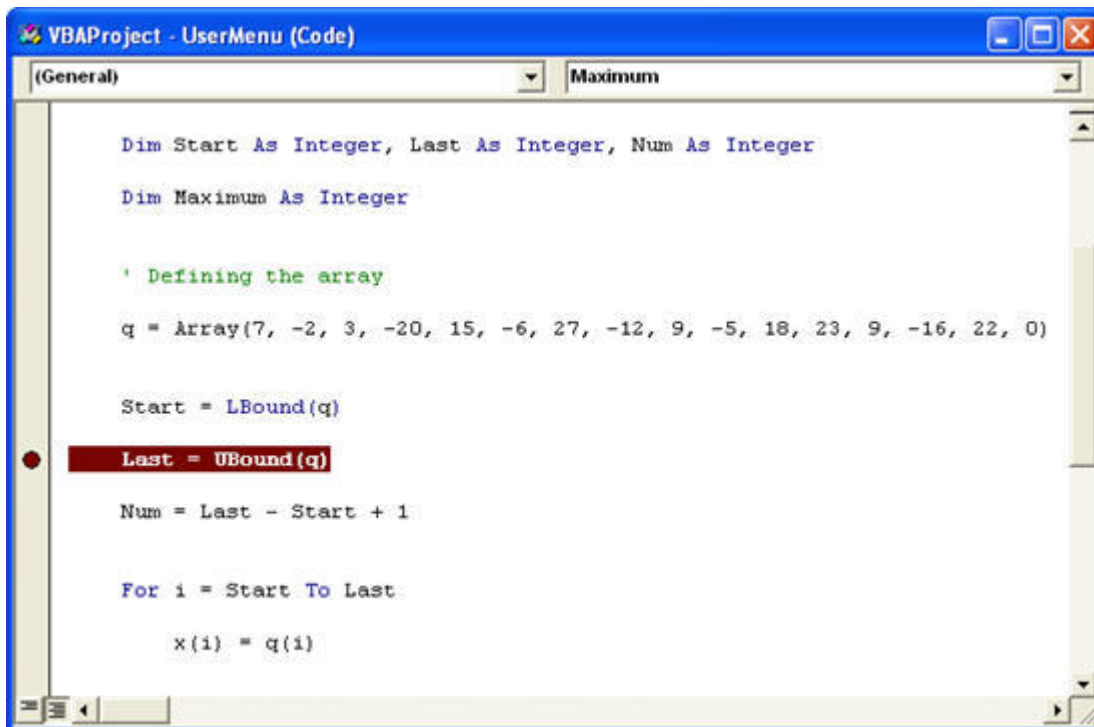
Setting a Break Point

By placing a break point at a particular statement in a VBA program, you can automatically suspend the program when it is executed to that statement.

When you put a break point at a line, the line is highlighted in amber as shown in the following figure. To set a break point, do one of the following:

- Place the cursor at the desired line of code, and click the "Set/clear break points" button on the debug toolbar.
- Click anywhere in the margin indicator bar of the code window.

Setting a break point



e5071c123

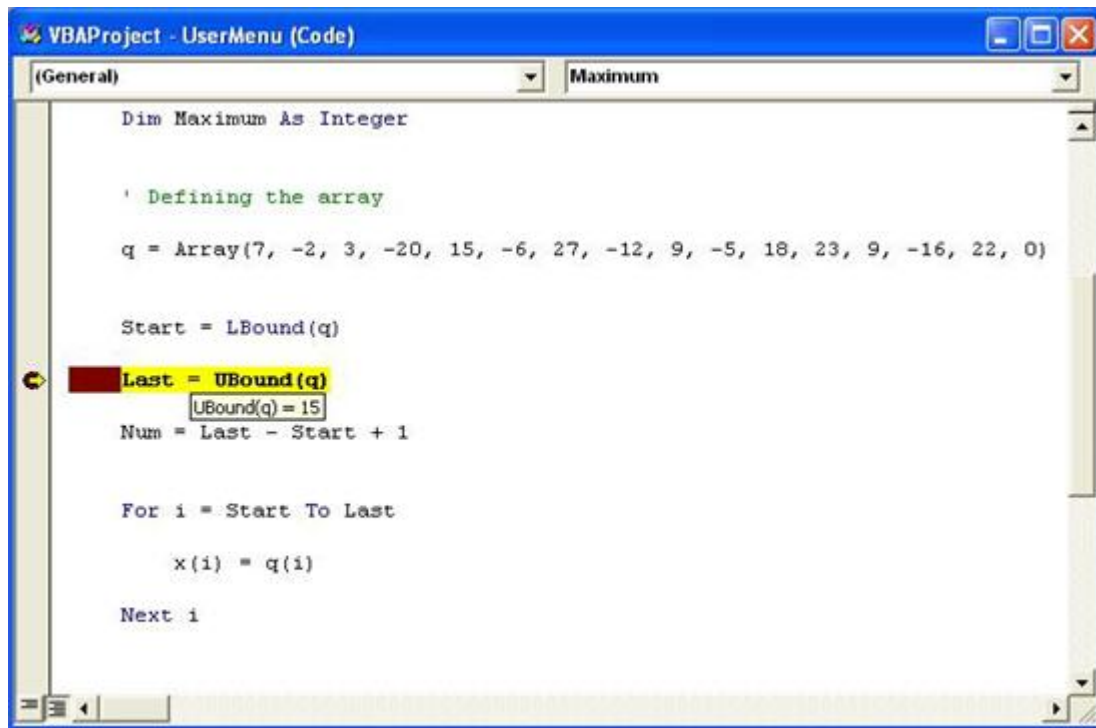
Monitoring Variable or Property Values

With your VBA program suspended, you can use the following debug tool to monitor variables or properties. To do this, you must set a break point, run the VBA program, and suspend it.

Data Hint

When you point to the variable or expression of interest, Data Hint shows the current value as shown in the following figure.

Data Hint



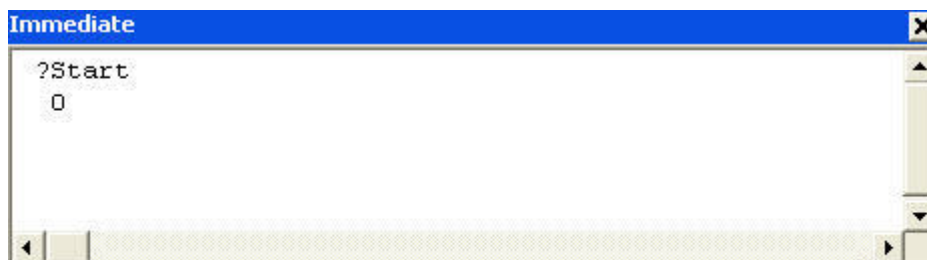
e5071c127

Immediate Window

To display the immediate window, click  on the debug toolbar.

In the immediate window, enter a question mark (?) followed by the variable or expression whose value you want to check, and press the Enter key, as shown in the following figure. The current value appears in the line that follows.

Immediate window

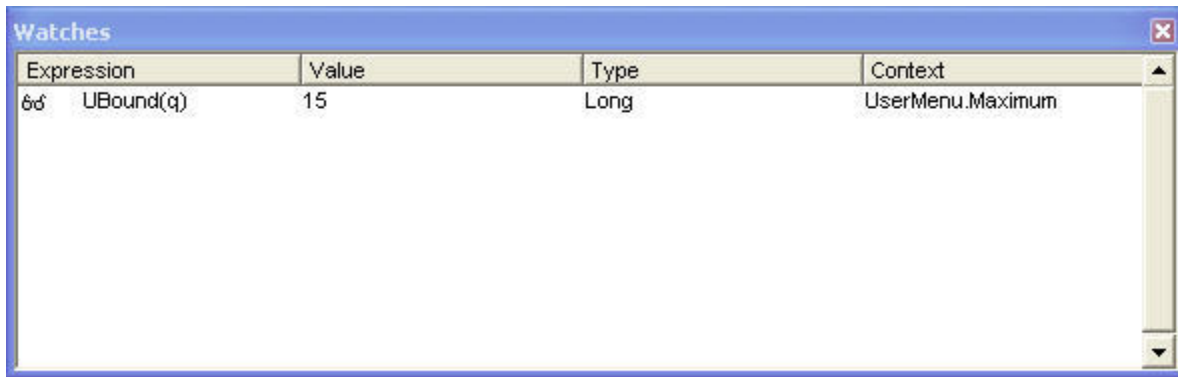


e5071c124

Watch Window

To display the watch window, click the "Watch Window" button on the debug toolbar.

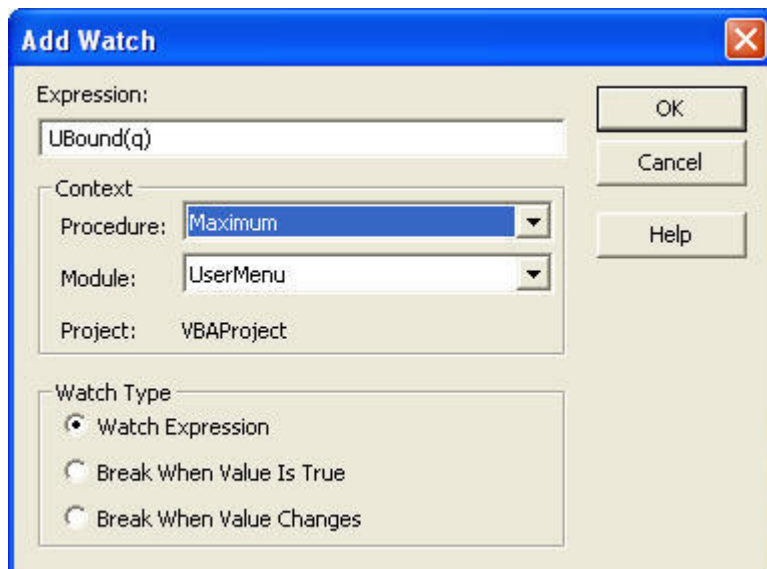
Watch window



e5071c126

1. On the **Debug** menu, click **Add Watch....** to open the Add Watch dialog box.
2. As shown in the following figure, you can specify an expression of interest as a watch expression to always monitor its value.
3. Click the **OK** button.

Add Watch dialog box



e5071c125

Quick Watch

In the code window, select a variable or expression whose value you want to watch. On the debug toolbar, click the "Quick Watch" button to open the Quick Watch dialog box. The dialog box displays the current value of your specified variable or expression.

Also, you can click the **Add** button in the Quick Watch dialog box to specify the current expression as a watch expression.

Printing Output Values in the Echo Window

- [Overview](#)
- [Entering Values Output to Echo Window](#)
- [Opening Echo Window](#)
- [Clearing Values Output from Echo Window](#)

Other topics about Operation Basics

Overview

The echo window, which appears in the lower part of the E5071C measurement screen, can be used to display a message or the return value (data) of an object.

Entering Values Output to Echo Window

You can use the COM objects listed below to enter values output to the echo window.

- ECHO
- SCPI.DISPlay.ECHO.DATA

Opening Echo Window

You can use the COM objects listed below to open the echo window.

- SCPI.DISPlay.TABLe.TYPE
- SCPI.DISPlay.TABLe.STATe

Alternatively, you can also open the echo window using the following key sequence:

Macro Setup > Echo Window (ON)

Clearing Values Output from Echo Window

You can use the COM object shown below to clear values output from the echo window.

- SCPI.DISPlay.ECHO.CLEAr

Alternatively, you can also clear values output from the echo window using the following key sequence:

Macro Setup > Clear Echo

Uses Advanced Techniques

- [Accessing a List of E5071C COM Objects](#)
- [Using Automatic Library References](#)

Other topics about Operation Basics

Accessing a List of E5071C COM Objects

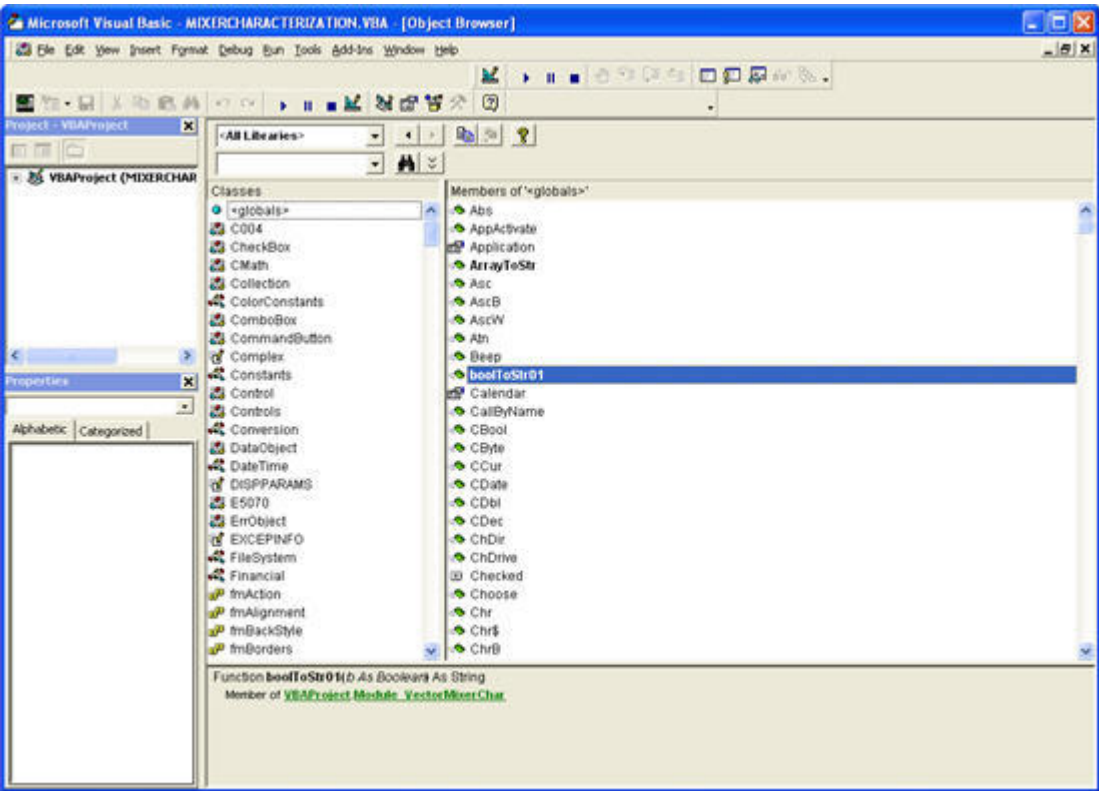
The E5071C VBA environment provides COM objects that support controlling the E5071C. When you are developing a program using E5071C COM objects, you can access a list of E5071C COM objects by opening Object Browser within Visual Basic Editor.

1. To open Object Browser, do one of the following:
 - On the **View** menu, click **Object Browser**.
 - On the toolbar, click "Object Browser" icon.
2. Select **E5071Lib** from the Project/Library box to display the E5071C library as shown in the following figure.

NOTE

There are some COM objects NOT used in controlling with E5071C VBA in the list of the E5071C COM objects displayed on the Object Browser. The COM objects NOT used in controlling with E5071C VBA are not described in the COM object reference.

How to use Object Browser



e5071c134

Using Automatic Library References

For libraries that satisfy the following conditions, the library reference will be automatically set whenever a new project is created and loaded (**Macro Setup > New Project**).

Automatically referenced libraries	Conditions
Directory where the library resides.	D:\Agilent
Extensions of libraries	olb, tlb, dll, or ocx

To check the library reference setting, you must use Visual Basic Editor. Follow these steps to check the library reference setting.

- On the **Tools** menu, click **References...**

NOTE

A project sets the library reference when the project is created. Therefore, if the existing project is loaded, libraries added after the development of the project are not automatically set in the library reference.

Using VBA Online Help

- [Overview](#)
- [Accessing VBA Online Help](#)

Other topics about Operation Basics

Overview

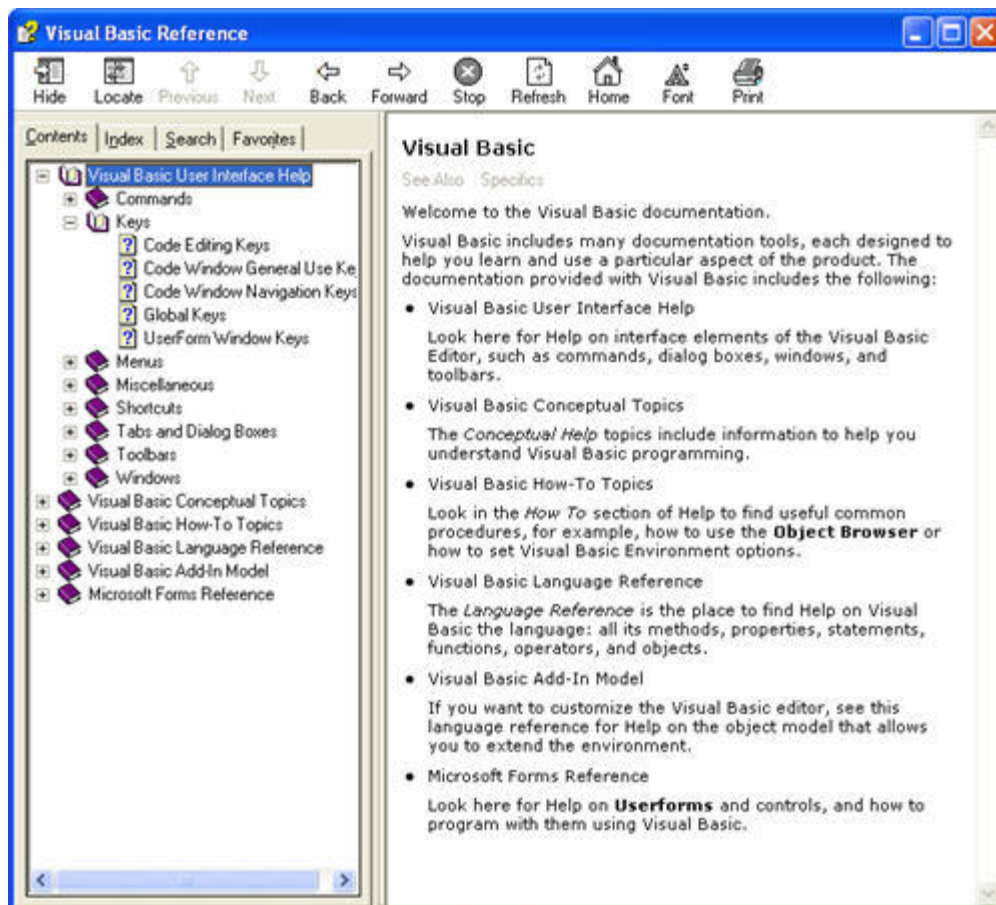
VBA Online Help provides useful topics, such as the VBA terminology or how to use a particular feature. In VBA Online Help, you can find a topic of interest through the Contents or by entering specific keywords.

Accessing VBA Online Help

From Visual Basic Editor, do one of the following to access the VBA Online Help screen.

- On the **Help** menu, click **Microsoft Visual Basic Help**.
- Press **F1** key on the keyboard.
- On the toolbar, click "VBA Help" icon.

VBA Online Help screen



e5071c135

Using the Contents Tab

Clicking the **Contents** tab in the VBA Online Help screen brings up the items listed below. The E5071C VBA Online Help has a hierarchical table of contents. Click an item to expand it, and then find a topic of interest.

- Visual Basic User Interface Help
- Visual Basic Conceptual Topics
- Visual Basic How-To Topics
- Visual Basic Language Reference
- Visual Basic Add-In Model
- Microsoft Forms Reference

When you need information on using Visual Basic Editor, use User Interface Help and How-To Topics as primary sources of information. Format of VBA program is covered in Visual Basic Conceptual Topics. Properties and methods supported by VBA are covered in Visual Basic Language Reference and Visual Basic Add-In Model. Information on using user forms is covered in Microsoft Forms Reference.

Using the Index Tab

In the VBA Online Help screen, click the **Index** tab, and enter a keyword(s) into the text box. For example, you may wish to search for "Sub" or "With" when you are writing your own code.

Looking up a Keyword in the Code within Visual Basic Editor

When you want to know the usage or meaning of a keyword contained in a sample program or some other code, you can quickly access the help topic on that keyword by moving the cursor to the keyword and pressing **F1** key.

Controlling E5071C

Controlling E5071C

- Detecting the End of Measurement
- Reading/Writing Measurement Data
- Executing a Procedure with a Softkey (User Menu Function)

Detecting End of Measurement

- [Overview](#)
- [Using Status Register](#)
- [Using SCPI.TRIGger.SEQuence.SINGle Object](#)

Other topics about Controlling E5071C

Overview

This chapter uses sample programs to demonstrate how to trigger the instrument to start a new measurement cycle and how to detect the end of a measurement cycle. The trigger system is responsible for such tasks as detecting the start of a measurement cycle (triggering) and enabling/disabling measurement on each channel.

You can detect the end of measurement by using either the status register or the SCPI.TRIGger.SEQuence.SINGle object.

Using Status Register

The status of the E5071C can be detected through the status register. If your program is based on SPC commands, you can use SRQ (Service Request) interrupts to detect the end of measurement.

On the other hand, if your program is based on COM objects, SRQ interrupts are not available; instead, you can use the following object to suspend the program until SRQs are generated upon completion of measurement.

- WaitOnSRQ

Sample program is available to download from the Agilent Support page, named **meas_srq.vba**, that demonstrates how to use the status register to suspend the program until the end of measurement. This VBA program consists of the following modules:

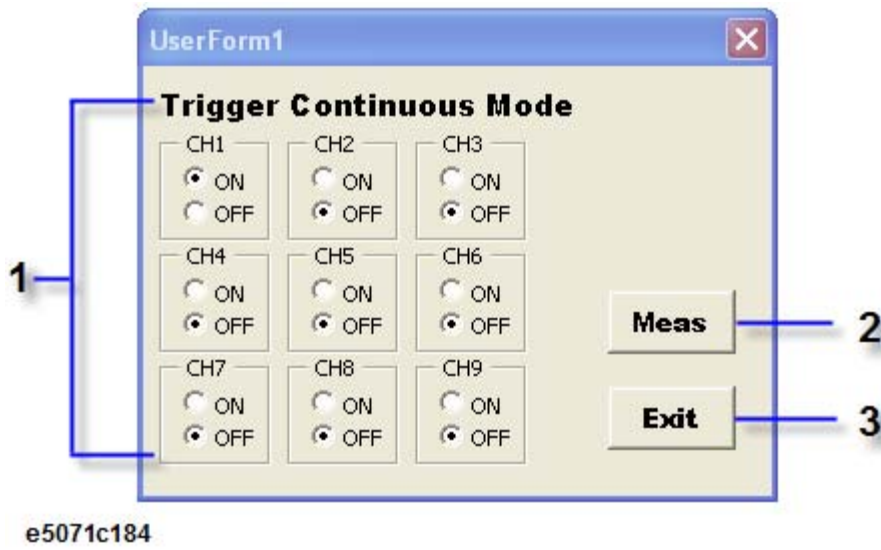
Object name	Module type	Content
frmSrqMeas	UserForm	Uses the status register to wait for the end of measurement.
mdlSrqMeas	Standard module	Invokes a UserForm.

NOTE

This sample program correctly runs when the maximum number of channels/traces is set 9 channels/9 traces.

When you run this VBA program, the following UserForm appears.

The UserForm when running the meas_srqr.vba program



For how to use each element, see the following description.

1. The program turns on/off Continuous Activation mode for each channel and determines whether to enable or disable each channel for measurement.
2. The program triggers the instrument to start a new measurement cycle, waits for the end of measurement, and then displays a message. For detail, see the description of the code window.
3. The program exits, and the UserForm disappears.

In Visual Basic Editor, open the UserForm (object name: frmSrqrMeas), and double-click the **Meas** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Exit** button on the UserForm (lines 10 to 50)

Line 30

Unloads the UserForm from the memory, and terminates the program.

Procedure called when the user clicks the **Meas** button on the UserForm (lines 70 to 390)

Line 110

Hides the UserForm (object name: frmSrqrMeas) from the screen.

Line 130

E5071C

Displays 9 channel windows.

Line 140

Sets the trigger source to "bus".

Lines 160 to 240

These lines turn on or off Continuous Activation mode for each channel depending on whether the corresponding option buttons are on or off. By default, the mode is turned on for channel 1 only.

Lines 260 to 270

These lines configure the instrument so that operation status event register's bit 4 is set to 1 only when operation status condition register's bit 4 is changed from 1 to 0 (negative transition).

Line 280

Enables the operation status event register's bit 4.

Line 290

Enables the status byte register's bit 7.

Line 300

Clears the status byte register and operation status event register.

Line 310

Triggers the instrument to start a measurement cycle.

Line 320

Verifies that the instrument is in a measurement cycle, and suspends the program until the end of measurement. The time-out is set to 100 seconds (maximum value).

Lines 330 to 350

These lines display a measurement completion message upon detecting the end of measurement.

Line 370

Displays the UserForm (object name :frmSrqMeas) on the screen.

Using SRQs to detect the end of measurement (object name: frmSrqMeas)

```

10| Private Sub cmdExit_Click()
20|
30| Unload Me
40|
50| End Sub
60|
70| Private Sub cmdMeas_Click()
80|
90| Dim Cond As Boolean
100|
110| frmSrqMeas.Hide
120|
130| SCPI.DISPlay.Split = "d123_456_789"
140| SCPI.TRIGger.SEQuence.Source = "bus"
150|
160| SCPI.INITiate(1).CONTInuous = optOn1.Value
170| SCPI.INITiate(2).CONTInuous = optOn2.Value
180| SCPI.INITiate(3).CONTInuous = optOn3.Value
190| SCPI.INITiate(4).CONTInuous = optOn4.Value
200| SCPI.INITiate(5).CONTInuous = optOn5.Value
210| SCPI.INITiate(6).CONTInuous = optOn6.Value
220| SCPI.INITiate(7).CONTInuous = optOn7.Value
230| SCPI.INITiate(8).CONTInuous = optOn8.Value
240| SCPI.INITiate(9).CONTInuous = optOn9.Value
250|
260| SCPI.STATus.OPERation.PTRansition = 0
270| SCPI.STATus.OPERation.NTRansition = 16
280| SCPI.STATus.OPERation.ENABLE = 16
290| SCPI.IEEE4882.SRE = 128
300| SCPI.IEEE4882.CLS
310| SCPI.IEEE4882.TRG
320| WaitOnSRQ Cond, 100000
330| If Cond = True Then
340| MsgBox "Measurement Completion"

```

```
350| End If
360|
370| frmSrqMeas.Show
380|
390| End Sub
```

Using the SCPI.TRIGger.SEQuence.SINGle Object

When you trigger the instrument by issuing the SCPI.TRIGger.SEQuence.SINGle object, you can use the SCPI.IEEE4882.OPC object to suspend the program until the end of measurement.

The sample program is available to download from the Agilent Support page, named "meas_sing.vba", that demonstrates how to use the SCPI.TRIGger.SEQuence.SINGle object to suspend the program until the end of measurement. This VBA program consists of the following modules:

Object name	Module type	Content
frmSingMeas	UserForm	Uses the SCPI.TRIGger.SEQuence.SINGle and SCPI.IEEE4882.OPC objects to suspend the program until the end of measurement.
mdlSingMeas	Standard module	Invokes a UserForm.

NOTE This sample program correctly runs when the maximum number of channels/traces is set 9 channels/9 traces.

When you run this VBA program, a Userform appears.

In Visual Basic Editor, open the UserForm (object name:frmSingMeas), and double-click the **Meas** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Exit** button on the UserForm (lines 10 to 50)

Line 30

Unloads the UserForm from the memory, and terminates the program.

Procedure called when the user clicks the **Meas** button on the UserForm (lines 70 to 330)

Line 110

Hides the UserForm (object name: frmSingMeas) from the screen.

Line 130

Displays 9 channel windows.

Line 140

Sets the trigger source to "bus".

Lines 160 to 240

These lines turn on or off Continuous Activation mode for each channel depending on whether the corresponding option buttons are on or off. By default, the mode is turned on for channel 1 only.

Line 260

Triggers the instrument to start a measurement cycle.

Line 270

Executes the SCPI.IEEE4882.OPC object to suspend the program until the value of 1 is returned indicating the end of measurement.

Line 290

Displays a measurement completion message.

Line 310

Displays the UserForm (object name: frmSingMeas) on the screen.

Using the SCPI.TRIGger.SEquence.SINGle object to suspend the program until the end of measurement (object name:frmSingMeas)

```
10| Private Sub cmdExit_Click()
20|
30| Unload Me
40|
50| End Sub
60|
70| Private Sub cmdMeas_Click()
80|
90| Dim Dmy As Long
100|
```

E5071C

```
110| frmSingMeas.Hide
120|
130| SCPI.DISPlay.Split = "d123_456_789"
140| SCPI.TRIGger.SEQuence.Source = "bus"
150|
160| SCPI.INITiate(1).CONTInuous = optOn1.Value
170| SCPI.INITiate(2).CONTInuous = optOn2.Value
180| SCPI.INITiate(3).CONTInuous = optOn3.Value
190| SCPI.INITiate(4).CONTInuous = optOn4.Value
200| SCPI.INITiate(5).CONTInuous = optOn5.Value
210| SCPI.INITiate(6).CONTInuous = optOn6.Value
220| SCPI.INITiate(7).CONTInuous = optOn7.Value
230| SCPI.INITiate(8).CONTInuous = optOn8.Value
240| SCPI.INITiate(9).CONTInuous = optOn9.Value
250|
260| SCPI.TRIGger.SEQuence.SINGle
270| Dmy = SCPI.IEEE4882.OPC
280|
290| MsgBox "Measurement Completion"
300|
310| frmSingMeas.Show
320|
330| End Sub
```


Reading/Writing Measurement Data

- [Overview](#)
- [Sample Program](#)

Other topics about Controlling E5071C

Overview

This section describes how to process the E5071C's internal data. You can use these internal data arrays: corrected data arrays, corrected memory arrays, formatted data arrays, formatted memory arrays, and stimulus data arrays. For more information on the internal data arrays, see Internal Data Processing.

To read/write a formatted data array, formatted memory array, corrected data array, or corrected memory array use the following objects:

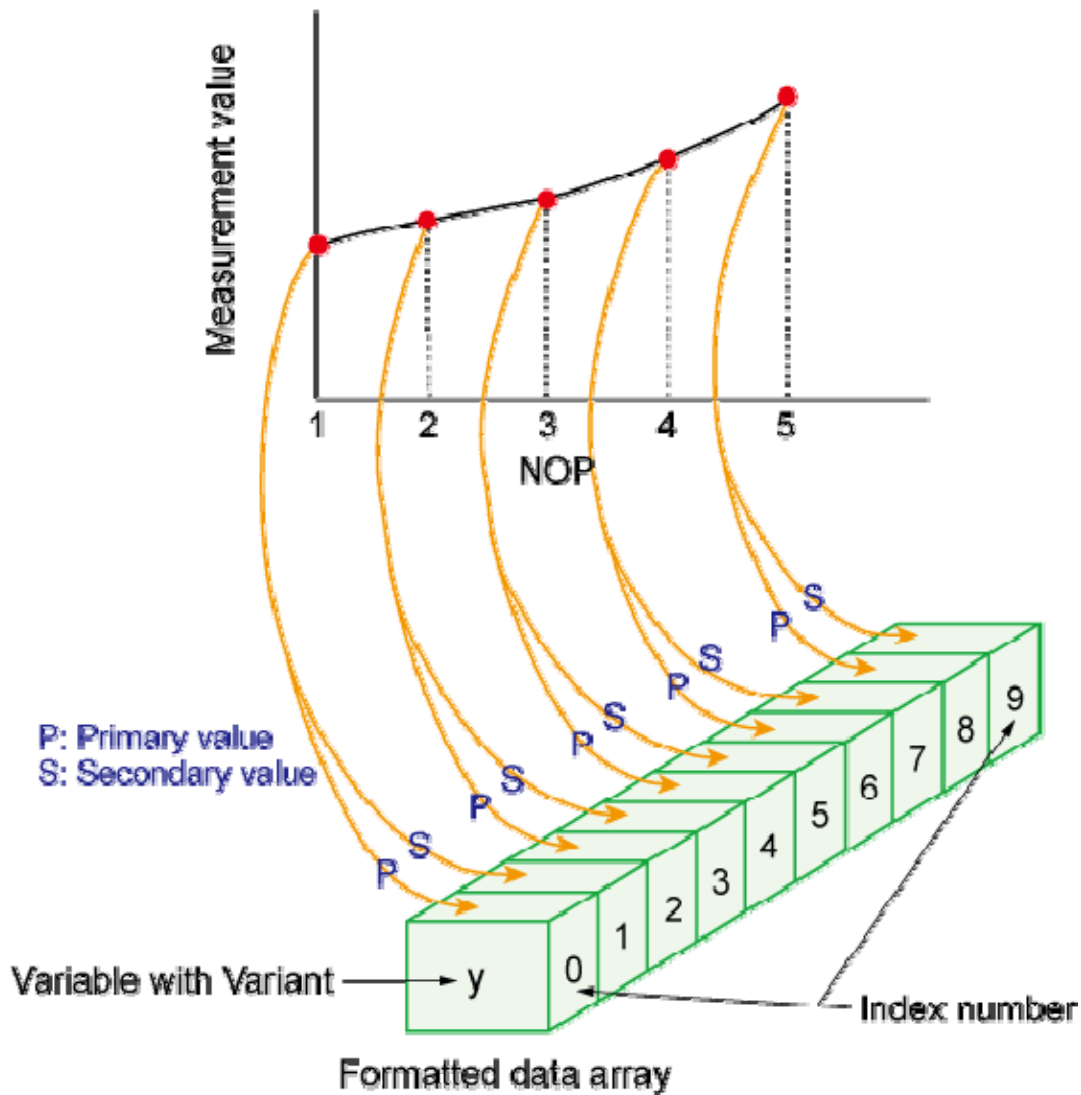
- SCPI.CALCulate(Ch).SElected.DATA.FDATA
- SCPI.CALCulate(Ch).SElected.DATA.FMEMory
- SCPI.CALCulate(Ch).SElected.DATA.SDATA
- SCPI.CALCulate(Ch).SElected.DATA.SMEMory

To read a stimulus data array, use the following objects:

- SCPI.SENSE(Ch).FREQuency.DATA

The E5071C VBA allows you to deal with multiple pieces of data through variables of Variant type. Variant variables can contain any type of data, allowing you to deal with array data without being aware of the number of elements. For example, a formatted data array that includes 5 measurement points is stored as shown in the following figure. Note that a formatted data array always contains 2 data items per measurement point, whichever data format is used. For more information on contained data, see Internal Data Processing. you can find a table that describes the relationship between contained data items and data formats.

Example storing data into a Variant variable



e5071c436

NOTE

When you use one of the objects listed above, the base index number of the array is always 0 even if the declaration section contains the "Option Base 1" statement, which specifies the use of the base array index of 1.

For example, you may wish to read the formatted data array for a particular trace in its entirety (including all measurement points), display the data in the echo window, and then write the data into another trace. How to implement such a process can be better understood with the aid of a sample program.

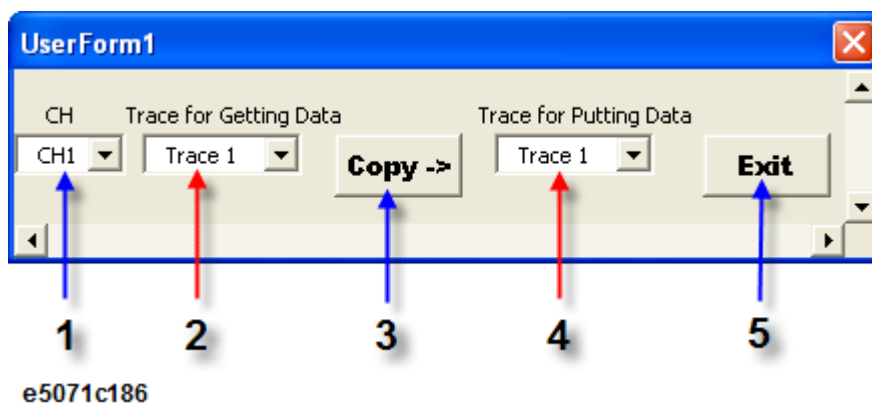
Sample program is available to download from the Agilent Support page, named "**read_write.vba**", that demonstrates how to read and write measurement data. This VBA program consists of the following modules:

Object name	Module type	Content
frmReadWrite	UserForm	Reads, displays, and writes a formatted data array.
mdlReadWrite	Standard module	Invokes a UserForm.

NOTE

This sample program correctly runs when the maximum number of channels/traces is set 9 channels/9 traces.

When you run this VBA program, a following window appears.

UserForm of read_write.vba program***The program lets the user specify the channel to be controlled.***

1. The program lets the user specify which trace's formatted data array to read (source trace).
2. The program reads the formatted data array for the trace specified by the user, display the measurement results in the echo window, and write the data into the trace specified by the user. For detail, see the description of the code window.
3. The program lets the user specify which trace's formatted data array to overwrite (target trace).
4. The program exits, and the window disappears.

In Visual Basic Editor, open the UserForm (object name: frmReadWrite), and double-click the entire UserForm or the **Copy ->** or **Exit** button to bring

E5071C

up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Copy** button on the UserForm (lines 10 to 520)

Lines 90 to 160

These lines identify the selected items in each list and store them into the variables TrGet, TrPut, and ActCh.

Lines 180 to 210

If the specified target trace is not displayed, these lines display that trace.

Lines 230 to 250

These lines make active the specified trace (TrGet: source trace) in the specified channel(ActCh) and hold the sweep.

Line 260

Reads the number of measurement points for the specified channel (ActCh) and stores that number into the Nop variable.

Line 280

Reads the formatted data array for the active trace (source trace) and store the data into the FmtData variable.

Line 290

Reads the stimulus array for the specified channel (ActCh) and stores the data into the Freq variable.

Line 330

Reads the data format for the active trace (source trace) and store it into the Fmt variable.

Lines 340 to 350

These lines display the echo window in the lower part of the LCD screen.

Lines 360 to 470

The lines display, in the echo window, each point along with one measured value (the odd part of the index is always 0) and a frequency if the Fmt is "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL", "IMAG", or "UPH"; or

along with two measured values and a frequency if Fmt\$ returns any other string.

Line 490

Makes active the specified trace (TrPut: target trace) in the specified channel(ActCh).

Line 500

Writes the formatted data array (FmtData) into the active trace (target trace).

Procedure called when the user clicks the **Exit** button on the UserForm (lines 540 to 580)

Line 560

Unloads the UserForm from the memory, and terminates the program.

Procedure that initializes the UserForm (lines 600 to 1020)

Lines 620 to 1000

When the program is launched, these lines add each list item and set the default value for each list.

Sample Program

Reading/displaying/writing a formatted data array (read_write.frm)

```

10| Private Sub cmdCopy_Click()
20|
30| Dim X As Integer, Y As Integer, Z As Integer, I As Integer
40| Dim ActCh As Long, TrGet As Long, TrPut As Long
50| Dim TrCont As Long, Nop As Long
60| Dim FmtData As Variant, Freq As Variant
70| Dim Fmt As String
80|
90| X = cboCh.ListIndex
100| ActCh = X + 1
110|
120| Y = cboGet.ListIndex
130| TrGet = Y + 1
140|
150| Z = cboPut.ListIndex

```

E5071C

```
160| TrPut = Z + 1
170|
180| TrCont = SCPI.CALCulate(ActCh).PARAmeter.Count
190| If TrCont < TrPut Then
200| SCPI.CALCulate(ActCh).PARAmeter.Count = TrPut
210| End If
220|
230| SCPI.CALCulate(ActCh).PARAmeter(TrGet).SElect
240| SCPI.INITiate(ActCh).CONTinuous = False
250| SCPI.ABORt
260| Nop = SCPI.SENSE(ActCh).SWEep.POINts
270|
280| FmtData = SCPI.CALCulate(ActCh).SElected.Data.FDATa
290| Freq = SCPI.SENSE(ActCh).FREQuency.Data
300|
310| ""Displays the formatted data
320|
330| Fmt = SCPI.CALCulate(ActCh).SElected.Format
340| SCPI.DISPlay.TABLe.TYPE = "ECHO"
350| SCPI.DISPlay.TABLe.STATe = True
360| Select Case Fmt
370| Case "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL", "IMAG", "UPH"
380| ECHO "Nop", "Frequency(GHz)", "Data"
390| For I = 0 To Nop - 1
400| ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I)
410| Next I
420| Case Else
430| ECHO "Nop", "Frequency(GHz)", "Data1", "Data2"
440| For I = 0 To Nop - 1
450| ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I), FmtData(2 * I + 1)
460| Next I
470| End Select
480|
490| SCPI.CALCulate(ActCh).PARAmeter(TrPut).SElect
500| SCPI.CALCulate(ActCh).SElected.Data.FDATa = FmtData
510|
520| End Sub
```

```

530|
540| Private Sub cmdExit_Click()
550|
560| Unload Me
570|
580| End Sub
590|
600| Private Sub UserForm_Initialize()
610|
620| With cboCh
630| .AddItem "CH1"
640| .AddItem "CH2"
650| .AddItem "CH3"
660| .AddItem "CH4"
670| .AddItem "CH5"
680| .AddItem "CH6"
690| .AddItem "CH7"
700| .AddItem "CH8"
710| .AddItem "CH9"
720| End With
730|
740| With cboGet
750| .AddItem "Trace 1"
760| .AddItem "Trace 2"
770| .AddItem "Trace 3"
780| .AddItem "Trace 4"
790| .AddItem "Trace 5"
800| .AddItem "Trace 6"
810| .AddItem "Trace 7"
820| .AddItem "Trace 8"
830| .AddItem "Trace 9"
840| End With
850|
860| With cboPut
870| .AddItem "Trace 1"
880| .AddItem "Trace 2"
890| .AddItem "Trace 3"

```

E5071C

```
900| .AddItem "Trace 4"  
910| .AddItem "Trace 5"  
920| .AddItem "Trace 6"  
930| .AddItem "Trace 7"  
940| .AddItem "Trace 8"  
950| .AddItem "Trace 9"  
960| End With  
970|  
980| cboCh.ListIndex = 0  
990| cboGet.ListIndex = 0  
1000| cboPut.ListIndex = 0  
1010|  
1020| End Sub
```


Executing a Procedure with a Softkey (User Menu Function)

- [Overview](#)
- [Preparing for User Menu Function](#)
- [Using User Menu Function](#)
- [Sample Program](#)

Other topics about Controlling E5071C

Overview

The E5071C lets you perform procedures assigned to specific softkeys (**Macro Setup > User Menu > Button 1/2/3/4/5/6/7/8/9/10**), without using user forms, when that softkey is pressed. This function is called the user menu function.

NOTE


You do not have to execute any VBA program when using the user menu function.

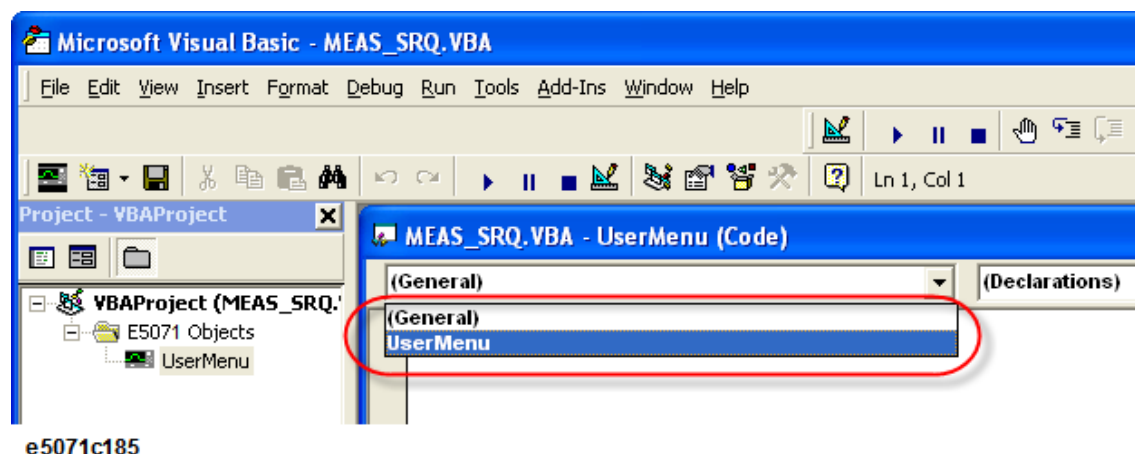
Preparing for User Menu Function

Before using the user menu function, perform the following preparation.

Coding of a Procedure Assigned to a Softkey

Follow these steps to create a procedure assigned to a specific softkey in the "UserMenu" object in the "E5071 Objects" folder.

1. Double-click  **UserMenu** to open the code window.
2. In the object box in the code window, select **UserMenu** as shown below:



3. In the UserMenu_OnPress(ByVal Key_id As Long) procedure, create a program you want to assign to a specific softkey (specify with the *id*

variable). For actual use example, see Line 70 to 430 in the Sample program using user menu UserMenu object.

4. During processing an event (during execution of a procedure for a key pressed), another event (an interrupt by a procedure for another softkey pressed) cannot be accepted.
5. You cannot save (export) the "UserMenu" object by module basis; save it by project basis.

Settings for Softkey Label and Softkey Enabled/Disabled

When you want to change the softkey labels for the user menu function, use the following COM object.

- [UserMenu.Item\(Key_id\).Caption](#)

When you want to set the softkey enabled/disabled for the user menu function, use the following COM object.

- [UserMenu.Item\(Key_id\).Enabled](#)

Moreover, when you want to preset the above settings for the user menu function, use the following COM object.

- [UserMenu.PRESet](#)

NOTE

The above user menu setting is also preset by pressing **Macro Setup** > **Preset User Menu** on the E5071C front panel.

Using User Menu Function

To execute a procedure assigned to a softkey, you need to generate an event of pressing the softkey. To generate an event, the manual method and the COM object method are available.

Method by Manual Operation

Click the specific softkey as follows:

Macro Setup > **User Menu** > **Button <No>**.

"No." represents a button number. You can set the label for "**Button No.**" as you like. For detail, refer to the "Settings for Softkey Label and Softkey Enabled/Disabled." section.

Method by COM Object

You can use the following COM object to perform the same operation as pressing a specific softkey.

- UserMenu.Press(Key_id)

Sample Program

Sample program is available to download from the Agilent Support page, named **meas_user.vba**, that demonstrates how to use the user menu function. This VBA program consists of the following standard module and the "UserMenu" object.

Object name	Module type	Content
mdlUserMenu	Standard module	Sets the softkey labels and enables interrupts from the softkeys.

The program (object name: mdlUserMenu) is described in detail below:

Line 70

Stores True into the State variable.

Lines 90 to 150

Sets the first to third softkey (*id*: 1 to 3) enabled, and sets the fourth to tenth softkey (*id*: 4 to 10) disabled.

Lines 170 to 190

Sets the first softkey label (*id*: 1) to "Setup" the second softkey label (*id*: 2) to "Meas" the third softkey label (*id*: 3) to "Exit".

Line 210

Displays the buttons for the user menu function in the softkey area.

Lines 230 to 250

Processing repeated until the State variable is True (State = True).

Line 240

Detects an event that a specific softkey is pressed and enables the interrupt from the event.

Sample program using user menu (object name: mdlUserMenu)

```

10| Public State As Boolean
20|
30| Sub Main()
40|
50| Dim I As Long, J As Long
60|

```

E5071C

```
70| State = True
80|
90| For I = 1 To 3
100| UserMenu.Item(I).Enabled = True
110| Next I
120|
130| For J = 4 To 10
140| UserMenu.Item(J).Enabled = False
150| Next J
160|
170| UserMenu.Item(1).Caption = "Setup"
180| UserMenu.Item(2).Caption = "Meas"
190| UserMenu.Item(3).Caption = "Exit"
200|
210| UserMenu.Show
220|
230| Do While State
240| DoEvents
250| Loop
260|
270| End Sub
```

UserMenu object

The procedures of the "UserMenu" object are described below.

Lines 70 to 190

The procedure when the first softkey (*id*: 1) is pressed.

Line 90: Returns the E5071C to the preset state.

Lines 110 to 130 For channel 1, sets the sweep start value to 1.73 GHz, the sweep stop value to 1.83 GHz, and the number of measurement points to 51.

Lines 150 to 170 After aborting the measurement, sets the trigger source to the bus trigger and turns on the continuous trigger startup mode for channel 1.

Line 190: Displays the buttons for the user menu function in the softkey area.

Lines 210 to 320

The procedure when the second softkey (*id*: 2) is pressed.

1400

Lines 230 to 240 Generates a trigger to start a single sweep and waits until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Line 260: Retrieves the number of points in channel 1 and stores that number into the Nop variable.

Lines 280 to 290 Specifies trace 1 of channel 1 to the active trace, retrieves the formatted data array, and stores the data into the FmtData variable.

Lines 310 to 320 Displays the echo window in the lower part of the LCD screen.

Lines 340 to 360: Displays 2 measurement data values (primary value and secondary value) for each measurement point in the echo window.

Lines 380 to 430

The procedure when the third softkey (*id*: 3) is pressed.

Line 400: Displays a program closing message.

Line 410: Stores False into the sta variable to terminate the main program.

Sample program using user menu ("UserMenu" object)

```

10| Private Sub UserMenu_OnPress(ByVal id As Long)
20|
30| Dim I As Integer
40| Dim Nop As Long, Dmy As Long
50| Dim FmtData As Variant
60|
70| If id = 1 Then
80|
90| SCPI.SYSTem.PRESet
100|
110| SCPI.SENSE(1).FREQuency.STARt = 1730000000#
120| SCPI.SENSE(1).FREQuency.STOP = 1830000000#
130| SCPI.SENSE(1).SWEep.POINts = 51
140|
150| SCPI.ABORt
160| SCPI.TRIGger.SEQuence.Source = "BUS"
170| SCPI.INITiate(1).CONTInuous = True
180|

```

E5071C

```
190| UserMenu.Show
200|
210| Elself id = 2 Then
220|
230| SCPI.TRIGger.SEQuence.SINGle
240| Dmy = SCPI.IEEE4882.OPC
250|
260| Nop = SCPI.SENSE(1).SWEep.POINts
270|
280| SCPI.CALCulate(1).PARAmeter(1).SElect
290| FmtData = SCPI.CALCulate(1).SElected.DATA.FDATa
300|
310| SCPI.DISPlay.TABLe.TYPE = "ECHO"
320| SCPI.DISPlay.TABLe.State = True
330|
340| For I = 1 To Nop - 1
350| ECHO FmtData(2 * I - 2), FmtData(2 * I - 1)
360| Next I
370|
380| Elself id = 3 Then
390|
400| MsgBox "Program ended!"
410| State = False
420|
430| End If
440|
450| End Sub
```

User Defined Variable

The E5071C is having an area in which a User can set any value. These area are divided by the different data format of the values. A maximum of ten (1 to 10) such areas can be used by each command.

For example, after setting the value (data) obtained using VBA of the E5071C to the User defined variable, this value is available to an external controller (program) through the use of these User defined variables.

NOTE

Turning E5071C power ON/OFF initializes the User defined variables. They are not initialized through executing Preset.

These commands does not refers to or change the results of the E5071C.

- SCPI.PROGram.VARiable.ARRay(Vnum).DATA
- SCPI.PROGram.VARiable.ARRay(Vnum).SIZE
- SCPI.PROGram.VARiable.DOUBle(Vnum).DATA
- SCPI.PROGram.VARiable.LONG(Vnum).DATA
- SCPI.PROGram.VARiable.STRing(Vnum).DATA

E5071C

Controlling Peripherals

Controlling Peripherals

- Overview
- Programming with VISA

Overview

- [Overview](#)
- [Preparation](#)

Other topics about Controlling Peripherals

Overview

The E5071C macro function (E5071C VBA) can be used not only to automate measurements but also to control external measurement instruments connected via USB/GPIB interface by acting as a self-contained system controller (see An Overview of a Control System Based on the Macro Function).

The E5071C macro function (E5071C VBA) performs communications via the COM interface when controlling the E5071C itself, but it communicates via VISA (Virtual Instrument Software Architecture) when controlling external measurement instruments.

Preparation

Importing Definition Files

To use the VISA library in the E5071C macro (E5071C VBA), you need to import two definition files into your project with the Visual Basic editor to define the VISA functions and perform other tasks. The definition files are stored on the sample programs disk under the following filenames (for information on importing modules, refer to Saving a Module (Exporting)).

- **visa32.bas**
- **vpptype.bas**

Programming with VISA

- [Overview](#)
- [Starting VISA](#)
- [Connection](#)
- [Communication](#)
- [Disconnection](#)
- [Sample Program](#)

Other topics about Controlling Peripherals

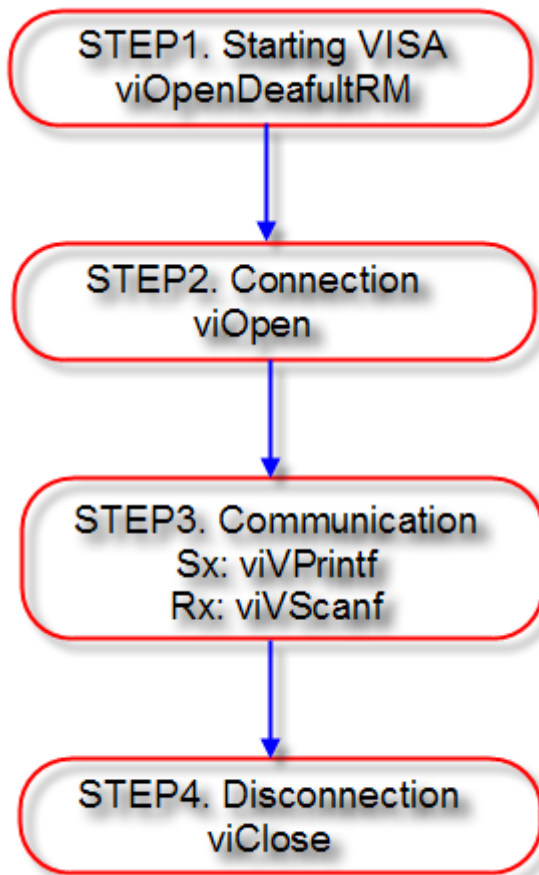
Overview

The following figure shows the flow of controlling the instrument with VISA. When developing a VISA program in the Visual Basic language, a special programming notice (in the readme text file listed below) must be reviewed.

For details on the use of the VISA library and the programming notice for using the VISA library with the E5071C macro (E5071C VBA), refer to the following files contained in IO library CD-ROM.

- **visa.hlp** (on-line help for the VISA library)
- **vbreadme.txt** (notes on using the VISA library with VB)

Flow of instrument control with VISA



e5071c183

STEP 1. Starting VISA

The VISA system startup session is processed in Line 90 in the sample program **ctrl_ext.vba**. VISA's **viOpenDefaultRM** function initializes and starts up the VISA system. The **viOpenDefaultRM** function must be executed before other VISA functions are called, and the parameter of this function is startup information (Defrm in **ctrl_ext.vba**).

Syntax

viOpenDefaultRM(*param*)

Parameter

Parameter	(<i>param</i>)
Description	Startup information (output)
Data type	Long integer type

STEP 2. Connection

The connection session is handled in Line 130. VISA's **viOpen** function makes connection with the specified instrument. The **viOpen** function returns a value so that the VISA functions can apply it to the specified

instrument. The parameters of this function are startup information (Defrm), the address information of the specified instrument ("GPIB0::17::INSTR" in **ctrl_ext.vba**), access mode (0 in **ctrl_ext.vba**), timeout (0 in **ctrl_ext.vba**), and connection information (Equip in **ctrl_ext.vba**).

Syntax

viOpen(*param1*, *param2*, *param3*, *param4*, *param5*)

Parameters

Parameter	(<i>param1</i>)
Description	Startup information (input)
Data type	Long integer type

Parameter	(<i>param2</i>)
Description	Address information of the specified instrument (input)
Data type	Character string type
Syntax	"GPIB0:: <i>gpiib address</i> ::INSTR" "USB0:: <i>manufacturer ID</i> :: <i>model code</i> :: <i>serial number</i> ::0::INSTR" (ex. "USB0::2391::2312::MY12345678::0::INSTR") "TCPIP0:: <i>IP address</i> ::inst0::INSTR"

Parameter	(<i>param3</i>)
Description	Access mode (Enter 0)

Parameter	(<i>param4</i>)
Description	Timeout (Enter 0)

Parameter	(<i>param5</i>)
Description	Connection information (output)

Data type	Long integer type
------------------	-------------------

STEP 3. Communication

The communication session is conducted in Line 170. VISA's [viVPrintf](#) function sends a program message (GPIB command) to the specified instrument. The parameters of this function are connection information (Equip), the program message (*IDN?), and the variable to be formatted (0 in **ctrl_ext.vba**).

- To input/output GPIB commands, the [viVPrintf](#) function and the [viVScanf](#) function are mainly used, but other VISA functions are also available. For more information, refer to **visa.hlp** (online help for the VISA library).

Syntax

`viVPrintf(param1, param2, param3)`

Parameters

Parameter	(<i>param1</i>)
Description	Connection information (input)
Data type	Long integer type

Parameter	(<i>param2</i>)
Description	Program message (input) When sending a program message of the GPIB command, a message terminator is required at the end of the message (Chr\$(10) in ctrl_ext.vba)
Data type	Character string type

Parameter	(<i>param3</i>)
Description	A variable to be formatted. If not applicable, enter 0.
Data type	Specified data type

The receiving session is controlled in Line 210. VISA's [viVScanf](#) function receives the result from the specified instrument and stores it in the output variable. The parameters of this function are connection information (Equip in **ctrl_ext.vba**), the format parameter for the output variable (%t in **ctrl_ext.vba**), and the output variable (Prod in **ctrl_ext.vba**).

Syntax

`viVScanf(param1, param2, param3)`

Parameters

Parameter	(<i>param1</i>)
Description	Connection information (input)
Data type	Long integer type

Parameter	(<i>param2</i>)
Description	Format parameter for the output variable
Data type	Character string type

Parameter	(<i>param3</i>)
Description	Output variable (output)
Data type	Character string type

STEP 4. Disconnection

The disconnection session is handled in Line 280. VISA's [viClose](#) function disconnects communication and terminates the VISA system. The parameter of this function is startup information (Defrm in **ctrl_ext.vba**).

Syntax

`viClose(param)`

Parameter

Parameter	(<i>param</i>)
Description	Startup information (input)
Data type	Long integer type

Sample Program to Read Out the Product Information of Peripheral (Instrument)

The **ctrl_ext.vba** is a sample program to control instruments connected through USB/GPIB interface cable using the E5071C as the system controller. This VBA program consists of the following modules.

Object name	Module type	Content
mdlVisa	Standard module	Reads out the product information of external instrument.
VISA32.bas	Standard module	Definition files to use VISA library

NOTE

When you control peripherals from E5071C VBA, use the GPIB commands provided for the instrument to communicate over VISA. On the other hand, when you control the E5071C itself from E5071C VBA, use the COM objects provided for the E5071C to communicate.

Lines 90 to 100

Initializes and starts up the VISA system and outputs the startup information to the Defrm variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Lines 130 to 140

Establishes the connection to the external instrument (GPIB address: 17) connected via GPIB and outputs the connection information to the Equip variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Lines 170 to 180

Queries the product information of the external instrument connected via USB/GPIB interface cable using VISA. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Lines 210 to 250

Retrieves the product information through VISA and outputs it into the Prod variable. Displays the read-out result in the message box. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Line 280

Breaks the communication and terminates the VISA system.

Lines 320 to 360

If an error occurs in a VISA function, displays the detail of the error and terminates the program.

Read out the product information (ctrl_ext.vba)

```
10| Sub Main()
20|
30| Dim status As Long 'VISA function status return code
40| Dim Defrm As Long 'Session to Default Resource Manager
50| Dim Equip As Long 'Session to instrument
60| Dim Prod As String * 100 'String to receive the result
70|
80| ' Initializes the VISA system.
90| status = viOpenDefaultRM(Defrm)
100| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
110|
120| ' Opens the session to the specified instrument.
130| status = viOpen(Defrm, "GPIB0::17::INSTR", 0, 0, Equip)
140| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
150|
160| ' Asks for the instrument's product information.
170| status = viVPrintf(Equip, "*IDN?" & Chr$(10), 0)
180| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
190|
200| ' Reads the result.
210| status = viVScanf(Equip, "%t", Prod)
220| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
230|
240| ' Displays the result.
250| MsgBox Prod
260|
270| ' Closes the resource manager session (which closes everything)
280| Call viClose(Defrm)
290|
300| GoTo Prog_end
310|
320| VisaErrorHandler:
330| Dim VisaErr As String * 200
340| Call viStatusDesc(Defrm, status, VisaErr)
```



```
350| MsgBox "Error : " & VisaErr, vbExclamation
360| Exit Sub
370|
380| Prog_end:
390|
400| End Sub
```

Application Programs

Application Programs

- Basic Measurement (measuring a band-pass filter)
- Measuring a Multi-port Device
- Measurement Using E5091A (measuring FEM)
- Executing Power Calibration
- Connecting Hard Disk of External PC (shared folder)

Basic Measurement (measuring a band-pass filter)

- [Overview](#)
- [Overview of the Program](#)
- [Description of the Program](#)

Other topics about Application Programs

Overview

The **apl_bsc.vba** shows a sample program (VBA program) that demonstrates how to perform the basic measurement of the band-pass filter. This VBA program consists of the following standard module.

Object name	Module type	Content
mdlBscMeas	Standard module	Performs basic measurement of band-pass filter

Overview of the Program

The sample program performs full 2-port calibration using the 85032F calibration kit, measures a band-pass filter (center frequency: 947.5 MHz), and calculates and displays its bandwidth, insertion loss, and so on. This measurement is the same as Measurement Example of a Bandpass Filter in the Quick Start.

Description of the Program

When you run this VBA program, reset is performed, the measurement conditions are automatically set, and the message "Perform the full 2-port calibration" is displayed. To perform the full 2-port calibration, click **Yes**; otherwise click **No**.

To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port, and then click **OK** to measure the calibration data. Click **Cancel** to return to the beginning of the calibration. You cannot skip the isolation calibration. When the calibration data measurement for all standards is complete, the message "All calibration data completion" is displayed, and the calibration coefficient is calculated.

NOTE

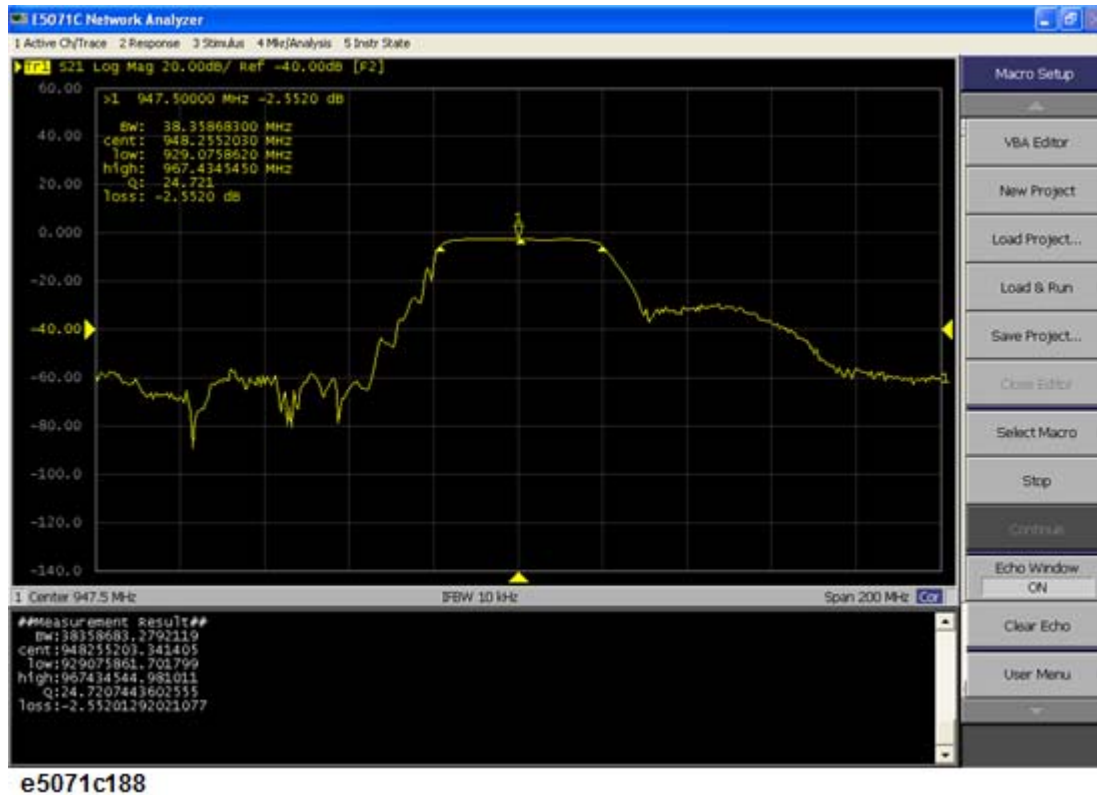
When you cancel the calibration data measurement before completing the measurement of necessary calibration data, the settings condition may not be returned to its former state.

Then, the message "Connect DUT, and then press **Macro Setup** > **Continue**" is displayed in the instrument status bar in the lower part of the LCD display. Connect a DUT and perform **Macro Setup** > **Continue**. After the

E5071C

measurement, the search result is displayed in the echo window, as shown below. If no bandwidth search target is found, only the result of the insertion loss obtained with the marker is displayed.

Example of display after executing the program in "apl_bsc.vba"



The basic measurement program (object name: mdlBscMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Lines 120 to 160

Store the sweep center value (947.5 MHz), the sweep span value (200 MHz), the number of measurement points (401), the IF bandwidth (10 kHz), and the power level (-10 dBm) into the variables Center, Span, Nop, IfBw, and Pow, respectively.

Lines 170 to 210

Store the number of traces (1), the measurement parameter (S21), the data format (log amplitude), the calibration kit number (4: 85032F), and the save file name (State08.sta) into the variables, NumTrac, Par, Fmt, CalKit, and File, respectively.

Line 250

Returns the E5071C to the preset state.

Lines 290 to 300

For channel 1, turn on the continuous trigger startup mode to On and set the trigger source to the bus trigger.

Lines 320 to 360

For channel 1, set the sweep center value to the Center variable, the sweep span value to the Span variable, the number of measurement points to the Nop variable, the IF bandwidth to the IfBw variable, and the power level to the Pow variable.

Lines 380 to 410

For channel 1, set the number of traces to the NumTrac variable, the measurement parameter to the Par variable, and the data format to the Fmt variable.

Line 450

Stores the calibration kit number for channel 1 into the CalKit variable.

Line 460

Stores 1 and 2 into the Port variable that indicates ports used for the full 2-port calibration.

Line 480

Calls the Calib_Solt procedure (lines 1200 to 2130).

Lines 520 to 530

Save the instrument setting and the calibration coefficient into a file whose name is specified with the File variable.

Line 580

Displays a message that prompts you to connect a DUT (Device Under Test) in the instrument status bar in the lower part of the LCD display and waits for the operation of **Macro Setup > Continue** after the connection.

Lines 620 to 630

Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Line 650

For trace 1 of channel 1, executes auto scale to set the optimum scale.

Lines 690 to 710

Display marker 1 and move it so that the stimulus value becomes equal to the value of the Center variable. Then, these lines read out the response value of marker 1 and store it into the MkrVal variable.

Line 730

E5071C

Enables the error handling routine starting from Bw_Err (lines 890 to 950). If a runtime error occurs, the program goes to the error handling routine.

Lines 750 to 770

Set the bandwidth definition value to -3 dB and the bandwidth search result display to on, read out the bandwidth search result (bandwidth, center frequency, Q value, and insertion loss), and store it into the BwData variable.

Lines 790 to 840

Based on the bandwidth search result, these lines store the bandwidth to the Bw variable, the center frequency to the Cent variable, the Q value to the Qfac variable, and the insertion loss to the Loss variable. Then, the program goes to the processing starting from Skip_Bw_Err.

Lines 880 to 960

Define a runtime error handler. These lines read out and display the error number and error message of the error that occurred and store 0 to the Bw, Cent, and Qfac variables and the response value of marker 1 (MkrVal(0) variable) to the Loss variable. Then, the program finishes the error handling and proceeds to the next processing.

Lines 1000 to 1010

Calculate the 2 (higher and lower) cutoff frequencies from the values in the Bw and Cent variables and store them into the CutLow and CutHigh variables.

Lines 1030 to 1110

Display the search result (the values of the Bw, Cent, CutLow, CutHigh, Qfac, and Loss variables) in the echo window.

Lines 1130 to 1160

Display the message asking whether you want to perform measurement again. Click **Yes** to return to the DUT connection section, otherwise click **No** to terminate the program.

Procedure: Calib_Solt (lines 1200 to 2130).

Lines 1260 to 1300

Display the message that prompts for the execution of the full n-port calibration (specified with the SoltType variable). Click **Cancel** to cancel the calibration.

Lines 1320 to 1410

Set the calibration type to the full n-port calibration for the port specified with the Port variable.

Lines 1450 to 1520

Display the message that prompts for connecting the open standard to the specified port. These lines start the measurement of the open calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1540 to 1610

Display the message that prompts for connecting the short standard to the specified port. These lines start the measurement of the short calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1630 to 1700

Display the message that prompts for connecting the load standard to the specified port. These lines start the measurement of the load calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1750 to 1840

Display the message that prompts for connecting the thru standard between the specified ports. These lines start the measurement of the thru calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1880 to 2060

When the calibration type is not the 1-port calibration (a value other than 1 is specified for the SoltType variable), displays the message asking you whether you want to measure the isolation calibration data. When **Yes** is clicked, displays the message that prompts for connecting the load standard to the specified two ports (specified with the Port(I-1) and Port(J-1) variables). These lines start the measurement of the isolation calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 2080 to 2090

Calculate the calibration coefficients from the measured calibration data and turn on the error correction function. Then, these lines display a calibration completion message.

Measuring a band-pass filter (object name: mdlBscMeas)

10| Sub Main()

E5071C

```
20|
30| Dim Par As String, Fmt As String, File As String
40| Dim Center As Double, Span As Double, IfBw As Double, Pow As Double
50| Dim Bw As Double, Cent As Double
60| Dim CutLow As Double, CutHigh As Double
70| Dim Qfac As Double, Loss As Double
80| Dim MkrVal As Variant, BwData As Variant
90| Dim Nop As Long, NumTrac As Long, CalKit As Long, Buff As Long
100| Dim Port As Variant, Error As Variant
110|
120| Center = 947500000# 'Center freq : 947.5 MHz
130| Span = 200000000# 'Span freq : 200 MHz
140| Nop = 401 'Number of points : 401
150| IfBw = 10000# 'IF bandwidth : 10 kHz
160| Pow = -10 'Power level : -10dBm
170| NumTrac = 1 'Number of traces : 1
180| Par = "S21" 'Meas. parameter : S21
190| Fmt = "MLOG" 'Data format : Log Mag
200| CalKit = 4 'Calibration kit : 85032F
210| File = "State08.sta" 'Saved file name : State08.sta
220|
230| ""Presetting the E5071C
240|
250| SCPI.SYSTem.PRESet
260|
270| ""Setting measurement conditions
280|
290| SCPI.INITiate(1).CONTinuous = True
300| SCPI.TRIGger.SEquence.Source = "BUS"
310|
320| SCPI.SENSE(1).FREQuency.Center = Center
330| SCPI.SENSE(1).FREQuency.Span = Span
340| SCPI.SENSE(1).SWEep.POINts = Nop
350| SCPI.SENSE(1).BANDwidth.RESolution = IfBw
360| SCPI.Source(1).POWER.LEVel.IMMEDIATE.AMPLitude = Pow
370|
380| SCPI.CALCulate(1).PARAmeter.Count = NumTrac
```



```

390| SCPI.CALCulate(1).PARAmeter(1).DEFine = Par
400| SCPI.CALCulate(1).PARAmeter(1).Select
410| SCPI.CALCulate(1).SElected.Format = Fmt
420|
430| ""Performing full 2-port calibration
440|
450| SCPI.SENSE(1).CORRection.COLlect.CKIT.Select = CalKit
460| Port = Array(1, 2)
470|
480| Calib_Solt 1, 2, Port
490|
500| ""Saving state & cal data
510|
520| SCPI.MMEMory.STORe.STYPe = "CST"
530| SCPI.MMEMory.STORe.STATe = File
540|
550| ""Connecting DUT
560|
570| Meas_Start:
580| Prompt ("Connect DUT, and then press [Macro Setup]-Continue button.")
590|
600| ""Performing single sweep
610|
620| SCPI.TRIGger.SEQuence.SINGLE
630| Dmy = SCPI.IEEE4882.OPC
640|
650| SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
660|
670| ""Analyzing the results
680|
690| SCPI.CALCulate(1).SElected.MARKer(1).STATe = True
700| SCPI.CALCulate(1).SElected.MARKer(1).X = Center
710| MkrVal = SCPI.CALCulate(1).SElected.MARKer(1).Y
720|
730| On Error GoTo Bw_Err
740|
750| SCPI.CALCulate(1).SElected.MARKer(1).BWIDth.THReshold = -3

```

E5071C

```
760| SCPI.CALCulate(1).SElected.MARKer(1).BWIDth.STATe = True
770| BwData = SCPI.CALCulate(1).SElected.MARKer(1).BWIDth.DATA
780|
790| Bw = BwData(0)
800| Cent = BwData(1)
810| Qfac = BwData(2)
820| Loss = BwData(3)
830|
840| GoTo Skip_Bw_Err
850|
860| Bw_Err:
870|
880| Error = SCPI.SYSTem.Error
890| MsgBox "Error No:" & Error(0) & " , Description:" & Error( 1)
900|
910| Bw = 0
920| Cent = 0
930| Qfac = 0
940| Loss = MkrVal(0)
950|
960| Resume Skip_Bw_Err
970|
980| Skip_Bw_Err:
990|
1000| CutLow = Cent - Bw / 2
1010| CutHigh = Cent + Bw / 2
1020|
1030| ECHO "##Measurement Result##"
1040| ECHO " BW:" & Bw
1050| ECHO "cent:" & Cent
1060| ECHO " low:" & CutLow
1070| ECHO "high:" & CutHigh
1080| ECHO " Q:" & Qfac
1090| ECHO "loss:" & Loss
1100| SCPI.DISPlay.TABLe.TYPE = "ECHO"
1110| SCPI.DISPlay.TABLe.STATe = True
1120|
```

```

1130| Buff = MsgBox("Do you make another measurement?", vbYesNo, "Bandpass fileter
measurement")
1140| If Buff = vbYes Then
1150| GoTo Meas_Start
1160| End If
1170|
1180| End Sub
1190|
1200| Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port As Variant)
1210|
1220| Dim Dmy As Long, I As Long, J As Long, Buff As Long
1230|
1240| Cal_Start:
1250|
1260| Buff = MsgBox("Perform the full " & SoltType & "-port cali bration.", vbOKCancel, "Full" & SoltType
& "-port calibration")
1270|
1280| If Buff = vbCancel Then
1290| GoTo Cal_Skip
1300| End If
1310|
1320| Select Case SoltType
1330| Case 1
1340| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT1 = Port(0)
1350| Case 2
1360| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT2 = Port
1370| Case 3
1380| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT3 = Port
1390| Case 4
1400| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT4 = Port
1410| End Select
1420|
1430| For I = 1 To SoltType
1440|
1450| Buff = MsgBox("Connect the Open standard to Port " & CS tr(Port(I - 1)) & ". ", _
1460| vbOKCancel, "Full" & SoltType & "-port calibration")
1470| If Buff = vbOK Then
1480| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.OPEN = Port(I - 1)

```

E5071C

```
1490| Dmy = SCPI.IEEE4882.OPC
1500| Else
1510| GoTo Cal_Start
1520| End If
1530|
1540| Buff = MsgBox("Connect the Short standard to Port " & CStr(Port(I - 1)) & ". ", _
1550| vbOKCancel, "Full" & SoltType & "-port calibration")
1560| If Buff = vbOK Then
1570| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.Short = Port(I - 1)
1580| Dmy = SCPI.IEEE4882.OPC
1590| Else
1600| GoTo Cal_Start
1610| End If
1620|
1630| Buff = MsgBox("Connect the Load standard to Port " & CStr(Port(I - 1)) & ". ", _
1640| vbOKCancel, "Full" & SoltType & "-port calibration")
1650| If Buff = vbOK Then
1660| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.Load = Port(I - 1)
1670| Dmy = SCPI.IEEE4882.OPC
1680| Else
1690| GoTo Cal_Start
1700| End If
1710| Next I
1720|
1730| For I = 1 To SoltType - 1
1740| For J = I + 1 To SoltType
1750| Buff = MsgBox("Connect the Thru standard between Por t " & CStr(Port(I - 1)) & _
1760| " and Port " & CStr(Port(J - 1)) & ". ", vbOKCancel, "Full" & SoltType & "-port calibration")
1770| If Buff = vbOK Then
1780| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.THru = Array(Port(I - 1), Port(J - 1))
1790| Dmy = SCPI.IEEE4882.OPC
1800| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.THru = Array(Port(J - 1), Port(I - 1))
1810| Dmy = SCPI.IEEE4882.OPC
1820| Else
1830| GoTo Cal_Start
1840| End If
1850| Next J
```

```

1860| Next I
1870|
1880| If SoltType <> 1 Then
1890| Buff = MsgBox("Do you measure the Isolation (Optional) ?", vbYesNo, "Full" & SoltType & "-port
calibration")
1900| If Buff = vbYes Then
1910| For I = 1 To SoltType - 1
1920| For J = I + 1 To SoltType
1930| Buff = MsgBox("Connect the Load standard to Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".",
-
1940| vbOKCancel, "Full" & Solt Type & "-port calibration")
1950| If Buff = vbOK Then
1960| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.ISOLation = Array(Port( I - 1), Port(J - 1))
1970| Dmy = SCPI.IEEE4882.OPC
1980| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.ISOLation = Array(Port( J - 1), Port(I - 1))
1990| Dmy = SCPI.IEEE4882.OPC
2000| Else
2010| GoTo Cal_Start
2020| End If
2030| Next J
2040| Next I
2050| End If
2060| End If
2070|
2080| SCPI.SENSE(1).CORRection.COLlect.SAVE
2090| MsgBox "All calibration data completion."
2100|
2110| Cal_Skip:
2120|
2130| End Sub

```

Measuring a Multi-port Device

- [Overview](#)
- [Overview of the Program](#)
- [Description of the Program](#)
- [Sample Program](#)

Other topics about Application Programs

Overview

The **apl_sys.vba** shows a sample program (VBA program) that demonstrates how to measure a (3-terminal) duplexer. This VBA program consists of the following modules:

NOTE

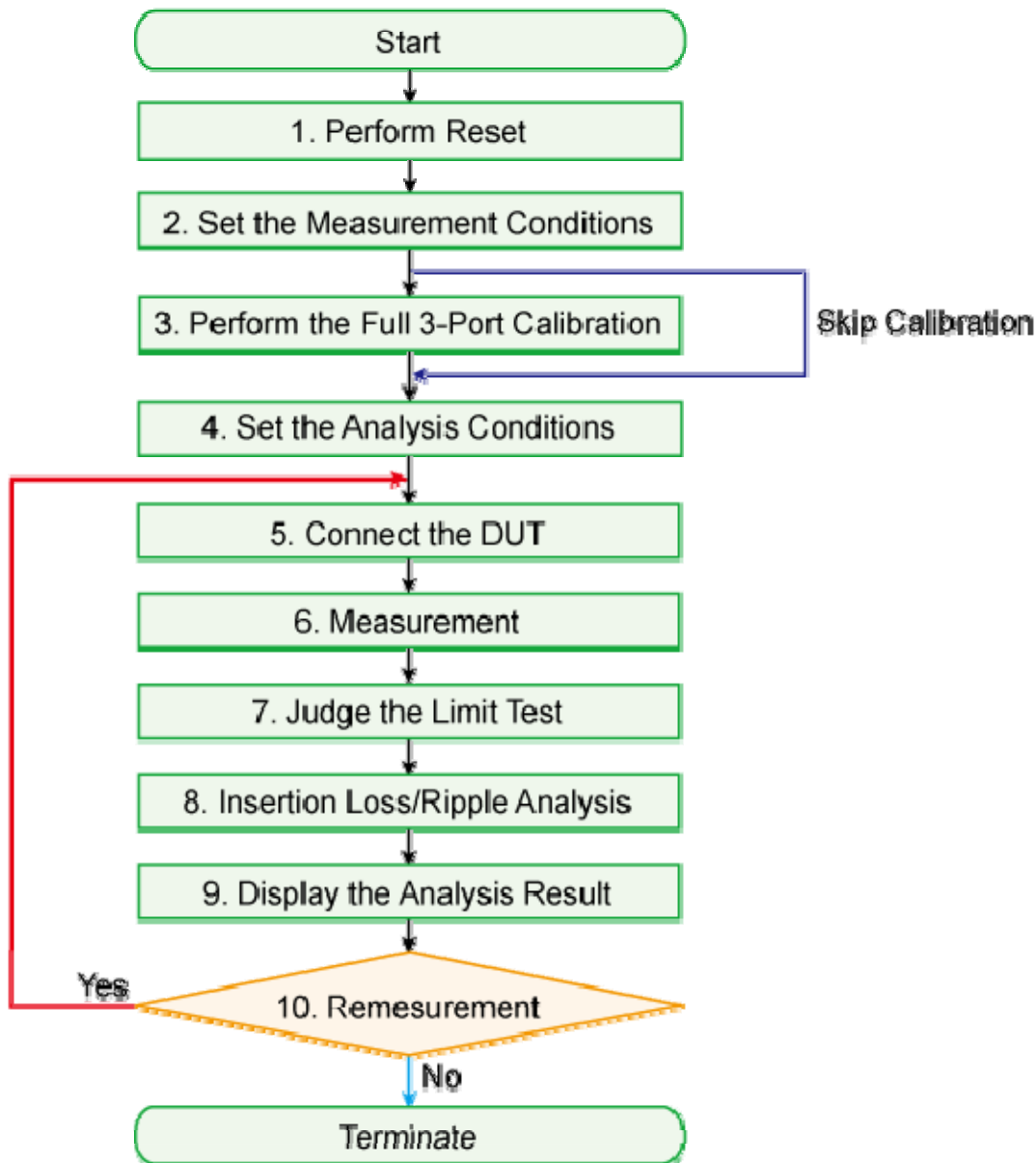
For the E5071C with 2-port S-parameter test set, a runtime error occurs because there are parameters that cannot be measured.

Object name	Module type	Description
frmDupRes	User form	Displays the analysis result
mdlDupMeas	Standard module	Performs duplexer measurement

Overview of the Program

The program performs full 3-port calibration using the 85032F calibration kit, measures a (3-terminal) duplexer, and calculates and displays the limit test result, insertion loss, and band-pass ripple. The following figure shows the simple flow of the (3-terminal) duplexer measurement program:

Flow of duplexer measurement program



e5071c435

Description of the Program

When you run this VBA program, reset is performed, the measurement conditions are set, and the message "Perform the full 3-port calibration" is displayed. To perform the full 3-port calibration, click **Yes**, otherwise click **No**.

To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port and then click **OK** to measure the calibration data. Click **Cancel** to return to the beginning of the calibration. You cannot skip the isolation calibration. When

the calibration data measurement for all standards is complete, the message "All calibration data completion" is displayed, and the calibration coefficient is calculated.

NOTE

When you cancel the calibration data measurement before completing the measurement of necessary calibration data, the settings conditions may not return to its original state.

Then, the limit line is set and the setting required for the limit test judgment is performed.

The message "Set the DUT between test cables." is displayed. Connect the DUT (duplexer) between the test cables and then click **Yes**. The limit line is set and a single sweep is executed. Then, for both trace 1 (Tx: S13) and trace 2 (Rx: S21), the Pass/Fail judgment of the limit test and the insertion loss and ripple analysis result within the pass band are displayed.

Click **Exit** on the user form displaying the analysis result. The message prompting for remeasurement is displayed. To perform remeasurement, click **Yes**, otherwise click **No** to terminate the program. The details of the program within the user form for displaying the analysis result (object name: frmDupRes) are not described here.

Display of the execution result of the program of "apl_sys.vba"

	Tx (S13)	Rx (S21)
Limit Line Test:	Pass	Pass
Insertion Loss (dB):	-3.9345	-5.3429
Pass Band Ripple (dB):	0.2901	0.0093
Total Result:	Pass	

Exit

e5071c187

The duplexer measurement program (object name: mdlDupMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Line 90

Stores the calibration kit number (4: 85032F) into the CalKit variable.

Line 110

Turns off display update. Turning off display update shortens drawing time and object processing time.

Line 140

Returns the E5071C to the preset state.

Lines 170 to 180

For channel 1, these lines turn the continuous trigger startup mode to On and sets the trigger source to the bus trigger.

Line 200

Calls the Setup_Parameter procedure (lines 910 to 1140). For information on the Setup_Parameter procedure, see the description later.

Line 220

Calls the Setup_Segment procedure (lines 1160 to 1530). For information on the Setup_Segment procedure, see the description later.

Line 250

Stores the calibration kit number for channel 1 into the CalKit variable.

Line 260

Stores 1, 2, and 3 into the Port variable that indicates the ports used for the full 3-port calibration.

Line 280

Calls the Calib_Solt procedure.

Line 310

Calls the Setup_Limitline procedure (lines 2420 to 3180). For information on the Setup_Limitline procedure, see the description later.

Line 330

Calls the Setup_Register procedure (lines 3200 to 3260). For information on the Setup_Register procedure, see the description later.

Line 380

Displays the message that prompts for connecting a DUT (Device Under Test) and waits for the **OK** button to be clicked after the connection.

Line 410

Clears the questionable limit status event register and questionable limit channel 1 status event register.

Lines 420 to 430

Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

E5071C

Lines 450 to 460

For traces 1 and 2, these lines executes the auto scale to set the optimum scale.

Lines 490 to 500

Read out the value of the questionable limit status event register and store the AND of the read-out value and 2 (the value in which only bit 1 is 1) into the Test_Ch1 variable.

Lines 510 to 530

Read out the value of the questionable limit channel 1 status event register and store the AND of the read-out value and 2 (the value in which only bit 1 is 1) into the Test_Tr1 variable and the AND of the read-out value and 4 (the value in which only bit 2 is 1) into the Test_Tr2 variable.

Lines 560 to 610

Specify trace 1 as the active trace and set the analysis range (start point: 1.85 GHz and stop point: 1.91 GHz). Then these lines sets the polarity of the peak search (both the positive peak and the negative peak) and the lower limit of the peak excursion value (0).

Lines 620 to 640

Search for the minimum value within the analysis range and store the analysis result into the IlossTx variables.

Line 650

Uses the ripple analysis function to store the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range into the RipTx variables.

Lines 670 to 720

Specify trace 2 as the active trace and set the analysis range (start point: 1.93 GHz and stop point: 1.99 GHz). Then these lines set the polarity of the peak search (both the positive peak and the negative peak) and the lower limit of the peak excursion value (0).

Lines 730 to 750

Search for the minimum value within the analysis range and store the analysis result into the IlossRx variables.

Line 760

Uses the ripple analysis function to store the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range into the RipRx variables.

Line 790

Calls the Display_Update procedure (lines 3280 to 3620). For information on the Display_Update procedure, see the description later.

Line 810

Displays the user form (object name: frmDupRes) on the screen to display the analysis result.

Lines 830 to 870

Display the message asking whether you want to perform measurement again. Click the **Yes** button to return to the DUT connection section. Click the **No** button to terminate the program.

Procedure: Setup_Parameter (lines 910 to 1140)

Lines 970 to 1020

Store the channel layout ("D1": 1-channel display), graph layout ("D1_2": upper/lower 2 part split display), measurement parameter of trace 1 (S13), measurement parameter of trace 2 (S21), data format of trace 1 (MLOG), and data format of trace 2 (MLOG) into the ChDisp, TracDisp, Par(0), Par(1), Fmt(0), and Fmt(1) variables, respectively.

Lines 1040 to 1060

Set the number of traces for channel 1 to 2, the channel layout to the ChDisp variable, and the graph layout for channel 1 to the TracDisp variable.

Lines 1080 to 1120

Set the measurement parameter for trace 1 to the Par(0) variable, the data format for trace 1 to the Fmt(0) variable, the measurement parameter for trace 2 to the Par(1) variable, and the data format for trace 2 to the Fmt(1) variable.

Procedure: Setup_Segment (lines 1160 to 1530)

Lines 1200 to 1260

Store the conditions for channel 1's segment table setting into the SegmData(0) to SegmData(6) variables. The settings are as follows. Stimulus setting mode: start/stop value. IF bandwidth setting for each segment: off. Power setting for each segment: off. Sweep delay time setting for each segment: off. Sweep time setting for each segment: off. Number of segments: 5.

Lines 1280 to 1470

Store the sweep start value, the sweep stop value, and the number of measurement points for channel 1's segments 1 through 5 into the SegmData(7) to SegmData(21) variables, respectively.

E5071C

Line 1490

Sets channel 1's segment table to the SegmData variable.

Line 1500

Sets channel 1's sweep type to "segment".

Line 1510

Sets the channel 1 graph's horizontal axis display method to the order base (the axis on which the measurement point numbers are placed evenly in the order of measurement).

Procedure: Calib_Solt (lines 1550 to 2420)

See Lines 1200 to 2130 of Calib_Solt.

Procedure: Setup_Limitline (lines 2440 to 3180)

Line 2490

Stores the number of lines (5) in trace 1 limit table into the LimDataS13(0) variable.

Lines 2500 to 2790

Store the settings in trace 1 limit table into the LimDataS13(1) to LimDataS13(25) variables.

Line 2820

Stores the number of lines (4) in trace 2 limit table into the LimDataS21(0) variable.

Lines 2830 to 3060

Store the settings in trace 2 limit table into the LimDataS21(1) to LimDataS21(20) variables.

Lines 3080 to 3110

Specify trace 1 as the active trace, store trace 1's limit line into the LimDataS13 variable, and display it. Then, these lines turn on the limit test function for trace 1.

Lines 3130 to 3160

Specify trace 2 as the active trace, store trace 2's limit line into the LimDataS21 variable, and display it. Then, these lines turn on the limit test function for trace 2.

Procedure: Setup_Register (lines 3200 to 3260)

Lines 3220 to 3230

1432

Set the instrument so that the questionable limit channel status event register's bits 1 and 2 are set to 1 only when the questionable limit channel status register's bits 1 and 2 are changed from 0 to 1 (positive transition).

Line 3240

Enables the questionable limit channel status event register's bits 1 and 2.

Procedure: Display_Update (lines 3280 to 3620).

Line 3300

Updates the display on the LCD screen once.

Lines 3320 to 3380

When trace 1's limit test result is Fail (Test_Tr1 = 2), these lines display Tx(S13) "Limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when trace 1's limit test result is Pass (Test_Tr1 \neq 2), they display Tx(S13) "Limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3400 to 3460

When trace 2's limit test result is Fail (Test_Tr2 = 4), these lines display Rx(S21) "Limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when trace 1's limit test result is Pass (Test_Tr2 \neq 4), they display Rx(S21) "Limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3480 to 3540

When channel 1's limit test result is Fail (Test_Ch1 = 2), these lines display "Overall limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when channel 1's limit test result is Pass (Test_Ch1 \neq 2), they display "Overall limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3560 to 3600

Display the analysis results for traces 1 and 2 (insertion loss and band-pass ripple) as Tx(S13) and Rx(S21), respectively, on the user form (object name: frmDupRes).

Sample Program

Duplexer measurement (object name: mdlDupMeas)

10| Sub Main()

20|

E5071C

```
30| Dim CalKit As Long, Dmy As Long, Rgst As Long, I As Long, Buff As Long
40| Dim Test_Ch1 As Integer, Test_Tr1 As Integer, Test_Tr2 As Integer
50| Dim IlossTx As Variant, IlossRx As Variant
60| Dim RipTx As Double, RipRx As Double
70| Dim Port As Variant
80|
90| CalKit = 4 'Calibration kit :85032F
100|
110| SCPI.DISPlay.ENABLE = False
120|
130| ""Presetting the E5071C
140| SCPI.SYSTem.PRESet
150|
160| ""Setting measurement conditions
170| SCPI.INITiate(1).CONTinuous = True
180| SCPI.TRIGger.SEQuence.Source = "BUS"
190|
200| Setup_Parameter
210|
220| Setup_Segment
230|
240| ""Full 3-port calibration
250| SCPI.SENSE(1).CORRection.COLlect.CKIT.Select = CalKit
260| Port = Array(1, 2, 3)
270|
280| Calib_Solt 1, 3, Port
290|
300| ""Setting analysis conditions
310| Setup_Limitline
320|
330| Setup_Register
340|
350| Meas_Start:
360|
370| ""Connecting DUT
380| MsgBox "Connect DUT between test cables."
390|
```

```

400| ""Performing single sweep
410| SCPI.IEEE4882.CLS
420| SCPI.TRIGger.SEQuence.SINGLe
430| Dmy = SCPI.IEEE4882.OPC
440|
450| SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
460| SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO
470|
480| ""Judging limit test
490| Rgst = SCPI.STATus.QUEStionable.LIMit.EVENT
500| Test_Ch1 = CInt(Rgst And 2)
510| Rgst = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENT
520| Test_Tr1 = CInt(Rgst And 2)
530| Test_Tr2 = CInt(Rgst And 4)
540|
550| ""Analyzing insertion loss/bandpass ripple
560| SCPI.CALCulate(1).PARAmeter(1).Select
570| SCPI.CALCulate(1).SElected.FUNcTION.DOMain.START = 1850000000#
580| SCPI.CALCulate(1).SElected.FUNcTION.DOMain.STOP = 1910000000#
590| SCPI.CALCulate(1).SElected.FUNcTION.DOMain.STATe = True
600| SCPI.CALCulate(1).SElected.FUNcTION.PPOLarity = "both"
610| SCPI.CALCulate(1).SElected.FUNcTION.PEXCursion = 0
620| SCPI.CALCulate(1).SElected.FUNcTION.TYPE = "MIN"
630| SCPI.CALCulate(1).SElected.FUNcTION.EXECute
640| llossTx = SCPI.CALCulate(1).SElected.FUNcTION.DATA
650| RipTx = MaxPeakToPeak(1)
660|
670| SCPI.CALCulate(1).PARAmeter(2).Select
680| SCPI.CALCulate(1).SElected.FUNcTION.DOMain.START = 1930000000#
690| SCPI.CALCulate(1).SElected.FUNcTION.DOMain.STOP = 1990000000#
700| SCPI.CALCulate(1).SElected.FUNcTION.DOMain.STATe = True
710| SCPI.CALCulate(1).SElected.FUNcTION.PPOLarity = "both"
720| SCPI.CALCulate(1).SElected.FUNcTION.PEXCursion = 0
730| SCPI.CALCulate(1).SElected.FUNcTION.TYPE = "MIN"
740| SCPI.CALCulate(1).SElected.FUNcTION.EXECute
750| llossRx = SCPI.CALCulate(1).SElected.FUNcTION.DATA
760| RipRx = MaxPeakToPeak(1)

```

E5071C

```
770|
780| ""Displaying the results
790| Display_Update Test_Tr1, Test_Tr2, Test_Ch1, llossTx, llossRx, RipTx, RipRx
800|
810| frmDupRes.Show
820|
830| Buff = MsgBox("Do you make another measurement?", vbYesNo, "Duplexer Measurement")
840|
850| If Buff = vbYes Then
860| GoTo Meas_Start
870| End If
880|
890| End Sub
900|
910| Private Sub Setup_Parameter()
920|
930| Dim I As Long
940| Dim ChDisp As String, TracDisp As String
950| Dim Par(1) As String, Fmt(1) As String
960|
970| ChDisp = "D1"
980| TracDisp = "D1_2"
990| Par(0) = "S13"
1000| Par(1) = "S21"
1010| Fmt(0) = "MLOG"
1020| Fmt(1) = "MLOG"
1030|
1040| SCPI.CALCulate(1).PARameter.Count = 2
1050| SCPI.DISPlay.Split = ChDisp
1060| SCPI.DISPlay.WINDow(1).Split = TracDisp
1070|
1080| For I = 1 To 2
1090| SCPI.CALCulate(1).PARameter(I).DEFine = Par(I - 1)
1100| SCPI.CALCulate(1).PARameter(I).Select
1110| SCPI.CALCulate(1).SELected.Format = Fmt(I - 1)
1120| Next I
1130|
```



```

1140| End Sub
1150|
1160| Private Sub Setup_Segment()
1170|
1180| Dim SegmData(21) As Variant
1190|
1200| SegmData(0) = 5 'Anytime 5 is set at seg ment settings
1210| SegmData(1) = 0 'Allows stimulus range to be set using Start/Stop frequency
1220| SegmData(2) = 0 'Not allows IF bandwidth to be set
1230| SegmData(3) = 0 'Not allows power to be set
1240| SegmData(4) = 0 'Not allows delay time to be set
1250| SegmData(5) = 0 'Not allows sweep time to be set
1260| SegmData(6) = 5 'Number of segments
1270|
1280| ""Segment 1
1290| SegmData(7) = 1730000000# 'Start frequency
1300| SegmData(8) = 1830000000# 'Stop frequency
1310| SegmData(9) = 50 'Number of points
1320| ""Segment 2
1330| SegmData(10) = 1830000000# 'Start frequency
1340| SegmData(11) = 2030000000# 'Stop frequency
1350| SegmData(12) = 400 'Number of points
1360| ""Segment 3
1370| SegmData(13) = 2030000000# 'Start frequency
1380| SegmData(14) = 2130000000# 'Stop frequency
1390| SegmData(15) = 50 'Number of points
1400| ""Segment 4
1410| SegmData(16) = 3650000000# 'Start frequency
1420| SegmData(17) = 4030000000# 'Stop frequency
1430| SegmData(18) = 38 'Number of points
1440| ""Segment 5
1450| SegmData(19) = 5500000000# 'Start frequency
1460| SegmData(20) = 6020000000# 'Stop frequency
1470| SegmData(21) = 52 'Number of points
1480|
1490| SCPI.SENSE(1).SEGMENT.DATA = SegmData
1500| SCPI.SENSE(1).SWEp.TYPE = "SEGM"

```

E5071C

```
1510| SCPI.DISPlay.WINDow(1).X.SPACing = "OBAS"
1520|
1530| End Sub
1540|
1550| Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port A s Variant)
1560|
1570| Dim Dmy As Long, I As Long, J As Long, Buff As Long
1580|
1590| Cal_Start:
1600|
1610| Buff = MsgBox("Perform the full " & SoltType & "-port calib ration.", vbOKCancel, "Full" & SoltType
& "-port calibration")
1620| If Buff = vbCancel Then
1630| GoTo Cal_Skip
1640| End If
1650|
1660| Select Case SoltType
1670| Case 1
1680| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT1 = Port(0)
1690| Case 2
1700| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT2 = Port
1710| Case 3
1720| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT3 = Port
1730| Case 4
1740| SCPI.SENSE(Chan).CORRection.COLlect.METHod.SOLT4 = Port
1750| End Select
1760|
1770| For I = 1 To SoltType
1780|
1790| Buff = MsgBox("Connect the Open standard to Port " & CS tr(Port(I - 1)) & ".", vbOKCancel, "Full"
& SoltType & "-port calibr ation")
1800| If Buff = vbOK Then
1810| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.OPEN = Port(I - 1)
1820| Dmy = SCPI.IEEE4882.OPC
1830| Else
1840| GoTo Cal_Start
1850| End If
1860|
```

```

1870| Buff = MsgBox("Connect the Short standard to Port " & C Str(Port(I - 1)) & ".", vbOKCancel, "Full"
& SoltType & "-port calibration")
1880| If Buff = vbOK Then
1890| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.Short = Port(I - 1)
1900| Dmy = SCPI.IEEE4882.OPC
1910| Else
1920| GoTo Cal_Start
1930| End If
1940|
1950| Buff = MsgBox("Connect the Load standard to Port " & C Str(Port(I - 1)) & ".", vbOKCancel, "Full" &
SoltType & "-port calibration")
1960| If Buff = vbOK Then
1970| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.Load = Port(I - 1)
1980| Dmy = SCPI.IEEE4882.OPC
1990| Else
2000| GoTo Cal_Start
2010| End If
2020| Next I
2030|
2040| For I = 1 To SoltType - 1
2050| For J = I + 1 To SoltType
2060| Buff = MsgBox("Connect the Thru standard between Por t " & CStr(Port(I - 1)) & " and Port " &
CStr(Port(J - 1)) & ".", vb OKCancel, "Full" & SoltType & "-port calibration")
2070| If Buff = vbOK Then
2080| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.THRU = Array(Port(I - 1), Port(J - 1))
2090| Dmy = SCPI.IEEE4882.OPC
2100| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.THRU = Array(Port(J - 1), Port(I - 1))
2110| Dmy = SCPI.IEEE4882.OPC
2120| Else
2130| GoTo Cal_Start
2140| End If
2150| Next J
2160| Next I
2170|
2180| If SoltType <> 1 Then
2190| Buff = MsgBox("Do you measure the Isolation (Optional)? ", vbYesNo, "Full" & SoltType & "-port
calibration")
2200| If Buff = vbYes Then

```

E5071C

```
2210| For I = 1 To SoltType - 1
2220| For J = I + 1 To SoltType
2230| Buff = MsgBox("Connect the Load standard to Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".",
vbOKCancel, "Full" & SoltType & "-port calibration")
2240| If Buff = vbOK Then
2250| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.ISOLation = Array(Port( I - 1), Port(J - 1))
2260| Dmy = SCPI.IEEE4882.OPC
2270| SCPI.SENSE(Chan).CORRection.COLlect.ACQuire.ISOLation = Array(Port( J - 1), Port(I - 1))
2280| Dmy = SCPI.IEEE4882.OPC
2290| Else
2300| GoTo Cal_Start
2310| End If
2320| Next J
2330| Next I
2340| End If
2350| End If
2360|
2370| SCPI.SENSE(1).CORRection.COLlect.SAVE
2380| MsgBox "All calibration data completion."
2390|
2400| Cal_Skip:
2410|
2420| End Sub
2430|
2440| Private Sub Setup_Limitline()
2450|
2460| Dim LimDataS13(25) As Variant, LimDataS21(20) As Variant
2470|
2480| ""Limit line for S13
2490| LimDataS13(0) = 5 'Number of segment
2500| ""Limit_line 1
2510| LimDataS13(1) = 1 'Maximum
2520| LimDataS13(2) = 1730000000# 'Beginning of stimulus
2530| LimDataS13(3) = 1930000000# 'End of stimulus
2540| LimDataS13(4) = 0 'Beginning of response
2550| LimDataS13(5) = 0 'End of response
2560| ""Limit_line 2
```

```

2570| LimDataS13(6) = 2 'Minimum
2580| LimDataS13(7) = 1850000000# 'Beginning of stimulus
2590| LimDataS13(8) = 1910000000# 'End of stimulus
2600| LimDataS13(9) = -8 'Beginning of response
2610| LimDataS13(10) = -8 'End of response
2620| ""Limit_line 3
2630| LimDataS13(11) = 1 'Maximum
2640| LimDataS13(12) = 1930000000# 'Beginning of stimulus
2650| LimDataS13(13) = 1990000000# 'End of stimulus
2660| LimDataS13(14) = -35 'Beginning of response
2670| LimDataS13(15) = -35 'End of response
2680| ""Limit_line 4
2690| LimDataS13(16) = 1 'Maximum
2700| LimDataS13(17) = 1990000000# 'Beginning of stimulus
2710| LimDataS13(18) = 2130000000# 'End of stimulus
2720| LimDataS13(19) = -40 'Beginning of response
2730| LimDataS13(20) = -40 'End of response
2740| ""Limit_line 5
2750| LimDataS13(21) = 1 'Maximum
2760| LimDataS13(22) = 2130000000# 'Beginning of stimulus
2770| LimDataS13(23) = 6020000000# 'End of stimulus
2780| LimDataS13(24) = -20 'Beginning of response
2790| LimDataS13(25) = -20 'End of response
2800|
2810| ""Limit line for S21
2820| LimDataS21(0) = 4 'Number of segment
2830| ""Limit_line 1
2840| LimDataS21(1) = 1 'Maximum
2850| LimDataS21(2) = 1730000000# 'Beginning of stimulus
2860| LimDataS21(3) = 1850000000# 'End of stimulus
2870| LimDataS21(4) = -40 'Beginning of response
2880| LimDataS21(5) = -40 'End of response
2890| ""Limit_line 2
2900| LimDataS21(6) = 1 'Maximum
2910| LimDataS21(7) = 1850000000# 'Beginning of stimulus
2920| LimDataS21(8) = 1910000000# 'End of stimulus
2930| LimDataS21(9) = -40 'Beginning of response

```

E5071C

```
2940| LimDataS21(10) = -40 'End of response
2950| ""Limit_line 3
2960| LimDataS21(11) = 1 'Maximum
2970| LimDataS21(12) = 1910000000# 'Beginning of stimulus
2980| LimDataS21(13) = 6020000000# 'End of stimulus
2990| LimDataS21(14) = 0 'Beginning of response
3000| LimDataS21(15) = 0 'End of response
3010| ""Limit_line 4
3020| LimDataS21(16) = 2 'Minimum
3030| LimDataS21(17) = 1930000000# 'Beginning of stimulus
3040| LimDataS21(18) = 1990000000# 'End of stimulus
3050| LimDataS21(19) = -10 'Beginning of response
3060| LimDataS21(20) = -10 'End of response
3070|
3080| SCPI.CALCulate(1).PARAmeter(1).Select
3090| SCPI.CALCulate(1).SELected.LIMit.DATA = LimDataS13
3100| SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe = True
3110| SCPI.CALCulate(1).SELected.LIMit.STATe = True
3120|
3130| SCPI.CALCulate(1).PARAmeter(2).Select
3140| SCPI.CALCulate(1).SELected.LIMit.DATA = LimDataS21
3150| SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe = True
3160| SCPI.CALCulate(1).SELected.LIMit.STATe = True
3170|
3180| End Sub
3190|
3200| Private Sub Setup_Register()
3210|
3220| SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition = 6
3230| SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition = 0
3240| SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABle = 6
3250|
3260| End Sub
3270|
3280| Sub Display_Update(Test_Tr1 As Integer, Test_Tr2 As Integer, Test_Ch1 As Integer, llossTx As
Variant, llossRx As Variant, RipTx As Variant, RipRx As Variant)
3290|
```

```

3300| SCPI.DISPlay.UPDate.IMMediate
3310|
3320| If Test_Tr1 = 2 Then
3330| frmDupRes.lblJudgeS13.BackColor = RGB(255, 0, 0)
3340| frmDupRes.lblJudgeS13.Caption = "Fail"
3350| Else
3360| frmDupRes.lblJudgeS13.BackColor = RGB(0, 0, 255)
3370| frmDupRes.lblJudgeS13.Caption = "Pass"
3380| End If
3390|
3400| If Test_Tr2 = 4 Then
3410| frmDupRes.lblJudgeS21.BackColor = RGB(255, 0, 0)
3420| frmDupRes.lblJudgeS21.Caption = "Fail"
3430| Else
3440| frmDupRes.lblJudgeS21.BackColor = RGB(0, 0, 255)
3450| frmDupRes.lblJudgeS21.Caption = "Pass"
3460| End If
3470|
3480| If Test_Ch1 = 2 Then
3490| frmDupRes.lblResult.BackColor = RGB(255, 0, 0)
3500| frmDupRes.lblResult.Caption = "Fail"
3510| Else
3520| frmDupRes.lblResult.BackColor = RGB(0, 0, 255)
3530| frmDupRes.lblResult.Caption = "Pass"
3540| End If
3550|
3560| frmDupRes.txtIlossS13.Text = Format(IlossTx(0), "0.#### ")
3570| frmDupRes.txtIlossS21.Text = Format(IlossRx(0), "0.#### ")
3580|
3590| frmDupRes.txtRipS13.Text = Format(RipTx, "0.####")
3600| frmDupRes.txtRipS21.Text = Format(RipRx, "0.####")
3610|
3620| End Sub

```

Measurement Using E5091A (measuring FEM)

- [Overview](#)
- [Sample Program](#)

Other topics about Application Programs

Overview

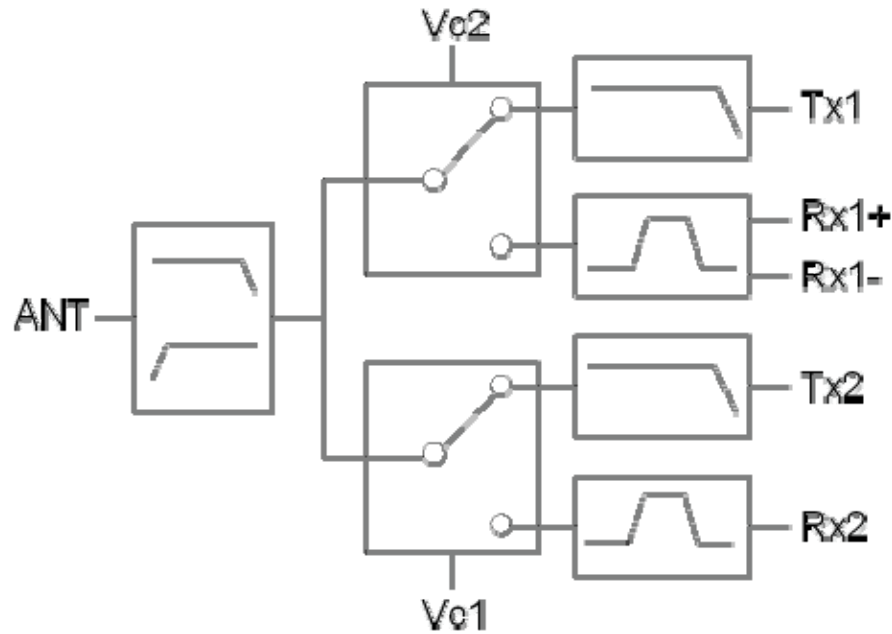
The **apl_fem.vba** shows front end module (FEM) measurement as a sample program of measurement using the E5091A.

NOTE If the E5071C does not have 4-port S parameter test set, a runtime error occurs because there are parameters that cannot be measured.

NOTE This VBA program cannot control the E5091A-016, the multiport test set.

Object name	Module type	Description
mdlFemMeas	Standard module	Performs measurement of FEM

This program calibrates each channel using the ECal module and then measures the transmission characteristics EGSM:Tx-Antenna (channel 1), EGSM:Antenna-Rx (channel 2), GSM1800:Tx-Antenna (channel 3), and GSM1800:Antenna-Rx (channel 4) of the 6-port dual-band FEM as shown in the below figure.



6070b-073

When you start the program, "Connect A and T1 to ECal Module" is displayed. Connect the cables from A and T1 of the E5091A to the ECal module and then press the **OK** key to calibrate channel 1. If trouble occurs due to a problem in the connection to the ECal module, an error message is displayed. You can execute calibration again by clicking the **Retry** button. If you want to abort the program, click the **Cancel** button. For channels 2 to 4, execute the calibration in the same way.

When calibration is complete, "Set DUT" is displayed. Connect the DUT (FEM) and the E5091A as shown below and click the **OK** button to start measurement.

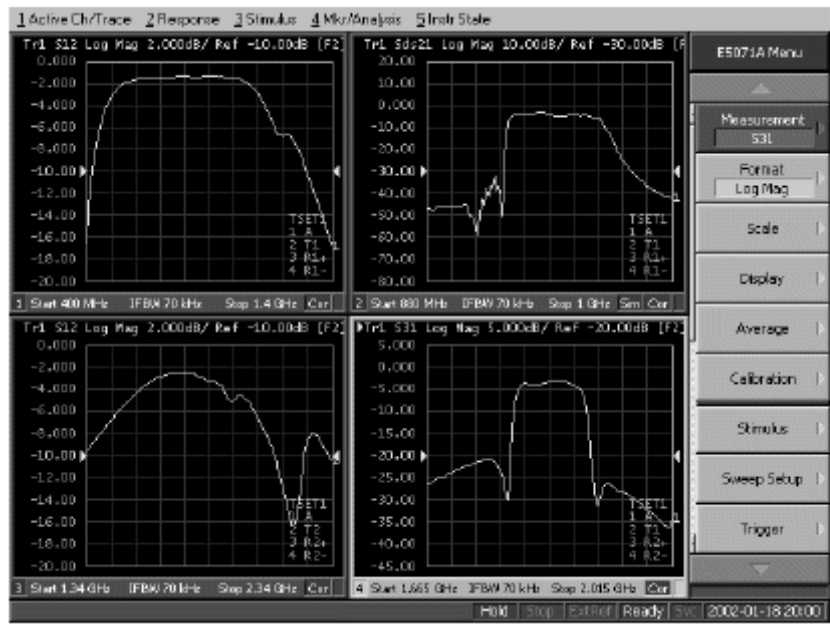
FEM		E5091A
Antenna		A
EGSM	Tx	T1
	Rx+	R1+
	Rx-	R1-
GSM1800	Tx	T2
	Rx	R2+

E5071C

Vc1	Control Line 1
Vc2	Control Line 2

The following figure shows a sample display of the LCD screen after the program exits execution.

Example of display after execution of program in "apl_fem.vba"



The FEM measurement program (object name: mdIFemMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Lines 140 to 330

Set the ports assigned to Port 1 to Port 4 of the E5091A and the control line setting (table below) into the variables.

Channel number	Port 1	Port 2	Port 3	Port 4	Control Lines
1	A	T1	R1+	R1-	0 (00000000)
2	A	T1	R1+	R1-	2 (00000010)

3	A	T2	R2+	R2-	0 (00000000)
4	A	T2	R2+	R2-	1 (00000001)

Lines 340 to 660

Set the settings required for the measurement conditions in the table below to the variables.

Channel number	Sweep range		Number of points	Number of traces	Measurement parameter
	Start	Stop			
1	400 MHz	1.4 GHz	51	1	S12
2	880 MHz	1 GHz	101	1	Sds21
3	1.34 GHz	2.34 GHz	201	1	S12
4	1.665 GHz	2.015 GHz	101	1	S31

Channel number	Fixture simulator		Title
	ON/OFF	Topology	
1	OFF	E/span> E/span>	[EGSM] Tx-Antenna
2	ON	SE:1, Bal:3,4	[EGSM] Antenna-Rx
3	OFF	E/span> E/span>	[GSM1800] Antenna-Rx
4	OFF	E/span> E/span>	[GSM1800] Tx-Antenna

Line 710

E5071C

Puts the instrument into preset state.

Line 720

Allocates the windows to the upper left, upper right, lower left, and lower right.

Lines 740 to 1020

Repeat the following for channels 1 to 4, where Ch is the channel number.

Lines 780 to 810: For the E5091A whose ID is 1, set the port assigned to port 1 to Port1(Ch-1), the port assigned to port 2 to Port2(Ch-1), the Port assigned to port 3 to Port3(Ch-1), and the port assigned to port 4 to Port4(Ch-1).

Line 820: Sets the control line of the E5091A whose ID is 1 to Clines(Ch-1).

Lines 860 to 890: Set the sweep start value to Freq_star(Ch-1), the sweep stop value to Freq_stop(Ch-1), the number of points to Nop(Ch-1), and the number of traces to N_tr(Ch-1).

Lines 910 to 950: If the fixture simulator function is ON (Fsim(Ch-1) is True), these lines set the fixture simulator function to ON, the device type to Dev(Ch-1), the port assignment to Tpl(Ch-1), the balance-unbalance conversion to ON, and the measurement parameter (mix mode S-parameter) to Trc(Ch-1).

Line 970: If the fixture simulator function is OFF (Fsim(i) is False), sets the measurement parameter (S-parameter) to Trc(Ch-1).

Lines 990 to 1010: Set the title label to Ttl(Ch-1), the title display to ON, and the continuous startup mode to ON.

Line 1040

Sets the trigger source to "Bus."

Lines 1050 to 1060

For the E5091A whose ID is 1, set the property display to ON and the control to ON.

Lines 1100 to 1120

Recall the Function procedure: ECal_solt (Lines 1410 to 1770) to execute the calibration of channel 1 with the ECal module (full 2-port calibration of ports A and T1). If the calibration is not completed correctly, these lines abort the program.

Lines 1130 to 1210

Execute the calibration of channels 2 to 4 in the same way.

Line 1260

Displays the message that prompts for connecting a DUT (Device Under Test) and waits for the **OK** button to be clicked after the connection.

Lines 1280 to 1290

Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Lines 1310 to 1330

Execute auto scale for trace 1 of channels 1 to 4.

Line 1350

Displays the message asking whether you want to perform measurement again.

Line 1360

If the **Yes** button is clicked, returns to the DUT connection section.

Function procedure: ECal_solt (lines 1410 to 1770)

Line 1460

Clears the error queue.

Lines 1460 to 1480

Display the message that prompts for connecting the Tset_Port of the E5091A to the ECal module and wait for the **OK** button to be clicked after the connection.

Line 1500

Enables the error handling routine starting from Ecal_Err (lines 1670 to 1740). If a runtime error occurs, the program goes to the error handling routine.

Line 1540

If solt is 1, executes the ECal command that performs 1-port calibration on port Ana_port(0) of channel Ch.

Line 1560

If solt is 2, executes the ECal command that performs full 2-port calibration on port Ana_port of channel Ch.

Line 1580

If solt is 3, executes the ECal command that performs full 3-port calibration on port Ana_port of channel Ch.

Line 1600

If solt is 4, executes the ECal command that performs full 4-port calibration on port Ana_port of channel Ch.

E5071C

Line 1630

Sets the return value of ECal_solt to 0.

Lines 1670 to 1740

Define a runtime error handler.

Lines 780 to 810: For the E5091A whose ID is 1, these lines set the port assigned to port 1 to Port1(Ch-1), the port assigned to port 2 to Port2(Ch-1), the Port assigned to port 3 to Port3(Ch-1), and the port assigned to port 4 to Port4(Ch-1).

Line 1670: Retrieves the error number and error message from the error queue.

Line 1680: Displays the error message.

Line 1700: When the **Retry** button is clicked, the program disables the error handler routine and then returns to the connection part and repeats ECal.

Lines 1720 to 1730: When the **Cancel** button is clicked, the program sets the return value of ECal_solt to the error number and disables the error handler routine.

Sample Program

Measurement of FEM (object name: mdlFemMeas)

```
10| Sub Main()
20|
30| Dim Port1(3) As String, Port2(3) As String
40| Dim Port3(3) As String, Port4(3) As String
50| Dim Trc(3) As String, Fsim(3) As Boolean, Dev(3) As String
60| Dim Tpl(3) As Variant, Ttl(3) As String, Inp_char As String
70| Dim Freq_star(3) As Double, Freq_stop(3) As Double
80| Dim CLines(3) As Long, Nop(3) As Long, N_tr(3) As Long
90| Dim Ch As Long, Res As Long, Buff As Long, Dmy As Long
100| Dim AnaPort As Variant
110| '
120| ' E5091A Setup
130| '
140| Port1(0) = "A" '[Ch1] Port1: A
150| Port2(0) = "T1" ' Port2: T1
160| Port3(0) = "R1" ' Port3: R1+
170| Port4(0) = "R1" ' Port4: R1-
180| CLines(0) = 0 ' Control Lines: 0
```

1450

```

190| Port1(1) = "A" '[Ch2] Port1: A
200| Port2(1) = "T1" ' Port2: T1
210| Port3(1) = "R1" ' Port3: R1+
220| Port4(1) = "R1" ' Port3: R1-
230| CLines(1) = 2 ' Control Lines: 2 (Line1:HIGH)
240| Port1(2) = "A" '[Ch3] Port1: A
250| Port2(2) = "T2" ' Port2: T2
260| Port3(2) = "R2" ' Port3: R2+
270| Port4(2) = "R2" ' Port4: R2- (Dummy)
280| CLines(2) = 0 ' Control Lines: 2 (Line1:HIGH)
290| Port1(3) = "A" '[Ch4] Port1: A
300| Port2(3) = "T2" ' Port2: T2
310| Port3(3) = "R2" ' Port3: R2+
320| Port4(3) = "R2" ' Port4: R2- (Dummy)
330| CLines(3) = 1 ' Control Lines: 1 (Line0:HIGH)
340| '
350| ' Measurement Condition
360| ' [Ch1]
370| Freq_star(0) = 400000000# ' Start frequency : 400 MHz
380| Freq_stop(0) = 1400000000# ' Stop frequency : 1.4 GHz
390| Nop(0) = 51 ' Number of points : 51
400| N_tr(0) = 1 ' Number of traces : 1
410| Fsim(0) = False ' Fixture Simulator : OFF
420| Trc(0) = "S12" ' Meas. parameter : S12
430| Ttl(0) = "[EGSM] Tx-Antenna" ' Title
440| ' [Ch2]
450| Freq_star(1) = 880000000# ' Start frequency : 880 MHz
460| Freq_stop(1) = 1000000000# ' Stop frequency : 1 GHz
470| Nop(1) = 101 ' Number of points : 101
480| N_tr(1) = 1 ' Number of traces : 1
490| Fsim(1) = True ' Fixture Simulator : ON
500| Dev(1) = "SBAL" ' Bal. Device Type : SE-Bal
510| Tpl(1) = Array(1, 3, 4) ' Topology : SE:1, Bal:3-4
520| Trc(1) = "SDS21" ' Meas. parameter : Sds21
530| Ttl(1) = "[EGSM] Antenna-Rx" ' Title
540| ' [Ch3]
550| Freq_star(2) = 1340000000# ' Start frequency : 1.34 GHz

```

E5071C

```
560| Freq_stop(2) = 2340000000# ' Stop frequency : 2.34 GHz
570| Nop(2) = 201 ' Number of points : 201
580| N_tr(2) = 1 ' Number of traces : 1
590| Fsim(2) = False ' Fixture Simulator : OFF
600| Trc(2) = "S12" ' Meas. parameter : S12
610| Ttl(2) = "[GSM1800] Tx-Antenna" ' Title
620| ' [Ch4]
630| Freq_star(3) = 1665000000# ' Start frequency : 1.665 GHz
640| Freq_stop(3) = 2015000000# ' Stop frequency : 2.015 GHz
650| Nop(3) = 101 ' Number of points : 101
660| N_tr(3) = 1 ' Number of traces : 1
670| Fsim(3) = False ' Fixture Simulator : OFF
680| Trc(3) = "S31" ' Meas. parameter : S31
690| Ttl(3) = "[GSM1800] Antenna-Rx" ' Title
700|
710| SCPI.SYSTem.PRESet
720| SCPI.DISPlay.Split = "D12_34"
730|
740| For Ch = 1 To 4
750| '
760| ' E5091A Setup
770| '
780| SCPI.SENSE(Ch).MULTiplexer(1).TSET9.Port1 = Port1(Ch - 1)
790| SCPI.SENSE(Ch).MULTiplexer(1).TSET9.Port2 = Port2(Ch - 1)
800| SCPI.SENSE(Ch).MULTiplexer(1).TSET9.Port3 = Port3(Ch - 1)
810| SCPI.SENSE(Ch).MULTiplexer(1).TSET9.Port4 = Port4(Ch - 1)
820| SCPI.SENSE(Ch).MULTiplexer(1).TSET9.OUTPUT.DATA = CLines(Ch - 1)
830| '
840| ' Measurement Condition
850| '
860| SCPI.SENSE(Ch).FREQuency.STARt = Freq_star(Ch - 1)
870| SCPI.SENSE(Ch).FREQuency.STOP = Freq_stop(Ch - 1)
880| SCPI.SENSE(Ch).SWEep.POINTs = Nop(Ch - 1)
890| SCPI.CALCulate(Ch).PARAmeter.Count = N_tr(Ch - 1)
900| If Fsim(Ch - 1) = True Then
910| SCPI.CALCulate(Ch).FSIMulator.STATe = True
920| SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice = Dev(Ch - 1)
```



```

930| SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SBALanced.PPORts = Tpl(Ch - 1)
940| SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter(1).STATe = Tr ue
950| SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter(1).SBALanced. DEFine = Trc(Ch - 1)
960| Else
970| SCPI.CALCulate(Ch).PARAmeter(1).DEFine = Trc(Ch - 1)
980| End If
990| SCPI.DISPlay.WINDow(Ch).TITLe.DATA = Ttl(Ch - 1)
1000| SCPI.DISPlay.WINDow(Ch).TITLe.STATe = True
1010| SCPI.INITiate(Ch).CONTInuous = True
1020| Next Ch
1030|
1040| SCPI.TRIGger.SEQuence.Source = "BUS"
1050| SCPI.SENSE.MULTiplexer(1).DISPlay.STATe = True
1060| SCPI.SENSE.MULTiplexer(1).STATe = True
1070| '
1080| ' Calibration
1090| '
1100| AnaPort = Array(1, 2)
1110| Res = ECal_Solt(1, 2, AnaPort, "A and T1")
1120| If Res <> 0 Then GoTo Prg_end
1130| AnaPort = Array(1, 3, 4)
1140| Res = ECal_Solt(2, 3, AnaPort, "A, R1+ and R1-")
1150| If Res <> 0 Then GoTo Prg_end
1160| AnaPort = Array(1, 2)
1170| Res = ECal_Solt(3, 2, AnaPort, "A and T2")
1180| If Res <> 0 Then GoTo Prg_end
1190| AnaPort = Array(1, 3)
1200| Res = ECal_Solt(4, 2, AnaPort, "A and R2+")
1210| If Res <> 0 Then GoTo Prg_end
1220| '
1230| ' Measurement
1240| '
1250| Meas_Start:
1260| MsgBox "Connect DUT.", vbOKOnly, "Measurement"
1270|
1280| SCPI.TRIGger.SEQuence.SINGle
1290| Dmy = SCPI.IEEE4882.OPC

```

E5071C

```
1300|
1310| For Ch = 1 To 4
1320| SCPI.DISPlay.WINDow(Ch).TRACe(1).Y.SCALe.AUTO
1330| Next Ch
1340|
1350| Buff = MsgBox("Do you make another measurement?", vbYesNo)
1360| If Buff = vbYes Then GoTo Meas_Start
1370|
1380| Prg_end:
1390| End Sub
1400|
1410| Function ECal_Solt(Ch As Long, Solt As Long, AnaPort As Variant, TsetPort As String) As Long
1420| Dim Err_info As Variant
1430| Dim Buff As Long
1440|
1450| Ecal_start:
1460| SCPI.IEEE4882.CLS
1470|
1480| MsgBox "Connect " + TsetPort + " to ECal Module."
1490|
1500| On Error GoTo Ecal_err
1510|
1520| Select Case Solt
1530| Case 1
1540| SCPI.SENSE(Ch).CORRection.COLLect.ECAL.SOLT1 = AnaPort(0)
1550| Case 2
1560| SCPI.SENSE(Ch).CORRection.COLLect.ECAL.SOLT2 = AnaPort
1570| Case 3
1580| SCPI.SENSE(Ch).CORRection.COLLect.ECAL.SOLT3 = AnaPort
1590| Case 4
1600| SCPI.SENSE(Ch).CORRection.COLLect.ECAL.SOLT4 = AnaPort
1610| End Select
1620|
1630| ECal_Solt = 0
1640| GoTo Ecal_end
1650|
1660| Ecal_err:
```

```

1670| Err_info = SCPI.SYSTem.Error
1680| Buff = MsgBox("Error: " + Err_info(1), vbRetryCancel)
1690| If Buff = vbRetry Then
1700| Resume Ecal_start
1710| Else
1720| ECal_Solt = Err_info(0)
1730| Resume Ecal_end
1740| End If
1750|
1760| Ecal_end:
1770| End Function

```

Executing Power Calibration

- [Overview](#)
- [Program Overview](#)
- [Program Description](#)

Other topics about Application Programs

Overview

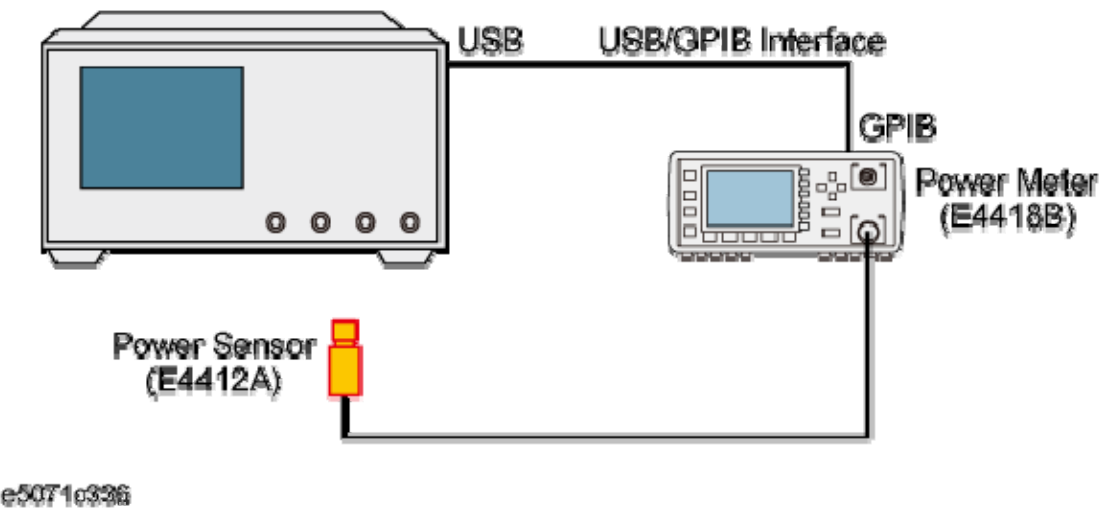
The **pow_cal.vba** shows a sample program (VBA program) for executing power calibration using the E4418B power meter and the E4412A power sensor. This VBA program consists of the following modules:

Object name	Module type	Description
mdlPowCal	Standard module	Executes power calibration
Module1 Module2	Standard module	Definition files for using VISA library

Program Overview

The program connects the E5071C and the E4418B (GPIB address: 13) through the USB/GPIB interface as shown below. Then, the program executes the zero adjustment and calibration of the power sensor (E4412A) connected to the power meter (E4418B) as necessary. Finally, it executes the power calibration of the E5071C and saves the obtained power calibration data array into a file.

Connection between E5071C and power meter



Program Description

When you run this VBA program, reset is performed, the GPIB address of the power meter to be controlled and the measurement conditions of the E5071C are set, and the message **"Do you want to zero and calibrate the power sensor?"** is displayed. If you want to execute the zero adjustment and the calibration of the power sensor, click **Yes**, otherwise click **No**.

If you want to execute the zero adjustment and calibration of the power sensor, follow the displayed messages to connect the power sensor connected to the A channel of the power meter to the POWER RF port of the power meter and click **OK**. When the zero adjustment and calibration of the power sensor is complete, the message **"Zeroing and calibrating the power sensor is complete"** is displayed.

When a message that prompts you to connect the power sensor to port 1 of the E5071C is displayed, make the connection and then click **OK**. The power calibration data measurement is executed immediately and the obtained power calibration data array is saved in a file named **CORR_DATA**.

The power calibration execution program (object name: mdlPowCal) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Lines 100 to 150

Assign the sweep type (power sweep), the number of points (41), the power range (-20 to +12 dBm), the sweep start value (-20 dBm), the sweep stop value (-10 dBm), and the fixed frequency (1 GHz) to the Swp_type, Nop, Pow_rang, Start_p, Stop_p, and Cw_freq variables, respectively.

Lines 160 to 170

Assign the number of power calibration data measurements for each measurement point (4) and the limit value of the power calibration data array (10 dBm) into the Num_avg and Limit variables, respectively.

Line 200

Returns the E5071C to the preset state.

Line 230

Sets the power meter's GPIB address (13) in the E5071C.

Lines 260 to 310

For channel 1, these lines assign the sweep type to the Swp_type variable, the number of points to the Nop variable, the power range to the Pow_rang variable, the sweep start value to the Start_p variable, the

E5071C

sweep stop value to the Stop_p variable, and the fixed frequency to the Cw_freq variable.

Lines 340 to 370

Display a message asking whether to execute the zero adjustment and calibration of the power sensor. If the **Yes** button is clicked, these lines call the procedure: Control_PowerMeter (Line 900 to Line 1340). On the other hand, if the **No** button is clicked, the program skips the zero adjustment and calibration of the power sensor.

Lines 410 to 420

Clear the error queue. Then, these lines prompt you to connect the power sensor to port 1 of the E5071C and wait for the **OK** button to be clicked after the connection.

Lines 450 to 470

For port 1 of channel 1, these lines set the number of power calibration data measurements for each measurement point to the Num_avg variable and then start the measurement of the power calibration data and wait for the completion of the measurement.

Line 500

Reads out an error that has occurred in the E5071C during the measurement of the power calibration data and sets it in the Err variable.

Lines 510 to 590

If no error has occurred, these lines read out the power calibration data array and set it in the Corr_data variable. In addition, the program uses the Limit_Test function to check whether the read out power calibration data array exceeds the specified limit value. If the limit value is exceeded, the return value of the Limit_Test function, False, is returned. Then, a message is displayed asking whether to perform the power calibration again. Click the **Yes** button to return to the start of the power calibration data measurement. Click the **No** button to terminate the program.

Lines 610 to 660

If an error occurs, display an error message and a message asking whether to execute the power calibration again. Click the **Yes** button to return to the start of the power calibration data measurement. Click the **No** button to terminate the program.

Lines 690 to 790

Write the read out power calibration data array into a file named **CORR_DATA**. Then, these lines display a message that saving into a file has been successfully completed.

Procedure: Control_PowerMeter (lines 900 to 1340)

Lines 990 to 1000

Initialize and start up the VISA system and output the startup information to the Defrm variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 1260 to 1300).

Lines 1030 to 1040

Establish the connection to the power meter in use (GPIB address: 13) and output the connection information to the E4418 variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 1260 to 1300).

Lines 1070 to 1090

Return the power meter to the preset state through VISA and clear the status byte register and the standard event status register. Then, these lines enable the standard event status register's bit 0.

Line 1100

Prompts you to connect the power sensor to the POWER REF port of the power meter and waits for the **OK** button to be clicked after the connection.

Lines 1110 to 1120

Execute the zero adjustment and calibration of the power sensor through VISA. These lines make the setting so that 1 is set to bit 0 of the standard event status register when all pending operations are completed.

Lines 1130 to 1190

Retrieve the value of the status byte register through VISA and set it into the Rgst variable. These lines set the AND of the read-out value and 32 (the value in which only bit 5 is 1) into the Rslt variable and display a message that the zero adjustment and calibration of the power sensor is completed when Rslt becomes 1 (when the zero adjustment and calibration of the power sensor is completed).

Line 1220

Breaks the communication and terminates the VISA system.

Lines 1260 to 1300

If an error occurs in a VISA function, these lines display the details of the error and terminate the program.

Function: Limit_Test (Lines 1360 to 1530)

Lines 1420 to 1470

If the absolute value of the read out power calibration data array exceeds the specified limit value, these lines turn off the power calibration function and return the value of False. Otherwise, the value of True is returned.

Measurement of power calibration (object name: mdlPowCal)

```

10| Sub Main()
20|
30| Dim Swp_type As String, File As String
40| Dim Start_p As Double, Stop_p As Double, Cw_freq As Double, Limit As Double
50| Dim Nop As Long, Pow_rang As Long, Num_avg As Long, Data_size As Long, Buff As Long, Dmy
   As Long
60| Dim Corr_data As Variant, Err As Variant
70| Dim Verifier As Boolean
80| Dim FileNo As Integer, I As Integer
90|
100| Swp_type = "POW" 'Sweep type : POWER
110| Nop = 41 'Number of points : 41
120| Pow_rang = 0 'Power range : -20 to +12 dBm
130| Start_p = -20# 'Start power : -20 dBm
140| Stop_p = -10# 'Stop power : -10 dBm
150| Cw_freq = 1000000000# 'CW frequency : 1 GHz
160| Num_avg = 4 'Number of averaging : 4
170| Limit = 10# 'limit for corrected data : 10 dBm
180|
190| ""Presetting the ENA
200| SCPI.SYSTem.PRESet
210|
220| ""Setting GPIB address of the power meter to ENA
230| SCPI.SYSTem.COMMunicate.GPIB.PMETer.address = 13
240|
250| ""Setting measurement conditions
260| SCPI.SENSE(1).SWEep.TYPE = Swp_type
270| SCPI.SENSE(1).SWEep.POINts = Nop

```



```

280| SCPI.Source(1).POWER.ATTenuation.DATA = Pow_rang
290| SCPI.Source(1).POWER.START = Start_p
300| SCPI.Source(1).POWER.STOP = Stop_p
310| SCPI.SENSE(1).FREQuency.CW = Cw_freq
320|
330| "Performing a calibration in the power meter
340| Buff = MsgBox("Do you perform zeroing and calibrating the power sensor?", vbYesNo, "Power
meter calibration")
350| If Buff = vbYes Then
360| Control_PowerMeter
370| End If
380|
390| Meas_Start:
400| "Connecting the power sensor to the port 1 in the ENA
410| SCPI.IEEE4882.CLS
420| MsgBox "Set the power sensor connected to the port 1 in the ENA.", vbOKOnly, "Power meter
calibration"
430|
440| "Performing power calibration measurement
450| SCPI.Source(1).POWER.PORT(1).CORRection.COLlect.AVERage.count = Num_avg
460| SCPI.Source(1).POWER.PORT(1).CORRection.COLlect.ACQuire = "ASEN "
470| Dmy = SCPI.IEEE4882.OPC
480|
490| "Error handling at power meter calibration
500| Err = SCPI.SYSTem.Error
510| If Err(0) = 0 Then
520| Corr_data = SCPI.Source(1).POWER.PORT(1).CORRection.DATA
530| Verifier = Limit_Test(Nop, Limit, Corr_data)
540| If Verifier = False Then
550| Buff = MsgBox("Do you perform the power meter calibration me asurement again?", vbYesNo,
"Power meter calibration")
560| If Buff = vbYes Then GoTo Meas_Start
570| If Buff = vbNo Then GoTo Prog_Stop
580| End If
590| MsgBox "Power meter calibration measurement is complete.", vbOKOnly, "Power meter calibration"
600|
610| Else
620| MsgBox "Error: " & Err(1)

```

E5071C

```
630| Buff = MsgBox("Do you perform the power meter calibration meas urement again?", vbYesNo,
"Power meter calibration")
640| If Buff = vbYes Then GoTo Meas_Start
650| If Buff = vbNo Then GoTo Prog_Stop
660| End If
670|
680| ""Installing the corrected data to a file
690| FileNo = FreeFile
700| File = "CORR_DATA"
710|
720| Open File For Output As FileNo
730|
740| For I = 0 To Nop - 1
750| Write #FileNo, Val(Corr_data(I))
760| Next I
770| Close #FileNo
780|
790| MsgBox "Installing the corrected data to the file is DONE.", vbO KOnly, "Power meter calibration"
800|
810| GoTo Prog_End
820|
830| Prog_Stop:
840| MsgBox "Program Interruption", vbOKOnly, "Power meter calibratio n"
850|
860| Prog_End:
870|
880| End Sub
890|
900| Private Sub Control_PowerMeter()
910|
920| Dim Status As Long 'VISA function status return
930| Dim Defrm As Long 'Session to default resource code
940| Dim E4418 As Long 'Session to power meter
950| Dim Rslt As Integer
960| Dim Rgst As String * 10
970|
980| ""Initializing the VISA system
```

```

990| Status = viOpenDefaultRM(Defrm)
1000| If (Status <> VI_SUCCESS) Then GoTo VisaErrorHandler
1010|
1020| ""Opening the session to the power meter
1030| Status = viOpen(Defrm, "GPIB0::13::INSTR", 0, 0, E4418)
1040| If (Status <> VI_SUCCESS) Then GoTo VisaErrorHandler
1050|
1060| ""Zeroing and calibrating the power meter
1070| Status = viVPrintf(E4418, "SYST:PRES" & Chr$(10), 0)
1080| Status = viVPrintf(E4418, "*CLS" & Chr$(10), 0)
1090| Status = viVPrintf(E4418, "ESE 1" & Chr$(10), 0)
1100| MsgBox "Set the power sensor connected to the POWER REF port in the power meter.",
vbOKOnly, "Power meter calibration"
1110| Status = viVPrintf(E4418, "CAL1:ALL" & Chr$(10), 0)
1120| Status = viVPrintf(E4418, "*OPC" & Chr$(10), 0)
1130| Do While Rslt = 0
1140| Status = viVPrintf(E4418, "*STB?" & Chr$(10), 0)
1150| Status = viVScanf(E4418, "%t", Rgst)
1160| Rslt = CInt(CInt(Rgst) And 32)
1170| Loop
1180|
1190| MsgBox "Zeroing and Calibrating the power sensor is complete.", vbOKOnly, "Power meter
calibration"
1200|
1210| ""Closing the resource manager session
1220| Call viClose(Defrm)
1230|
1240| GoTo Prog_End
1250|
1260| VisaErrorHandler:
1270| Dim VisaErr As String * 200
1280| Call viStatusDesc(Defrm, Status, VisaErr)
1290| MsgBox "Error : " & VisaErr, vbExclamation
1300| End
1310|
1320| Prog_End:
1330|
1340| End Sub

```

E5071C

```
1350|
1360| Function Limit_Test(Nop As Long, Limit As Double, Corr_data As Var iant) As Boolean
1370|
1380| Dim I As Integer
1390|
1400| For I = 0 To Nop - 1
1410|
1420| If Abs(Corr_data(I)) > Limit Then
1430| SCPI.Source(1).POWer.PORT(1).CORRection.STATe = False
1440| MsgBox "The corrected data is out of limit!", vbExclamation, "Power meter calibration"
1450| Limit_Test = False
1460| Exit Function
1470| End If
1480|
1490| Next I
1500|
1510| Limit_Test = True
1520|
1530| End Function
```

Connecting Hard Disk of External PC (shared folder)

- [Overview](#)
- [Using VBA Program](#)
- [Description of VBA Program](#)

Other topics about Application Programs

Overview

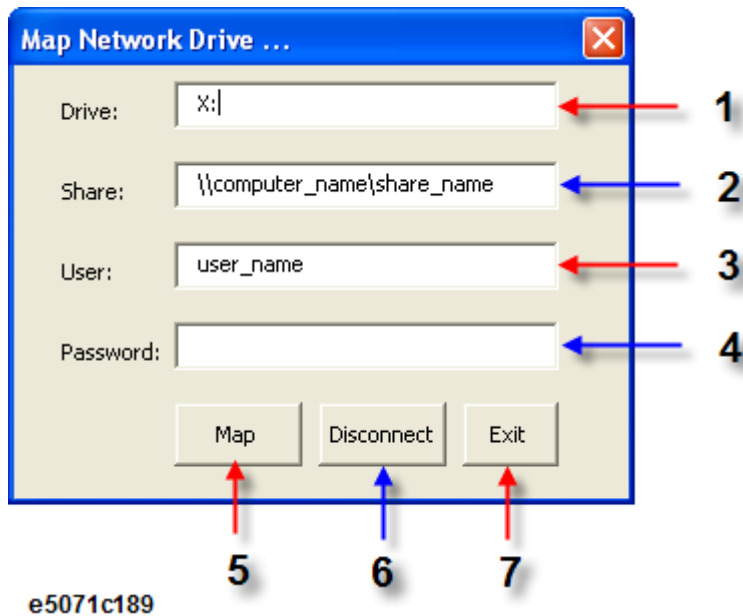
The **map_drive.vba** shows a sample program (VBA program) that demonstrates how to connect a hard disk (a shared folder) of an external PC to the E5071C. This VBA program consists of the following modules:

Object name	Module type	Description
frmMapDrive	User form	Connects or disconnects a hard disk
Module1	Standard module	Displays frmMapDrive

Using VBA Program

Load the **map_drive.vba** and press **Macro Run** key. The following macro appears.

Shared folder connection macro



Connecting (Mapping)

Enter the drive letter for the shared folder (1), share name of the shared folder (2), user name (3), password (4), and then click **Map** (5).

NOTE

Consult your network administrator and enter the settings in the same way as done in the Windows PC operating system. If you enter an incorrect setting, an error might occur and the program might be interrupted.

Disconnecting

1. Enter the drive letter for the shared folder (1), and then click **Disconnect** (6).
2. Click **Exit** (7) to exit from the program.

Description of VBA Program

The program (object name: frmMapDrive) is described in detail below:

Sub CommandButton1_Click

This procedure is called when the user clicks the **Map** button. It checks whether the drive letter is used by using the IsDriveNameInUse procedure. Then the procedure connects the shared folder using the MapDrive procedure if the drive letter is not used or otherwise displays a message to show the drive letter is used.

Sub CommandButton2_Click

This procedure is called when the user clicks the **Disconnect** button. The procedure disconnects the shared folder by using the DisconnectDrive procedure.

Function IsDriveNameInUse

This procedure checks if the txtDrive.Text (the drive letter specified by 1) is used.

Sub MapDrive

This procedure connects the shared folder as the txtDrive.Text (the drive letter specified by 1) drive by using the parameters: txtShare.Text (the share name specified by 2), txtUser.Text (the user name specified by 3), and txtPasswd.Text (the password specified by 4).

Sub DisconnectDrive

This procedure disconnects the txtDrive.Text (the drive letter specified by 1) drive.

Sub CommandButton3_Click

This procedure is called when the user clicks the **Exit** button. This procedure ends the program.

Connecting the hard disk of an external PC (Object name: frmMapDrive)

```
Private Sub CommandButton1_Click()
```

```
If Not IsDriveNameInUse Then
```

```
Call MapDrive
```

```
Else
```

```
MsgBox "Drive "" & txtDrive.Text & "" is Already used", vb Critical
```

```
End If
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
Call DisconnectDrive
```

```
End Sub
```

```
Private Function IsDriveNameInUse() As Boolean
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
IsDriveNameInUse = fso.DriveExists(txtDrive.Text)
```

```
End Function
```

```
Private Sub MapDrive()
```

```
Set network = CreateObject("wscript.network")
```

```
Call network.MapNetworkDrive(txtDrive.Text, txtShare.Text, vbFalse, txtUser.Text, txtPasswd.Text)
```

```
End Sub
```

```
Private Sub DisconnectDrive()
```

```
Set network = CreateObject("wscript.network")
```

```
network.RemoveNetworkDrive txtDrive.Text
```

```
End Sub
```

```
Private Sub CommandButton3_Click()
```

```
Unload Me
```

```
End Sub
```

Complex Operation Library

Complex Operation Library

By using the complex operation library, you can perform operations of complex numbers.

Data of the complex type

In the complex operation library, you can use the complex type (Complex) as a data type. Data of the complex type consists of a real part (.real) and an imaginary part (.imag) as shown in the following example.

```
Dim Num as Complex
Num.real=1.0
Num.imag=2.0
```

List of procedures

The following table lists the procedures included in the complex operation library.

Procedure name	Function
ComplexSet(x,y)	Sets a complex number. (Specify a real part and an imaginary part.)
ComplexPolar(x,y)	Sets a complex number. (Specify an absolute value and a phase angle.)
ComplexSetArray(x)	Converts a variant type or double floating point type array to a complex type array.
ComplexAdd(x,y)	Returns the result of the addition.
ComplexSub(x,y)	Returns the result of the subtraction.
ComplexMul(x,y)	Returns the result of the multiplication.
ComplexDiv(x,y)	Returns the result of the division.
ComplexAbs(x)	Returns the absolute value.
ComplexArg(x)	Returns the phase angle.
ComplexNorm(x)	Returns the square of the absolute value.
ComplexConj(x)	Returns the conjugate complex number.
ComplexCos(x)	Returns the cosine.

ComplexCosh(x)	Returns the hyperbolic cosine.
ComplexSin(x)	Returns the sine.
ComplexSinh(x)	Returns the hyperbolic sine.
ComplexExp(x)	Returns e^x .
ComplexLog(x)	Returns the natural logarithm.
ComplexLog10(x)	Returns the common logarithm.
ComplexSqrt(x)	Returns the square root.

Sample Program

```

:
:

```

```

Dim Dmy As Long
Dim s21_raw As Variant
Dim s31_raw As Variant
Dim s21_Comp As Complex
Dim s31_Comp As Complex
Dim trAce_ratio_comp As Complex
Dim trAce_ratio(401) As Double

```

```

SCPI.DISPlay.Split = "D1"
SCPI.DISPlay.WINDow(1).Split = "D12_34"
SCPI.CALCulate(1).PARAmeter.Count = 2
SCPI.CALCulate(1).PARAmeter(1).DEFine = "s21"
SCPI.CALCulate(1).PARAmeter(2).DEFine = "s31"
SCPI.SENSE(1).SWEep.POINts = 201

```

```

:
:
:

```

```

SCPI.TRIGger.SEQuence.Source = "bus"
SCPI.TRIGger.SEQuence.SINGLE
Dmy = SCPI.IEEE4882.OPC

```

E5071C

```
"" Get corrected data array
SCPI.CALCulate(1).PARAmeter(1).SElect
s21_raw = SCPI.CALCulate(1).SElected.DATA.SDATa
SCPI.CALCulate(1).PARAmeter(2).SElect
s31_raw = SCPI.CALCulate(1).SElected.DATA.SDATa

For i = 0 To 200

"" Copy corrected data array to the complex data array
"" to take advantage of complex operation library
s21_Comp = ComplexSet(s21_raw(2 * i), s21_raw(2 * i + 1))
s31_Comp = ComplexSet(s31_raw(2 * i), s31_raw(2 * i + 1))

"" Calculate the ratio of S31 and S21
"" S31/S21
trAce_ratio_comp = ComplexDiv(s31_Comp, s21_Comp)

trAce_ratio(2 * i) = trAce_ratio_comp.real
trAce_ratio(2 * i + 1) = trAce_ratio_comp.imag

Next i

SCPI.CALCulate(1).PARAmeter.Count = 4

"" Write "S31/S21" data to corrected data array for the trace 3 (LogMag)
SCPI.CALCulate(1).PARAmeter(3).SElect
SCPI.CALCulate(1).SElected.Format = "MLOG"
SCPI.CALCulate(1).SElected.DATA.SDATa = trAce_ratio

"" Write "S31/S21" data to corrected data array for the trace 4 (Phase)
SCPI.CALCulate(1).PARAmeter(4).SElect
SCPI.CALCulate(1).SElected.Format = "PHASe"
SCPI.CALCulate(1).SElected.DATA.SDATa = trAce_ratio

:
:
:
```

Complex Operation Library

By using the complex operation library, you can perform operations of complex numbers.

Data of the complex type

In the complex operation library, you can use the complex type (Complex) as a data type. Data of the complex type consists of a real part (.real) and an imaginary part (.imag) as shown in the following example.

```
Dim Num as Complex
Num.real=1.0
Num.imag=2.0
```

List of procedures

The following table lists the procedures included in the complex operation library.

Procedure name	Function
ComplexSet(x,y)	Sets a complex number. (Specify a real part and an imaginary part.)
ComplexPolar(x,y)	Sets a complex number. (Specify an absolute value and a phase angle.)
ComplexSetArray(x)	Converts a variant type or double floating point type array to a complex type array.
ComplexAdd(x,y)	Returns the result of the addition.
ComplexSub(x,y)	Returns the result of the subtraction.
ComplexMul(x,y)	Returns the result of the multiplication.
ComplexDiv(x,y)	Returns the result of the division.
ComplexAbs(x)	Returns the absolute value.
ComplexArg(x)	Returns the phase angle.
ComplexNorm(x)	Returns the square of the absolute value.
ComplexConj(x)	Returns the conjugate complex number.
ComplexCos(x)	Returns the cosine.
ComplexCosh(x)	Returns the hyperbolic cosine.

ComplexSin(x)	Returns the sine.
ComplexSinh(x)	Returns the hyperbolic sine.
ComplexExp(x)	Returns e^x .
ComplexLog(x)	Returns the natural logarithm.
ComplexLog10(x)	Returns the common logarithm.
ComplexSqrt(x)	Returns the square root.

Sample Program

```

:
:

```

```

Dim Dmy As Long
Dim s21_raw As Variant
Dim s31_raw As Variant
Dim s21_Comp As Complex
Dim s31_Comp As Complex
Dim trAce_ratio_comp As Complex
Dim trAce_ratio(401) As Double

```

```

SCPI.DISPlay.Split = "D1"
SCPI.DISPlay.WINDow(1).Split = "D12_34"
SCPI.CALCulate(1).PARAmeter.Count = 2
SCPI.CALCulate(1).PARAmeter(1).DEFine = "s21"
SCPI.CALCulate(1).PARAmeter(2).DEFine = "s31"
SCPI.SENSE(1).SWEep.POINts = 201

```

```

:
:
:

```

```

SCPI.TRIGger.SEQuence.Source = "bus"
SCPI.TRIGger.SEQuence.SINGLE
Dmy = SCPI.IEEE4882.OPC

```

```

"" Get corrected data array

```

```

SCPI.CALCulate(1).PARAmeter(1).SElect
s21_raw = SCPI.CALCulate(1).SElected.DATA.SDATa
SCPI.CALCulate(1).PARAmeter(2).SElect
s31_raw = SCPI.CALCulate(1).SElected.DATA.SDATa

For i = 0 To 200

    "" Copy corrected data array to the complex data array
    "" to take advantage of complex operation library
    s21_Comp = ComplexSet(s21_raw(2 * i), s21_raw(2 * i + 1))
    s31_Comp = ComplexSet(s31_raw(2 * i), s31_raw(2 * i + 1))

    "" Calculate the ratio of S31 and S21
    "" S31/S21
    trAce_ratio_comp = ComplexDiv(s31_Comp, s21_Comp)

    trAce_ratio(2 * i) = trAce_ratio_comp.real
    trAce_ratio(2 * i + 1) = trAce_ratio_comp.imag

Next i

SCPI.CALCulate(1).PARAmeter.Count = 4

"" Write "S31/S21" data to corrected data array for the trace 3 (LogMag)
SCPI.CALCulate(1).PARAmeter(3).SElect
SCPI.CALCulate(1).SElected.Format = "MLOG"
SCPI.CALCulate(1).SElected.DATA.SDATa = trAce_ratio

"" Write "S31/S21" data to corrected data array for the trace 4 (Phase)
SCPI.CALCulate(1).PARAmeter(4).SElect
SCPI.CALCulate(1).SElected.Format = "PHASe"
SCPI.CALCulate(1).SElected.DATA.SDATa = trAce_ratio

:
:

```

Procedure Reference

ComplexAbs(*x*)

Syntax

Result = ComplexAbs(*x*)

Description

Returns the absolute value of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double
a = ComplexSet(1.5, 2.0)
b = ComplexAbs(a)
```

ComplexAdd(x,y)**Syntax**

Result = ComplexAdd(x,y)

Description

Returns the result (x+y) of the addition of a complex number x and another y.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexAdd(a, b)
```

ComplexArg(x)

Syntax

Result = ComplexArg(*x*)

Description

Returns the phase angle (radian) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double, c As Double, pi As Double
a = ComplexSet(1.5, 2.0)
b = ComplexArg(a)
pi = 3.14159265
c = b * 180 / pi    ' radian -> degree
```


ComplexConj(*x*)**Syntax**

Result = ComplexConj(*x*)

Description

Returns the conjugate complex number of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex  
a = ComplexSet(1.5, 2.0)  
b = ComplexConj(a)
```

E5071C

ComplexCos(*x*)

Syntax

Result = ComplexCos(*x*)

Description

Returns the cosine (cos(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex  
a = ComplexSet(1.5, 2.0)  
b = ComplexCos(a)
```

ComplexCosh(*x*)**Syntax**

Result = ComplexCosh(*x*)

Description

Returns the hyperbolic cosine (cosh(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexCosh(a)
```

ComplexDiv(x,y)

Syntax

Result = ComplexDiv(*x*,*y*)

Description

Returns the result (x/y) of the division of a complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexDiv(a, b)
```

ComplexExp(*x*)

Syntax

Result = ComplexExp(*x*)

Description

Returns *ex*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexExp(a)
```

E5071C

ComplexLog(x)

Syntax

Result = ComplexLog(x)

Description

Returns the natural logarithm ($\log(x)$) of a complex number x .

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexLog(a)
```

ComplexLog10(x)**Syntax**

Result = ComplexLog(x)

Description

Returns the common logarithm (log
10
(x)) of a complex number x.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexLog10(a)
```

ComplexMul(x,y)

Syntax

Result = ComplexMul(*x*,*y*)

Description

Returns the result (*x*\ *y*) of the multiplication of a complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexMul(a, b)
```


ComplexNorm(*x*)**Syntax**

Result = ComplexNorm(*x*)

Description

Returns the square of the absolute value of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double
a = ComplexSet(1.5, 2.0)
b = ComplexNorm(a)
```

ComplexPolar(x,y)

Syntax

$z = \text{ComplexPolar}(x,y)$

Description

Sets a complex number to a complex type variable z . Specify a complex number with an absolute value x and a phase angle y (*radian*).

Data type

x : Double precision floating point type (Double)

y : Double precision floating point type (Double)

z : Complex type (Complex)

Example of use

```
Dim a As Complex, pi As Double
pi = 3.14159265
a = ComplexPolar(2.5, 60 * pi / 180)
```

ComplexSet(x,y)**Syntax**

```
z = ComplexSet(x,y)
```

Description

Sets a complex number to a complex type variable *z*. Specify a complex number with a real part *x* and an imaginary part *y*. (Sets *x* and *y* to *z.real* and *z.imag* respectively.)

Data type

x : Double precision floating point type (Double)

y : Double precision floating point type (Double)

z : Complex type (Complex)

Example of use

```
Dim a as Complex
```

```
a = ComplexSet(1.5, 2.0)
```

ComplexSetArray(x)

Syntax

$y = \text{ComplexSetArray}(x)$

Description

Converts a variant type or double floating point type array x that contains complex numbers using 2 elements to store each complex number in the order of the real part and imaginary part to a complex type array y .

Data type

x : Variant type (Variant) array or Double precision floating point type (Double) array

y : Complex type (Complex) array

Example of use

```
Dim a as Variant, b as Complex
a = SCPI.CALCulate(1).SElected.DATA.SDATa
b = ComplexSetArray(a)
```

ComplexSin(*x*)**Syntax**

Result = ComplexSin(*x*)

Description

Returns the sine (sin(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex  
a = ComplexSet(1.5, 2.0)  
b = ComplexSin(a)
```

E5071C

ComplexSinh(*x*)

Syntax

Result = ComplexSinh(*x*)

Description

Returns the hyperbolic sine ($\sinh(x)$) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSinh(a)
```

ComplexSqrt(*x*)**Syntax**

Result = ComplexSqrt(*x*)

Description

Returns the square root () of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex  
a = ComplexSet(1.5, 2.0)  
b = ComplexSqrt(a)
```

ComplexSub(x,y)

Syntax

Result = ComplexSub(x,y)

Description

Returns the result ($x - y$) of the subtraction of a complex number x and another y .

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexSub(a, b)
```


Waveform Analysis Library

Waveform (Ripple) Analysis Library

- [Overview](#)
- [Flow of Programming using Ripple Analysis Library](#)
- [Condition Setting before using Ripple Analysis Library](#)
- [List of Ripple Analysis Library](#)
- [Sample Program](#)

Other topics about VBA Programming

Overview

By combining the COM objects provided for the E5071C and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

Flow of Programming using Ripple Analysis Library

Below table shows the flow of program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

1. Condition setting before using the ripple analysis library
 - Specifying the analysis range
 - Setting the peak definition
2. Using the ripple analysis library

Condition Setting before using Ripple Analysis Library

Since the analysis conditions are not specified in the ripple analysis library, before using the procedure for ripple analysis, set up the analysis range and the peak definition using COM objects.

Specifying the Analysis Range

Use the following COM objects to specify the analysis range for ripple analysis.

- SCPI.CALCulate(Ch).SElected.FUNction.DOMain.START
- SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STOP
- SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STATe
- SCPI.CALCulate(Ch).SElected.FUNction.DOMain.COUPLE

Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis.

- SCPI.CALCulate(Ch).SElected.FUNction.PEXCursion

- SCPI.CALCulate(Ch).SElected.FUNCTION.PPOLarity

List of Ripple Analysis Library

Use the provided procedures for ripple analysis to analyze the ripple of waveforms and output the result. All procedures perform analysis only within the stimulus range for the specified channel.

Function Name	Description
MaxPeakToPeak(Chan)	Returns the maximum value of the difference between a positive peak and a negative peak
MaxRightGap(Chan)	Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.
MaxLeftGap(Chan)	Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.
MaxGap(Chan)	Returns the maximum value of the difference between a positive peak and its adjacent negative peak.
MaxEnvelopeGap(Chan)	Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.
GapMean(Chan)	Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRippleValue(Chan)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRipplePoint(Chan,Stim)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks and the stimulus value (<i>Stim</i>) of the valley of the ripple.
Pole(Chan,D,LeftStim,LeftValue,RightStim,RightValue)	Returns the values (<i>LeftValue</i> and <i>RightValue</i>) and the stimulus values (<i>LeftStimulus</i> and <i>RightStimulus</i>) of the right and left negative peaks detected first below the specified value (<i>D</i>) relative to the maximum value.
FirstRightGap(Chan)	Returns the difference between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.

FirstLeftGap(Chan)	Returns the difference between the positive peak detected first when searched from the right edge toward the left edge and its left adjacent negative peak.
FirstRightInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its left adjacent negative peak.

Sample Program

Here is a simple sample program using the ripple analysis procedures.

```
Sub Sample()
```

```
Dim Val As Double  (1)
```

```
SCPI.CALCulate(1).SElected.FUNCTION.PEXCursion = 1.5  (2)
SCPI.CALCulate(1).SElected.FUNCTION.PPOLarity = "BOTH"  (2)
SCPI.CALCulate(1).SElected.FUNCTION.DOMain.START = 935E6  (3)
SCPI.CALCulate(1).SElected.FUNCTION.DOMain.STOP = 960E6  (3)
SCPI.CALCulate(1).SElected.FUNCTION.DOMain.STATe = True  (3)
```

```
.
```

```
Val = MaxPeakToPeak(1)  (4)
```

```
End Sub
```

Let us break down the code into a number of blocks and see what they do.

1. Defines a variable Val as Double.
2. Sets the lower limit of the peak excursion value and polarity for the peak search to 1.5 and both of positive peak and negative peak, respectively.
3. Sets the analysis range for channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitutes the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

Waveform (Ripple) Analysis Library

- [Overview](#)
- [Flow of Programming using Ripple Analysis Library](#)
- [Condition Setting before using Ripple Analysis Library](#)
- [List of Ripple Analysis Library](#)
- [Sample Program](#)

Other topics about VBA Programming

Overview

By combining the COM objects provided for the E5071C and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

Flow of Programming using Ripple Analysis Library

Below table shows the flow of program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

1. Condition setting before using the ripple analysis library
 - Specifying the analysis range
 - Setting the peak definition
2. Using the ripple analysis library

Condition Setting before using Ripple Analysis Library

Since the analysis conditions are not specified in the ripple analysis library, before using the procedure for ripple analysis, set up the analysis range and the peak definition using COM objects.

Specifying the Analysis Range

Use the following COM objects to specify the analysis range for ripple analysis.

- SCPI.CALCulate(Ch).SElected.FUNcTion.DOMain.START
- SCPI.CALCulate(Ch).SElected.FUNcTion.DOMain.STOP
- SCPI.CALCulate(Ch).SElected.FUNcTion.DOMain.STATe
- SCPI.CALCulate(Ch).SElected.FUNcTion.DOMain.COUPLe

Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis.

- SCPI.CALCulate(Ch).SElected.FUNcTion.PEXCursion
- SCPI.CALCulate(Ch).SElected.FUNcTion.PPOLarity

List of Ripple Analysis Library

Use the provided procedures for ripple analysis to analyze the ripple of waveforms and output the result. All procedures perform analysis only within the stimulus range for the specified channel.

Function Name	Description
MaxPeakToPeak(Chan)	Returns the maximum value of the difference between a positive peak and a negative peak
MaxRightGap(Chan)	Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.
MaxLeftGap(Chan)	Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.
MaxGap(Chan)	Returns the maximum value of the difference between a positive peak and its adjacent negative peak.
MaxEnvelopeGap(Chan)	Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.
GapMean(Chan)	Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRippleValue(Chan)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRipplePoint(Chan,Stim)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks and the stimulus value (<i>Stim</i>) of the valley of the ripple.
Pole(Chan,D,LeftStim,LeftValue,RightStim,RightValue)	Returns the values (<i>LeftValue</i> and <i>RightValue</i>) and the stimulus values (<i>LeftStimulus</i> and <i>RightStimulus</i>) of the right and left negative peaks detected first below the specified value (<i>D</i>) relative to the maximum value.
FirstRightGap(Chan)	Returns the difference between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.

FirstLeftGap(Chan)	Returns the difference between the positive peak detected first when searched from the right edge toward the left edge and its left adjacent negative peak.
FirstRightInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its left adjacent negative peak.

Sample Program

Here is a simple sample program using the ripple analysis procedures.

```
Sub Sample()
```

```
Dim Val As Double  (1)
```

```
SCPI.CALCulate(1).SElected.FUNCTION.PEXCursion = 1.5  (2)
SCPI.CALCulate(1).SElected.FUNCTION.PPOLarity = "BOTH"  (2)
SCPI.CALCulate(1).SElected.FUNCTION.DOMain.START = 935E6  (3)
SCPI.CALCulate(1).SElected.FUNCTION.DOMain.STOP = 960E6  (3)
SCPI.CALCulate(1).SElected.FUNCTION.DOMain.STATe = True  (3)
```

```
.
```

```
Val = MaxPeakToPeak(1)  (4)
```

```
End Sub
```

Let us break down the code into a number of blocks and see what they do.

1. Defines a variable Val as Double.
2. Sets the lower limit of the peak excursion value and polarity for the peak search to 1.5 and both of positive peak and negative peak, respectively.
3. Sets the analysis range for channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitutes the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

Procedure Reference

FirstLeftGap(*Chan*)

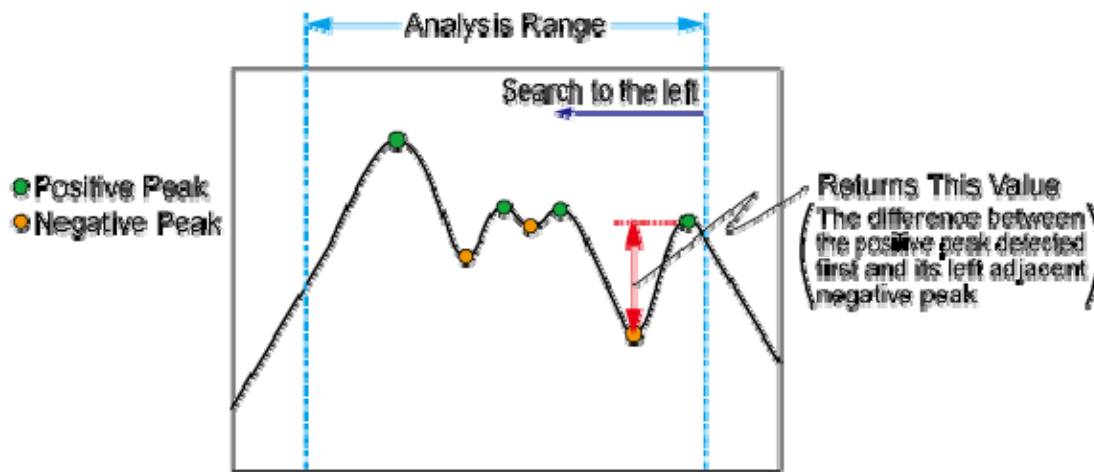
Syntax

Value = FirstLeftGap(*Chan*)

Description

Returns the response difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

FirstLeftGap



e5071c431

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

E5071C

Return value

Parameter	<i>Value</i>
Description	Returns the response difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

Dim Value As Double

Value = FirstLeftGap(1)

MsgBox "First Left Gap =" & Value

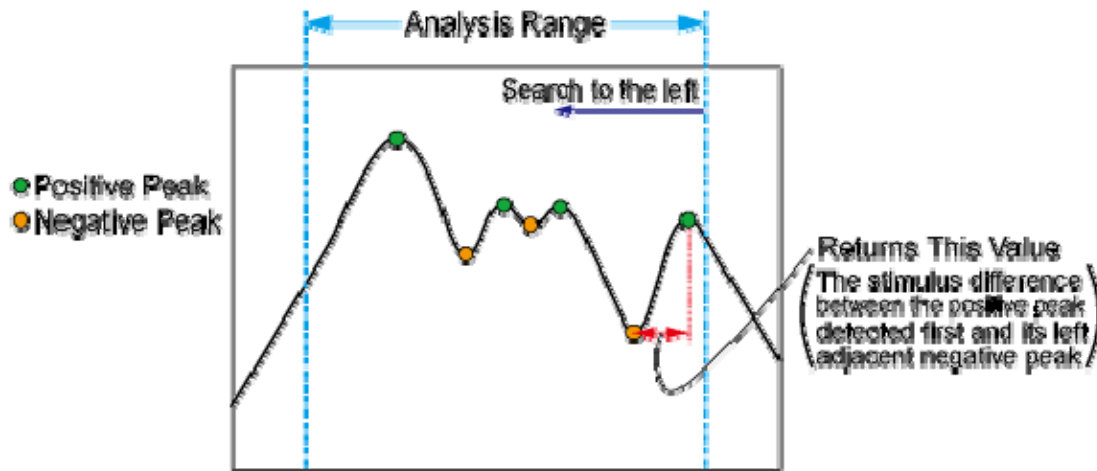
FirstLeftInterval(*Chan*)

Syntax

Value = FirstLeftInterval(*Chan*)

Description

Returns the stimulus difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

FirstLeftInterval

e5071c432

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

E5071C

Return value

Parameter	<i>Value</i>
Description	Returns the stimulus difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

Dim Value As Double

Value = FirstLeftInterval(1)

MsgBox "First Left Interval =" & Value

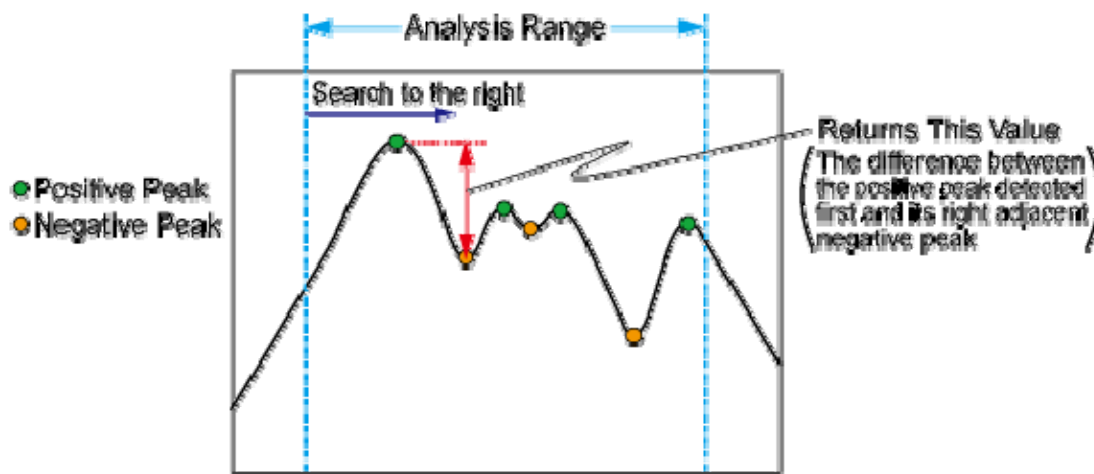
FirstRightGap(*Chan*)

Syntax

Value = FirstRightGap(*Chan*)

Description

Returns the response difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

FirstRightGap

e5071c434

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

E5071C

Return value

Parameter	<i>Value</i>
Description	Returns the response difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = FirstRightGap(1)
MsgBox "First Right Gap =" & Value
```

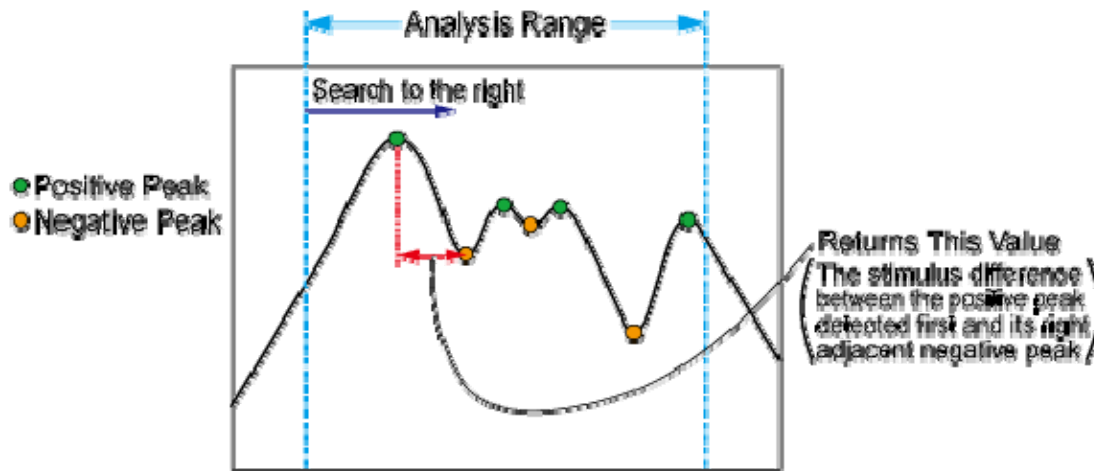
FirstRightInterval(*Chan*)

Syntax

Value = FirstRightInterval(*Chan*)

Description

Returns the stimulus difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

FirstRightInterval

e6071c433

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	<i>Value</i>
Description	Returns the stimulus difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

Dim Value As Double

Value = FirstRightInterval(1)

MsgBox "First Right Interval =" & Value

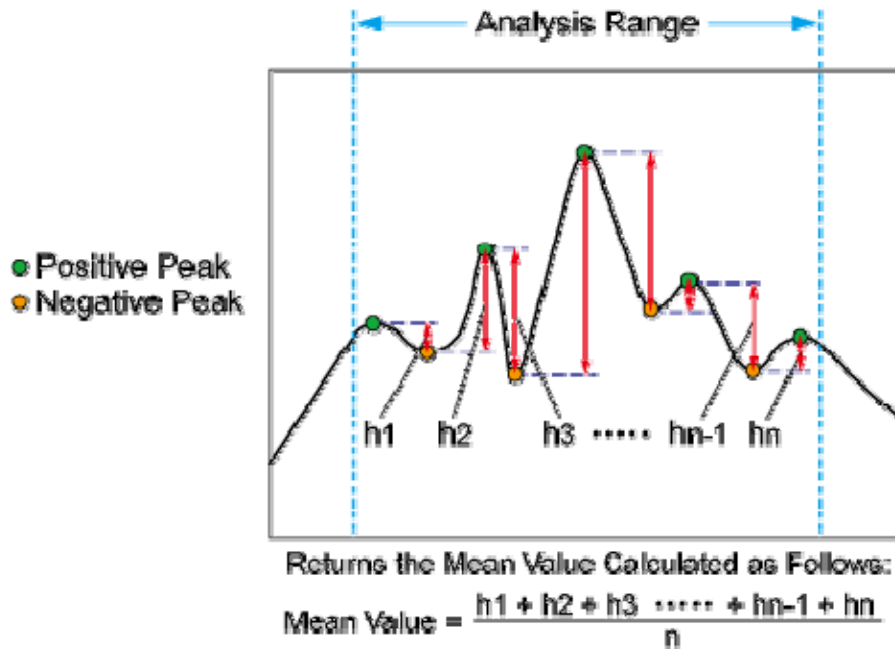
GapMean(*Chan*)

Syntax

Value = GapMean(*Chan*)

Description

Returns the mean value of the response differences between the negative peaks and its adjacent positive peaks within the analysis range.

GapMean

e5071c427

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.
-------------	-----------------------------------------------------------------------------------------------

Return value

Parameter	<i>Value</i>
Description	Returns the mean value of the response differences between the negative peaks and its right and left adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = GapMean(1)
MsgBox "Gap Mean =" & Value
```

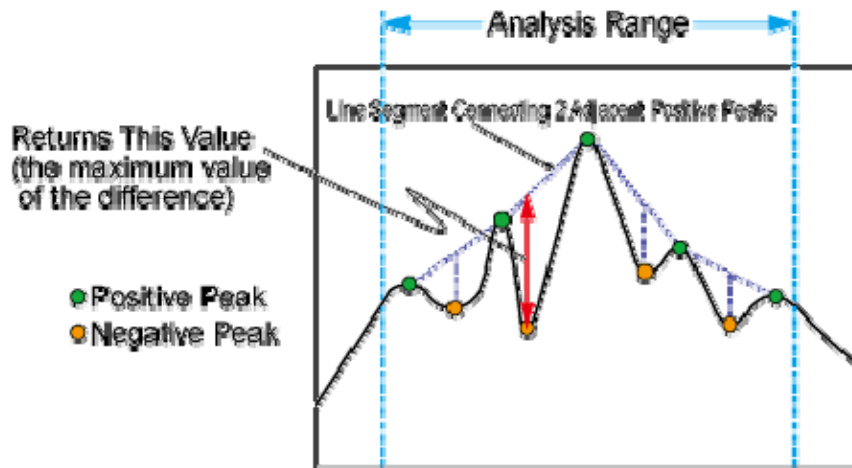

MaxEnvelopeGap(*Chan*)

Syntax

$$Value = \text{MaxEnvelopeGap}(\textit{Chan})$$

Description

Returns the maximum value of the vertical distance between the line segments connecting 2 adjacent positive peaks and the negative peaks between them within the analysis range.

MaxEnvelopeGap

e50719426

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the vertical distance between the line segments connecting 2 adjacent positive peaks and the negative peaks between them.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double  
Value = MaxEnvelopeGap(1)  
MsgBox "Max Envelope Gap =" & Value
```

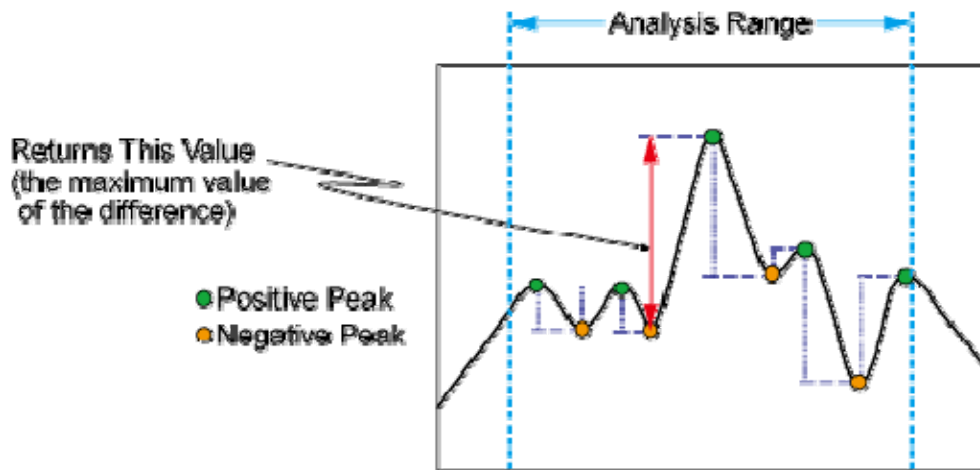
MaxGap(*Chan*)

Syntax

Value = MaxGap(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its adjacent negative peaks within the analysis range.

MaxGap

e5071c425

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the response differences between the positive peaks and its adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double  
Value = MaxGap(1)  
MsgBox "Max Gap =" & Value
```

MaxLeftGap(*Chan*)

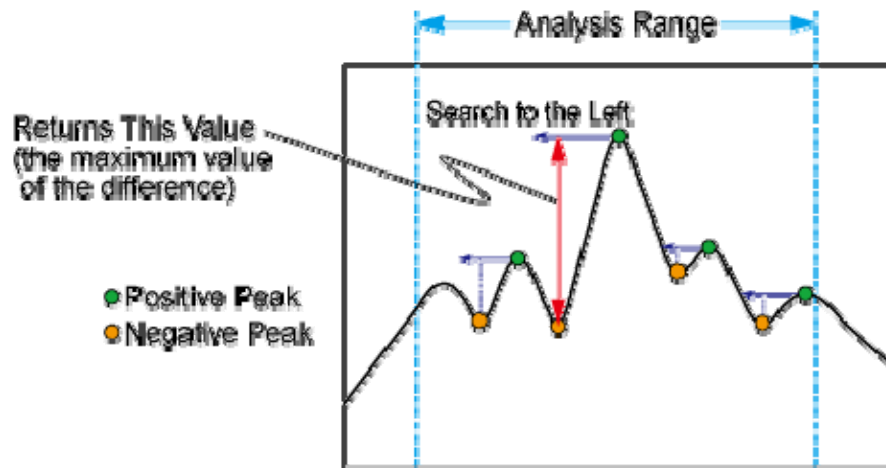
Syntax

Value = MaxLeftGap(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its left adjacent negative peaks within the analysis range.

MaxLeftGap



g5071g424

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the response differences between the positive peaks and its left adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double  
Value = MaxLeftGap(1)  
MsgBox "Max Left Gap =" & Value
```

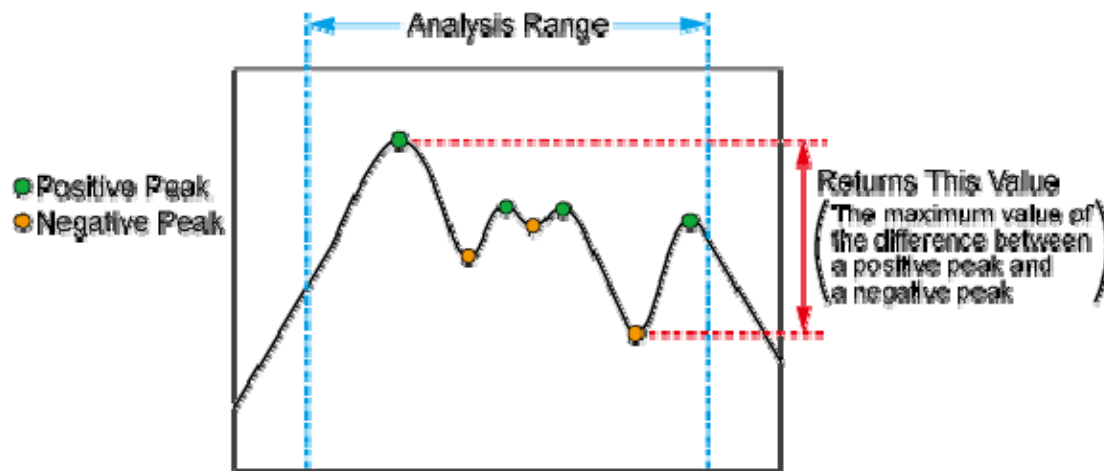
MaxPeakToPeak(*Chan*)

Syntax

Value = MaxPeakToPeak(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range.

MaxPeakToPeak

e5071c422

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the response differences between the positive peaks and the negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

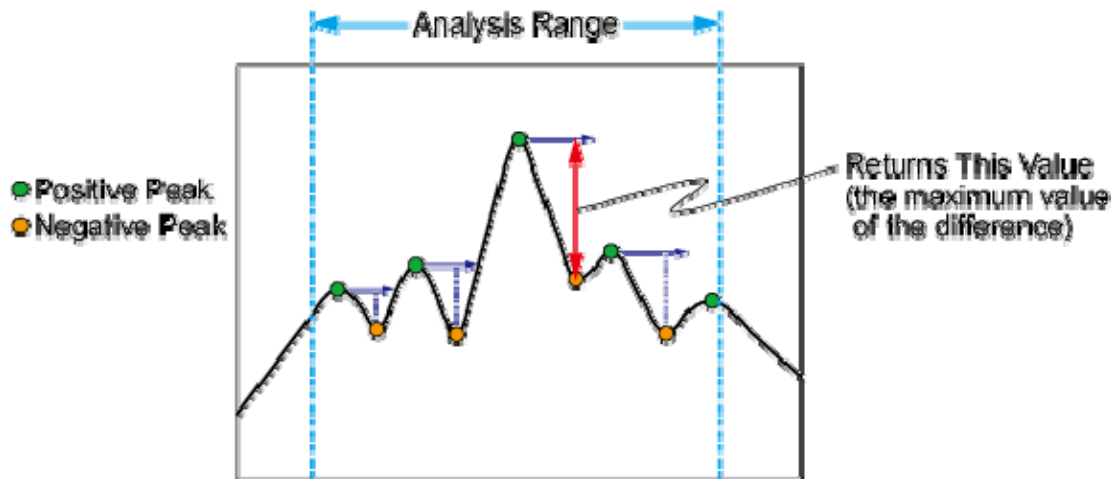
```
Dim Value As Double  
Value = MaxPeakToPeak(1)  
MsgBox "Max Peak To Peak =" & Value
```


MaxRightGap(*Chan*)**Syntax**

Value = MaxRightGap(*chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its right adjacent negative peaks within the analysis range.

MaxRightGap

e5071c423

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the response differences between the positive peaks and its right adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

Dim Value As Double

Value = MaxRightGap(1)

MsgBox "Max Right Gap =" & Value

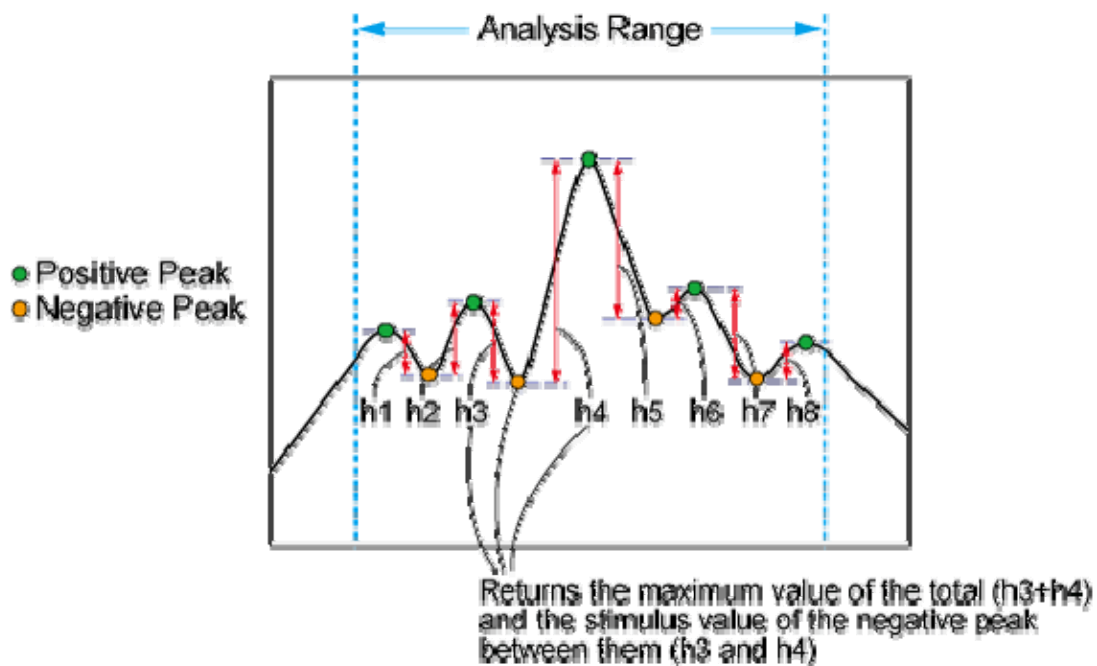
MaxRipplePoint(*Chan*,*Stim*)

Syntax

Value = MaxRipplePoint(*Chan*,*Stim*)

Description

Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks and the stimulus value of the applicable negative peaks within the analysis range.

MaxRipplePoint

e5071c428

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.
-------------	-----------------------------------------------------------------------------------------------

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	<i>Stim</i>
Description	Returns the stimulus value of the negative peak at which the sum of the response differences between the negative peak and its adjacent positive peaks is maximum.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Dim Stim As Double
```

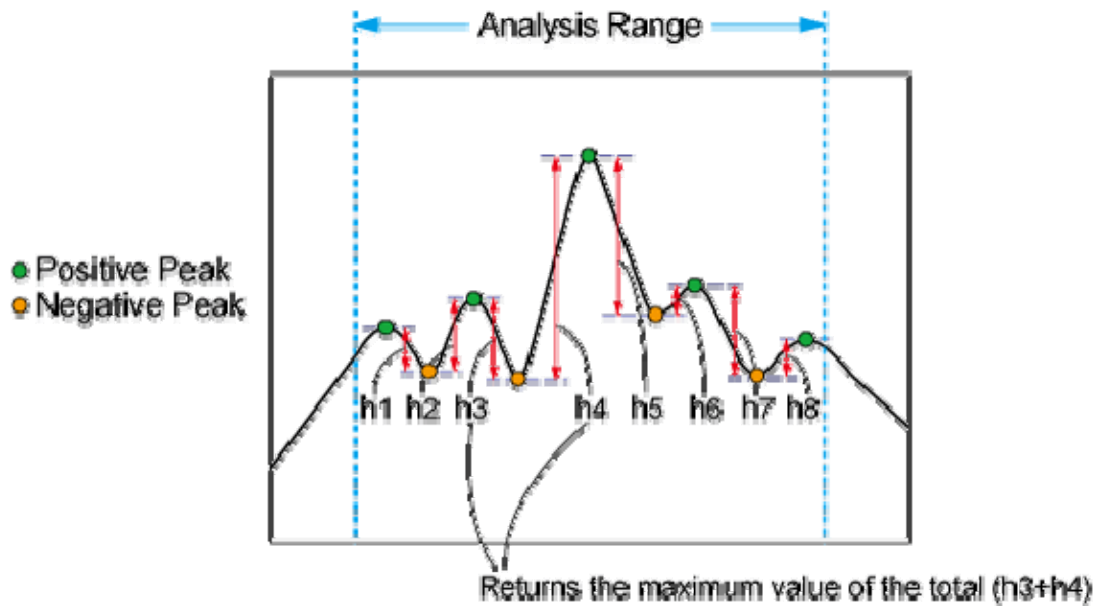
```
Value = MaxRipplePoint(1, Stim)
MsgBox "Max Ripple Value =" & Value & " , Stimulus =" & Stim
```

MaxRippleValue(*Chan*)**Syntax**

Value = MaxRippleValue(*Chan*)

Description

Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks within the analysis range.

MaxRippleValue

e5071c429

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.
-------------	-----------------------------------------------------------------------------------------------

Return value

Parameter	<i>Value</i>
Description	Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxRippleValue(1)
MsgBox "Max Ripple Value =" & Value
```

Pole(*Chan,D,LeftStim,LeftValue,RightStim,RightValue*)

Syntax

Call Pole(*Chan,D,LeftStim,LeftValue,RightStim,RightValue*)

Description

For the negative peaks below the specified value (*D*) relative to the maximum value of the positive peaks within the analysis range, returns the response value (*LeftValue*) and stimulus value (*LeftStimulus*) of the negative peak first detected when searched to the left from the maximum value of the positive peaks, and the response value (*RightValue*) and stimulus value (*RightStimulus*) of the negative peak first detected when searched to the right from the maximum value of the positive peaks.

Pole



e5971c430

Variable

Parameter	<i>Chan</i>
Description	Specifies the channel number.

Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>D</i>
Description	Specifies the difference from the maximum value.
Data type	Double precision floating point type (Double)

Return value (arguments)

Parameter	<i>LeftStim</i>
Description	Returns the stimulus value of the negative peak first detected to the left from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	<i>LeftValue</i>
Description	Returns the response value of the negative peak first detected to the left from the maximum value of the positive peaks.

Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	<i>RightStim</i>
Description	Returns the stimulus value of the negative peak first detected to the right from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	<i>RightValue</i>
Description	Returns the response value of the negative peak first detected to the right from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim LeftStim As Double
Dim LeftValue As Double
Dim RightStim As Double
Dim RightValue As Double
```

```
Call Pole(1, 1, LeftStim, LeftValue, RightStim, RightValue)
```

E5071C

MsgBox "Left Pole =" & LeftStim & ":" & LeftValue

MsgBox "Right Pole =" & RightStim & ":" & RightValue

TDR Remote Control

Option TDR Remote Control

- Concepts of Remote Control in TDR
- Setup and Connection of TDR
- TDR Sample Program

Other topics about Programming

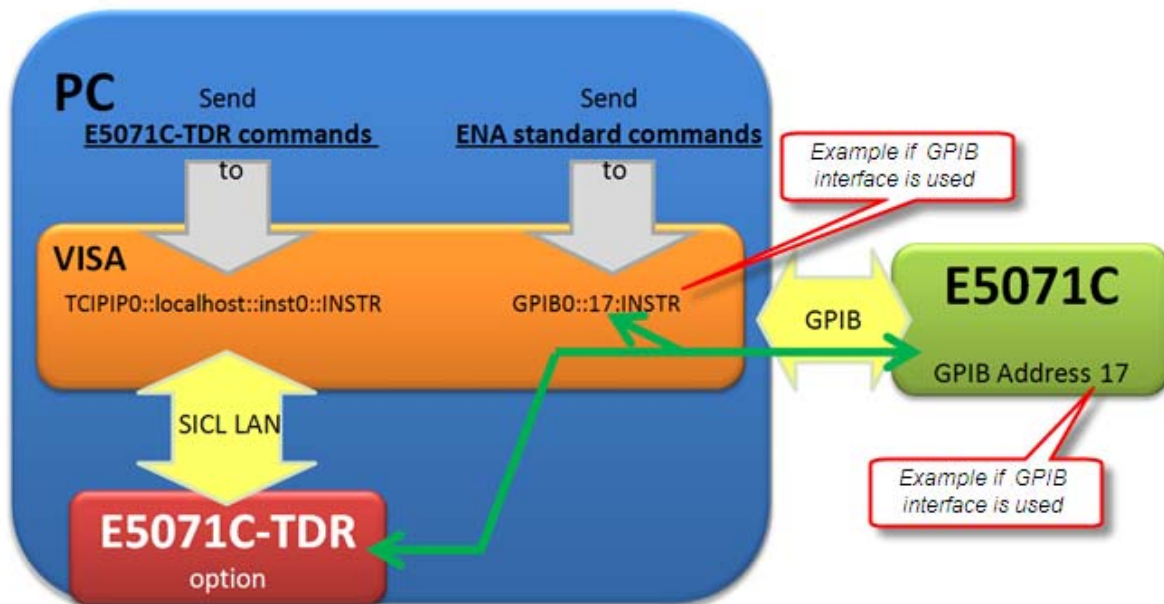
Concept of Remote Control in TDR

Enhanced Time Domain Analysis (E5071C-TDR) can be controlled remotely.

The figure below shows the connection between a PC, E5071C and Enhanced Time Domain Analysis option. VISA application treats both E5071C and Enhanced Time Domain Analysis option as instruments.

As such, VISA application connects to E5071C via GPIB through GPIB0::17:INSTR address. Standard E5071C commands are sent through GPIB to the E5071C. USB and LAN are alternatives to GPIB.

Whereas, VISA application connects to Enhanced Time Domain Analysis option via SICL LAN through TCPIP0::localhost::inst0::INSTR address. E5071C-TDR commands are sent through SICL LAN to the Enhanced Time Domain Analysis option.



To learn more about other remote control system, refer to Types of remote control system.

SCPI commands and COM commands are used to control E5071C remotely. However, only SCPI commands can be used to control the Enhanced Time Domain Analysis option. Therefore, it is not possible to control it by built-in VBA.

Other topics about TDR Remote Control

Setup and Connection of TDR

- [Installing E5071C-TDR application to PC](#)
- [Connecting E5071C-TDR with PC](#)

Other topics about TDR Remote Control

Installing E5071C-TDR application to PC

Three applications need to be installed in order for E5071C-TDR to be worked correctly:

Installing .Net Framework Version 2.0 and above

1. Ensure that .Net Framework Version 2.0 and above is installed in the PC. Go to Windows Control Panel > Add or Remove Programs to confirm if it is installed. If it is not installed, go to below URL and download the latest version.

<http://www.microsoft.com/downloads/>

Installing Agilent IO Library

1. Ensure that Agilent IO Library is installed in the PC. If it is not installed, go to below URL and download the latest Agilent IO Libraries Suite executable file:

[Download IO library](#)

2. Double-click on the executable file for installation. Follow the instruction of the installation wizard.

Installing E5071C-TDR Software

1. Download the latest TDR software from the http://www.agilent.com/find/ena-tdr_firmware.

NOTE

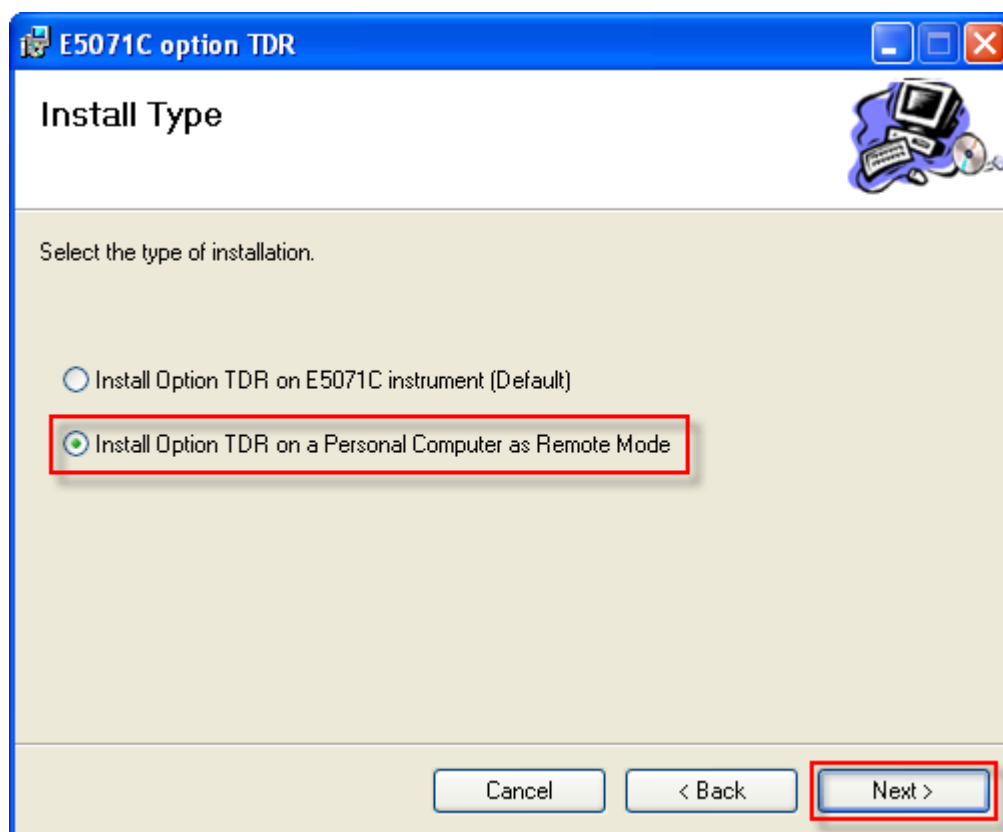
Install the same revision of the E5071C-TDR application into the instrument and the PC. This ensure the remote access to the instrument works correctly

2. Download the execution file (**E5071C_opt_TDR_xx.xx.exe**).
3. Double-click on the executable file, **E5071C_opt_TDR.msi**.
4. Click **Next >**.



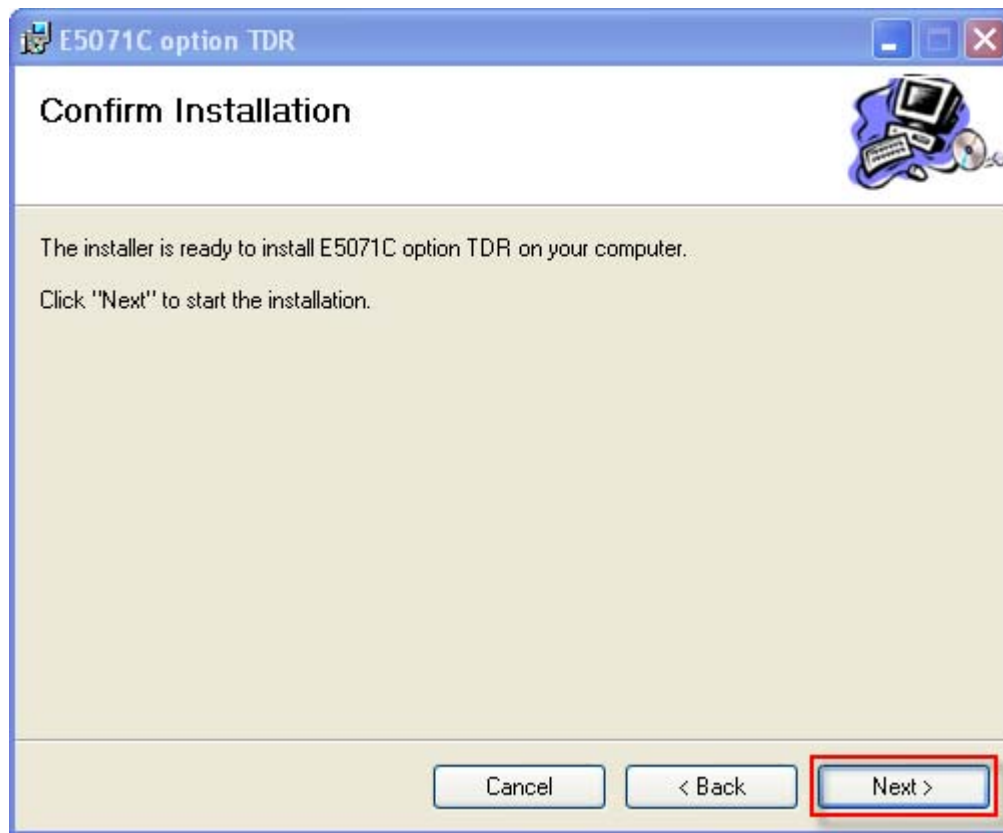
e5071c521

3. Select Install Option TDR on a Personal Computer as Remote Mode and click **Next >**.



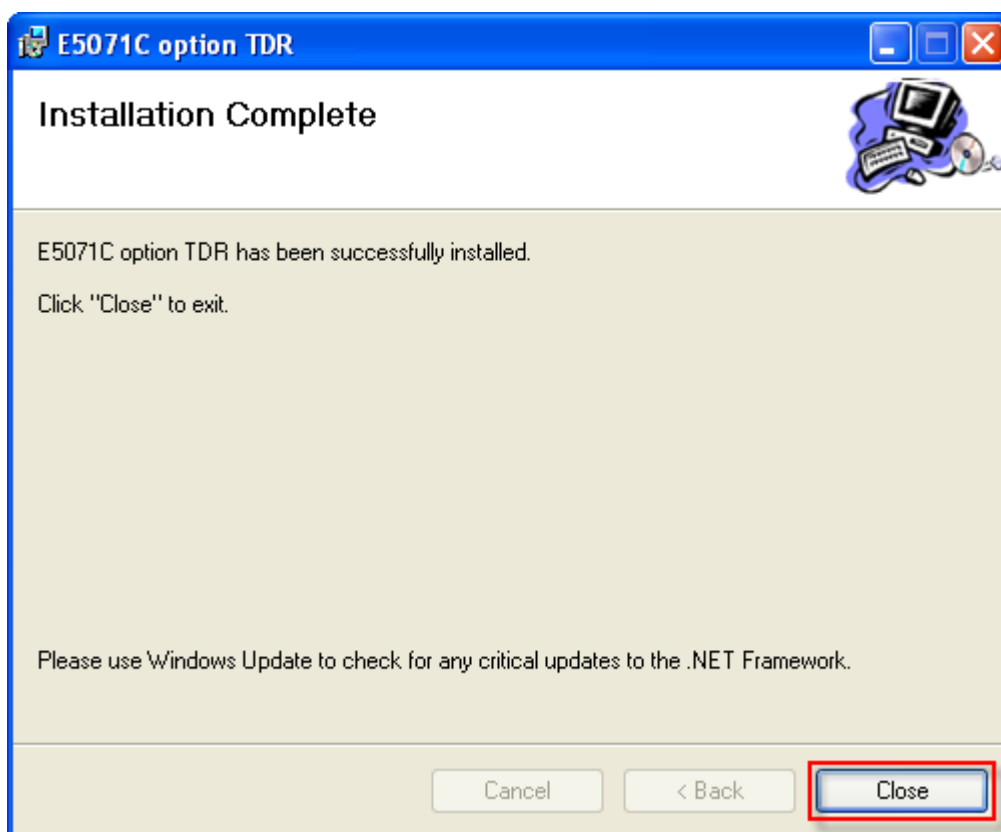
e5071c522

4. Click **Next >**.



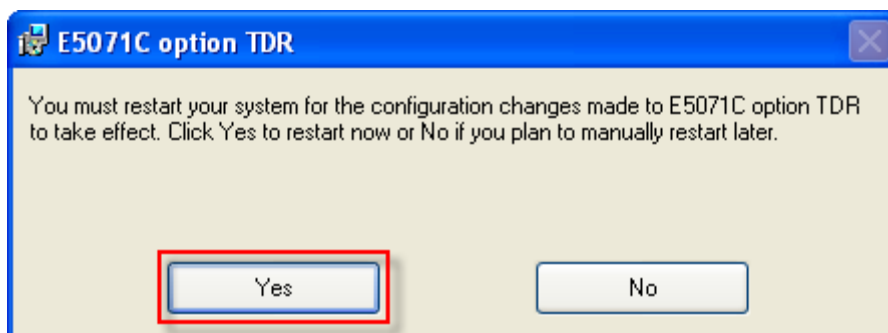
e5071c523

5. Click **Close** to complete the installation.



e5071c524

6. E5071C option TDR window appears. Press **Yes** when ready to reboot.



e5071c525

Connecting E5071C-TDR with PC

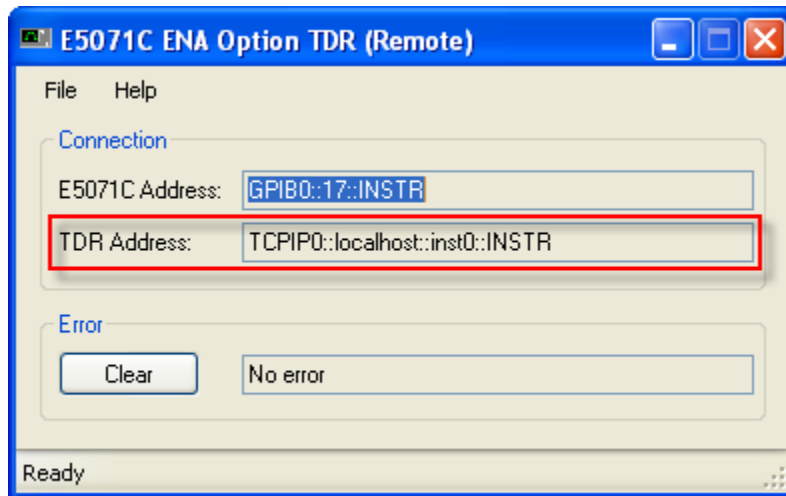
Below steps are needed to access the instrument remotely.

1. Connect the E5071C with the PC through either GPIB, LAN or USB.
2. From Windows Start on your PC, **Start > All Programs > Agilent E5071C ENA Option TDR > TDR Remote**.
3. Type the VISA address of E5071C and click **OK**. (This dialog box is displayed once. The next time TDR Remote is launched, the previous VISA address is used to connect to the instrument.)



e5071c526

4. E5071C ENA Option TDR (Remote) window appears. Control the E5071C TDR application with the address in TDR address.



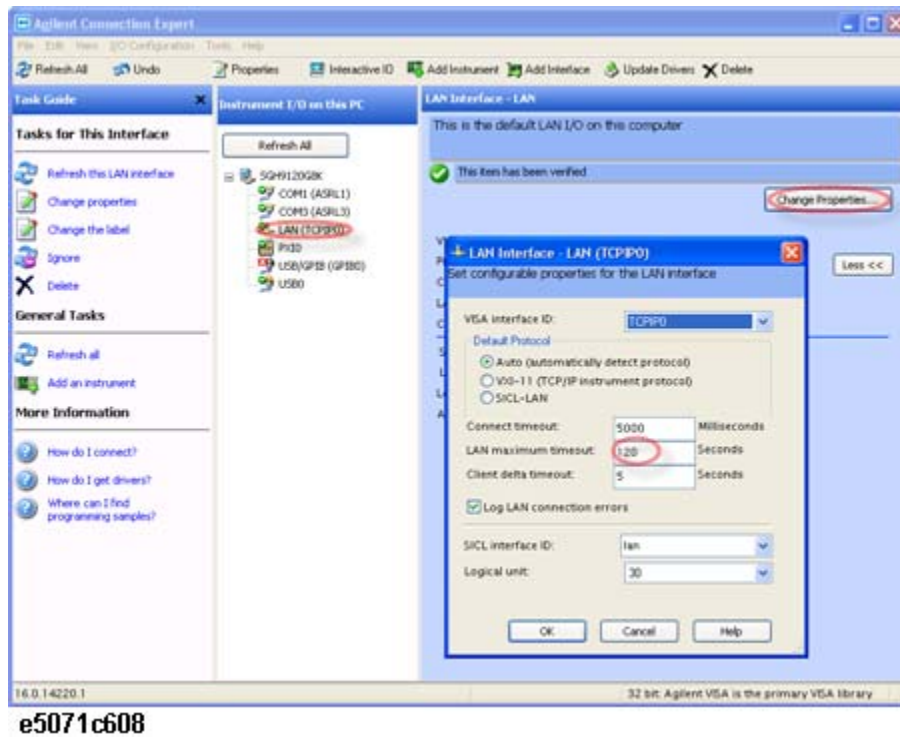
e5071c527

5. E5071C-TDR GUI is closed in the instrument when TDR Remote is launched.

NOTE GUI mode can be used in exchange to the Remote mode by clicking **Analysis** > **Start TDR** (Firmware is restarted). ENA Option TDR can be controlled by using either one of two modes, GUI or Remote mode.

Note for connecting with PC through LAN

- Turn on SICL-LAN Server.
- If you specify the timeout time for the E5071C, you need to consider the time out setting for LAN. On Agilent connection expert, you can specify the timeout for LAN and its default is 120 sec. If you specify the timeout longer than 120 sec. The LAN minimum timeout setting for LAN interface should be extended.

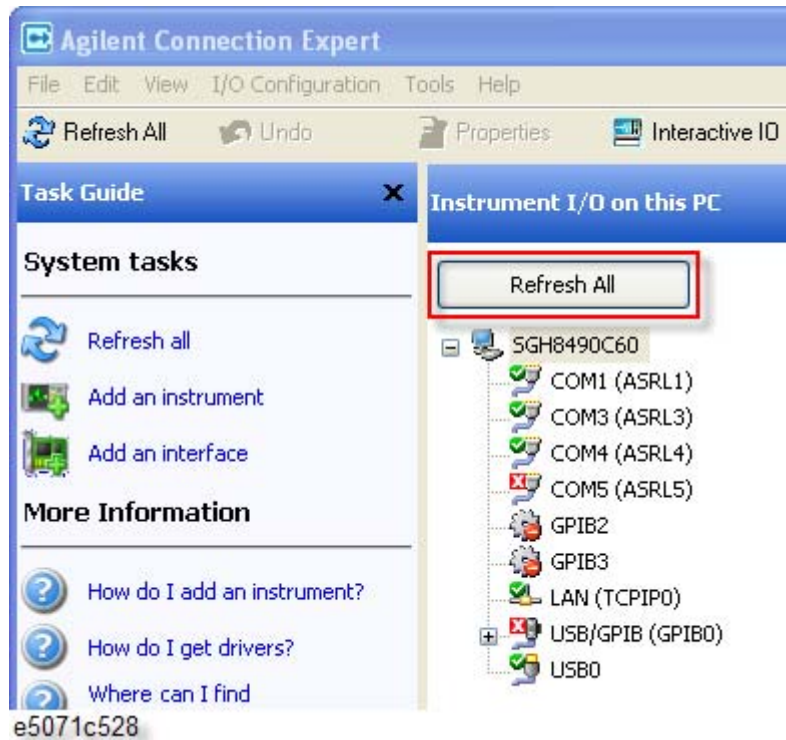


e5071c608

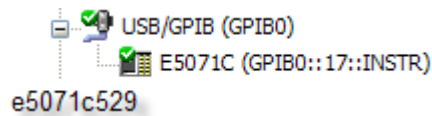
Configuring IO Library Setup

This is optional as it needs Agilent Connection Expert.

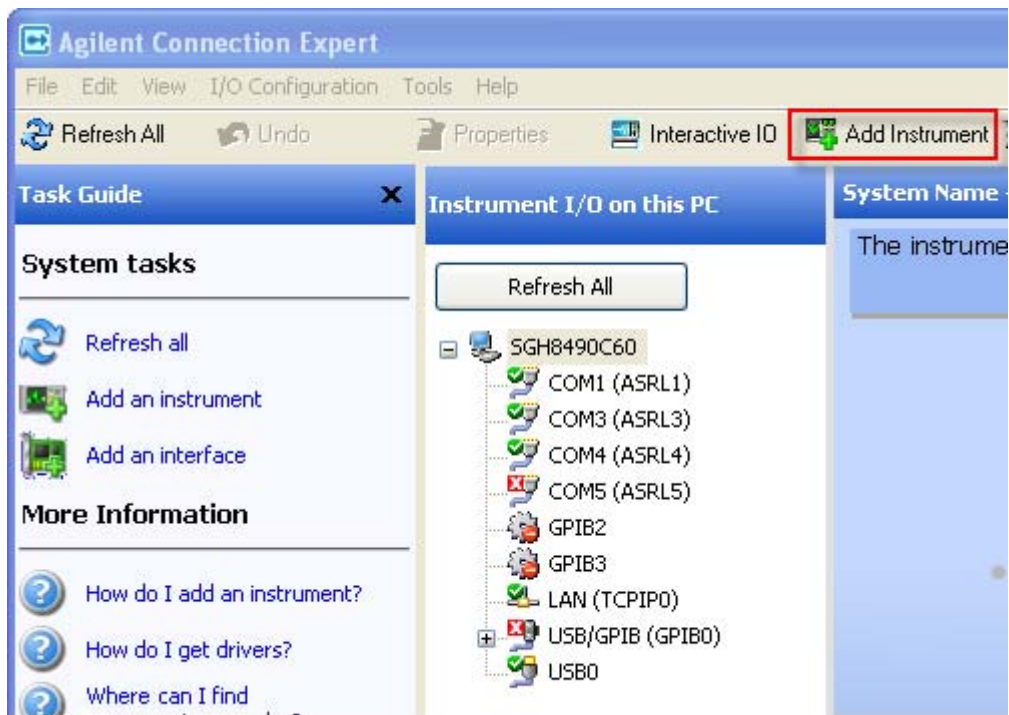
1. From Windows Start on your PC, click **Start > All Programs > Agilent IO Libraries Suite > Agilent Connection Expert**.
2. Click **Refresh All**.



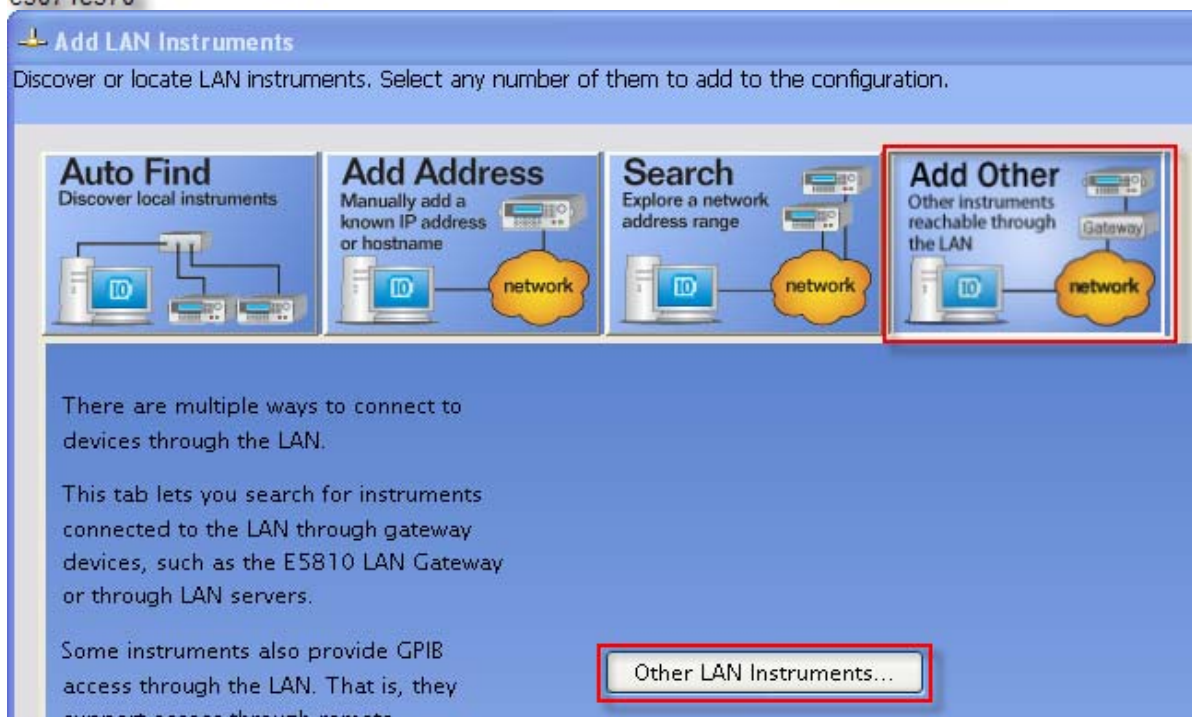
3. Refresh the system. The E5071C is now available under GPIB/USB category.



4. Click on LAN (TCPIP0).
5. Select **Add Instrument > Add Other > Other LAN Instruments...** to connect to E5071C-TDR.

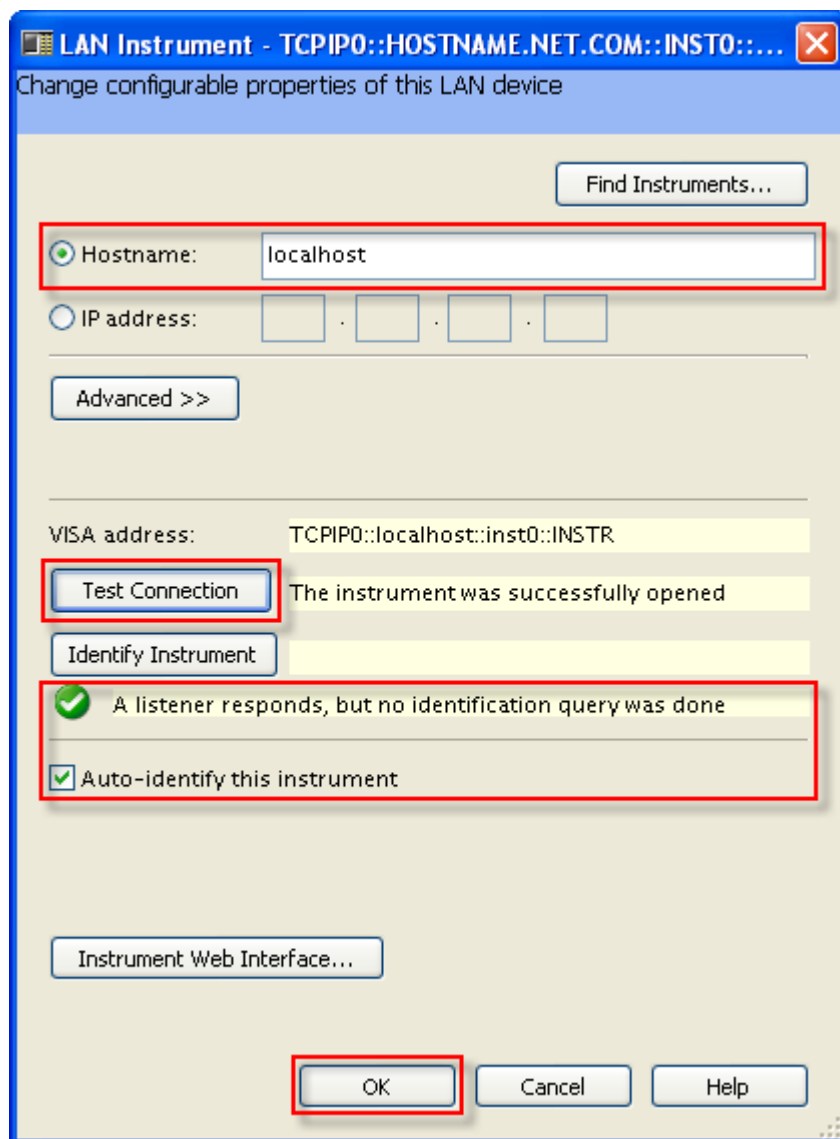


e5071c570



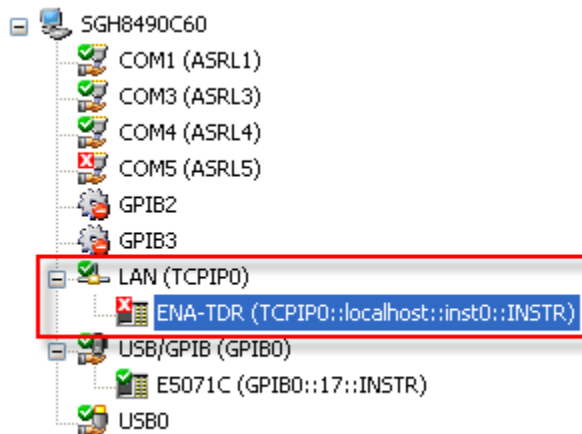
e5071c530

5. Type localhost in Hostname.
6. Click **Test Connection** to confirm that the connection is working.
7. Select out Auto-identify this instrument and click **OK**.



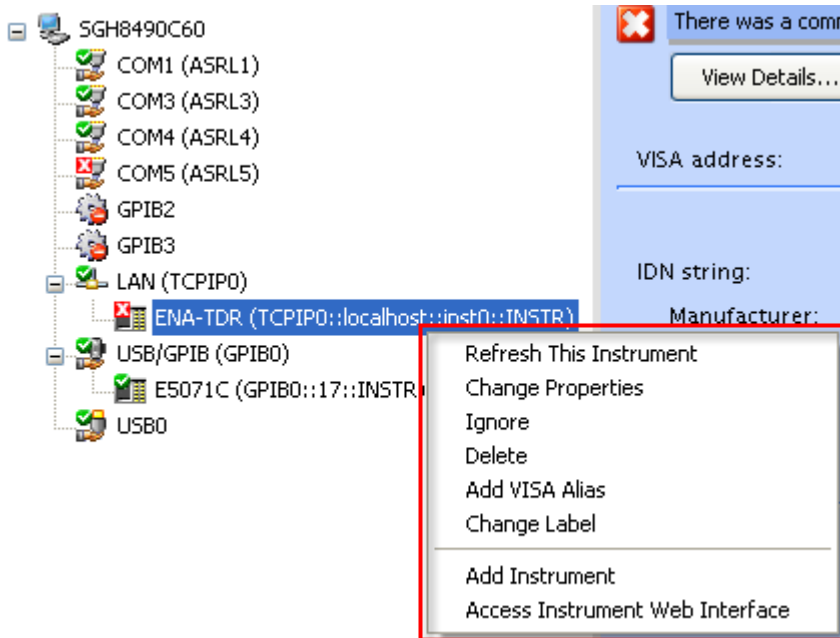
e5071c531

8. Notice that ENA-TDR connection fails in Agilent Connection Expert main window.



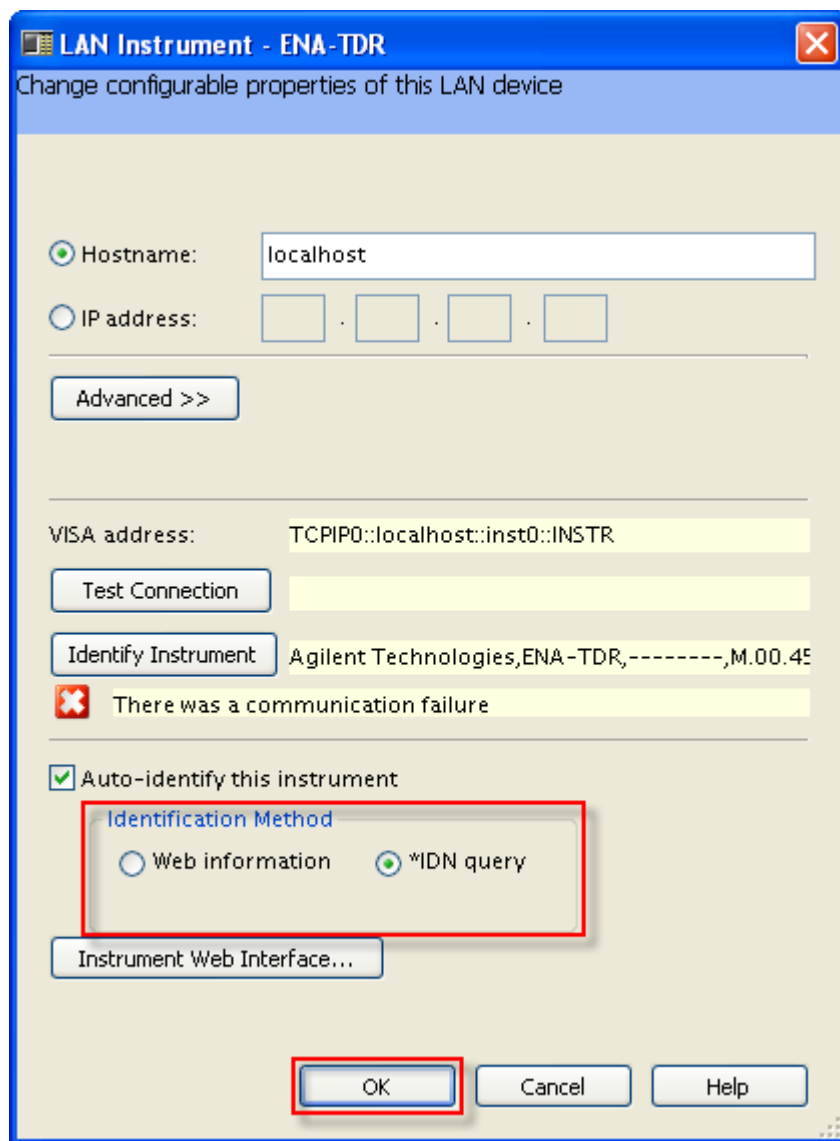
e5071c532

9. Right-click on ENA-TDR option and select **Change Properties**.



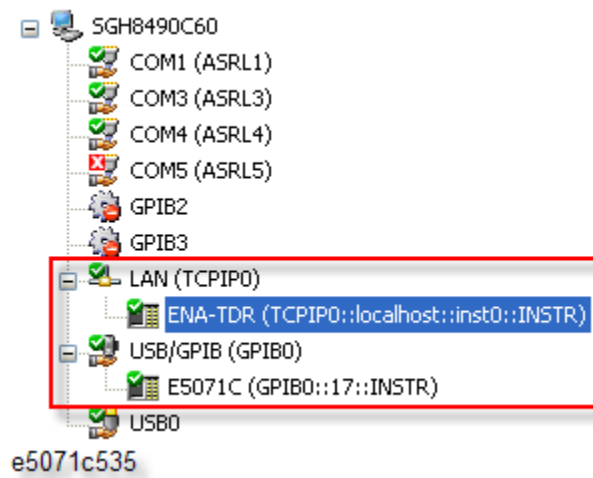
e5071c533

10. Select ***IDN query** under **Identification Method** at LAN Instrument window and click **OK**.



e5071c534

11. Both E5071C and ENA-TDR should be active in Agilent Connection Expert main window as shown below.



TDR Sample Programs

TDR Sample Programs

- TDR/TDT Measurement
- Simulated Eye Diagram
- 2 Channels Measurement

TDR/TDT Measurement

Overview

This sample program demonstrates how to perform TDR/TDT setup.

[Sample Program in Excel VBA \(VISA-COM\)](#)

Global variables

```
Dim rm As VisaComLib.ResourceManager
Dim ENATDR As VisaComLib.FormattedIO488
Dim NumDmy As Integer
```

TDR-TDT Measurement

```
Sub TDRTDTMeasure()
    On Error GoTo errorhandler
    Set rm = New VisaComLib.ResourceManager
    Set ENATDR = New VisaComLib.FormattedIO488
    ' TDR address is not necessary to change.
    Set ENATDR.IO = rm.Open("TCPIP0::localhost::inst0::INSTR")
    ENATDR.IO.Timeout = 50000
    ' Clear Excel Sheet Cells
    Range("F5:F5").ClearContents
    With ENATDR
        'Preset the ENA
        .WriteString ":SYST:PRES"
        'TDR Trigger Hold
        .WriteString ":TRIG:MODE HOLD"
        ' Set DUT Topology to Differential 2-Port
        .WriteString ":CALC:DEV DIF2"
        .WriteString "**OPC?"
        NumDmy = .ReadNumber
        ' Execute Deskew (Deskew & Loss)
        MsgBox "[Deskew] Disconnect the cables from DUT to make open condition.", vbOKOnly
        .WriteString ":SENS:CORR:EXT:AUTO:STAN OPEN"
        .WriteString ":SENS:CORR:EXT:AUTO:PORT1 ON"
        .WriteString ":SENS:CORR:EXT:AUTO:PORT2 ON"
        .WriteString ":SENS:CORR:EXT:AUTO:PORT3 ON"
        .WriteString ":SENS:CORR:EXT:AUTO:PORT4 ON"
        .WriteString ":SENS:CORR:EXT:AUTO:IMM"
        .WriteString "**OPC?"
    End With
end Sub
```

```

NumDmy = .ReadNumber
' Execute Thru Measurement: Connect a Thru between Port 1 and Port 3
MsgBox "Connect a Thru between Port 1 and Port 3.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:THRU TH13"
.WriteString "**OPC?"
NumDmy = .ReadNumber
' Execute Thru Measurement: Connect a Thru between Port 2 and Port 4
MsgBox "Connect a Thru between Port 2 and Port 4.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:THRU TH24"
.WriteString "**OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 1
MsgBox "Connect a Load to Port 1.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 1"
.WriteString "**OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 2
MsgBox "Connect a Load to Port 2.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 2"
.WriteString "**OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 3
MsgBox "Connect a Load to Port 3.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 3"
.WriteString "**OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 4
MsgBox "Connect a Load to Port 4.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 4"
' Save the result of loss compensation
.WriteString ":SENS:CORR:COLL:DLC:SAVE"
.WriteString "**OPC?"
NumDmy = .ReadNumber

'Measure DUT Length
MsgBox "Connect the DUT to the cables.", vbOKOnly
.WriteString ":SENS:DLEN:AUTO:IMM"

```

```
.WriteString "**OPC?"
NumDmy = .ReadNumber
'Set Rise Time
.WriteString ":CALC:TRAC1:TIME:STEP:RTIM:DATA 35E-12"
.WriteString ":CALC:TRAC3:TIME:STEP:RTIM:DATA 35E-12"
.WriteString ":CALC:TRAC5:TIME:STEP:RTIM:DATA 35E-12"
.WriteString ":CALC:TRAC7:TIME:STEP:RTIM:DATA 35E-12"
'Set Definition
.WriteString ":CALC:TRAC1:TIME:STEP:RTIM:THR T1_9"
.WriteString ":CALC:TRAC3:TIME:STEP:RTIM:THR T1_9"
.WriteString ":CALC:TRAC5:TIME:STEP:RTIM:THR T1_9"
.WriteString ":CALC:TRAC7:TIME:STEP:RTIM:THR T1_9"
.WriteString ":TRIG:SING"
.WriteString "**OPC?"
NumDmy = .ReadNumber
```

```
'Set Auto Scale for all traces
.WriteString ":DISP:ATR:SCAL:AUTO"
.WriteString "**OPC?"
NumDmy = .ReadNumber
```

```
'Read Rise Time
.WriteString ":CALC:TRAC3:TTIM:STAT ON"
.WriteString ":CALC:TRAC3:TTIM:THR T1_9"
.WriteString ":CALC:TRAC3:TTIM:DATA?"
Cells(5, 6) = .ReadNumber
```

```
MsgBox "TDT/TDT Measurement Setup is completed.", vbOKOnly
End With
ENATDR.IO.Close
Exit Sub
```

errorhandler:

```
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
```

End Sub

Simulated Eye Diagram

Overview

This sample program demonstrates how to perform simulated eye diagram.

[Sample Program in Excel VBA \(VISA-COM\)](#)

Global variables

```
Dim rm As VisaComLib.ResourceManager
Dim ENATDR As VisaComLib.FormattedIO488
Dim NumDmy As Integer
```

Open

```
Sub OpenInstrument()
    On Error GoTo errorhandler
    Set rm = New VisaComLib.ResourceManager
    Set ENATDR = New VisaComLib.FormattedIO488

    ' TDR address is not necessary to change.
    Set ENATDR.IO = rm.Open("TCPIP0::localhost::inst0::INSTR")
    ENATDR.IO.Timeout = 50000

    ' Clear Excel Sheet Cells
    Range("F5:F22").ClearContents
    MsgBox "E5071C-TDR is opened.", vbOKOnly

    Exit Sub
errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Setup

```
Sub setup()
    On Error GoTo errorhandler
    With ENATDR
        'Preset the ENA
        .WriteString ":SYST:PRES"
        .WriteString "**OPC?"
```

```
NumDmy = .ReadNumber
```

```
'TDR Trigger Hold
```

```
.WriteString ":TRIG:MODE HOLD"
```

```
' Set DUT Topology to Differential 2-Port
```

```
.WriteString ":CALC:DEV DIF2"
```

```
.WriteString "**OPC?"
```

```
NumDmy = .ReadNumber
```

```
' Execute Deskew (Deskew)
```

```
MsgBox "[Deskew] Disconnect the cables from DUT to make open condition.", vbOKOnly
```

```
.WriteString ":SENS:CORR:EXT:AUTO:STAN OPEN"
```

```
.WriteString ":SENS:CORR:EXT:AUTO:PORT1 ON"
```

```
.WriteString ":SENS:CORR:EXT:AUTO:PORT2 ON"
```

```
.WriteString ":SENS:CORR:EXT:AUTO:PORT3 ON"
```

```
.WriteString ":SENS:CORR:EXT:AUTO:PORT4 ON"
```

```
.WriteString ":SENS:CORR:EXT:AUTO:IMM"
```

```
.WriteString "**OPC?"
```

```
NumDmy = .ReadNumber
```

```
'Measure DUT Length
```

```
MsgBox "Connect the DUT to the cables.", vbOKOnly
```

```
.WriteString ":SENS:DLEN:AUTO:IMM"
```

```
.WriteString "**OPC?"
```

```
NumDmy = .ReadNumber
```

```
'Set Rise Time
```

```
.WriteString ":CALC:TRAC1:TIME:STEP:RTIM:DATA 35E-12"
```

```
'Set Definition
```

```
.WriteString ":CALC:TRAC1:TIME:STEP:RTIM:THR T1_9"
```

```
'Activate Trace 3
```

```
.WriteString ":CALC:ATR:ACT 3"
```

```
.WriteString ":TRIG:SING"
```

```
.WriteString "**OPC?"
```

```
NumDmy = .ReadNumber
```

```
MsgBox "Setup is completed.", vbOKOnly
```

```
End With
```

```
Exit Sub
```

errorhandler:

```
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Simulated Eye Diagram

```
Sub SimEyeDiagram()
Dim EyeResult() As Double
Dim NoR As Integer, i As Integer
On Error GoTo errorHandler
With ENATDR
    ' Set Bit Pattern Parameters
    ' Set Type to PRBS
    .WriteString ":CALC:EYE:INP:BPAT:TYPE PRBS"
    ' Set Length
    .WriteString ":CALC:EYE:INP:BPAT:LENG 7"
    ' Set Amplitude
    .WriteString ":CALC:EYE:INP:OLEV 200e-3"
    ' Set Data Rate
    .WriteString ":CALC:EYE:INP:RTIM:THR T1_9"
    .WriteString "**OPC?"
    NumDmy = .ReadNumber

    ' Execute Draw Eye
    .WriteString ":CALC:EYE:STAT ON"
    .WriteString ":CALC:EYE:EXEC"
    .WriteString "**OPC?"
    NumDmy = .ReadNumber

    ' Read Eye Result
    .WriteString ":CALC:EYE:RES:DATA?"
    EyeResult() = .ReadList(ASCIIType_R8, ",")

    NoR = 18
    i = 0
    Do While i < NoR
        Cells(i + 5, 6) = EyeResult(i)
```



```

        i = i + 1
    Loop
    .WriteString "**OPC?"
    NumDmy = .ReadNumber
    MsgBox "Eye Diagram Simulation is completed.", vbOKOnly
End With
Exit Sub

```

errorhandler:

```

    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub

```

Close

```

Sub CloseInstrument()
    ENATDR.IO.Close
End Sub

```

2 Channels Measurement

- Overview
- Sample Program in Excel VBA (VISA-COM)

Other topics about TDR Sample Program

Overview

2 channel measurement allows you to makes TDR measurement on channel 1 and more customized S-parameter measurement on channel2. This is a program example of the 2 Channel Measurement Example. In 2 channel measurement, you need to setup two logical instrument, E5071C and E5071C-TDR. In this sample program, the E5071C is assigned as ENA and E5071C-TDR is assigned as ENATDR.

The TDR commands should be sent to ENATDR and The E5071C standard commands should be sent to ENA.

Note for 2 Channels Measurement

- Put *OPC? to make sure to finish the process before you change the control from ENA to ENATDR or ENATDR to ENA.
- DUT topology setting should be executed at the beginning of sequence, because changing DUT Topology presets the E5071C.

Sample Program in Excel VBA (VISA-COM)

Grobal variables

```
Dim rm As VisaComLib.ResourceManager
Dim ENA As VisaComLib.FormattedIO488
Dim ENATDR As VisaComLib.FormattedIO488
Dim NumDmy As Integer
```

Open Instruments

```
Sub OpenInstrument()
    On Error GoTo errorhandler
    Set rm = New VisaComLib.ResourceManager
    Set ENA = New VisaComLib.FormattedIO488
    Set ENATDR = New VisaComLib.FormattedIO488
```

' Change the VISA address for your configuration. For example, if using USB interface, an example would be `rm.Open("USB0::0x0957::0x0D09::{serial number}::0::INSTR")`, if using LAN interface, an example would be `rm.Open("TCPIP0::{ip address}::inst0::INSTR")`. If using LAN interface, SICL-LAN Server in the E5071C should be turned on.

```
Set ENA.IO = rm.Open("GPIB0::17::INSTR")
```

```
' Set time out time for ENA at 30 second
```

```
ENA.IO.Timeout = 30000
```

```
' TDR address is not necessary to change.
```

```
Set ENATDR.IO = rm.Open("TCPIP0::localhost::inst0::INSTR")
```

```
' Set time out time for ENATDR at 70 second
```

```
ENATDR.IO.Timeout = 70000
```

```
' Clear Excel Sheet Cells
```

```
Range("D7:H10010").ClearContents
```

```
Range("E4:E4").ClearContents
```

```
Range("H4:H4").ClearContents
```

```
MsgBox "E5071C and E5071C-TDR are opened.", vbOKOnly
```

```
Exit Sub
```

```
errorhandler:
```

```
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
```

```
End Sub
```

Preparation for 2 Channels

```
Sub Preparation()
```

```
On Error GoTo errorhandler
```

```
' Note: Must put *OPC? when you change the control between ENA and ENATDR.
```

```
With ENA
```

```
    ' ENA- Preset ENA
```

```
    .WriteString "**RST"
```

```
    ' *OPC? for checking the process finish
```

```
    .WriteString "**OPC?"
```

```
    NumDmy = .ReadNumber
```

```
End With
```

```
With ENATDR
```

```
    ' TDR- Set Diff 2 port as DUT
```

```
    .WriteString ":CALC:DEV DIF2"
```

```
    ' TDR- Wait Until Topology Setting is finished.
```

```
    .WriteString "**OPC?"
```

```
    NumDmy = .ReadNumber
```

E5071C

```
'
' TDR- Trigger Hold
.WriteString ":TRIG:MODE HOLD"
'
' *OPC? for checking the process finish
.WriteString "**OPC?"
NumDmy = .ReadNumber
End With

' ENA- Setup ENA
With ENA
' Allocate channel must be done after TDR DUT Type setting.
.WriteString ":DISP:SPL D12"
'
' ENA- Set Trigger at BUS
.WriteString ":TRIG:SOUR BUS"

' ENA- Set Trigger scope at Active channel only
.WriteString ":TRIG:SCOP ACT"
'
' ENA- Beep Warning OFF
.WriteString ":SYST:BEEP:WARN:STAT OFF"
'
' *OPC? for checking the process finish
.WriteString "**OPC?"
NumDmy = .ReadNumber
End With

MsgBox "Preparation for 2ch is done.", vbOKOnly
Exit Sub

errorhandler:
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext

End Sub
```

Setup for Channel 1

1552

Sub SetupCh1()

On Error GoTo errorhandler

With ENATDR

,

' TDR- Execute Deskew

FrmDeSkew.Show ' Call DeSkew subroutine in Form

,

' TDR- Execute Auto DUT Length

FrmDUTLength.Show ' Call DUT Length subroutine in From

,

' TDR- Set rise time for Traces 1 and 2

.WriteString ":CALC:TRAC1:TIME:STEP:RTIM:THR T2_8"

.WriteString ":CALC:TRAC1:TIME:STEP:RTIM:DATA 50e-12"

.WriteString ":CALC:TRAC2:TIME:STEP:RTIM:THR T2_8"

.WriteString ":CALC:TRAC2:TIME:STEP:RTIM:DATA 50e-12"

,

' *OPC? for checking the process finish

.WriteString "**OPC?"

NumDmy = .ReadNumber

End With

,

' Advance mode setting (such as Limit Test) for channel 1 (TDR) should be setup through ENA

,

With ENA

' ENA- Limit Test ON

.WriteString ":CALC1:LIM ON"

' ENA- Limit Line On

.WriteString ":CALC1:LIM:DISP ON"

' ENA- Edit Limit Line

.WriteString ":CALC1:LIM:DATA 2,1,0,1e-9,105,105,2,0,1e-9,75,75"

,

' *OPC? for checking the process finish

.WriteString "**OPC?"

NumDmy = .ReadNumber

End With

E5071C

```
MsgBox "Setup for Ch1 is done.", vbOKOnly
Exit Sub
```

errorhandler:

```
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Deskew (This is called from SetupCh1)

```
Sub DeSkew()
    With ENATDR
        ' TDR- Execute Descrow
        .WriteString ":SENS:CORR:EXT:AUTO:IMM"
        ' *OPC? for checking the process finish
        .WriteString "**OPC?"
        NumDmy = .ReadNumber
    End With
End Sub
```

DutLength (This is called from SetupCh1)

```
Sub DUTLength()
    With ENATDR
        ' TDR- Execute Auto DUT length
        .WriteString ":SENS:DLEN:AUTO:IMM"
        ' *OPC? for checking the process finish
        .WriteString "**OPC?"
        NumDmy = .ReadNumber
    End With
End Sub
```

Measurement for Channel 1

```
Sub MeasCh1()
    On Error GoTo errorHandler
    '
    Dim TimeData() As Double, Impedance() As Double
    Dim Nop As Integer, PassFail As Integer, i As Integer, k As Integer
    '
end Sub
```

' Making Measurement for TDR (Channel 1)

With ENA

```
' ENA- Set active channel at channel 1
.WriteString ":DISP:WIND1:ACT"
' *OPC? for checking the process finish
.WriteString "**OPC?"
NumDmy = .ReadNumber
```

End With

,

With ENATDR

```
' TDR- Single Trigger
.WriteString ":TRIG:SING"
' TDR- Wait for measurement end
.WriteString "**OPC?"
NumDmy = .ReadNumber
' TDR- Autoscale
.WriteString ":DISP:ATR:SCAL:AUTO"
' *OPC? for checking the process finish
.WriteString "**OPC?"
NumDmy = .ReadNumber
```

End With

,

' Read TDR data (Channel1) - TDR Data should be read through ENA.

,

With ENA

```
' Query Number of Test Points
.WriteString ":SENS1:SWE:POIN?"
Nop = .ReadNumber
ReDim TimeData(Nop - 1)
ReDim Impedance(Nop * 2 - 1)
' Get the data for X axis.
.WriteString ":CALC1:SEL:DATA:XAX?"
TimeData() = .ReadList(ASCIIType_R8, ",", ",")
' Select the trace 1.
.WriteString ":CALC1:PAR1:SEL"
' Get the data for Y axis for active trace.
```

E5071C

```
.WriteString ":CALC1:DATA:FDAT?"  
Impedance() = .ReadList(ASCIIType_R8, ",")  
' Get limit line test result  
.WriteString ":CALC1:LIM:FAIL?"  
PassFail = .ReadNumber
```

```
k = 0  
For i = 0 To Nop - 1  
    Cells(i + 7, 4) = TimeData(i)  
    Cells(i + 7, 5) = Impedance(k)  
    k = k + 2  
Next i  
Cells(4, 5) = PassFail  
.WriteString "**OPC?"  
NumDmy = .ReadNumber  
End With  
  
'MsgBox "Measurement for Ch1 is done.", vbOKOnly  
Exit Sub
```

errorhandler:

```
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
```

End Sub

Setup for Ch2

```
Sub SetupCh2()  
    On Error GoTo errorHandler  
    With ENA  
        ,  
        ' Setup Network measurement(S-Parameter) on channel 2.  
        ,  
        ' ENA- Set Start Frequency  
.WriteString ":SENS2:FREQ:STAR 1E9"  
        ' ENA- Set Stop Frequency  
.WriteString ":SENS2:FREQ:STOP 3E9"
```



```

' ENA- Set IFBW
.WriteString ":SENS2:BAND 1E3"
' ENA- Fixture Simulator ON
.WriteString ":CALC2:FSIM:STAT ON"
' ENA- Set topology at Balance-Balance
.WriteString ":CALC2:FSIM:BAL:DEV BBAL"
' ENA- Assign Physical Ports
.WriteString ":CALC2:FSIM:BAL:TOP:BBAL 1,2,3,4"
' ENA- Balance-Balance Conversion ON
.WriteString ":CALC2:FSIM:BAL:PAR1:STAT ON"
' ENA- Select Measurement Parameter at Sdd21
.WriteString ":CALC2:FSIM:BAL:PAR1:BBAL SDD21"
'

' ENA- Limit Test ON
.WriteString ":CALC2:LIM ON"
' ENA- Limit Line On
.WriteString ":CALC2:LIM:DISP ON"
' ENA- Edit Limit Table
.WriteString ":CALC2:LIM:DATA 3,2,100e6,1.25e9,-1.5,-5,2,1.25e9,2.5e9,-5,-7.5,2,2.5e9,7.5e9,-
7.5,-25"
'

' *OPC? for checking the process finish
.WriteString "**OPC?"
NumDmy = .ReadNumber
End With

FrmEcal.Show

MsgBox "Setup Ch2 is done.", vbOKOnly
Exit Sub

errorhandler:
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub

```

EcalCalibration (This is called from SetupCh2)

E5071C

Sub EcalCalibration()

With ENA

' ENA- 4 Port Ecal

.WriteString ":SENS2:CORR:COLL:ECAL:SOLT4 1,2,3,4"

' *OPC? for checking the process finish

.WriteString "**OPC?"

NumDmy = .ReadNumber

End With

End Sub

Measurement for channel 2

Sub MeasCh2()

On Error GoTo errorhandler

Dim FreqData() As Double, InserionLoss() As Double

Dim Nop As Integer, PassFail As Integer, i As Integer, k As Integer

,

With ENA

' ENA- Set active channel at channel 2

.WriteString ":DISP:WIND2:ACT"

' ENA- Single Trigger

.WriteString ":INIT2:TRIG:SING"

' ENA- Wait for measurement end

.WriteString "**OPC?"

NumDmy = .ReadNumber

End With

,

' Read Newtwork Meaurement (Channel 2) data

,

With ENA

' ENA- Get Number of Points of data

.WriteString ":SENS2:SWE:POIN?"

Nop = .ReadNumber

' Get the data for x-axis (Frequency)

ReDim FreqData(Nop - 1)

.WriteString ":SENS2:FREQ:DATA?"

FreqData() = .ReadList(ASCIIType_R8, ",")

' Select the trace 1.

```
.WriteString ":CALC2:PAR1:SEL"
' Get the data for Y-axis (Sdd21 - LogMag)
ReDim InersionLoss(Nop * 2 - 1)
.WriteString ":CALC2:DATA:FDATA?"
InersionLoss() = .ReadList(ASCIIType_R8, ",")
' Get limit line test result
.WriteString ":CALC2:LIM:FAIL?"
PassFail = .ReadNumber
'
k = 0
For i = 0 To Nop - 1
    Cells(i + 7, 7) = FreqData(i)
    Cells(i + 7, 8) = InersionLoss(k)
    k = k + 2
Next i
Cells(4, 8) = PassFail
' *OPC? for checking the process finish
.WriteString "**OPC?"
NumDmy = .ReadNumber
End With

Exit Sub
```

errorhandler:

```
MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Close Instrument

```
Sub CloseInstrument()

    ENA.IO.Close
    ENATDR.IO.Close
    MsgBox "E5071C and E5071C-TDR are closed.", vbOKOnly

End Sub
```

Command Reference

Command Reference

- Notational Conventions
- COM Object Model
- Command Finder

Other topics about Programming

Notational Conventions

This section describes the notational conventions used for the description of commands reference.

Object Type

Object type describes different types of E5071C COM objects. The E5071C provides properties and methods as COM objects. COM objects which set (send)/read (return) the state of the E5071C using variables are defined as property and COM objects which does other processing are defined as method.

COM objects used only to read the state of the E5071C are indicated with "**Read-only**" and ones used only to set the state of the E5071C are indicated by "**Write-only**". COM object that can both read and write data to the E5071C are indicated by '**Read-Write**'.

Syntax

Syntax describes the syntax for sending a COM object from the E5071C VBA to the E5071C. The syntax consists of two parts: the object part and the set/read part, with an equal "=" inserted between them. Variables are indicated by italicized letters. Variables with () are indices. For indices with () having their preset values, you can omit "(*variable*)," and, if omitted, the preset values are automatically set.

The following table describes the 3 types of syntax for coding using objects:

Type	Description
"Object (property) = <i>variable</i> ":	Set the stat of the E5071C.
<i>variable</i> =object (property):	Read the stat of the E5071C.
"Object (method)":	Perform some processing in the E5071C.

Description

Description describes how to use the COM object or the operation when executed.

Variable

Variable provides description about different variables that can be used with the COM objects. It gives the description, data type, allowable range,

preset value, unit, resolution, and notes for *variable (italic)* shown in the syntax.

NOTE

Variables declared as the string data type (String) are not case-sensitive. For variables of the string type that indicate arguments (written as *Param* in the syntax), you can omit lower-case letters.

The data types of the E5071C COM objects include 5 types as shown in the following table. Before using variables, declare the data type of each variable. If you do not declare the data type of a variable, it is automatically processed as a variant type.

Data type	Name	Consumed memory	Range
Long	Long integer type	4 bytes	-2,147,483,648 to 2,147,483,647
Double	Double precision floating point type	8 bytes	For a negative value: -1.79769313486232E+308 to -4.94065645841247E-324 For a positive value: -1.79769313486232E+308 to -4.94065645841247E-324
Boolean	Boolean type	2 bytes	For COM: True or False (For SCPI: ON or OFF)
String	Character string type	1 byte / alphanumeric character	Up to approximately 2 billion characters
Variant	Variant type	16 bytes	No limitation

Examples

Examples provides a sample example of using the object through coding with the E5071C VBA.

Related Objects

Related objects provides information about other objects that are similar/related with the object.

Equivalent Key

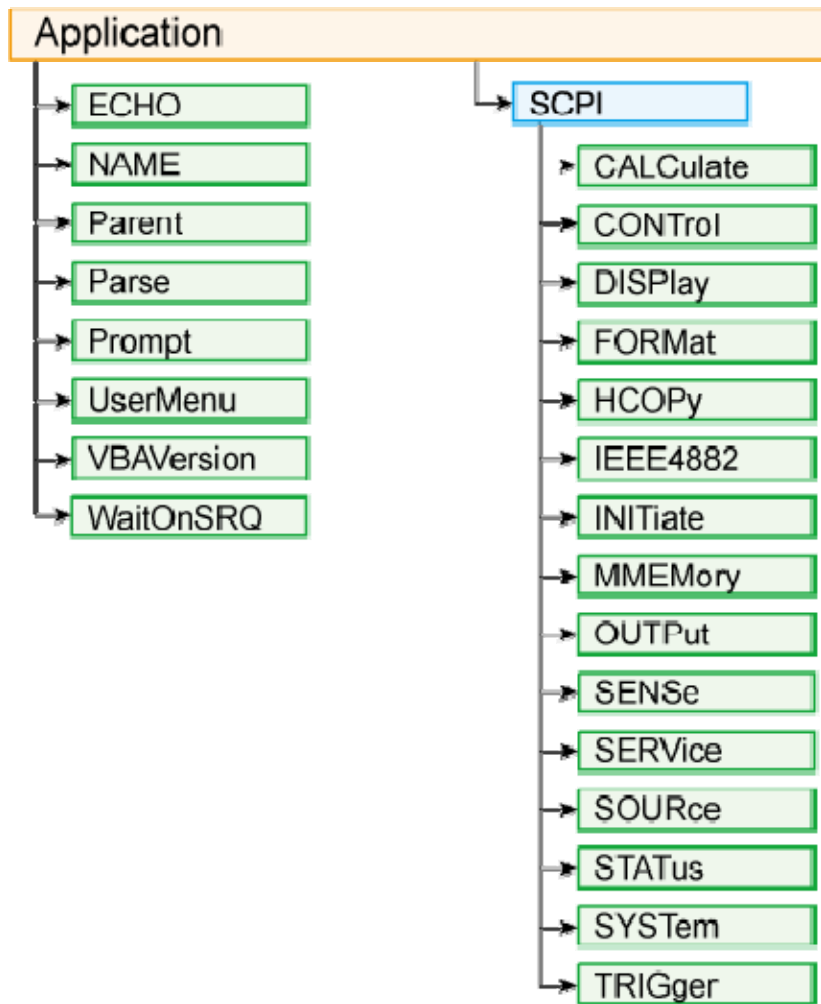
Equivalent key shows the operational procedure of the front panel keys that has the same effect as this object.

Equivalent SCPI command

Equivalent SCPI command shows the SCPI command to execute from an external controller. Its syntax, query response, and example of use are provided.

COM Object Model

The COM objects provided for the E5071C are structured hierarchically as shown below.



e5071c473

Application Objects

The Application objects are at the top of the hierarchy of the E5071C COM object model. They consist of 7 objects dedicated to the COM interface and SCPI objects corresponding to SCPI commands.

SCPI Objects

The SCPI objects are created to realize the SCPI commands of the E5071C with the COM interface.

The conversion rules from the SCPI commands when writing SCPI object messages are as follows:

- SCPI. must be at the beginning. Notice that the IEEE common commands start with SCPI.IEEE4882. and "*" is omitted.
- Replace colons (:) used as the hierarchical separator symbol with dots (.).
- The number written in the object message is specified with ().
- You cannot omit the command message in the syntax.

SCPI command	COM object
OUTPUT 717;":SOUR1:POW -10"	SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude = -10
OUTPUT 717;":SENS1:CORR:COLL:METH:TYPE?" ENTER 717;A\$	A = SCPI.SENSE(1).CORRection.COLLect.METHOD:TYPE
OUTPUT 717;"*CLS"	SCPI.IEEE4882.CLS

Application Objects

ECHO

Object type

Method (**Write-only**)

Syntax

ECHO *V1,V2, ,V10*

ECHO *SCPI object*

Description

Provides display in the echo window. (No read)

This command is difference from SCPI.DISPlay.ECHO.DATA.

- Up to 10 data items can be displayed.
- Data is displayed as the declared data type without a cast.

Parameter	<i>V1,V2, ,V10</i>
Description	Data you want to display in the echo window.
Data type	Variant type (Variant)

Examples

```
Dim Nop As Long
Dim i As Integer
Dim Fdata As Variant
Nop = SCPI.SENSE(1).SWEp.POINTs
Fdata = SCPI.CALCulate(1).SElected.DATA.FDATA
ECHO "Test Results"
For i=1 to Nop
    ECHO i, Fdata(2*i-2), Fdata(2*i-1)
Next i
```

ECHO SCPI.SYSTem.ERRor

Related objects

SCPI.DISPlay.ECHO.DATA

Equivalent key

No equivalent key is available on the front panel.

NAME**Object type**

Property

Syntax

App = NAME

Description

Reads out the application name of VBA. E5071C is always read out. (Read only)

Variable

Parameter	<i>App</i>
Description	Application name
Data type	Character string type (String)

Examples

```
Dim Inst As String  
Inst = NAME  
ECHO Inst
```

Equivalent key

No equivalent key is available on the front panel.

Parse**Object type**Method (**Write-only**)**Syntax**Parse(*Scpi*)*Return* = Parse(*Scpi*?)**Description**

Executes an SCPI command of the E5071C.

The Parse object is a little slower in the execution speed than the COM object which has the same function as the SCPI command because it must parse the message string of the SCPI command.

Variable

Parameter	<i>Scpi</i>
Description	SCPI command
Data type	Character string type (String)

Parameter	<i>Return</i>
Description	Response (query) of the SCPI command
Data type	Character string type (String)

Examples

```
Dim Start As String
Parse(":SENS1:FREQ:STAR 100E6")
Start = Parse(":SENS1:FREQ:STAR?")
```

```
Dim TtlLbl As String
Parse(":DISP:WIND1:TITL:DATA ""filter""")
TtlLbl = Parse(":DISP:WIND1:TITL:DATA?")
```

```
Dim Fmt As String
Parse(":CALC1:PAR2:SEL")
```

```
Parse(":CALC1:FORM SMIT")  
Fmt = Parse(":CALC1:FORM?")
```

```
Dim BckLght As String  
Parse(":SYST:BACK OFF")  
BckLght = Parse(":SYST:BACK?")
```

Equivalent key

No equivalent key is available on the front panel.

Prompt

Object type

Method (**Write-only**)

Syntax

Prompt(*Mes*)

Description

Displays the message you specify on the instrument status bar (at the bottom of the LCD display) and suspends the program until the **Macro Setup > Continue** button is pressed. (No read)

- When using this object, execute the program with the Visual Basic closed since you need to press the **Macro Setup > Continue**. For more information, see Running a Program from the E5071C Measurement Screen. If you need to abort the program, see Stopping with the Dialog Box Appeared.

Variable

Parameter	<i>Mes</i>
Description	Message
Data type	Character string type (String)

Examples

Prompt ("Connect DUT, and then press [Continue]")

Equivalent key

No equivalent key is available on the front panel.

UserMenu.Item(*Key_id*).Caption**Object type**

Property

SyntaxUserMenu.Item(*Key_id*).Caption = *Lbl**Lbl* = UserMenu.Item(*Key_id*).Caption**Description**Sets the label name of the user menu function softkeys 1 to 10 (*Key_id*).**Variable**

Parameter	<i>Key_id</i>
Description	Softkey number for the user menu function
Data type	Long integer type (Long)
Range	1 to 10
Note	You cannot omit this because it does not have a preset value. If the specified variable is out of the valid setting range, an error occurs when executed.

Parameter	<i>Lbl</i>
Description	Softkey label name for the user menu function
Data type	Character string type (String)
Preset value	Varies depending on the specified softkey number.

Examples

```
Dim KeyLbl As String
UserMenu.Item(1).Caption = "Meas"
KeyLbl = UserMenu.Item(1).Caption
```

Equivalent key

No equivalent key is available on the front panel.

UserMenu.Item(Key_id).Enabled**Object type**

Property

SyntaxUserMenu.Item(Key_id).Enabled = *Status**Status* = UserMenu.Item(Key_id).Enabled**Description**

Makes the user menu function softkeys 1 to 10 (*Key_id*) enabled/disabled. The softkey label disabled is displayed with the grey color and its softkey cannot be pressed.

Variable

Parameter	<i>Status</i>
Description	Enabled/disabled for the user menu function softkey
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or -1 Makes the softkey enabled. • False or 0 Makes the softkey disabled.
Preset value	True or -1

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
Dim KeyEna As Boolean
UserMenu.Item(10).Enabled = False
KeyEna = UserMenu.Item(10).Enabled
```

Related objects

UserMenu.Press

Equivalent key

No equivalent key is available on the front panel.

UserMenu.PRESet

Object type

Method (**Write-only**)

Syntax

UserMenu.PRESet

Description

Presets the label name and enabled/disabled settings for the user menu softkeys. (No read)

Examples

UserMenu.PRESet

Related objects

UserMenu.Item.Caption

UserMenu.Item.Enabled

Equivalent key

Macro Setup > Preset User Menu

E5071C

UserMenu.Press(*Key_id*)

Object type

Method (**Write-only**)

Syntax

UserMenu.Press(*Key_id*)

Description

Presses one of the user menu function softkeys 1 to 10 (*id*). (No read)

Variable

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

UserMenu.Press(1)

Related objects

UserMenu.Item.Enabled

Equivalent key

Macro Setup > **User Menu** > **Button 1 to Button 10**

UserMenu.Show

Object type

Method (**Write-only**)

Syntax

UserMenu.Show

Description

Displays the user menu function softkeys in the softkey area. (No read)

Examples

UserMenu.Show

Equivalent key

Macro Setup > **User Menu**

UserMenu_OnPress(ByVal *Key_id* As Long)**Object type**

Event

Description

Executes the processing when one of the user menu function softkeys 1 to 10 (*Key_id*) is pressed. Write the processing in the "UserMenu" object. For more information on its use, see Executing a Procedure with a Softkey (User Menu Function).

Variable

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
Private Sub UserMenu_OnPress(ByVal id As Long)
If id = 1 Then
MsgBox "Button 1 was pressed."
ElseIf id = 10 Then
MsgBox "Button 10 was pressed."
End If
End Sub
```

Equivalent key

No equivalent key is available on the front panel.

VBAVersion

Object type

Property

Syntax

Vers = VBAVersion

Description

Reads out the version information of VBA installed in the E5071C. (Read only)

Variable

Parameter	<i>Vers</i>
Description	VBA version information
Data type	Character string type (String)

Examples

```
Dim Version As String
Version = VBAVersion
ECHO Version
```

Equivalent key

From the **Help** menu of the Visual Basic editor, click **About Microsoft Visual Basic....**

WaitOnSRQ**Object type**Method (**Write-only**)**Syntax***WaitOnSRQ Status, Timeout***Description**

Suspends the program for specified time until the RQS/MSS bit (bit 6) of the status byte register changes to 1. (No read)

Variable

Parameter	<i>Status</i>
Description	State of the RQS/MSS bit (read only)
Data type	Boolean type (Boolean)
Range	<p>One of the following is returned.</p> <ul style="list-style-type: none"> • True or -1 <p>1 has been received within the specified time.</p> <ul style="list-style-type: none"> • False or 0 <p>1 has not been received within the specified time due to timeout or abort.</p>

Parameter	<i>Timeout</i>
Description	Timeout time
Data type	Long integer type (Long)
Range	0 to 2,147,483,647
Preset value	-1 (infinity)
Unit	ms (millisecond)

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Stat As Boolean
SCPI.IEEE4882.CLS
SCPI.STATus.OPERation.PTRansition = 0
SCPI.STATus.OPERation.NTRansition = 16
SCPI.STATus.OPERation.ENABLE = 16
SCPI.IEEE4882.SRE = 128
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMEDIATE
WaitOnSRQ Stat, 10000
If Stat = True Then
  MsgBox "Done"
End If
```

Equivalent key

No equivalent key is available on the front panel.

ABORT

SCPI.ABORT

Object type

Method (**Write-only**)

Syntax

SCPI.ABORT

Description

This command aborts the measurement and changes the trigger sequence for all channels to idle state.

The channels for which the continuous startup mode is set to *ON* (setting to start up the trigger system continuously), changes immediately from idle to startup state. See Trigger System for details.

Examples

SCPI.ABORT

Related objects

SCPI.INITiate(Ch).IMMediate

SCPI.INITiate(Ch).CONTinuous

Equivalent key

Trigger > **Restart**

Equivalent SCPI command

Syntax

:ABORT

Example of use

10 OUTPUT 717;":ABORT"

CALCULATE

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORT(*Bpt*).IMAGinary

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORT(*Bpt*).IMAGinary = *Value*

Value =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORT(*Bpt*).IMAGinary

Description

This command sets/gets the impedance value (imaginary part) for the common port impedance conversion function, for balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

Variable

Parameter	<i>Ch</i>
Description	Channel number
Data type	Long integer type (Long)
Range	1 to 160
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.
Parameter	<i>Bpt</i>
Description	Balance port number
Data type	Long integer type (Long)

Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Value</i>
Description	Impedance value (imaginary part) for the common port impedance conversion function
Data type	Double precision floating point type (Double)
Range	-1E+18 to 1E+18
Preset value	0
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CImag As Double
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORT(1).IMAGinary = 1E5
CImag = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORT(1).IMAGinary
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).REAL
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).Z0.R
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > **Fixture Simulator** > **Cmn ZConversion** > **Port1(bal) Imag|Port2(bal) Imag|Port3(bal) Imag**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-
160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:IMAGinary
<numeric>
```

```
:CALCulate{[1]-
160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:IMAGinary?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:IMAG 30E6"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:IMAG?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).REAL

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).REAL =
Value

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).REAL

Description

This command sets/gets the impedance value (real part) for the common port impedance conversion function, for balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

- This command performs in the same way as SCPI.CALCulate(Ch).FSIM-ulator.BALun.CZConversion.BPORT(Bpt).Z0.R

Variable

Parameter	Value
Description	Impedance value (real part) for the common port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	25
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CReal As Double
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORT(1).REAL = 30
CReal = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORT(1).REAL
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).IMAGinary
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).Z0.R
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > **Fixture Simulator** > **Cmn ZConversion** > **Port1(bal) Real|Port2(bal) Real|Port3(bal) Real**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-
160}:FSIMulator:BALun:CZConversion:BPORT{[1]|2}:REAL <numeric>
:CALCulate{[1]-
160}:FSIMulator:BALun:CZConversion:BPORT{[1]|2}:REAL?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:REAL 30E6"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:REAL?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).Z0.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).Z0.R =
Value

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).Z0.R

Description

This command sets/gets the impedance value for the common port impedance conversion function, for balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

- This command clears setting value of
 SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.
 BPORT(Bpt).IMAGinary

Variable

Parameter	<i>Value</i>
Description	Impedance value for the common port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	25
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CZ0 As Double
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORT(1).Z0.R = 30
CZ0 = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORT(1).Z0.R
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).IMAGinary
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).REAL
SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > Fixture Simulator > Cmn ZConversion > Port1(bal) Real|Port2(bal) Real|Port3(bal) Real

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-
160}:FSIMulator:BALun:CZConversion:BPORT{[1]|2}:Z0[:R] <numeric>
:CALCulate{[1]-
160}:FSIMulator:BALun:CZConversion:BPORT{[1]|2}:Z0[:R]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:Z0 30"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:Z0?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe = *Status*
Status = SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe

Description

This command turns ON/OFF the common port impedance conversion function when the fixture simulator function is *ON*, for all the balance ports of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the common port impedance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the common port impedance conversion function. • False or OFF: Turns OFF the common port impedance conversion function.
Preset value	False or OFF

Examples

```
Dim ComZcon As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.STATe = True
ComZcon = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).IMAGinary
 SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).REAL
 SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORT(Bpt).Z0.R
 SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Cmn ZConversion > Cmn ZConversion**Equivalent SCPI command****Syntax**

```
:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:STATe  
{ON|OFF|1|0}  
:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:STAT ON"  
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:STAT?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVICE = *Param**Param* = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVICE

Description

This command selects the balance device type of the fixture simulator function, for the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Balance device type
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "SBALanced" Specifies the unbalance-balance (3 ports). • "BBALanced" Specifies the balance-balance (4 ports). • "SSBalanced" Specifies the unbalance-unbalance-balance (4 ports). • "BALanced" Specifies the unbalance (2 ports). (Rev. A.9.50 and above)
Preset value	For 4 port models, "SBALanced" For 2 port models, "BALanced"

Examples

```
Dim BalDev As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "bbal"
BalDev = SCPI.CALCulate(1).FSIMulator.BALun.DEVICE
```

Related objects

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BBALanced.PPORTsSCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORTsSCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SSBalanced.PPORTs

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Device**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DEvice
{SBALanced|BBALanced|SSBalanced|BALanced}
:CALCulate{[1]-160}:FSIMulator:BALun:DEvice?
```

Query response

```
{SBAL|BBAL|SSB|BAL}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DEV BBAL"
20 OUTPUT 717;":CALC1:FSIM:BAL:DEV?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.C

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.
C = *Value*

Value =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.
C

Description

This command sets/gets the C value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).TYPE object), for balance port 1 and balance port 2 (*Bpt*) of the selected channel (*Ch*),

Variable

Parameter	<i>Value</i>
Description	C value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	F (farad)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcC As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARAMeters.C = 12E-12
DmcC = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARAMeters.C
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
G
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
L
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
R
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe
```

Equivalent key

Analysis > Fixture Simulator > Diff Matching > C

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:PARAMeters:C <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:PARAMeters:C?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C?"
30 ENTER 717;A
```

**SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
G**

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
G = *Value*

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
G

Description

This command sets/gets the G value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit. BPORT(Bpt).TYPE object), for the balance port 1 and balance port 2 (*Bpt*) of selected channel (*Ch*),

Variable

Parameter	<i>Value</i>
Description	G value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	S (siemens)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcG As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARAMeters.G = 12E-12
DmcG = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARAMeters.G
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
C
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
L
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
R
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe
```

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **G**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:PARAMeters:G <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:PARAMeters:G?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.L

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
L = *Value*

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
L

Description

This command sets/gets the L value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE object), for the balance port 1 and balance port 2 (*Bpt*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	L value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	H (henry)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcL As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARameters.L = 12E-12
DmcL = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARameters.L
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.  
C
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.  
G
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.  
R
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe
```

Equivalent key

Analysis > Fixture Simulator > Diff Matching > L

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}  
:PARameters:L <numeric>  
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}  
:PARameters:L?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;" :CALC1:FSIM:BAL:DMC:BPOR1:PAR:L 12.3"  
20 OUTPUT 717;" :CALC1:FSIM:BAL:DMC:BPOR1:PAR:L?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.
R = *Value*

Value =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.
R

Description

This command sets/gets the R value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE object), for the balance port 1 and balance port 2 (*Bpt*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	R value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	ohm
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcR As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARAMeters.R = 12E-12
DmcR = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).PARAMeters.R
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
C
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
G
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
L
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe
```

Equivalent key

Analysis > Fixture Simulator > Diff Matching > R

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:PARAMeters:R <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:PARAMeters:R?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE =
Param

Param =

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE

Description

For balance ports 1 and 2 (*Bpt*) of selected channel (*Ch*), selects the type of the differential matching circuit.

Variable

Parameter	<i>Param</i>
Description	Type of the differential matching circuit
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "NONE": Specifies no-circuit. • "PLPC": Specifies the circuit that consists of shunt L and shunt C. • "USER": Specifies the user-defined circuit.
Preset value	"NONE"
Note	If you want to select the user-defined circuit, you must specify the 2-port touchstone file in which the proper information on the user-defined circuit is saved in advance. If you do not specify the appropriate file and you select the user-defined circuit, an error occurs when executed and NONE is automatically selected.

Examples

```
Dim CirType As String
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).TYPE = "plpc"
CirType = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).TYPE
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARAMeters.
C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).USER.FILena
me

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **Select Circuit**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
[:TYPE] {NONE|PLPC|USER}
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
[:TYPE]?
```

Query response

```
{NONE|PLPC|USER}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1 PLPC"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1?"
30 ENTER 717;A$
```

**SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORTt(Bpt).USER.FILena
me****Object type**Property (**Read-Write**)**Syntax**SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORTt(*Bpt*).USER.FILena
me = *File**File* =SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORTt(*Bpt*).USER.FILena
me**Description**

This command specifies the file in which the information on the user-defined differential matching circuit is saved (2-port touchstone file with the .s2p extension), for the balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Even if the specified file does not exist, no error occurs when you execute this object. However, when you set the type of the differential matching circuit to the user-defined circuit with the SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit. BPORTt(*Bpt*).TYPE object, an error occurs when executed.

Variable

Parameter	<i>File</i>
Description	2-port touchstone file name (extension: .s2p) for the differential matching circuit
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim DmcUser As String
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORTt(1).USER.FILename = "dmc.s2p"
```

```
DmcUser = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).USER.FILename
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORT(1).TYPE = "user"
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE
SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe
```

Equivalent key

Analysis > Fixture Simulator > Diff Matching > User File

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:USER:FILename <string>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORT{[1]|2}
:USER:FILename?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:USER:FIL ""Match_d.s2p""
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:USER:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe = *Status**Status* = SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Description

This command turns ON/OFF the differential matching circuit embedding function when the fixture simulator function is ON, for all the balance ports of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the differential matching circuit embedding function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the differential matching circuit embedding function. • False or OFF: Turns OFF the differential matching circuit embedding function.
Preset value	False or OFF

Examples

```
Dim DifMch As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.STATe = True
DifMch = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).PARameters.
R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).USER.FILena
me

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter(Tr).STATE

SCPI.CALCulate(Ch).FSIMulator.STATE

Equivalent key

Analysis > Fixture Simulator > Diff Matching > Diff Matching

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:STATE {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:STATE?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).IMAGinary

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).IMAGinary = Value

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).IMAGinary

Description

This command sets/gets the impedance value (imaginary part) for the differential port impedance conversion function, for the balance ports 1 and 2 (Bpt) of selected channel (Ch),

Variable

Parameter	Value
Description	Impedance value (imaginary part) for the differential port impedance conversion function
Data type	Double precision floating point type (Double)
Range	-1E+18 to 1E+18
Preset value	0
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim DImag As Double

SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORT(1).IMAGinary = 200

DImag = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORT(1).IMAGinary

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).REAL

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > Fixture Simulator > Diff ZConversion > Port1(bal) Imag|Port2(bal) Imag|Port3(bal) Imag

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:IMAGinary <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:IMAGinary?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:IMAG 300"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:IMAG?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).REAL

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).REAL =
Value

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).REAL

Description

This command sets/gets the impedance value (real part) for the differential port impedance conversion function, for the balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

- This command performs in the same way as "SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R"

Variable

Parameter	Value
Description	Impedance value (real part) for the differential port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	100
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DReal As Double
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORT(1).REAL = 200
DReal = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORT(1).REAL
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).IMAGinary
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > Fixture Simulator > Diff ZConversion > Port1(bal) Real|Port2(bal) Real|Port3(bal) Real

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORT{[1]|2}:REAL <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORT{[1]|2}:REAL?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:REAL 300"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:REAL?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R =
Value

Value =

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R

Description

This command sets/gets the impedance value for the differential port impedance conversion function, for the balance ports 1 and 2 (*Bpt*) of selected channel (*Ch*).

- This command clears setting value of
"SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.
BPORT(Bpt).IMAGinary"

Variable

Parameter	Value
Description	Impedance value for the differential port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	100
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DZ0 As Double
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORT(1).Z0.R = 200
DZ0 = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORT(1).Z0.R
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).IMAGinary
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).REAL
SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > **Fixture Simulator** > **Diff ZConversion** > **Port1(bal) Real|Port2(bal) Real|Port3(bal) Real**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORT{[1]|2}:Z0[:R] <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORT{[1]|2}:Z0[:R]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:Z0 300"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:Z0?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe**Object type**Property (**Read-Write**)**Syntax**

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe = *Status*
Status = SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe

Description

This command turns ON/OFF the differential port impedance conversion function when the fixture simulator function is ON, for all the balance ports of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the differential port impedance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the differential port impedance conversion function. • False or OFF: Turns OFF the differential port impedance conversion function.
Preset value	False or OFF

Examples

```
Dim DifZcon As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.STATe = True
DifZcon = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).IMAGinary
 SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).REAL
 SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORT(Bpt).Z0.R
 SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Diff ZConversion > Diff ZConversion
Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion:STATe
{ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).XXXX.PARAmeter.TNAME.XXXX

As mentioned in Selecting the Active Channel/Trace, if you are using E5071C revision A.9.60 and above, you can select a trace by the trace name, provided it has been defined earlier. To define a trace name, use the following command:

```
:CALC{[1]-160}:PAR{[1]-16}:TNAME:DATA
```

If the trace name is not defined but used, the following error occurs:

'51, Specified trace does not exist'

If the defined trace name already exist, the following error occurs:

'63, Duplicate trace name'

The following commands allow you to select the active trace directly by using **TNAME**:

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.BALanced.DEFin  
e
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.BBALanced.DEFi  
ne
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SBALanced.DEFi  
ne
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SSBALanced.DE  
Fine
```

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.STATe
```

```
SCPI.CALCulate(Ch).PARAmeter.TNAME.DEFine
```

```
SCPI.CALCulate(Ch).PARAmeter.TNAME.SELect
```

```
SCPI.CALCulate(Ch).PARAmeter.TNAME.SPORT
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.BALanced.DEFin
e(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.BALanced.DEFin
e(*Name*) = *Param*

Param =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.BALanced.DEFin
e(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "balance", for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "SDD11": Specifies Sdd11. • "SCD11": Specifies Scd11. • "SDC11": Specifies Sdc11. • "SCC11": Specifies Scc11.

Preset value	"SDD11"
---------------------	---------

Examples

```
Dim BalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "bal"
SCPI.CALCulate(1).PARAMeter(1).TNAME.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARAMeter.TNAME.BALanced.DEFine("duptrace") = "sdd11"
BalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAMeter.TNAME.BALanced.DEFine("duptrace")
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE
```

Equivalent key

Analysis > **Fixture Simulator** | **[Meas]** > **Sdd11** | **Scd11** | **Sdc11** | **Sc11**

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-
160}:FSIMulator:BALun:PARAMeter:TNAME:BALanced[:DEFine] <string>,
{SDD11|SCD11|SDC11|SCC11}
:CALCulate{[1]-
160}:FSIMulator:BALun:PARAMeter:TNAME:BALanced[:DEFine]? <string>
```

Query response

```
{SDD11|SCD11|SDC11|SCC11}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPTRACE""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:BAL ""DUPTRACE"",SDD21"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:BAL? ""DUPTRACE""
40 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.BBALanced.DEFi

ne(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.BBALanced.DEFi
ne(*Name*) = *Param*

Param =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.BBALanced.DEFi
ne(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "balance-balance", for the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(*Ch*).PARAmeter(Tr).TNAME.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "SDD11": Specifies Sdd11. • "SDD21": Specifies Sdd21. • "SDD12": Specifies Sdd12.

	<ul style="list-style-type: none"> • "SDD22": Specifies Sdd22. • "SCD11": Specifies Scd11. • "SCD21": Specifies Scd21. • "SCD12": Specifies Scd12. • "SCD22": Specifies Scd22. • "SDC11": Specifies Sdc11. • "SDC21": Specifies Sdc21. • "SDC12": Specifies Sdc12. • "SDC22": Specifies Sdc22. • "SCC11": Specifies Scc11. • "SCC21": Specifies Scc21. • "SCC12": Specifies Scc12. • "SCC22": Specifies Scc22. • "IMB1": Specifies Imbalance1. • "IMB2": Specifies Imbalance2. • "CMRR": Specifies CMRR (Sdd21/Scc21).
Preset value	"SDD11"

Examples

```
Dim BbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).PARAmeter(1).TNAME.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter.TNAME.BBALanced.DEFine("duptrace") = "sdd21"
BbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter.BBALanced.DEFine("duptrace")
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice
```

Equivalent key

Analysis > **Fixture Simulator** | **[Meas]** > **Sdd11|Sdd21|Sdd12|Sdd22|Scd11|Scd21|Scd12|Scd22|Sdc11|Sdc21|Sdc12|Sdc22|Scc11|Scc21|Scc12|Scc22|Imbalance1|Imbalance2|Sdd21|Scc21**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-
160}:FSIMulator:BALun:PARAmeter:TNAME:BBALanced[:DEFine] <string>,
{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC
21|SDC12|SDC22|SCC11|
SCC21|SCC12|SCC22|IMB1|IMB2| CMRR}
:CALCulate{[1]-
160}:FSIMulator:BALun:PARAmeter:TNAME:BBALanced[:DEFine]?
<string>
```

Query response

```
{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC
21|SDC12|SDC22|
SCC11|SCC21|SCC12|SCC22|IMB1|IMB2|CMRR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPTRACE""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:BBAL ""DUPTRACE"",SDD21"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:BBAL? ""DUPTRACE""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SBALanced.DEFi
ne(Name)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.SBALanced.DEFi
ne(*Name*) = *Param**Param* =SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.SBALanced.DEFi
ne(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-balance", for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA command.

This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "SSS11": Specifies Sss11. • "SDS21": Specifies Sds21. • "SSD12": Specifies Ssd12.

	<ul style="list-style-type: none"> • "SCS21": Specifies Scs21. • "SSC12": Specifies Ssc12. • "SDD22": Specifies Sdd22. • "SCD22": Specifies Scd22. • "SDC22": Specifies Sdc22. • "SCC22": Specifies Scc22. • "IMB": Specifies Imbalance. • "CMRR": Specifies CMRR (Sds21/Scs21). • "CMRR2": Specifies CMRR2 (Ssd12/Ssc12).
Preset value	"SSS11"

Examples

```
Dim SbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "sbal"
SCPI.CALCulate(1).PARAMeter(1).TNAME.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARAMeter.TNAME.SBALanced.DEFINE("duptrace") = "scs21"
SbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAMeter.TNAME.SBALanced.DEFINE("duptrace")
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE
```

Equivalent key

Analysis > Fixture Simulator|[Meas] > Sss11|Sds21|Ssd12|Scs21|Ssc12|Sdd22|Scd22|Sdc22|Scc22|Imbalance|Sds21/Scs21|Ssd12/Ssc12

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-
160}:FSIMulator:BALun:PARAMeter:TNAME:SBALanced[:DEFINE] <string>,
{SSS11|SDS21|
SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|CMRR|CMRR2}
```

E5071C

:CALCulate{[1]-
160}:FSIMulator:BALun:PARameter:TNAME:SBALanced[:DEFine]?
<string>

Query response

{SSS11|SDS21|SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|
CMRR|CMRR2}<newline> <^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPTRACE""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:SBAL ""DUPTRACE"",SDS21"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:SBAL? ""DUPTRACE""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SSBALanced.DEFine(*Name*)

Object type

Property (**Read-Write**)

Syntax

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SSBALanced.DEF
ine(Name) = Param
```

Param =

```
SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SSBALanced.DEF
ine(Name)
```

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-unbalance-balance", for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARAmeter(Tr).TNAME.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "SSS11": Specifies Sss11. • "SSS21": Specifies Sss21. • "SSS12": Specifies Sss12. • "SSS22": Specifies Sss22.

	<ul style="list-style-type: none"> • "SDS31": Specifies Sds31. • "SDS32": Specifies Sds32. • "SSD13": Specifies Ssd13. • "SSD23": Specifies Ssd23. • "SCS31": Specifies Scs31. • "SCS32": Specifies Scs32. • "SSC13": Specifies Ssc13. • "SSC23": Specifies Ssc23. • "SDD33": Specifies Sdd33. • "SCD33": Specifies Scd33. • "SDC33": Specifies Sdc33. • "SCC33": Specifies Scc33. • "IMB1": Specifies Imbalance1. • "IMB2": Specifies Imbalance2. • "IMB3": Specifies Imbalance3. • "IMB4": Specifies Imbalance4. • "CMRR1": Specifies CMRR (Sds31/Scs31). • "CMRR2": Specifies CMRR (Sds32/Scs32).
Preset value	"SSS11"

Examples

```
Dim SsbPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "ssb"
SCPI.CALCulate(1).PARAMeter(1).TNAME.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARAMeter.TNAME.SSBalanced.DEFINE("duptrace") = "sds31"
SsbPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAMeter.TNAME.SSBalanced.DEFINE("duptrace")
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE
```

Equivalent key

Analysis > **Fixture Simulator** | **[Meas]** > **Sss11** | **Sss21** | **Sss12** | **Sss22** | **Sds31** | **Sds32** | **Ssd13** | **Ssd23** | **Scs31** | **Scs32** | **Ssc13** | **Ssc23** | **Sdd33** | **Scd33** | **Sdc33** | **Scc33** | **Imbalance1** | **Imbalance2** | **Imbalance3** | **Imbalance4** | **Sds31/Scs31** | **Sds32/Scs32**

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-
160}:FSIMulator:BALun:PARameter:TNAME:SSBalanced[:DEFine]
<string>,
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS3
2|SSC13|SSC23|SDD33|SCD33|
SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CMRR1|CMRR2}
:CALCulate{[1]-
160}:FSIMulator:BALun:PARameter:TNAME:SSBalanced[:DEFine]?
<string>
```

Query response

```
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS3
2|SSC13|SSC23|SDD33|SCD33|SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CM
RR1|CMRR2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPTRACE""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:SSB ""DUPTRACE"",SDS31"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:SSB? ""DUPTRACE""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.STATe(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.STATe(*Name*) =
Status

Status =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter.TNAME.STATe(*Name*)

Description

This command turns ON/OFF the balance-unbalance conversion function when the fixture simulator function is ON, for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Status</i>
Description	ON/OFF of the balance-unbalance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the balance-unbalance conversion function. • False or OFF: Turns OFF the balance-unbalance conversion function.
Preset value	False or OFF

Examples

```
Dim BalMode As Boolean
SCPI.CALCulate(1).PARAmeter(1).TNAME.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter.TNAME.STATe("duptrace") = True
BalMode = SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter.TNAME.STATe("duptrace")
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > BalUn

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARAmeter:TNAME:STATe
<string>, {ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:BALun:PARAmeter:TNAME:STATe?
<string>
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPTRACE""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAME:STAT ""DUPTRACE"",ON"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:STAT? ""DUPTRACE""
40 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter(Tr).BALanced.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter(Tr).BALanced.DEFine =
Param

Param =

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter(Tr).BALanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "balance", for the selected trace (*Tr*) of the selected channel (*Ch*).

This command is available at the firmware revision A.9.50 and above.

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "SDD11": Specifies Sdd11. • "SCD11": Specifies Scd11. • "SDC11": Specifies Sdc11. • "SCC11": Specifies Scc11.
Preset value	"SDD11"

Examples

```
Dim BalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bal"
SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).BALanced.DEFine = "sdd11"
BalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).BALanced.DEFine
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > Fixture Simulator|[Meas] > Sdd11|Scd11|Sdc11|Scc11

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}  
:BALanced[:DEFine] {SDD11|SCD11|SDC11|SCC11}  
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}  
:BALanced[:DEFine]?
```

Query response

```
{SDD11|SCD11|SDC11|SCC11}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BAL SDD21"  
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BAL?"  
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).BBALanced.DEFine **Object type**

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).BBALanced.DEFine
 = *Param*

Param =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).BBALanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "balance-balance", for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	<p>Select from the following.</p> <ul style="list-style-type: none"> • "SDD11": Specifies Sdd11. • "SDD21": Specifies Sdd21. • "SDD12": Specifies Sdd12. • "SDD22": Specifies Sdd22. • "SCD11": Specifies Scd11. • "SCD21": Specifies Scd21. • "SCD12": Specifies Scd12. • "SCD22": Specifies Scd22. • "SDC11": Specifies Sdc11. • "SDC21": Specifies Sdc21. • "SDC12": Specifies Sdc12. • "SDC22": Specifies Sdc22. • "SCC11": Specifies Scc11. • "SCC21": Specifies Scc21. • "SCC12": Specifies Scc12.

	<ul style="list-style-type: none"> • "SCC22": Specifies Scc22. • "IMB1": Specifies Imbalance1. • "IMB2": Specifies Imbalance2. • "CMRR": Specifies CMRR (Sdd21/Scc21).
Preset value	"SDD11"

Examples

```
Dim BbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).BBALanced.DEFINE = "sdd21"
BbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).BBALanced.DEFINE
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE
```

Equivalent key

Analysis > **Fixture Simulator** | **[Meas]** > **Sdd11|Sdd21|Sdd12|Sdd22|Scd11|Scd21|Scd12|Scd22|Sdc11|Sdc21|Sdc12|Sdc22|Scc11|Scc21|Scc12|Scc22|Imbalance1|Imbalance2|Sdd21/Scc21**

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARAmeter{[1]-16}
:BBALanced[:DEFine]
{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC
21|SDC12|SDC22|SCC11|
SCC21|SCC12|SCC22|IMB1|IMB2| CMRR}
:CALCulate{[1]-160}:FSIMulator:BALun:PARAmeter{[1]-16}
:BBALanced[:DEFine]?
```

Query response

```
{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC
21|SDC12|SDC22|
SCC11|SCC21|SCC12|SCC22|IMB1|IMB2|CMRR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BBAL SDD21"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BBAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SBALanced.DEFine **Object type**

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SBALanced.DEFine
 = *Param*

Param =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SBALanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-balance", for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	<p>Select from the following.</p> <ul style="list-style-type: none"> • "SSS11": Specifies Sss11. • "SDS21": Specifies Sds21. • "SSD12": Specifies Ssd12. • "SCS21": Specifies Scs21. • "SSC12": Specifies Ssc12. • "SDD22": Specifies Sdd22. • "SCD22": Specifies Scd22. • "SDC22": Specifies Sdc22. • "SCC22": Specifies Scc22. • "IMB": Specifies Imbalance. • "CMRR": Specifies CMRR (Sds21/Scs21). • "CMRR2": Specifies CMRR2 (Ssd12/Ssc12).
Preset value	"SSS11"

Examples

```
Dim SbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "sbal"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).SBALanced.DEFine = "scs21"
SbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).SBALanced.DEFine
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator** | **[Meas]** > **Sss11|Sds21|Ssd12|Scs21|Ssc12|Sdd22|Scd22|Sdc22|Scc22|Imbalance|Sds21/Scs21|Ssd12/Ssc12**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:SBALanced[:DEFine] {SSS11|SDS21|
SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|CMRR|CMRR2}
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:SBALanced[:DEFine]?
```

Query response

```
{SSS11|SDS21|SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|
CMRR|CMRR2}<newline> <^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SBAL SDS21"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SBAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SSBalanced.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SSBalanced.DEFine
= *Param*

Param =SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SSBalanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-unbalance-balance", for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	<p>Select from the following.</p> <ul style="list-style-type: none"> • "SSS11": Specifies Sss11. • "SSS21": Specifies Sss21. • "SSS12": Specifies Sss12. • "SSS22": Specifies Sss22. • "SDS31": Specifies Sds31. • "SDS32": Specifies Sds32. • "SSD13": Specifies Ssd13. • "SSD23": Specifies Ssd23. • "SCS31": Specifies Scs31. • "SCS32": Specifies Scs32. • "SSC13": Specifies Ssc13. • "SSC23": Specifies Ssc23. • "SDD33": Specifies Sdd33. • "SCD33": Specifies Scd33. • "SDC33": Specifies Sdc33.

	<ul style="list-style-type: none"> • "SCC33": Specifies Scc33. • "IMB1": Specifies Imbalance1. • "IMB2": Specifies Imbalance2. • "IMB3": Specifies Imbalance3. • "IMB4": Specifies Imbalance4. • "CMRR1": Specifies CMRR (Sds31/Scs31). • "CMRR2": Specifies CMRR (Sds32/Scs32).
Preset value	"SSS11"

Examples

```
Dim SsbPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "ssb"
SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).SSBalanced.DEFine = "sds31"
SsbPara = SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).SSBalanced.DEFine
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice
```

Equivalent key

Analysis > **Fixture Simulator**|[Meas] > **Sss11|Sss21|Sss12|Sss22|Sds31|Sds32|Ssd13|Ssd23|Scs31|Scs32|Ssc13|Ssc23|Sdd33|Scd33|Sdc33|Scc33|Imbalance1|Imbalance2|Imbalance3|Imbalance4|Sds31/Scs31|Sds32/Scs32**

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARAmeter{[1]-16}
:SSBalanced[:DEFine]
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS32|SSC13|SSC23|SDD33|SCD33|SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CMRR1|CMRR2}
:CALCulate{[1]-160}:FSIMulator:BALun:PARAmeter{[1]-16}
:SSBalanced[:DEFine]?
```

Query response

```
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS32|SSC13|SSC23|SDD33|SCD33|SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CMRR1|CMRR2}<newline><^END>
```

Example of use

E5071C

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SSB SDS31"  
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SSB?"  
30 ENTER 717;A$
```


SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*). STATE**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).STATE = *Status**Status* = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).STATE**Description**

This command turns ON/OFF the balance-unbalance conversion function when the fixture simulator function is ON, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the balance-unbalance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the balance-unbalance conversion function. • False or OFF: Turns OFF the balance-unbalance conversion function.
Preset value	False or OFF

Examples

Dim BalMode As Boolean

SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).STATE = True

BalMode = SCPI.CALCulate(1).FSIMulator.BALun.PARAmeter(1).STATE

Related objectsSCPI.CALCulate(*Ch*).FSIMulator.STATE**Equivalent key****Analysis > Fixture Simulator > BalUn****Equivalent SCPI command****Syntax**

E5071C

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}:STATe
{ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:STAT ON"  
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:STAT?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BALanced.PPORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BALanced.PPORTs = *Ports**Ports* = SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BALanced.PPORTs

Description

This command assigns each port when the balance device type is "balance", for the selected channel (*Ch*).

To set the balance device type to "balance," specify BAL with the SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVICE object.

This command is available at the firmware revision A.9.50 and above.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Balance device type. <i>Ports(1)</i> :Port number assigned to port b in Balance device type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 2
Preset value	Ports(0):1 / Ports(1):2
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

E5071C

```
Dim BbalPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORTs = Array(1,4)
BbalPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORTs
```

```
Dim BbalPort(1) As Variant
Dim Ref As Variant
BbalPort(0) = 1
BbalPort(1) = 4
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORTs = BbalPort
Ref = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORTs
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > Fixture Simulator > Topology > Port1(bal)

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BALanced
[:PPORTs] <numeric 1>, <numeric 2>
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BALanced
[:PPORTs]?
```

Query response

```
{numeric 1},{numeric 2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BAL 1,4"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BAL?"
30 ENTER 717;A,B
```

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.BBALanced.PPORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BBALanced.PPORTs =
Ports

Ports =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BBALanced.PPORTs

Description

This command assigns each port when the balance device type is "balance-balance", for the selected channel (*Ch*).

To set the balance device type to "balance-balance," specify BBAL with the SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVICE object.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Balance device type. <i>Ports(1)</i> :Port number assigned to port b in Balance device type. <i>Ports(2)</i> :Port number assigned to port c in Balance device type. <i>Ports(3)</i> :Port number assigned to port d in Balance device type. <p>The index of the array starts from 0.</p>
	Data type
Range	Variant type (Variant)
Preset value	1 to 4
Resolution	Ports(0):1 / Ports(1):2 / Ports(2):3 / Ports(3):4
Note	1
	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same</p>

port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim BbalPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORTs = Array(3,4,1,2)
BbalPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORTs
```

```
Dim BbalPort(3) As Variant
Dim Ref As Variant
BbalPort(0) = 3
BbalPort(1) = 4
BbalPort(2) = 1
BbalPort(3) = 2
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORTs = BbalPort
Ref = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORTs
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(bal)**

Analysis > **Fixture Simulator** > **Topology** > **Port2(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BBALanced
[:PPORTs] <numeric 1>, <numeric 2>,<numeric 3>,<numeric 4>
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BBALanced
[:PPORTs]?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BBAL 1,2,3,4"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BBAL?"
30 ENTER 717;A,B,C,D
```

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.PROPerTy.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.PROPerTy.STATe = *Status*
Status = SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.PROPerTy.STATe

Description

This command turns on/off the property display for the topology setting when using the balance-unbalance conversion, for the selected channel (Ch).

Variable

Parameter	<i>Status</i>
Description	On/off of the property display of the topology setting
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns on the property display. • False or OFF :Turns off the property display.
Preset value	False or OFF

Examples

```
Dim TopProp As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.PROPerTy.STATe = True
TopProp = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.PROPerTy.STATe
```

Equivalent key

Analysis > Fixture Simulator > Topology > Property

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology  
:PROPerTy:STATe {ON|OFF|1|0}
```

E5071C

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology
:PROPerTy:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:PROP:STAT ON"  
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:PROP:STAT?"  
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORTs =
Ports

Ports =

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORTs

Description

This command assigns each port when the balance device type is "unbalance-balance", for the selected channel (*Ch*).

To set the balance device type to "unbalance-balance," specify SBAL with the SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVICE object.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 3-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Balance device type. <i>Ports(1)</i> :Port number assigned to port b in Balance device type. <i>Ports(2)</i> :Port number assigned to port c in Balance device type. <p>The index of the array starts from 0.</p>
	Data type
Range	Variant type (Variant)
Preset value	1 to 4
Resolution	Ports(0):1 / Ports(1):2 / <i>Ports(2)</i> :3
	1

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim SbalPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "sbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.SBALanced.PPORTs = Array(1,3,4)
SbalPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORTs
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE
```

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(se)**

Analysis > **Fixture Simulator** > **Topology** > **Port2(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SBALanced
[:PPORTs] <numeric 1>, <numeric 2>,<numeric 3>
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SBALanced
[:PPORTs]?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SBAL 1,2,3"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SBAL?"
30 ENTER 717;A,B,C
```

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SSBalanced.PPORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SSBalanced.PPORTs =
Ports

Ports =

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SSBalanced.PPORTs

Description

This channel assigns/reads each port when the balance device type is "unbalance-unbalance-balance", for the selected channel (*Ch*).

To set the balance device type to "unbalance-unbalance-balance," specify SSB with the SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE object.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Balance device type. <i>Ports(1)</i> :Port number assigned to port b in Balance device type. <i>Ports(2)</i> :Port number assigned to port c in Balance device type. <i>Ports(3)</i> :Port number assigned to port d in Balance device type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Preset value	<i>Ports(0)</i> :1 / <i>Ports(1)</i> :2 / <i>Ports(2)</i> :3 / <i>Ports(3)</i> :4
Resolution	1

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim SsbPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVICE = "ssb"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.SSBalanced.PPORTs = Array(1,4,2,3)
SsbPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.SSBalanced.PPORTs
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE
```

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(se)**

Analysis > **Fixture Simulator** > **Topology** > **Port2(se)**

Analysis > **Fixture Simulator** > **Topology** > **Port3(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SSBalanced
[:PPORTs] <numeric 1>, <numeric 2>,<numeric 3>,<numeric 4>
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SSBalanced
[:PPORTs]?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SSB 1,2,3,4"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SSB?"
30 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename = *File**File* = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename

Description

This command specifies a file in which the information of network (*Nwk*) (which you want to embed/de-embed using the 4-port network embedding/de-embedding feature) is saved for the selected channel (*Ch*). The file is saved as a 4-port touchstone file with the ".s4p" extension.

- Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash) or "/" (slash).

For information on network numbers, refer to Connection Type.

This function is available with the Firmware revision 3.50 or greater.

Variable

Parameter	<i>File</i>
Description	4-port touchstone file name (extension: .s4p) for the 4-port network embedding/de-embedding feature
Data type	Character string type (String)
Range	254 characters or less
Preset value	""
Note	When the processing type is set to NONE, even if the specified file does not exist, no error occurs when you execute this object. However, when you set the processing type to embedding/de-embedding with the SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).TYPE object, an error occurs.

Examples

E5071C

```
Dim Emb As String
SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).FILename = "network.s4p"
Emb = SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).FILename
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE
SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe
```

Equivalent key

Analysis > Fixture Simulator > De-Embedding S4P > Topology > User File (nwk1)|User File (nwk2) | User File (nwk3) | User File (nwk4)

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}: FILename
<string>
:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}: FILename?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:FIL ""Network.s4p""
20 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE = *File**File* = SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE

Description

This command sets/gets the processing type for networks (*Nwk*)., for the 4-port network embedding/de-embedding feature for the selected channel (*Ch*).

- This function is available with the Firmware revision 3.50 or greater.

Variable

Parameter	<i>Param</i>
Description	Processing type
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "NONE": Specifies no-processing. • "EMBed": Specifies embedding. • "DEEMbed": Specifies de-embedding.
Preset value	"NONE"
Note	Before selecting embedding/de-embedding, use the SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).FILENAME object to specify the 4-port touchstone file in which the information on the network is saved. If you do not specify the appropriate file and you select embedding/de-embedding, a runtime error occurs and NONE is automatically selected.

Examples

```
Dim EmbType As String
SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).FILENAME = "network.s4p"
```

E5071C

```
SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).TYPE = "deem"  
EmbType = SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).TYPE
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).FILENAME  
SCPI.CALCulate(Ch).FSIMulator.EMBed.STATE
```

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** >
Type (nwk1)|Type (nwk2)|Type (nwk3)|Type (nwk4) > **None|Embed|De-Embed**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}:TYPE  
{NONE|EMBed|DEEMbed}  
:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}:TYPE?
```

Query response

```
{NONE|EMB|DEEM}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:TYPE DEEM"  
20 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:TYPE?"  
30 ENTER 717;A$
```


SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe = *Status**Status* = SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe**Description**

This command turns ON/OFF the 4-port network embedding/de-embedding feature when the fixture simulator feature is *ON*, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	Turns ON/OFF the 4-port network embedding/de-embedding feature.
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the 4-port network embedding/de-embedding feature. • False or OFF: Turns OFF the 4-port network embedding/de-embedding feature.
Preset value	False or 0

Examples

```
Dim Emb As Boolean
SCPI.CALCulate(1).FSIMulator.EMBed.STATe = True
Emb = SCPI.CALCulate(1).FSIMulator.EMBed.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).FILEname

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key**Analysis > Fixture Simulator > De-Embedding S4P > De-Embedding S4P****Equivalent SCPI command**

E5071C

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:STATe {ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:EMBed:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:STAT ON"  
20 OUTPUT 717;":CALC1:FSIM:EMB:STAT?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs = *Ports**Ports* = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs

Description

This command sets/gets the test port assignment when the connection type (Topology) is set to A, for the 4-port network embedding/de-embedding feature for selected channel (*Ch*).

For information on the connection type (Topology), refer to Connection Type.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port numbers).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Connection Type. <i>Ports(1)</i> :Port number assigned to port b in Connection Type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2
Resolution	1
Note	If the specified variable is out of the allowable setting range, an error occurs when executed. If you specify an identical port number to multiple ports, a runtime error occurs.

Examples

```
Dim EnbPort As Variant
SCPI.CALCulate(1).FSIMulator.ENBed.TYPE = "a"
SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.A.PORTs = Array(2,1)
EnbPort = SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.A.PORTs
```

E5071C

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Equivalent key

Analysis > Fixture Simulator > De-Embedding S4P > Topology > Ports > 1-2|1-3|1-4|2-1|2-3|2-4|3-1|3-2|3-4|4-1|4-2|4-3

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:A:PORTs <value 1>,<value 2>

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:A:PORTs?

Query response

{value 1},{value 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TOP:A:PORT 1,2"
20 OUTPUT 717;":CALC1:FSIM:EMB:TOP:A:PORT?"
30 ENTER 717;A,B
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.B.PORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.B.PORTs = *Ports**Ports* = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.B.PORTs

Description

This command sets/gets test port assignment when the connection type (Topology) is set to B, for the 4-port network embedding/de-embedding feature for selected channel (*Ch*).

For information on the connection type (Topology), refer to Connection Type.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 3-element array data (port numbers).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Connection Type. <i>Ports(1)</i> :Port number assigned to port b in Connection Type. <i>Ports(2)</i> :Port number assigned to port c in Connection Type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2 / <i>Ports(2)</i> :3
Resolution	1
Note	If the specified variable is out of the allowable setting range, an error occurs when executed. If you specify an identical port number to multiple ports, a runtime error occurs.

Examples

E5071C

```
Dim EnbPort As Variant
SCPI.CALCulate(1).FSIMulator.ENBed.TYPE = "b"
SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.B.PORTs = Array(1,3,2)
EnbPort = SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.B.PORTs
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Equivalent key

Analysis > Fixture Simulator > De-Embedding S4P > Topology > Ports >
1-2-3|1-2-4|1-3-2|1-3-4|1-4-2|1-4-3|2-1-3|2-1-4|2-3-1|2-3-4|2-4-1|2-4-3| 3-1-2|3-
1-4|3-2-1|3-2-4|3-4-1|3-4-2|4-1-2|4-1-3|4-2-1|4-2-3|4-3-1|4-3-2

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:B:PORTs <value  
1>,<value 2>,<value 3>  
:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:B:PORTs?
```

Query response

```
{value 1},{value 2},{value 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TOP:B:PORT 1,2,3"  
20 OUTPUT 717;":CALC1:FSIM:EMB:TOP:B:PORT?"  
30 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.C.PORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.C.PORTs = *Ports**Ports* = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.C.PORTs

Description

This command sets/gets test port assignment when the connection type (Topology) is set to C, for the 4-port network embedding/de-embedding feature for selected channel (*Ch*).

For information on the connection type (Topology), refer to Connection Type.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port numbers).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Port number assigned to port a in Connection Type . <i>Ports(1)</i> :Port number assigned to port b in Connection Type . <i>Ports(2)</i> :Port number assigned to port c in Connection Type . <i>Ports(3)</i> :Port number assigned to port d in Connection Type . <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Preset value	<i>Ports(0)</i> :1 / <i>Ports(1)</i> :2 / <i>Ports(2)</i> :3 / <i>Ports(3)</i> :4
Resolution	1
Note	If the specified variable is out of the allowable setting range, an error occurs when executed. If you specify an identical port number to multiple ports, a runtime error occurs.

E5071C

Examples

```
Dim EnbPort As Variant
SCPI.CALCulate(1).FSIMulator.ENBed.TYPE = "c"
SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.C.PORTs = Array(1,4,2,3)
EnbPort = SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.C.PORTs
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE
```

Equivalent key

Analysis > Fixture Simulator > De-Embedding S4P > Topology > Ports >
1-2-3-4|1-2-4-3|1-3-2-4|1-3-4-2|1-4-2-3|1-4-3-2|2-1-3-4|2-1-4-3|2-3-1-4|2-3-4-1| 2-
4-1-3|2-4-3-1|3-1-2-4|3-1-4-2|3-2-1-4|3-2-4-1|3-4-1-2|3-4-2-1|4-1-2-3|4-1-3-2| 4-2-
1-3|4-2-3-1|4-3-1-2|4-3-2-1

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:C:PORTs <value  
1>,<value 2>,<value 3>,<value 4>  
:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:C:PORTs?
```

Query response

```
{value 1},{value 2},{value 3},{value 4}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TOP:C:PORT 1,2,3,4"  
20 OUTPUT 717;":CALC1:FSIM:EMB:TOP:C:PORT?"  
30 ENTER 717;A,B,C,D
```


SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Object type

Property (**Read-Write**)

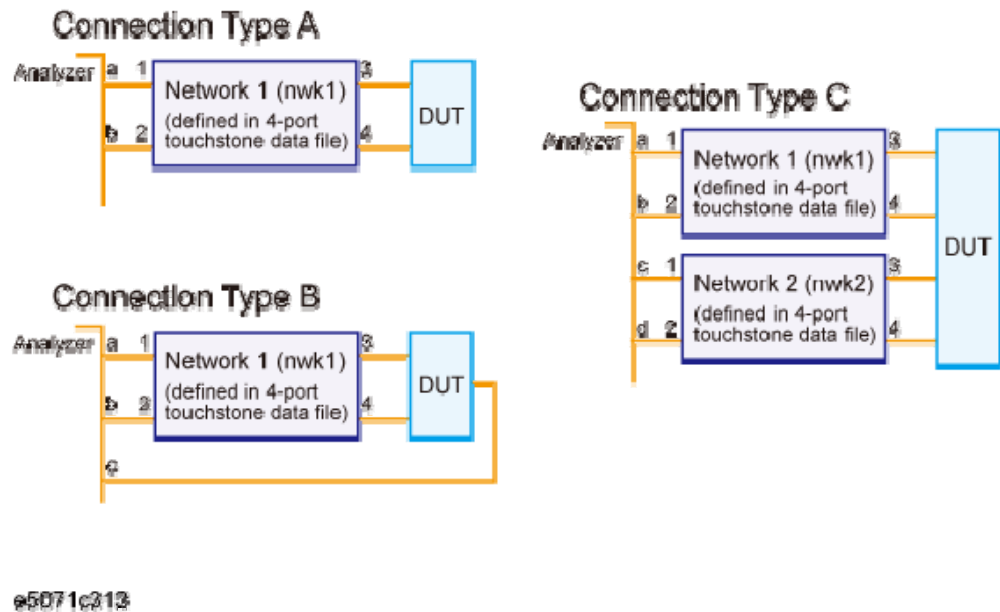
Syntax

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE = *File**File* = SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Description

This command selects a connection type (Topology), for the 4-port network embedding/de-embedding feature for channels 1 to 36 (*Ch*).

- This function is available with the Firmware revision 3.50 or greater.

Connection Type

Variable

Parameter	<i>Param</i>
Description	Connection type (refer to Connection Type)
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> "A": Specifies connection type A.

Preset value	<ul style="list-style-type: none">• "B": Specifies connection type B.• "C": Specifies connection type C.
	"A"

Examples

```
Dim EmbType As String
SCPI.CALCulate(1).FSIMulator.EMBed.TYPE = "b"
EmbType = SCPI.CALCulate(1).FSIMulator.EMBed.TYPE
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.A.PORTs
SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.B.PORTs
SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.C.PORTs
```

Equivalent key

Analysis > Fixture Simulator > De-Embedding S4P > Topology > Select Topology > A|B|C

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:EMBed:TYPE {A|B|C}
:CALCulate{[1]-160}:FSIMulator:EMBed:TYPE?
```

Query response

```
{A|B|C}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TYPE A"
20 OUTPUT 717;":CALC1:FSIM:EMB:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed.PORT(*Pt*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed.PORT(*Pt*).TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed.PORT(*Pt*).TYPE

Description

This command sets/gets the type of the network de-embedding, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Pt</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Param</i>
Description	Type of the network de-embedding
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "NONE": Specifies no network de-embedding. • "USER": Specifies the user-defined network de-embedding.
Preset value	"NONE"

Note

If you want to select the user-defined network de-embedding, you must specify the 2-port touchstone file in which the information on the user-defined network is saved in advance. If you do not specify the appropriate file and you select the user-defined network de-embedding, an error occurs when executed and NONE is automatically selected.

Examples

```
Dim DeemType As String
SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.PORT(1).USER.FILename = "network.s2p"
SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.PORT(1).TYPE = "user"
DeemType = SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.PORT(1).TYPE
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).USER.FILename
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe
SCPI.SENSE(Ch).CORRection.COLLEct.ECAL.ORIentation.STATe
```

Equivalent key

Analysis > Fixture Simulator > De-Embedding > Select Type

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:FSIMulator:SENDEd:DEEMbed:PORT{[1]|2|3|4}[:TYPE] {NONE|USER}
:CALCulate{[1]-160}:FSIMulator:SENDEd:DEEMbed:PORT{[1]|2|3|4}[:TYPE]?
```

Query response

```
{NONE|USER}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1 USER"
20 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1?"
30 ENTER 717;A$
```

**SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed.PORT(*Pt*).USER.FILenam
e**

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed.PORT(*Pt*).USER.FILen
ame = *File*

File =

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed.PORT(*Pt*).USER.FILen
ame

Description

This command specifies the file in which the information on the user-defined network for the network de-embedding function is saved for ports 1 and 4 (*Pt*) of selected channel (*Ch*). This file is saved as a 2-port touchstone file with the .s2p extension.

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Even if the specified file does not exist, no error occurs when you execute this object. However, when you set the type of the network de-embedding to the user-defined network with the SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.DEEMbed. PORT(*Pt*).TYPE object, an error occurs.

Variable

Parameter	<i>File</i>
Description	2-port touchstone file name (extension: .s2p) for the network de-embedding function
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim DeemUser As String
SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.PORT(1).USER.FILename = "D:\folder\network.s2p"
```

E5071C

```
DeemUser = SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.PORT(1).USER.FILename  
SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.PORT(1).TYPE = "user"
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).TYPE  
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe
```

Equivalent key

Analysis > Fixture Simulator > De-Embedding > User File

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDEd:DEEMbed:PORT{[1]|2|  
3|4}:USER:FILename <string>  
:CALCulate{[1]-160}:FSIMulator:SENDEd:DEEMbed:PORT{[1]|2|  
3|4}:USER:FILename?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1:USER:FIL ""D:\folder\Network.s2p""  
20 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1:USER:FIL?"  
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe = *Status**Status* = SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe

Description

This command turns ON/OFF the network de-embedding function when the fixture simulator function is ON, for all the ports of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the network de-embedding function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the network de-embedding function. • False or OFF: Turns OFF the network de-embedding function.
Preset value	False or 0

Examples

Dim Deemb As Boolean

SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.STATe = True

Deemb = SCPI.CALCulate(1).FSIMulator.SENDEd.DEEMbed.STATe

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > De-Embedding > De-Embedding

Equivalent SCPI command

Syntax

E5071C

:CALCulate{[1]-160}:FSIMulator:SENDEd:DEEMbed:STATe {ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:SENDEd:DEEMbed:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:STAT ON"  
20 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:STAT?"  
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).PARameters.C(*El*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).PARameters.C(*El*) = *Value*

Value =

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).PARameters.C(*El*)

Description

This command sets the C value of the matching circuit specified with the SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit. PORT(*Pt*).TYPE object, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

(*El*) shows the element number of C in the matching circuit. (*El*) is supported in revision A.11.20 and above.

Variable

Parameter	<i>Value</i>
Description	C value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	F (farad)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

E5071C

Dim PmcC As Double

SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.C(1) = 12E-12

PmcC = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.C(1)

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Equivalent key

Analysis > Fixture Simulator > Port Matching > C(C1)|C2

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}:PARameters:C{[1]|  
2} <numeric>  
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}:PARameters:C{[1]|  
2}? 
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:C1 12.3"  
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:C1?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G(EI)

Object type

Property (**Read-Write**)

Syntax

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G(EI) = Value
```

Value =

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G(EI)
```

Description

This command sets/gets the G value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit. PORT(Pt).TYPE object, for the ports 1 and 4 (Pt) of selected channel (Ch).

(EI) shows the element number of G in the matching circuit. (EI) is supported in revision A.11.20 and above.

Variable

Parameter	Value
Description	G value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	S (siemens)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

E5071C

Dim PmcG As Double

SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.G(1) = 12E-12

PmcG = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.G(1)

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.C
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Equivalent key

Analysis > Fixture Simulator > Port Matching > G(G1)|G2

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMCircuit:PORT{[1]|2|3|4}:PARameters:G{[1]|  
2} <numeric>  
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMCircuit:PORT{[1]|2|3|4}:PARameters:G{[1]|  
2}? 
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:G1 12.3"  
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:G1?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L(EI)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L(EI) = Value

Value =

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L(EI)

Description

This command sets the L value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit. PORT(Pt).TYPE object, for ports 1 and 4 (Pt) of selected channel (Ch).

(EI) shows the element number of L in the matching circuit. (EI) is supported in revision A.11.20 and above.

Variable

Parameter	Value
Description	L value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	H (henry)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

E5071C

Dim PmcL As Double

SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.L(1) = 12E-12

PmcL = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.L(1)

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.C
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **L(L1)|L2**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMCircuit:PORT{[1]|2|3|4}:PARameters:L{[1]|2  
} <numeric>  
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMCircuit:PORT{[1]|2|3|4}:PARameters:L{[1]|2  
}? 
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:L1 12.3"  
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:L1?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R(EI)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R(EI) = *Value*

Value = SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(EI).PARameters.R(Pt)

Description

This command sets/gets the R value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit. PORT(Pt).TYPE object, for ports 1 and 4 (Pt) of selected channel (Ch).

(EI) shows the element number of R in the matching circuit. (EI) is supported in revision A.11.20 and above.

Variable

Parameter	<i>Value</i>
Description	R value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	ohm
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

E5071C

Dim PmcR As Double

SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.R(1) = 12E-12

PmcR = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).PARameters.R(1)

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.C
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L
(EI)

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **R(R1)|R2**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}:PARameters:R{[1]|  
2} <numeric>  
:CALCulate{[1]-  
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}:PARameters:R{[1]|  
2}? 
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:R1 12.3"  
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:R1?"  
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).TYPE = *Param*
Param = SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).TYPE

Description

This command selects the type of the matching circuit, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Type of the matching circuit
Data type	Character string type (String)
Range	<p>Select from the following.</p> <ul style="list-style-type: none"> • "NONE": Specifies no-circuit. • "SLPC": Specifies the circuit that consists of series L and shunt C. • "PCSL": Specifies the circuit that consists of shunt C and series L. • "PLSC": Specifies the circuit that consists of shunt L and series C. • "SCPL": Specifies the circuit that consists of series C and shunt L. • "PLPC": Specifies the circuit that consists of shunt L and shunt C. • "SCPC": Specifies the circuit that consists of series C and shunt C. • "PCSC": Specifies the circuit that consists of shunt C and series C. • "SLPL": Specifies the circuit that consists of series L and shunt L. • "PLSL": Specifies the circuit that consists of shunt L and series L.

	<ul style="list-style-type: none">• "USER": Specifies the user-defined circuit. <div>NOTE "SCPC", "PCSC", "SLPL" and "PLSL" are available at revision 11.20 and above.</div>
Preset value	"NONE"
Note	If you want to select the user-defined circuit, you must specify the 2-port touchstone file in which the proper information on the user-defined circuit is saved in advance. If you do not specify the appropriate file and you select the user-defined circuit, an error occurs when executed and NONE is automatically selected.

Examples

```
Dim CirType As String
SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).TYPE = "slpc"
CirType = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).TYPE
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.C
(EI)
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G
(EI)
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L
(EI)
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R
(EI)
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).USER.FILena
me
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe
```

Equivalent key

Analysis > Fixture Simulator > Port Matching > Select Circuit

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}{:TYPE} {NONE|
SLPC|PCSL|PLSC|SCPL|PLPC|SCPC|PCSC|SLPL|PLSL|USER}
:CALCulate{[1]-
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}{:TYPE}?
```

Query response

{NONE|SLPC|PCSL|PLSC|SCPL|PLPC|SCPC|PCSC|SLPL|PLSL|USER}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1 SLPC"  
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1?"  
30 ENTER 717;A$
```

**SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).USER.FILenam
e**

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).USER.FILena
me = *File*

File =

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.PORT(*Pt*).USER.FILena
me

Description

This command specifies the file in which the information on the user-defined matching circuit is saved (2-port touchstone file), for the port 1 and 4 (*Pt*) of selected channel (*Ch*).

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Even if the specified file does not exist, no error occurs when you execute this object. However, when you set the type of the matching circuit to the user-defined circuit with the SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit. PORT(*Pt*).TYPE object, an error occurs.

Variable

Parameter	<i>File</i>
Description	2-port touchstone file name (extension: .s2p) for the matching circuit
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim PmcUser As String
SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).USER.FILename = "match.s2p"
PmcUser = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).USER.FILename
SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.PORT(1).TYPE = "user"
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Equivalent key

Analysis > Fixture Simulator > Port Matching > User File

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}:USER:FILEname
<string>

:CALCulate{[1]-
160}:FSIMulator:SENDEd:PMcircuit:PORT{[1]|2|3|4}:USER:FILEname?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:USER:FIL ""Match.s2p""
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:USER:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe = *Status**Status* = SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATe

Description

This command turns ON/OFF the matching circuit embedding function when the fixture simulator function is ON, for all the ports of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the matching circuit embedding function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the matching circuit embedding function. • False or OFF: Turns OFF the matching circuit embedding function.
Preset value	False or OFF

Examples

Dim Pmcir As Boolean

SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.STATe = True

Pmcir = SCPI.CALCulate(1).FSIMulator.SENDEd.PMCircuit.STATe

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).USER.FILena
me

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Port Matching > Port Matching

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDEd:PMcircuit:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:SENDEd:PMcircuit:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).IMAGinary

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).IMAGinary
= *Value*

Value =

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).IMAGinary

Description

This command sets/gets the impedance value (imaginary part) for the port impedance conversion function, for the ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Impedance value (imaginary part) for the port impedance conversion function
Data type	Double precision floating point type (Double)
Range	-1E+18 to 1E+18
Preset value	0
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ZImag As Double
```

```
SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.PORT(1).IMAGinary = -9.2E10
```

```
ZImag = SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.PORT(1).IMAGinary
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).REAL

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe
 SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Port ZConversion > Port1 Z0 Imag|Port2 Z0 Imag|Port3 Z0 Imag|Port4 Z0 Imag

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion
:PORT{[1]|2|3|4}:IMAGinary <numeric>
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion
:PORT{[1]|2|3|4}:IMAGinary?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:IMAG 75"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:IMAG?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).REAL

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).REAL =
Value

Value =

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).REAL

Description

This command sets/gets the impedance value (real part) for the port impedance conversion function, for ports 1 and 4 (*Pt*) of the selected channel (*Ch*).

- This command performs in the same way as "SCPI.CALCulate(Ch).FSIM-ulator.SENDEd.ZCONversion.PORT(Pt).Z0.R"

Variable

Parameter	<i>Value</i>
Description	Impedance value (real part) for the port impedance conversion function
Data type	Double precision floating point type (Double)
Range	0.001 to 1E7
Preset value	50
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ZReal As Double
SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.PORT(1).REAL = 3.7E5
ZReal = SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.PORT(1).REAL
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).IMAGinary
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > Fixture Simulator > Port ZConversion > Port1 Z0 Real|Port2 Z0 Real|Port3 Z0 Real|Port4 Z0 Real

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion
:PORT{[1]|2|3|4}:REAL <numeric>
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion
:PORT{[1]|2|3|4}:REAL?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:REAL 75"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:REAL?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R =
Value

Value =

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R

Description

This command sets/gets the impedance value for the port impedance conversion function, for ports 1 and 4 (*Pt*) of the selected channel (*Ch*).

- This command clears setting value of
 SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.
 PORT(Pt).IMAGinary

Variable

Parameter	<i>Value</i>
Description	Impedance value for the port impedance conversion function
Data type	Double precision floating point type (Double)
Range	0.001 to 1E7
Preset value	50
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ZconR As Double
SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.PORT(1).Z0.R = 75
ZconR = SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.PORT(1).Z0.R
```

Related objects

```
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).IMAGinary
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).REAL
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe
SCPI.CALCulate(Ch).FSIMulator.STATe
```

Equivalent key

Analysis > Fixture Simulator > Port ZConversion > Port1 Z0 Real|Port2 Z0 Real|Port3 Z0 Real|Port4 Z0 Real

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion
:PORT{[1]|2|3|4}:Z0[:R] <numeric>
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion
:PORT{[1]|2|3|4}:Z0[:R]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:Z0 75"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:Z0?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe = *Status*
Status = SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe

Description

This command turns ON/OFF the port impedance conversion function when the fixture simulator function is ON, for all the ports of the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the port impedance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the port impedance conversion function. • False or OFF: Turns OFF the port impedance conversion function.
Preset value	False or OFF

Examples

```
Dim Zcon As Boolean
SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.STATe = True
Zcon = SCPI.CALCulate(1).FSIMulator.SENDEd.ZCONversion.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).IMAGinary
 SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).REAL
 SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R
 SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Port ZConversion > Port ZConversion

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion:STATe
{ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:SENDEd:ZCONversion:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).FSIMulator.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.CALCulate(Ch).FSIMulator.STATe = *Status**Status* = SCPI.CALCulate(Ch).FSIMulator.STATe**Description**

This command turns ON/OFF the fixture simulator function of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the fixture simulator function
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the fixture simulator function. • False or OFF: Turns OFF the fixture simulator function.
Preset value	False or OFF

Examples

```
Dim FxtSim As Boolean
SCPI.CALCulate(1).FSIMulator.STATe = True
FxtSim = SCPI.CALCulate(1).FSIMulator.STATe
```

Equivalent key**Analysis > Fixture Simulator > Fixture Simulator****Equivalent SCPI command****Syntax**

```
:CALCulate{[1]-160}:FSIMulator:STATe {ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:STATe?
```

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:STAT ON"  
20 OUTPUT 717;":CALC1:FSIM:STAT?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARAmeter.TNAME.DEFine(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARAmeter.TNAME.DEFine(*Name*) = *Param**Param* = SCPI.CALCulate(*Ch*).PARAmeter.TNAME.DEFine(*Name*)

Description

This command sets/gets the measurement parameter of the selected trace for the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	<p>Select either one of the following:</p> <ul style="list-style-type: none"> • "S<XY>" Where: x=1to 4 Y=1to 4 • A B C D <ul style="list-style-type: none"> • R<X> (X=1-4) • AUX1 or AUX2

Preset value	"S11"
---------------------	-------

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARAmeter(1).TNAME.DATA = "duptrace"
SCPI.CALCulate(1).PARAmeter.TNAME.DEFine("duptrace") = "S21"
MeasPara = SCPI.CALCulate(1).PARAmeter.TNAME.DEFine("duptrace")
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).TNAME.DATA
SCPI.CALCulate(Ch).PARAmeter.TNAME.SELect
SCPI.CALCulate(Ch).PARAmeter.TNAME.SPORT
```

Equivalent key

```
Meas > S<XY> {X=1-4;Y=1-4}|
Meas > Absolute > R<X> {X=1-4}|A|B|C|D|
Meas > Aux Input 1|Aux Input 2
```

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:PARAmeter:TNAME:DEFine <string>,
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S4
4|A|B|C|D|R
1|R2|R3|R4|Aux1|AUX2}
:CALCulate{[1]-160}:PARAmeter:TNAME:DEFine? <string>
```

Query response

```
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S4
4|A|B|C|D|R1|R2|R3|R4|AUX1|AUX2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPTRACE""
20 OUTPUT 717;":CALC1:PAR:TNAME:DEF ""DUPTRACE"", S21"
30 OUTPUT 717;":CALC1:PAR:TNAME:DEF? ""DUPTRACE""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).PARAmeter.TNAME.SELect

Object type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).PARAmeter.TNAME.SELect = *Name*

Description

This command sets the measurement parameter of the selected trace for the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARAmeter(Tr).TNAME.DATA command.

Variable

Parameter	<i>Name</i>
Description	Name of the trace
Data type	Character string type (String)

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARAmeter.TNAME.SELect = "duptrace"
```

Equivalent key

Trace Prev / Trace Next

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARAmeter:TNAME:SELect <string>

Example of use

```
10 OUTPUT 717;":CALC1:PAR:TNAME:SEL ""DUPTRACE""
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARAmeter.TNAME.SPORT(*Name*)

Type of object

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARAmeter.TNAME.SPORT(*Name*) = *Value**Value* = SCPI.CALCulate(*Ch*).PARAmeter.TNAME.SPORT(*Name*)

Description

This command sets/gets the output port used for absolute or AUX measurements, for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace Name
Data type	Character string type (String)

Parameter	<i>Value</i>
Description	Setting of the output port
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	You need to set the measurement parameter for absolute/AUX measurements with the "SCPI.CALCulate(<i>Ch</i>).PARAmeter(<i>Tr</i>).DEFine" command.

Example of use

```
Dim Sport As Long
SCPI.CALCulate(1).PARAmeter(1).TNAME.DATA = "duptrace"
```

E5071C

```
SCPI.CALCulate(1).PARAmeter.TNAME.DEFine("duptrace") = "B"  
SCPI.CALCulate(1).PARAmeter.TNAME.SPORT("duptrace") = 4  
Sport = SCPI.CALCulate(1).PARAmeter.TNAME.SPORT("duptrace")
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).DEFine
```

Equivalent key

Meas > **Absolute** > **A(x) ...D(x)...R1(x) ...R4(x)** (x: 1 to 4)

Meas > **AUX Input 1** or **AUX Input 2** > **Sweep Port**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:PARAmeter:TNAME:SPORT <string>,<numeric>  
:CALCulate{[1]-160}:PARAmeter:TNAME:SPORT? <string>
```

Query response

```
{value}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR2:TNAME:DATA ""DUPTRACE""  
20 OUTPUT 717;":CALC1:PAR:TNAME:DEF ""DUPTRACE"", B"  
30 OUTPUT 717;":CALC1:PAR:TNAME:SPOR ""DUPTRACE"", 4"  
40 OUTPUT 717;":CALC1:PAR:TNAME:SPOR? ""DUPTRACE""  
50 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).DEFine = *Param**Param* = SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).DEFine

Description

This command sets/gets the measurement parameter of the selected trace (*Tr*), for the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select either one of the following: <ul style="list-style-type: none"> • "S<XY>" Where: x=1to 4 Y=1to 4 • A B C D <ul style="list-style-type: none"> • R<X> (X=1-4) • AUX1 or AUX2
Preset value	"S11"

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARAmeter(1).DEFine = "s21"
MeasPara = SCPI.CALCulate(1).PARAmeter(1).DEFine
```

Equivalent key

Meas > **S<XY>** {X=1-4;Y=1-4}|**Meas** > **Absolute** > **R<X>** {X=1-4}|**A|B|C|D|****Meas** > **Aux Input 1|Aux Input 2**

Equivalent SCPI command

Syntax

E5071C

```
:CALCulate{[1]-160}:PARameter{[1]-16}:DEFine  
<string>{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S2  
4|S34|S44|A|B|C|D|R  
1|R2|R3|R4|Aux1|AUX2}  
:CALCulate{[1]-160}:PARameter{[1]-16}:DEFine?
```

Query response

```
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S4  
4|A|B|C|D|R1|R2|R3|R4|AUX1|AUX2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:DEF S21"  
20 OUTPUT 717;":CALC1:PAR1:DEF?"  
30 ENTER 717;A$
```


SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect**Object type**Method (**Write-only**)**Syntax**SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect**Description**

This command sets/gets the selected trace (*Tr*) of selected channel (*Ch*) to the active trace.

You can set only a trace displayed to the active trace. If this object is used to set a trace not displayed to the active trace, an error occurs when executed and the object is ignored. (No read).

In revision A.9.60 and above, you can select the trace by the trace name, provided it has been defined earlier. To define a trace name, execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

In revision A.9.60 and above, you can execute the action directly by using direct channel and trace assign commands like SCPI.CALCulate(*Ch*).TRACE(*Tr*).XXXX.XXXX.

Variable

Parameter	<i>Tr</i>
Description	Trace number
Data type	Long integer type (Long)
Range	1 to 16
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

SCPI.CALCulate(2).PARAmeter(2).SElect

Related objectsSCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.MEDium

E5071C

SCPI.CALCulate(Ch).SElected.CORRection.EDElay.TIME

SCPI.CALCulate(Ch).SElected.CORRection.EDElay.WGCutoff

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.DCOffset

[Equivalent key](#)

[Trace Prev](#) / [Trace Next](#)

[Equivalent SCPI command](#)

Syntax

:CALCulate{[1]-160}:PARameter{[1]-16}:SElect

Example of use

10 OUTPUT 717;":CALC1:PAR1:SEL"

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA**Object type**Property (**Read-Write**)**Syntax**

This option is only available In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA = *Name*

Name = SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).TNAME.DATA

Description

This command defines/gets the name for the active trace of selected channel (*Ch*).

This command is only available in revision A.9.60 and above.

Variable

Parameter	<i>Name</i>
Description	Trace name for the active trace
Data type	Character string type (String)

Examples

```
Dim TraceName As String
```

```
SCPI.CALCulate(1).PARAmeter(1).TNAME.DATA = "DuplexTrace"
```

```
TraceName = SCPI.CALCulate(1).PARAmeter(Tr).TNAME.DATA = Param
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}:PARAmeter{[1]-16}:TNAME:DATA <string>
```

```
:CALCulate{[1]-160}:PARAmeter{[1]-16}:TNAME:DATA?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAME:DATA ""DUPLXTRACE""
```

```
20 OUTPUT 717;":CALC1:PAR1:TNAME:DATA?"
```

```
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SPORT

Type of object

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SPORT = *Value**Value* = SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SPORT

Description

This command sets/gets the output port used for absolute or AUX measurements, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of the output port
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	You need to set the measurement parameter for absolute/AUX measurements with the "SCPI.CALCulate(<i>Ch</i>).PARAmeter(<i>Tr</i>).DEFine" command.

Example of use

```
Dim Sport As Long
SCPI.CALCulate(1).PARAmeter(1).DEFine = "B"
SCPI.CALCulate(1).PARAmeter(1).SPORT = 4
Sport = SCPI.CALCulate(1).PARAmeter(1).SPORT
```

Related objects

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).DEFine

Equivalent key

Meas > Absolute > A(x) ...D(x)...R1(x) ...R4(x) (x: 1 to 4)**Meas > AUX Input 1 or AUX Input 2 > Sweep Port**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARAmeter{[1]-16}:SPORT <numeric>

:CALCulate{[1]-160}:PARameter{[1]-16}:SPORT?

Query response

{value}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:DEF B"  
20 OUTPUT 717;":CALC1:PAR1:SPOR 4"  
30 OUTPUT 717;":CALC1:PAR1:SPOR?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARAmeter.COUNT**Object type**Property (**Read-Write**)**Syntax**SCPI.CALCulate(*Ch*).PARAmeter.COUNT = *Value**Value* = SCPI.CALCulate(*Ch*).PARAmeter.COUNT**Description**This command sets/gets the number of traces of selected channel (*Ch*).**Variable**

Parameter	<i>Value</i>
Description	Number of traces
Data type	Long integer type (Long)
Range	Varies depending on the upper limit setting for the channel/trace number.
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TraceNum As Long
SCPI.CALCulate(1).PARAmeter.COUNT = 4
TraceNum = SCPI.CALCulate(1).PARAmeter.COUNT
```

Equivalent key**Display > Num of Traces****Equivalent SCPI command****Syntax**

:CALCulate{[1]-160}:PARAmeter:COUNT <numeric>

:CALCulate{[1]-160}:PARAmeter:COUNT?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR:COUN 4"  
20 OUTPUT 717;":CALC1:PAR:COUN?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).TRACe(Tr).XXXX.XXXX

As mentioned in Selecting the Active Channel/Trace, if you are using E5071C revision A.9.60 and above, you can select the trace directly by using **TRAC{1-16}** for all **SElected** commands. As such, you do not need to make a trace active before assigning a command to it.

The following commands allow you to select the trace directly by using **TRAC{1-16}**. Note that the active trace is not changed by this command:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DB
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.MARKer
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.VALue
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.FAIL
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MAXimum
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MINimum
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.REPort.DATA
 SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.STATe
 SCPI.CALCulate(Ch).TRACe(Tr).CONVersion.FUNCTion
 SCPI.CALCulate(Ch).TRACe(Tr).CONVersion.STATe
 SCPI.CALCulate(Ch).TRACe(Tr).CORRection.EDElay.MEDium
 SCPI.CALCulate(Ch).TRACe(Tr).CORRection.EDElay.TIME
 SCPI.CALCulate(Ch).TRACe(Tr).CORRection.EDElay.WGCutoff
 SCPI.CALCulate(Ch).TRACe(Tr).CORRection.OFFSet.PHASE
 SCPI.CALCulate(Ch).TRACe(Tr).DATA.FDATA
 SCPI.CALCulate(Ch).TRACe(Tr).DATA.FMEMory
 SCPI.CALCulate(Ch).TRACe(Tr).DATA.SDATA
 SCPI.CALCulate(Ch).TRACe(Tr).DATA.SMEMory
 SCPI.CALCulate(Ch).TRACe(Tr).DATA.XAXis
 SCPI.CALCulate(Ch).TRACe(Tr).EQUation.STATE
 SCPI.CALCulate(Ch).TRACe(Tr).EQUation.TEXT
 SCPI.CALCulate(Ch).TRACe(Tr).EQUation.VALid
 SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. CENTER
 SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. SHAPE
 SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. SPAN

SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. START
 SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. STATE
 SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. STOP
 SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. TYPE
 SCPI.CALCulate(Ch).TRACe(Tr).FORMat
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.DATA
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.DOMain.COUPle
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.DOMain.START
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.DOMain.STATE
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.DOMain.STOP
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.EXECute
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.PEXCursion
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.POINTs
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.PPOLarity
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.TARGET
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.TTRansition
 SCPI.CALCulate(Ch).TRACe(Tr).FUNCTion.TYPE
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.DATA
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.DISPlay.STATE
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.FAIL
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.AMPLitude
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.MARKer
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.STIMulus
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.ALL
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.DATA
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.POINTs
 SCPI.CALCulate(Ch).TRACe(Tr).LIMit.STATE
 SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).ACTivate
 SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).BWIDth.DATA
 SCPI.CALCulate(Ch).TRACe(Tr).MARKer.BWIDth.STATE
 SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).BWIDth.THReshold
 SCPI.CALCulate(Ch).TRACe(Tr).MARKer.COUPle
 SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).DATA

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).DISCcrete
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNcTion.DOMain.COUPle
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.DOMain.MULTiple.R
ANGe
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.DOMain.MULTiple.S
TARt
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.DOMain.MULTiple.S
TATe
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.DOMain.MULTiple.S
TOP
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNcTion.DOMain.STARt
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNcTion.DOMain.STATe
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNcTion.DOMain.STOP
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.EXECute
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.PEXCursion
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.PPOLarity
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.TARGet
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.TRACking
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.TTRansition
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNcTion.TYPE
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.NOTCh
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).NOTCh.DATA
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).NOTCh.THReshold
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).SET
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).STATe
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).X
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).Y
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.REFerence.STATe
SCPI.CALCulate(Ch).TRACe(Tr).MATH.FUNcTion
SCPI.CALCulate(Ch).TRACe(Tr).MATH.MEMorize
SCPI.CALCulate(Ch).TRACe(Tr).MIXer.XAXis
SCPI.CALCulate(Ch).TRACe(Tr).MSTatistics.DATA
SCPI.CALCulate(Ch).TRACe(Tr).MSTatistics.STATe
SCPI.CALCulate(Ch).TRACe(Tr).OFFset.XAXis

SCPI.CALCulate(Ch).SElected.PLIMit.DATA
SCPI.CALCulate(Ch).SElected.PLIMit.DISPlay.LINE
SCPI.CALCulate(Ch).SElected.PLIMit.FAIL
SCPI.CALCulate(Ch).SElected.PLIMit.REPort.DATA
SCPI.CALCulate(Ch).SElected.PLIMit.STATe
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DATA
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.LINE
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.SELect
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.VALue
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.FAIL
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.REPort.DATA
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.STATe
SCPI.CALCulate(Ch).TRACe(Tr).SMOothing.APERture
SCPI.CALCulate(Ch).TRACe(Tr).SMOothing.STATe
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.CENTer
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.IMPulse.WIDTH
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.KBESsel
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.LPFRequency
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.SPAN
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.START
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.STATe
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.STEP.RTIME
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.STIMulus
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.STOP
SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.TYPE

SCPI.CALCulate(*Ch*).SElected.BLIMit.DB

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.BLIMit.DB = ValueValue = SCPI.CALCulate(*Ch*).SElected.BLIMit.DB

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.DB = ValueValue = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.DB

Description

This command sets/gets the bandwidth threshold value (attenuation from the peak) of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect command).

In revision A.9.60 and above, you can select the trace and set/get the bandwidth threshold value for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Bandwidth N dB points.
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	0
Unit	dB

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimDB As Double
SCPI.CALCulate(1).SElected.BLIMit.DB = 3
BLimDB = SCPI.CALCulate(1).SElected.BLIMit.DB
```

Related objects

```
SCPI.CALCulate(Ch).PARameter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
```

Equivalent key

Analysis > Bandwidth Limit > N dB Points

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:BLIMit:DB <numeric>
:CALCulate{[1]-160}{:SElected}:BLIMit:DB?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DB <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DB?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DB 3"
20 OUTPUT 717;":CALC1:BLIM:DB?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.MARKer

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.MARKer = Status

Status = SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.MARKer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.MARKer = Status

Status = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.MARKer

Description

This command turns ON/OFF the marker display of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the marker display of the bandwidth tests for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the bandwidth marker.
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the bandwidth marker. • False or OFF: Turns OFF the bandwidth marker.
Preset value	False or OFF

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimMk As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.BLIMit.DISPlay.MARKer = True
BLimMk = SCPI.CALCulate(1).SElected.BLIMit.DISPlay.MARKer
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.VALue
```

Equivalent key

Analysis > **Bandwidth Limit** > **BW Marker**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:BLIMit:DISPlay:MARKer {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:BLIMit:DISPlay:MARKer?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:MARKer
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:MARKer?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DISP:MARK ON"
20 OUTPUT 717;":CALC1:BLIM:DISP:MARK?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.VALue

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.VALue = Status

Status = SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.VALue

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.VALue = Status

Value = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.VALue

Description

This command turns ON/OFF the bandwidth value display of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the bandwidth value display of the bandwidth tests for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the bandwidth display of the bandwidth test.
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the bandwidth display. • False or OFF: Turns OFF the bandwidth display.
Preset value	False or OFF

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimVal As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElectSCPI.CALCulate(1).SElected.BLIMit.DISPlay.VALue = True
BLimVal = SCPI.CALCulate(1).SElected.BLIMit.DISPlay.VALue
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.MARKer
```

Equivalent key

Analysis > **Bandwidth Limit** > **BW Display**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:BLIMit:DISPlay:VALue {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:BLIMit:DISPlay:VALue?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:VALue
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:VALue?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DISP:VAL ON"
20 OUTPUT 717;":CALC1:BLIM:DISP:VAL?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.BLIMit.FAIL**Object type**Property (**Read Only**)**Syntax**

All revision:

Status = SCPI.CALCulate(Ch).SElected.BLIMit.FAIL

In revision A.9.60 and above, the following syntax is also supported:

Status = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.FAIL

Description

This command get the bandwidth limit test results, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and get the bandwidth limit test results for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	Status
Description	The bandwidth limit test result
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the bandwidth limit test result is FAIL. • False or OFF: Turns OFF the bandwidth limit test result is PASS.
Note	When the bandwidth limit test if set to OFF, False or OFF is always read out.

For information on the variable (*Ch*), see Variable.

Examples

```
Dim Result As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.BLIMit.STATe = True
Result = SCPI.CALCulate(1).SElected.BLIMit.FAIL
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{[:SElected]:BLIMit:FAIL?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:FAIL?
```

Query response

```
{1|0}<newline><^END>
```

	Description
1	The bandwidth test result is FAIL.
0	The bandwidth test result is PASS.

When the bandwidth limit test is set to OFF, 0 is always read out.

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.BLIMit.MAXimum

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.BLIMit.MAXimum = Value

Value = SCPI.CALCulate(Ch).SElected.BLIMit.MAXimum

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MAXimum = Value

Value = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MAXimum

Description

This command sets/gets the upper limit value of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and set/get the upper limit value of the bandwidth test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	Maximum bandwidth
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	3E5
Unit	Hz (hertz), dBm or second

- For the variable (*Ch*), the channel number can be selected from 1 to 16, not 1 to 36.

Examples

```
Dim BLimMax As Double
SCPI.CALCulate(1).SElected.BLIMit.MAXimum = 1E9
BLimMax = SCPI.CALCulate(1).SElected.BLIMit.MAXimum
```

Related objects

```
SCPI.CALCulate(Ch).PARameter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
SCPI.CALCulate(Ch).SElected.BLIMit.MINimum
```

Equivalent key

Analysis > **Bandwidth Limit** > **Max Bandwidth**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:BLIMit:MAXimum <numeric>
:CALCulate{[1]-160}{:SElected}:BLIMit:MAXimum?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MAXimum <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MAXimum?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:MAX 3E5"
20 OUTPUT 717;":CALC1:BLIM:MAX?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.BLIMit.MINimum

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.BLIMit.MINimum = Value

Value = SCPI.CALCulate(Ch).SElected.BLIMit.MINimum

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MINimum = Value

Value = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MINimum

Description

This command sets/gets the lower limit value of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and set/get the lower limit value of the bandwidth test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	Minimum bandwidth
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	1E4
Unit	Hz (hertz), dBm or second

For information on the variable (Ch), see Variable.

Examples

```
Dim BLimMin As Double
SCPI.CALCulate(1).SElected.BLIMit.MINimum = 1E6
BLimMin = SCPI.CALCulate(1).SElected.BLIMit.MINimum
```

Related objects

```
SCPI.CALCulate(Ch).PARameter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
SCPI.CALCulate(Ch).SElected.BLIMit.MAXimum
```

Equivalent key

Analysis > Bandwidth Limit > Min Bandwidth

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:BLIMit:MINimum <numeric>
:CALCulate{[1]-160}{:SElected}:BLIMit:MINimum?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MINimum <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MINimum?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:MIN 1E4"
20 OUTPUT 717;":CALC1:BLIM:MIN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.BLIMit.REPort.DATA**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.BLIMit.REPort.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.REPort.DATA**Description**

This command reads the bandwidth value of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect command).

In revision A.9.60 and above, you can select the trace and read the bandwidth test just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	The bandwidth value of the bandwidth
Data type	Double precision floating point type (Double)

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BWData As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
BWData = SCPI.CALCulate(1).SElected.BLIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect
 SCPI.CALCulate(*Ch*).SElected.BLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

:CALCulate{[1]-160}{:SElected}:BLIMit:REPort[:DATA]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:REPort[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:REP?"
20 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.BLIMit.STATe**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.BLIMit.STATe = Status

Status = SCPI.CALCulate(Ch).SElected.BLIMit.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.STATe = Value

Value = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.STATe

Description

This command turns ON/OFF the bandwidth test function, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the bandwidth test function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF the bandwidth test function.
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the bandwidth test function. • False or OFF: Turns OFF the bandwidth test function.
Preset value	False or OFF

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimTest As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.BLIMit.STATe = True
BLimTest = SCPI.CALCulate(1).SElected.BLIMit.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.BLIMit.DB
SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.MARKer
SCPI.CALCulate(Ch).SElected.BLIMit.DISPlay.VAlue
SCPI.CALCulate(Ch).SElected.BLIMit.FAIL
SCPI.CALCulate(Ch).SElected.BLIMit.MAXimum
SCPI.CALCulate(Ch).SElected.BLIMit.MINimum
SCPI.CALCulate(Ch).SElected.BLIMit.REPort.DATA
```

Equivalent key

Analysis > Bandwidth Limit > BW Test

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}:SElected:BLIMit[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}:SElected:BLIMit[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:BLIM ON"
20 OUTPUT 717;":CALC1:BLIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.CONVersion.FUNction

Object type

Property (**Read-Write**)

Syntax

All Revision

SCPI.CALCulate(*Ch*).SElected.CONVersion.FUNction = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.CONVersion.FUNction

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.FUNction = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.FUNction

Description

This command sets/gets the parameter after conversion using the parameter conversion function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the parameter after conversion using the parameter conversion function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	The parameter after conversion
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "ZREFlection": Specifies the equivalent impedance in reflection measurement. • "ZTRansmit": Specifies the equivalent impedance(series) in transmission measurement. • "YREFlection": Specifies the equivalent admittance in reflection measurement. • "YTRansmit": Specifies the equivalent admittance(series) in transmission measurement.

	<ul style="list-style-type: none"> • "INVersion": Specifies the inverse S-parameter. • "ZTSHunt": Specifies the equivalent impedance(shunt) in transmission measurement. • "YTSHunt": Specifies the equivalent admittance(shunt) in transmission measurement. • "CONJugation": Specifies the conjugate.
Preset value	"ZREFlection"

Examples

```
Dim Func As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.CONVersion.FUNcTion = "ztr"
Func = SCPI.CALCulate(1).SElected.CONVersion.FUNcTion
```

Related objects

```
SCPI.CALCulate(Ch).SElected.CONVersion.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.DCOFFset
```

Equivalent key

Analysis > Conversion >
Z:Reflection|Z:Transmission|Y:Reflection|Y:Transmission|1/S| Z:Trans-Shunt|Y:Trans-Shunt|Conjugation

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:CONVersion:FUNcTion {ZREFlection|
ZTRansmit|YREFlection|YTRansmit|INVersion|ZTSHunt|YTSHunt|CONJugati
on}
:CALCulate{[1]-160}{:SElected}:CONVersion:FUNcTion?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion:FUNcTion {ZREFlection|
ZTRansmit|YREFlection|YTRansmit|INVersion|ZTSHunt|YTSHunt|CONJugati
on}
:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion:FUNcTion?
```

Query response

E5071C

{ZREF|ZTR|YREF|YTR|INV|ZTSH|YTSH|CONJ}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CONV:FUNC ZTR"  
20 OUTPUT 717;":CALC1:CONV:FUNC?"  
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.CONVersion.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.CONVersion.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.CONVersion.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.STATe

Description

This command turns ON/OFF the parameter conversion function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the parameter conversion function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the parameter conversion function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the parameter conversion function. • False or OFF: Turns OFF the parameter conversion function.
Preset value	False or OFF

Examples

E5071C

```
Dim Conv As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.CONVersion.STATe = True
Conv = SCPI.CALCulate(1).SElected.CONVersion.STATe
```

Related objects

```
SCPI.CALCulate(Ch).SElected.CONVersion.FUNction
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > **Conversion** > **Conversion**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:CONVersion[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:CONVersion[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:CONV ON"
20 OUTPUT 717;":CALC1:CONV?"
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.MEDium

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.MEDium = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.MEDium

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.MEDium = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.MEDium

Description

This command select the media type, for calculating the electrical delay time of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the media type for calculating the electrical delay time just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Select the media type for calculating the electrical delay time.
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "COAXial": Selects coaxial as a media type. • "WAVEguide": Selects waveguide as a media type.
Preset value	"COAXial"

Examples

E5071C

```
Dim EdelMed As String
SCPI.CALCulate(1).SElected.CORRection.EDElay.MEDium = "WAVeguide"
EdelMed = SCPI.CALCulate(1).SElected.CORRection.EDElay.MEDium
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.CORRection.EDElay.TIME
SCPI.CALCulate(Ch).SElected.CORRection.EDElay.WGCutoff
```

Equivalent key

Scale > Electrical Delay > Media

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:CORRection:EDElay:MEDium
{COAXial|WAVeguide}
:CALCulate{[1]-160}{:SElected}:CORRection:EDElay:MEDium?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDElay:MEDium
{COAXial|WAVeguide}
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDElay:MEDium?
```

Query response

```
{COAXial|WAVeguide}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:MED WAV"
20 OUTPUT 717;":CALC1:CORR:EDEL:MED?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.CORRection.EDELay.TIME

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.TIME = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.TIME

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.TIME = *Value**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.MEDium

Description

This command sets/gets the electrical delay time of the active trace of channels 1 to 160 (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the electrical delay time for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Electrical delay time
Data type	Double precision floating point type (Double)
Range	-10 to 10
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

upper limit of the range is exceeded) is set.

Examples

```
Dim Edel As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.CORRection.EDElay.TIME = 0.2
Edel = SCPI.CALCulate(1).SElected.CORRection.EDElay.TIME
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.CORRection.EDElay.MEDium
SCPI.CALCulate(Ch).SElected.CORRection.EDElay.WGCutoff
```

Equivalent key

Scale > Electrical Delay

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:CORRection:EDElay:TIME <numeric>
:CALCulate{[1]-160}{:SElected}:CORRection:EDElay:TIME?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDElay:TIME
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDElay:TIME?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:TIME 0.2"
20 OUTPUT 717;":CALC1:CORR:EDEL:TIME?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.WGCutoff

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.WGCutoff = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.CORRection.EDELay.WGCutoff

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.WGCutoff = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.WGCutoff

Description

This command sets/gets the cut-off frequency when waveguide is selected as an electrical delay time of channels 1 to 160 (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the cut-off frequency just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAMeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Cut-off frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

upper limit of the range is exceeded) is set.

Examples

```
Dim EdelWgc As Double
SCPI.CALCulate(1).SElected.CORRection.EDELay.WGCutoff = 1E9
Edel = SCPI.CALCulate(1).SElected.CORRection.EDELay.WGcutoff
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.CORRection.EDELay.MEDium
SCPI.CALCulate(Ch).SElected.CORRection.EDELay.TIME
```

Equivalent key

Scale > **Electrical Delay** > **Cutoff Frequency**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:CORRection:EDELay:WGCutoff
<numeric>
:CALCulate{[1]-160}{:SElected}:CORRection:EDELay:WGCutoff?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:WGCutoff
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:WGCutoff?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:WGC 1E9"
20 OUTPUT 717;":CALC1:CORR:EDEL:WGC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.CORRection.OFFSet.PHASE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.CORRection.OFFSet.PHASE = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.CORRection.OFFSet.PHASE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.OFFSet.PHASE = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.OFFSet.PHASE

Description

This command sets/gets the phase offset of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the phase offset of the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Phase offset
Data type	Double precision floating point type (Double)
Range	-360 to 360
Preset value	0
Unit	° (degree)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

upper limit of the range is exceeded) is set.

Examples

```
Dim Offset As Double
SCPI.CALCulate(2).PARAmeter(1).SElect
SCPI.CALCulate(2).SElected.CORRection.OFFSet.PHASe = 2.5
Offset = SCPI.CALCulate(2).SElected.CORRection.OFFSet.PHASe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Scale > Phase Offset

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{[:SElected]:CORRection:OFFSet:PHASe <numeric>
:CALCulate{[1]-160}{[:SElected]:CORRection:OFFSet:PHASe?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:OFFSet:PHASe
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:OFFSet:PHASe?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:CORR:OFFS:PHAS 2.5"
20 OUTPUT 717;":CALC1:CORR:OFFS:PHAS?"
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.DATA.FDATa

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.DATA.FDATa = *Data**Data* = SCPI.CALCulate(*Ch*).SElected.DATA.FDATa

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FDATa = *Data**Data* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FDATa

Description

This command sets/gets the formatted data array, for the active trace of selected channel (*Ch*).

The array data element varies in the data format (specified with the SCPI.CALCulate(*Ch*).SElected.FORMat object). For more information on the formatted data array, see Internal Data Processing.

NOTE

If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the formatted data array for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (formatted data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP.</p> <ul style="list-style-type: none"> <i>Data</i>(<i>n</i>×2-2) :Data (primary value) at the n-th measurement point. <i>Data</i>(<i>n</i>×2-1) :Data (secondary value) at the n-th measurement point. Always 0 when the data format is not the Smith chart format or the polar format. <p>The index of the array starts from 0.</p>

Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))×2 when setting a formatted data array, an error occurs when executed and the object is ignored.

Examples

```
Dim FmtData As Variant
SCPI.SENSE(1).SWEep.POINTs = 201
SCPI.CALCulate(1).PARAmeter(1).SElect
FmtData = SCPI.CALCulate(1).SElected.DATA.FDATa
SCPI.CALCulate(1).PARAmeter(2).SElect
SCPI.CALCulate(1).SElected.DATA.FDATa = FmtData
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).SWEep.POINTs
SCPI.CALCulate(Ch).SElected.FORMat
SCPI.CALCulate(Ch).SElected.DATA.FMEMory
SCPI.CALCulate(Ch).SElected.DATA.SDATa
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}[:SElected]:DATA:FDATa <numeric1>,... ,<numeric  
NOP*2>  
:CALCulate{[1]-160}[:SElected]:DATA:FDATa?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FDATa  
<numeric1>,... ,<numeric NOP*2>  
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FDATa?
```

Query response

```
{numeric 1},... ,{numeric NOP×2}<newline><^END>
```

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:FDAT?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(Ch).SElected.DATA.FMEMory

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.DATA.FMEMory = *Data**Data* = SCPI.CALCulate(*Ch*).SElected.DATA.FMEMory

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FMEMory = *Data**Data* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FMEMory

Description

This command sets/gets the formatted memory array, for the active trace of selected channel (*Ch*).

The array data element varies in the data format (specified with the SCPI.CALCulate(*Ch*).SElected.FORMat object). For more information on the formatted memory array, see Internal Data Processing.

- If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the formatted memory array for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (formatted memory array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP.</p> <ul style="list-style-type: none"> • <i>Data</i>(<i>n</i>×2-2) :Data (primary value) at the n-th measurement point. • <i>Data</i>(<i>n</i>×2-1) :Data (secondary value) at the n-th measurement point. Always 0 when the data format is not the Smith chart format or the polar format. <p>The index of the array starts from 0.</p>

Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))×2 when setting a formatted memory array, an error occurs when executed and the object is ignored.

Examples

```
Dim FmtMem As Variant
SCPI.SENSE(1).SWEep.POINTs = 201
SCPI.CALCulate(1).PARAmeter(1).SElect
FmtMem = SCPI.CALCulate(1).SElected.DATA.FMEMory
SCPI.CALCulate(1).PARAmeter(2).SElect
SCPI.CALCulate(1).SElected.DATA.FMEMory = FmtMem
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).SWEep.POINTs
SCPI.CALCulate(Ch).SElected.FORMat
SCPI.CALCulate(Ch).SElected.DATA.FDATa
SCPI.CALCulate(Ch).SElected.DATA.SMEMory
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:DATA:FMEMory <numeric  
1>,... ,<numeric NOP*2>  
:CALCulate{[1]-160}{:SElected}:DATA:FMEMory?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FMEMory  
<numeric1>,... ,<numeric NOP*2>  
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FMEMory?
```

Query response

```
{numeric 1},... ,{numeric NOP×2}<newline><^END>
```

Example of use

E5071C

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;"CALC1:DATA:FMEM?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SElected.DATA.SDATa

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.DATA.SDATa = *Data**Data* = SCPI.CALCulate(*Ch*).SElected.DATA.SDATa

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.SDATa = *Data**Data* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.SDATa

Description

This command sets/gets the corrected data array, for the active trace of selected channel (*Ch*).

For more information on the corrected data array, see Internal Data Processing

NOTE

If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the corrected data array for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (corrected data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP.</p> <ul style="list-style-type: none"> <i>Data</i>(<i>n</i>×2-2) :Real part of the data (complex number) at the n-th measurement point. <i>Data</i>(<i>n</i>×2-1) :Imaginary part of the data (complex number) at the n-th measurement point. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Note

If there is no array data of NOP (number of measurement point)×2 when setting a corrected data array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorData As Variant
SCPI.SENSE(1).SWEep.POINts = 201
CorData = SCPI.CALCulate(1).SElected.DATA.SDATa
SCPI.SENSE(2).SWEep.POINts = 201
SCPI.CALCulate(2).SElected.DATA.SDATa = CorData
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).SWEep.POINts
SCPI.CALCulate(Ch).SElected.DATA.SMEMory
SCPI.CALCulate(Ch).SElected.DATA.FDATa
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:DATA:SDATa <numeric 1>,... ,<numeric NOP*2>
```

```
:CALCulate{[1]-160}{:SElected}:DATA:SDATa?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SDATa
<numeric1>,... ,<numeric NOP*2>
```

```
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SDATa?
```

Query response

```
<numeric 1>,... ,<numeric NOP×2><^END>
```

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:SDAT?"
30 ENTER 717:A(*)
```


SCPI.CALCulate(Ch).SElected.DATA.SMEMory**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.DATA.SMEMory = *Data**Data* = SCPI.CALCulate(Ch).SElected.DATA.SMEMory

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).DATA.SMEMory = *Data**Data* = SCPI.CALCulate(Ch).TRACe(Tr).DATA.SMEMory**Description**

This command sets/gets the corrected memory array, for the active trace of selected channel (*Ch*).

For more information on the corrected memory array, see Section Internal Data Processing.

- If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the corrected memory array for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (corrected memory array) of NOP (number of measurement points)x2. Where n is an integer between 1 and NOP.</p> <ul style="list-style-type: none"> • <i>Data</i>($n \times 2 - 2$) :Real part of the data (complex number) at the n-th measurement point. • <i>Data</i>($n \times 2 - 1$) :Imaginary part of the data (complex number) at the n-th measurement point. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Note

If there is no array data of NOP (number of measurement point))*2 when setting a corrected memory array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorMem As Variant
SCPI.SENSE(1).SWEep.POINts = 201
CorMem = SCPI.CALCulate(1).SElected.DATA.SMEMory
SCPI.SENSE(2).SWEep.POINts = 201
SCPI.CALCulate(1).SElected.DATA.SMEMory = CorMem
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).SWEep.POINts
SCPI.CALCulate(Ch).SElected.DATA.SDATA
SCPI.CALCulate(Ch).SElected.DATA.FMEMory
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:DATA:SMEMory <numeric
1>,... ,<numeric NOP*2>
:CALCulate{[1]-160}{:SElected}:DATA:SMEMory?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SMEMory
<numeric1>,... ,<numeric NOP*2>
:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SMEMory?
```

Query response

```
<numeric 1>,... ,<numeric NOP*2><^END>
```

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:SMEM?"
30 ENTER 717:A(*)
```

SCPI.CALCulate(*Ch*).SELEcted.DATA.XAXis

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELEcted.DATA.XAXis

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.XAXis

Description

This command reads the data of measurement points of X axis, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the data of measurement points of X axis for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SELEct.

NOTE

Selection of the data format with SCPI.FORMat.DATA determines the format of the query result of this command.

Variable

Parameter	<i>Data</i>
Description	Indicates the array data (measurement points) of X axis
Data type	Variant type (Variant)

Examples

```
Dim AnaData As Variant
AnaData = SCPI.CALC1.SEL.DATA.XAX
```

Related objects

SCPI.FORMat.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

E5071C

All revision:

:CALCulate{[1]-160}{:SElected}:DATA:XAXis?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:XAXis?

Query response

<numeric 1>, ... , <numeric N> <^END>

Example of use

```
10 OUTPUT 717;":CALC1:SEL:DATA:XAX?"
20 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SElected.EQUation.STATE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.EQUation.STATE= *Data**Data* = SCPI.CALCulate(*Ch*).SElected.EQUation.STATE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.STATE= *Data**Data* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.STATE

Description

This command enable/disable the Equation Editor.

In revision A.9.60 and above, you can select the trace and enable/disable the Equation Editor for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	Ch
Description	Channel Number
Range	1 to 36

Parameter	<i>Data</i>
Description	Sets/Gets the state of equation in the Equation Editor
Data type	Boolean type (Boolean)

Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the Equation Editor. • False or OFF: Turns OFF the Equation Editor.
Preset value	False or OFF

Examples

```
Dim EqState As Boolean
EqState = ON
SCPI.CALCulate(1).SElected.EQUation.STATE = EqState
EqState = SCPI.CALCulate(1).SElected.EQUation.TEXT
```

Related objects

```
SCPI.CALCulate(Ch).SElected.EQUation.TEXT
SCPI.CALCulate(Ch).SElected.EQUation.VALid
```

Equivalent key**Display > Equation****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}[:SElected]:EQUation:STATE {ON|OFF|1|0}
:CALCulate{[1]-160}[:SElected]:EQUation:STATE?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-160}:EQUation:STATE {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-160}:EQUation:STATE?
```

Query response

{1/0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SElected:EQUation:STATE ON"
20 OUTPUT 717;":CALC1:SElected:EQUation:STATE?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.EQUation.TEXT

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.EQUation.TEXT = *Data**Data* = SCPI.CALCulate(*Ch*).SElected.EQUation.TEXT

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.TEXT = *Data**Data* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.TEXT

Description

This command sets/gets the equation in the Equation Editor. For valid parameters that can be used in this equation, refer to the Equation Editor.

In revision A.9.60 and above, you can select the trace and set/get the equation in the Equation Editor for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	Ch
Description	Channel Number
Range	1 to 36

Parameter	<i>Data</i>
Description	Sets/Gets the equation in the Equation Editor
Data type	Character string type (String)

Examples

Dim strEq As String

E5071C

```
SCPI.CALCulate(1).SElected.EQUation.TEXT = "Example=S21/(1-S11)"  
strEq = SCPI.CALCulate(1).SElected.EQUation.TEXT
```

Related objects

```
SCPI.CALCulate(Ch).SElected.EQUation.STATE
```

```
SCPI.CALCulate(Ch).SElected.EQUation.VALid
```

Equivalent key

No equivalent key is available on the front panel for Equation Text but Equation Editor can be accessed through **Display** > **Equation Editor**.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:EQUation:TEXT <string1>
```

```
:CALCulate{[1]-160}{:SElected}:EQUation:TEXT?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:EQUation:TEXT <string1>
```

```
:CALCulate{[1]-160}:TRACe{[1]-16}:EQUation:TEXT?
```

Query response

```
<string 1><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:SElected:EQUation:TEXT?"  
30 ENTER 717;A$
```


SCPI.CALCulate(Ch).SElected.EQUation.VALid

Object type

Property (**Read Only**)

Syntax

In revision A.9.5x and below:

Data = SCPI.CALCulate(Ch).SElected.EQUation.VALid

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(Ch).TRACe(Tr).EQUation.VALid

Description

This command returns False when the equation expression and label are correct but the required S-parameter data is not measured or if it refers the invalid corrected memory array. Annotation of '**Equ!**' is displayed when this command returns a False value.

- Equation Editor can refer S parameter data and data present in corrected memory array.

In revision A.9.60 and above, you can select the trace and check the validity of the equation for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	Ch
Description	Channel Number
Range	1 to 36

Parameter	<i>Data</i>
Description	A boolean value which gets the state of the equation in the Equation Editor as invalid (False) or valid (True)
Data type	Boolean type (Boolean)

Range	True or ON: Valid Spara data False or OFF: Invalid Spara data
Preset value	False or OFF

Examples

```
Dim EqState As Boolean
```

```
EqState = SCPI.CALCulate(1).SElected.EQUation.VALid
```

Related objects

```
SCPI.CALCulate(Ch).SElected.EQUation.STATE
```

```
SCPI.CALCulate(Ch).SElected.EQUation.TEXT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{[:SElected]:EQUation:VALID?}
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:EQUation:VALID?
```

Query response

```
<1/0><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:EQUation:VALid?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.CENTer

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.CENTer = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.CENTer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.CENTer = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.CENTer

Description

This command sets/gets the center value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the center value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	The center value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.
--	--------------------------------------------------------------------------------------------

Examples

```
Dim FilCent As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.CENTer = 1E-8
FilCent = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.CENTer
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.SPAN
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key**Analysis > Gating > Center****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:CENTer <numeric>
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:CENTer?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:CENTer
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:CENTer?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:CENT 1E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME. SHAPE**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.SHAPE = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.SHAPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SHAPE = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SHAPE**Description**

This command set/get the shape of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	The shape of the gate
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "MAXimum": Specifies the maximum shape. • "WIDE": Specifies the wide shape. • "NORMal": Specifies the normal shape. • "MINimum": Specifies the minimum shape.
Preset value	"NORMal"

Examples

E5071C

```
Dim FilShape As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.SHAPe = "wide"
FilShape = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.SHAPe
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.TYPE
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Gating > Shape > Maximum|Wide|Normal|Minimum

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:SHAPe {MAXimum|
WIDE|NORMAl|MINimum}
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:SHAPe?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SHAPe
{MAXimum| WIDE|NORMAl|MINimum}
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SHAPe?
```

Query response

```
{MAX|WIDE|NORM|MIN}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:SHAP WIDE"
20 OUTPUT 717;":CALC1:FILT:TIME:SHAP?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.SPAN**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.SPAN = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.SPAN

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SPAN = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SPAN**Description**

This command sets/gets the span value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the span value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	The span value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	2E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim FilStar As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.SPAN = 1E-8
FilStar = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.SPAN
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.CENTer
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Gating > Span

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:SPAN <numeric>
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:SPAN?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SPAN
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SPAN?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:SPAN 1E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:SPAN?"
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.START

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.START = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.START

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.START = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.START

Description

This command sets/gets the start value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the start value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	The start value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	-1E-8
Unit	s (second)

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim FilCent As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.START = 0
FilCent = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.START
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STOP
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Gating > Start

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:START <numeric>
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:START?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:START
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:START?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STAR 0"
20 OUTPUT 717;":CALC1:FILT:TIME:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME. STATE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.STATE = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.STATE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STATE = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STATE

Description

This command turns ON/OFF the gating function of the time domain function, for the active trace of selected channel (*Ch*).

You can turn ON the gating function only when the sweep type is the linear sweep and the number of points is 3 or more. If you execute this object to try to turn ON the gating function when the sweep type is other than the linear sweep or the number of points is less than 3, an error occurs and the object is ignored.

When the sweep type is the power sweep, you cannot turn on the gating function. If you execute this object trying to turn on the gating function during the power sweep, an error occurs and the object is ignored.

In revision A.9.60 and above, you can select the trace and turn ON/OFF the gating function of the time domain function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the gating function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the gating function. • False or OFF: Turns OFF the gating function.

Preset value	False or OFF
---------------------	--------------

Examples

```
Dim Gating As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.STATe = True
Gating = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).SWEep.TYPE
SCPI.SENSE(Ch).SWEep.POINts
```

Equivalent key

Analysis > Gating > Gating

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:STATe
{ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:STATe?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STATe
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STAT ON"
20 OUTPUT 717;":CALC1:FILT:TIME:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME. STOP**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FILTer.GATE.TIME.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STOP**Description**

This command sets/gets the stop value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the stop value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	The stop value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.
--	--------------------------------------------------------------------------------------------

Examples

```
Dim FilStop As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.STOP = 2E-8
FilStop = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.STOP
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.START
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key**Analysis > Gating > Stop****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:STOP <numeric>
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME:STOP?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STOP
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STOP?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STOP 2E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME. TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.TYPE = *Param**Param* = SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME.TYPE = *Param**Param* = SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME.TYPE

Description

This command sets/gets the gate type used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the gate type used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Param</i>
Description	The gate type
Data type	Character string type (String)
Range	Select from the following. <ul style="list-style-type: none"> • "BPASs": Specifies the band-pass type. • "NOTCh": Specifies the notch type.
Preset value	"BPASs"

Examples

```
Dim FilType As String
SCPI.CALCulate(1).PARAmeter(1).SElect
```

E5071C

```
SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.SHAPe = "notc"  
FilType = SCPI.CALCulate(1).SElected.FILTer.GATE.TIME.SHAPe
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.SHAPe  
SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe  
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Gating > Type

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME[:TYPE]  
{BPASs|NOTCh}  
:CALCulate{[1]-160}{:SElected}:FILTer[:GATE]:TIME[:TYPE]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME[:TYPE]  
{BPASs|NOTCh}  
:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME[:TYPE]?
```

Query response

```
{BPAS|NOTC}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME NOTC"  
20 OUTPUT 717;":CALC1:FILT:TIME?"  
30 ENTER 717;A$
```


SCPI.CALCulate(*Ch*).SElected.FORMat

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FORMat = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.FORMat

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FORMat = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FORMat

Description

This command sets/gets the data format of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the data format of the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Data format
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "MLOGarithmic": Specifies the log magnitude format. • "PHASe": Specifies the phase format. • "GDElay": Specifies the group delay format. • "SLINear": Specifies the Smith chart format (Lin/Phase). • "SLOGarithmic": Specifies the Smith chart format (Log/Phase). • "SCOMplex": Specifies the Smith chart format

	<p>(Re/Im).</p> <ul style="list-style-type: none"> • "SMITH": Specifies the Smith chart format (R+jX). • "SADMittance": Specifies the Smith chart format (G+jB). • "PLINear": Specifies the polar format (Lin/Phase). • "PLOGarithmic": Specifies the polar format (Log/Phase). • "POLar": Specifies the polar format (Re/Im). • "MLINear": Specifies the linear magnitude format. • "SWR": Specifies the SWR format. • "REAL": Specifies the real format. • "IMAGinary": Specifies the imaginary format. • "UPHase": Specifies the expanded phase format. • "PPHase": Specifies the positive phase format.
Preset value	"MLOGarithmic"

Examples

```
Dim Fmt As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FORMat = "smit"
Fmt = SCPI.CALCulate(1).SElected.FORMat
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Format > **Log Mag**|**Phase**|**Group Delay**|**Lin Mag**|**SWR**|**Real**|**Imaginary**|**Expand Phase**| **Positive Phase**

Format > **Smith** > **Lin/Phase**|**Log/Phase**|**Real/Imag**|**R+jX**|**G+jB**

Format > **Polor** > **Lin/Phase**|**Log/Phase**|**Real/Imag**

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FORMat {MLOGarithmic|PHASe|GDElay|
SLINear|SLOGarithmic|SCOMplex|SMITH|SADMittance|PLINear|PLOGarith
mic|POLar|MLINear|SWR|REAL| IMAGinary|UPHase|PPHase}
:CALCulate{[1]-160}{:SElected}:FORMat?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FORMat
{MLOGarithmic|PHASe|GDElay|
SLINear|SLOGarithmic|SCOMplex|SMITH|SADMittance|PLINear|PLOGarith
mic|POLar|MLINear|SWR|REAL| IMAGinary|UPHase|PPHase}
:CALCulate{[1]-160}:TRACe{[1]-16}:FORMat?
```

Query response

```
{MLOG|PHAS|GDEL|SLIN|SLOG|SCOM|SMIT|SADM|PLIN|PLOG|POL|MLIN|
SWR| REAL|IMAG|UPH|PPH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FORM SLIN"
20 OUTPUT 717;":CALC1:FORM?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.FUNCTION.DATA**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.FUNCTION.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.DATA**Description**

This command reads the analysis result of the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the analysis result of the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (analysis result) of N (number of data pairs)x2. N (number of data pairs) can be read out with the SCPI.CALCulate(Ch).SElected.FUNCTION.POINTs object. Where n is an integer between 1 and N.</p> <ul style="list-style-type: none"> <i>Data</i>(<i>n</i>*2-2) :Response value or analysis result of the searched n-th measurement point. <i>Data</i>(<i>n</i>*2-1) :Stimulus value of the searched n-th measurement point. Always 0 for the analysis of the mean value, the standard deviation, and the difference between the maximum value and the minimum value. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

```
Dim AnaData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNction.TYPE = "mean"
SCPI.CALCulate(1).SElected.FUNction.EXECute
AnaData = SCPI.CALCulate(1).SElected.FUNction.DATA
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNction.TYPE
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
SCPI.CALCulate(Ch).SElected.FUNction.POINTs
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:DATA?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DATA?
```

Query response

```
<numeric 1>, ... , <numeric N*2> <^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:POIN?"
20 ENTER 717;A
30 REDIM B(1:2*A)
40 OUTPUT 717;":CALC1:FUNC:DATA?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SElected.FUNction.DOMain.COUPle

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNction.DOMain.COUPle = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.FUNction.DOMain.COUPle

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.DOMain.COUPle = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.DOMain.COUPle

Description

This command specifies whether to set the coupling of the analysis range of the SCPI.CALCulate(*Ch*).SElected.FUNction.EXECute object for all traces, for the selected channel (*Ch*).

In revision A.9.60 and above, even though you can select the trace and set the coupling of the analysis range of the SCPI.CALCulate(*Ch*).SElected.FUNction.EXECute object for all traces just by executing the above TRACe(*Tr*) command, the *Tr* value is omitted. This is because the command is applied to all the traces.

Variable

Parameter	<i>Status</i>
Description	On/off of the trace coupling of the analysis range.
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Specifies the analysis range with the trace coupling. • False or OFF: Specifies the analysis range for each trace.
Preset value	True or ON

Examples

```
Dim TrCpl As Boolean
SCPI.CALCulate(1).SElected.FUNction.DOMain.COUPle = False
TrCpl = SCPI.CALCulate(1).SElected.FUNction.DOMain.COUPle
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}[:SElected]:FUNction:DOMain:COUPle {ON|OFF|1|0}
:CALCulate{[1]-160}[:SElected]:FUNction:DOMain:COUPle?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:COUPle
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:COUP OFF"
20 OUTPUT 717;":CALC1:FUNC:DOM:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.START**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNCTION.DOMain.START = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FUNCTION.DOMain.START

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.DOMain.START = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.DOMain.START**Description**

This command sets/gets the start value of the analysis range of the SCPI.CALCulate(*Ch*).SElected.FUNCTION.EXECute object, for the selected channel (*Ch*).

When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the start value of the analysis range of the SCPI.CALCulate(*Ch*).SElected.FUNCTION.EXECute object for the trace (when trace coupling is off) just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Start value of the analysis range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples


```
Dim AnaStar As Double
SCPI.CALCulate(1).SElected.FUNction.DOMain.START = 1.5E9
AnaStar = SCPI.CALCulate(1).SElected.FUNction.DOMain.START
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STOP
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STAtE
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.COUPle
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}[:SElected]:FUNction:DOMain:START <numeric>
:CALCulate{[1]-160}[:SElected]:FUNction:DOMain:START?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:START <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:START?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:STAR 1.7E9"
20 OUTPUT 717;":CALC1:FUNC:DOM:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.STATe = *Status**Status* = SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).FUNCTION.DOMain.STATe = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).FUNCTION.DOMain.STATe

Description

This command sets/gets whether to use an arbitrary range when executing the analysis with the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object, for the selected channel (*Ch*).

When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get whether to use an arbitrary range when executing the analysis with the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object for the trace (when trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	Selection of the analysis range
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Specifies an arbitrary range • False or OFF: Specifies the entire sweep range
Preset	False or OFF

value	
--------------	--

Examples

```
Dim AnaRnge As Boolean
SCPI.CALCulate(1).SElected.FUNction.DOMain.START = 1.5E9
SCPI.CALCulate(1).SElected.FUNction.DOMain.STOP = 1.8E9
SCPI.CALCulate(1).SElected.FUNction.DOMain.STATe = True
AnaRnge = SCPI.CALCulate(1).SElected.FUNction.DOMain.STATe
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.START
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STOP
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.COUPLE
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:DOMain[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:FUNction:DOMain[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.STOP

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNCTION.DOMain.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FUNCTION.DOMain.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.DOMain.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.DOMain.STOP

Description

This command sets/gets the stop value of the analysis range of the SCPI.CALCulate(*Ch*).SElected.FUNCTION.EXECute object, for the selected channel (*Ch*).

When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the stop value of the analysis range of the SCPI.CALCulate(*Ch*).SElected.FUNCTION.EXECute object for the trace (when trace coupling is off) just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Stop value of the analysis range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim AnaStop As Double
SCPI.CALCulate(1).SElected.FUNction.DOMain.STOP = 1.8E9
AnaStop = SCPI.CALCulate(1).SElected.FUNction.DOMain.STOP
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.START
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STAtE
SCPI.CALCulate(Ch).SElected.FUNction.DOMain.COUPle
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:DOMain:STOP <numeric>
:CALCulate{[1]-160}{:SElected}:FUNction:DOMain:STOP?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:STOP <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:STOP 1.8E9"
20 OUTPUT 717;":CALC1:FUNC:DOM:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute**Object type**Method (**Write-only**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).FUNCTION.EXECute

Description

This command executes the analysis specified with the SCPI.CALCulate(Ch).SElected.FUNCTION.TYPE object, for the active trace of selected channel (Ch).

In revision A.9.60 and above, you can select the trace and execute the analysis specified with the SCPI.CALCulate(Ch).SElected.FUNCTION.TYPE object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Examples

```
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNCTION.EXECute
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNCTION.DATA
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNCTION.TYPE
SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.STATe
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

:CALCulate{[1]-160}[:SElected]:FUNCTION:EXECute

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCTION:EXECute

Example of use

10 OUTPUT 717;":CALC1:FUNC:EXEC"

SCPI.CALCulate(*Ch*).SElected.FUNction.PEXCursion

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNction.PEXCursion = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FUNction.PEXCursion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.PEXCursion = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value (the minimum value of the difference relative to the right and left adjacent measurement points) when executing the peak search with the SCPI.CALCulate(*Ch*).SElected.FUNction.EXECute object, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the lower limit of peak excursion value when executing the peak search with the SCPI.CALCulate(*Ch*).SElected.FUNction.EXECute object, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Lower limit of peak excursion value
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3

Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> • Log magnitude (MLOG) : dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree) • Group delay (GDEL) : s (second) • Others : No unit
Note	<p>If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNction.TYPE = "peak"
SCPI.CALCulate(1).SElected.FUNction.PEXCursion = 1.5
PeakExc = SCPI.CALCulate(1).SElected.FUNction.PEXCursion
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNction.TYPE
SCPI.CALCulate(Ch).SElected.FUNction.PPOLarity
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}[[:SElected]:FUNction:PEXCursion <numeric>
:CALCulate{[1]-160}[[:SElected]:FUNction:PEXCursion?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:PEXCursion <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:PEXCursion?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:PEXC 0.2"  
20 OUTPUT 717;":CALC1:FUNC:PEXC?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.FUNCTION.POINTs**Object type**Property (**Read Only**)**Syntax**

In revision A.9.5x and below:

Value = SCPI.CALCulate(Ch).SElected.FUNCTION.POINTs

In revision A.9.60 and above, the following syntax is also supported:

Value = SCPI.CALCulate(Ch).TRACE(Tr).FUNCTION.POINTs**Description**

This command reads the number of data pairs of the analysis result of the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object, for the active trace of selected channel (*Ch*).

For the analysis of the mean value or the search of the maximum value, 1 is always read out; for the search of all peaks or the search of all targets, the total number of searched measurement points is read out.

In revision A.9.60 and above, you can select the trace and read the number of data pairs of the analysis result of the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object for the trace just by executing the above TRACE(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAMeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	Number of analyzed data pairs
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim AnaPoin As Long
SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.FUNCTION.TYPE = "ape"
SCPI.CALCulate(1).SElected.FUNCTION.EXECute
AnaPoin = SCPI.CALCulate(1).SElected.FUNCTION.POINTs
```

Related objects

SCPI.CALCulate(Ch).SELEcted.FUNCtion.DATA
 SCPI.CALCulate(Ch).PARAmeter(Tr).SELEct
 SCPI.CALCulate(Ch).SELEcted.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SELEcted}:FUNCtion:POINts?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:POIN?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.FUNCTION.PPOLarity

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNCTION.PPOLarity = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.FUNCTION.PPOLarity

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.PPOLarity = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.PPOLarity

Description

This command sets/gets the polarity when performing the peak search with the SCPI.CALCulate(*Ch*).SElected.FUNCTION.EXECute object, for the active trace of the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the polarity when performing the peak search with the SCPI.CALCulate(*Ch*).SElected.FUNCTION.EXECute object for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Polarity for peak search
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "POSitive": Specifies the positive peak. • "NEGative": Specifies the negative peak. • "BOTH": Specifies both the positive peak and the negative peak.
Preset	"POSitive"

value	
--------------	--

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNction.TYPE = "peak"
SCPI.CALCulate(1).SElected.FUNction.PPOLarity = "both"
PeakPol = SCPI.CALCulate(1).SElected.FUNction.PPOLarity
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNction.TYPE
SCPI.CALCulate(Ch).SElected.FUNction.PEXCursion
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:PPOLarity {POSitive|
NEGative|BOTH}
:CALCulate{[1]-160}{:SElected}:FUNction:PPOLarity?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:PPOLarity {POSitive|
NEGative|BOTH}
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:PPOLarity?
```

Query response

```
{POS|NEG|BOTH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:PPOL BOTH"
20 OUTPUT 717;":CALC1:FUNC:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.FUNCTION.TARGET

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNCTION.TARGET = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.FUNCTION.TARGET

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.TARGET = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCTION.TARGET

Description

This command sets/gets the target value when performing the target search with the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object, for the active trace of the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the target value when performing the target search with the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute object for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Target value
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0

Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> • Log magnitude (MLOG) : dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree) • Group delay (GDEL) : s (second) • Others : No unit
Note	<p>If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNction.TYPE = "atar"
SCPI.CALCulate(1).SElected.FUNction.TARGet = -12.5
TargVal = SCPI.CALCulate(1).SElected.FUNction.TARGet
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNction.TYPE
SCPI.CALCulate(Ch).SElected.FUNction.TTRansition
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:TARGet <numeric>
:CALCulate{[1]-160}{:SElected}:FUNction:TARGet?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:TARGet <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:TARGet?
```

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TARG -12.5"  
20 OUTPUT 717;":CALC1:FUNC:TARG?"  
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.FUNction.TTRansition

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNction.TTRansition = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.FUNction.TTRansition

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.TTRansition = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.TTRansition

Description

This command sets/gets the transition type when performing the target search with the SCPI.CALCulate(*Ch*).SElected.FUNction.EXECute object, for the active trace of the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the transition type when performing the target search with the SCPI.CALCulate(*Ch*).SElected.FUNction.EXECute object for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Transition type for search
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.
Preset	"BOTH"

value

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNction.TYPE = "atar"
SCPI.CALCulate(1).SElected.FUNction.TTRansition = "pos"
TargTran = SCPI.CALCulate(1).SElected.FUNction.TTRansition
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNction.TYPE
SCPI.CALCulate(Ch).SElected.FUNction.TARGet
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:TTRansition {POSitive|
NEGative|BOTH}
:CALCulate{[1]-160}{:SElected}:FUNction:TTRansition?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:TTRansition {POSitive|
NEGative|BOTH}
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:TTRansition?
```

Query response

```
{POS|NEG|BOTH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TTR NEG"
20 OUTPUT 717;":CALC1:FUNC:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SElected.FUNction.TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.FUNction.TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.FUNction.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNction.TYPE

Description

This command sets/gets the type of analysis, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the type of analysis for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Analysis type
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "PTPeak": Specifies the analysis of the difference between the maximum value and the minimum value (Peak to Peak). • "STDEV": Specifies the analysis of the standard deviation. • "MEAN": Specifies the analysis of the mean value. • "MAXimum": Specifies the search for the maximum value. • "MINimum": Specifies the search for the

	<p>minimum value.</p> <ul style="list-style-type: none"> • "PEAK": Specifies the search for the peak. • "APEak": Specifies the search for all peaks. • "ATARget": Specifies the search for all targets.
Preset value	"PTPeak"

Examples

```
Dim AnaType As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.FUNction.TYPE = "atar"
AnaType = SCPI.CALCulate(1).SElected.FUNction.TYPE
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FUNction.DATA
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.FUNction.PEXCursion
SCPI.CALCulate(Ch).SElected.FUNction.PPOLarity
SCPI.CALCulate(Ch).SElected.FUNction.TARGet
SCPI.CALCulate(Ch).SElected.FUNction.TTRansition
SCPI.CALCulate(Ch).SElected.FUNction.EXECute
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:FUNction:TYPE {PTPeak| STDEV|MEAN|
MAXimum|MINimum|PEAK|APEak|ATARget}
:CALCulate{[1]-160}{:SElected}:FUNction:TYPE?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNctionTYPE {PTPeak|
STDEV|MEAN| MAXimum|MINimum|PEAK|APEak|ATARget}
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:TYPE?
```

Query response

{PTP|STDEV|MEAN|MAX|MIN|PEAK|APE|ATAR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TYPE PEAK"
20 OUTPUT 717;":CALC1:FUNC:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.LIMit.DATA

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.LIMit.DATA = *Data**Data* = SCPI.CALCulate(Ch).SElected.LIMit.DATA

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).LIMit.DATA = *Data**Data* = SCPI.CALCulate(Ch).TRACe(Tr).LIMit.DATA

Description

This command sets/gets the limit table for the limit test, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the limit table for the limit test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (for limit line) of 1 + Num (number of limit lines)*5. Where n is an integer between 1 and Num.</p> <ul style="list-style-type: none"> <i>Data(0)</i> :The number of limit lines you want to set. Specify an integer ranging 0 to 100. When the number of limit lines is set to 0 (clears the limit table), the variable Data is only required with <i>Data(0)</i>. <i>Data(n*5-4)</i> :The type of the n-th line. Specify an integer 0 to 2 as follows. 0: OFF 1: Upper limit line 2: Lower limit line <i>Data(n*5-3)</i> :The value on the horizontal axis (frequency/power/time) of the start point of the

	<p>n-th line.</p> <ul style="list-style-type: none"> • <i>Data(n*5-2)</i> :The value on the horizontal axis (frequency/power/time) of the end point of the n-th line. • <i>Data(n*5-1)</i> :The value on the vertical axis of the start point of the n-th line. • <i>Data(n*5)</i> :The value on the vertical axis of the end point of the n-th line. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Note	<p>If there is no array data of 1+Num (number of set lines))*5 when setting a formatted memory array, an error occurs when executed and the object is ignored. For <i>Data(n*5-4)</i> in the array data, if you specify an integer other than 0, 1 or 2, an error occurs when executed. For <i>Data(n*5-3)</i>, <i>Data(n*5-2)</i>, <i>Data(n*5-1)</i>, and <i>Data(n*5)</i> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```

Dim LimData As Variant
SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.DATA = Array(1,1,1e6,1e9,0,0)
LimData = SCPI.CALCulate(1).SElected.LIMit.DATA

SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.DATA = Array(0) 'Clear Limit Table

```

```

Dim LimData(5) As Variant
Dim Ref As Variant
LimData(0) = 1
LimData(1) = 1
LimData(2) = 1e6
LimData(3) = 1e9
LimData(4) = 0
LimData(5) = 0
SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.DATA = LimData
Ref = SCPI.CALCulate(1).SElected.LIMit.DATA

```

E5071C

```
Dim LimData(0) As Variant
LimData(0) = 0
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.DATA = LimData 'Clear Limit Table
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.LIMit.STATe
SCPI.CALCulate(Ch).SElected.LIMit.DISPlay.STATe
SCPI.CALCulate(Ch).SElected.LIMit.FAIL
```

Equivalent key

Analysis > Limit Test > Edit Limit Line

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:LIMit:DATA <numeric 1>, ... ,<numeric
1+(N*5)>
:CALCulate{[1]-160}{:SElected}:LIMit:DATA?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DATA <numeric 1>, ...
,<numeric 1+(N*5)>
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DATA?
```

Query response

```
{numeric 1}, ... ,{numeric 1+(N*5)}<newline><^END>
```

Example of use

```
10 DIM B(1:2,1:5)
20 OUTPUT 717;":CALC1:LIM:DATA 2,1,1E9,3E9,0,0,2,1E9,3E9,-3,-3"
30 OUTPUT 717;":CALC1:LIM:DATA?"
40 ENTER 717;A,B(*)
10 OUTPUT 717;":CALC1:LIM:DATA 0" ! Clear Limit Table
```


SCPI.CALCulate(*Ch*).SELEcted.LIMit.DISPlay.STATe**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SELEcted.LIMit.DISPlay.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SELEcted.LIMit.DISPlay.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.DISPlay.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.DISPlay.STATe**Description**

This command turns ON/OFF the limit line display, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the limit line display for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SELEct.

Variable

Parameter	<i>Status</i>
Description	Limit line display
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the limit line display. • False or OFF: Turns OFF the limit line display.
Preset value	False or OFF

Examples

E5071C

```
Dim LimDisp As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit = True
LimDisp = SCPI.CALCulate(1).SElected.LIMit.DISPlay.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.LIMit.STATe
```

Equivalent key

Analysis > **Limit Test** > **Limit Line**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:LIMit:DISPlay[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:LIMit:DISPlay[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DISPlay[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DISPlay[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:LIM:DISP ON"
20 OUTPUT 717;":CALC1:LIM:DISP?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.LIMit.FAIL

Object type

Property (**Read Only**)

Syntax

All revision:

Status = SCPI.CALCulate(Ch).SElected.LIMit.FAIL

In revision A.9.60 and above, the following syntax is also supported:

Status = SCPI.CALCulate(Ch).TRACe(Tr).LIMit.FAIL

Description

This command reads the limit test result, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the limit test result for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	Limit test result
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: The limit test result is FAIL. • False or OFF: The limit test result is PASS.
Note	When the limit test is set to OFF, False or OFF is always read out.

Examples

```

Dim Result As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.STATe = True
Result = SCPI.CALCulate(1).SElected.LIMit.FAIL

```

Related objects

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect

E5071C

SCPI.CALCulate(Ch).SELEcted.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SELEcted}:LIMit:FAIL?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:FAIL?

Query response

{1|0}<newline><^END>

	Description
1	The limit test result is FAIL.
0	The limit test result is PASS.

When the limit test is set to OFF, 0 is always read out.

Example of use

10 OUTPUT 717;":CALC1:LIM:FAIL?"
20 ENTER 717;A

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude = *Value**Value* = SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.AMPLitude = *Value**Value* = SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.AMPLitude

Description

This command sets/gets the limit line offset of response for the selected active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command) of the selected channel (Ch).

The setting of the limit line doesn't change even if the offset value is changed.

In revision A.9.60 and above, you can select the trace and set/get the limit line offset of response for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	The limit line offset of Response (Vertical offset)
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	dB

For information on the variable (*Ch*), see Variable.

Examples

```
Dim LimOffset As Double
SCPI.CALCulate(1).SElected.LIMit.OFFSet.AMPLitude = -10
LimOffset = SCPI.CALCulate(1).SElected.LIMit.OFFSet.AMPLitude
```

Related objects

```
SCPI.CALCulate(Ch).SElected.LIMit.STATe
SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.MARKer
SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.STIMulus
```

Equivalent key

Analysis > Limit Test > Limit Line Offsets > Amplitude Offset

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:LIMit:OFFSet:AMPLitude <numeric>
:CALCulate{[1]-160}{:SElected}:LIMit:OFFSet:AMPLitude?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:AMPLitude <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:AMPLitude?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:AMPL -10"
20 OUTPUT 717;":CALC1:LIM:OFFS:AMPL?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.MARKer**Object type**Method (**Write-only**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.MARKer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.MARKer

Description

This command sets the *active marker* value to amplitude offset using the limit line, for the selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

- The setting of the limit line does not change even if the offset value is changed.

When the markers are not displayed, this command does not operate.

In revision A.9.60 and above, you can select the trace and set the active marker value to amplitude offset using the limit line for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

For information on the variable (*Ch*), see Variable.

Examples

```
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.OFFSet.MARKer
```

Related objects

```
SCPI.CALCulate(Ch).SElected.LIMit.STATe
SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude
SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.STIMulus
```

Equivalent key

Analysis > Limit Test > Limit Line Offsets > Marker -> Amplitude Offset

Equivalent SCPI command**Syntax**

All revision:

:CALCulate{[1]-160}[:SElected]:LIMit:OFFSet:MARKer

E5071C

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:MARKer

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:MARK"
```


SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.STIMulus**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.STIMulus = *Value**Value* = SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.STIMulus

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.STIMulus = *Value**Value* = SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.STIMulus**Description**

This command sets/gets the stimulus offset of the limit line, for the selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

- The setting of the limit line doesn't change even if the offset value is changed.

In revision A.9.60 and above, you can select the trace and set/get the stimulus offset of the limit line for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	The stimulus offset of the limit line
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz), dBm or second

For information on the variable (*Ch*), see Variable.

Examples

```
Dim LimOffset As Double
SCPI.CALCulate(1).SElected.LIMit.OFFSet.STIMulus = 1E9
LimOffset = SCPI.CALCulate(1).SElected.LIMit.OFFSet.STIMulus
```

Related objects

```
SCPI.CALCulate(Ch).SElected.LIMit.STATe
SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude
SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.MARKer
```

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Stimulus Offset**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:LIMit:OFFSet:STIMulus <numeric>
:CALCulate{[1]-160}{:SElected}:LIMit:OFFSet:STIMulus?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:STIMulus <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:STIMulus?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:STIM 5E3"
20 OUTPUT 717;":CALC1:LIM:OFFS:STIM?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.LIMit.REPort.ALL

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(Ch).SElected.LIMit.REPort.ALL

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.ALL

Description

This command reads the bandwidth test results (stimulus value, limit test result, upper limit value and lower limit value of all measurement points), for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command).

In revision A.9.60 and above, you can select the trace and read the bandwidth test results for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	Data
Description	<p>Indicates the array data (for limit line) of NOP (number of measurement points)x4. Where n is an integer between 1 and NOP.</p> <ul style="list-style-type: none"> <i>Data</i>(n×4-3) The stimulus value for the measurement point. <i>Data</i>(n×4-2) The limit test result. Specify an integer -1 to 1 as follows. -1: No limit 0: Fail 1: Pass <i>Data</i>(n×4-1) The upper limit value at the measurement point. (If there is no limit at this point, reads out the 0.) <i>Data</i>(n×4) The lower limit value at the measurement point. (If there is no limit at this point, reads out the 0.)

	The index of the array starts from 0.
Data type	Variant type (Variant)

For information on the variable (*Ch*), see Variable.

Examples

```
Dim LimData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
LimData = SCPI.CALCulate(1).SElected.LIMit.REPort.ALL
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.LIMit.STATe
SCPI.CALCulate(Ch).SElected.LIMit.REPort.DATA
SCPI.CALCulate(Ch).SElected.LIMit.REPort.POINTs
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{[:SElected]:LIMit:REPort:ALL?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:REPort:ALL?
```

Example of use

```
10 OUTPUT 717;":SENS1:SWE:POIN?"
20 ENTER 717;A
30 REDIM B(1:4*A)
40 OUTPUT 717;":CALC1:LIM:REP:ALL?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(Ch).SElected.LIMit.REPort.DATA**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.LIMit.REPort.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.REPort.DATA**Description**

This command reads the stimulus values (frequency, power level or time) at all the measurement points that failed the limit test, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the stimulus values at all the measurement points that failed the limit test, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	Indicates the array data for failed measurement points (can be read out with the SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.REPort.POINts object).
Data type	Variant type (Variant)

Examples

```
Dim FailData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.STATe = True
FailData = SCPI.CALCulate(1).SElected.LIMit.REPort.DATA
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.LIMit.REPort.POINts
SCPI.CALCulate(Ch).SElected.LIMit.STATe
```

Equivalent key

No equivalent key is available on the front panel.

E5071C

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SElected]:LIMit:REPort[:DATA]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:REPort[:DATA]?

Query response

{numeric 1},...,{numeric N}<newline><^END>

Where N is the number of the measurement points that failed (can be read out with the :CALC{1-160}:LIM:REP:POIN? command).

Example of use

```
10 OUTPUT 717;":CALC1:LIM:REP:POIN?"
20 ENTER 717;A
30 REDIM B(1:A)
40 OUTPUT 717;":CALC1:LIM:REP?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(Ch).SElected.LIMit.REPort.POINts**Object type**Property (**Read Only**)**Syntax**

All revision:

Value = SCPI.CALCulate(*Ch*).SElected.LIMit.REPort.POINts

In revision A.9.60 and above, the following syntax is also supported:

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.REPort.POINts**Description**

This command reads the number of the measurement points that failed the limit test, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the number of the measurement points that failed the limit test, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Number of measurement points that failed
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim FailPoin As Long
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.STATe = True
FailPoin = SCPI.CALCulate(1).SElected.LIMit.REPort.POINts
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.LIMit.STATe
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:LIMit:REPort:POINts?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:REPort:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:REP:POIN?"  
20 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.LIMit.STATe**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.LIMit.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.LIMit.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.STATe**Description**

This command turns ON/OFF the limit test function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the limit test function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the limit test function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the limit test function. • False or OFF: Turns OFF the limit test function.
Preset value	False or OFF

Examples

E5071C

```
Dim LimTest As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.LIMit.STATe = True
LimTest = SCPI.CALCulate(1).SElected.LIMit.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.LIMit.DISPlay.STATe
SCPI.DISPlay.FSIGN
```

Equivalent key

Analysis > Limit Test > Limit Test

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:LIMit[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:LIMit[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:LIM ON"
20 OUTPUT 717;":CALC1:LIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).ACTivate**Object type**Method (**Write-only**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).ACTivate

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).ACTivate**Description**

This command sets the marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the *active marker*, for the active trace of selected channel (*Ch*).

- If you set a marker not displayed to the active marker, the marker display is automatically set to ON.

In revision A.9.60 and above, you can select the trace and set the marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the *active marker*, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Mk</i>
Description	Marker number
Data type	Long integer type (Long)
Range	1 to 10 Notice that 10 is for the reference marker.
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).ACTivate
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.DISPlay.WINDow(Ch).ACTivate
```

E5071C

Equivalent key

Marker > **Marker 1|Marker 2|Marker 3|Marker 4|Ref Marker**

Marker > **More Markers** > **Marker 5|Marker 6|Marker 7|Marker 8|Marker 9**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{[:SElected]:MARKer{[1]-10}:ACTivate

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:ACTivate

Example of use

10 OUTPUT 717;":CALC1:MARK1:ACT"

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).BWIDth.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).BWIDth.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).BWIDth.DATA

Description

This command reads the bandwidth search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

If the bandwidth search is impossible, an error occurs when executed and the object is ignored.

In revision A.9.60 and above, you can select the trace and read the bandwidth search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 4-element array data (bandwidth search result).</p> <ul style="list-style-type: none"> <i>Data</i>(0) :The bandwidth. <i>Data</i>(1) :Center point frequency of the 2 cutoff frequency points. <i>Data</i>(2) :The Q value. <i>Data</i>(3) :Insertion loss <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

E5071C

Dim BandData As Variant

SCPI.CALCulate(1).PARAmeter(1).SElect

BandData = SCPI.CALCulate(1).SElected.MARKer(1).BWIDth.DATA

Related objects

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect

SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe

SCPI.CALCulate(Ch).SElected.MARKer(Mk).BWIDth.THReshold

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{[:SElected]:MARKer{[1]-10}:BWIDth:DATA?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:BWIDth:DATA?

Query Response

{numeric1},{numeric2},{numeric3},{numeric4},<newline><^END>

Example of use

10 OUTPUT 717;":CALC1:MARK1:BWID:DATA?"

20 ENTER 717;A,B,C,D

SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe = *Status**Status* = SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.BWIDth.STATe = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).MARKer.BWIDth.STATe

Description

This command turns ON/OFF the bandwidth search result display, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the bandwidth search result display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the bandwidth search result display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the bandwidth search result display. • False or OFF: Turns OFF the bandwidth search result display.
Preset value	False or OFF

Examples

E5071C

```
Dim BandSrch As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer.BWIDth.STATe = True
BandSrch = SCPI.CALCulate(1).SElected.MARKer.BWIDth.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).BWIDth.DATA
SCPI.CALCulate(Ch).SElected.MARKer(Mk).BWIDth.THReshold
```

Equivalent key

Marker Search > Bandwidth

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{[:SElected]:MARKer:BWIDth[:STATe] {ON|OFF|1|0}}
:CALCulate{[1]-160}{[:SElected]:MARKer:BWIDth[:STATe]}?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:BWIDth[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:BWIDth[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:BWID ON"
20 OUTPUT 717;":CALC1:MARK:BWID?"
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).BWIDth.THReshold

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).BWIDth.THReshold = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).BWIDth.THReshold

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).BWIDth.THReshold = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).BWIDth.THReshold

Description

This command sets/gets the bandwidth definition value (the value to define the pass-band of the filter) of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the bandwidth definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Bandwidth definition value (the value to define the pass band of the filter)
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	-3

Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> Log magnitude (MLOG): dB (decibel) Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) Group delay (GDEL): s (second) Others: No unit
Note	<p>If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim BandVal As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).BWIDth.THReshold = -6
BandVal = SCPI.CALCulate(1).SElected.MARKer(1).BWIDth.THReshold
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe
```

Equivalent key**Marker Search > Bandwidth Value****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:BWIDth: THReshold
<numeric>
```

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:BWIDth: THReshold?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BWIDth: THReshold <numeric>
```

```
:CALCulate{[1]-160}:TRACe{[1]-16}:BWIDth: THReshold?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:BWID:THR 6"  
20 OUTPUT 717;":CALC1:MARK1:BWID:THR?"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.MARKer.COUPle**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer.COUPle = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.MARKer.COUPle

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.COUPle = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.COUPle**Description**

This command turns ON/OFF the marker coupling between traces, for the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the marker coupling between traces for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the marker coupling between traces
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the marker coupling. • False or OFF: Turns OFF the marker coupling.
Preset value	True or ON

Examples

```
Dim MkrCpl As Boolean
SCPI.CALCulate(1).SElected.MARKer.COUPle = False
MkrCpl = SCPI.CALCulate(1).SElected.MARKer.COUPle
```

Equivalent key

Marker Fctn > Couple**Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer:COUPle {ON|OFF|1|0}
```

```
:CALCulate{[1]-160}{:SElected}:MARKer:COUPle?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:COUPle {ON|OFF|1|0}
```

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:COUP OFF"
20 OUTPUT 717;":CALC1:MARK:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).DATA**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).DATA**Description**

This command reads the response and stimulus value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

When the reference marker mode is ON ("True" is specified with the SCPI.CALCulate(*Ch*).SElected.MARKer.REFerence.STATe object), the readout value is the value relative to the reference marker.

In revision A.9.60 and above, you can select the trace and read the response and stimulus value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 3-element array data (response and stimulus of marker).</p> <ul style="list-style-type: none"> <i>Data</i>(0) :Response value (primary value) at the marker position. <i>Data</i>(1) :Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format. <i>Data</i>(2) :Stimulus value at the marker position. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

```
Dim MkrData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
MkrData = SCPI.CALCulate(1).SElected.MARKer(1).DATA
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe
SCPI.CALCulate(Ch).SElected.MARKer(Mk).X
SCPI.CALCulate(Ch).SElected.MARKer(Mk).Y
```

Equivalent key

Marker > **Marker 1|Marker 2|Marker 3|Marker 4|Ref Marker**

Marker > **More Markers** > **Marker 5|Marker 6|Marker 7|Marker 8|Marker 9**

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:DATA?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:DATA?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:DATA?"
30 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).DISCcrete

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).DISCcrete = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).DISCcrete

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).DISCcrete = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).DISCcrete

Description

This command turns ON/OFF the discrete mode (mode in which the marker moves only at the measurement points) with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the discrete mode with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the marker discrete mode
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the discrete mode. • False or OFF: Turns OFF the discrete mode.
Preset value	False or OFF

Examples

```
Dim MkrDsc As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
```



```
SCPI.CALCulate(1).SElected.MARKer(1).DISCcrete = True
MkrDsc = SCPI.CALCulate(1).SElected.MARKer(1).DISCcrete
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Marker Fctn > Discrete

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:DISCcrete
{ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:DISCcrete?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:DISCcrete
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:DISCcrete?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:DISC OFF"
20 OUTPUT 717;":CALC1:MARK1:DISC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer.FUNction.DOMain.COUPle

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer.FUNction.DOMain.COUPle = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.MARKer.FUNction.DOMain.COUPle

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNction.DOMain.COUPle = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNction.DOMain.COUPle

Description

This command sets/gets the coupling of the marker search range for all traces, for the selected channel (*Ch*).

In revision A.9.60 and above, even though you can select the trace and set/get the coupling of the marker search range for all traces just by executing the above TRACe(*Tr*) command, the *Tr* value is omitted. This is because the command is applied to all the traces.

Variable

Parameter	<i>Status</i>
Description	On/off of the trace coupling of the marker search range.
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Specifies the search range with the trace coupling. • False or OFF :Specifies the search range for each trace.
Preset value	True or ON

Examples

```
Dim TrCpl As Boolean
SCPI.CALCulate(1).SElected.MARKer.FUNction.DOMain.COUPle = False
TrCpl = SCPI.CALCulate(1).SElected.MARKer.FUNction.DOMain.COUPle
```

Related objects

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.EXECute

Equivalent key

Marker Search > Search Range > Couple

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}[:SElected]:MARKer:FUNction:DOMain:COUPle
{ON|OFF|1|0}
:CALCulate{[1]-160}[:SElected]:MARKer:FUNction:DOMain:COUPle?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNction:DOMain:COUPle
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNction:DOMain:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:COUP OFF"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.DOMain.MULTiple.RANGe

Object type

Property (**Read-Write**)

Syntax

```
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.RANGe = Param
```

Param =

```
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.RANGe
```

The following syntax is also supported:

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNction.DOMain.MULTiple.RANGe = Param
```

Param =

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNction.DOMain.MULTiple.RANGe
```

Description

This command sets/gets the target range of search analysis for the selected marker, for the selected channel (*Ch*) in revision A.9.60 and above.

Even though you can select the trace and set/get the target range of search analysis for all traces just by executing the above TRACe(*Tr*) command, the *Tr* value is omitted. This is because the command is applied to all the traces.

Variable

Parameter	<i>Param</i>
Description	Target search range of search analysis.
Data type	Integer
Range	1 to 16

Examples

```
Sub Multiple Range()
```

'Turn ON partial search feature and multiple search range

SCPI.CALCulate.TRACe(2).MARKer.FUNcTion.DOMain.STATe = True

SCPI.CALCulate.TRACe(2).MARKer.FUNcTion.DOMain.MULTiple.STATe = True

'Set the start and stop value for search range 3

SCPI.CALCulate.TRACe(2).MARKer(2).FUNcTion.DOMain.MULTiple.STARt(3) = 1000000000#

SCPI.CALCulate.TRACe(2).MARKer(2).FUNcTion.DOMain.MULTiple.STOP(3) = 3000000000#

'Assign the target range to the active marker

SCPI.CALCulate.TRACe(2).MARKer(2).ACTivate

SCPI.CALCulate.TRACe(2).MARKer(2).FUNcTion.DOMain.MULTiple.RANGe = 3

'Move marker to max within the search range

SCPI.CALCulate.TRACe(2).MARKer(2).FUNcTion.TYPE = "MAXimum"

SCPI.CALCulate.TRACe(2).MARKer(2).FUNcTion.EXECute

End Sub

Refer to Setting Multiple Range for the operations procedure to place multiple markers within a search range

Related objects

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNcTion.DOMain.MULTiple.STARt

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNcTion.DOMain.MULTiple.STOP

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNcTion.DOMain.MULTiple.STATe

Equivalent key

Marker Search > Search Range > Multiple Range > Target Range[1~16]

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNcTion:DOMAIN:MULTiple:RANGe <numeric>

:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNcTion:DOMAIN:MULTiple:RANGe?

The following syntax is also supported:

E5071C

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCTion:DOMain:MULTiple:RANGe <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCTion:DOMain:MULTiple:RANGe?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":CALC1:MARK2:FUNC:DOM:MULT:RANG 2"
20 OUTPUT 717;":CALC1:MARK2:FUNC:DOM:MULT:RANG?"
30 ENTER 717;A

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.DOMain.MULTiple.STARt(*Param*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.DOMain.MULTiple.STARt(*Param*) = *Value*

Value =

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.DOMain.MULTiple.STARt(*Param*)

The following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.DOMain.MULTiple.STARt(*Param*) = *Value*

Value =

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.DOMain.MULTiple.STARt(*Param*)

Description

This command sets/gets the start value of the target range in the multiple search range for the selected channel (*Ch*) in revision A.9.60 and above.

When the trace coupling is off, the active trace is the target to be set.

You can select the trace and set/get the start value of the target range in the multiple search range for the trace (when trace coupling is off) just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Target search range of search analysis.
Data type	Integer
Range	1 to 16

Parameter	Value
Description	Start value of the target range in the multiple search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

see the example in SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.RANGe

Related objects

SCPI.CALCULATE(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.RANGe

SCPI.CALCULATE(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.STATe

SCPI.CALCULATE(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.STOP

Equivalent key

Marker Search > Search Range > Multiple Range > Start

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:START
<numeric(Range)>,<numeric(Frequency)>
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:START? <numeric(Range)>
```

The following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:START
<numeric(Range)>,<numeric(Frequency)>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:START? <numeric(Range)>
```


Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STAR 2,1.7E9"  
20 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STAR? 2"  
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.STATe

Object type

Property (**Read-Write**)

Syntax

```
SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.STATe
= Status
```

Status =

```
SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.STATe
```

The following syntax is also supported:

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNction.DOMain.MULTiple.STATe
= Status
```

Status =

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNction.DOMain.MULTiple.STATe
```

Description

This command sets/gets the state of the multiple search range for the active trace of the selected channel (*Ch*) in revision A.9.60 and above.

You can select the trace and set/get the multiple search range for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	Selection of the analysis range
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Specifies an arbitrary range • False or OFF: Specifies the entire sweep range
Preset value	False or OFF

Examples

see the example in SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.RANGe

Related objects

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.ST
ARt

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.ST
OP

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.RA
NGe

Equivalent key

Marker Search > Search Range > Multiple Range > Multiple Range

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}{:SElected}:FUNction:DOMain:MULTiple[:STATe]
{ON|OFF|1|0}
```

```
:CALCulate{[1]-160}{:SElected}:FUNction:DOMain:MULTiple[:STATe]?
```

The following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNction:DOMain:MULTiple[:STATe]
{ON|OFF|1|0}
```

```
:CALCulate{[1]-160}:TRACe{[1]-
16}:FUNction:DOMain:MULTiple[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.DOMain.MULTiple.STOP(*Param*)

Object type

Property (**Read-Write**)

Syntax

```
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.STOP(Param) = Value
```

Value =

```
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.STOP(Param)
```

The following syntax is also supported:

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNction.DOMain.MULTiple.STOP(Param) = Value
```

Value =

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNction.DOMain.MULTiple.STOP(Param)
```

Description

This command sets/gets the stop value of the target range in the multiple search range, for the selected channel (*Ch*) in revision A.9.60 and above.

- When the trace coupling is off, the active trace is the target to be set.

You can select the trace and set/get the stop value of the target range in the multiple search range for the trace (when the trace coupling is off) just by executing the above TRACE(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAMeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Target search range of search analysis.
Data type	Integer
Range	1 to 16

Parameter	Value
Description	Stop value of the target range in the multiple search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

see the example in SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.DOMain.MULTiple.RANGe

Related objects

SCPI.CALCULATE(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.RANGe

SCPI.CALCULATE(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.STATe

SCPI.CALCULATE(Ch).SElected.MARKer.FUNction.DOMain.MULTiple.START

Equivalent key

Marker Search > Search Range > Multiple Range > Stop

Equivalent SCPI command**Syntax**

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:STOP
<numeric(Range)>,<numeric(Frequency)>
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:STOP? <numeric(Range)>
```

The following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:STOP
<numeric(Range)>,<numeric(Frequency)>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:DOMain:MULTiple:STOP? <numeric(Range)>
```

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STOP 2,1.8E9"  
20 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STOP? 2"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.START**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.START = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.START

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.START = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.START**Description**

This command sets/gets the start value of the marker search range, for the selected channel (*Ch*).

- When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the start value of the marker search range for the trace (when the trace coupling is off) just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	The start value of the search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

Dim SchStar As Double

SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.START = 1.7E9

SchStar = SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.START

Related objectsSCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.STOP

E5071C

SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STATe

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.EXECute

Equivalent key

Marker Search > **Search Range** > **Start**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:MARKer:FUNction:DOMain:STARt

<numeric>

:CALCulate{[1]-160}{:SElected}:MARKer:FUNction:DOMain:STARt?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNction:DOMain:STARt

<numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNction:DOMain:STARt?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STAR 1.7E9"

20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STAR?"

30 ENTER 717;A

SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STATe = *Status**Status* = SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNction.DOMain.STATe = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNction.DOMain.STATe

Description

This command sets/gets whether to use an arbitrary range when executing the marker search, for the selected channel (*Ch*).

- When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get whether to use an arbitrary range when executing the marker search for the trace (when the trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	Selects the search range.
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Specifies an arbitrary range. • False or OFF: Specifies the entire sweep range.
Preset value	False or OFF

Examples

E5071C

```
Dim SchRnge As Boolean
SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.START = 1.5E9
SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.STOP = 1.8E9
SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.STATe = True
SchRnge = SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.STATe
```

Related objects

```
SCPI.CALCulate(Ch).SElected.MARKer.FUNCtion.DOMain.START
SCPI.CALCulate(Ch).SElected.MARKer.FUNCtion.DOMain.STOP
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.EXECute
```

Equivalent key

Marker Search > Search Range > Search Range [ON/OFF]

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer:FUNCtion:DOMAIN[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:MARKer:FUNCtion:DOMAIN[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMAIN[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMAIN[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.STOP**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.MARKer.FUNCtion.DOMain.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STOP**Description**

This command sets/gets the stop value of the marker search range, for the selected channel (*Ch*).

- When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the stop value of the marker search range for the trace (when the trace coupling is off) just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Stop value of the search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

Dim SchStop As Double

SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.STOP = 1.8E9

SchStop = SCPI.CALCulate(1).SElected.MARKer.FUNCtion.DOMain.STOP

Related objects

E5071C

SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STARt

SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STATe

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.EXECute

Equivalent key

Marker Search > Search Range > Stop

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:MARKer:FUNction:DOMain:STOP
<numeric>

:CALCulate{[1]-160}{:SElected}:MARKer:FUNction:DOMain:STOP?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNction:DOMain:STOP
<numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNction:DOMain:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STOP 1.8E9"  
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STOP?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.EXECute

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.EXECute

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.EXECute

Description

This command executes search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

To specify the type of the search, use the SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction. TYPE object. (No read)

In revision A.9.60 and above, you can select the trace and execute search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Examples

```
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TYPE = "maximum"
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.EXECute
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE
SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STATE
```

Equivalent key

Marker Search > Max|Min**Marker Search > Peak > Search Peak|Search Left|Search Right****Marker Search > Target > Search Target|Search Left|Search Right**

When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI command

Syntax

All revision:

E5071C

:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:EXECute

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:EXECute

Example of use

10 OUTPUT 717;":CALC1:MARK1:FUNC:EXEC"

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PEXCursion

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PEXCursion = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PEXCursion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.PEXCursion = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value when executing the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*). Peak excursion value is the minimum value of the difference relative to the right and left adjacent measurement points.

In revision A.9.60 and above, you can select the trace and set/get the lower limit of peak excursion value when executing the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Lower limit of peak excursion value
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3

Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	<p>If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TYPE = "peak"
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.PEXCursion = 0.2
PeakExc = SCPI.CALCulate(1).SElected.MARKer(1).FUNction.PEXCursion
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.PPOLarity
```

Equivalent key

Marker Search > Peak > Peak Excursion

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction: PEXCursion
<numeric>
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction: PEXCursion?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:
PEXCursion <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:
PEXCursion?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:PEXC 0.2"  
20 OUTPUT 717;":CALC1:MARK1:FUNC:PEXC?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PPOLarity

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PPOLarity = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PPOLarity

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.PPOLarity = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.PPOLarity

Description

This command set/get the polarity of the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the polarity of the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Polarity for peak search
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "POSitive" Specifies the positive peak. • "NEGative" Specifies the negative peak. • "BOTH" Specifies both the positive peak and the negative peak.
Preset	"POSitive"

value	
--------------	--

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNctioN.TYPE = "peak"
SCPI.CALCulate(1).SElected.MARKer(1).FUNctioN.PPOLarity = "both"
PeakPol = SCPI.CALCulate(1).SElected.MARKer(1).FUNctioN.PPOLarity
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNctioN.TYPE
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNctioN.PEXCursion
```

Equivalent key**Marker Search > Peak > Peak Polarity****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNctioN:PPOLarity
{POSitive|NEGative|BOTH}
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNctioN:PPOLarity?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-
10}:FUNctioN:PPOLarity {POSitive|NEGative|BOTH}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-
10}:FUNctioN:PPOLarity?
```

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:PPOL NEG"
20 OUTPUT 717;":CALC1:MARK1:FUNC:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TARGet

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TARGet = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TARGet

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.TARGet = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.TARGet

Description

This command sets/gets the target value to be searched with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the target value to be searched with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Target value for target search
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0

Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	<p>If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TARGET = -12.5
TargVal = SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TARGET
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TTRansition
```

Equivalent key

Marker Search > Target > Target Value

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:TARGET
<numeric>
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:TARGET?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:TARGET
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:TARGET?
```

Query response

```
{numeric}<newline><^END>
```

E5071C

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TARG -12.5"  
20 OUTPUT 717;":CALC1:MARK1:FUNC:TARG?"  
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TRACKing

Object type

Property (**Read-Write**)

Syntax

All revision:

```
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TRACKing = Status
Status = SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TRACKing
```

In revision A.9.60 and above, the following syntax is also supported:

```
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNction.TRACKing = Status
Status = SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNction.TRACKing
```

Description

This command turns ON/OFF the search tracking (function to repeat search for each sweep) for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In case of multiple peak/target, use

```
SCPI.CALCulate(Ch).SElected.MARKer.FUNction.MULTi.TRACKing.
```

In revision A.9.60 and above, you can select the trace and turn ON/OFF the search tracking for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the marker search tracing
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the search tracking. • False or OFF: Turns OFF the search tracking.

Preset value	False or OFF
---------------------	--------------

Examples

```
Dim SrchTrac As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNctioN.TYPE = "targ"
SCPI.CALCulate(1).SElected.MARKer(1).FUNctioN.TRACKing = True
SrchTrac = SCPI.CALCulate(1).SElected.MARKer(1).FUNctioN.TRACKing
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNctioN.TYPE
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNctioN.EXECute
```

Equivalent key

Marker Search > Tracking

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNctioN:TRACking
{ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNctioN:TRACking?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNctioN:TRACking {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNctioN:TRACking?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TRAC ON"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TRAC?"
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TTRansition

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TTRansition = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TTRansition

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.TTRansition =
*Param**Param* =SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.TTRansition

Description

This command selects the transition type of the target search, for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and select the transition type of the target search, for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Transition type for search
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.

Preset value	"BOTH"
---------------------	--------

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TYPE = "targ"
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TTRansition = "neg"
TargTran = SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TTRansition
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TARGET
```

Equivalent key

Marker Search > Target > Target Transition

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:TTRansition
{POSitive| NEGative|BOTH}
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:TTRansition?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:TTRansition {POSitive| NEGative|BOTH}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNction:TTRansition?
```

Query response

```
{POS|NEG|BOTH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TTR NEG"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNction.TYPE

Description

This command selects the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and select the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Search type of marker
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "MAXimum": Sets the search type to the maximum value. • "MINimum": Sets the search type to the minimum value. • "PEAK": Sets the search type to the peak search. • "LPEak": Sets the search type to the peak search to the left from the marker position. • "RPEak": Sets the search type to the peak

	<p>search to the right from the marker position.</p> <ul style="list-style-type: none"> • "TARGet": Sets the search type to the target search. • "LTARget": Sets the search type to the target search to the left from the marker position. • "RTARget": Sets the search type to the target search to the right from the marker position.
Preset value	"MAXimum"

Examples

```
Dim SrchType As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TYPE = "targ"
SrchType = SCPI.CALCulate(1).SElected.MARKer(1).FUNction.TYPE
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.PEXCursion
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.PPOLarity
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TARGet
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TTRansition
SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.EXECute
```

Equivalent key

Marker Search > Max|Min

Marker Search > Peak > Search Peak|Search Left|Search Right

Marker Search > Target > Search Target|Search Left|Search Right

When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction
:TYPE {MAXimum|
MINimum|PEAK|LPEak|RPEak|TARGet|LTARget|RTARget}
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:FUNction:TYPE?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCTion
:TYPE {MAXimum|
MINimum|PEAK|LPEak|RPEak|TARGet|LTARget|RTARget}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCTion:TYPE?
```

Query response

```
{MAX|MIN|PEAK|LPE|RPE|TARG|LTAR|RTAR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TYPE PEAK"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.MARKer(Mk).NOTCh.DATA**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).NOTCh.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).NOTCh.DATA**Description**

This command reads the notch search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

If the notch search is impossible, an error occurs and the command is ignored. In this case, no query response is obtained.

In revision A.9.60 and above, you can select the trace and read the notch search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 4-element array data (notch bandwidth search result).</p> <ul style="list-style-type: none"> <i>Data</i>(0) :The bandwidth. <i>Data</i>(1) :Center point frequency of the 2 cutoff frequency points. <i>Data</i>(2) :The Q value. <i>Data</i>(3) :Insertion loss <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

```
Dim NotchData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
BandData = SCPI.CALCulate(1).SElected.MARKer(1).NOTCH.DATA
```

Related ObjectsSCPI.CALCulate(*Ch*).SElected.MARKer.NOTCh

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).NOTCh.THReshold

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SELEcted}:MARKer{[1]-10}:NOTCh:DATA?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:NOTCh:DATA?

Query response

{value 1},{value 2},{value 3},{value 4}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1 :NOTC :DATA?"
20 ENTER 717;A,B,C,D
```

SCPI.CALCulate(Ch).SElected.MARKer.NOTCh**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.MARKer.NOTCh = *Status**Status* = SCPI.CALCulate(Ch).SElected.MARKer.NOTCh

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.NOTCh = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).MARKer.NOTCh**Description**

This command turns ON/OFF the notch search result display, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the notch search result display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the notch search result display
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • ON or 1: Turns ON the notch search result display. • OFF or 0: Turns OFF the notch search result display.
Preset value	OFF or 0

Examples

```
Dim NotchSrch As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
```



```
SCPI.CALCulate(1).SElected.MARKer.NOTCH = True
NotchSrch = SCPI.CALCulate(1).SElected.MARKer.NOTCH
```

Related Objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer.NOTCh.DATA
SCPI.CALCulate(Ch).SElected.MARKer(Mk).NOTCh.THReshold
```

Equivalent key

Marker Search > Notch

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer:NOTCh[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:MARKer:NOTCh[:STATe]?
```

In revision A.9.60 and above:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:NOTCh[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:NOTCh[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK :NOTC ON"
20 OUTPUT 717;":CALC1:MARK :NOTC?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.MARKer(Mk).NOTCh.THReshold

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.MARKer(Mk).NOTCh.THReshold = *Value**Value* = SCPI.CALCulate(Ch).SElected.MARKer(Mk).NOTCh.THReshold

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).NOTCh.THReshold = *Value**Value* = SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).NOTCh.THReshold

Description

This command sets/gets the notch definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk:10*), for the active trace of selected channel (*Ch*), specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command.

In revision A.9.60 and above, you can select the trace and set/get the notch definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk:10*), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	Notch definition value
Range	-5E8 to 5E8
Preset Value	-3
Unit	Varies depending on the data format as follows: <ul style="list-style-type: none"> Amplitude (MLOG):dB (decibel) Phase (PHAS), Expanded phase (UPH),Positive phase (PPH): ° (degree) Group delay (GDEL): s (second)

- Others: No unit

- If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim NotchVal As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).NOTCh.THReshold = -6
NotchVal = SCPI.CALCulate(1).SElected.MARKer(1).NOTCh.THReshold
```

Related Objects

```
SCPI.CALCulate(Ch).SElected.MARKer.NOTCh
SCPI.CALCulate(Ch).SElected.MARKer.NOTCh.DATA
```

Equivalent key**Marker Search > Notch Value****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:NOTCh:THReshold
<value>
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:NOTCh:THReshold?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-
10}:NOTCh:THReshold <value>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-
10}:NOTCh:THReshold?
```

Query response

```
{value}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK :NOTC :THR 5"
20 OUTPUT 717;":CALC1:MARK :NOTC :THR?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.MARKer(Mk).SET

Object type

Property (**Write Only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).SET = *Param*

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).SET = *Param*

Description

This command sets the value at the position of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the value of the instrument setting item (*Param*), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set the value at the position of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the value of the instrument setting item (*Param*), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Instrument setting item
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "START": Sets the sweep start value to the stimulus value at the selected marker position. • "STOP": Sets the sweep stop value to the stimulus value at the selected marker position. • "CENTer": Sets the sweep center value to the stimulus value at the selected marker position. • "RLEVel": Sets the reference line value to the response value at the selected marker position. • "DELay": Sets the electrical delay time value

to the value of the group delay at the selected marker position (a value smoothed with the aperture of 20%).

- "RMARker": Sets the reference marker position to the stimulus value at the marker position.
- "SPAN": Sets the span at the between the selected marker and reference marker. If reference marker ($Mk=10$) is selected, span is set at 0.

Examples

```
Dim MkrTo As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).SET = "cent"
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe
```

Equivalent key

Marker Fctn > **Marker -> Start** | **Marker -> Stop** | **Marker -> Center** | **Marker -> Reference** | **Marker -> Delay**

- There are no equivalent keys for the parameters of "RMARker" and "SPAN".

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{[:SElected]:MARKer{[1]-10}:SET {START|STOP|CENTer|RLEVel|DELay|RMARker|SPAN}}
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:SET {START|STOP|CENTer|RLEVel|DELay|RMARker|SPAN}
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:SET CENT"
```

SCPI.CALCulate(Ch).SElected.MARKer(Mk).STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.MARKer(Mk).STATe = *Status**Status* = SCPI.CALCulate(Ch).SElected.MARKer(Mk).STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).STATe = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).STATe

Description

This command turns ON/OFF the display of marker 1 to 9 (*Mk*) and reference marker (*Mk:10*), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the display of marker 1 to 9 (*Mk*) and reference marker (*Mk:10*), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of display of markers 1 to 9 and reference marker
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the display of the marker. • False or OFF: Turns OFF the display of the marker.
Preset value	False or OFF

Examples

```
Dim Mkr As Boolean
SCPI.CALCulate(1).PARAmeter(2).SElect
SCPI.CALCulate(1).SElected.MARKer(10).STATe = True
Mkr = SCPI.CALCulate(1).SElected.MARKer(10).STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

When turning ON the display of the marker

Marker > **Marker 1|Marker 2|Marker 3|Marker 4|Ref Marker**

Marker > **More Markers** > **Marker 5|Marker 6|Marker 7|Marker 8|Marker 9**

When performing the operation from the front panel, a marker set to ON is automatically set to the *active marker*.

When turning OFF the display of the marker

Marker > **Clear Marker Menu** > **Marker 1|Marker 2|Marker 3|Marker 4|Marker 5|Marker 6|Marker 7|Marker 8|Marker 9|Ref Marker**

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}{:STATe} {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}{:STATe}?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}{:STATe}
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}{:STATe}?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1 ON"
20 OUTPUT 717;":CALC1:MARK1?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.MARKer(Mk).X

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.MARKer(Mk).X = *Value**Value* = SCPI.CALCulate(Ch).SElected.MARKer(Mk).X

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).X = *Value**Value* = SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).X

Description

This command set the stimulus value for marker 1 to 9 (*Mk*) and reference marker (*Ch:10*), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set the stimulus value for marker 1 to 9 (*Mk*) and reference marker (*Ch:10*), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	Stimulus value of the marker
Data type	Double precision floating point type (Double)
Range	Sweep start value to sweep stop value
Preset value	Sweep start value
Unit	Hz (hertz), dBm or s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

upper limit of the range is exceeded) is set.

Examples

```
Dim MkrX As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer(1).X = 1E9
MkrX = SCPI.CALCulate(1).SElected.MARKer(1).X
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe
SCPI.CALCulate(Ch).SElected.MARKer(Mk).Y
SCPI.CALCulate(Ch).SElected.MARKer(Mk).DATA
```

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

When performing the operation from the front panel, you turn ON the marker and set the stimulus value at the same time.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:X <numeric>
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:X?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:X <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:X?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:X 1E9"
20 OUTPUT 717;":CALC1:MARK1:X?"
30 ENTER 717:A
```

SCPI.CALCulate(Ch).SElected.MARKer(Mk).Y**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).Y

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).Y**Description**

This command reads the response value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

When the reference marker mode is ON ("True" is specified with the SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe object), the readout value is the value relative to the reference marker.

In revision A.9.60 and above, you can select the trace and read the response value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 2-element array data (response value of marker).</p> <ul style="list-style-type: none"> <i>Data</i>(0) :Response value (primary value) at the marker position. <i>Data</i>(1) :Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

```
Dim MkrY As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
MkrY = SCPI.CALCulate(1).SElected.MARKer(1).Y
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe
SCPI.CALCulate(Ch).SElected.MARKer(Mk).X
SCPI.CALCulate(Ch).SElected.MARKer(Mk).DATA
```

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

When performing the operation from the front panel, you turn ON the marker and read the response value on the display.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:MARKer{[1]-10}:Y?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:Y?
```

Query response

```
{numeric 1},{numeric 2}<newline><^END>
```

	Description
{numeric 1}	Response value (primary value) at the marker position.
{numeric 2}	Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format.

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:Y?"
30 ENTER 717;A,B
```

SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.MARKer.REFerence.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.MARKer.REFerence.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.REFerence.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.REFerence.STATe**Description**

This command turns ON/OFF the reference marker mode , for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the reference marker mode for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the reference marker mode
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the reference marker mode. • False or OFF: Turns OFF the reference marker mode.
Preset value	False or OFF

Examples

```
Dim RefMode As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MARKer.REFerence.STATe = True
RefMode = SCPI.CALCulate(1).SElected.MARKer.REFerence.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Marker > **Ref Marker Mode**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}[:SElected]:MARKer:REFerence[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}[:SElected]:MARKer:REFerence[:STATe]?
```

In revision A.9.60 and above:

```
:CALCulate{[1]-160}:TRACe(Tr):MARKer:REFerence[:STATe]
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe(Tr):MARKer:REFerence[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:REF ON"
20 OUTPUT 717;":CALC1:MARK:REF?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.MATH.FUNcTion

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MATH.FUNcTion = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.MATH.FUNcTion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MATH.FUNcTion = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MATH.FUNcTion

Description

This command sets/gets the data trace display method (math method between measurement data and memory trace data), for the active trace of selected channel (*Ch*).

- The math result according to this setting is displayed on the data trace.

In revision A.9.60 and above, you can select the trace and set/get the data trace display method for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Math method between measurement data and memory trace data
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "NORMal": Specifies <i>Data</i> (no math). • "DIVide": Specifies <i>Data</i> / <i>Mem</i>. • "MULTiply": Specifies <i>Data</i> x <i>Mem</i>. • "SUBTract": Specifies <i>Data</i> - <i>Mem</i>. • "ADD": Specifies <i>Data</i> + <i>Mem</i>. <p>Where <i>Data</i> is the measurement data (corrected data</p>

	array) and <i>Mem</i> is the data stored in the memory trace (corrected memory array).
Preset value	"NORMal"

Examples

```
Dim MathFunc As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MATH.FUNcTion = "div"
MathFunc = SCPI.CALCulate(1).SElected.MATH.FUNcTion
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Display > **Data Math** > **OFF|Data / Mem|Data * Mem|Data - Mem|Data + Mem**

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:MATH:FUNcTion {NORMal|
SUBTract|DIVide|ADD|MULTiply}
:CALCulate{[1]-160}{:SElected}:MATH:FUNcTion?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MATH:FUNcTion {NORMal|
SUBTract|DIVide|ADD|MULTiply}
:CALCulate{[1]-160}:TRACe{[1]-16}:MATH:FUNcTion?
```

Query response

```
{NORM|DIV|MULT|SUBT|ADD}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MATH:FUNC DIV"
20 OUTPUT 717;":CALC1:MATH:FUNC?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SElected.MATH.MEMorize**Object type**Method (**Write-only**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.MATH.MEMorize

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MATH.MEMorize**Description**

This command copies the measurement data to the memory trace, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and copy the measurement data to the memory trace, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Examples

```
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MATH.MEMorize
```

Related objectsSCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect**Equivalent key****Display > Data - > Mem****Equivalent SCPI command****Syntax**

All revision:

:CALCulate{[1]-160}{:SElected}:MATH:MEMorize

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MATH:MEMorize

Example of use

10 OUTPUT 717;":CALC1:MATH:MEM"

SCPI.CALCulate(*Ch*).SElected.MIXer.XAXis

Type of object

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MIXer.XAXis = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.MIXer.XAXis

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MIXer.XAXis = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MIXer.XAXis

Description

This command sets/gets the X-axis frequency when the frequency offset feature is off, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the X-axis frequency when the frequency offset feature is off for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Selects the X-axis frequency (frequency offset: off)
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "NORMal": Specifies the normal frequency. • "RFPLo": Specifies the frequency $RF + LO$. • "RFMLo": Specifies the frequency $RF - LO$. • "LOMRf": Specifies the frequency $LO - RF$.
Preset value	"NORMal"

Example of use

```
Dim Xaxis As String
SCPI.SENSE.OFFSet.STATe = False
```

E5071C

SCPI.CALCulate(1).SElected.MIXer.XAXis = "RFPLo"
Xaxis = SCPI.CALCulate(1).SElected.MIXer.XAXis

Related objects

SCPI.SENSE(Ch).OFFSet.STATe

SCPI.CALCulate(Ch).SElected.OFFset.XAXis

Equivalent key

Sweep Setup > **Frequency Offset** > **X-Axis** > **Normal|RF+LO|RF-LO|LO-RF**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:MIXer:XAXis {NORMal|RFPLo|
RFMLo|LOMRf}
:CALCulate{[1]-160}{:SElected}:MIXer:XAXis?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MIXer:XAXis {NORMal|RFPLo|
RFMLo|LOMRf}
:CALCulate{[1]-160}:TRACe{[1]-16}:MIXer:XAXis?

Query response

{NORM|RFPL|RFML|LOMR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MIX:XAX RFPL"  
20 OUTPUT 717;":CALC1:MIX:XAX?"  
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.MSTatistics.DATA**Object type**Property (**Read Only**)**Syntax**

All revision:

Data = SCPI.CALCulate(*Ch*).SElected.MSTatistics.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MSTatistics.DATA**Description**

This command reads the statistics values of the active trace of selected channel (*Ch*). The statistical values contains: mean value, standard deviation and the difference between the maximum value and the minimum value.

In revision A.9.60 and above, you can select the trace and read the statistics values of the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 3-element array data (statistics value).</p> <ul style="list-style-type: none"> <i>Data</i>(0) :Mean value <i>Data</i>(1) :Standard deviation <i>Data</i>(2) :Difference between the maximum value and the minimum value (Peak to Peak) <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

```
Dim MstData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
MstData = SCPI.CALCulate(1).SElected.MSTatistics.DATA
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MSTatistics.STATe
```

E5071C

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:MStatisticks:DATA?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MStatisticks:DATA?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

	Description
{numeric 1}	Mean value
{numeric 2}	Standard deviation
{numeric 3}	Difference between the maximum value and the minimum value (Peak to Peak)

Example of use

```
10 OUTPUT 717;":CALC1:MST:DATA?"
20 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).SElected.MSTatistics.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.MSTatistics.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.MSTatistics.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MSTatistics.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MSTatistics.STATe

Description

This command turns ON/OFF the statistics values display, for the active trace of selected channel (*Ch*). The statistical values contains: mean value, standard deviation and the difference between the maximum value and the minimum value.

In revision A.9.60 and above, you can select the trace and turn ON/OFF the statistics values display for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the statistics value display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the statistics value display. • False or OFF: Turns OFF the statistics value display.

Preset value	False or OFF
---------------------	--------------

Examples

```
Dim Mst As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.MSTatistics.STATe = True
Mst = SCPI.CALCulate(1).SElected.MSTatistics.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.MSTatistics.DATA
```

Equivalent key**Marker Fctn > Statistics****Equivalent SCPI command****Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected]:MSTatistics[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected]:MSTatistics[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:MSTatistics[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:MSTatistics[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MST ON"
20 OUTPUT 717;":CALC1:MST?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.OFFset.XAXis

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.OFFSet.XAXis = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.OFFSet.XAXis

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).OFFSet.XAXis = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).OFFSet.XAXis

Description

This command sets/gets the X-axis frequency when the frequency offset feature is on, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the X-axis frequency when the frequency offset feature is on for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	Selects the X-axis frequency (frequency offset: on)
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "BASE": Sets the normal frequency. • "STIMulus": Sets the frequency for the <i>source port</i> for the specified measurement parameter. • "RESPonse": Sets the frequency for the <i>receiver port for the specified measurement parameter</i>.
Preset value	"RESPonse"

Example of use

E5071C

```
Dim Xaxis As String
SCPI.SENSE.OFFSet.STATe = True
SCPI.CALCulate(1).SElected.OFFSet.XAXis = "STIMulus"
Xaxis = SCPI.CALCulate(1).SElected.OFFSet.XAXis
```

Related objects

```
SCPI.SENSE(Ch).OFFSet.STATe
SCPI.CALCulate(Ch).SElected.MIXer.XAXis
```

Equivalent key

Sweep Setup > Frequency Offset > X-Axis > Base|Stimulus|Response

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:OFFSet:XAXis {BASE|STIMulus|
RESPonse}
:CALCulate{[1]-160}{:SElected}:OFFSet:XAXis?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:OFFSet:XAXis {BASE|STIMulus|
RESPonse}
:CALCulate{[1]-160}:TRACe{[1]-16}:OFFSet:XAXis?
```

Query response

```
{BASE|STIMulus|RESPonse}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:OFFS:XAX RESP"
20 OUTPUT 717;":CALC1:OFFS:XAX?"
30 ENTER 717;A$
```


SCPI.CALCulate(*Ch*).SElected.RLIMit.DATA

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.RLIMit.DATA = *Data**Data* = SCPI.CALCulate(*Ch*).SElected.RLIMit.DATA

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DATA = *Data**Data* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DATA

Description

This command sets/gets the ripple limit table for the active trace (specified with the SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect command) of selected channel (*Ch*).

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.

In revision A.9.60 and above, you can select the trace and set/get the ripple limit table for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (for ripple line) of 1 + Num (number of limit lines) \ 4. Where n is an integer between 1 and Num.</p> <ul style="list-style-type: none"> <i>Data</i>(0) : The number of limit lines you want to set. Specify an integer ranging 0 to 12. When the number of limit lines is set to 0 (clears the limit table), the variable <i>Data</i> is only required with <i>Data</i>(0). <i>Data</i>(<i>n</i>×4-3) : The type of the <i>n</i>-th line. Specify an integer 0 to 1 as follows. 0: OFF 1: ON <i>Data</i>(<i>n</i>×4-2) : The value on the horizontal axis (frequency/power/time) of the start point of the <i>n</i>-

	<p>th line.</p> <ul style="list-style-type: none"> • <i>Data(n×4-1)</i> :The value on the horizontal axis (frequency/power/time) of the end point of the n-th line. • <i>Data(n×4)</i> :The ripple line value (dB) of the n-th line. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Note	<p>If there is no array data of 1+Num (number of set lines))*4 when setting a formatted memory array, an error occurs when executed and the object is ignored. For <i>Data(n \ 4-3)</i> in the array data, if you specify an integer other than 0 or 1, an error occurs when executed. For <i>Data(n*4-2)</i> and <i>Data(n*4-1)</i> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples (1)

```

Dim RLimData As Variant
SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.RLIMit.DATA = Array(1,1,1E6,1E9,0)
RLimData = SCPI.CALCulate(1).SElected.RLIMit.DATA

SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.RLIMit.DATA = Array(0) "" Clear Ripple Limit Table

```

Examples (2)

```

Dim RlimData(5) As Variant
Dim Ref As Variant
RLimData(0) = 1
RLimData(1) = 1
RLimData(2) = 1e6
RLimData(3) = 1e9
RLimData(4) = 0
SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.RLIMit.DATA = RLimData
Ref = SCPI.CALCulate(1).SElected.RLIMit.DATA

Dim RLimData(0) as Variant
RLimData(0) = 0
SCPI.CALCulate(1).PARAMeter(1).SElect
SCPI.CALCulate(1).SElected.RLIMit.DATA = RLimData "" Clear Ripple Limit Table

```

Related objects

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
 SCPI.CALCulate(Ch).SElected.RLIMit.STATe

Equivalent key

Analysis > Ripple Limit > Edit Ripple Limit > Add

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:RLIMit:DATA <numeric 1>,... ,<numeric 1+(N\ 4)>
```

```
:CALCulate{[1]-160}{:SElected}:RLIMit:DATA?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DATA <numeric 1>,... ,<numeric 1+(N\ 4)>
```

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DATA?
```

Query response

```
{numeric 1},... ,{numeric 1+(Nx4)}<newline><^END>
```

Example of use

```
10 DIM B(1:2,1:4)
20 OUTPUT 717;":CALC1:RLIM:DATA 2,1,1E9,3E9,3,1,5E9,7E9,3"
30 OUTPUT 717;":CALC1:RLIM:DATA?"
40 ENTER 717;A,B(*)
10 OUTPUT 717;":CALC1:RLIM:DATA 0" ! Clear Ripple Limit Table
```

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE = *Status**Status* = SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.LINE = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.LINE

Description

This command turns ON/OFF the ripple limit line display, for the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command) of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the ripple limit line display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF the ripple limit line display.
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the ripple limit line display. • False or OFF: Turns OFF the ripple limit line display.
Preset value	False or OFF

Examples

```
Dim RLimDisp As Boolean
SCPI.CALCulate(1).SElected.RLIMit.DISPlay.LINE = True
RLimDisp = SCPI.CALCulate(1).SElected.RLIMit.DISPLay.LINE
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.RLIMit.STATe
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SElect
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue
```

Equivalent key

Analysis > Ripple Limit > Ripple Limit

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:RLIMit:DISPlay:LINE {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:RLIMit:DISPlay:LINE?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:LINE {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:LINE?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:LINE ON"
20 OUTPUT 717;":CALC1:RLIM:DISP:LINE?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SELect**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SELect = *Value**Value* = SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SELect

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.SELect = *Value**Value* = SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.SELect**Description**

This command sets/gets the ripple limit band for ripple value display for selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the ripple limit band for ripple value display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SELect.

Variable

Parameter	<i>Value</i>
Description	The ripple limit band.
Data type	Long integer type (Long)
Range	1 to 12
Preset value	1

Examples

```
Dim RBand As Long
SCPI.CALCulate(1).SElected.RLIMit.DISPlay.SELect = 2
RBand = SCPI.CALCulate(1).SElected.RLIMit.DISPlay.SELect
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SELect
SCPI.CALCulate(Ch).SElected.RLIMit.STATe
```

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE
 SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue

Equivalent key

Analysis > Ripple Limit > Ripple Band

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:RLIMit:DISPlay:SElect <numeric>
:CALCulate{[1]-160}{:SElected}:RLIMit:DISPlay:SElect?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:SElect <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:SElect?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:SEL 5"
20 OUTPUT 717;":CALC1:RLIM:DISP:SEL?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue = *Param**Param* = SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.VALue = *Param**Param* = SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.VALue

Description

This command sets/gets the display type of ripple value for the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command) of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the display type of ripple value for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Param</i>
Description	The displaying type of ripple value.
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "OFF": Specifies the display off. • "ABSolute": Specifies the absolute value for display type. • "MARgin": Specifies the margin for display type.
Preset	"OFF"

value	
--------------	--

Examples

```
Dim RDisp As String
SCPI.CALCulate(1).SElected.RLIMit.DISPlay.VALue = "ABSolute"
RDisp = SCPI.CALCulate(1).SElected.RLIMit.DISPlay.VALue
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.RLIMit.STATe
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SElect
```

Equivalent key

Analysis > **Ripple Limit** > **Ripple Value** > **OFF|Absolute|Margin**

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:RLIMit:DISPlay:VALue {OFF|ABS|MAR}
:CALCulate{[1]-160}{:SElected}:RLIMit:DISPlay:VALue?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:VALue
{OFF|ABS|MAR}
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:VALue?
```

Query response

```
{OFF|ABS|MAR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:VAL ABS"
20 OUTPUT 717;":CALC1:RLIM:DISP:VAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(Ch).SElected.RLIMit.FAIL**Object type**Property (**Read Only**)**Syntax**

All revision:

Status = SCPI.CALCulate(Ch).SElected.RLIMit.FAIL

In revision A.9.60 and above, the following syntax is also supported:

Status = SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.FAIL**Description**

This command reads the ripple test result for the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command) of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the ripple test result for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	The ripple test result
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the ripple test result is FAIL. • False or OFF: Turns OFF the ripple test result is FAIL.
Note	When the ripple test is set to OFF, False or OFF is always read out.

Examples

Dim Result As Boolean
 Result = SCPI.CALCulate(1).SElected.RLIMit.FAIL

Related objects

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.RLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:RLIMit:FAIL?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:FAIL?

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.RLIMit.REPort.DATA**Object type**

Property

Syntax

All revision:

Data = SCPI.CALCulate(Ch).SElected.RLIMit.REPort.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.REPort.DATA**Description**

This command reads the ripple value of the ripple test for the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command) of selected channel (*Ch*).

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.(Read only)

In revision A.9.60 and above, you can select the trace and read the ripple value of the ripple test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (for ripple line) of 1 + Num (number of limit lines)*3. Where n is an integer between 1 and 12.</p> <ul style="list-style-type: none"> • <i>Data</i>(0) :Number of ripple limit line. • <i>Data</i>(<i>n</i>*3-2) :Number of ripple limit bands. • <i>Data</i>(<i>n</i>*3-1) :Ripple value. • <i>Data</i>(<i>n</i>*3) :Results of ripple test. <p>Select from the following. 0:PASS 1:FAIL.</p> <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Examples

```
Dim RData As Variant
SCPI.CALCulate(1).PARAmeter(1).SElect
RData = SCPI.CALCulate(1).SElected.RLIMit.REPort.DATA
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.RLIMit.STATe
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:RLIMit:REPort[:DATA]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:REPort[:DATA]?
```

Query response

```
{numeric 1},...,{numeric 1+Nx3}<newline><^END>
```

Type	Description
<numeric 1>	Number of ripple limit line (1 to 12)
<numeric 1+(n\ 3)-2>	Number of ripple limit bands
<numeric 1+(n\ 3)-1>	Ripple value
<numeric 1+(n\ 3)>	Results of ripple test 0: Pass 1: Fail

Where N is the number of lines (specified with <numeric 1>) and n is an integer between 1 and 12.

Example of use

```
10 DIM B(1:2,1:3)
20 OUTPUT 717;":CALC1:RLIM:REP?"
30 ENTER 717;A,B(*)
```

E5071C

SCPI.CALCulate(Ch).SElected.RLIMit.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SElected.RLIMit.STATe = *Status**Status* = SCPI.CALCulate(Ch).SElected.RLIMit.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.STATe = *Status**Status* = SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.STATe

Description

This command turns ON/OFF the ripple test function for the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect command) of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the ripple test function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF the ripple test function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the ripple test function. • False or OFF: Turns OFF the ripple test function.
Preset value	False or OFF

E5071C

Examples

```
Dim RLimTest As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.RLIMit.STATe = True
RLimTest = SCPI.CALCulate(1).SElected.RLIMit.STATe
```

Related Objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.RLIMit.DATA
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SElect
SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue
SCPI.CALCulate(Ch).SElected.RLIMit.FAIL
SCPI.CALCulate(Ch).SElected.RLIMit.REPort.DATA
```

Equivalent key

Analysis > Ripple Limit > Ripple Limit Test

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:RLIMit[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:RLIMit[:STATe]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:RLIM ON"
20 OUTPUT 717;":CALC1:RLIM?"
30 ENTER 717;A
```


SCPI.CALCulate(*Ch*).SElected.SMOothing.APERture

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.SMOothing.APERture = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.SMOothing.APERture

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.APERture = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.APERture

Description

This command sets/gets the smoothing aperture (percentage to the sweep span value) of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the smoothing aperture of the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Smoothing aperture
Data type	Double precision floating point type (Double)
Range	0.05 to 25
Preset value	1.5
Unit	% (percent)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

upper limit of the range is exceeded) is set.

Examples

```
Dim SmoAper As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.SMOothing.APERture = 2.5
SmoAper = SCPI.CALCulate(1).SElected.SMOothing.APERture
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.CALCulate(Ch).SElected.SMOothing.STATe
```

Equivalent key

Avg > **Smo Aperture**

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}[:SElected]:SMOothing:APERture <numeric>
:CALCulate{[1]-160}[:SElected]:SMOothing:APERture?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing:APERture <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing:APERture?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:SMO:APER 2.5"
20 OUTPUT 717;":CALC1:SMO:APER?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.SMOothing.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.SMOothing.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.SMOothing.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.STATe

Description

This command turns ON/OFF the smoothing, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the smoothing for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the smoothing
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the smoothing. • False or OFF: Turns OFF the smoothing.
Preset value	False or 0

Examples

```

Dim Smo As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.SMOothing.STATe = True
Smo = SCPI.CALCulate(1).SElected.SMOothing.STATe

```

E5071C

Related objects

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect

SCPI.CALCulate(Ch).SElected.SMOothing.APERture

Equivalent key

Avg > Smoothing

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}{:SElected}:SMOothing[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}{:SElected}:SMOothing[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SMO:STAT ON"
20 OUTPUT 717;":CALC1:SMO:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.CENter

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.CENter = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.CENter

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.CENter = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.CENter

Description

This command sets/gets the center value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the center value for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Center value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Cent As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.CENTer = 1E-8
Cent = SCPI.CALCulate(1).SElected.TRANSform.TIME.CENTer
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.SPAN
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Center

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:CENTer <numeric>
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:CENTer?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:CENTer
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:CENTer?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:CEN 1E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:CEN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.IMPulse.WIDTH

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.IMPulse.WIDTH = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.IMPulse.WIDTH

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.IMPulse.WIDTH = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.IMPulse.WIDTH

Description

This command sets/gets the shape of the Kayser Bessel window using the impulse width used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the Kayser Bessel window using the impulse width used for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Impulse width
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and transformation type.
Preset value	Varies depending on the frequency span and transformation type.
Unit	s (second)

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ImpWid As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.IMPulse.WIDTH = 1E-10
ImpWid = SCPI.CALCulate(1).SElected.TRANSform.TIME.IMPulse.WIDTH
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME. KBESsel
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STEP.RTIME
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATE
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Center

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:IMPulse:WIDTH
<numeric>
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:IMPulse:WIDTH?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:IMPulse:WIDTH
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:IMPulse:WIDTH?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:IMP:WIDT 1E-10"
20 OUTPUT 717;":CALC1:TRAN:TIME:IMP:WIDT?"
30 ENTER 717;A
```


SCPI.CALCulate(Ch).SElected.TRANSform.TIME.KBESsel**Object Type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.KBESsel = *Value**Value* = SCPI.CALCulate(Ch).SElected.TRANSform.TIME.KBESsel

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.KBESsel = *Value**Value* = SCPI.CALCulate(Ch).TRACe(Tr).TRANSform.TIME.KBESsel**Description**

This command sets/gets the shape of the Kayser Bessel window used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the Kayser Bessel window used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARAmeter(Tr).SElect.

Variable

Parameter	<i>Value</i>
Description	The value of b
Data type	Double precision floating point type (Double)
Range	0 to13
Preset value	6
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

E5071C

Dim Beta As Double

SCPI.CALCulate(1).PARAmeter(1).SElect

SCPI.CALCulate(1).SElected.TRANSform.TIME.KBESsel = 3

Beta = SCPI.CALCulate(1).SElected.TRANSform.TIME.KBESsel

Related objects

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.IMPulse.WIDTH

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STEP.RTIME

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATE

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect

Equivalent key

Analysis > Transform > Center

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SElected]:TRANSform:TIME:KBESsel <numeric>

:CALCulate{[1]-160}[:SElected]:TRANSform:TIME:KBESsel?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:KBESsel
<numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:KBESsel?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":CALC1:TRAN:TIME:KBES 3"

20 OUTPUT 717;":CALC1:TRAN:TIME:KBES?"

30 ENTER 717;A

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.LPFRequency**Object type**Method (**Write-only**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.LPFRequency = *Value*

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.LPFRequency = *Value***Description**

This command changes the frequency range to match with the low-pass type transformation of the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and change the frequency range to match with the low-pass type transformation of the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Examples

```
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.LPFRequency
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.TYPE
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key**Analysis > Transform > Set Freq Low pass****Equivalent SCPI command****Syntax**

All revision:

:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:LPFRequency

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:LPFRequency

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.SPAN**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.SPAN = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.SPAN

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.SPAN = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.SPAN**Description**

This command sets/gets the span value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the span value for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Span value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	2E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Span As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.SPAN = 1E-8
Cent = SCPI.CALCulate(1).SElected.TRANSform.TIME.SPAN
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.CENTer
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Center

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:SPAN <numeric>
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:SPAN?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:SPAN <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:SPAN?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:SPAN 1E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.START

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.START = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.START

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.START = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.START

Description

This command sets/gets the start value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the start value for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Start value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	-1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Star As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.START = 0
Star = SCPI.CALCulate(1).SElected.TRANSform.TIME.START
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STOP
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Start

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:START <numeric>
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:START?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:START <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:START?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STAR 0"
20 OUTPUT 717;":CALC1:TRAN:TIME:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STATe = *Status**Status* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STATe

Description

This command turns ON/OFF the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

You can enable the transformation function only when the sweep type is the linear sweep and the number of points is three or more. If you execute this object to try to enable the transformation function when the sweep type is other than the linear sweep or the number of points is less than three, an error occurs and the object is ignored.

- When the sweep type is the power sweep, you cannot turn on the transformation function. If you execute this object trying to turn on the transformation function during the power sweep, an error occurs and the object is ignored.

In revision A.9.60 and above, you can select the trace and turn ON/OFF the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the gating function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the transformation function.

	<ul style="list-style-type: none"> False or OFF: Turns OFF the transformation function.
Preset value	False or OFF

Examples

```
Dim Trans As Boolean
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.STATe = True
Trans = SCPI.CALCulate(1).SElected.TRANSform.TIME.STATe
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.SENSE(Ch).SWEep.TYPE
SCPI.SENSE(Ch).SWEep.POINTs
```

Equivalent key

Analysis > **Transform** > **Transform**

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:STATe {ON|OFF|1|0}
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:STATe?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STATe
{ON|OFF|1|0}
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STAT ON"
20 OUTPUT 717;":CALC1:TRAN:TIME:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STEP.RTIME

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STEP.RTIME = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STEP.RTIME

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STEP.RTIME = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STEP.RTIME

Description

This command sets/gets the shape of the Kayser Bessel window using the rise time of step signal used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the Kayser Bessel window using the rise time of step signal used for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	The rise time of step signal
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span.
Preset value	Varies depending on the frequency span.
Unit	s (second)

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim RTime As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.STEP.RTIME = 1E-10
RTime = SCPI.CALCulate(1).SElected.TRANSform.TIME.STEP.RTIME
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.IMPulse.WIDTH
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.KBESsel
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Center

Equivalent SCPI command**Syntax**

All revision:

```
:CALCulate{[1]-160}{:SElected]:TRANSform:TIME:STEP:RTIME
<numeric>
:CALCulate{[1]-160}{:SElected]:TRANSform:TIME:STEP:RTIME?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STEP:RTIME
<numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STEP:RTIME?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STEP:RTIM 1E-10"
20 OUTPUT 717;":CALC1:TRAN:TIME:STEP:RTIM?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STIMulus**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STIMulus = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STIMulus

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STIMulus = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STIMulus**Description**

This command sets/gets the stimulus type used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the stimulus type used for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	The stimulus type
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "IMPulse": Specifies the impulse • "STEP": Specifies the step
Preset value	"IMPulse"

Examples

```

Dim StimType As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.STIMulus = "step"
StimType = SCPI.CALCulate(1).SElected.TRANSform.TIME.STIMulus

```

Related objects

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.TYPE
 SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
 SCPI.CALCulate(Ch).PARAmeter(Tr).SElect

Equivalent key

Analysis > Transform > Type > Bandpass|Lowpass Step|Lowpass Imp.

- When performing this operation from the front panel, you select the transformation type at the same time.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:STIMulus
{IMPulse|STEP}
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:STIMulus?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STIMulus
{IMPulse|STEP}
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STIMulus?
```

Query response

```
{IMP|STEP}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME LPAS"
20 OUTPUT 717;":CALC1:TRAN:TIME:STIM STEP"
30 OUTPUT 717;":CALC1:TRAN:TIME:STIM?"
40 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STOP**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STOP = *Value**Value* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.STOP**Description**

This command sets/gets the stop value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the stop value used for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Value</i>
Description	Stop value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Span As Double
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.STOP = 2E-8
Cent = SCPI.CALCulate(1).SElected.TRANSform.TIME.STOP
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.START
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Stop

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:STOP <numeric>
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME:STOP?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STOP <numeric>
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STOP 2E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.TYPE**Object type**Property (**Read-Write**)**Syntax**

All revision:

SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).SElected.TRANSform.TIME.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.TYPE = *Param**Param* = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANSform.TIME.TYPE**Description**

This command sets/gets the transformation type used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the transformation type used for the transformation function of the time domain function, for the trace just by executing the above TRACe(*Tr*) command. You do not need to execute SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect.

Variable

Parameter	<i>Param</i>
Description	The transformation type
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "BPASs": Specifies the band-pass • "LPASs": Specifies the low-pass
Preset value	"BPASs"

Examples


```
Dim Typ As String
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.CALCulate(1).SElected.TRANSform.TIME.SHAPE = "lpas"
Typ = SCPI.CALCulate(1).SElected.TRANSform.TIME.SHAPE
```

Related objects

```
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STIMulus
SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STATe
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key

Analysis > Transform > Type > Bandpass|Lowpass Step|Lowpass Imp.

When performing this operation from the front panel, you select the stimulus type at the same time.

Equivalent SCPI command

Syntax

All revision:

```
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME[:TYPE] {BPASs|LPASs}
:CALCulate{[1]-160}{:SElected}:TRANSform:TIME[:TYPE]?
```

In revision A.9.60 and above, the following syntax is also supported:

```
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME[:TYPE]
{BPASs|LPASs}
:CALCulate{[1]-160}:TRACe{[1]-16}:TRANSform:TIME[:TYPE]?
```

Query response

```
{BPAS|LPAS}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME LPAS"
20 OUTPUT 717;":CALC1:TRAN:TIME?"
30 ENTER 717;A$
```

CONTROL**SCPI.CONTrol.HANDler.A.DATA****Object type**Method (**Write Only**)**Syntax**SCPI.CONTrol.HANDler.A.DATA = *Value***Description**

This command sets/gets information of output port A (A0 to A7) of the handler I/O. Port information is output as 8-bit binary data using A0 as LSB and A7 as MSB.

Variable

Parameter	<i>Value</i>
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

SCPI.CONTrol.HANDler.A.DATA = 15

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:HANDler:A[:DATA] <numeric>

Example of use

```
10 OUTPUT 717;":CONT:HAND:A 15"
20 OUTPUT 717;":CONT:HAND:A:DATA 15"
```

SCPI.CONTrol.HANDler.B.DATA**Object type**Method (**Write Only**)**Syntax**SCPI.CONTrol.HANDler.B.DATA = *Value***Description**

This command sets/gets information of output port B (B0 to B7) of the handler I/O. Port information is output as 8-bit binary data using B0 as LSB and B7 as MSB.

- The bit 6 of the data outputted by this project is ignored when outputting the INDEX signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.INDeX.STATe object).
- The bit 7 of the data outputted by this project is ignored when outputting the READY FOR TRIGGER signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe object).

Variable

Parameter	<i>Value</i>
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

SCPI.CONTrol.HANDler.B.DATA = 15

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:HANDler:B[:DATA] <numeric>

Example of use

E5071C

10 OUTPUT 717;":CONT:HAND:B 15"

20 OUTPUT 717;":CONT:HAND:B:DATA 15"

SCPI.CONTrol.HANDler.C.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.HANDler.C.DATA = *Value* (for output port)*Value* = SCPI.CONTrol.HANDler.C.DATA (for input port)**Description**

When input/output port C of the handler I/O is set to the output port, outputs port information to output port C (C0 to C3).

When input/output port C of the handler I/O is set to the input port, reads out port information inputted to port C (C0 to C3).

Port information is input/output as 4-bit binary data, using C0 as LSB and C3 as MSB.

Variable

Parameter	<i>Value</i>
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 15
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

SCPI.CONTrol.HANDler.C.MODE = "outp"

SCPI.CONTrol.HANDler.C.DATA = 8

Dim HdlCinp As Long

SCPI.CONTrol.HANDler.C.MODE = "inp"

HdlCinp = SCPI.CONTrol.HANDler.C.DATA

Related objects

SCPI.CONTrol.HANDler.C.MODE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

E5071C

:CONTRol:HANDler:C[:DATA] <numeric>
:CONTRol:HANDler:C[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"  
20 OUTPUT 717;":CONT:HAND:C 15"  
10 OUTPUT 717;":CONT:HAND:C:MODE INP"  
20 OUTPUT 717;":CONT:HAND:C?"  
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.C.MODE**Object type**

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.C.MODE = *Param*

Param = SCPI.CONTrol.HANDler.C.MODE

Description

This command sets/gets the input/output direction of port C of the handler I/O.

Variable

Parameter	<i>Param</i>
Description	Input/output direction of port C
Data type	Character string type (String)
Range	Select from the following: "INPut":Sets the port C to input. "OUTPut":Sets the port C to output.
Preset value	"INPut"

Examples

```
Dim HdlCmode As String
SCPI.CONTrol.HANDler.C.MODE = "OUTP"
HdlCmode = SCPI.CONTrol.HANDler.C.MODE
```

Related objects

SCPI.CONTrol.HANDler.C.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:CONTrol:HANDler:C:MODE {INPut|OUTPut}
:CONTrol:HANDler:C:MODE?
```

Query response

```
{INP|OUTP}<newline><^END>
```

E5071C

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"  
20 OUTPUT 717;":CONT:HAND:C:MODE?"  
30 ENTER 717;A$
```


SCPI.CONTrol.HANDler.D.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.HANDler.D.DATA = *Value* (for output port)*Value* = SCPI.CONTrol.HANDler.D.DATA (for input port)**Description**

When input/output port D of the handler I/O is set to the output port, outputs port information to output port D (D0 to D3).

When input/output port D of the handler I/O is set to the input port, reads out port information inputted to port D (D0 to D3).

Port information is output as 4-bit binary data using D0 as LSB and D3 as MSB.

Variable

Parameter	<i>Value</i>
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 15
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

SCPI.CONTrol.HANDler.D.MODE = "outp"

SCPI.CONTrol.HANDler.D.DATA = 8

Dim HdlDinp As Long

SCPI.CONTrol.HANDler.D.MODE = "inp"

HdlDinp = SCPI.CONTrol.HANDler.D.DATA

Related objects

SCPI.CONTrol.HANDler.D.MODE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

E5071C

:CONTRol:HANDler:D[:DATA] <numeric>
:CONTRol:HANDler:D[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:D:MODE OUTP"  
20 OUTPUT 717;":CONT:HAND:D 15"  
10 OUTPUT 717;":CONT:HAND:D:MODE INP"  
20 OUTPUT 717;":CONT:HAND:D?"  
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.D.MODE**Object type**

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.D.MODE = *Param*

Param = SCPI.CONTrol.HANDler.D.MODE

Description

This command sets/gets the input/output direction of port D of the handler I/O.

Variable

Parameter	<i>Param</i>
Description	Input/output direction of port D
Data type	Character string type (String)
Range	Select from the following: "INPut": Sets the port D to input. "OUTPut":Sets the port D to output.
Preset value	"INPut"

Examples

```
Dim HdlDmode As String
SCPI.CONTrol.HANDler.D.MODE = "OUTP"
HdlDmode = SCPI.CONTrol.HANDler.D.MODE
```

Related objects

SCPI.CONTrol.HANDler.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:HANDler:C:MODE {INPut|OUTPut}

:CONTrol:HANDler:C:MODE?

Query response

E5071C

{INP|OUTP}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:D:MODE OUTP"  
20 OUTPUT 717;":CONT:HAND:D:MODE?"  
30 ENTER 717;A$
```

SCPI.CONTrol.HANDler.E.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.HANDler.E.DATA = *Value* (for output)*Value* = SCPI.CONTrol.HANDler.E.DATA (for input port)**Description**

When input/output port E (port C + port D) of the handler I/O is set to the output port, outputs port information to output port E (C0 to D3).

When input/output port E of the handler I/O is set to the input port, reads out port information inputted to port E (C0 to D3).

Port information is output as 8-bit binary data using C0 as LSB and D3 as MSB.

Variable

Parameter	<i>Value</i>
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
SCPI.CONTrol.HANDler.C.MODE = "outp"
SCPI.CONTrol.HANDler.D.MODE = "outp"
SCPI.CONTrol.HANDler.E.DATA = 128

Dim HdlEinp As Long
SCPI.CONTrol.HANDler.C.MODE = "inp"
SCPI.CONTrol.HANDler.D.MODE = "inp"
HdlEinp = SCPI.CONTrol.HANDler.E.DATA
```

Related objects

```
SCPI.CONTrol.HANDler.C.MODE
SCPI.CONTrol.HANDler.D.MODE
SCPI.CONTrol.HANDler.C.DATA
```

E5071C

SCPI.CONTRol.HANDler.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTRol:HANDler:E[:DATA] <numeric>

:CONTRol:HANDler:E[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"  
20 OUTPUT 717;":CONT:HAND:D:MODE OUTP"  
30 OUTPUT 717;":CONT:HAND:E 128"  
  
10 OUTPUT 717;":CONT:HAND:C:MODE INP"  
20 OUTPUT 717;":CONT:HAND:D:MODE INP"  
30 OUTPUT 717;":CONT:HAND:E?"  
40 ENTER 717;A
```

SCPI.CONTrol.HANDler.EXTension.INDeX.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.HANDler.EXTension.INDeX.STATe = *Status**Status* = SCPI.CONTrol.HANDler.EXTension.INDeX.STATe**Description**

Turns ON/OFF output of the INDEX signal to B6 of the handler I/O.

- When you use port B6 as the output port, turn OFF the INDEX signal output. When outputting the INDEX signal is turned ON, the bit 6 of the data output by the SCPI.CONTrol.HANDler.B.DATA object (the bit 14 of the data outputted by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the INDEX signal output
Data type	Boolean type (Boolean)
Range	Select from the following: True or ON: Turns ON the INDEX signal output. False or OFF: Turns OFF the INDEX signal output.
Preset value	False or OFF

Examples

```
Dim Indx As Boolean
SCPI.CONTrol.HANDler.EXTension.INDeX.STATe = 1
Indx = SCPI.CONTrol.HANDler.EXTension.INDeX.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

E5071C

:CONTrol:HANDler[:EXTension]:INDex:STATe {ON|OFF|1|0}
:CONTrol:HANDler[:EXTension]:INDex:STATe?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":CONT:HAND:IND:STAT ON"
20 OUTPUT 717;":CONT:HAND:IND:STAT?"
30 ENTER 717;A

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = *Status**Status* = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe**Description**

This command turns ON/OFF the output of READY FOR TRIGGER signal to B7 of the handler I/O.

- When you use port B7 as the output port, turn OFF the READY FOR TRIGGER signal output. When outputting the READY FOR TRIGGER signal is turned ON, the bit 7 of the data output by the SCPI.CONTrol.HANDler.B.DATA object (the bit 15 of the data output by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the READY FOR TRIGGER signal output
Data type	Boolean type (Boolean)
Range	<p>Select from the following:</p> <p>True or ON: Turns ON the READY FOR TRIGGER signal output.</p> <p>False or OFF: Turns OFF the READY FOR TRIGGER signal output.</p>
Preset value	False or OFF

Examples

```
Dim RdyTrig As Boolean
SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = 0
RdyTrig = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

```
:CONTrol:HANDler[:EXTension]:RTRigger:STATe {ON|OFF|1|0}  
:CONTrol:HANDler[:EXTension]:RTRigger:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:RTR:STAT ON"  
20 OUTPUT 717;":CONT:HAND:RTR:STAT?"  
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.F.DATA**Object type**Method (**Write Only**)**Syntax**SCPI.CONTrol.HANDler.F.DATA = *Value***Description**

Outputs port information to output port F (port A + port B) of the handler I/O. Port information is output as 16-bit binary using A0 as LSB and B7 as MSB.

- The bit 14 of the data output by this project is ignored when outputting the INDEX signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.INDeX.STATe object).
- The bit 15 of the data output by this project is ignored when outputting the READY FOR TRIGGER signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe object).

Variable

Parameter	<i>Value</i>
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 65535
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

SCPI.CONTrol.HANDler.F.DATA = 511

Related objects

SCPI.CONTrol.HANDler.A.DATA

SCPI.CONTrol.HANDler.B.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

:CONTrol:HANDler:F[:DATA] <numeric>

Example of use

10 OUTPUT 717;":CONT:HAND:F 511"

SCPI.CONTrol.HANDler.OUTPUT(*Num*).DAT-**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.HANDler.OUTPUT(*Num*) = *Value**Value* = SCPI.CONTrol.HANDler.OUTPUT(*Num*)**Description**

This command sets/gets data to OUTPUT1 or OUTPUT2 of the handler I/O.

Variable

Parameter	<i>Num</i>
Description	Number of the OUTPUT terminal
Data type	Long integer type (Long)
Range	1 or 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Value</i>
Description	Polarity (High/Low)
Data type	Long integer type (Long)
Range	Select from the following: <ul style="list-style-type: none"> • 1: Specifies LOW. • 0: Specifies HIGH.

Examples

E5071C

```
Dim HdIPol As Long
SCPI.CONTRol.HANDler.OUTPUT(1).DATA = 1
HdIPol = SCPI.CONTRol.HANDler.OUTPUT(1).DATA
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CONTRol:HANDler:OUTPut{[1]|2}[:DATA] {1|0}
:CONTRol:HANDler:OUTPut{[1]|2}[:DATA]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:OUTP1 1"
20 OUTPUT 717;":CONT:HAND:OUTP1?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATA**Description**

This command sets/gets output port data for group A with ID of the multiport test set .

Variable

Parameter	<i>Value</i>
Description	Output port data for group A
Data type	Long integer type (Long)
Range	0 to 255
Note	This command is available only for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSESe.MULTiplexer(1).OUTPut.A.DATA = 150
Obj = SCPI.SENSESe.MULTiplexer(1).OUTPut.A.DATA
```

Related objectsSCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A[:DATA]?

Query response

{numeric}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:A 0"  
20 OUTPUT 717;":CONT:MULT1:OUTP:A?"  
30 ENTER 717;A
```


SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group A with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group A
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSE.MULTiplexer(1).OUTPut.A.VOLTage.DATA = 5
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.A.VOLTage.DATA
```

Related objectsSCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A:VOLTage[:DATA] <numeric>

E5071C

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:A:VOLT 2.3"  
20 OUTPUT 717;":CONT:MULT1:OUTP:A:VOLT?"  
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATA

Description

This command sets/gets the output port data for group B with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output port data for group B
Data type	Long integer type (Long)
Range	0 to 15
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.B.DATA = 12
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.B.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B[:DATA]?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:B 10"  
20 OUTPUT 717;":CONT:MULT1:OUTP:B?"  
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group B with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group B
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSE.MULTiplexer(1).OUTPut.B.VOLTage.DATA = 0
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.B.VOLTage.DATA
```

Related objectsSCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B:VOLTage[:DATA] <numeric>

E5071C

:CONTRol:MULTiplexer{[1]|2}:OUTPut:B:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:B:VOLT 0.5"  
20 OUTPUT 717;":CONT:MULT1:OUTP:B:VOLT?"  
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATA**Description**

This command sets/gets the output port data for group C with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output port data for group C
Data type	Long integer type (Long)
Range	0 to 15
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.C.DATA = 10
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.C.DATA
```

Related objectsSCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C[:DATA]?

Query response

{numeric}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:C 5"  
20 OUTPUT 717;":CONT:MULT1:OUTP:C?"  
30 ENTER 717;A
```


SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group C with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group C
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSE.MULTiplexer(1).OUTPut.C.VOLTage.DATA = 5.2
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.C.VOLTage.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA
 SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA
 SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:CONTrol:MULTiplexer{[1]|2}:OUTPut:C:VOLTage[:DATA] <numeric>
```

E5071C

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:C:VOLT 4.5"  
20 OUTPUT 717;":CONT:MULT1:OUTP:C:VOLT?"  
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA**Description**

This command sets/gets the output port data for group D with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output port data for group D
Data type	Long integer type (Long)
Range	0 to 15
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.D.DATA = 5
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.D.DATA
```

Related objectsSCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D[:DATA]?

Query response

{numeric}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:D 4"  
20 OUTPUT 717;":CONT:MULT1:OUTP:D?"  
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA = *Value**Value* = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group D with ID of the multiport test set.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group D
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

Dim Obj As Double

SCPI.SENSE.MULTiplexer(1).OUTPut.D.VOLTage.DATA = 5.2

Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.D.VOLTage.DATA

Related objectsSCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATASCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D:VOLTage[:DATA] <numeric>

E5071C

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:D:VOLT 1.2"  
20 OUTPUT 717;":CONT:MULT1:OUTP:D:VOLT?"  
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).PORT(Pt).SElect**Object type**

Property (**Write Only**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).PORT(*Pt*).SElect

Description

This command sets the multiport test set port.

Variable

Parameter	<i>Pt</i>
Description	Test set port
Data type	
Note	This command is available for E5092A multiport test set.

Examples

SCPI.CONTrol.MULTiplexer(1).PORT(1).SElect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:CONTrol:MULTiplexer{[1]|2}:PORT:SElect

Example of use

10 OUTPUT 717;":CONT:MULT1:PORT:SEL"

DISPLAY

SCPI.DISPlay.ANNotation.FREQuency.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.ANNotation.FREQuency.STATe = *Status*

Status = SCPI.DISPlay.ANNotation.FREQuency.STATe

Description

This command turns ON/OFF the frequency display on the LCD display.

Variable

Parameter	<i>Status</i>
Description	Sets/Gets ON/OFF state of the frequency display
Data type	Boolean type (Boolean)
Range	Select either of the following: <ul style="list-style-type: none">• True or ON: Turns ON the frequency display.• False or OFF: Turns OFF the frequency display.
Preset value	True or ON

Examples

Dim DispFreq As Boolean
SCPI.DISPlay.ANNotation.FREQuency.STATe = 0
DispFreq = SCPI.DISPlay.ANNotation.FREQuency.STATe

Equivalent key

Display > Frequency

Equivalent SCPI command

Syntax

:DISPlay:ANNotation:FREQuency[:STATe] {ON|OFF|1|0}
:DISPlay:ANNotation:FREQuency[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:ANN:FREQ OFF"  
20 OUTPUT 717;":DISP:ANN:FREQ?"  
30 ENTER 717;A
```

SCPI.DISPlay.CCLear

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.CCLear

Description

This command clears the error message displayed in the status bar (at the bottom of the LCD display).

Examples

SCPI.DISPlay.CCLear

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:CCLear

Example of use

10 OUTPUT 717;":DISP:CCL"

SCPI.DISPlay.CLOCK**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.CLOCK = *Status**Status* = SCPI.DISPlay.CLOCK**Description**

This command turns ON/OFF the clock display in the instrument status bar (at the right bottom of the LCD display).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the clock display
Data type	Boolean type (Boolean)
Range	Select either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the clock display. • False or OFF: Turns OFF the clock display.
Preset value	True or ON

Examples

```
Dim DispTime As Boolean
SCPI.DISPlay.CLOCK = ON
DispTime = SCPI.DISPlay.CLOCK
```

Equivalent key**System > Misc Setup > Clock Setup > Show Clock****Equivalent SCPI command****Syntax**

:DISPlay:CLOCK {ON|OFF|1|0}

:DISPlay:CLOCK?

Query response

E5071C

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:CLOC OFF"  
20 OUTPUT 717;":DISP:CLOC?"  
30 ENTER 717;A
```

SCPI.DISPlay.COLOr(*Dnum*).BACK

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLOr(*Dnum*).BACK = *Data**Data* = SCPI.DISPlay.COLOr(*Dnum*).BACK

Description

This command sets/gets the background color for normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	<i>Dnum</i>
Description	Select either of the following: 1: Normal display 2: Inverted display
Data type	Long integer type (Long)
Range	1 or 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when the command is executed.

Parameter	<i>Data</i>
Description	Indicates 3-element array data. <ul style="list-style-type: none"> • <i>Data</i>(0) : Sets amount of red. • <i>Data</i>(1) : Sets amount of green. • <i>Data</i>(2) : Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)

Range	<ul style="list-style-type: none">• <i>Data(0)</i> 0 to 5• <i>Data(1)</i> 0 to 5• <i>Data(2)</i> 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim BackColor As Variant
SCPI.DISPlay.COLOr(1).BACK = Array(1,2,3)
BackColor = SCPI.DISPlay.COLOr(1).BACK
```

Related objects

SCPI.DISPlay.COLOr(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Display Setup** > **Color Setup** > **Normal|Invert** > **Background**

Equivalent SCPI command

Syntax

```
:DISPlay:COLOr{[1]|2}:BACK <numeric 1>,<numeric 2>,<numeric 3>
:DISPlay:COLOr{[1]|2}:BACK?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:COL:BACK 1,2,3"
20 OUTPUT 717;":DISP:COL:BACK?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLOr(*Dnum*).GRATicule(*Gnum*)

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLOr(*Dnum*).GRATicule(*Gnum*) = *Data**Data* = SCPI.DISPlay.COLOr(*Dnum*).GRATicule(*Gnum*)

Description

This command sets/gets:

1. Color of the graticule label.
 2. Outer frame line of the graph (*Gnum*:1).
 3. Color of the grid line of the graph (*Gnum*:2).
- for the normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	<i>Gnum</i>
Description	The number of items: 1: The outer frame line of the graph 2: The color of the grid line of the graph
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Data</i>
Description	Indicates 3-element array data. <ul style="list-style-type: none"> • <i>Data</i>(0) : Sets amount of red. • <i>Data</i>(1) : Sets amount of green.

	<ul style="list-style-type: none"> • <i>Data(2)</i> : Sets amount of blue. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<ul style="list-style-type: none"> • <i>Data(0)</i> : 0 to 5 • <i>Data(1)</i> : 0 to 5 • <i>Data(2)</i> : 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim GritColor As Variant
SCPI.DISPlay.COLOr(1).GRATicule(1) = Array(1,2,3)
GritColor = SCPI.DISPlay.COLOr(1).GRATicule(1)
```

Related objects

```
SCPI.DISPlay.COLOr(Dnum).RESet
```

Equivalent key

System > **Misc Setup** > **Display Setup** > **Color Setup** > **Normal|Invert** > **Graticule Main|Graticule Sub**

Equivalent SCPI command**Syntax**

```
:DISPlay:COLOr{[1]|2}:GRATicule{[1]|2} <numeric 1>,<numeric 2>,<numeric 3>
:DISPlay:COLOr{[1]|2}:GRATicule{[1]|2}?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:COL1:GRAT1 1,2,3"
20 OUTPUT 717;":DISP:COL1:GRAT1?"
30 ENTER 717;A,B,C
```


SCPI.DISPlay.COLOr(*Dnum*).LIMit(*Lnum*)

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLOr(*Dnum*).LIMit(*Lnum*) = *Data**Data* = SCPI.DISPlay.COLOr(*Dnum*).LIMit(*Lnum*)

Description

This command sets/gets:

- Fail display color used for the limit test result, Bandwidth test result and Ripple test result (*Lnum*:1)
- Color of the limit line (*Lnum*:2)
for normal display (*Dnum*:1) and inverted display (*Dnum*: 2).

Variable

Parameter	<i>Lnum</i>
Description	The number of item 1: The limit test result (Fail/Pass) 2: The limit line
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Data</i>
Description	Indicates 3-element array data. <ul style="list-style-type: none"> • <i>Data</i>(0) : Sets amount of red. • <i>Data</i>(1) : Sets amount of green.

	<ul style="list-style-type: none"> <i>Data(2)</i> : Sets amount of blue. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<ul style="list-style-type: none"> <i>Data(0)</i> : 0 to 5 <i>Data(1)</i> : 0 to 5 <i>Data(2)</i> : 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim LimColor As Variant
SCPI.DISPlay.COLOr(1).LIMit(1) = Array(1,2,3)
LimColor = SCPI.DISPlay.COLOr(1).LIMit(1)
```

Related objects

```
SCPI.DISPlay.COLOr(Dnum).RESet
```

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Limit Fail|Limit Line**

Equivalent SCPI command**Syntax**

```
:DISPlay:COLOr{[1]|2}:LIMit{[1]|2} <numeric 1>,<numeric 2>,<numeric 3>
```

```
:DISPlay:COLOr{[1]|2}:LIMit{[1]|2}?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:COL1:LIM1 1,2,3"
20 OUTPUT 717;":DISP:COL1:LIM1?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLOr(*Dnum*).RESet**Object type**Method (**Write-only**)**Syntax**SCPI.DISPlay.COLOr(*Dnum*).RESet**Description**

This command resets the display color settings for all the items to the factory preset state, for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Examples

SCPI.DISPlay.COLOr(1).RESet

Related objectsSCPI.DISPlay.COLOr(*Dnum*).BACKSCPI.DISPlay.COLOr(*Dnum*).GRATicule(*Gnum*)SCPI.DISPlay.COLOr(*Dnum*).LIMit(*Lnum*)SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).DATASCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).MEMory**Equivalent key****System > Misc Setup > Color Setup > Normal|Invert > Reset Color > OK****Equivalent SCPI command****Syntax**

:DISPlay:COLOr{[1]|2}:RESet

Example of use

10 OUTPUT 717;":DISP:COL1:RES"

SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).DATA = *Data**Data* = SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).DATA

Description

This command sets/gets the color of the data trace of traces 1 to 36 (*Tr*), for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 3-element array data.</p> <ul style="list-style-type: none"> • <i>Data</i>(0) :Sets amount of red. • <i>Data</i>(1) :Sets amount of green. • <i>Data</i>(2) :Sets amount of blue. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<ul style="list-style-type: none"> • <i>Data</i>(0) : 0 to 5 • <i>Data</i>(1) : 0 to 5 • <i>Data</i>(2) : 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TrColor As Variant
SCPI.DISPlay.COLOr(1).TRACe(1).DATA = Array(1,2,3)
TrColor = SCPI.DISPlay.COLOr(1).TRACe(1).DATA
```

Related objects

SCPI.DISPlay.COLOr(*Dnum*).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Data Trace 1|Data Trace 2|Data Trace 3|Data Trace 4|Data Trace 5|Data Trace 6|Data Trace 7|Data Trace 8|Data Trace 9**

Equivalent SCPI command

Syntax

```
:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:DATA <numeric 1>,<numeric 2>,<numeric 3>
```

```
:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:DATA?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:COL1:TRAC1:DATA 1,2,3"
20 OUTPUT 717;":DISP:COL1:TRAC1:DATA?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).MEMory

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).MEMory = *Data**Data* = SCPI.DISPlay.COLOr(*Dnum*).TRACe(*Tr*).MEMory

Description

This command sets/gets the color of the memory trace of traces 1 to 36 (*Tr*), for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 3-element array data.</p> <ul style="list-style-type: none"> • <i>Data</i>(0) :Sets amount of red. • <i>Data</i>(1) :Sets amount of green. • <i>Data</i>(2) :Sets amount of blue. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<ul style="list-style-type: none"> • <i>Data</i>(0) :0 to 5 • <i>Data</i>(1) :0 to 5 • <i>Data</i>(2) :0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TrColor As Variant
SCPI.DISPlay.COLOr(1).TRACe(1).MEMory = Array(1,2,3)
TrColor = SCPI.DISPlay.COLOr(1).TRACe(1).MEMory
```

Related objects

SCPI.DISPlay.COLOr(*Dnum*).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Mem Trace 1|Mem Trace 2|**
Mem Trace 3|Mem Trace 4|Mem Trace 5|Mem Trace 6|Mem Trace 7|Mem Trace 8|Mem Trace 9

Equivalent SCPI command

Syntax

```
:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:MEMory <numeric 1>,<numeric 2>,<numeric 3>
```

```
:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:MEMory?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:COL1:TRAC1:MEM 1,2,3"
20 OUTPUT 717;":DISP:COL1:TRAC1:MEM?"
30 ENTER 717;A,B,C
```

E5071C

SCPI.DISPlay.ECHO.CLEAr

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.ECHO.CLEAr

Description

This command clears all character strings displayed in the echo window.

Examples

SCPI.DISPlay.ECHO.CLEAr

Related objects

SCPI.DISPlay.ECHO.DATA

Equivalent key

Macro Setup > Clear Echo

Equivalent SCPI command

Syntax

:DISPlay:ECHO:CLEAr

Example of use

10 OUTPUT 717;":DISP:ECHO:CLE"

SCPI.DISPlay.ECHO.DATA**Object type**Property (**Write Only**)**Syntax**SCPI.DISPlay.ECHO.DATA = *Cont***Description**

This command displays a character string in the echo window. This command is different from ECHO command as it displays a single character string.

Variable

Parameter	<i>Cont</i>
Description	String you want to display in the echo window.
Data type	Character string type (String)
Range	254 characters or less

Examples

```
SCPI.DISPlay.ECHO.DATA = "Test Result"
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
```

Related objects

ECHO

SCPI.DISPlay.TABLe.TYPE

SCPI.DISPlay.TABLe.STATe

SCPI.DISPlay.ECHO.CLEAr

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:DISPlay:ECHO[:DATA] <string>

Example of use

```
10 OUTPUT 717;":DISP:ECHO ""TEST RESULT""
```

SCPI.DISPlay.ENABLE

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.ENABLE = *Status**Status* = SCPI.DISPlay.ENABLE

Description

This command turns ON/OFF the display update on the E5071C measurement screen.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the display update of the E5071C measurement screen
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the display update. • False or OFF : Turns OFF the display update.
Preset value	True or ON

Examples

```
Dim DispUpdt As Boolean
SCPI.DISPlay.ENABLE = False
DispUpdt = SCPI.DISPlay.ENABLE
```

Equivalent key

Display > Update

Equivalent SCPI command

Syntax

:DISPlay:ENABLE {ON|OFF|1|0}

:DISPlay:ENABLE?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:ENAB OFF"  
20 OUTPUT 717;":DISP:ENAB?"  
30 ENTER 717;A
```

SCPI.DISPlay.FSIGN**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.FSIGN = *Status**Status* = SCPI.DISPlay.FSIGN**Description**

This command turns ON/OFF the "Fail" display on the LCD screen when the limit test, bandwidth test and ripple test fails.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the "Fail" display when the limit test fails
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the "Fail" display. • False or OFF : Turns OFF the "Fail" display.
Preset value	True or ON

On/off of the Fail display cannot be set at each test. When the Fail display of either of test is turned on, the Fail display of other tests turns on, too.

Examples

```
Dim DispFail As Boolean
SCPI.DISPlay.FSIGN = False
DispFail = SCPI.DISPlay.FSIGN
```

Related objects

```
SCPI.CALCulate(Ch).SElected.LIMit.STATe
SCPI.CALCulate(Ch).SElected.RLIMit.STATe
SCPI.CALCulate(Ch).SElected.BLIMit.STATe
SCPI.CALCulate(Ch).SElected.PLIMit.STATe
```

Equivalent key

Analysis > **Limit Test** > **Fail Sign**

Analysis > Ripple Limit > Fail Sign

Analysis > Bandwidth limit > Fail Sign

Analysis > Point Limit Test > Fail Sign

Equivalent SCPI command

Syntax

:DISPlay:FSIGn {ON|OFF|1|0}

:DISPlay:FSIGn?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:FSIG OFF"  
20 OUTPUT 717;":DISP:FSIG?"  
30 ENTER 717;A
```

SCPI.DISPlay.IMAGe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.IMAGe = *Param**Param* = SCPI.DISPlay.IMAGe

Description

This command sets/gets the display type of the LCD display.

Variable

Parameter	<i>Param</i>
Description	Display type of the LCD display
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "NORMal" : Specifies the normal display (background color: black). • "INVert": Specifies the display in which the color of the normal display is inversed (background color: white).
Preset value	"NORMal"

Examples

```
Dim Displmg As String
SCPI.DISPlay.IMAGe = "inv"
Displmg = SCPI.DISPlay.IMAGe
```

Equivalent key

Display > Invert Color

Equivalent SCPI command

Syntax

```
:DISPlay:IMAGe {NORMal|INVert}
:DISPlay:IMAGe?
```

Query response

{NORM|INV}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:IMAG INV"  
20 OUTPUT 717;":DISP:IMAG?"  
30 ENTER 717;A$
```

SCPI.DISPlay.MAXimize**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.MAXimize = *Status**Status* = SCPI.DISPlay.MAXimize**Description**

This command turns ON/OFF the window maximization of the active channel.

If the maximization is set to ON, only the window of the active channel is maximized on the LCD display and the windows of the other channels are not displayed.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the window maximization
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the window maximization. • False or OFF: Turns OFF the window maximization.
Preset value	False or OFF

Examples

```
Dim ChMax As Boolean
SCPI.DISPlay.SPLit = "d1_2"
SCPI.DISPlay.WINDow(2).ACTivate
SCPI.DISPlay.MAXimize = True
ChMax = SCPI.DISPlay.MAXimize
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key**Channel Max****Equivalent SCPI command**

Syntax

:DISPlay:MAXimize {ON|OFF|1|0}

:DISPlay:MAXimize?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":DISP:MAX ON"

20 OUTPUT 717;":DISP:MAX?"

30 ENTER 717;A

SCPI.DISPlay.SKEY.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.SKEY.STATe = *Status**Status* = SCPI.DISPlay.SKEY.STATe**Description**

This command turns ON/OFF the display of the softkey menu bar.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the softkey menu bar display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none">• True or ON: Turns ON the softkey menu bar display.• False or OFF: Turns OFF the softkey menu bar display.
Preset value	True or ON

Examples

```
Dim DispSKey As Boolean
SCPI.DISPlay.SKEY.STATe = False
DispSKey = SCPI.DISPlay.SKEY.STATe
```

Equivalent key**Entry Off****Equivalent SCPI command****Syntax**

:DISPlay:SKEY[:STATe] {ON|OFF|1|0}

:DISPlay:SKEY[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:MAX ON"  
20 OUTPUT 717;":DISP:MAX?"  
30 ENTER 717;A
```

SCPI.DISPlay.SPLit

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.SPLit = *Param**Param* = SCPI.DISPlay.SPLit

Description

Sets the layout of the channel windows on the LCD display.

Variable

Parameter	<i>Param</i>
Description	Layout of channel windows
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "D1": See Channel graph window layouts. • "D12": See Channel graph window layouts. • "D1_2": See Channel graph window layouts. • "D112": See Channel graph window layouts. • "D1_1_2": See Channel graph window layouts. • "D123": See Channel graph window layouts. • "D1_2_3": See Channel graph window layouts. • "D12_33": See Channel graph window layouts. • "D11_23": See Channel graph window layouts. • "D13_23": See Channel graph window layouts. • "D12_13": See Channel graph window layouts. • "D1234": See Channel graph window

layouts.

- "D1_2_3_4": See Channel graph window layouts.
- "D12_34": See Channel graph window layouts.
- "D123_456": See Channel graph window layouts.
- "D12_34_56": See Channel graph window layouts.
- "D1234_5678": See Channel graph window layouts.
- "D12_34_56_78": See Channel graph window layouts.
- "D123_456_789": See Channel graph window layouts.
- "D123__ABC": See Channel graph window layouts.
- "D1234__9ABC": See Channel graph window layouts.
- "D1234__CDEF": See Channel graph window layouts.
- "D1234__DEFG": See Channel graph window layouts.
- "D1X1"
- "D1X2"
- "D1X3"
- "D1X4"
- "D2X1"
- "D2X2"
- "D2X3"
- "D2X4"
- "D3X1"
- "D3X2"
- "D3X3"
- "D3X4"

Preset value	<ul style="list-style-type: none"> • "D4X1" • "D4X2" • "D4X3" • "D4X4" • "D4X5" • "D4X6" • "D4X7" • "D4X8" • "D4X9" • "D8X9" • "D6X12" • "D8X12" • "D10X16"
	"D1"

- From Firmware revision A.08.10, channel display configuration of 8x9, 6x12 and 8x12 is added.

Examples

```
Dim ChanAloc As String
SCPI.DISPlay.SPLit = "d12_34"
ChanAloc = SCPI.DISPlay.SPLit
```

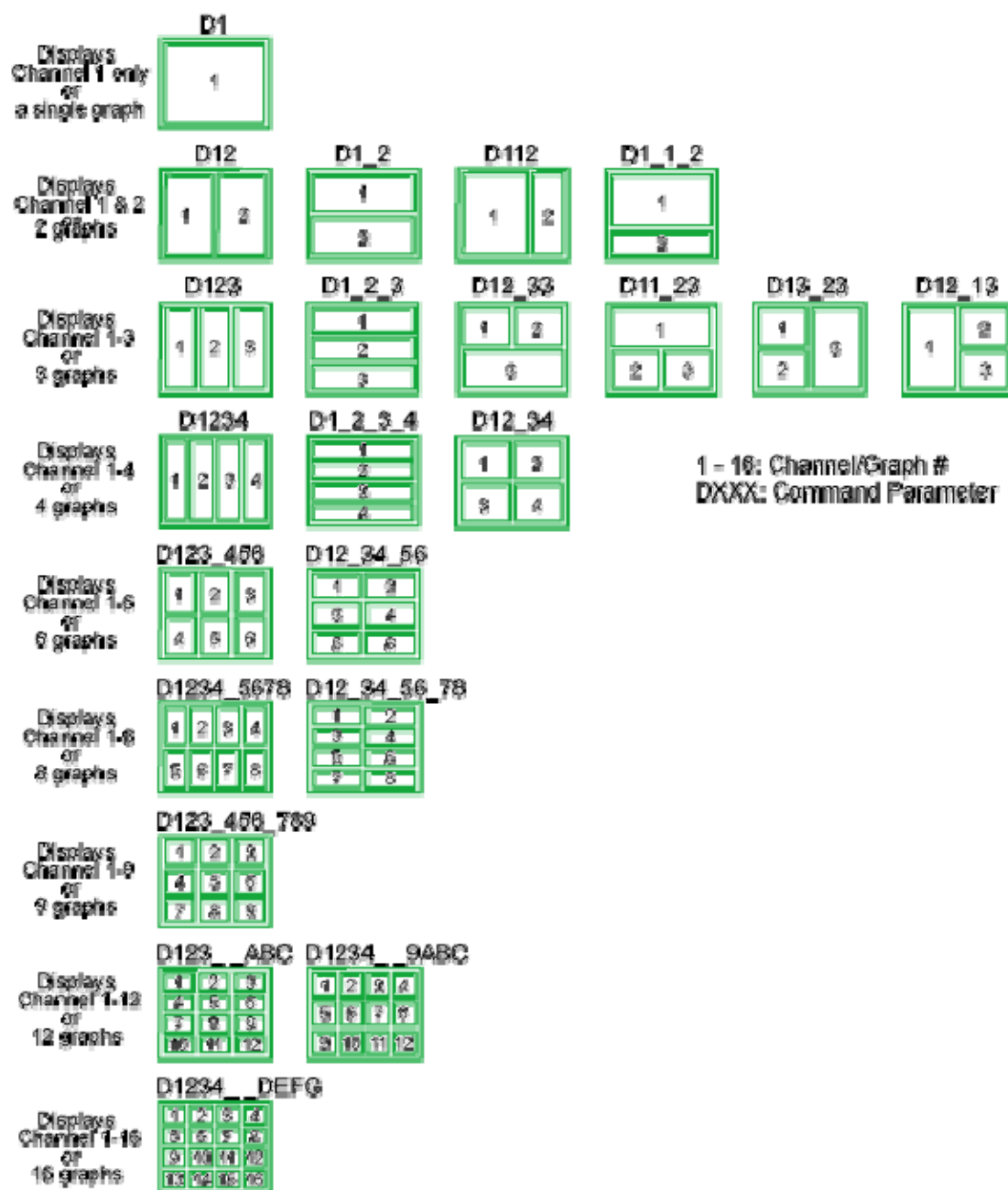
Related objects

SCPI.DISPlay.WINDow(Ch).SPLit

Equivalent key

Display > Allocate Channels

Channel/graph window layouts



e5071c314

Equivalent SCPI command

Syntax

:DISPlay:SPLit

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|D1234|D1_2_3_4|D12_34|D123_456|D12_34_56|D1234_5678|D12_34_56_78|D123_456_789|

E5071C

```
D123__ABC|D1234__9ABC|D1234__CDEF|D1234__DEFG|D1X1|D1X2|D1
X3|D1X4|D2X1|D2X2|D2X3|D2X4|D3X1|
D3X2|D3X3|D3X4|D4X1|D4X2|D4X3|D4X4|D4X5|D4X6|D4X7|D4X8|D4X9
}
```

:DISPlay:SPLit?

Query response

```
{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D1
2_13|
D1234|D1_2_3_4|D12_34|D123_456|D12_34_56|D1234_5678|D12_34_5
6_78|D123_456_789|
D123__ABC|D1234__9ABC|D1234__CDEF|D1234__DEFG|D1X1|D1X2|D1
X3|D1X4|D2X1|D2X2|D2X3|D2X4|D3X1|
D3X2|D3X3|D3X4|D4X1|D4X2|D4X3|D4X4|D4X5|D4X6|D4X7|D4X8|D4X9
}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:SPL D1_2"
20 OUTPUT 717;":DISP:SPL?"
30 ENTER 717;A$
```


SCPI.DISPlay.TABLE.POSition.RECTangle**Object type**Property (**Read Only**)**Syntax***Param* = SCPI.DISPlay.TABLE.POSition.RECTangle**Description**

This command reads the display coordinates position of Table area (the top left of the display is [0, 0]). If SCPI.DISPlay.TABLE.STATE is OFF, 0, 0, 0, 0 will be returned. .

Variable

Parameter	<i>Param</i>
Description	<p>Indicates the coordinates position of Table Area.</p> <ul style="list-style-type: none"> <i>Param(0)</i> : coordinates X position of top left of Table Area. <i>Param(1)</i> : coordinates Y position of top left of Table Area. <i>Param(2)</i> : coordinates X position of bottom right of Table Area. <i>Param(3)</i> : coordinates Y position of bottom right of Table Area. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Resolution	1

Examples

```
Dim TablePos() As Variant
SCPI.DISPlay.TABLE.STATE = True
TablePos = SCPI.DISPlay.TABLE.POSition.RECTangle
```

Related objects

SCPI.DISPlay.TABLE.STATE

Equivalent key

None

Equivalent SCPI command

E5071C

Syntax

:DISPlay:TABLE:POSition[:RECTangle]?

Query response

{numeric1},{numeric2},{numeric3},{numeric4},<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL:STAT ON"  
20 OUTPUT 717;":DISP:TABL:POS?"  
20 ENTER 717;A, B, C, D
```

SCPI.DISPlay.TABLE.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.TABLE.STATe = *Status**Status* = SCPI.DISPlay.TABLE.STATe

Description

This command turns ON/OFF the display of the window that appears in the lower part of the LCD display (specified by SCPI.DISPlay.TABLE.TYPE object).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the display of the window that appears in the lower part of the LCD display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the display. • False or OFF: Turns OFF the display.
Preset value	False or OFF

Examples

```
Dim DispTbl As Boolean
SCPI.DISPlay.TABLE.TYPE = "echo"
SCPI.DISPlay.TABLE.STATe = True
DispTbl = SCPI.DISPlay.TABLE.STATe
```

Related objects

SCPI.DISPlay.TABLE.TYPE

Equivalent key

Sweep Setup > **Edit Segment Table****Marker Fctn** > **Marker Table****Analysis** > **Limit Test** > **Edit Limit Line**

Analysis > **Ripple Limit** > **Edit Ripple Line**

Macro Setup > **Echo Window**

Cal > **Power Calibration** > **Loss Compen**

Cal > **Power Calibration** > **Sensor A Settings | Sensor B Settings**

When performing the operation from the front panel, you select the type of the window that appears in the lower part of the LCD display and turn ON/OFF the display at the same time.

Equivalent SCPI command

Syntax

```
:DISPlay:TABLE[:STATe] {ON|OFF|1|0}  
:DISPlay:TABLE[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:TABL ON"  
20 OUTPUT 717;":DISP:TABL?"  
30 ENTER 717;A
```

SCPI.DISPlay.TABLe.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.TABLe.TYPE = *Param**Param* = SCPI.DISPlay.TABLe.TYPE

Description

This command selects the type of the window that appears in the lower part of the LCD display.

Variable

Parameter	<i>Param</i>
Description	Window type
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "MARKer": Specifies the marker table window. • "LIMit": Specifies the limit test table window. • "SEGment": Specifies the segment table window. • "ECHO": Specifies the echo window. • "PLOSs": Specifies the loss compensation table window. • "SCFactor": Specifies the power sensor's calibration factor table window. • "RLIMit": Specifies the ripple test table window.
Preset value	"MARKer"

Examples

```
Dim TbType As String
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
TbType = SCPI.DISPlay.TABLe.TYPE
```

Related objects

SCPI.DISPlay.TABLe.STATe

Equivalent key

Sweep Setup > **Edit Segment Table**

Marker Fctn > **Marker Table**

Analysis > **Limit Test** > **Edit Limit Line**

Analysis > **Ripple Limit** > **Edit Ripple Line**

Macro Setup > **Echo Window**

Cal > **Power Calibration** > **Loss Compen**

Cal > **Power Calibration** > **Sensor A Settings|Sensor B Settings**

When performing the operation from the front panel, you select the type of the window that appears in the lower part of the LCD display and turn ON/OFF the display at the same time.

Equivalent SCPI command

Syntax

```
:DISPlay:TABLe:TYPE
{MARKEr|LIMit|SEGMENT|ECHO|PLOSs|SCFactor|RLIMit}
:DISPlay:TABLe:TYPE?
```

Query response

```
{MARK|LIM|SEGM|ECHO|PLOS|SCF|RLIM}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:TABL:TYPE SEGM"
20 OUTPUT 717;":DISP:TABL:TYPE?"
30 ENTER 717;A$
```

SCPI.DISPlay.UPDate.IMMediate**Object type**

Method (**Write-only**)

Syntax

SCPI.DISPlay.UPDate.IMMediate

Description

This command executes the display update once when the display update of the LCD screen is set to OFF (specifying False with the SCPI.DISPlay.ENABLE object),

Examples

```
SCPI.DISPlay.ENABLE = False  
SCPI.DISPlay.UPDate.IMMediate
```

Related objects

SCPI.DISPlay.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:DISPlay:UPDate[:IMMediate]

Example of use

```
10 OUTPUT 717;":DISP:UPD"
```

SCPI.DISPlay.WINDow(*Ch*).ACTivate

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ACTivate

Description

This command specifies selected channel (*Ch*) as the active channel.

- You can set only a channel displayed to the active channel. If this object is used to set a channel not displayed to the active channel, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.SPLit = "d1_2"  
SCPI.DISPlay.WINDow(2).ACTivate
```

Related objects

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect

Equivalent key

Channel Prev / Channel Next

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:ACTivate

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ACT"
```


SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe = *Status**Status* = SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

Description

This command turn ON/OFF the mode that align the marker display position of each trace based on trace 1, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF the mode that align the marker display position of each trace based on trace 1
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the mode that align marker display position based on trace 1. • False or OFF: Turns OFF the alignment.
Preset value	True or ON

Examples

Dim AnnMarkAlig As Boolean

SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe = False

AnnMarkAlig = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > Annotation Options > Align

Equivalent SCPI command

Syntax

E5071C

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:ALIGn[:STATe]
{ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:ALIGn[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ANN:MARK:ALIG OFF"  
20 OUTPUT 717;":DISP:WIND1:ANN:MARK:ALIG?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe = *Status**Status* = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe**Description**

This command turns ON/OFF the display of the marker value of only active traces, for the selected channel (*Ch*).

- If the function is turned OFF, marker values of all traces (markers) are displayed.

Variable

Parameter	<i>Status</i>
Description	ON/OFF the display of the marker value of only active
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Displays the marker values of only active traces.(ON) • False or OFF: Displays the marker values of all traces. (OFF)
Preset value	True or ON

Examples

```
Dim AnnMarkAlig As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe = False
AnnMarkAlig = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe
```

Related objectsSCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATeSCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.XSCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y**Equivalent key****Marker Fctn > Annotation Options > Active Only****Equivalent SCPI command**

E5071C

Syntax

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:SINGle[:STATe]
{ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:SINGle[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ANN:MARK:SING OFF"  
20 OUTPUT 717;":DISP:WIND1:ANN:MARK:SING?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(Ch).LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).LABel = *Status**Status* = SCPI.DISPlay.WINDow(*Ch*).LABel

Description

This command turns ON/OFF the graticule label display of the graph of channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the graticule label display of the graph
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the graticule label display. • False or OFF: Turns OFF the graticule label display.
Preset value	True or ON

Examples

```
Dim DispGrat As Boolean
SCPI.DISPlay.WINDow(1).LABel = False
DispGrat = SCPI.DISPlay.WINDow(1).LABel
```

Equivalent key

Display > Graticule Label

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:LABel {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:LABel?

Query response

E5071C

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;";DISP:WIND1:LAB ON"  
20 OUTPUT 717;";DISP:WIND1:LAB?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(Ch).MAXimize**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.WINDow(*Ch*).MAXimize = *Status**Status* = SCPI.DISPlay.WINDow(*Ch*).MAXimize**Description**

This command turns ON/OFF the maximization of the active trace of selected channel (*Ch*).

- If you turned ON the maximization, only the maximized active trace is displayed in the window and the other traces are not displayed.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the maximization of the active trace
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the maxim display. • False or OFF: Turns OFF the maxim display.
Preset value	False or OFF

Examples

```
Dim TracMax As Boolean
SCPI.CALCulate(1).PARAmeter(2).SElect
SCPI.DISPlay.WINDow(1).MAXimize = True
TracMax = SCPI.DISPlay.WINDow(1).MAXimize
```

Related objects

```
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.DISPlay.MAXimize
```

Equivalent key**Trace Max**

E5071C

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:MAXimize {ON|OFF|1|0}  
:DISPlay:WINDow{[1]-160}:MAXimize?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:MAX ON"  
20 OUTPUT 717;":DISP:WIND1:MAX?"  
30 ENTER 717;A
```


SCPI.DISPlay.WINdow(Ch).SPLit

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINdow(Ch).SPLit = *Param**Param* = SCPI.DISPlay.WINdow(Ch).SPLit

Description

This command sets/gets the graph layout of selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Graph layout
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "D1": See Channel graph window layouts. • "D12": See Channel graph window layouts. • "D1_2": See Channel graph window layouts. • "D112": See Channel graph window layouts. • "D1_1_2": See Channel graph window layouts. • "D123": See Channel graph window layouts. • "D1_2_3": See Channel graph window layouts. • "D12_33": See Channel graph window layouts. • "D11_23": See Channel graph window layouts. • "D13_23": See Channel graph window layouts. • "D12_13": See Channel graph window layouts. • "D1234": See Channel graph window layouts. • "D1_2_3_4": See Channel graph window layouts.

	<p>layouts.</p> <ul style="list-style-type: none">• "D12_34": See Channel graph window layouts.• "D123_456": See Channel graph window layouts.• "D12_34_56": See Channel graph window layouts.• "D1234_5678": See Channel graph window layouts.• "D12_34_56_78": See Channel graph window layouts.• "D123_456_789": See Channel graph window layouts.• "D123__ABC": See Channel graph window layouts.• "D1234__9ABC": See Channel graph window layouts.• "D1234__DEFG": See Channel graph window layouts.
Preset value	"D1"

Examples

```
Dim TracAloc As String
SCPI.DISPlay.WINDow(1).SPLit = "d1_2"
TracAloc = SCPI.DISPlay.WINDow(1).SPLit
```

Related objects

```
SCPI.DISPlay.SPLit
```

Equivalent key

Display > Allocate Traces

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-36}:SPLit {D1|D12|D1_2|D112|D1_1_2|
D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|D1234|D1_2_3_4|D12_3
4|D123_456|D12_34_56|
D1234_5678|D12_34_56_78|D123_456_789|D123__ABC|D1234__9ABC|
D1234__DEFG}
```

:DISPlay:WINDow{[1]-36}:SPLit?

Query response

```
{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D1
2_13|D1234|D1_2_3_4|
D12_34|D123_456|D12_34_56|D1234_5678|D12_34_56_78|D123_456_7
89|D123__ABC|
D1234__9ABC|D1234__DEFG}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND:SPL D1_2"
20 OUTPUT 717;":DISP:WIND:SPL?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA = *Lbl*
Lbl = SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Description

This command sets/gets the title label displayed in the title area of selected channel (*Ch*).

Variable

Parameter	<i>Lbl</i>
Description	Title label
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim TtlLbl As String
SCPI.DISPlay.WINDow(1).TITLe.DATA = "Filter"
SCPI.DISPlay.WINDow(1).TITLe.STATe = True
TtlLbl = SCPI.DISPlay.WINDow(1).TITLe.DATA
```

Related objects

SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Equivalent key

Display > Edit Title Label

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TITLe:DATA <string>
:DISPlay:WINDow{[1]-160}:TITLe:DATA?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TITL:DATA ""Title""  
20 OUTPUT 717;":DISP:WIND1:TITL?"  
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(Ch).TITLe.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe = *Status**Status* = SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe**Description**

This command turns ON/OFF the title label display in the title area of channels 1 to 160 (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the title label display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none">• True or ON: Turns ON the title label display.• False or OFF: Turns ON the title label display.
Preset value	False or OFF

Examples

```
Dim DispTtl As Boolean
SCPI.DISPlay.WINDow(1).TITLe.DATA = "Filter"
SCPI.DISPlay.WINDow(1).TITLe.STATe = True
DispTtl = SCPI.DISPlay.WINDow(1).TITLe.STATe
```

Related objectsSCPI.DISPlay.WINDow(*Ch*).TITLe.DATA**Equivalent key****Display > Title Label****Equivalent SCPI command****Syntax**

:DISPlay:WINDow{[1]-160}:TITLe[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:TITLe[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TITL ON"  
20 OUTPUT 717;":DISP:WIND1:TITL?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X =
Value

Value =SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

This command sets/gets the display position of the marker value on the X-axis by a percentage of a width of the display span, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Display position of the marker value on the X-axis.
Data type	Double precision floating point type (Double)
Range	-15 to 100
Preset value	1
Unit	% (percent)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim AnnMPosX As Double

SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.X = 15

AnnMPosX = SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.X

Related objects

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATeSCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATeSCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Marker Info X Pos**

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:TRACe{[1]-
16}:ANNotation:MARKer:POSition:X <numeric>
:DISPlay:WINDow{[1]-160}:TRACe{[1]-
16}:ANNotation:MARKer:POSition:X?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:X 33"
20 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:X?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y =
Value

Value =

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

This command sets/gets the display position of the marker value on the X-axis by a percentage of a width of the display span, for the selected trace (*Tr*) of selected channel (*Ch*), and the marker value on Y axis by a percentage of a height of the display span.

Variable

Parameter	<i>Value</i>
Description	Display position of the marker value on the Y-axis.
Data type	Double precision floating point type (Double)
Range	-15 to 100
Preset value	1
Unit	% (percent)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim AnnMPosY As Double

SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.Y = 23

AnnMPosY = SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.Y

Related objects

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATeSCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

Equivalent key

Marker Fctn > Annotation Options > Marker Info X Pos

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:TRACe{[1]-
16}:ANNotation:MARKer:POSition:Y <numeric>
:DISPlay:WINDow{[1]-160}:TRACe{[1]-
16}:ANNotation:MARKer:POSition:Y?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:Y 33"
20 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:Y?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe = *Status**Status* = SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe

Description

This command turns ON/OFF the memory trace display, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the memory trace display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the memory trace display. • False or OFF: Turns OFF the memory trace display.
Preset value	False or OFF

Examples

```
Dim DispMem As Boolean
SCPI.DISPlay.WINDow(1).TRACe(2).MEMory.STATe = True
DispMem = SCPI.DISPlay.WINDow(1).TRACe(2).MEMory.STATe
```

Related objects

SCPI.CALCulate(Ch).SElected.MATH.MEMorize

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe

Equivalent key

Display > **Display** > **Mem** (when the data trace display is OFF)**Display** > **Display** > **Data & Mem** (when the data trace display is ON)

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :MEMory[:STATe]  
{ON|OFF|1|0}  
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :MEMory[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:MEM ON"  
20 OUTPUT 717;":DISP:WIND1:TRAC1:MEM?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe = *Status**Status* = SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe**Description**

This command turns ON/OFF the data trace display, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the data trace display
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the data trace display. • False or OFF: Turns OFF the data trace display.
Preset value	True or ON

Examples

```
Dim DispTrac As Boolean
SCPI.DISPlay.WINDow(1).TRACe(2).STATe = False
DispTrac = SCPI.DISPlay.WINDow(1).TRACe(2).STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe

Equivalent key**Display** > **Display** > **Data** (when the memory trace display is OFF)**Display** > **Display** > **Data & Mem** (when the memory trace display is ON)**Equivalent SCPI command****Syntax**

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :STATe {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:STAT ON"  
20 OUTPUT 717;":DISP:WIND1:TRAC1:STAT?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO**Object type**Method (**Write-only**)**Syntax**SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO**Description**

This command executes the auto scale function, for the selected trace (*Tr*) of selected channel (*Ch*). The Auto Scale function automatically adjusts the value of the reference division line and the scale per division to display the trace appropriately.

Examples

SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO

Related objectsSCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVisionSCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel**Equivalent key****Scale > Auto Scale****Equivalent SCPI command****Syntax**

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:AUTO

Example of use

10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:AUTO"

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision = *Value**Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision

Description

For the selected trace (*Tr*) of selected channel (*Ch*), when the data format is not the Smith chart format or the polar format, sets the scale per division. When the data format is the Smith chart format or the polar format, sets the full scale value (the value of the outermost circumference).

Variable

Parameter	<i>Value</i>
Description	Scale value
Data type	Double precision floating point type (Double)
Range	1E-18 to 1E8
Preset value	<p>Varies depending the data format.</p> <ul style="list-style-type: none"> • Log magnitude: 10 • Phase, Expanded phase or Positive phase: 90 • Group delay: 1E-8 • Smith chart or Polar or SWR: 1 • Linear magnitude: 0.1 • Real or Imaginary: 0.2
Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> • Log magnitude: dB (decibel) • Phase, Expanded phase or Positive phase: ° (degree) • Group delay: s (second) • Others: No unit

Note

If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pdiv As Double
SCPI.CALCulate(1).PARAmeter(2).SElect
SCPI.CALCulate(1).SElected.FORMat = "gdel"
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALE.PDIVision = 1E-9
Pdiv = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALE.PDIVision
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FORMat
SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALE.RLEVel
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALE.RPOSition
```

Equivalent key

Scale > **Scale/Div**

Equivalent SCPI command**Syntax**

```
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALE]:PDIVision
<numeric>
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALE]:PDIVision?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:PDIV 2.5"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:PDIV?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel = *Value**Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Description

This command sets/gets the value of the reference division line, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of reference division line
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	<p>Varies depending on the data format.</p> <ul style="list-style-type: none"> Log magnitude (MLOG): dB (decibel) Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) Group delay (GDEL): s (second) Others: No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim RefLvl As Double
SCPI.CALCulate(1).PARAmeter(2).SElect
SCPI.CALCulate(1).SElected.FORMAT = "phas"
```

E5071C

```
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RLEVel = 90  
Pdiv = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RLEVel
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FORMat  
SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions  
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision  
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition
```

Equivalent key

[Scale] > Reference Value

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:RLEVel  
<numeric>  
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:RLEVel?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RLEV 1E2"  
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RLEV?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*T36r*).Y.SCALe.RPOSition = *Value**Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Description

This command specifies the position of a reference division line with its number (an integer assigned starting from 0 from the lowest division), for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Position of reference division line
Data type	Long integer type (Long)
Range	0 to the number of divisions
Preset value	5
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim RefPos As Long
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RPOSition = 6
RefPos = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RPOSition
```

Related objects

```
SCPI.CALCulate(Ch).SElected.FORMat
SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel
```

Equivalent key

Scale > Reference Position

E5071C

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-36} :Y[:SCALe]:RPOSition
<numeric>

:DISPlay:WINDow{[1]-160}:TRACe{[1]-36} :Y[:SCALe]:RPOSition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RPOS 6"  
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RPOS?"  
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(Ch).X.SPACing

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(Ch).X.SPACing = *Param**Param* = SCPI.DISPlay.WINDow(Ch).X.SPACing

Description

This command selects the display type of the graph horizontal axis of selected channel (*Ch*) for segment sweep.

Variable

Parameter	<i>Param</i>
Description	Horizontal axis display type of the graph for segment sweep
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "LINear": Specifies the frequency base (linear frequency axis with the minimum frequency at the left edge and the maximum frequency at the right edge). • "OBASe" : Specifies the order base (axis in which the measurement point numbers are positioned evenly in the order of measurement).
Preset value	"OBASe"

Examples

```

Dim DispSegm As String
SCPI.SENSE(1).SWEep.TYPE = "segm"
SCPI.DISPlay.WINDow(1).X.SPACing = "obas"
DispSegm = SCPI.DISPlay.WINDow(1).X.SPACing

```

Related objects

SCPI.SENSE(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > Segment Display

Equivalent SCPI command

E5071C

Syntax

:DISPlay:WINDow{[1]-160}:X:SPACing {LINear|OBASe}
:DISPlay:WINDow{[1]-160}:X:SPACing?

Query response

{LIN|OBAS}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:X:SPAC OBAS"  
20 OUTPUT 717;":DISP:WIND1:X:SPAC?"  
30 ENTER 717;A$
```


SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions**Object type**Property (**Read-Write**)**Syntax**SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions = *Value**Value* = SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions**Description**

This command sets/gets the number of divisions in all the graphs, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Number of divisions of graph
Data type	Long integer type (Long)
Range	4 to 30
Preset value	10
Resolution	2
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim Divs As Long

SCPI.DISPlay.WINDow(1).Y.SCALE.DIVisions = 12

Divs = SCPI.DISPlay.WINDow(1).Y.SCALE.DIVisions

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALE.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALE.RLEVel

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALE.RPOSition

Equivalent key**Scale > Divisions**

E5071C

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:Y[:SCALe]:DIVisions <numeric>  
:DISPlay:WINDow{[1]-160}:Y[:SCALe]:DIVisions?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:Y:DIV 12"  
20 OUTPUT 717;":DISP:WIND1:Y:DIV?"  
30 ENTER 717;A
```

FORMAT

SCPI.FORMat.BORDer

Object type

Property (**Read-Write**)

Syntax

SCPI.FORMat.BORDer = *Param*

Param = SCPI.FORMat.BORDer

Description

This command sets/gets the transfer order of each byte in the output data (byte order), when the data transfer format is set to binary mode (by specifying "REAL" with SCPI.FORMat.DATA object).

NOTE

This object is NOT used when controlling the E5071C using COM objects through E5071C VBA.

Variable

Parameter	<i>Param</i>
Description	Byte order
Data type	Character string type (String)
Range	<p>Select from the following:</p> <p>"NORMal": Specifies the byte order in which transfer starts from the byte including MSB (Most Significant Bit).</p> <p>"SWAPped": Specifies the byte order in which transfer starts from the byte including LSB (Least Significant Bit).</p>
Preset value	"NORMal"

Examples

```
Dim BitOrd As String
SCPI.FORMat.BORDer "swap"
BitOrd = SCPI.FORMat.BORDer
```

Related objects

SCPI.FORMat.DATA

Equivalent key

E5071C

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:FORMat:BORDER {NORMal|SWAPped}

:FORMat:BORDER?

Query response

{NORM|SWAP}<newline><^END>

Example of use

```
10 OUTPUT 717;":FORM:BORD SWAP"  
20 OUTPUT 717;":FORM:BORD?"  
30 ENTER 717;A$
```

SCPI.FORMat.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.FORMat.DATA = *Param**Param* = SCPI.FORMat.DATA

Description

This command can be used to set/get format data using the following SCPI commands:

```
:CALC{1-160}:DATA:FDAT
:CALC{1-160}:DATA:FMEM
:CALC{1-160}:DATA:SDAT?
:CALC{1-160}:DATA:SMEM?
:CALC{1-160}:FUNC:DATA?
:CALC{1-160}:LIM:DATA
:CALC{1-160}:LIM:REP?
:CALC{1-160}:LIM:REP:ALL?
:CALC{1-160}:BLIM:REP?
:CALC{1-160}:RLIM:DATA?
:CALC{1-160}:RLIM:REP?
:SENS{1-160}:CORR:COEF
:SENS{1-160}:FREQ:DATA?
:SENS{1-160}:SEGM:DATA
:SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA
:SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA
:SOUR{1-160}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS:DATA
:SOUR{1-160}:POW:PORT{1-4}:CORR:DATA
```

NOTE

ASCII transfer format must be specified when controlling the E5071C using SCPI commands with the Parse object in the E5071C VBA.

NOTE

Selection of the data format determines the result format of SCPI.CALCulate(Ch).SElected.DATA.XAXis.

Variable

Parameter	<i>Param</i>
Description	Data transfer format
Data type	Character string type (String)

Range	Select from the following: "ASCIi": Specifies the ASCII transfer format. "REAL": Specifies the IEEE 64-bit floating point binary transfer format. "REAL32": Specifies the IEEE 32-bit floating point binary transfer format.
Preset value	"ASCIi"

Examples

```
Dim Fmt As String
SCPI.FORMat.DATA = "ASC"
Fmt = SCPI.FORMat.DATA
```

Related objects

```
SCPI.FORMat.BORDer
Parse
SCPI.CALCulate(Ch).SElected.DATA.XAXis
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:FORMat:DATA {ASCIi|REAL|REAL32}
:FORMat:DATA?
```

Query response

```
{ASC|REAL|REAL32}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":FORM:DATA REAL"
20 OUTPUT 717;":FORM:DATA?"
30 ENTER 717;A$
```

HCOPY

SCPI.HCOPy.ABORT

Object type

Method (**Write-only**)

Syntax

SCPI.HCOPy.ABORT

Description

This command aborts the print output.

Examples

SCPI.HCOPy.ABORT

Related objects

SCPI.HCOPy.IMMediate

Equivalent key

System > **Abort Printing**

Equivalent SCPI command

Syntax

:HCOPy:ABORT

Example of use

10 OUTPUT 717;":HCOP:ABOR"

SCPI.HCOPy.IMAGe**Object type**Property (**Read-Write**)**Syntax**`SCPI.HCOPy.IMAGe = Param``Param = SCPI.HCOPy.IMAGe`**Description**

This command sets/gets the print color for output (to the printer).

Variable

Parameter	<i>Param</i>
Description	Print color for output to the printer.
Data type	Character string type (String)
Range	Select from the following: <ul style="list-style-type: none"> • "NORMal": Specifies printing in close color to the display color. • "INVert": Specifies printing in the inverted color of the display color.
Preset value	"INVert"

Examples

```
Dim Img As String
SCPI.HCOPy.IMAGe = "norm"
Img = SCPI.HCOPy.IMAGe
```

Related objects`SCPI.HCOPy.IMMediate`**Equivalent key****System > Invert Image****Equivalent SCPI command****Syntax**

```
:HCOPy:IMAGe {NORMal|INVert}
:HCOPy:IMAGe?
```


Query response

{NORM|INV}<newline><^END>

Example of use

```
10 OUTPUT 717;":HCOP:IMAG NORM"  
20 OUTPUT 717;":HCOP:IMAG?"  
30 ENTER 717;A$
```

SCPI.HCOPy.IMMediate

Object type

Method (**Write-only**)

Syntax

Value = SCPI.HCOPy.IMMediate

Description

This command outputs the display image on the LCD display to the printer connected to the E5071C.

NOTE

When printing the E5071C measurement screen, execute the VBA program with the Visual Basic editor closed. For the method, see Running a Program from the E5071C Measurement Screen.

Examples

SCPI.HCOPy.IMMediate

Related objects

SCPI.HCOPy.ABORT

SCPI.HCOPy.IMAGe

Equivalent key

[System] > Print

When performing the operation from the front panel, the image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **[Capture] ([System])** key is pressed) is printed. If no image is memorized in the clipboard, in the same way as the SCPI.HCOPy.IMMediate object, the image on the LCD display at the execution is memorized in the clipboard and then it is printed.

Equivalent SCPI command

Syntax

:HCOPy[:IMMediate]

Example of use

10 OUTPUT 717;":HCOP"

IEEE**SCPI.IEEE4882.CLS****Object type**Method (**Write-only**)**Syntax**

SCPI.IEEE4882.CLS

Description

This command clears the following:

- Error Queue
- Status Byte Register
- Standard Event Status Register
- Operation Status Event Register
- Questionable Status Event Register
- Questionable Limit Status Event Register
- Questionable Limit Extra Status Event Register
- Questionable Limit Channel Status Event Register
- Questionable Limit Channel Extra Status Event Register
- Questionable Bandwidth Limit Status Event Register
- Questionable Bandwidth Limit Extra Status Event Register
- Questionable Bandwidth Limit Channel Status Event Register
- Questionable Bandwidth Limit Channel Extra Status Event Register
- Questionable Ripple Limit Status Event Register
- Questionable Ripple Limit Extra Status Event Register
- Questionable Ripple Limit Channel Status Event Register
- Questionable Ripple Limit Channel Extra Status Event Register

Examples

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*CLS

Example of use

E5071C

10 OUTPUT 717;"*CLS"

SCPI.IEEE4882.ESE**Object type**Property (**Read-Write**)**Syntax**SCPI.IEEE4882.ESE = *Value**Value* = SCPI.IEEE4882.ESE**Description**

This command sets/gets the value of the Standard Event Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Standard Event Status Enable Register
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the result of bitwise AND with 255 (0xff) is set.

Examples

```
Dim Stat As Long
SCPI.IEEE4882.ESE = 16
Stat = SCPI.IEEE4882.ESE
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*ESE <numeric>

*ESE?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*ESE 16"  
20 OUTPUT 717;"*ESE?"  
30 ENTER 717;A
```

SCPI.IEEE4882.ESR**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.IEEE4882.ESR

Description

This command reads the value of the Standard Event Status Register. Execution of this command clears the register value.

Variable

Parameter	<i>Value</i>
Description	Value of the Standard Event Status Register
Data type	Long integer type (Long)
Preset value	128

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.ESR
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*ESR?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*ESR?"
20 ENTER 717;A
```

SCPI.IEEE4882.IDN

Object type

Property (**Read-only**)

Syntax

Cont = SCPI.IEEE4882.IDN

Description

This command reads the product information (manufacturer, model number, serial number, and firmware revision number) of the E5071C.

Variable

Parameter	<i>Cont</i>
Description	<p>Product information ("{string 1},{string 2},{string 3},{string 4}")</p> <ul style="list-style-type: none">• {string 1}: Manufacturer. Agilent Technologies is always read out.• {string 2}: Model number. E5071C is always read out.• {string 3}: Serial number (example: MY123400101).• {string 4}: Firmware revision number (example: A.07.00).
Data type	Character string type (String)

Examples

Dim Who As String
Who = SCPI.IEEE4882.IDN

Equivalent key

System > **Firmware Revision**

System > **Service Menu** > **Enable Options** > **Serial Number**

Equivalent SCPI command

Syntax

*IDN?

Query response

{string 1},{string 2},{string 3},{string 4}<newline><^END>

Example of use


```
10 OUTPUT 717;"*IDN?"  
20 ENTER 717;A$
```

SCPI.IEEE4882.OPC**Object type**Property (**Read-Write**)**Syntax**(1) SCPI.IEEE4882.OPC = *Dummy*(2) *Value* = SCPI.IEEE4882.OPC**Description**

Case (1):

This command sets/gets 1 the OPC bit (bit 0) of the Standard Event Status Register when all of pending operations complete. For information on the structure of the status register, see Status Reporting System.

Variable

Case (2):

Parameter	<i>Value</i>
Description	1 returned when all pending operations are complete
Data type	Long integer type (Long)

Examples

SCPI.IEEE4882.OPC = 1

Dmy = SCPI.IEEE4882.OPC

Related objects

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.ISOLation

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.LOAD

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OPEN

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.SHORt

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.THRU

SCPI.TRIGger.SEQuence.SINGle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*OPC

Example of use

10 OUTPUT 717;"*OPC"

E5071C

SCPI.IEEE4882.OPT

Object type

Property (**Read-only**)

Syntax

Cont = SCPI.IEEE4882.OPT

Description

This command reads the identification numbers of options installed in the E5071C.

Variable

Parameter	<i>Cont</i>
Description	Identification numbers of installed options
Data type	Character string type (String)
Note	If there is no installed option, 0 is read out.

Examples

```
Dim OptNum As String
OptNum = SCPI.IEEE4882.OPT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*OPT?

Query response

{numeric}<newline><^END>

Example of use

```
Call viVPrintf(vi, "**OPT?" + vbLf, 0)
Call viVScanf(vi, "%t", Result)
```

SCPI.IEEE4882.RST**Object type**

Method (**Write-only**)

Syntax

SCPI.IEEE4882.RST

Description

This command presets the E5071C to its default settings and is different from setting state preset with the SCPI.SYSTem.PRESet object as the continuous initiation mode (see SCPI.INITiate(Ch).CONTInuous) of channel 1 is set to OFF.

Examples

SCPI.IEEE4882.RST

Related objects

SCPI.INITiate(Ch).CONTInuous

SCPI.SYSTem.PRESet

SCPI.SYSTem.UPReset

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*RST

Example of use

10 OUTPUT 717;"*RST"

SCPI.IEEE4882.SRE**Object type**Property (**Read-Write**)**Syntax**SCPI.IEEE4882.SRE = *Value**Value* = SCPI.IEEE4882.SRE**Description**

This command sets/gets the value of Service Request Enable Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Service Request Enable Register
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the result of bitwise AND with 255 (0xff) is set. Note that bit 6 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.IEEE4882.SRE = 8
Stat = SCPI.IEEE4882.SRE
```

Related objects

SCPI.IEEE4882.ESE

SCPI.STATus.OPERation.ENABLE

SCPI.STATus.QUEStionable.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*SRE <numeric>

*SRE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*SRE 128"  
20 OUTPUT 717;"*SRE?"  
30 ENTER 717;A
```

SCPI.IEEE4882.STB**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.IEEE4882.STB

Description

This command reads the value of Status Byte Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Status Byte Register
Data type	Long integer type (Long)
Preset Value	0

Examples

Dim Stat As Long
Stat = SCPI.IEEE4882.STB

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*STB?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*STB?"
20 ENTER 717;A
```


SCPI.IEEE4882.TRG**Object type**

Method (**Write-only**)

Syntax

SCPI.IEEE4882.TRG

Description

This command triggers the E5071C if the trigger source is set to GPIB/LAN (set to BUS with the SCPI.TRIGger.SEQuence.SOURce).

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"  
SCPI.IEEE4882.TRG
```

Related objects

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

*TRG

Example of use

```
10 OUTPUT 717;"*TRG"
```

SCPI.IEEE4882.WAI

Object type

Method (**Write-only**)

Syntax

SCPI.IEEE4882.WAI

Description

This command waits for the execution of all objects sent before this command to be completed.

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"  
SCPI.TRIGger.SEQuence.SINGle  
SCPI.IEEE4882.WAI  
MsgBox "Done"
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*WAI

Example of use

```
10 OUTPUT 717;"*WAI"
```

INIT**SCPI.INITiate(*Ch*).CONTInuous**

Object type

Property (**Read-Write**)

Syntax

SCPI.INITiate(*Ch*).CONTInuous = *Status**Status* = SCPI.INITiate(*Ch*).CONTInuous

Description

This command turns ON/OFF the continuous initiation mode (setting by which the trigger system initiates continuously) of selected channel (*Ch*) in the trigger system. For more information on the trigger system, see Section Trigger System.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the continuous initiation mode
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the continuous initiation mode. • False or OFF: Turns OFF the continuous initiation mode.
Preset value	Varies depending on [<i>variable (Ch)</i>]

Examples

```
Dim ContMode As Boolean
SCPI.INITiate(2).CONTInuous = True
ContMode = SCPI.INITiate(2).CONTInuous
```

Related objects

SCPI.INITiate(*Ch*).IMMEDIATE

Equivalent key

Trigger > **Continuous** (continuous initiation mode ON)**Trigger** > **Hold** (continuous initiation mode OFF)

Equivalent SCPI command

E5071C

Syntax

:INITiate{[1]-160}:CONTInuous {ON|OFF|1|0}

:INITiate{[1]-160}:CONTInuous?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":INIT1:CONT OFF"  
20 OUTPUT 717;":INIT1:CONT?"  
30 ENTER 717;A
```

SCPI.INITiate(*Ch*).IMMediate**Object type**Method (**Write-only**)**Syntax**SCPI.INITiate(*Ch*).IMMediate**Description**

This command changes the state of each channel of selected channel (*Ch*) to the initiation state in the trigger system.

When this object is executed for a channel in the idle state in the trigger system, it goes into the initiation state immediately. Then, after measurement is executed once, it goes back to the idle state.

If this object is executed for a channel that is not in the idle state or a channel for which the continuous initiation mode is set to ON (setting by which the trigger system initiates continuously) in the trigger system, an error occurs when executed and the object is ignored. For more information on the trigger system, see Trigger System.

Examples

```
SCPI.INITiate(1).CONTinuous = False
SCPI.INITiate(1).IMMediate
```

Related objectsSCPI.INITiate(*Ch*).CONTinuous**Equivalent key****Trigger > Single****Equivalent SCPI command****Syntax**

:INITiate{[1]-160}[:IMMediate]

Example of use

10 OUTPUT 717;":INIT1"

MEMORY

:MMEMory:LOAD:PROGram

No equivalent COM Commands

Syntax

:MMEMory:LOAD:PROGram <string>

Description

This command loads (or imports) a VBA project (a file with the .vba extension), a module (a file with the .bas extension), a user form (a file with the .frm extension) or a class module (a file with the .cls extension). If the specified file does not exist, an error occurs and the command is ignored.

This command appears the VBA editor with Auto_complete_Feature, however, this is not supported on VBA/COM.

Variables

Parameter	<i>String</i>
Description	File name in which you want to load the VBA project
Range	254 characters or less
Preset value	""

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:PROG ""Test1/Test1_01.vba""
```

```
10 OUTPUT 717;":MMEM:LOAD:PROG ""A:Test1_01.vba""
```

Related commands

:MMEM:STOR:PROG

Equivalent key

Macro Setup > Load VBA Project

:MMEMory:STORe:PROGram

No equivalent COM Commands

Syntax

:MMEMory:STORe:PROGram <string>

Description

This command saves a VBA project opened on the VBA editor into a file. The file name need to have a .vba extension. If a file with the specified file name exists, its contents are overwritten.

This command appears the VBA editor with Auto_complete_Feature, however, this is not supported on VBA/COM.

Variables

Parameter	<i>String</i>
Description	File name in which you want to save the VBA project
Range	254 characters or less
Preset value	""

Example of use

```
10 OUTPUT 717;":MMEM:STOR:PROG ""Test1/Test1_01.vba""
```

```
10 OUTPUT 717;":MMEM:STOR:PROG ""D:Test1_01.vba""
```

Related commands

:MMEM:LOAD:PROG

Equivalent key

Macro Setup > Save VBA Project

:MMEMory:TRANSfer

No equivalent COM Commands

Syntax

:MMEMory:TRANSfer <string>,<block>

:MMEMory:TRANSfer? <string>

Description

This command sets/gets data to/from a file on the built-in storage device of the E5071C. By reading out data with this command and writing it to a file on the external controller, file transfer from the E5071C to the external controller can be realized.

On the other hand, by reading out data from the external controller and writing it to a file on the E5071C with this command, file transfer from the external controller to the E5071C can be realized.

When you use directory names and file name, separate them with "/" (slash) or "\" (backslash). If a file with the specified file name already exists for writing or if the specified file does not exist for reading out (Query), an error occurs and the command is ignored.

This command appears the VBA editor with Auto_complete_Feature, however, this is not supported on VBA/COM.

Variables

Parameter	<i>String</i>
Description	File name on the built-in storage device
Range	254 characters or less

Parameter	<i>block</i>
Description	Data written on/read out from the file
Range	GPIB: 20 Mbytes or less LAN: 100 Kbytes or less

Query response

{block}<newline><^END>

Example of use

```

10 OUTPUT 717;":MMEM:TRAN ""Trace01.csv"";#6012345";Dat$
10 OUTPUT 717;":MMEM:TRAN? ""Trace01.csv""
20 ENTER 717 USING "#,A";A$
30 ENTER 717 USING "#,A";Digit$
40 Img$="#,"&Digit$&"A"
50 ENTER 717 USING Img$;Byte$
60 Img$=Byte$&"A"
70 ALLOCATE Dat$[VAL(Byte$)]
80 ENTER 717 USING Img$;Dat$

```

Equivalent key

No equivalent key is available on the front panel.

SCPI.MMEMemory.CATalog(*Dir*)

Object type

Property (**Read Only**)

Syntax

Cont = SCPI.MMEMemory.CATalog(*Dir*)

Description

This command reads the following information on the built-in storage device of the E5071C:

- Space in use
- Available space
- Name and size of all files (including directories) in the specified directory
 - To read out the information in the root directory (folder), specify "\" (backslash). Separate between directory names (file name) with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>Cont</i>
Description	<p>Directory information ("{A},{B},{Name 1},{Size 1},{Name 2},{Size 2}, ... ,{Name N},{Size N}")</p> <p>Where N is the number of all files in the specified directory and n is an integer between 1 and N.</p> <p>{A}: Space in use of the built-in storage device (byte).</p> <p>{B}: Available space of the built-in storage device (byte).</p> <p>{Name n}: Name of the n-th file (directory).</p> <p>{Size n}: Size (byte) of the n-th file (directory). Always 0 for directories.</p>
Data type	Character string type (String)

Parameter	<i>Dir</i>
Description	Directory name whose information you want to

	read out
Data type	Character string type (String)
Range	254 characters or less

Examples

```
Dim DirCont As String
DirCont = SCPI.MMEemory.CATalog("d:\")
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:MMEemory:CATalog? <string 1>
```

Query response

```
{string 2}<newline><^END>
```

The format of the readout character string is as follows:

```
"{used_size},{free_size},{name 1},{size 1}, ... ,{name N},{size N}"
```

Where:

N is the number of all files in the specified directory and n is an integer between 1 and N.

{used_size}:

Space in use of the built-in **storage device** (byte).

{free_size}:

Available space of the built-in **storage device** (byte).

{name n}:

Name of the n-th file (directory).

{size n}:

Size (byte) of the n-th file (directory). Always 0 for directories.

Example of use

E5071C

```
10 DIM A$(1000)
20 OUTPUT 717;"MMEM:CAT? ""\""
30 ENTER 717;A$
```

SCPI.MMEemory.COPY**Object type**Method (**Write Only**)**Syntax**SCPI.MMEemory.COPY = *File***Description**

This command copies a file.

- Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	Indicates 2 file names (copy source and copy destination). <i>File(0)</i> : Copy source file name <i>File(1)</i> : Copy destination file name The index of the array starts from 0.
Data type	Variant type (Variant)
Range	254 characters or less
Note	If the specified copy source file does not exist, an error occurs when executed and the object is ignored. Notice that, if a file with the same name as the specified copy destination file name exists, its contents are overwritten.

Examples

SCPI.MMEemory.COPY = Array("test/state01.sta","d:test01.sta")

Dim File(1) As Variant

File(0) = "test/state01.sta"

File(1) = "d:test01.sta"

SCPI.MMEemory.COPY = File

Equivalent key**Save/Recall > Save State > File Dialog...****Equivalent SCPI command**

E5071C

Syntax

:MMEMory:COPY <string 1>,<string 2>

Example of use

```
10 OUTPUT 717;":MMEM:COPY ""Test1/State01.sta"",""d:Test1_01.sta"""
```

SCPI.MMEMemory.DElete**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.DElete = *File***Description**

This command deletes an existing file or directory (folder).

- When you delete a directory, all the files and directories in it are deleted.

Specify the file name with the extension.

When you specify a file (directory) under an existing directory, separate them with "\" (back slash), or "/" (slash).

To delete all files in the directory (folder), specify "\" (backslash).

Variable

Parameter	<i>File</i>
Description	File name or directory name you want to delete
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file or directory does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.MMEMemory.DElete = "d:\"
```

```
SCPI.MMEMemory.DElete = "test/state01.sta"
```

Equivalent key

Save/Recall > Save State > File Dialog...

Equivalent SCPI command**Syntax**

```
:MMEMemory:DElete <string>
```

E5071C

Example of use

```
10 OUTPUT 717;":MMEM:DEL ""Test1/State01.sta""
```

```
10 OUTPUT 717;":MMEM:DEL ""D:State01.sta""
```


SCPI.MMEemory.LOAD.ASCFactor**Object type**Property (**Write Only**)**Syntax**SCPI.MMEemory.LOAD.ASCFactor = *File***Description**

This command recalls the file (file with the ".csv" extension saved with the SCPI.MMEemory.STORE.ASCFactor object) you want to specify as the table for the reference calibration coefficient and calibration coefficient table for power sensor A.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

Parameter	<i>File</i>
Description	A file name (extension ".csv") of the reference calibration coefficient and the calibration coefficient table for power sensor A.
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples

SCPI.MMEemory.LOAD.ASCFactor = "d:\sensor01.csv"

SCPI.MMEemory.LOAD.ASCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEemory.STORE.ASCFactor

Equivalent key**Cal > Power Calibration > Sensor A Settings > Import from CSV File****Equivalent SCPI command****Syntax**

:MMEemory:LOAD:ASCFactor <string>

E5071C

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:ASCF ""Test1/Sensor01.csv""
```

```
10 OUTPUT 717;":MMEM:LOAD:ASCF ""D:Sensor01.csv""
```

SCPI.MMEemory.LOAD.BSCFactor**Object type**Property (**Write Only**)**Syntax**SCPI.MMEemory.LOAD.BSCFactor = *File***Description**

This command recalls the file (file with the ".csv" extension saved with the SCPI.MMEemory.STORE.BSCFactor) you want to specify as the table for the reference calibration coefficient and the calibration coefficient table for power sensor B.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	A file name (extension ".csv") of the reference calibration coefficient and the calibration coefficient table for power sensor B.
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples

SCPI.MMEemory.LOAD.BSCFactor = "d:\sensor01.csv"

SCPI.MMEemory.LOAD.BSCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEemory.STORE.BSCFactor

Equivalent key**Cal > Power Calibration > Sensor B Settings > Import from CSV File****Equivalent SCPI command****Syntax**

:MMEemory:LOAD:BSCFactor <string>

E5071C

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:BSCF ""Test1/Sensor01.csv""
```

```
10 OUTPUT 717;":MMEM:LOAD:BSCF ""D:Sensor01.csv""
```

SCPI.MMEMemory.LOAD.CHANnel.STATe

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMemory.LOAD.CHANnel.STATe = *Register*

Description

This command recalls the instrument state for an individual channel (saved with the SCPI.MMEMemory.STORE.CHANnel.STATe object) from the specified register as the setting of the active channel.

Variable

Parameter	<i>Register</i>
Description	Register
Data type	Character string type (String)
Range	Select from the following: "A": Specifies register A. "B": Specifies register B. "C" :Specifies register C. "D" :Specifies register D.
Note	If no instrument state has been saved in the specified register, an error occurs and the object is ignored.

Examples

SCPI.MMEMemory.LOAD.CHANnel.STATe = "a"

Related objects

SCPI.MMEMemory.STORE.CHANnel.STATe

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Save/Recall > Recall Channel > A|B|C|D

Equivalent SCPI command

Syntax

:MMEMemory:LOAD:CHANnel[:STATe] {A|B|C|D}

E5071C

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:CHAN A"
```

SCPI.MMEemory.LOAD.CKIT(*Ckit*)**Object type**Property (**Write Only**)**Syntax**SCPI.MMEemory.LOAD.CKIT(*Ckit*) = *File***Description**

This command recalls the definition file of the specified calibration kit (file with the ".ckx" extension saved with the SCPI.MMEemory.STORE.CKIT(*Ckit*)).

Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>Ckit</i>
Description	Number of calibration kit
Data type	Long integer type (Long)
Range	1 to 30
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>File</i>
Description	File name of the definition table of a calibration kit
Data type	Character string type (String)
Range	254 characters or less

Note

If the specified file does not exist, a runtime error occurs.

Examples

```
SCPI.MMEemory.LOAD.CKIT(1) = "Test1/Ckit01.ckx"
```

```
SCPI.MMEemory.LOAD.CKIT(1) = "A:\Ckit01.ckx"
```

Related objects

```
SCPI.MMEemory.STORE.CKIT(Ckit)
```

Equivalent key

Cal > Modify Cal Kit > Import Cal Kit...

Equivalent SCPI command**Syntax**

```
:MMEemory:LOAD:CKIT{[1]-30} <string>
```

Example of use

```
10 OUTPUT 717;":MMEemory:LOAD:CKIT1 ""Test1/Ckit01.ckx"""
```

```
10 OUTPUT 717;":MMEemory:LOAD:LIM ""A:Ckit01.ckx"""
```


SCPI.MMEMemory.LOAD.LIMit**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.LOAD.LIMit = *File***Description**

This command recalls the specified limit table file (file with the .csv extension saved with SCPI.MMEMemory.STORE.LIMit), from the limit table for the active trace of the active channel.

NOTE

Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name of limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.LOAD.LIMit = "d:\limit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.LOAD.LIMit = "test/limit01.csv"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.MMEMemory.STORE.LIMit
```

Equivalent key

Analysis > Limit Test > Edit Limit Line > Import from CSV File

Equivalent SCPI command

E5071C

Syntax

:MMEMory:LOAD:LIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:LIM ""Test1/Limit01.csv""
```

```
10 OUTPUT 717;":MMEM:LOAD:LIM ""D:Limit01.csv""
```

SCPI.MMEMemory.LOAD.PLOSs(*Pt*)**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.LOAD.PLOSs(*Pt*) = *File***Description**

This command recalls the specified loss compensation table file (a file with the ".csv" extension saved with SCPI.MMEMemory.STORE.PLOSs(*Pt*)), as the loss compensation table for the active channel, for the selected port (*Pt*).

Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name of the loss compensation table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMemory.LOAD.PLOSs(1) = "d:\loss01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMemory.LOAD.PLOSs(1) = "test/loss01.csv"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.MMEMemory.STORE.PLOSs(Pt)
```

Equivalent key**Cal > Power Calibration > Loss Compens > Import from CSV File****Equivalent SCPI command****Syntax**

:MMEMemory:LOAD:PLOSs{[1]|2|3|4} <string>

E5071C

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:PLOS1 ""Test1/Loss01.csv""
```

```
10 OUTPUT 717;":MMEM:LOAD:PLOS1 ""D:Loss01.csv""
```

SCPI.MMEMemory.LOAD.RLIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMemory.LOAD.RLIMit = *File*

Description

This command recalls the specified ripple limit table file (file with the .csv extension saved with SCPI.MMEMemory.STORE.RLIMit) of the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command), as ripple limit table for the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SElect).

NOTE

Specify the file name with the extension. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).
If the specified file does not exist, an error occurs and the command is ignored.

Variable

Parameter	<i>File</i>
Description	File name of the ripple limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.LOAD.RIMit = "D:\Rlimit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.LOAD.RLIMit = "test/Rlimit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

E5071C

SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.MMEMory.STORe.RLIMit

Equivalent key

Analysis > Ripple Limit > Edit Ripple Line > Import from CSV File

Equivalent SCPI command

Syntax

:MMEMory:LOAD:RLIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:RLIM ""RTest1/Rlim01.csv""  
10 OUTPUT 717;":MMEM:LOAD:RLIM ""D:Rlim01.csv""
```

SCPI.MMEMemory.LOAD.SEGMent**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.LOAD.SEGMent = *File***Description**

This command recalls the specified segment sweep table file (file with a .csv extension saved with the SCPI.MMEMemory.STORE.SEGMent), as the segment sweep table of the active channel.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name of segment sweep table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMemory.LOAD.SEGMent = "d:\segm01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMemory.LOAD.SEGMent = "test/segm01.csv"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.MMEMemory.STORE.SEGMent
```

Equivalent key**Sweep Setup > Edit Segment Table > Import from CSV File****Equivalent SCPI command****Syntax**

:MMEMemory:LOAD:SEGMENT <string>

E5071C

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:SEGM ""Test1/Segm01.csv""  
10 OUTPUT 717;":MMEM:LOAD:SEGM ""D:Segm01.csv""
```


SCPI.MMEMemory.LOAD.STATE**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.LOAD.STATE = *File***Description**

This command recalls the specified instrument state file (file with a .sta extension saved with SCPI.MMEMemory.STORE.STATE).

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name of instrument state (extension ".sta")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

SCPI.MMEMemory.LOAD.STATE = "d:\state01.sta"

SCPI.MMEMemory.LOAD.STATE = "test/state01.sta"

Related objects

SCPI.MMEMemory.STORE.STATE

Equivalent key**Save/Recall > Recall State****Equivalent SCPI command****Syntax**

:MMEMemory:LOAD[:STATE] <string>

Example of use

10 OUTPUT 717;":MME:LOAD ""Test1/State01.sta""

10 OUTPUT 717;":MME:LOAD ""d:State01.sta""

SCPI.MMEMory.MDIRectory

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.MDIRectory = *File*

Description

This command creates a new directory (folder).

NOTE When you create a directory under an existing directory, separate between the directory names with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	Directory name you want to create
Data type	Character string type (String)
Range	254 characters or less
Note	If a directory with the same name as the specified directory name exists, an error occurs when executed and the object is ignored.

Examples

SCPI.MMEMory.MDIRectory = "d:\test"

SCPI.MMEMory.MDIRectory = "test"

Equivalent key

Save/Recall > Save State > File Dialog...

Equivalent SCPI command

Syntax

:MMEMory:MDIRectory <string>

Example of use

10 OUTPUT 717;":MMEM:MDIR ""Test1""

10 OUTPUT 717;":MMEM:MDIR ""d:Test1""

SCPI.MMEemory.STORE.ASCFactor**Object type**Property (**Write Only**)**Syntax**SCPI.MMEemory.STORE.ASCFactor = *File***Description**

This command saves the reference calibration coefficient and the calibration coefficient table for power sensor A into a CSV file (extension ".csv").

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	A file name (extension ".csv") to save the reference calibration coefficient and the calibration coefficient table for power sensor A.
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

SCPI.MMEemory.STORE.ASCFactor = "d:\sensor01.csv"

SCPI.MMEemory.STORE.ASCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEemory.LOAD.ASCFactor

Equivalent key**Cal > Power Calibration > Sensor A Settings > Export to CSV File****Equivalent SCPI command****Syntax**

:MMEemory:STORE:ASCFactor <string>

Example of use

E5071C

10 OUTPUT 717;";MMEM:STOR:ASCF ""Test1/Sensor01.csv""

10 OUTPUT 717;";MMEM:STOR:ASCF ""D:Sensor01.csv""

SCPI.MMEemory.STORE.BSCFactor**Object type**Property (**Write Only**)**Syntax**SCPI.MMEemory.STORE.BSCFactor = *File***Description**

This command saves the reference calibration coefficient and the calibration coefficient table for power sensor B into a CSV file (extension ".csv").

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	A file name (extension ".csv") to save the reference calibration coefficient and the calibration coefficient table for power sensor B.
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

SCPI.MMEemory.STORE.BSCFactor = "d:\sensor01.csv"

SCPI.MMEemory.STORE.BSCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEemory.LOAD.BSCFactor

Equivalent key**Cal > Power Calibration > Sensor B Settings > Export to CSV File****Equivalent SCPI command****Syntax**

:MMEemory:STORE:BSCFactor <string>

Example of use

E5071C

10 OUTPUT 717;";MMEM:STOR:BSCF ""Test1/Sensor01.csv""

10 OUTPUT 717;";MMEM:STOR:BSCF ""D:Sensor01.csv""

SCPI.MMEMemory.STORE.CHANnel.CLEar**Object type**

Method (**Write-only**)

Syntax

SCPI.MMEMemory.STORE.CHANnel.CLEar

Description

This command deletes the instrument state for each channel (saved with the SCPI.MMEMemory.STORE.CHANnel.STATe object) in all the registers.

Examples

SCPI.MMEMemory.STORE.CHANnel.CLEar

Related objects

SCPI.MMEMemory.STORE.CHANnel.STATe

Equivalent key

Save/Recall > Save Channel > Clear States > OK

Equivalent SCPI command**Syntax**

:MMEMemory:STORE:CHANnel:CLEar

Example of use

10 OUTPUT 717;":MME:STOR:CHAN:CLE"

SCPI.MMEMemory.STORE.CHANnel.STATe**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.STORE.CHANnel.STATe = *Register***Description**

This command saves the instrument state of the items set for the active channel specific to that channel only into the specified register (volatile memory).

Variable

Parameter	<i>Register</i>
Description	Register
Data type	Character string type (String)
Range	Select from the following: "A": Specifies register A. "B": Specifies register B. "C": Specifies register C. "D": Specifies register D.
Note	If an instrument state has been saved already in the specified register, its contents are overwritten.

Examples

SCPI.MMEMemory.STORE.CHANnel.STATe = "a"

Related objects

SCPI.MMEMemory.LOAD.CHANnel.STATe

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key**Save/Recall > Save Channel > A|B|C|D****Equivalent SCPI command****Syntax**

:MMEMemory:STORE:CHANnel[:STATe] {A|B|C|D}

Example of use

10 OUTPUT 717;":MMEM:STOR:CHAN A"

SCPI.MMEMemory.STORE.CKIT(*Ckit*)

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMemory.STORE.CKIT(*Ckit*) = *File*

Description

This command saves the definition table of the calibration kit to a file.

- Specify the file name with the .ckx extension. When you use a directory name and file name, separate them with "/" (slash) or "\" (backslash).

If a file with the specified file name exists, its contents are overwritten.

Variable

Parameter	<i>File</i>
Description	A file name used to save the definition of the calibration kit.
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

- For information on the variable (*Ckit*), see Val Ckit.

Examples

```
SCPI.MMEMemory.STORE.CKIT(1) = "d:\Ckit01.ckx"
```

```
SCPI.MMEMemory.STORE.CKIT(1) = "test/trace01.csv"
```

Related objects

SCPI.MMEMemory.LOAD.CKIT(*Ckit*)

Equivalent key

Cal > Modify Cal Kit > Export Cal Kit...

Equivalent SCPI command

Syntax

```
:MMEMemory:STORE:CKIT{[1]-30} <string>
```

Example of use

```
10 OUTPUT 717;":MMEM:STOR:CKIT1 ""Test1/Ckit01.ckx""  
10 OUTPUT 717;":MMEM:STOR:CKIT1 ""d:Ckit01.ckx""
```

SCPI.MMEMemory.STORE.FDATA**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.STORE.FDATA = *File***Description**

This command saves the formatted data array into a file in the CSV format (extension ".csv"), for the active trace of the active channel.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name in which you want to save the formatted data array (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.STORE.FDATA = "d:\trace01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.STORE.FDATA = "test/trace01.csv"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
```

Equivalent key**Save/Recall > Save Trace Data****Equivalent SCPI command****Syntax**

:MMEMory:STORe:FDATa <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:FDAT ""Result/Trace01.csv""
```

```
10 OUTPUT 717;":MMEM:STOR:FDAT ""D:Trace01.csv""
```

SCPI.MMEMemory.STORE.IMAGE**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.STORE.IMAGE = *File***Description**

This command saves the display image on the LCD display at the execution of the object into a file in the bitmap (extension ".bmp") or portable network graphics (extension ".png") format.

- When saving the E5071C measurement screen, execute the VBA program with the Visual Basic editor closed. For more information, see Running a Program from the E5071C Measurement Screen.

Specify the file name with the extension When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name in which you want to save the display image on the LCD display (extension ".bmp" or ".png")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

SCPI.MMEMemory.STORE.IMAGE = "d:\image01.bmp"

SCPI.MMEMemory.STORE.IMAGE = "test/image01.png"

Equivalent key**System > Dump Screen Image**

When performing the operation from the front panel, the image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture (System)** key is pressed) is saved.

If no image is memorized in the clipboard, in the same way as the SCPI.MMEMory.STORe.IMAGe object, the image on the LCD display at the execution is memorized in the clipboard and then it is saved.

Equivalent SCPI command

Syntax

:MMEMory:STORe:IMAGe <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:IMAG ""Result/Image01.bmp""
```

```
10 OUTPUT 717;":MMEM:STOR:IMAG ""D:image01.png""
```

SCPI.MMEMemory.STORE.LIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMemory.STORE.LIMit = *File*

Description

This command saves the limit table of the active trace of the active channel into a file in the CSV format (extension ".csv").

NOTE

Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name to save the limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.STORE.LIMit = "d:\limit01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SElect
SCPI.MMEMemory.STORE.LIMit = "test/limit01.csv"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.CALCulate(Ch).PARAmeter(Tr).SElect
SCPI.MMEMemory.LOAD.LIMit
```

Equivalent key

Analysis > Limit Test > Edit Limit Line > Export to CSV File

Equivalent SCPI command

Syntax

:MMEMory:STORe:LIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:LIM ""Test1/Limit01.csv""  
10 OUTPUT 717;":MMEM:STOR:LIM ""D:Limit01.csv""
```

SCPI.MMEMemory.STORE.PLOSs(*Pt*)**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.STORE.PLOSs(*Pt*) = *File***Description**

This command saves the loss compensation table of the active channel into a file in the CSV format (extension ".csv"), for the selected port (*Pt*).

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	A file name to save the loss compensation table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINdow(1).ACTivate
SCPI.MMEMemory.STORE.PLOSs(1) = "d:\loss01.csv"

SCPI.DISPlay.WINdow(1).ACTivate
SCPI.MMEMemory.STORE.PLOSs(1) = "test/loss01.csv"
```

Related objects

```
SCPI.DISPlay.WINdow(Ch).ACTivate
SCPI.MMEMemory.LOAD.PLOSs(Pt)
```

Equivalent key**Cal > Power Calibration > Loss Compens > Export to CSV File****Equivalent SCPI command****Syntax**

:MMEMemory:STORE:PLOSs{[1]|2|3|4} <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:PLOS1 ""Test1/Loss01.csv""  
10 OUTPUT 717;":MMEM:STOR:PLOS1 ""D:Loss01.csv""
```

SCPI.MMEMemory.STORE.RLIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMemory.STORE.RLIMit = *File*

Description

This command saves the ripple limit table of the active trace (specified with the SCPI.CALCulate(Ch).PARAmeter(Tr).SELEct command) of the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the CSV format.

NOTE

Specify the file name with the .sta extension. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).

If a file with the specified file name already exists, its contents will be overwritten.

Variable

Parameter	<i>File</i>
Description	File name used to save the ripple limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples (1)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SELEct
SCPI.MMEMemory.STORE.RLIMit = "D:\Rlimit01.csv"
```

Examples (2)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARAmeter(1).SELEct
SCPI.MMEMemory.STORE.RLIMit = "test/Rlimit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARAmeter(Tr).SELEct
SCPI.MMEMory.LOAD.RLIMit

Equivalent key

Analysis > Ripple Limit > Edit Ripple Line > Export to CSV File

Equivalent SCPI command

Syntax

:MMEMory:STORe:RLIMit <string>

Example of use

10 OUTPUT 717;":MMEM:STOR:RLIM ""RTest1/Rlim01.csv""
10 OUTPUT 717;":MMEM:STOR:RLIM ""D:Rlim01.csv""

SCPI.MMEMemory.STORE.SALL**Object type**Property (**Read-Write**)**Syntax**SCPI.MMEMemory.STORE.SALL = *Status**Status* = SCPI.MMEMemory.STORE.SALL**Description**

This command sets/gets whether to save the settings of all channels/traces or that of the displayed channels/traces only, as the instrument state to be saved.

Variable

Parameter	<i>Status</i>
Description	Selecting content to be saved as the instrument state setting.
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Specifies the setting of all channels/traces as the target to be saved. • False or OFF: Specifies the setting of displayed channels/traces only as the target to be saved.
Preset value	False or OFF

Examples

```
Dim Obj As Boolean
SCPI.MMEMemory.STORE.SALL = True
Obj = SCPI.MMEMemory.STORE.SALL
```

Related objects

SCPI.MMEMemory.STORE.STATe

Equivalent key**Save/Recall > Channel/Trace****Equivalent SCPI command****Syntax**

:MMEMory:STORe:SALL {ON|OFF|1|0}
:MMEMory:STORe:SALL?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SALL ON"  
20 OUTPUT 717;":MMEM:STOR:SALL?"  
30 ENTER 717;A
```

SCPI.MMEMemory.STORE.SEGMent**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.STORE.SEGMent = *File***Description**

This command saves the segment sweep table of the active channel into a file in the CSV format (extension ".csv").

- Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name to save segment sweep table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMemory.STORE.SEGMent = "d:\segm01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMemory.STORE.SEGMent = "test/segm01.csv"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.MMEMemory.LOAD.SEGMent
```

Equivalent key**Sweep Setup > Edit Segment Table > Export to CSV File****Equivalent SCPI command****Syntax**

:MMEMemory:STORE:SEGMent <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SEGM ""Test1/Segm01.csv""  
10 OUTPUT 717;":MMEM:STOR:SEGM ""D:Segm01.csv""
```

SCPI.MMEMemory.STORE.SNP.DATA**Object type**Property (**Write Only**)**Syntax**SCPI.MMEMemory.STORE.SNP.DATA = *File***Description**

Saves the measurement data for the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

You need to specify a file format and file type before saving a file. The extension differs depending on file types as shown in the follownig table:

<file type>	<extension>
When specifying one port	s1p
When specifying two ports	s2p
When specifying three ports	s3p
When specifying four ports	s4p

- When you use directory names and file name, separate them with "/" (slash) or "\" (back slash).

If a file with the specified file name already exists, its contents are overwritten.

Variable

Parameter	<i>File</i>
Description	File name you want to use when saving file in the touchstone format
Range	254 characters or less

- When invalid extension is specified, an error message appears and the command is ignored.

Examples

```
Dim SnpPorts(2) As Variant
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SNP.FORMat = "RI"
SnpPorts(0) = 1
SnpPorts(1) = 2
SnpPorts(2) = 4
SCPI.MMEMory.STORe.SNP.TYPE.S3P = SnpPorts
SCPI.MMEMory.STORe.SNP.DATA = "SNP01.s3p"
```

Related objects

```
SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.MMEMory.STORe.SNP.FORMat
SCPI.MMEMory.STORe.SNP.TYPE.S1P
SCPI.MMEMory.STORe.SNP.TYPE.S2P
SCPI.MMEMory.STORe.SNP.TYPE.S3P
SCPI.MMEMory.STORe.SNP.TYPE.S4P
```

Equivalent key

After a file type is specified, a dialog box will appear.

Equivalent SCPI command

Syntax

```
:MMEMory:STORe:SNP[:DATA] <string>
```

Example of use

```
10 OUTPUT 717;"DISP:WIND1:ACT"
20 OUTPUT 717;":MMEM:STOR:SNP:FORM RI"
30 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P 1,2,4"
40 OUTPUT 717;":MMEM:STOR:SNP ""SNP01.s3p"""
```

SCPI.MMEemory.STORE.SNP.FORMat**Object type**Property (**Read-Write**)**Syntax**SCPI.MMEemory.STORE.SNP.FORMat = *Param**Param* = SCPI.MMEemory.STORE.SNP.FORMat**Description**

This command sets/gets the data format for saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	<i>Param</i>
Description	Touchstone file format
Data type	Character string type (String)
Range	Select from the following: "AUTO": Specifies data format automatically according to the display format of the active trace. "MA": Specifies data format "log magnitude > angle". "DB": Specifies data format "linear magnitude > angle". "RI": Specifies data format "real part > imaginary part".
Preset value	"AUTO"

Examples

```
Dim Fmt As String
SCPI.MMEemory.STORE.SNP.FORMat = "MA"
Fmt = SCPI.MMEemory.STORE.SNP.FORMat
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEemory.STORE.SNP.DATA

Equivalent key

**Save/Recall > Save Snp > Snp Format >
 AUTO|LogMag/Angle|LinMag/Angle|Real/Imaginary**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:FORMat {AUTO|MA|DB|RI}
:MMEMory:STORe:SNP:FORMat?

Query response

{AUTO|MA|DB|RI}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:FORM MA"  
20 OUTPUT 717;":MMEM:STOR:SNP:FORM?"  
30 ENTER 717;A$
```

SCPI.MMEMemory.STORE.SNP.TYPE.S1P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMemory.STORE.SNP.TYPE.S1P = *Port*

Port = SCPI.MMEMemory.STORE.SNP.TYPE.S1P

Description

This command sets/gets the specified port to the file type (1 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	<i>Port</i>
Description	Port number
Range	1 to 4
Resolution	1

Examples

10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P 2"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P?"
30 ENTER 717;A\$

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate
SCPI.MMEMemory.STORE.SNP.DATA
SCPI.MMEMemory.STORE.SNP.FORMat

Equivalent key

Save/Recall > Save SnP > S1P > 1|2|3|4

Equivalent SCPI command

Syntax

:MMEMemory:STORE:SNP:TYPE:S1P <numeric>
:MMEMemory:STORE:SNP:TYPE:S1P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P 2"  
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P?"  
30 ENTER 717;A$
```

SCPI.MMEemory.STORE.SNP.TYPE.S2P**Object type**Property (**Read-Write**)**Syntax**SCPI.MMEemory.STORE.SNP.TYPE.S2P = *Ports**Ports* = SCPI.MMEemory.STORE.SNP.TYPE.S2P**Description**

This command sets/gets the specified port to the file type (2 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for file type. <i>Ports(1)</i>: Specifies the other port for file type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two ports, an error occurs during execution. The order of the two port numbers to be specified is arbitrary.</p>

Examples

```
Dim Ports(1) As Long
Ports(0) = 2
Ports(1) = 3
SCPI.MMEemory.STORE.SNP.TYPE.S2P = Ports
Ports = SCPI.MMEemory.STORE.SNP.TYPE.S2P
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S2p** > **1-2|1-3|1-4|2-3|2-4|3-4**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S2P <numeric1>, <numeric 2>

:MMEMory:STORe:SNP:TYPE:S2P?

Example of use

10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P 2,3"

20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P?"

30 ENTER 717;A\$

SCPI.MMEemory.STORE.SNP.TYPE.S3P**Object type**Property (**Read-Write**)**Syntax**SCPI.MMEemory.STORE.SNP.TYPE.S3P = *Ports**Ports* = SCPI.MMEemory.STORE.SNP.TYPE.S3P**Description**

This command sets/gets the specified port to the file type (3 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 3-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Specifies a port for file type. <i>Ports(1)</i> :Specifies a port for file type. <i>Ports(2)</i> :Specifies a port for file type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two port numbers, an error occurs during execution. The order of the three port numbers to be specified is arbitrary.</p>

Examples

```
Dim Ports(2) As Long
Ports(0) = 2
Ports(1) = 3
Ports(2) = 4
SCPI.MMEemory.STORE.SNP.TYPE.S3P = Ports
Ports = SCPI.MMEemory.STORE.SNP.TYPE.S3P
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S3p** > **1-2-3|1-2-4|1-3-4|2-3-4**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S3P <numeric 1>, <numeric 2>, <numeric 3>

:MMEMory:STORe:SNP:TYPE:S3P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P 2,3,4"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P?"
30 ENTER 717;A$
```

SCPI.MMEemory.STORE.SNP.TYPE.S4P**Object type**Property (**Read-Write**)**Syntax**SCPI.MMEemory.STORE.SNP.TYPE.S4P = *Ports**Ports* = SCPI.MMEemory.STORE.SNP.TYPE.S4P**Description**

This command sets/gets the specified port to the file type (4 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i> :Specifies a port for file type. <i>Ports(1)</i> :Specifies a port for file type. <i>Ports(2)</i> :Specifies a port for file type. <i>Ports(3)</i> :Specifies a port for file type. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two port numbers, an error occurs during execution. The order of the four port numbers to be specified is arbitrary.</p>

Examples

```
Dim Ports(3) As Long
Ports(0) = 1
Ports(1) = 2
Ports(2) = 3
Ports(3) = 4
```

SCPI.MMEMemory.STORe.SNP.TYPE.S4P = Ports
 Ports = SCPI.MMEMemory.STORe.SNP.TYPE.S4P

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate
 SCPI.MMEMemory.STORe.SNP.DATA
 SCPI.MMEMemory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S4p** > **1-2-3-4**

Equivalent SCPI command

Syntax

:MMEMemory:STORe:SNP:TYPE:S4P 1, 2, 3, 4
 :MMEMemory:STORe:SNP:TYPE:S4P?

Example of use

```
10 OUTPUT 717;":MME:STOR:SNP:TYPE:S4P 1,2,3,4"
20 OUTPUT 717;":MME:STOR:SNP:TYPE:S3P?"
30 ENTER 717;A$
```

SCPI.MMEMemory.STORE.STATE

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMemory.STORE.STATE = *File*

Description

This command saves the instrument state (contents to be saved specified with the SCPI.MMEMemory.STORE.STYPE object) into a file (file with the .sta extension).

NOTE

Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	<i>File</i>
Description	File name to save the instrument state (extension ".sta")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
Dim StaType As String
SCPI.MMEMemory.STORE.STYPE = "cdst"
SCPI.MMEMemory.STORE.STATE = "d:\state01.sta"
```

```
Dim StaType As String
SCPI.MMEMemory.STORE.STYPE = "cdst"
SCPI.MMEMemory.STORE.STATE = "test/state01.sta"
```

Related objects

SCPI.MMEMemory.STORE.STYPE

SCPI.MMEMemory.LOAD.STATE

Equivalent key

Save/Recall > Save State

Equivalent SCPI command

Syntax

:MMEMory:STORe[:STATe] <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR ""Test1/State01.sta""  
10 OUTPUT 717;":MMEM:STOR ""D:State01.sta""
```

SCPI.MMEemory.STORE.STYPE**Object type**Property (**Read-Write**)**Syntax**SCPI.MMEemory.STORE.STYPE = *Param**Param* = SCPI.MMEemory.STORE.STYPE**Description**

Selects the contents saved when saving the instrument state into a file with the SCPI.MMEemory.STORE.STATE object.

Variable

Parameter	<i>Param</i>
Description	Data of instrument state
Data type	Character string type (String)
Range	<p>Select from the following.</p> <ul style="list-style-type: none"> • "STATE": Specifies the save of the measurement conditions only. • "CState": Specifies the save of the measurement conditions and the calibration state. • "DState": Specifies the save of the measurement conditions and the formatted data array. • "CDState": Specifies the save of the measurement conditions, the calibration state, and the formatted data array.
Preset value	"CState"

Examples

```
Dim StaType As String
SCPI.MMEemory.STORE.STYPE = "cdst"
StaType = SCPI.MMEemory.STORE.STYPE
```

Related objects

SCPI.MMEemory.STORE.STATE

Equivalent key

Save/Recall > **Save Type** > **State Only|State & Cal|State & Trace|All**

Equivalent SCPI command

Syntax

:MMEMory:STORe:STYPe {STATe|CSTate|DSTate|CDSTate}

:MMEMory:STORe:STYPe?

Query response

{STAT|CST|DST|CDST}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:STYP CDST"  
20 OUTPUT 717;":MMEM:STOR:STYP?"  
30 ENTER 717;A$
```

OUTPUT

SCPI.OUTPUT.STATE

Object type

Property (**Read Write**)

Syntax

SCPI.OUTPUT.STATE = *Status*

Status = SCPI.OUTPUT.STATE

Description

This command turns on/off of the stimulus signal output. Measurement cannot be made until the stimulus signal output is turned ON.

Variable

Parameter	<i>Status</i>
Description	On/off of the stimulus signal output
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns on the stimulus signal. • False or OFF: Turns off the stimulus signal.
Preset value	True or ON

Examples

```
Dim Outp As Boolean
SCPI.OUTPUT.STATE = True
Outp = SCPI.OUTPUT.STATE
```

Equivalent key

Sweep Setup > Power > RF Out

Equivalent SCPI command

Syntax

```
:OUTPUT[:STATE] {ON|OFF|1|0}
```

```
:OUTPUT[:STATE]?
```

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":OUTP ON"  
20 OUTPUT 717;":OUTP?"  
30 ENTER 717;A
```

PROGRAM

:PROGram:CATalog?

No equivalent COM Commands

Syntax

:PROGram:CATalog?

Description

This command reads the list of all executable VBA macros (procedures defined by Public including the VBA project loaded on the VBA editor).

Query response

{string}<newline><^END>

The character string in the following format, in which each macro is separated by a comma (.), is read out.

"{macro 1},{macro 2}, ... ,{macro N}"

Where N is the total number of VBA macros.

{macro n}: VBA macro name (module name.procedure name)

Example of use

```
10 DIM A$[1000]
20 OUTPUT 717;":PROG:CAT?"
30 ENTER 717;A$
```

:PROG:NAME

No equivalent COM Commands

Syntax

:PROG:NAME <string>

:PROG:NAME?

Description

This command sets/gets the VBA macro controlled with the :PROG:STAT command.

- Selectable VBA macro names can be read with the :PROG:CAT? command.

Variable

Parameter	<i>String</i>
Description	VBA macro name (module name.procedure name)
Range	254 characters or less
Preset value	""

Query response

{string}<newline><^END>

Example of use

10 OUTPUT 717;":PROG:NAME ""Module1.main""

20 OUTPUT 717;":PROG:NAME?"

30 ENTER 717;A\$

Related commands

:PROG:CAT?

:PROG:STAT

Equivalent key**Macro Setup > Select Macro**

- When performing a similar operation from the front panel, you should select the VBA macro and execute it at the same.

:PROGram[:SElected]:STATe

No equivalent COM Commands

Syntax

:PROGram[:SElected]:STATe {STOP|RUN}

:PROGram[:SElected]:STATe?

Description

This command sets/gets the control/state of the VBA macro selected with the **:PROG:STAT** command.

Variable

Range	STOP: Stop the program. RUN: Run the Program.
Preset value	STOP

Query response

{STOP|RUN}<newline><^END>

Example of use

10 OUTPUT 717;":PROG:STAT RUN"

20 OUTPUT 717;":PROG:STAT?"

30 ENTER 717;A\$

Related commands

:PROG:NAME

Equivalent key

Macro Break (to stop)

Macro Setup > **Select Macro** (to run)

- When performing the operation from the front panel, you select the VBA macro and execute it at the same time.

SCPI.PROGram.VARiable.ARRay(Vnum).DATA**Object type**Property (**Read-Write**)**Syntax**

SCPI.PROGram.VARiable.ARRay(Vnum).DATA = Data

Data = SCPI.PROGram.VARiable.ARRay(Vnum).DATA

Description

This command sets/gets an array of Data that can be exchanged between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	<i>Vnum</i>
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	<i>Data</i>
Description	<p>"n" is the number obtained from the value specified with the SCPI.PROGram.VARiable.ARRay(Vnum).SIZE object.</p> <ul style="list-style-type: none"> • Data(0): The first array data • Data(n-1): The n-th array data <p>The index of the array starts from 0.</p>
Data Type	Variant type (Variant)
Note	If the data size is not specified, an error occurs when executed.

Examples

Dim Var1(2) As Long

Dim Var2 as Variant

Dim ArraySize as Long

ArraySize=3

E5071C

```
Var1(1) = 2  
Var1(2) = 56  
SCPI.PROG:VAR:ARR:SIZE = ArraySize  
SCPI.PROG:VAR:ARR:DATA = Var1  
Var2= SCPI.PROG:VAR:ARR:DATA
```

Related objects

```
SCPI.PROG:VAR:ARR(Vnum).SIZE  
SCPI.PROG:VAR:DOUB(Vnum).DATA  
SCPI.PROG:VAR:LONG(Vnum).DATA  
SCPI.PROG:VAR:STR(Vnum).DATA
```

Equivalent key

None

Equivalent SCPI command

Syntax

```
:PROG:VAR:ARR{[1] - 10}:DATA <Data Array>  
:PROG:VAR:ARR{[1] - 10}:DATA ?
```

Example of use

```
10 OUTPUT 717;"PROG:VAR:ARR2:SIZE 4"  
20 OUTPUT 717;"PROG:VAR:ARR2 1.0,2.0,3.0,4.0"  
30 OUTPUT 717;"PROG:VAR:ARR2?"  
40 ENTER 717;A(*)
```


SCPI.PROGram.VARiable.ARRay(Vnum).SIZE**Object type**Property (**Read-Write**)**Syntax**

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE = Value

Value = SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

Description

This command sets/gets the size of an array of Data, specified by SCPI.PROGram.VARiable.ARRay(1-10).DATA, that can be exchanged between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	<i>Vnum</i>
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	<i>Value</i>
Description	The value of data size
Data Type	Long integer type (long)
Range	1 to 40002
Preset value	402
Resolution	1

Examples

Dim Var1(2) As Long

Dim Var2 as Variant

Dim ArraySize as Long

E5071C

ArraySize=3

Var1(1) = 2

Var1(2) = 56

SCPI.PROGrama.VARiable.ARRay(1).SIZE = ArraySize

SCPI.PROGrama.VARiable.ARRay(1).DATA = Var1

Var2= SCPI.PROGrama.VARiable.ARRay(1).DATA

Related objects

SCPI.PROGrama.VARiable.ARRay(Vnum).DATA

SCPI.PROGrama.VARiable.DOUBLE(Vnum).DATA

SCPI.PROGrama.VARiable.LONG(Vnum).DATA

SCPI.PROGrama.VARiable.STRING(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGrama:VARiable:ARRay{[1] - 10}:SIZE <Data Array Size>

:PROGrama:VARiable:ARRay{[1] - 10}:SIZE ?

Example of use

10 OUTPUT 717;"PROG:VAR:ARR2:SIZE 4"

20 OUTPUT 717;"PROG:VAR:ARR2 1.0,2.0,3.0,4.0"

30 OUTPUT 717;"PROG:VAR:ARR2?"

40 ENTER 717;A(*)

SCPI.PROGrama.VARiable.DOUBle(Vnum).DATA**Object type**Property (**Read-Write**)**Syntax**

SCPI.PROGrama.VARiable.DOUBle(Vnum).DATA = Value

Value = SCPI.PROGrama.VARiable.DOUBle(Vnum).DATA

Description

This command can be used to exchange Double type data between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	<i>Vnum</i>
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	<i>Value</i>
Description	The value of the double precision floating point type
Data type	Double precision floating point type (Double)
Range	Compliant with the double precision floating point type
Preset value	0

Examples

```
Dim DbIVal As Double
DbIVal =55.7890
```

```
SCPI.PROGrama.VARiable.DOUBle(1).DATA = DbIVal
DbIVal= SCPI.PROGrama.VARiable.DOUBle(1).DATA
```

Related objects

SCPI.PROGrama.VARiable.ARRay(Vnum).DATA

E5071C

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:DOUBle{[1]-10}:DATA <Numeric value, Double Type>

:PROGram:VARiable:DOUBle{[1]-10}:DATA ?

Example of use

10 OUTPUT 717;"PROGram:VARiable:DOUBle1:DATA 12345.89607"

20 OUTPUT 717;" :PROGram:VARiable:DOUBle1:DATA?"

30 ENTER 717;A\$

SCPI.PROGram.VARiable.LONG(Vnum).DATA**Object type**Property (**Read-Write**)**Syntax**

SCPI.PROGram.VARiable.LONG(Vnum).DATA = Value

Value = SCPI.PROGram.VARiable.LONG(Vnum).DATA

Description

This command can be used to exchange Long type data between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

The SCPI.PROGram.VARiable.LONG(Vnum).DATA command accepts the values from -2,147,483,648 to 2,147,483,647 as a long type variable. However, the range of scpi command (:PROG:VAR:LONG:DATA) is -65536 to 65536. Therefore, you can exchange the value from -65536 to 65536. If you set the value out of the range by the SCPI.PROGram.VARiable.LONG(Vnum).DATA command, the returned value of :PROG:VAR:LONG:DATA is -65536 or 65536.

Variable

Parameter	<i>Vnum</i>
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	<i>Value</i>
Description	The value of the long integer type
Data type	Long integer type (Long)
Range	-65536 to 65536
Preset value	0

E5071C

Examples

```
Dim LngVal As Long
LngVal = 65536
SCPI.PROGrama.VARiable.LONG(1).DATA = LngVal
LngVal= SCPI.PROGrama.VARiable.LONG(1).DATA
```

Related objects

```
SCPI.PROGrama.VARiable.ARRay(Vnum).DATA
SCPI.PROGrama.VARiable.ARRay(Vnum).SIZE
SCPI.PROGrama.VARiable.DOUBle(Vnum).DATA
SCPI.PROGrama.VARiable.STRing(Vnum).DATA
```

Equivalent key

None

Equivalent SCPI command

Syntax

```
:PROGrama:VARiable:LONG{[1] - 10}:DATA <Numeric value>
:PROGrama:VARiable:LONG{[1] - 10}:DATA?
```

Example of use

```
10 OUTPUT 717;":PROGrama:VARiable:LONG1:DATA 65536"
20 OUTPUT 717;":PROGrama:VARiable:LONG1:DATA?"
30 ENTER 717;A
```

SCPI.PROGram.VARiable.STRING(Vnum).DATA**Object type**Property (**Read-Write**)**Syntax**

SCPI.PROGram.VARiable.STRING(Vnum).DATA = Value

Value = SCPI.PROGram.VARiable.STRING(Vnum).DATA

Description

This command can be used to exchange String type data between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	<i>Vnum</i>
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	<i>Value</i>
Description	The value of the character string type
Data type	Character string type (String)
Range	Compliant with the character string type
Preset value	""

Examples

```
Dim StrVal As String
StrVal = "TestS11"
```

```
SCPI.PROGram.VARiable.STRING(1).DATA = StrVal
StrVal= SCPI.PROGram.VARiable.STRING(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

E5071C

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:STRing{[1] - 10}:DATA <Character value, String
Type>

:PROGram:VARiable:STRing{[1] - 10}:DATA ?

Example of use

```
10 OUTPUT 717;" PROG:VAR:STR ""TEST DATA""
20 OUTPUT 717;" PROG:VAR:STR?"
30 ENTER 717;A$
```


SENSE**SCPI.SENSE(*Ch*).AVERage.CLEar**

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).AVERage.CLEar

Description

This command resets the data count to 0, used for averaging of selected channel (*Ch*). Measurement data before the execution of this object is not used for averaging.

Examples

SCPI.SENSE(1).AVERage.CLEar

Related objects

SCPI.SENSE(*Ch*).AVERage.COUNTSCPI.SENSE(*Ch*).AVERage.STATe

Equivalent key

Avg > Averaging Restart

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:AVERage:CLEar

Example of use

10 OUTPUT 717;":SENS1:AVER:CLE"

SCPI.SENSE(*Ch*).AVERage.COUNT**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).AVERage.COUNT = *Value**Value* = SCPI.SENSE(*Ch*).AVERage.COUNT**Description**This command sets/gets the averaging factor of selected channel (*Ch*).**Variable**

Parameter	<i>Value</i>
Description	Averaging factor
Data type	Long integer type (Long)
Range	1 to 999
Preset value	16
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AvgCnt As Long
SCPI.SENSE(1).AVERage.COUNT = 4
AvgCnt = SCPI.SENSE(1).AVERage.COUNT
```

Related objectsSCPI.SENSE(*Ch*).AVERage.STATeSCPI.SENSE(*Ch*).AVERage.CLEAr**Equivalent key****Avg > Avg Factor****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:AVERage:COUNT <numeric>

:SENSe{[1]-160}:AVERage:COUNT?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:AVER:COUN 4"  
20 OUTPUT 717;":SENS1:AVER:COUN?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).AVERage.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).AVERage.STATe = *Status**Status* = SCPI.SENSE(*Ch*).AVERage.STATe

Description

This command sets/gets the averaging function of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the averaging function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the averaging function. • False or OFF: Turns OFF the averaging function.
Preset value	False or OFF

Examples

```
Dim Avg As Boolean
SCPI.SENSE(1).AVERage.STATe = True
Avg = SCPI.SENSE(1).AVERage.STATe
```

Related objects

SCPI.SENSE(*Ch*).AVERage.COUNTSCPI.SENSE(*Ch*).AVERage.CLEAr

Equivalent key

Avg > Averaging

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:AVERage[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:AVERage[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:AVER ON"  
20 OUTPUT 717;":SENS1:AVER?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).BANDwidth.RESolution**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).BANDwidth.RESolution = *Value**Value* = SCPI.SENSE(*Ch*).BANDwidth.RESolution**Description**This command sets/gets the IF bandwidth of selected channel (*Ch*).

- This command is similar to SCPI.SENSE(*Ch*).BWIDth.RESolution.

Variable

Parameter	<i>Value</i>
Description	IF bandwidth
Data type	Double precision floating point type (Double)
Range	10 to 1500000
Preset value	70000
Unit	Hz (hertz)
Resolution	In steps of 1, 1.5, 2, 3, 4, 5, or 7
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim IfBw As Double
SCPI.SENSE(1).BANDwidth.RESolution = 1.5E3
IfBw = SCPI.SENSE(1).BANDwidth.RESolution
```

Related objectsSCPI.SENSE(*Ch*).BWIDth.RESolution**Equivalent key****Avg > IF Bandwidth**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:BANDwidth[:RESolution] <numeric>  
:SENSe{[1]-160}:BANDwidth[:RESolution]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:BAND 1.5E3"  
20 OUTPUT 717;":SENS1:BAND?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).BWIDth.RESolution

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).BWIDth.RESolution = *Value**Value* = SCPI.SENSE(*Ch*).BWIDth.RESolution

Description

This command sets/gets the IF bandwidth of selected channel (*Ch*).

- This command is similar to SCPI.SENSE(*Ch*).BANDwidth.RESolution.

Variable

Parameter	<i>Value</i>
Description	IF bandwidth
Data type	Double precision floating point type (Double)
Range	10 to 1500000
Preset value	70000
Unit	Hz (hertz)
Resolution	In steps of 1, 1.5, 2, 3, 4, 5, or 7
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim IfBw As Double

SCPI.SENSE(1).BWIDth.RESolution = 1.5E3

IfBw = SCPI.SENSE(1).BWIDth.RESolution

Related objects

SCPI.SENSE(*Ch*).BANDwidth.RESolution

Equivalent key

Avg > **IF Bandwidth**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:BWIDth[:RESolution] <numeric>
:SENSe{[1]-160}:BWIDth[:RESolution]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:BWID 1.5E3"
20 OUTPUT 717;":SENS1:BWID?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.CLEar

Description

This command clears the error coefficient for calibration when the frequency offset feature is set to OFF, for the selected channel (*Ch*).

Example of use

SCPI.SENSE(1).CORRection.CLEar

Related objects

SCPI.SENSE(*Ch*).OFFSet.STATe

SCPI.SENSE(*Ch*).CORRection.OFFSet.CLEar

Equivalent key

Cal > **Clear** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:CLE"

SCPI.SENSE(Ch).CORRection.COEfficient.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COEfficient.DATA (Str, Int1 , Int2) = Array

Array = SCPI.SENSE(Ch).CORRection.COEfficient.DATA (Str, Int1, Int2)

Description

This command sets/gets the calibration coefficient data for specified channel.

- When the calibration factor is interpolated, the interpolated calibration coefficient array is read. Similarly, when the calibration factor is not interpolated, a non-interpolated calibration coefficient array is read.

After writing the calibration coefficient array, the written value becomes effective only after the (SCPI.SENSE(Ch).CORRection.COEfficient.SAVE) command is executed.

Variable

Parameter	Array
Description	<p>Indicates the array data (corrected data array) of NOP (number of measurement points)X2. Where n is an integer between 1 and NOP.</p> <p><i>Data(nX2-2)</i></p> <p>Real part of data (complex number) at the n-th measurement point.</p> <p><i>Data(nX2-1)</i></p> <p>Imaginary part of data (complex number) at the n-th measurement point.</p> <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)

Parameter	Str
-----------	-----

Description	Calibration type
Data type	Character string type (String)
Range	<p>Select from the following:</p> <ul style="list-style-type: none"> • "ES": Source match • "ER": Reflection tracking • "ED": Directivity • "EL": Load match • "ET": Transmission tracking • "EX": Isolation

Parameter	<i>Int1</i>
Description	Response port
Data type	Integer type (Integer)
Range	1 to 4
Note	If ES, ER, or ED is used, the response port and the stimulus port must be the same, while EL, ET, or EX is used, the response port and the stimulus port must be different.

Parameter	<i>Int2</i>
Description	Stimulus port
Data type	Integer type (Integer)

Range	1 to 4
Note	If ES, ER, or ED is used, the response port and the stimulus port must be the same, while EL, ET, or EX is used, the response port and the stimulus port must be different.

For information on the variable (*Ch*), see Ch

Examples

DIM Array As Variant
 Array = SCPI.SENS(1).CORRection.COEFFicient.DATA("EL", 1, 2)

Related objects

SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.ERESponse
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.RESponse.OPEN
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.RESponse.SHORT
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.RESponse.THRU
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.SOLT1
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.SOLT2
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.SOLT3
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.SOLT4
 SCPI.SENSE(*Ch*).CORRection.COEFFicient.SAVE
 SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL2
 SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL3
 SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COEFFicient[:DATA]
{ES|ER|ED|EL|ET|EX},<numeric 1>, <numeric 2>, <numeric 3>,...,
<numeric 3 n*2>

:SENSe{[1]-160}:CORRection:COEFFicient[:DATA]?
{ES|ER|ED|EL|ET|EX},<numeric 1>, <numeric 2>
```

Query response

```
{numeric 1}, ... ,{numeric NOP*2}<newline><^END>
```

	Description
{numeric nX 2-1}	Real part of data (complex number) at the n-th measurement point.
{numeric nX 2}	Imaginary part of data (complex number) at the n-th measurement point.

Because the calibration coefficient array is expressed by a complex number, the real part and the imaginary part of one measurement point are returned and obtained as a value. Here, NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:CORR:COEF? EL,1,2"
30 ENTER 717;A(*)
```

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.ERESponse**Object type**Method (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.ERESponse = Ports

Description

This command sets the calibration type to the enhanced response calibration between the two specified ports when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEfficient.DATA command, for the selected channel Ch.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the response port. <i>Ports(1)</i>: Specifies the stimulus port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESport(1) As Variant
ERESport(0) = 1
ERESport(1) = 2
SCPI.SENSE(1).CORRection.COEfficient.METHod.ERESponse = ERESport
```

Related objects

E5071C

SCPI.SENSE(Ch).CORRection.COEFficient.DATA

SCPI.SENSE(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod:ERESponse <numeric
1>,<numeric 2>

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:ERES 1,2"

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.RESPOnse.OPEN**Object type**

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.RESPOnse.OPEN = Port

Description

This command sets the calibration type to the response calibration (open) of the specified port when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEfficient.DATA command, for the selected channel Ch.

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Val Ch (Ch) and Val Port respectively.

Examples

SCPI.SENSE(1).CORRection.COEfficient.METHod.RESPOnse.OPEN = 1

Related objects

SCPI.SENSE(Ch).CORRection.COEfficient.DATA

SCPI.SENSE(Ch).CORRection.COEfficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COEfficient:METHod[:RESPOnse]:OPEN
<numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:OPEN 1"
```

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.RESponse.SHORt**Object type**Method (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.RESponse.SHORt = Port

Description

This command sets the calibration type to the response calibration (short) of the specified port for the selected channel, when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEfficient.DATA command.

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Val Ch (*Ch*) and Val Port.

Examples

SCPI.SENSE(1).CORRection.COEfficient.METHod.RESponse.SHORt = 1

Related objects

SCPI.SENSE(Ch).CORRection.COEfficient.DATA

SCPI.SENSE(Ch).CORRection.COEfficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COEfficient:METHod[:RESponse]:SHORt
<numeric>
```

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:SHOR 1"

SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPOnse.THRU**Object type**Method (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPOnse.THRU = Ports

Description

This command sets the calibration type to the response calibration (thru) between the two specified ports when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEFficient.DATA command, for the selected channel.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the input port. <i>Ports(1)</i>: Specifies the output port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

SCPI.SENSE(1).CORRection.COEFficient.METHod.RESPOnse.THRU = Array(2, 1)

Dim ThruPort(1) As Variant

ThruPort(0) = 2

ThruPort(1) = 1

SCPI.SENSE(1).CORRection.COEFficient.METHod.RESPOnse.THRU = ThruPort

Related objects

SCPI.SENSE(Ch).CORRection.COEFficient.DATA

SCPI.SENSE(Ch).CORRection.COEFficient.SAVE

E5071C

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COEFficient:METHod[:RESPonse]:THRU  
<numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:THRU 2,1"
```

SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT1**Object type**

Method (**Write-only**)

Syntax

```
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT1 = Port
```

Description

This command sets the calibration type to the full 1-port calibration of the specified port, when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEFficient.DATA command, for the selected channel.

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Val Ch (*Ch*) and Val Port respectively.

Examples

```
SCPI.SENSE(1).CORRection.COEFficient.METHod.SOLT1 = 1
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COEFficient.DATA
```

```
SCPI.SENSE(Ch).CORRection.COEFficient.SAVE
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COEFficient:METHod:SOLT1 <numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT1 1"
```

SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT2

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT2 = Ports

Description

This command sets the calibration type to full 2-port calibration between the two specified ports, when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEFficient.DATA command for the selected channel.

Variable

Parameter	Ports
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for full 2-port calibration. <i>Ports(1)</i>: Specifies the other port for full 2-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples (1)

SCPI.SENSE(1).CORRection.COEFficient.METHod.SOLT2 = Array(1, 2)

Examples (2)

Dim CalPort(1) As Variant
CalPort(0) = 1

```
CalPort(1) = 2
SCPI.SENSE(1).CORRection.COEFFicient.METHod.SOLT2 = CalPort
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COEFFicient.DATA
SCPI.SENSE(Ch).CORRection.COEFFicient.SAVE
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COEFFicient:METHod:SOLT2 <numeric
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT2 1,2"
```

SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.SOLT3

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.SOLT3 = Ports

Description

This command sets the calibration type to full 3-port calibration between the three specified ports, when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEFFicient.DATA command for the selected channel.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 3-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for full 3-port calibration. <i>Ports(1)</i>: Specifies a port for full 3-port calibration. <i>Ports(2)</i>: Specifies a port for full 3-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples (1)

SCPI.SENSE(1).CORRection.COEFFicient.METHod.SOLT3 = Array(1, 2, 3)

Examples (2)


```
Dim CalPort(2) As Variant
CalPort(0) = 1
CalPort(1) = 2
CalPort(2) = 3
SCPI.SENSE(1).CORRection.COEFFicient.METHod.SOLT3 = CalPort
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COEFFicient.DATA
SCPI.SENSE(Ch).CORRection.COEFFicient.SAVE
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COEFFicient:METHod:SOLT3 <numeric
1>,<numeric 2>,<numeric 3>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT3 1,2,3"
```

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.SOLT4

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.COEfficient.METHod.SOLT4 = Ports

Description

This command sets the calibration type to full 4-port calibration, when the calibration coefficient data array is written with the SCPI.SENSE(Ch).CORRection.COEfficient.DATA command for the selected channel.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for full 4-port calibration. <i>Ports(1)</i>: Specifies a port for full 4-port calibration. <i>Ports(2)</i>: Specifies a port for full 4-port calibration. <i>Ports(3)</i>: Specifies a port for full 4-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples (1)

SCPI.SENSE(1).CORRection.COEfficient.METHod.SOLT4 = Array(1, 2, 3, 4)

Examples (2)

```
Dim CalPort(3) As Variant
CalPort(0) = 1
CalPort(1) = 2
CalPort(2) = 3
CalPort(3) = 4
SCPI.SENSE(1).CORRection.COEFfient.METHod.SLOT4 = CalPort
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COEFficient.DATA
SCPI.SENSE(Ch).CORRection.COEFficient.SAVE
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COEFficient:METHod:SOLT4 <numeric 1>,
<numeric 2>, <numeric 3>, <numeric 4>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT4 1,2,3,4"
```

SCPI.SENSE(Ch).CORRection.COEFficient.SAVE**Object type**Method (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COEFficient.SAVE

Description

This command enables the calibration coefficients depending on the selected calibration type from the writing calibration data.

- Enabling the calibration coefficients clears all calibration data regardless of whether the data are used for the calculation and also clears the calibration type selections.

If you execute this command before all calibration data needed for calculating the calibration coefficients are written, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Dmy As Long
Dim Data(3) as Variant
Data(0) = -1.123
Data(1) = 2.456
Data(2) = -2.249
Data(3) = 2.608
SCPI.SENSE(1).CORRection.COEFficient.METHod.RESPOnse.THRU = Array(2, 1)
SCPI.SENSE(1).CORRection.COEFficient("ET", 2, 1) = Data
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSE(1).CORRection.COEFficient.SAVE
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COEFficient.DATA
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.ERESponse
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPOnse.OPEN
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPOnse.SHORT
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPOnse.THRU
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT1
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT2
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT3
SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT4
SCPI.SENSE(Ch).CORRection.COLLECT.METHod.TRL2
SCPI.SENSE(Ch).CORRection.COLLECT.METHod.TRL3
```

SCPI.SENSE(Ch).CORRection.COLLEct.METHod.TRL4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFFicient:SAVE

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:SAVE"

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.ISOLation

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.ISOLation = *Ports*

Description

This command measures the calibration data of the isolation from the specified stimulus port to the specified response port, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the response port number. <i>Ports(1)</i>: Specifies the stimulus port number. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLlect.ACQuire.ISOLation = Array(1,2)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim IsPort(1) As Variant
Dim Dmy As Long
IsPort(0) = 1
IsPort(1) = 2
```

```
SCPI.SENSE(1).CORRection.COLLection.ACQuire.ISOLation = IsPort
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Thru)** > **Isolation (Optional)**

Cal > **Calibrate** > **n-Port Cal** > **Isolation (Optional)** > **Port m-n Isol**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection[:ACQuire]:ISOLation <numeric
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ISOL 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.LOAD**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.LOAD = *Port***Description**

This command measures the calibration data of the load standard for the specified port, for the selected channel (*Ch*).

Variable

Parameter	<i>Port</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.LOAD = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key**Cal > Calibrate > Response (Open)|Response (Short) > Load (Optional)****Cal > Calibrate > 1-Port Cal > Load****Cal > Calibrate > n-Port Cal > Reflection > Port m Load****Equivalent SCPI command****Syntax**

:SENSE{[1]-160}:CORRection:COLLection[:ACQuire]:LOAD <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:LOAD 1"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.COLLEct.ACQuire.OLOad.LINE**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLEct.ACQuire.OLOad.LINE = *Standard**Standard* = SCPI.SENSE(*Ch*).CORRection.COLLEct.ACQuire.OLOad.LINE**Description**

This command selects the offset line to be connected with the load for the selected channel (*Ch*).

Variable

Parameter	<i>Standard</i>
Description	Standard number
Data type	Long integer type (Long)
Range	1 to 30
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLEct.ACQuire.OLOad.LINE = 1
```

Related objects**Equivalent key****Cal > Calibrate > Response (Open)|Response (Short) > Load (Optional)****Cal > Calibrate > 1-Port Cal > Load****Cal > Calibrate > n-Port Cal > Reflection > Port m Load****Equivalent SCPI command****Syntax**

:SENSE{[1]-160}:CORRection:COLLEct[:ACQuire]:OLOad:LINE <numeric>

:SENSE{[1]-160}:CORRection:COLLEct[:ACQuire]:OLOad:LINE?

Query response

{numeric}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;".SENS1:CORR:COLL:OLO:LINE 1"  
20 OUTPUT 717;".SENS1:CORR:COLL:OLO:LINE?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.CLEAr**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.CLEAr = *Port***Description**

This command clears the acquired offset load data for the selected port and channel (*Ch*).

Variable

Parameter	<i>Port</i>
Description	Port number for clear
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLlect.ACQuire.OLOad.LOAD.CLEAr = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OLOad.LOAD.DONE
```

Equivalent key**Cal > Calibrate > Enhanced Response > Load > Load n - Offset Load > Clear****Cal > Calibrate > 1-Port Cal > Load > Load n - Offset Load > Clear****Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Offset Load > Clear****Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:CORRection:COLlect[:ACQuire]:OLOad:LOAD:CLEAr
<numeric>
```

Example of use

E5071C

10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD:CLE 1"
30 ENTER 717;A

SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OLOad.LOAD.COUNT(Pt)**Object type**Property (**Read-only**)**Syntax**

Count =
 SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OLOad.LOAD.COUNT(Pt)

Description

This command returns the number of count for the acquired offset calibration data for the offset load standard for the selected port (*Pt*) and channel (*Ch*).

In the other word, number of times SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute is executed. The maximum number of count for the acquired offset calibration data is two. If it is exceeded, the error occurs.

Variable

Parameter	<i>Count</i>
Description	Number of count for the offset calibration acquisition
Data type	Long integer type (Long)
Range	
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Count As Long
Count = SCPI.SENSE(1).CORRection.COLlect.ACQuire.OLOad.LOAD.COUNT(1)
```

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute

Equivalent key

No Equivalent Softkey

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COLlect[:ACQuire]:OLOad:LOAD:COUNT?
<numeric>
```

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;" :SENS1:CORR:COLL:OLO:LOAD:COUN? 1"  
20 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.DONE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.DONE = *Port*

Description

This command finalizes the offset load calibration for the selected channel (*Ch*), then calculate the load correction coefficient data.

Variable

Parameter	<i>Port</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

See SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute

Equivalent key

Cal > Calibrate > Enhanced Response > Load > Load n - Offset Load > Done**Cal > Calibrate > 1-Port Cal > Load > Load n - Offset Load > Done****Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Offset Load > Done**

Equivalent SCPI command

Syntax

```
:SENSE{[1]-160}:CORRection:COLlect[:ACQuire]:OLOad:LOAD:DONE
<numeric>
```

Example of use

E5071C

10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD:DONE 1"
20 OUTPUT 717;""*OPC?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.OLOad.LOAD.EXECute =
Port

Description

This command measures the calibration data of a offset load standard for the specified port, for the selected channel (*Ch*).

Variable

Parameter	<i>Port</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```

Dim Dmy As Long
' Select X11644 Calibration Kit
SCPI.SENSE.CORRection.COLlect.CKIT.SELect = 15
' Select Load 2 in Load
SCPI.SENSE.CORRection.COLlect.ACQuire.SUBClass = 2
' Select Offset Line 1 (Thru). The "Thru" is defined as the standard No. 6 in the Define Standard.
SCPI.SENSE.CORRection.COLlect.ACQuire.OLOad.LINE = 6
MsgBox "Connect Offset Load (Thru)"
' Make a measurement for Offset Line 1
SCPI.SENSE.CORRection.COLlect.ACQuire.OLOad.Load.EXECute = 1
' Wait for measurement end.
Dmy = SCPI.IEEE4882.OPC
' Select Offset Line 2 (X-Band Delay Line). The "X-Band Delay Line" is defined as the standard No. 7 in
the Define Standard.
SCPI.SENSE.CORRection.COLlect.ACQuire.OLOad.LINE = 7
MsgBox "Connect Offset Load (X-Band Delay Line)"
' Make a measurement for Offset Line 2
SCPI.SENSE.CORRection.COLlect.ACQuire.OLOad.Load.EXECute = 1
' Wait for measurement end.
Dmy = SCPI.IEEE4882.OPC

```

E5071C

' Finalize the Offset load measurement.

SCPI.SENSE.CORRection.COLLection.ACQuire.OLOad.LoAd.DONE = 1

Related objects

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OLOAD.LINE

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OLOAD.LOAD.DONE

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OLOAD.LOAD.COUNT

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OLOAD.LOAD.CLEAr

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Enhanced Response** > **Load** > **Load n - Offset Load** > **Offset Line n**

Cal > **Calibrate** > **1-Port Cal** > **Load** > **Load n - Offset Load** > **Offset Line n**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Load** > **Load n - Offset Load** > **Offset Line n**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLection[:ACQuire]:OLOad:LOAD[:EXECute]
<numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD 1"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

Refer to Example for COM.

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.OPEN**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.OPEN = *Port***Description**

This command measures the calibration data of the open standard for the specified port, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see *Ch* and *Port* respectively.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.OPEN = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key**Cal > Calibrate > Response (Open)|1-Port Cal > Open****Cal > Calibrate > n-Port Cal > Reflection > Port m Open****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLLection[:ACQuire]:OPEN <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OPEN 1"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SHORt**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SHORt = *Port***Description**

This command measures the calibration data of the short standard for the specified port, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see *Ch* and *Port*.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.SHORt = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key**Cal > Calibrate > Response (Short)|1-Port Cal > Short****Cal > Calibrate > n-Port Cal > Reflection > Port m Short****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLLection[:ACQuire]:SHORt <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SHOR 1"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SLOad.CLEar**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SLOad.CLEar**Description**

This command clears the acquired sliding load data for the selected channel (*Ch*).

Examples

SCPI.SENSE(1).CORRection.COLLection.ACQuire.SLOad.CLEar

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SLOad.LOAD.EXECuteSCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SLOad.TRLLine.EXECuteSCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SLOad.DONE**Equivalent key**

Cal > Calibrate > Response (Open)|Response (Short) > Load > Load n - Sliding Load > Clear

Cal > Calibrate > Enhanced Response > Load > Load n - Sliding Load > Clear

Cal > Calibrate > 1-Port Cal > Load > Load n - Sliding Load > Clear

Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Sliding Load > Clear

Cal > Calibrate > n-Port TRL Cal > Line/Match > x-y Line/Match | x-y Fwd (sxy) | x-y Rvs (syx) > Line/Match n - Sliding Load > Clear

Equivalent SCPI command**Syntax**

:SENSe{[1]-160}:CORRection:COLLection[:ACQuire]:SLOad:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:SLO:CLE"

SCPI.SENSE(Ch).CORRection.COLLECT.ACQuire.SLOad.COUNT

Object type

Property (**Read-only**)

Syntax

Count = SCPI.SENSE(*Ch*).CORRection.COLLECT.ACQuire.SLOad.COUNT

Description

This command returns the number of count for the acquired sliding calibration data for the sliding load standard for the selected channel (*Ch*).

In the other word, the number of times

SCPI.SENSE(*Ch*).CORRection.COLLECT.ACQuire.SLOad.LOAD.EXECute or SCPI.SENSE(*Ch*).CORRection.COLLECT.ACQuire.SLOad.TRLLine.EXECute is executed. The maximum number of count for the acquired offset calibration data is ten. If it is exceeded, the error occurs.

Variable

Parameter	<i>Count</i>
Description	Number of count for the sliding calibration acquisition
Data type	Long integer type (Long)
Range	
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Count As Long
Count = SCPI.SENSE(1).CORRection.COLLECT.ACQuire.SLOad.COUNT
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLECT.ACQuire.SLOad.LOAD.EXECuteSCPI.SENSE(*Ch*).CORRection.COLLECT.ACQuire.SLOad.TRLLine.EXECute

Equivalent key

No Equivalent Softkey

Equivalent SCPI command

Syntax

:SENSE{[1]-160}:CORRection:COLLECT[:ACQuire]:SLOad:COUNT?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;" :SENS1:CORR:COLL:SLO:COUN?"
20 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.DONE**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.DONE**Description**

This command finalizes the sliding load calibration for the selected channel (*Ch*), then calculate the load correction coefficient data.

Examples

See SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.LOAD.EXECute

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.LOAD.EXECute

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.TRLLine.EXECute

SCPI.IEEE4882.OPC

Equivalent key

Cal > Calibrate > Response (Open)|Response (Short) > Load > Load n - Sliding Load > Done

Cal > Calibrate > Enhanced Response > Load > Load n - Sliding Load > Done

Cal > Calibrate > 1-Port Cal > Load > Load n - Sliding Load > Done

Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Sliding Load > Done

Cal > Calibrate > n-Port TRL Cal > Line/Match > x-y Line/Match | x-y Fwd (sxy) | x-y Rvs (syx) > Line/Match n - Sliding Load > Done

Equivalent SCPI command**Syntax**

:SENSe{[1]-160}:CORRection:COLlect[:ACQuire]:SLOad:DONE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:DONE"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.LOAD.EXECute**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.EXECute = *Port***Description**

This command measures the calibration data of a sliding load standard for the specified port, for the selected channel (*Ch*). From 5 to 10 times of the sliding measurement is required.

Variable

Parameter	<i>Port</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
' Select the 85052B Calibration Kit
SCPI.SENSE.CORRection.COLlect.CKIT.SELect = 19
' Select Load 5 (Sliding Load (m)) in load
SCPI.SENSE.CORRection.COLlect.ACQuire.SUBClass = 5
For i = 1 To 5
    MsgBox "Set Sliding Load position"
    ' Make a measurement for sliding load calibration
    SCPI.SENSE.CORRection.COLlect.ACQuire.SLOad.LOAD.EXECute = 1
    Dmy = SCPI.IEEE4882.OPC
Next i
SCPI.SENSE.CORRection.COLlect.ACQuire.SLOad.DONE
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.SLOad.DONE
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.SLOad.COUNT
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.SLOad.CLEAr
SCPI.IEEE4882.OPC
```

Equivalent key

Cal > Calibrate > Response (Open)|Response (Short) > Load > Load n - Sliding Load > Sliding Load

Cal > Calibrate > Enhanced Response > Load > Load n - Sliding Load > Sliding Load

Cal > Calibrate > 1-Port Cal > Load > Load n - Sliding Load > Sliding Load

Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Sliding Load > Sliding Load

Equivalent SCPI command

Syntax

**:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SLOad:LOAD[:EXECute]
<numeric>**

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:LOAD 1"  
20 OUTPUT 717;"*OPC?"  
30 ENTER 717;A
```

See Example for COM

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.TRLLine.EXECute

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.SLOad.TRLLine.EXECute =
Ports

Description

This command measures the calibration data of a sliding load standard for TRL line from for the specified ports, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the input port. <i>Ports(1)</i>: Specifies the output port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLlect.ACQuire.SLOad.TRLLine.EXECute = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim Port(1) As Variant, Dmy As Long
Port(0) = 2
Port(1) = 1
SCPI.SENSE(1).CORRection.COLlect.ACQuire.SLOad.TRLLine.EXECute = Port
Dmy = SCPI.IEEE4882.OPC
```

Related objects

E5071C

SCPI.SENSE(Ch).CORRection.COLLEct.ACQuire.SLOad.LOAD.EXECute
SCPI.SENSE(Ch).CORRection.COLLEct.ACQuire.SLOad.DONE
SCPI.SENSE(Ch).CORRection.COLLEct.ACQuire.SLOad.CLEar
SCPI.SENSE(Ch).CORRection.COLLEct.ACQuire.SLOad.COUNT
SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **n-Port TRL Cal** > **Line/Match** > **x-y Line/Match | x-y Fwd (sxy) | x-y Rvs (syx) > Line/Match n - Sliding Load**

Equivalent SCPI command

Syntax

:SENSe{[1]-
160}:CORRection:COLLEct[:ACQuire]:SLOad:TRLLine[:EXECute] <numeric
(input port)>,<numeric (output port)>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:TRLL 2,1"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SUBClass**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SUBClass = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.SUBClass**Description**

This command sets/gets the subclass which is used for the calibration of the selected channel (*Ch*). For example, if two different subclasses are set in advance, say Thru 1 & Thru 2, which are visible at the calibration softkey, this command can select either Thru1 or Thru2. When performing Thru cal, either Thru 1 or Thru 2 set with this command is used for the calibration.

Variable

Parameter	<i>Value</i>
Description	The setting number of the standard subclass for the calibration.
Data type	Long integer type (Long)
Range	1 to 30
Preset value	1
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Subc As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.SUBClass = 3
Subc = SCPI.SENSE(1).CORRection.COLLection.ACQuire.SUBClass
```

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.ORDER.SELECT**Equivalent key**

E5071C

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SUBClass <numeric>  
:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SUBClass?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SUBC 1"
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.THRU

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.THRU = *Ports*

Description

This command measures the calibration data of the Thru standard from the specified stimulus port to the specified response port, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the input port. <i>Ports(1)</i>: Specifies the output port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.THRU = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim ThruPort(1) As Variant
Dim Dmy As Long
ThruPort(0) = 2
ThruPort(1) = 1
SCPI.SENSE(1).CORRection.COLLection.ACQuire.THRU = ThruPort
Dmy = SCPI.IEEE4882.OPC
```

E5071C

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Thru)** > **Thru**

Cal > **Calibrate** > **n-Port Cal** > **Transmission** > **Port m-n Thru**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:THRU <numeric
1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:THRU 1,2"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLLine

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLLine = *Ports*

Description

This command executes LINE or MATCH measurement of the TRL calibration for the selected calibration kit, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the input port. <i>Ports(1)</i>: Specifies the output port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.TRLLine = Array(1,2)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim Trll(1) As Variant
Dim Dmy As Long
Trll(0) = 1
Trll(1) = 2
SCPI.SENSE(1).CORRection.COLLection.ACQuire.TRLLine = Trll
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLReflect

SCPI.SENSE(Ch).CORRection.COLLEct.ACQuire.TRLThru

Equivalent key

Cal > **Calibrate** > **2-Port TRL Cal** > **Line/Match** > **x-y Line/Match|x-y Fwd (Syx)|x-y Rvs (Sxy)**

Cal > **Calibrate** > **3-Port TRL Cal** > **Line/Match** > **x-y Line/Match|x-y Fwd (Syx)|x-y Rvs (Sxy)|x-z Line/Match|x-z Fwd (Szx)|x-z Rvs (Sxz)|y-z Line/Match|y-z Fwd (Szy)|y-z Rvs (Syz)**

Cal > **Calibrate** > **4-Port TRL Cal** > **Line/Match** > **x-y Line/Match|x-y Fwd (Syx)|x-y Rvs (Sxy) > x-z Line/Match|x-z Fwd (Szx)|x-z Rvs (Sxz) > x-w Line/Match|x-w Fwd (Swx)|x-w Rvs (Sxw) > y-z Line/Match|y-z Fwd (Szy)|y-z Rvs (Syz) > y-w Line/Match|y-w Fwd (Swy)|y-w Rvs (Syw) > z-w Line/Match|z-w Fwd (Swz)|z-w Rvs (Szw)**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLEct[:ACQuire]:TRLLine <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:TRLL 1,2"
20 OUTPUT 717;""OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLReflect

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLReflect = *Port*

Description

This command executes the reflection measurement of the TRL calibration for the selected calibration kit, for the selected channel (*Ch*).

Variable

Parameter	<i>Port</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLLection.ACQuire.TRLReflect = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLLineSCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLThru

Equivalent key

Cal > Calibrate > 2-Port TRL Cal > Reflect > Portx Reflect|Porty Reflect**Cal > Calibrate > 3-Port TRL Cal > Reflect > Portx Reflect|Porty Reflect|Portz Reflect****Cal > Calibrate > 4-Port TRL Cal > Reflect > Portx Reflect|Porty Reflect|Portz Reflect|Portw Reflect**

E5071C

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:TRLReflect <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:TRLR 1"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.TRLThru

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.TRLThru = *Ports*

Description

This command executes the THRU measurement of the TRL calibration for the selected calibration kit, for the selected channel (*Ch*).

- Both forward and backward way are required .

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the input port. <i>Ports(1)</i>: Specifies the output port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLlect.ACQuire.TRLThru = Array(1,2)
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSE(1).CORRection.COLlect.ACQuire.TRLThru = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
```

```
Trlt1(0) = 1
Trlt1(1) = 2
```

E5071C

Trlt2(0) = 2

Trlt2(1) = 1

SCPI.SENSE(1).CORRection.COLLection.ACQuire.TRLThru = Trlt1

Dmy = SCPI.IEEE4882.OPC

SCPI.SENSE(1).CORRection.COLLection.ACQuire.TRLThru = Trlt2

Dmy = SCPI.IEEE4882.OPC

Related objects

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.TRLLine

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.TRLReflect

Equivalent key

Cal > **Calibrate** > **2-Port TRL Cal** > **Thru/Line** > **Port x-y Thru**

Cal > **Calibrate** > **3-Port TRL Cal** > **Thru/Line** > **Port x-y Thru|Port x-z Thru|Port y-z Thru**

Cal > **Calibrate** > **4-Port TRL Cal** > **Thru/Line** > **Port x-y Thru|Port x-z Thru|Port x-w Thru|Port y-z Thru|Port y-w Thru|Port z-w Thru**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLection[:ACQuire]:TRLThru <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:TRLT 1,2"
20 OUTPUT 717;"*OPC?"
30 OUTPUT 717;":SENS1:CORR:COLL:TRLT 2,1"
40 OUTPUT 717;"*OPC?"
50 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).LENGth

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).LENGth=*length*

Description

This command sets/displays the approximate length of the adaptor, for the selected channel (*Ch*) and for the selected port.

Variable

Parameter	<i>length</i>
Description	Adapter Length
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Note	Adapter length is positive for adaptor removal and negative for adaptor insertion.

Examples

SCPI.SENSE(1).CORRection.COLlect.ADAPter(2).LENGth = 0.01

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.METHod.ADAPter.REMOval

Equivalent key

Cal > Calibrate > Adapter Removal > Adapter Length

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLlect:ADAPter{[1]-4}:LENGth <+ or -
><value of length>
```

Query response

{numeric}<newline><^END>

Example of use

E5071C

10 OUTPUT 717;":SENS1:CORR:COLL:ADAP2:LENG 0.01"
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP2:LENG?"
30 ENTER 717;A\$

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).ROTate**Object type**Method (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).ROTate

Description

This command executes Adapter Removal/Insertion along with moving the phase of adapter (which is removed or inserted) to 180 degrees. This command is useful in cases where auto judgement of phase fails. This command can be executed several times while the calibration remains valid.

- If user cannot execute this command, then "Execution error" is displayed.

Examples

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).ROTate

Related Object

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVEguide.CUToff

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVEguide.LENGth

Equivalent key**Cal > Calibrate > Adapter Removal > Rotate Adapter****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLlect:ADAPter{[1]-4}:ROTate

Example of use

10 OUTPUT 717;"SENS1:CORR:COLL:ADAP2:ROT"

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVEguide.CUTOff

Object type

Property (**Read-Write**)

Syntax

```
SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVEguide.CUTOff =
Cfreq
```

```
Cfreq =
```

```
SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVEguide.CUTOff
```

Description

This command sets/gets approximate cutoff frequency of the waveguide, for the selected channel (Ch) and for the selected port.

Variable

Parameter	<i>Cfreq</i>
Description	Waveguide cutoff frequency
Data type	Double precision floating point type (Double)
Range	0 to 20E9
Unit	Hz
Preset value	0 Hz

Examples

```
Dim CFreq As Double
```

```
SCPI.SENSE(1).CORRection.COLlect.ADAPter(2).WAVEguide.CUTOff = 700
```

```
CFreq= SCPI.SENSE(1).CORRection.COLlect.ADAPter(2).WAVEguide.CUTOff
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).ROTate
```

```
SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVEguide.LENGth
```

Equivalent key

Cal > Calibrate > Adapter Removal > Cutoff Frequency

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLlect:ADAPter{[1]-4}:WAVEguide:CUTOff <numeric>
```

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:WAVeguide:CUToff?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:CUT 2000"  
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:CUT?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVeguide.LENGth

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVeguide.LENGth =
length

length =

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVeguide.LENGth

Description

This command sets/gets approximate adapter length of the waveguide, for the selected channel (Ch) and for the selected port.

Variable

Parameter	<i>length</i>
Description	Waveguide Adapter Length
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Preset value	0 second
Note	Adapter length is positive for adaptor removal and negative for adaptor insertion.

Examples

```
Dim CLen As Double
SCPI.SENSE(1).CORRection.COLlect.ADAPter(2).WAVeguide.LENGth = -5
CLen= SCPI.SENSE(1).CORRection.COLlect.ADAPter(2).WAVeguide.LENGth
```

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).ROTate

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).WAVeguide.CUToff

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Waveguide Length**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-
4}:WAVeguide:LENGth <numeric>
:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-
4}:WAVeguide:LENGth?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:LEN 5"
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:LEN?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.LABel = *Lbl**Lbl* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.LABel

Description

This command sets/gets the calibration kit name for the calibration kit selected for channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Lbl</i>
Description	Calibration kit name
Data type	Character string type (String)
Range	254 characters or less
Preset value	<p>Varies depending on the calibration kit number:</p> <ul style="list-style-type: none"> • 85031B • 85032B/E • 85032F • 85033D • 85033E • 85036B • 85036E • 85038A/F/M • 85039B • 85050C • 85050D • 85052D • 85052C • 85054D • 85056D

- 85056K
- K11644A
- P11644A
- X11644A
- <User Defined Cal Kits>

Examples

```
Dim CalLbl As String
SCPI.SENSE(1).CORRection.COLLection.CKIT.LABel = "User 1"
CalLbl = SCPI.SENSE(1).CORRection.COLLection.CKIT.LABel
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLLection.CKIT.SELect
```

Equivalent key

Cal > Modify Cal Kit > Label Kit

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COLLection:CKIT:LABel <string>
:SENSe{[1]-160}:CORRection:COLLection:CKIT:LABel?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:LAB ""MY_KIT""
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:LAB?"
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.LOAD(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.LOAD(*Cpt*) = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.LOAD(*Cpt*)

Description

This command sets/gets the standard used for the load measurement of the specified port (*Cpt*), for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	<i>Cpt</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples


```
Dim StanLoad As Long
SCPI.SENSE(1).CORRection.COLLEct.CKIT.ORDer.LOAD(1) = 10
StanLoad = SCPI.SENSE(1).CORRection.COLLEct.CKIT.ORDer.LOAD(1)
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.SELect
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.OPEN(Cpt)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.SELect
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.SHORT(Cpt)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.THRU(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLReflect
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)
```

Equivalent key

Cal > Modify Cal Kit > Specify CLSs > Load > Set All|Port 1|Port 2|Port 3|Port 4

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:ORDer:LOAD <numeric
1>,<numeric 2>
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:ORDer:LOAD? <numeric 1>
```

Query response

```
{numeric 2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:LOAD 1,9"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:LOAD? 1"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt) = *Value**Value* = SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)

Description

This command sets/gets the standard used for the open measurement of the specified port (*Cpt*), for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Examples

```
Dim StanOpen As Long
SCPI.SENSE(1).CORRection.COLlect.CKIT.ORDER.OPEN(1) = 10
StanOpen = SCPI.SENSE(1).CORRection.COLlect.CKIT.ORDER.OPEN(1)
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THUR(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLReflect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)
```

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Open** > **Port 1|Port 2|Port 3|Port 4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:OPEN <numeric
1>,<numeric 2>
```

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:OPEN? <numeric 1>
```

Query response

```
{numeric 2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:OPEN 1,2"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:OPEN? 1"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect = *Value**Value* = SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect

Description

This command sets/gets a subclass in the subclass menu. When sending this command, it selects the subclass from 1 to 8.

Variable

Parameter	<i>Value</i>
Description	The number of the standard subclass for the calibration.
Data type	Long integer type (Long)
Range	1 to 8
Preset Value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim CKitSel As Long
SCPI.SENSE(1).CORRection.COLlect.CKIT.ORDER.SELect = 3
CKitSel = SCPI.SENSE(1).CORRection.COLlect.CKIT.ORDER.SELect
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.SUBClass
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THRU(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLReflect
```

SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > Modify Cal Kit > Specify CLSs > Sub Class > Sub Class 1 | ... | Sub Class 8

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:ORDer:SElect <numeric>
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:ORDer:SElect?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SEL 1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SEL?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.SHORt(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.SHORt(*Cpt*) = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.SHORt(*Cpt*)

Description

This command sets/gets the standard used for the short measurement of the specified port (*Cpt*), for the calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Examples

```
Dim StanShor As Long
SCPI.SENSE(1).CORRection.COLlect.CKIT.ORDER.SHORt(1) = 10
StanShor = SCPI.SENSE(1).CORRection.COLlect.CKIT.ORDER.SHORt(1)
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THUR(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLReflect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)
```

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Short** > **Port 1|Port 2|Port 3|Port 4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:SHORt <numeric
1>,<numeric 2>
```

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:SHORt? <numeric 1>
```

Query response

```
{numeric 2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SHOR 1,1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SHOR? 1"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.ORDER.THURU(*Cpt_m*,*Cpt_n*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.ORDER.THURU(*Cpt_m*,*Cpt_n*) =
Value

Value =

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.ORDER.THURU(*Cpt_m*,*Cpt_n*)

Description

This set/get the standard used for the thru measurement between the specified 2 ports (*Cpt_m* and *Cpt_n*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Cpt_m</i> , <i>Cpt_n</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variables (*Cpt_m* and *Cpt_n*) have no preset value, you cannot omit them. If you omit the variables (*Cpt_m* and *Cpt_n*) or if you specify the same port number to 2 port numbers, an error occurs when executed. Notice that when you specify 2 ports with the variables (*Cpt_m* and *Cpt_n*), the order of the 2 port numbers is arbitrary.

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)

Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanThru As Long
SCPI.SENSE(1).CORRection.COLLEct.CKIT.ORDer.THru(1,2) = 10
StanThru = SCPI.SENSE(1).CORRection.COLLEct.CKIT.ORDer.THru(1,2)
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.SELect
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.LOAD(Cpt)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.OPEN(Cpt)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.SELect
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.SHORT(Cpt)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLReflect
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)
```

Equivalent key

Cal > Modify Cal Kit > Specify CLSs > Thru > Port 1-2|Port 1-3|Port 1-4|Port 2-3|Port 2-4|Port 3-4

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:ORDer:THRU <numeric 1>,<numeric 2>,<numeric 3>
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:ORDer:THRU? <numeric 1>,<numeric 2>
```

Query response

```
{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:THRU 1,2,11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:THRU? 1,2"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(*Cpt_m*,*Cpt_n*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.TRLLine(*Cpt_m*,*Cpt_n*)
= *Value*

Value =

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.TRLLine(*Cpt_m*,*Cpt_n*)

Description

This command sets/gets the standard used for the line measurement of TRL calibration between the specified 2 ports (*Cpt_m* and *Cpt_n*), for the calibration kit selected for selected channel (*Ch*).

- If the standard number is 0, the standard is invalid for the subclass.

Variable

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanTrll As Long
SENSe(1).CORRection.COLlect.CKIT.ORDER.TRLLine(1,2) = 10
StanTrll = SENSe(1).CORRection.COLlect.CKIT.ORDER.TRLLine(1,2)
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.SELect
 SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.LOAD(*Cpt*)
 SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.OPEN(*Cpt*)
 SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.SELect
 SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.SHORT(*Cpt*)
 SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.THURU(*Cpt_m*,*Cpt_n*)

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLReflect

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > Modify Cal Kit > Specify CLSs > TRL Line/Match > Set All|Port 1-2|Port 1-3|Port 1-4|Port 2-3|Port 2-4|Port 3-4

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLection:CKIT:ORDer:TRLLine <numeric 1>,<numeric 2>,<numeric 3>

:SENSe{[1]-160}:CORRection:COLLection:CKIT:ORDer:TRLLine? <numeric 1>,<numeric 2>

Query response

{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRL 1,2,11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRL? 1,2"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.TRLReflect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.TRLreflect = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.TRLreflect

Description

This command sets/gets the standard used for the reflection measurement of the TRL calibration between the specified 2 ports, for the calibration kit selected for selected channel (*Ch*).

- If the standard number is 0, the standard is invalid for the subclass.

Variable

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanTrlr As Long
SENSe(1).CORRection.COLlect.CKIT.ORDER.TRLReflect = 5
StanTrlr = SENSe(1).CORRection.COLlect.CKIT.ORDER.TRLReflect
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THRU(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)
```

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **TRL Reflect**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLReflect
<numeric>
```

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLReflect?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLR 11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLR?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)
= *Value*

Value =

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)

Description

This command sets/gets the standard for the THRU measurement of the TRL calibration between the specified 2 ports (*Cpt_m* and *Cpt_n*).

- If the standard number is 0, the standard is invalid for the subclass.

Variable

Parameter	<i>Value</i>
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanTrlt As Long
SENSe(1).CORRection.COLlect.CKIT.ORDER.TRLThru(1,2) = 3
StanTrlt = SENSe(1).CORRection.COLlect.CKIT.ORDER.TRLThru(1,2)
```

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Cpt)
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Cpt)
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THRU(Cpt_m,Cpt_n)
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Cpt_m,Cpt_n)
 SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLReflect

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **TRL Thru** > **Set All|Port 1-2|Port 1-3|Port 1-4|Port 2-3|Port 2-4|Port 3-4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLThru <numeric 1>, <numeric 2>, <numeric 3>
```

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLThru? <numeric 1>,<numeric 2>
```

Query response

```
{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLT 1,2,5"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLT? 1,2"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.RESet

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.RESet

Description

This command resets the calibration kit selected for selected channel (*Ch*) to the default factory setting state.

Examples

SCPI.SENSE(1).CORRection.COLLection.CKIT.RESet

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLection:CKIT:RESet

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:RES"

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect = *Value**Value* = SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect**Description**This command sets/gets the calibration kit of selected channel (*Ch*).**Variable**

Parameter	<i>Value</i>
Description	Number of calibration kit
Data type	Long integer type (Long)
Range	1 to 30
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim CalKit As Long
SCPI.SENSE(1).CORRection.COLlect.CKIT.SELect = 3
CalKit = SCPI.SENSE(1).CORRection.COLlect.CKIT.SELect
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Cpt)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THRU(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLReflect
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Cpt_m,Cpt_n)
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).DELay
```

E5071C

Equivalent key

Cal > **Cal Kit**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT[:SElect] <numeric>  
:SENSe{[1]-160}:CORRection:COLLect:CKIT[:SElect]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT 3"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).ARBitrary

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).ARBitrary = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).ARBitrary

Description

This command sets/gets the value of the arbitrary impedance of the standards (*Std*), for the calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	<i>Std</i>
Description	Standard number
Data type	Long integer type (Long)
Range	1 to 21
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Value</i>
Description	Value of arbitrary impedance
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.

Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanArbt As Double
SCPI.SENSE(1).CORRection.COLLEct.CKIT.STAN(5).ARBitrary = 50.5
StanArbt = SCPI.SENSE(1).CORRection.COLLEct.CKIT.STAN(5).ARBitrary
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.SELect
```

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Arb. Impedance**

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:STAN{[1]-30}:ARBitrary
<numeric>
:SENSe{[1]-160}:CORRection:COLLEct:CKIT:STAN{[1]-30}:ARBitrary?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:ARB 50.5"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:ARB?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C0**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C0 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C0**Description**

This command sets/gets the value of the C0 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	C0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	fF (femto farad): 1E-15 F (farad)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanC0 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C0 = 12.3

StanC0 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C0

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect**Equivalent key****Cal > Modify Cal Kit > Define STDs > no. name > C0****Equivalent SCPI command****Syntax**

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C0
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C0?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C0 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C0?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C1 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C1

Description

This command sets/gets the value of the C1 value of the standards (*Std*), for the selected calibration kit and channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	C1
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-27 F/Hz (1E-27 farad / hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanC1 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C1 = 12.3

StanC1 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C1

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C1**

Equivalent SCPI command

E5071C

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C1
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C1?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C1 12.3"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C1?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C2**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C2 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C2**Description**

This command sets/gets the value of the C2 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	C2
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-36 F/Hz ² (1E-36 farad /hertz ²)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanC2 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C2 = 12.3

StanC2 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C2

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect**Equivalent key****Cal > Modify Cal Kit > Define STDs > no. name > C2****Equivalent SCPI command****Syntax**

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C2
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C2?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C2 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C2?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C3 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).C3

Description

This command sets/gets the value of the C3 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	C3
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-45 F/Hz ³ (1E-45 farad / hertz ³)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanC3 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C3 = 12.3

StanC3 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).C3

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C3**

Equivalent SCPI command

Syntax

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C3
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C3?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C3 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C3?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).CHARacter

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).CHARacter = *Param**Param* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).CHARacter

Description

This command sets/gets the media type of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Select media type of standard.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "COAXial": Selects coaxial as the media type. • "WAVEguide": Selects waveguide as the media type.
Preset value	"COAXial"

Examples

```
Dim StanChar As Double
SCPI.SENSE(1).SENSe.CORRection.COLlect.CKIT.STAN.CHARacter = "WAVEguide"
StanChar = SCPI.SENSE.CORRection.COLlect.CKIT.STAN.CHARacter
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).DELay

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Media**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLlect:CKIT:STAN{[1]-30}:CHARacter
{COAXial|WAVEguide}
```

```
:SENSe{[1]-160}:CORRection:COLlect:CKIT:STAN{[1]-30}:CHARacter?
```

E5071C

Query response

{COAXial|WAVeguide}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:CHAR WAV"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:CHAR?"  
30 ENTER 717;A$
```

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).DELay**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).DELay = *Value**Value* = SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).DELay**Description**

This command sets/gets the value of the offset delay of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Offset delay
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanDel As Double

SCPI.SENSE(1).CORRection.COLlect.CKIT.STAN(5).DELay = 12.3

StanDel = SCPI.SENSE(1).CORRection.COLlect.CKIT.STAN(5).DELay

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELECT

SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).CHARacter

Equivalent key**Cal > Modify Cal Kit > Define STDs > no. name > Offset Delay****Equivalent SCPI command**

E5071C

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:DELay
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:DELay?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:DEL 12.3"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:DEL?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).FMAXimum

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).FMAXimum = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).FMAXimum

Description

This command sets/gets the value of the stop frequency of the standard (*Std*), for the calibration kit selected for selected channel (*Ch*).

- When media type of standard is "Waveguide", sets the stop frequency of the cutoff frequency.

Variable

Parameter	<i>Value</i>
Description	Stop frequency of the selected standard.
Data type	Double precision floating point type (Double)
Range	0 to 999E9
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanFMax As Double

SCPI.SENSE(1).CORRection.COLlect.CKIT.STAN(5).FMAXimum = 13.2E9

StanFMax = SCPI.SENSE(1).CORRection.COLlect.CKIT.STAN(5).FMAXimum

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).FMINimum

Equivalent key

[Cal] > Modify Cal Kit > Define STDs > no. name > Max. Frequency

E5071C

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMAXimum
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMAXimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMAX 5E9"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMAX?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).FMINimum

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).FMINimum = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).FMINimum

Description

This command sets/gets the value of the start frequency of the standard (*Std*), for the selected calibration kit and channel (*Ch*).

- When media type of standard is "Waveguide", sets the start frequency of the cutoff frequency.

Variable

Parameter	<i>Value</i>
Description	Start frequency of the selected standard.
Data type	Double precision floating point type (Double)
Range	0 to 999E9
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanFMin As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).FMINimum = 600E6

StanFMax = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).FMINimum

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).FMAXimum

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Min. Frequency**

E5071C

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMINimum
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMINimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMIN 1E9"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMIN?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L0 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L0

Description

This command sets/gets the value of the L0 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	L0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	pH (pico henry)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanL0 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L0 = 12.3

StanL0 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L0

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > Modify Cal Kit > Define STDs > no. name > L0

Equivalent SCPI command

Syntax

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L0
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L0?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L0 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L0?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L1 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L1

Description

This command sets/gets the value of the L1 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	L1
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-24 H/Hz (1E-24 henry / hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanL1 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L1 = 12.3

StanL1 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L1

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > Modify Cal Kit > Define STDs > no. name > L1

Equivalent SCPI command

Syntax

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L1
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L1?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L1 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L1?"
30 ENTER 717;A

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).L2

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).L2 = *Value**Value* = SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).L2

Description

This command sets/gets the value of the L2 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	L2
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-33 H/Hz ² (1E-33 henry / hertz ²)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanL2 As Double
SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L2 = 12.3
StanL2 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L2
```

Related objects

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L2**

Equivalent SCPI command

E5071C

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L2
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L2?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L2 12.3"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L2?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L3 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).L3

Description

This command sets/gets the value of the L3 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	L3
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-42 H/Hz ³ (1E-42 henry / hertz ³)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanL3 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L3 = 12.3

StanL3 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).L3

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > Modify Cal Kit > Define STDs > no. name > L3

Equivalent SCPI command

Syntax

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L3
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L3?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L3 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L3?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).LABel = *Lbl**Lbl* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).LABel

Description

This command sets/gets the name of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Lbl</i>
Description	Standard name
Data type	Character string type (String)
Range	254 characters or less
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanLbl As Double
SCPI.SENSE(1).CORRection.COLlect.CKIT.STAN(5).LABel = "OPEN 3.5mm"
StanLbl = SCPI.SENSE(1).CORRection.COLlect.CKIT.STAN(5).LABel
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.SELECT

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Label**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLlect:CKIT:STAN{[1]-30}:LABel
<string>
:SENSe{[1]-160}:CORRection:COLlect:CKIT:STAN{[1]-30}:LABel?
```

Query response

{string}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LAB ""OPEN"""  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LAB?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).LOSS

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).LOSS = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).LOSS

Description

This command sets/gets the value of the offset loss of the standards 1 to 21 (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Offset loss
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	ohm/s
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanLoss As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).LOSS = 12.3

StanLoss = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).LOSS

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > Modify Cal Kit > Define STDs > no. name > Offset Loss

Equivalent SCPI command

Syntax

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-21}:LOSS
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-21}:LOSS?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LOSS 12.3"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LOSS?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).LTYPe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).LTYPe = *Type**Type* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).LTYPe

Description

This command sets/gets the length type for the load standard of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Type</i>
Description	Length type
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "FIXed": Normal load • "SLIDing": Sliding load • "OFFSet": Offset load.
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanType As String
SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).LTYPe = "SLID"
StanType = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).LTYPe
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Length Type**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection:CKIT:STAN{[1]-30}:LTYPe
{FIXed|SLIDing|OFFSet}
:SENSe{[1]-160}:CORRection:COLLection:CKIT:STAN{[1]-30}:LTYPe?
```

E5071C

Query response

{FIX|SLID|OFFS}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LTYP SLID"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LTYP?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLLEct.CKIT.STAN(*Std*).TYPE**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLEct.CKIT.STAN(*Std*).TYPE = *Param**Param* = SCPI.SENSE(*Ch*).CORRection.COLLEct.CKIT.STAN(*Std*).TYPE**Description**

This command sets/gets the standard type of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Standard type
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "OPEN": Specifies open. • "SHORT": Specifies short. • "LOAD": Specifies load. • "THRU": Specifies Thru. • "UTHRu": Specifies Unknown Thru. • "ARBI": Specifies arbitrary impedance. • "NONE": Specifies DUT of which theoretical value is 0.
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanType As String
SCPI.SENSE(1).CORRection.COLLEct.CKIT.STAN(5).TYPE = "OPEN"
StanType = SCPI.SENSE(1).CORRection.COLLEct.CKIT.STAN(5).TYPE
```

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLEct.CKIT.SELect**Equivalent key****Cal** > **Modify Cal Kit** > **Define STDs** > **no. name** > **STD Type**

E5071C

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:TYPE  
{OPEN|SHORT|LOAD|THRU|UTHRu|ARBI|NONE}  
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:TYPE?
```

Query response

```
{OPEN|SHORT|LOAD|THRU|UTHRu|ARBI|NONE}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TYPE OPEN"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TYPE?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).Z0**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).Z0 = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).Z0**Description**

This command sets/gets the value of the offset Z0 of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Offset Z0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim StanZ0 As Double

SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).Z0 = 50

StanZ0 = SCPI.SENSE(1).CORRection.COLLection.CKIT.STAN(5).Z0

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.SELect**Equivalent key****Cal > Modify Cal Kit > Define STDs > no. name > Offset Z0****Equivalent SCPI command****Syntax**

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:Z0
<numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:Z0?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:Z0 50"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:Z0?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.IMPedance**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.IMPedance = *Param**Param* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.IMPedance**Description**

This command sets/gets the reference impedance during the TRL calibration, for the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Selects the reference impedance during the TRL calibration.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> "SYSTEM": Calculate the error coefficients by setting the system impedance to the reference impedance. "LINE": Calculate the error coefficients by setting the characteristic impedance of the line standard to the reference impedance.
Preset value	"SYSTEM"

Examples

```
Dim TrlImp As String
SCPI.SENSE(1).CORRection.COLlect.CKIT.TRLoption.IMPedance = "LINE"
TrlImp = SCPI.SENSE(1).CORRection.COLlect.CKIT.TRLoption.IMPedance
```

Related objectsSCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.RPLane**Equivalent key****Cal > Modify Cal Kit > TRL Option > Impedance****Equivalent SCPI command****Syntax**

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLoption:IMPedance
{LINE|SYSTem}

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLoption:IMPedance?

Query response

{LINE|SYST}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:IMP LINE"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:IMP?"  
30 ENTER 717;A$
```


SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.RPLane**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.RPLane = *Param**Param* = SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.RPLane**Description**

This command sets/gets the calculation method of the calibration plane, for the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Selects the calculation method of the calibration plane.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "THRU": Uses the length of the THRU and LINE standard to calculate the calibration plane. • "REFlect": Uses the reflection coefficient of the reflection standard to calculate the calibration plane.
Preset value	"THRU"

Examples

```
Dim TrlRpl As String
SCPI.SENSE(1).SENSe(1).CORRection.COLlect.CKIT.TRLoption.RPLane = "REFlect"
TrlRpl = SCPI.SENSE(1).CORRection.COLlect.CKIT.TRLoption.RPLane
```

Related objectsSCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.TRLoption.IMPedance**Equivalent key****Cal > Modify Cal Kit > TRL Option > Reference Plane****Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:CORRection:COLlect:CKIT:TRLoption:RPLane
{THRU|REFlect}
```

E5071C

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLOption:RPLANE?

Query response

{THRU|REFL}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:RPL REFL"  
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:RPL?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLlect.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.CLEar

Description

This command clears the calibration measurement data when the frequency offset feature is off, for the selected channel (*Ch*).

- Settings that have been temporarily changed due to measurement for each standard (number of traces, measurement parameters, and so on) return to their original values.

Example of use

SCPI.SENSE(1).CORRection.COLlect.CLEar

Related objects

SCPI.SENSE(*Ch*).OFFSet.STATe

Equivalent key

Cal > Calibrate > Responce(Open) > Cancel > OK**Cal > Calibrate > Responce(Short) > Cancel > OK****Cal > Calibrate > Responce(Thru) > Cancel > OK****Cal> Calibrate > 1-Port Cal > Cancel > OK****Cal> Calibrate > 2-Port Cal > Cancel > OK****Cal> Calibrate > 3-Port Cal > Cancel > OK****Cal> Calibrate > 4-Port Cal > Cancel > OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLlect:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CLE"

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.CCHeck.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.CCHeck.ACQuire

Description

This command executes the confidence check of the calibration coefficients for selected channel (*Ch*), using ECal (Electronic Calibration). In other words, this command sets the data measured with the analyzer and the data stored in ECal so that they can be compared).

- If you execute this object when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

This function is available with the Firmware revision 3.50 or greater.

Example of use

SCPI.SENSE(1).CORRection.COLLect.ECAL.CCHeck.ACQuire

Equivalent key

Cal > ECal > Confidence Check

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:CCHeck[:ACQuire]

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:CCH"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.ECAL.ERESponse**Object type**Property (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COLlect.ECAL.ERESponse = Eports

Description

This command executes enhanced response calibration between the two specified ports of selected channel using the ECal (Electrical Calibration) module.

- If you execute this command when the ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	<i>Eports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> • <i>EPorts(0)</i>: Specifies the response port. • <i>EPorts(1)</i>: Specifies the stimulus port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESport(1) As Variant
ERESport(0) = 1
ERESport(1) = 2
SCPI.SENSE(1).CORRection.COLlect.ECAL.ERESponse = ERESport
```

Equivalent key

E5071C

Cal > ECal > Enhanced Response > 2-1 (S21 S11)|3-1 (S31 S11)|...|3-4 (S34 S44)

Equivalent SCPI command

Syntax

**:SENSe{[1]-160}:CORRection:COLLect:ECAL:ERESponse <numeric
1>,<numeric 2>**

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ERES 1,2"  
20 OUTPUT 717;""OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.INFOrmation**Object type**

Property (**Read Only**)

Syntax

Param = SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.INFOrmation?

Description

This command gets information of the connected ECAL modules to the E5071C.

Variable

The command return information in a string variable <param> in the following syntax:

[For 2-port ECal]

- ModelNumber
- SerialNumber
- PortAConnector
- PortBConnector
- MinFreq
- MaxFreq
- NumberOfPoints
- Calibrated
- ID
- PortAExtension
- PortBExtension
- Analyzer
- Operator
- Location

[For 4-port ECal]

- ModelNumber
- SerialNumber
- PortAConnector
- PortBConnector
- PortCConnector
- PortDConnector
- MinFreq
- MaxFreq

E5071C

- NumberOfPoints
- Calibrated
- ID
- PortAExtension
- PortBExtension
- PortCExtension
- PortDExtension
- Analyzer
- Operator
- Location

Examples

```
Dim Cal_Info As String  
Cal_Info = SCPI.SENS1.CORRection.COLLection.ECAL.INFormation
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLLection.SAVE  
SCPI.SENSE(Ch).CORRection.TYPE(Tr)
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection:ECAL:INFormation?
```

Query response

```
{Model number,Serial number,Connector type, Calibration date, Min and  
max frequency}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:INF?"  
20 ENTER 717;A$
```


SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ISOLation.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ISOLation.STATe = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ISOLation.STATe**Description**

From revision A.9.00, this command ignores ON/OFF of the isolation measurement when executing Ecal for the selected channel (*Ch*), because the isolation of the ENA is better than that of the ECal. This command takes no action and only exists to maintain backward compatibility.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the isolation measurement when executing ECal
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the isolation measurement. • False or OFF: Turns OFF the isolation measurement.
Preset value	False or OFF

Examples

```
Dim Ecallso As Boolean
SCPI.SENSE1.CORRection.COLlect.ECAL.ISOLation.STATe = False
Ecallso = SCPI.SENSE1.CORRection.COLlect.ECAL.ISOLation.STATe
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT1
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT2
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT3
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT4
```

Equivalent key

E5071C

No equivalent key is available on front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:ISOLation[:STATe]  
{ON|OFF|1|0}  
:SENSe{[1]-160}:CORRection:COLLect:ECAL:ISOLation[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ISOL.STATE OFF"  
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ISOL.STATE?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ORIentation.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ORIentation.STATe = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ORIentation.STATe

Description

This command turns ON/OFF the ECal auto detect function.

Variable

Parameter	<i>Status</i>
Description	ON/OFF the ECal auto detect funcion.
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the auto detect function. • False or OFF: Turns OFF the auto detect function.
Preset value	True or ON

For information on the variable (*Ch*), see *Ch*.

Examples

Dim EcalOri As Boolean

SCPI.SENSE(1).CORRection.COLlect.ECAL.ORIentation.STATe = True

EcalOri = SCPI.SENSE(1).CORRection.COLlect.ECAL.ORIentation.STATe

Related objects

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.PATH(*Cpt*)SCPI.CALCulate(*Ch*).FSIMulator.SENDeD.DEEMbed.PORT(*Pt*).TYPE

Equivalent key

Cal > ECal > Orientation

Equivalent SCPI command

Syntax

E5071C

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ORientation[:STATe]
{ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ORientation[:STATe]?

Query response

{0|1}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ORI ON"  
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ORI?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.ECAL.PATH(Cpt)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.COLlect.ECAL.PATH(Cpt) = *Value**Value* = SCPI.SENSE(Ch).CORRection.COLlect.ECAL.PATH(Cpt)

Description

This command set/get the ECal module n port number which is connected to a specified port.

Variable

Parameter	<i>Value</i>
Description	Port of ECal module.
Data type	Long integer type (Long)
Range	<p>One of the following is set/get:</p> <ul style="list-style-type: none"> • 0: Nothing is connected. • 1: Port A is connected. • 2: Port B is connected. • 3: Port C is connected. • 4: Port D is connected.

For information on the variable (*Cpt*), see *Cpt*.

Examples

```
Dim ECalPort As Long
SCPI.SENSE.CORRection.COLlect.ECAL.PATH(1) = 3
ECalPort = SCPI.SENSE.CORRection.COLlect.ECAL.PATH(1)
```

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.ECAL.ORIentation.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLlect:ECAL:PATH <Cpt>,<Value>

:SENSe{[1]-160}:CORRection:COLlect:ECAL:PATH? <Cpt>

E5071C

Example of use

10 OUTPUT 717;".SENS1:CORR:COLL:ECAL:PATH 1,2"

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT1**Object type**Property (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT1 = *Eport***Description**

This command executes 1-port calibration of the specified port of selected channel (*Ch*) using the ECal (Electronic Calibration) module.

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	<i>Eport</i>
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

SCPI.SENSE(1).CORRection.COLlect.ECAL.SOLT1 = 1

Equivalent key**Cal > ECal > 1-Port Cal > Port 1|Port 2|Port 3|Port 4****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLlect:ECAL:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT1 1"
20 OUTPUT 717;""*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT2**Object type**Property (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT2 = *Eports***Description**

Executes full 2-port calibration between the specified 2 ports of channels 1 to 36 (*Ch*) using the ECal (Electronic Calibration) module.

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	<i>Eports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>EPorts(0)</i> <i>EPorts(1)</i> <p>Specifies the port numbers for 2-port ECal. The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. the order of the 2 port numbers to be specified is arbitrary.

Examples

SCPI.SENSE(1).CORRection.COLlect.ECAL.SOLT2 = Array(1,2)

Dim EcalPort(1) As Variant

EcalPort(0) = 1

EcalPort(1) = 2

SCPI.SENSE(1).CORRection.COLlect.ECAL.SOLT2 = EcalPort

Equivalent key

Cal > ECal > 2-Port Cal > Port 1-2|Port 1-3|Port 1-4|Port 2-3|Port 2-4|Port 3-4
Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:SOLT2 <numeric  
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT2 1,2"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT3

Object type

Property (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT3 = *Eports*

Description

This command executes full 3-port calibration between the specified 3 ports of channels 1 to 36 (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the 4-port ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	<i>Eports</i>
Description	<p>Indicates 3-element array data (port number).</p> <ul style="list-style-type: none"> • <i>EPorts(0)</i> <i>EPorts(1)</i> <i>EPorts(2)</i> <p>Specifies the port numbers for 3-port ECal. The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to more than 2 port numbers, an error occurs when executed. the order of the 3 port numbers to be specified is arbitrary.

Examples

SCPI.SENSE(1).CORRection.COLlect.ECAL.SOLT3 = Array(1,2,3)

Equivalent key

Cal > ECal > 3-Port Cal > Port 1-2-3|Port 1-2-4|Port 1-3-4|Port 2-3-4

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:SOLT3 <numeric  
1>,<numeric 2>,<numeric 3>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT3 1,2,3"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A  
Dim EcalPort(2) As Variant  
EcalPort(0) = 1  
EcalPort(1) = 2  
EcalPort(2) = 3  
SCPI.SENSE(1).CORRection.COLLect.ECAL.SOLT3 = EcalPort
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT4

Object type

Property (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.SOLT4 = *Eports*

Description

This command executes full 4-port calibration for selected channel (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the 4-port ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	<i>Eports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>EPorts(0)</i> <i>EPorts(1)</i> <i>EPorts(2)</i> <i>EPorts(3)</i> <p>Specifies the port numbers for 4-port ECal. The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to more than 2 port numbers, an error occurs when executed. the order of the 4 port numbers to be specified is arbitrary.

Examples

SCPI.SENSE(1).CORRection.COLlect.ECAL.SOLT4 = Array(1,2,3,4)

Dim EcalPort(3) As Variant

EcalPort(0) = 1

EcalPort(1) = 2

```
EcalPort(2) = 3
EcalPort(3) = 4
SCPI.SENSE(1).CORRection.COLLection.ECAL.SOLT4 = EcalPort
```

Equivalent key

Cal > ECal > 4-Port Cal

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection:ECAL:SOLT4 1,2,3,4
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT4 1,2,3,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.THRU

Object type

Property (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.THRU = *Eports*

Description

This command executes response calibration (thru) between the specified 2 ports of selected channel (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	<i>Eports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the response port number. <i>Ports(1)</i>: Specifies the stimulus port number. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.</p>

Examples

SCPI.SENSE(1).CORRection.COLlect.ECAL.THRU = Array(1,2)

Dim EcalPort(1) As Variant

EcalPort(0) = 1

```
EcalPort(1) = 2
SCPI.SENSE(1).CORRection.COLLection.ECAL.THRU = EcalPort
```

Equivalent key

Cal > ECal > Thru Cal > 2-1 (S21)|3-1 (S31)|4-1 (S41)| 1-2 (S12)|3-2 (S32)| 4-2 (S42)| 1-3 (S13)|2-3 (S23)|4-3 (S43)| 1-4 (S14)|2-4 (S24)| 3-4 (S34)

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection:ECAL:THR U <numeric  
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT2 1,2"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ECAL.UCHar

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.ECAL.UCHar = *Param*

Description

This command sets the ECal characteristic used when executing the user-defined ECal, for the selected channel (*Ch*).

- The user-defined ECal is a type of ECal that is executed using the characteristic that has been acquired by the user and stored in the memory for ECal.

When the ECal module is not connected or the characteristic is not stored at the specified location number, executing this object will cause a runtime error.

- This command is available with the Firmware revision 3.50 or higher.

Variable

Parameter	<i>Param</i>
Description	Characteristic used when executing ECal (user characterization)
Data type	Character string type (String)
Range	<p>Select from either of the following:</p> <ul style="list-style-type: none"> • "CHAR0": Uses the factory-default characteristic. (Normal ECal) • "CHAR1": Uses the characteristic stored at location number 1 in the ECal's flash memory. • "CHAR2": Uses the characteristic stored at location number 2 in the ECal's flash memory. • "CHAR3": Uses the characteristic stored at location number 3 in the ECal's flash memory. • "CHAR4": Uses the characteristic stored at location number 4 in the ECal's flash memory. • "CHAR5": Uses the characteristic stored at

	location number 5 in the ECal's flash memory.
Preset value	"CHAR0"

Examples

```
Dim UserChar As String
SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar = "CHAR2"
UserChar = SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar
```

Equivalent key

Cal > ECal > Characterization > Factory|User1|User2|User3|User4|User5

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:UCHar {CHAR0| CHAR1
|CHAR2|CHAR3|CHAR4|CHAR5}
:SENSe{[1]-160}:CORRection:COLLect:ECAL:UCHar?
```

Query response

```
{CHAR0|CHAR1|CHAR2|CHAR3|CHAR4|CHAR5}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UCH CHAR1"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UCH?"
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLLection.ECAL.UTHRU.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.ECAL.UTHRU.STATe = *Status*
Status= SCPI.SENSE(*Ch*).CORRection.COLLection.ECAL.UTHRU.STATe

Description

This command sets Unknown Thru On/Off for Ecal.

Variable

Parameter	<i>Status</i>
Description	Sets/gets Unknown Thru ON/Off for Ecal
Data type	Boolean
Range	Select from either of the following: <ul style="list-style-type: none">• True or ON: Turns ON the Unknown Thru function.• False or OFF: Turns OFF the Unknown Thru function.

Examples

```
Dim EcalPort As Boolean
EcalPort = 0
SCPI.SENSE(1).CORRection.COLLection.ECAL.UTHRU.STATe = 0
EcalPort=SCPI.SENSE(1).CORRection.COLLection.ECAL.UTHRU.STATe
```

Equivalent key

Cal > ECal > Unknown Thru

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLection:ECAL:UTHRU:STATe
{ON|OFF|1|0}

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UTHRU:STAT 1"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UTHRU:STAT?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.COLlect.METHod.ADAPter.REMoval**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(Ch).CORRection.COLlect.METHod.ADAPter.REMoval = *Port***Description**

This command sets the port in which adaptor needs to be added/removed.

Variable

Parameter	<i>Port</i>
Description	Port in which Adaptor needs to be added/removed
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1

Examples

SCPI.SENSE(Ch).CORRection.COLlect.METHod.ADAPter.REMoval = 1

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.ADAPter(Pt).LENGth

Equivalent key**Cal > Calibrate > Adapter Removal > Select Port****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLlect:METHod:ADAPter:REMoval {1-4}

Example of use

Call viVPrintf(vi, ":SENS1:CORR:COLL:METH:ADAPter:REMoval 1"+vbcr, 0)

SCPI.SENSE(Ch).CORRection.COLlect.METHod.ERESponse**Object type**Method (**Write-only**)**Syntax**

SCPI.SENSE(Ch).CORRection.COLlect.METHod.ERESponse = Ports

Description

This command sets the calibration type to the enhanced response calibration between the two specified ports, for the selected channel.

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies the response port. <i>Ports(1)</i>: Specifies the stimulus port. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESport(1) As Variant
ERESport(0) = 1
ERESport(1) = 2
SCPI.SENSE(1).CORRection.COLlect.METHod.ERESponse = ERESport
```

Related objects

SCPI.SENSE(Ch).CORRection.COLlect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Enhanced Response** > **Select Ports** > **2-1 (S21 S11)|3-1 (S31 S11)|...|3-4 (S34 S44)**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:METHod:ERESponse <numeric  
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:ERES 1,2"
```

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.RESPOnse.OPEN

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.RESPOnse.OPEN = *Port*

Description

This command sets the calibration type to the response calibration (open) of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSE(1).CORRection.COLLection.METHod.RESPOnse.OPEN = 1

Related objects

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.TYPE

Equivalent key

Cal > Calibrate > Response (Open) > Select Port

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection:METHod[:RESPOnse]:OPEN
<numeric>
```

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:OPEN 1"

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.RESPOnse.SHORt**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.RESPOnse.SHORt = *Port***Description**

This command sets the calibration type to the response calibration (short) of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSE(1).CORRection.COLLection.METHod.RESPOnse.SHORt = 1

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.METHod.TYPE**Equivalent key****Cal > Calibrate > Response (Short) > Select Port****Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:CORRection:COLLection:METHod[:RESPOnse]:SHORt
<numeric>
```

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:SHOR 1"

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.RESPOnse.THRU**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.RESPOnse.THRU = *Ports***Description**

This command sets the calibration type to the response calibration (thru) between the specified 2 ports, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Ch and Port.

Examples

```
SCPI.SENSE(1).CORRection.COLLection.METHod.RESPOnse.THRU = Array(2,1)
Dim ThruPort(1) As Variant
ThruPort(0) = 2
ThruPort(1) = 1
SCPI.SENSE(1).CORRection.COLLection.METHod.RESPOnse.THRU = ThruPort
```

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.METHod.TYPE**Equivalent key****Cal > Calibrate > Response (Thru) > Select Ports****Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:CORRection:COLLection:METHod[:RESPOnse]:THRU
<numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:THRU 1,2"
```


SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.SOLT1**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.SOLT1 = *Port***Description**

This command sets the calibration type to the 1-port calibration of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSE(1).CORRection.COLLection.METHod.SOLT1 = 1

Related objectsSCPI.SENSE(*Ch*).CORRection.COLLection.METHod.TYPE**Equivalent key****Cal > Calibrate > 1-Port Cal > Select Port****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLLection:METHod:SOLT1 <numeric>

Example of use

10 OUTPUT 717";:SENS1:CORR:COLL:METH:SOLT1 1"

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT2**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT2 = *Ports***Description**

This command sets the calibration type to the full 2-port calibration between the specified 2 ports, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for full 2-port calibration. <i>Ports(1)</i>: Specifies the other port for full 2-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.</p>

Examples

```
SCPI.SENSE(1).CORRection.COLlect.METHod.SOLT2 = Array(1,2)
Dim CalPort(1) As Variant
CalPort(0) = 1
CalPort(1) = 2
SCPI.SENSE(1).CORRection.COLlect.METHod.SOLT2 = CalPort
```

Related objectsSCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TYPE**Equivalent key**

Cal > Calibrate > 2-Port Cal > Select Ports**Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:CORRection:COLLect:METHod:SOLT2 <numeric  
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT2 1,2"
```

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT3

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT3 = *Ports*

Description

This command sets the calibration type to the full 3-port calibration between the specified 3 ports, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 3-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for full 3-port calibration. <i>Ports(1)</i>: Specifies a port for full 3-port calibration. <i>Ports(2)</i>: Specifies a port for full 3-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 3 port numbers to be specified is arbitrary.</p>

Examples

```
SCPI.SENSE(1).CORRection.COLlect.METHod.SOLT3 = Array(1,2,3)
Dim CalPort(2) As Variant
CalPort(0) = 1
CalPort(1) = 2
CalPort(2) = 3
SCPI.SENSE(1).CORRection.COLlect.METHod.SOLT3 = CalPort
```

Related objects

SCPI.SENSE{Ch}.CORRection.COLLection.METHod.TYPE

Equivalent key

Cal > Calibrate > 3-Port Cal > Select Ports

Equivalent SCPI command**Syntax**

:SENSe{[1]-160}:CORRection:COLLection:METHod:SOLT3 <numeric 1>,
<numeric 2>,<numeric 3>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT3 1,2,3"

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.SOLT4**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.SOLT4 = *Ports***Description**

This command sets the calibration type to the full 4-port calibration, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for full 4-port calibration. <i>Ports(1)</i>: Specifies a port for full 4-port calibration. <i>Ports(2)</i>: Specifies a port for full 4-port calibration. <i>Ports(3)</i>: Specifies a port for full 4-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 4 port numbers to be specified is arbitrary.</p>

Examples

SCPI.SENSE(1).CORRection.COLLection.METHod.SOLT4 = Array(1,2,3,4)

Dim CalPort(3) As Variant

CalPort(0) = 1

CalPort(1) = 2

CalPort(2) = 3

CalPort(3) = 4

SCPI.SENSE(1).CORRection.COLLection.METHod.SOLT4 = CalPort

Related objects

SCPI.SENSE(Ch).CORRection.COLLection.METHod.TYPE

Equivalent key

Cal > Calibrate > 4-Port Cal

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLection:METHod:SOLT4 1,2,3,4

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT3 1,2,3,4"

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL2

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL2 = *Ports*

Description

This command sets the calibration type to the TRL calibration between the 2 specified ports, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 2-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for TRL 2-port calibration. <i>Ports(1)</i>: Specifies a port for TRL 2-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.</p>

Examples

```
SCPI.SENSE(1).CORRection.COLlect.METHod.TRL2 = Array(1,2)
Dim TrlCalPort(1) As Variant
TrlCalPort(0) = 1
TrlCalPort(1) = 2
SCPI.SENSE(1).CORRection.COLlect.METHod.TRL2 = TrlCalPort
```

Related objects

SCPI.SENSE(*Ch*).CORRection.COEfficient.DATASCPI.SENSE(*Ch*).CORRection.COEfficient.SAVE

SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL3
 SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL4
 SCPI.SENSE(Ch).CORRection.COLLection.SAVE
 SCPI.SENSE(Ch).CORRection.TYPE(Tr)

Equivalent key

Cal > **Calibrate** > **2-Port TRL Cal** > **Select Ports** > **1-2|1-3|1-4|2-3|2-4|3-4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection
:METHod:TRL2 <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TRL2 1,2"
```

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL3

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL3 = *Ports*

Description

This command sets the calibration type to the TRL calibration between the 3 specified ports, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 3-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for TRL 3-port calibration. <i>Ports(1)</i>: Specifies a port for TRL 3-port calibration. <i>Ports(2)</i>: Specifies a port for TRL 3-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 3 port numbers to be specified is arbitrary.</p>

Examples

```
SCPI.SENSE(1).CORRection.COLlect.METHod.TRL3 = Array(1,2,3)
Dim TrlCalPort(2) As Variant
TrlCalPort(0) = 1
TrlCalPort(1) = 2
TrlCalPort(2) = 3
SCPI.SENSE(1).CORRection.COLlect.METHod.TRL3 = TrlCalPort
```

Related objects

SCPI.SENSE(Ch).CORRection.COEFficient.DATA
 SCPI.SENSE(Ch).CORRection.COEFficient.SAVE
 SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL2
 SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL4
 SCPI.SENSE(Ch).CORRection.COLLection.SAVE
 SCPI.SENSE(Ch).CORRection.TYPE(Tr)

Equivalent key

Cal > **Calibrate** > **3-Port TRL Cal** > **Select Ports** > **1-2-3|1-2-4|1-3-4|2-3-4**

Equivalent SCPI command**Syntax**

```

:SENSe{[1]-160}:CORRection:COLLection
:METHod:TRL3 <numeric 1>,<numeric 2>,<numeric 3>

```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TRL3 1,2,3"
```

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.TRL4

Object type

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CCORRection.COLLection.METHod.TRL4 = *Ports*

Description

This command sets the calibration type to the TRL calibration between the 4 specified ports, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Indicates 4-element array data (port number).</p> <ul style="list-style-type: none"> <i>Ports(0)</i>: Specifies a port for TRL 4-port calibration. <i>Ports(1)</i>: Specifies a port for TRL 4-port calibration. <i>Ports(2)</i>: Specifies a port for TRL 4-port calibration. <i>Ports(3)</i>: Specifies a port for TRL 4-port calibration. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 4 port numbers to be specified is arbitrary.</p>

Examples

SCPI.SENSE(1).CORRection.COLLection.METHod.TRL4 = Array(1,2,3,4)

Dim TrlCalPort(3) As Variant

TrlCalPort(0) = 1

```
TrlCalPort(1) = 2
TrlCalPort(2) = 3
TrlCalPort(3) = 4
SCPI.SENSE(1).CORRection.COLLection.METHod.TRL4 = TrlCalPort
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COEFficient.DATA
SCPI.SENSE(Ch).CORRection.COEFficient.SAVE
SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL2
SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL3
SCPI.SENSE(Ch).CORRection.COLLection.SAVE
SCPI.SENSE(Ch).CORRection.TYPE(Tr)
```

Equivalent key

Cal > **Calibrate** > **4-Port TRL Cal**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLection
:METHod:TRL4 <numeric 1>,<numeric 2>,<numeric 3>,<numeric 4>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TRL4 1,2,3,4"
```

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TYPE

Object type

Property (**Read Only**)

Syntax

Param = SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TYPE

Description

This command reads the selected calibration type of selected channel (*Ch*).

- This object is used to check the selected calibration type for calculating the calibration coefficients. To check the applied calibration type (error correction on), use the SCPI.SENSE(*Ch*).CORRection.TYPE(*Tr*) object.

Variable

Parameter	<i>Param</i>	
Description	Calibration type	
Range	AREM	Adaptor Removal
	ERES	Enhanced response calibration
	NONE	None
	RESPO	Response calibration (open)
	RESPS	Response calibration (short)
	RESPT	Response calibration (thru)
	SOLT1	1-port calibration.
	SOLT2	Full 2-port calibration.
	SOLT3	Full 3-port calibration.
	SOLT4	Full 4-port calibration.

	TRL2	TRL 2-port calibration.
	TRL3	TRL 3-port calibration.
	TRL4	TRL 4-port calibration.
Preset Value	NONE	
Data type	Character string type (String)	

Examples

```
Dim CalType As String
CalType = SCPI.SENSE(1).CORRection.COLLection.METHod.TYPE
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLLection.SAVE
SCPI.SENSE(Ch).CORRection.TYPE(Tr)
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:COLLection:METHod:TYPE?
```

Query response

```
{AREM|ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|SOLT3|SOLT4|TRL2|TRL3|TRL4}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TYPE?"
20 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.COLLection.PARTial.SAVE**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLLection.PARTial.SAVE**Description**

This command, used for partial overwrite, recalculates the calibration coefficients depending on the selected calibration type from the measured calibration data.

- Calculating the calibration coefficients clears all calibration data regardless of whether they are used for the calculation and also clears the calibration type selections.

If partial overwrite is executed before selecting the calibration type, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

SCPI.SENSE(1).CORRection.COLLection.PARTial.SAVE

Equivalent key**Cal > Calibrate > n-Port Cal > Overwrite****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:COLLection:PARTial:SAVE

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:PART:SAVE"

SCPI.SENSE(*Ch*).CORRection.COLlect.SAVE**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.SAVE**Description**

This command calculates the calibration coefficients depending on the calibration type selection, from the measured calibration data.

- Calculating the calibration coefficients clears all the measured calibration data whether or not used for the calculation and also clears the calibration type selection.

If this command is executed before all necessary calibration data for calculating the calibration coefficients is measured, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Dmy As Long
SCPI.SENSE(1).CORRection.COLlect.METHod.RESPOnse.THRU = Array(2,1)
SCPI.SENSE(1).CORRection.COLlect.ACQuire.THRU = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSE(1).CORRection.COLlect.SAVE
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.METHod.RESPOnse.OPEN
SCPI.SENSE(Ch).CORRection.COLlect.METHod.RESPOnse.SHORT
SCPI.SENSE(Ch).CORRection.COLlect.METHod.RESPOnse.THRU
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT1
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT2
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT3
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT4
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL2
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL3
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL4
```

Equivalent key

Cal > **Calibrate** > **Response|n-Port Cal** > **Done**

Equivalent SCPI command**Syntax**

E5071C

:SENSe{[1]-160}:CORRection:COLLect:SAVE

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:SAVE"

SCPI.SENSE(*Ch*).CORRection.COLlect.SIMPLified.SAVE**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.COLlect.SIMPLified.SAVE**Description**

This command calculates the calibration coefficients for the simple full 3 port calibration or the simple full 4 port calibration from the measured calibration data, when the full 3/4 port calibration is selected as the calibration type.

- If the response calibration or the 1/2 port calibration is selected as the calibration type, this command behaves similarly to SCPI.SENSE(*Ch*).CORRection.COLlect.SAVE.

After the calibration coefficients are calculated, the measured data and the calibration type setting are cleared.

If this command is executed before all the necessary calibration data for calculating the calibration coefficients for the simple full 3 port calibration or the simple full 4 port calibration is measured, a runtime error occurs.

This function is available with the Firmware revision 3.50 or greater.

Variable

For information on the variable (*Ch*), refer to Ch.

Examples

```
SCPI.SENSE(1).CORRection.COLlect.METHod.SOLT3 = Array(1,2,3)
SCPI.SENSE(1).CORRection.COLlect.SIMPLified.SAVE
```

Related objects

```
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT3
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT4
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL3
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL4
SCPI.SENSE(Ch).CORRection.COLlect.SAVE
```

Equivalent key

Cal > Calibrate > n-Port Cal|n-Port TRL > Done

Equivalent SCPI command**Syntax**

E5071C

:SENSe{[1]-160}:CORRection:COLLect:SIMPlied:SAVE

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:SIMP:SAVE"

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.CONFig**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.CONFig = *Param**Param* = SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.CONFig**Description**

This command sets/gets the frequency point to calculate the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	The frequency point to calculate the auto port extension
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "CSPN": Uses the frequency of the current sweep range. • "AMKR": Use the frequency of the active marker. This is applied to Loss 1 and Loss 2 is ignored. • "USPN": This is executed with the arbitrary specified start frequency and stop frequency.
Preset value	"CSPN"

Examples

Dim Conf As String

SCPI.SENSE(1).CORRection.EXTension.AUTO.CONFig = "AMKR"

Conf = SCPI.SENSE(1).CORRection.EXTension.AUTO.CONFig

Related objectsSCPI.SENSE(*Ch*).CORRection.EXTension.STATeSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.DCOffsetSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSSSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.MEASure

E5071C

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.PORT(Pt)

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.RESet

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.STARt

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.STOP

Equivalent key

Cal > Port Extensions > Auto Port Extension > Method > Current Span|Active Marker|User Span

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:CONFig
{CSPN|AMKR|USPN}

:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:CONFig?

Query response

{CSPN|AMKR|USPN}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:CONF CSPN"  
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:CONF?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.DCOffset

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.DCOffset = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.DCOffset

Description

This command enables/disables or gets the usage of DC Offset value for the results of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF the usage of DC Offset value for the results of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Uses the DC Offset value for the results. • False or OFF: Does not use the DC Offset value for the results.
Preset value	False or OFF

Examples

Dim Dcof As Boolean

SCPI.SENSE(1).CORRection.EXTeNsion.AUTO.DCOffset = True

Dcof = SCPI.SENSE(1).CORRection.EXTeNsion.AUTO.DCOffset

Related objects

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElectSCPI.CALCulate(*Ch*).SElected.CONVersion.FUNcTionSCPI.SENSE(*Ch*).CORRection.EXTeNsion.STATeSCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.CONFigSCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.LOSSSCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.MEASureSCPI.SENSE(*Ch*).CORRection.EXTeNsion.AUTO.PORT(*Pt*)

E5071C

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.RESet

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.STARt

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Adjust Mismatch**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:DCOOffset {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:DCOOffset?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:DCOF ON"

20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:DCOF?"

30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSS**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSS = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSS**Description**

This command turns ON/OFF or gets the status of the loss compensation for the results of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF the loss compensation for the results of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the loss compensation • False or OFF: Turns off the loss compensation
Preset value	False or OFF

Examples

```
Dim AutoLoss As Boolean
SCPI.SENSE(1).CORRection.EXTension.AUTO.LOSS = True
AutoLoss = SCPI.SENSE(1).CORRection.EXTension.AUTO.LOSS
```

Related objects

```
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.CONFig
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.DCOffset
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.MEASure
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.PORT(Pt)
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.RESet
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.STARt
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.STOP
```

Equivalent key

Cal > Port Extensions > Auto Port Extension > Include Loss

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:AUTO:LOSS {ON|OFF|1|0}  
:SENSe{[1]-160}:CORRection:EXTension:AUTO:LOSS?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:LOSS ON"  
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:LOSS?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.MEASure

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.MEASure = *Param**Param* = SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.MEASure

Description

This command measures the calibration data of the OPEN standard or SHORT standard of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Standard type of the auto port extension
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "OPEN": Measures the calibration data of the OPEN standard • "SHORT": Measures the calibration data of the SHORT standard
Preset value	"SHORT"

Examples

Dim AutoMeas As String

SCPI.SENSE(1).CORRection.EXTension.AUTO.MEASure = "OPEN"

AutoLoss = SCPI.SENSE(1).CORRection.EXTension.AUTO.LOSS

Related objects

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.CONFigSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.DCOffsetSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSSSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.RESetSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.STARTSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.STOP

E5071C

Equivalent key

Cal > Port Extensions > Auto Port Extension > Measure OPEN|Measure Short- All|Port 1|Port 2|Port 3|Port 4

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:AUTO:MEASure {OPEN|SHORT}  
:SENSe{[1]-160}:CORRection:EXTension:AUTO:MEASure?
```

Query response

```
{OPEN|SHOR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:MEAS OPEN"  
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:MEAS?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*) = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)

Description

This command turns ON/OFF or gets the status of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	On/off of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the auto port extension • False or OFF: Turns OFF the auto port extension
Preset value	True or ON

For information on the variable (*Ch*) or *Pt*, refer to *Ch* or *Pt*.

Examples

Dim APort As Boolean

SCPI.SENSE(1).CORRection.EXTension.AUTO.PORT(1) = True

APort = SCPI.SENSE(1).CORRection.EXTension.AUTO.PORT(1)

Related objects

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.CONFigSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.DCOffsetSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSSSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.MEASureSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.RESetSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.STARtSCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.STOP

E5071C

Equivalent key

Cal > Port Extensions > Auto Port Extension > Select Ports > Port 1|Port 2|Port 3|Port 4

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:AUTO:PORT{[1]|2|3|4}  
{ON|OFF|1|0}  
:SENSe{[1]-160}:CORRection:EXTension:AUTO:PORT{[1]|2|3|4}?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:PORT1 ON"  
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:PORT1?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.RESet**Object type**Method (**Write-only**)**Syntax**SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.RESet**Description**

This command deletes the finished measurement data (OPEN and SHORT), for the selected channel (*Ch*).

Examples

SCPI.SENSE(1).CORRection.EXTension.AUTO.RESet

Related objects

SCPI.SENSE(*Ch*).CORRection.EXTension.STATe
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.CONFig
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.DCOffset
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.LOSS
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.MEASure
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.STARt
 SCPI.SENSE(*Ch*).CORRection.EXTension.AUTO.STOP

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SENSe{[1]-160}:CORRection:EXTension:AUTO:RESet

Example of use

10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:RESet"

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.START**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).CORRection.EXTension.AUTO.START = *Value**Value* = SCPI.SENSE(Ch).CORRection.EXTension.AUTO.START**Description**

This command get/set the start frequency within the frequency range of the user specified auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Start frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	3E5
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AStart As Double
SCPI.SENSE(1).CORRection.EXTension.AUTO.START = 1E9
AStart = SCPI.SENSE(1).CORRection.EXTension.AUTO.START
```

Related objects

SCPI.SENSE(Ch).CORRection.EXTension.STATe
 SCPI.SENSE(Ch).CORRection.EXTension.AUTO.CONFig
 SCPI.SENSE(Ch).CORRection.EXTension.AUTO.DCOffset
 SCPI.SENSE(Ch).CORRection.EXTension.AUTO.LOSS
 SCPI.SENSE(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.PORT(Pt)

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.RESet

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.STOP

Equivalent key

Cal > Port Extensions > Auto Port Extension > Method > User Span Start

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:STARt <numeric>

:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:STARt?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STAR 1.2E9"

20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STAR?"

30 ENTER 717;A

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.STOP = *Value**Value* = SCPI.SENSE(Ch).CORRection.EXTension.AUTO.STOP

Description

This command sets/gets the stop frequency within the frequency range of the user specified auto port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Stop frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	Maximum frequency in the range
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim AStop As Double

SCPI.SENSE(1).CORRection.EXTension.AUTO.STOP = 250E8

AStop = SCPI.SENSE(1).CORRection.EXTension.STOP.START

Related objects

SCPI.SENSE(Ch).CORRection.EXTension.STATe

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.DCOffset

SCPI.SENSE(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.MEASure
 SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.PORT(Pt)
 SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.RESet
 SCPI.SENSE(Ch).CORRection.EXTeNsion.AUTO.STARt

Equivalent key

Cal > Port Extensions > Auto Port Extension > Method > User Span Stop

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:STOP <numeric>
:SENSe{[1]-160}:CORRection:EXTeNsion:AUTO:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STOP 1.5E9"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STOP?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*) = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)

Description

This command sets/gets the frequency used for calculation of the loss value of the frequency 1 and 2 (*Fq*) of the selected port (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Fq</i>
Description	Frequency number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Value</i>
Description	Frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	0

Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PortFreq As Double
SCPI.SENSE(1).CORRection.EXTension.PORT(1).FREQuency(1) = 500E6
PortFreq = SCPI.SENSE(1).CORRection.EXTension.PORT(1).FREQuency(1)
```

Related objects

```
SCPI.SENSE(Ch).CORRection.EXTension.STATe
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).INCLude(II).STATe
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)
```

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3**
| **Extension Port4** > **Loss** > **Freq1**|**Freq2**

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:FREQuency{[1]|2}
<numeric>
:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:FREQuency{[1]|2}?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:FREQ1 10E6"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:FREQ1?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe =
Status

Status =

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe

Description

This command turns ON/OFF the set of loss value and frequency value of include 1 and 2 (*Il*) of the port 1 to 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Il</i>
Description	Include number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Status</i>
Description	ON/OFF the set of loss value and frequency value.
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the loss value and frequency value. • False or OFF: Turns OFF the loss value and

	frequency value.
Preset value	False or OFF

Examples

```
Dim PortIncl As Double
SCPI.SENSE(1).CORRection.EXTension.PORT(1).INCLude(1).STATe = 500E6
PortIncl = SCPI.SENSE(1).CORRection.EXTension.PORT(1).INCLude(1).STATe
```

Related objects

```
SCPI.SENSE(Ch).CORRection.EXTension.STATe
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME
```

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3**
| **Extension Port4** > **Loss** > **Loss1**|**Loss2**

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:INCLude{[1]|2}[:STATe]
{ON|OFF|1|0}

:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:INCLude{[1]|2}[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:INCL1 ON"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:INCL1?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC**Description**

This command sets/gets the DC loss value of the port 1 to 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	The loss value of DC.
Data type	Double precision floating point type (Double)
Range	-90 to 90
Preset value	0
Unit	dBm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim PLdc As Double

SCPI.SENSE(1).CORRection.EXTension.PORT(1).LDC = 45

PLdc = SCPI.SENSE(1).CORRection.EXTension.PORT(1).LDC

Related objectsSCPI.SENSE(*Ch*).CORRection.EXTension.STATeSCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATeSCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Loss*)SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3**
 | **Extension Port4** > **Loss** > **Loss at DC**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LDC
<numeric>
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LDC?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LDC 1.2"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LDC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(Loss)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(Loss) = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(Loss)

Description

This command sets/gets the loss value of the loss 1 to 4 of the port 1 to 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Loss</i>
Description	Loss number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	<i>Value</i>
Description	The loss value
Data type	Double precision floating point type (Double)
Range	-90 to 90
Preset value	0

Unit	dBm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PLoss As Double
SCPI.SENSE(1).CORRection.EXTension.PORT(1).LOSS(1) = -45
PLoss = SCPI.SENSE(1).CORRection.EXTension.PORT(1).LOSS(1)
```

Related objects

```
SCPI.SENSE(Ch).CORRection.EXTension.STATe
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).INCLude(II).STATe
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME
```

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3**
| **Extension Port4** > **Loss** > **Loss1**|**Loss2**

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LOSS{[1]|2}
<numeric>
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LOSS{[1]|2}?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LOSS1 0.8"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LOSS1?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Description

This command sets/gets or gets the status of the delay time for the port extension of ports 1 and 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Delay time
Data type	Double precision floating point type (Double)
Range	-10 to 10
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PortExt As Double
SCPI.SENSE(1).CORRection.EXTension.PORT(1).TIME = 1E-3
PortExt = SCPI.SENSE(1).CORRection.EXTension.PORT(1).TIME
```

Related objects

SCPI.SENSE(*Ch*).CORRection.EXTension.STATe
 SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)
 SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe
 SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC
 SCPI.SENSE(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Loss*)

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3**
 | **Extension Port4** > **Coax. Extension**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:TIME
<numeric>
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:TIME?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:TIME 1e-3"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:TIME?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff =
Cfreq

Cfreq =

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

Description

This command sets/gets cutoff frequency of the waveguide of the selected port (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Cfreq</i>
Description	Waveguide cutoff frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Unit	Hz
Preset value	0 Hz

Examples

Dim CFreq As Double

SCPI.SENSE(2).CORRection.EXTension.PORT(1).WAVeguide.CUToff = 1E5

CFreq = SCPI.SENSE(1).CORRection.EXTension.PORT(3).WAVeguide.CUToff

Related objects

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME

Equivalent key

Cal > Port Extensions > Extension Port 1|Extension Port 2|Extension Port 3|Extension Port 4 > Cutoff Frequency

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:CUToff
<numeric>

:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:CUToff?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:CUT 1e3"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:CUT?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME**Object type**Property (**Read-Write**)**Syntax**

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME =
Value

Value =

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME

Description

This command sets/gets the port extension of the selected port (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Port extension
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Preset value	0 second

Examples

Dim WgTime As Double

SCPI.SENSE(2).CORRection.EXTension.PORT(1).WAVeguide.TIME = 5

WgTime = SCPI.SENSE(1).CORRection.EXTension.PORT(3).WAVeguide.TIME

Related objects

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

Equivalent key

Cal > Port Extensions > Extension Port 1|Extension Port 2|Extension Port 3|Extension Port 4 > Waveguide

Equivalent SCPI command**Syntax**

:SENSe{[1]-
 160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:TIME
 <numeric>


```
:SENSe{[1]-  
160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:TIME?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:TIME -4"  
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:TIME?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.EXTension.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.EXTension.STATe = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.EXTension.STATe**Description**

This command turns ON/OFF or returns the status of the port extension, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the port extension correction
Data type	Boolean type (Boolean)
Range	Select from either of the following. <ul style="list-style-type: none"> • True or ON: Turns ON the port extension. • False or OFF: Turns OFF the port extension.
Preset value	False or OFF

Examples

```
Dim Ext As Boolean
SCPI.SENSE(1).CORRection.EXTension.STATe = True
Ext = SCPI.SENSE(1).CORRection.EXTension.STATe
```

Related objects

```
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.DCOffset
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.LOSS
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.MEASure
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.PORT(Pt)
SCPI.SENSE(Ch).CORRection.EXTension.AUTO.RESet
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).INCLude(II).STATe
```

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC
 SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)
 SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > Port Extensions > Extensions

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension[:STATe] {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:EXTension[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT ON"
20 OUTPUT 717;":SENS1:CORR:EXT?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.IMPedance.INPut.MAGNitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).CORRection.IMPedance.INPut.MAGNitude = *Value**Value* = SCPI.SENSE(Ch).CORRection.IMPedance.INPut.MAGNitude

Description

This command sets/gets the system characteristic impedance (Z0) value.

Variable

Parameter	<i>Value</i>
Description	System Z0 value
Data type	Double precision floating point type (Double)
Range	1E-3 to 1000
Preset value	50
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim SysZ0 As Double

SCPI.SENSE.CORRection.IMPedance.INPut.MAGNitude = 75

SysZ0 = SCPI.SENSE.CORRection.IMPedance.INPut.MAGNitude

Equivalent key

Cal > Set Z0

Equivalent SCPI command

Syntax

```
:SENSe:CORRection:IMPedance[:INPut][:MAGNitude] <numeric>  
:SENSe:CORRection:IMPedance[:INPut][:MAGNitude]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:CORR:IMP 75"  
20 OUTPUT 717;":SENS:CORR:IMP?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.OFFSet.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.OFFSet.CLEar

Description

This command clears the error coefficient for calibration when the frequency offset feature is on, for the selected channel (*Ch*).

NOTE

This command does not clear the error coefficient when the frequency offset mode is OFF.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.CLEar

Related objects

SCPI.SENSE(Ch).OFFSet.STATe

SCPI.SENSE(Ch).CORRection.CLEar

Equivalent key

Cal > Mixer/Converter Calibration > Clear > OK

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:CLE"

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.LOAD

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.LOAD = *Ports*

Description

This command measures the calibration data of the load standard of the specified port when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	Provides 2-element array data (port). Ports(0): Measurement port number Ports(1): Frequency port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSE(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency and its subcommands" for the port specified by the frequency port number.

Example of use (1)

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.LOAD = Array(1,2)

Example of use (2)

Dim Port As Variant

Port(0) = 1

Port(1) = 2

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.LOAD = Port

Related objects

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.OPENSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.SHORT

Equivalent key

E5071C

Cal > Mixer/Converter Calibration > Scalar Cal(Manual) > Reflection > PortX@FreqY Broadband

- The value of PortX and FreqY changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:LOAD <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:LOAD 1,4"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.OPEN

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.OPEN = *Ports*

Description

This command measures the calibration data of the open standard of the specified port when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	Provides 2-element array data (port). Ports(0): Measurement port number Ports(1): Frequency port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSE(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency and its subcommands" for the port specified by the frequency port number.

Example of use (1)

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.OPEN = Array(1,2)

Example of use (2)

Dim Port As Variant

Port(0) = 1

Port(1) = 2

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.OPEN = Port

Related objects

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.LOADSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.SHORT

Equivalent key

Cal > Mixer/Converter Calibration > Scalar Cal(Manual) > Reflection > PortX@FreqY Open

- The value of PortX and FreqY changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:OPEN <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:OPEN 1,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.PMETer

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.PMETer = *Params*

Description

This command measures the scalar-mixer calibration data using the power meter when the frequency offset feature is *ON*, for the selected port and channel (*Ch*).

Variable

Parameter	<i>Params</i>
Description	<p>Provides 3-element array data (port).</p> <p>Params(0): Measurement port number</p> <p>Params(1): Frequency port number</p> <p>Params(2): Power sensor number in use</p> <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<p>Measurement port number/frequency port number 1 to 4</p> <p>Power sensor number in use</p> <ul style="list-style-type: none"> "ASENsor": Specifies power sensor A. "BSENsor": Specifies power sensor B.
Note	<p>The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency and its subcommands" for the port specified by the frequency port number.</p> <p>The setting of the power sensor is common to that for power meter calibration.</p>

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.PMETer = Array(1,2,"ASENsor")

E5071C

Dim Params As Variant

Params(0) = 1

Params(1) = 2

Params(2) = "ASENSor"

SCPI.SENSE(1).CORRection.OFFSet.COLLect.ACQuire.PMETer = Params

Related objects

SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORT

Equivalent key

Cal > Mixer/Converter Caribration > Scalar Cal(Manual) > Power Meter > Use Sensor A|Use Sensor B

Cal > Mixer/Converter Caribration > Scalar Cal(Manual) > Power Meter > PortX@FreqY

- The values of PortX and FreqY change depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSE{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:PMETer
<numeric 1>,<numeric 2>,{ASENSor|BSENSor}

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:PMET 1,4,ASEN"

20 OUTPUT 717;":*OPC?"

30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.SHORt

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.SHORt = *Ports*

Description

This command measures the calibration data of the short standard of the specified port when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	Provides 2-element array data (port). Ports(0): Measurement port number Ports(1): Frequency port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSE(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency and its subcommands" for the port specified by the frequency port number.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.SHORt = Array(1,2)

Dim Port As Variant

Port(0) = 1

Port(1) = 2

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.SHORt = Port

Related objects

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.LOADSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.OPEN

Equivalent key

Cal > Mixer/Converter Calibration > Scalar Cal(Manual) > Reflection > PortX@FreqY Short

- The value of PortX and FreqY changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

**:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:SHORT
<numeric 1>,<numeric 2>**

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:SHOR 1,4"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.THURU

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.THURU = *Ports*

Description

This command measures the calibration data of the thru standard of the specified port when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	Provides 2-element array data (port). Ports(0): Response port number Ports(1): Stimulus port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	For example, when THRU 1,2 is specified, S22 and S12 are measured; when THRU 2,1 is specified, S11 and S21 are measured. If you specify the same port number to the 2 port numbers, an error occurs when executed.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.THURU = Array(1,2)

Dim Port As Variant

Port(0) = 1

Port(1) = 2

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ACQuire.THURU = Port

Related objects

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.LOADSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.OPENSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.SHORT

Equivalent key

Cal > **Mixer/Converter Calibration** > **Scalar Cal(Manual)** > **Transmission** > **PortX-Y@FreqZ Thru**

- The value of PortX-Y and FreqZ changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:THRU <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:THRU 1,4"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.CLEar

Description

This command clears the calibration measurement data when the frequency offset feature is *ON*, for the selected channel (*Ch*).

- This command also clears the measurement data of the power meter.

Settings that have been temporarily changed due to measurement for each standard (number of traces, measurement parameters, and so on) return to their original values.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLLect.CLEar

Related objects

SCPI.SENSE(*Ch*).OFFSet.STATeSCPI.SENSE(*Ch*).CORRection.OFFSet.CLEar

Equivalent key

Cal > Mixer/Converter Calibration > Scalar Cal(Manual) > Cancel > OK

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:CLE"

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ECAL.SMIX2

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ECAL.SMIX2 = *Ports*

Description

This command executes calibration for the specified 2 ports when the frequency offset feature is on for selected channel (*Ch*) using ECal (Electronic Calibration).

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

Variable

Parameter	<i>Ports</i>
Description	Provides 2-element array data (port). Ports(0): Port number 1 Ports(1): Port number 2 The direction (forward or reverse) is determined by the presence/absence of power meter measurement data instead of port 1 or port 2 specified here. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	Before executing this command, the "SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.PM ETer" command must be executed. If you specify the same port number to the 2 port numbers, an error occurs when executed.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ECAL.SMIX2 = Array(1,2)

Dim Port As Variant
Port(0) = 1

Port(1) = 2
 SCPI.SENSE(1).CORRection.OFFSet.COLLect.ECAL.SMIX2 = Port

Related objects

SCPI.SENSE(Ch).OFFSet.STATe
 SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.PMETer
 SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ECAL.SOLT1

Equivalent key

Cal > Mixer/Converter Caribration > Scalar Cal(ESal) > ESal & Done

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:ECAL:SMIX2 <numeric
 1>,<numeric 2>

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:ECAL:SMIX2 1,2"
 20 OUTPUT 717;":*OPC?"
 30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ECAL.SOLT1

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ECAL.SOLT1 = *EPort*

Description

This command executes 1-port calibration for the specified port when the frequency offset feature is on for selected channel (*Ch*) using ECal (Electronic Calibration).

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

Variable

Parameter	<i>EPort</i>
Description	Executes 1-port calibration for frequency offset measurement.
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use (1)

SCPI.SENSE(1).CORRection.OFFSet.COLlect.ECAL.SOLT1 = 1

Related objects

SCPI.SENSE(*Ch*).OFFSet.STATeSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ECAL.SMIX2

Equivalent key

Cal > Mixer/Converter Caribration > Scalar Cal(ESol) > Select Ports > Port n**Cal > Mixer/Converter Caribration > Scalar Cal(ESol) > ECal & Done**

Equivalent SCPI command

Syntax

:SENSE{[1]-160}:CORRection:OFFSet:COLlect:ECAL:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:ECAL:SOLT1 2"  
20 OUTPUT 717;"*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.METHod.SMIX2

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.METHod.SMIX2 = *Ports*

Description

This command executes 2-port calibration for the specified 2 ports when the frequency offset feature is on for selected channel (*Ch*).

Variable

Parameter	<i>Ports</i>
Description	<p>Provides 2-element array data (port). Ports(0): Port number 1 Ports(1): Port number 2</p> <p>The direction (forward, reverse, or both) is determined depending on the port number specified with the "SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.LO AD", "SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.OP EN", "SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.SH ORt", and "SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.TH RU" command instead of port number 1 or port number 2 specified here.</p> <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	1 to 4
Note	If you specify the same port number to the 2 port numbers, an error occurs when executed.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLlect.METHod.SMIX2 = Array(1,2)

Dim Port As Variant
Port(0) = 1

```
Port(1) = 2
SCPI.SENSE(1).CORRection.OFFSet.COLLect.METHod.SMIX2 = Port
```

Related objects

```
SCPI.SENSE(Ch).OFFSet.STATe
SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD
SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN
SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORT
SCPI.SENSE(Ch).CORRection.OFFSet.COLLect.ACQuire.THURU
```

Equivalent key

Cal > Mixer/Converter Caribration > Scalar Cal(Manual) > Select Ports > 2-1(fwd)|1-2(rev)|1-2(both

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:OFFSet:COLLect:METHod:SMIX2 <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:METH:SMIX2 1,2"
```

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.METHod.SOLT1

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.METHod.SOLT1 = *Port*

Description

Executes 1-port calibration for the specified port when the frequency offset feature is on for selected channel (*Ch*). (No read)

Variable

Parameter	<i>Port</i>
Description	Executes 1-port calibration for frequency offset measurement.
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use (1)

SCPI.SENSE(1).CORRection.OFFSet.COLlect.METHod.SOLT1 = 2

Related objects

SCPI.SENSE(Ch).OFFSet.STATe

SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.METHod.SMIX2

Equivalent key

Cal > Mixer/Converter Caribration > Scalar Cal(Manual) > Select Ports > Port n

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:OFFSet:COLlect:METHod:SOLT1
<numeric>
```

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:METH:SOLT1 2"

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.SAVE

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.SAVE

Description

This command calculates the calibration coefficient for the selected calibration type from the calibration data measured with the frequency offset feature, for the selected channel (*Ch*).

- After the calibration coefficient is calculated, the measured data and the calibration type setting are cleared.

If this command is executed before all necessary calibration data for calculating the calibration coefficient is measured, an error occurs when executed.

Example of use

SCPI.SENSE(1).CORRection.OFFSet.COLlect.SAVE

Related objects

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.METHod.SMIX2SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.METHod.SOLT1SCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.LOADSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.OPENSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.SHORTSCPI.SENSE(*Ch*).CORRection.OFFSet.COLlect.ACQuire.THRU

Equivalent key

Cal > Mixer/Converter Caribration > Scalar Cal(Manual) > Done

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLlect:SAVE

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:SAVE"

SCPI.SENSE(*Ch*).CORRection.PROPerTy

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.PROPerTy = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.PROPerTy

Description

This command turns ON/OFF or returns the display of the calibration property, for the active trace of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the display of the calibration property
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none">• True or ON: Turns ON the display of the calibration property.• False or OFF: Turns OFF the display of the calibration property.
Preset value	False or OFF

Examples

```
Dim CalProp As Boolean
SCPI.SENSE(1).CORRection.PROPerTy = True
CalProp = SCPI.SENSE(1).CORRection.PROPerTy
```

Equivalent key

Cal > Property

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:PROPerTy {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:PROPerTy?
```

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:PROP ON"  
20 OUTPUT 717;":SENS1:CORR:PROP?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.ACQuire = *Src*

Description

This command executes receiver calibration, for measurement the selected port (*Pt*) of selected channel (*Ch*).

- Since information of power calibration for both the measurement port and the source port is used for error coefficient calculation, the precision of receiver calibration is improved by executing power calibration for both ports before executing receiver calibration.

Variable

Parameter	<i>Src</i>
Description	Specifies the source port for receiver correction.
Data type	Long integer type (Long)
Range	1 to 4
Note	If you specify the same port number as the measurement port number, an error occurs when executed.

Example of use

SCPI.SENSE(1).CORRection.RECeiver(2).COLLect.ACQuire = 4

Related objects

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).STATe

Equivalent key

Cal > Receiver Calibration > Calibrate Both

- From Firmware rev 9.2, the **Take Cal Sweep** softkey is renamed to **Calibrate Both** softkey.

Equivalent SCPI command

Syntax

```
:SENSE{[1]-160}:CORRection:RECeiver{[1]|2|3|4}:COLLect:ACQuire
<numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:COLL:ACQ 2"  
20 OUTPUT 717;"*OPC?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.RCHannel.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.RCHannel.ACQuire

Description

This command executes receiver calibration (R channel only), for measurement the selected port (*Pt*) of selected channel (*Ch*).

- Since information of power calibration for both the measurement port and the source port is used for error coefficient calculation, the precision of receiver calibration is improved by executing power calibration for both ports before executing receiver calibration.
- When **Calibrate R** is selected, the **Select Port** setting is used and the **Source Port** setting is ignored.

Example of use

SCPI.SENSE(1).CORRection.RECeiver(2).COLLect.RCHannel.ACQuire

Related objects

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).STATeSCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.ACQuireSCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.TCHannel.ACQuire

Equivalent key

Cal > Receiver Calibration > Calibrate R

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:CORRection:RECeiver{[1]|2|3|4}:COLLect:RCHannel:ACQuire
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:COLL:RCH:ACQ"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.TCHannel.ACQuire

Type of object

Method (**Write-only**)

Syntax

```
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).COLLect.TCHannel.ACQuire =
Src
```

Description

This command executes receiver calibration (T channel only), for measurement the selected port (*Pt*) of selected channel (*Ch*).

- Since information of power calibration for both the measurement port and the source port is used for error coefficient calculation, the precision of receiver calibration is improved by executing power calibration for both ports before executing receiver calibration.
- When **Calibrate Both** or **Calibrate T** is selected, the source port should be different from measurement port.

Variable

Parameter	<i>Src</i>
Description	Specifies the source port for receiver correction.
Data type	Long integer type (Long)
Range	1 to 4
Note	If you specify the same port number as the measurement port number, an error occurs when executed.

Example of use

```
SCPI.SENSE(1).CORRection.RECeiver(2).COLLect.TCHannel.ACQuire = 4
```

Related objects

```
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).STATe
```

```
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).COLLect.ACQuire
```

```
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).COLLect.RCHannel.ACQuire
```

Equivalent key

Cal > Receiver Calibration > **Calibrate T**

Equivalent SCPI command

E5071C

Syntax

```
:SENSe{[1]-  
160}:CORRection:RECeiver{[1]|2|3|4}:COLLect:TCHannel:ACQuire  
<numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:COLL:TCH:ACQ 2"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.RECeiver(Pt).OFFset.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.RECeiver(Pt).OFFset.AMPLitude = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.RECeiver(Pt).OFFset.AMPLitude

Description

This command sets/gets power offset value for Receiver Calibration. Receiver calibration is done at the condition of (source power)+(Specified offset), for selected channel (*Ch*).

For example, if the source Power is -10 dBm, then:

- If specified offset is 0 dBm, then Source Power Reading after calibration would be -10 dBm.
- If specified offset is 10 dBm, then Source Power Reading after calibration would be 0 dBm.
- If specified offset is -10 dBm, then Source Power Reading after calibration would be -20 dBm.

Variable

Parameter	<i>Value</i>
Description	Specified Offset
Data type	Double precision floating point type (Double)
Range	-200 to 200
Preset value	0
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

E5071C

Dim Vel As Double

SCPI.SENSE(1).CORRection.RECeiver(1).OFFset.AMPLitude = 50

Vel = SCPI.SENSE(1).CORRection.RECeiver(1).OFFset.AMPLitude

Equivalent key

Cal > Receiver Calibration > Power Offset

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:RECeiver([1]-4):OFFset:AMPLitude
<numeric>

:SENSe{[1]-160}:CORRection:RECeiver([1]-4):OFFset:AMPLitude?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:REC1:OFF:AMP 50"

20 OUTPUT 717;":SENS1:CORR:REC1:OFF:AMP?"

30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).STATe = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).STATe

Description

This command turns ON/OFF or gets the error correction for receiver calibration, for the measurement the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	On/off of error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the receiver calibration error correction. • False or OFF: Turns off the receiver calibration error correction.
Preset value	False or OFF

Example of use

```
Dim Corr As Boolean
SCPI.SENSE(1).CORRection.RECeiver(1).STATe = True
Corr = SCPI.SENSE(1).CORRection.RECeiver(1).STATe
```

Related objects

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLlect.ACQuire

Equivalent key

Cal > Receiver Calibration > Correction

Equivalent SCPI command

Syntax

```
:SENSE{[1]-160}:CORRection:RECeiver{[1]|2|3|4}[:STATe]
{ON|OFF|1|0}
```

E5071C

:SENSe{[1]-160}:CORRection:RECeiver{[1]|2|3|4}{:STATe}?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1 ON"  
20 OUTPUT 717;":SENS1:CORR:REC1?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.RVELocity.COAX**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.RVELocity.COAX = *Value**Value* = SCPI.SENSE(*Ch*).CORRection.RVELocity.COAX**Description**This command sets/gets the velocity factor, for selected channel (*Ch*).**Variable**

Parameter	<i>Value</i>
Description	Velocity factor
Data type	Double precision floating point type (Double)
Range	0 to 10
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim Vel As Double

SCPI.SENSE(1).CORRection.RVELocity.COAX = 0.5

Vel = SCPI.SENSE(1).CORRection.RVELocity.COAX

Equivalent key**Cal > Velocity Factor****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:CORRection:RVELocity:COAX <numeric>

:SENSe{[1]-160}:CORRection:RVELocity:COAX?

Query response

{numeric}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;":SENS1:CORR:RVEL:COAX 0.7"  
20 OUTPUT 717;":SENS1:CORR:RVEL:COAX?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).CORRection.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.STATe = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.STATe**Description**

This turns ON/OFF or gets the status of the error correction, for the active trace of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the error correction. • False or OFF: Turns OFF the error correction.
Preset value	False or OFF

Examples

```
Dim Corr As Boolean
SCPI.SENSE(1).CORRection.STATe = True
Corr = SCPI.SENSE(1).CORRection.STATe
```

Equivalent key**Cal > Correction****Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:CORRection:STATe {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:STATe?
```

Query response

```
{1|0}<newline><^END>
```

E5071C

Example of use

```
10 OUTPUT 717;".SENS1:CORR:STAT ON"  
20 OUTPUT 717;".SENS1:CORR:STAT?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).CORRection.TRIGger.FREE.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).CORRection.TRIGger.FREE.STATe = *Status**Status* = SCPI.SENSE(*Ch*).CORRection.TRIGger.FREE.STATe**Description**

This command sets/gets the trigger source for calibration to internal (ON) or to the state of the trigger source (SCPI.TRIGger.SEQUENCE.SOURCE) when measurement is made (OFF).

- When you change the trigger source during sweep, the sweep is canceled.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the trigger source at calibration
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Specifies internal for the trigger source for calibration. This corresponds to the softkey "Internal." • False or OFF: Matches the state of the trigger source when measurement is made. This corresponds to the softkey "System."
Preset value	True or ON

Examples

```
Dim TrigSour As Boolean
SCPI.SENSE(1).CORRection.TRIGger.FREE.STATe = False
TrigSour = SCPI.SENSE(1).CORRection.TRIGger.FREE.STATe
```

Equivalent key**Cal > Cal Trigger Source > Internal|System****Equivalent SCPI command****Syntax**

E5071C

:SENSe{[1]-160}:CORRection:TRIGger:FREE[:STATe] {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:TRIGger:FREE[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:CORR:TRIG:FREE OFF"
20 OUTPUT 717;":SENS1:CORR:TRIG:FREE?"
30 ENTER 717;A

SCPI.SENSE(*Ch*).CORRection.TYPE(*Tr*)

Object type

Property (**Read-only**)

Syntax

Data = SCPI.SENSE(*Ch*).CORRection.TYPE(*Tr*)

Description

This command reads the information (calibration type, port numbers) of the applied calibration coefficients for the actual error correction, for traces 1 to 36 (*Tr*) of selected channel (*Ch*).

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 5 array data items (the calibration type and the port information to which the calibration is applied).</p> <ul style="list-style-type: none"> • <i>Data(0)</i>: The calibration type applied. For detail, refer to the Range section. • <i>Data(1)</i>: The port number to which the calibration is applied (0 when the calibration type is NONE). • <i>Data(2)</i>: The port number to which the calibration is applied (0 when the calibration type is not SOLT2, SOLT3, or SOLT4). • <i>Data(3)</i>: The port number to which the calibration is applied (0 when the calibration type is not SOLT3 or SOLT4). • <i>Data(4)</i>: The port number to which the calibration is applied (0 when the calibration type is not SOLT4). <p>The array index starts from 0.</p>
Range	<p>One of the following is read out as <i>Data(0)</i>.</p> <ul style="list-style-type: none"> • "ERES": The enhanced response calibration is applied. • "NONE": Nothing is applied. • "RESPO": The response calibration (open) is applied. • "RESPS": The response calibration (short) is

	<p>applied.</p> <ul style="list-style-type: none">• "RESPT": The response calibration (thru) is applied.• "SMIX2": The scalar-mixer calibration is applied.• "SOLT1": The 1-port calibration is applied.• "SOLT2": The full 2-port calibration is applied.• "SOLT3": The full 3-port calibration is applied.• "SOLT4": The full 4-port calibration is applied.
Data type	Variant type (Variant)

Examples

Dim CalType As Variant
CalType = SCPI.SENSE(1).CORRection.TYPE(1)

Related objects

SCPI.SENSE(Ch).CORRection.COLLect.METHod.TRL2
SCPI.SENSE(Ch).CORRection.COLLect.METHod.TRL3
SCPI.SENSE(Ch).CORRection.COLLect.METHod.TRL4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSE{[1]-160}:CORRection:TYPE{[1]-36}?

Query response

{ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|SOLT3|SOLT4|SMIX2},
{numeric 1},{numeric 2}, {numeric 3},{numeric 4}<newline><^END>
{ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|SOLT3|SOLT
4|SMIX2}:

	Description
ERES	The enhanced response calibration is applied.
NONE	Any calibration is not applied.

RESPO	The response calibration (open) is applied.
RESPS	The response calibration (short) is applied.
RESPT	The response calibration (thru) is applied.
SMIX2	The scalar-mixer calibration is applied.
SOLT1	The 1-port calibration is applied.
SOLT2	The full 2-port calibration is applied.
SOLT3	The full 3-port calibration is applied.
SOLT4	The full 4-port calibration is applied.

{numeric 1}:

the calibration port number

(This parameter is 0 when the first parameter is NONE.)

the response port number when the SMIX2 is selected.

{numeric 2}:

the calibration port number

(This parameter is 0 when the first parameter is not ERES, RESPT, SOLT2, SOLT3, SOLT4 and SMIX2.)

the stimulus port number when the SMIX2 is selected.

{numeric 3}:

the calibration port number

(This parameter is 0 when the first parameter is not SOLT3 and SOLT4.)

{numeric 4}:

the calibration port number

(This parameter is 0 when the first parameter is not SOLT4.)

Example of use

```
10 OUTPUT 717;":SENS1:CORR:TYPE1?"
20 ENTER 717;A$
```

SCPI.SENSE(Ch).DATA.CORRdata(Param)

Object type

Property (**Read only**)

Syntax

Data = SCPI.SENSE(Ch).DATA.CORRdata(*Param*)

Description

This command gets S-Parameter data. Refer to Internal Data Processing.

The SCPI.SENSE(Ch).CORRection.DATA.CDATA command allows you to get the multiple S-parameter data by one command.

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select the following: <ul style="list-style-type: none"> • "S<XY>" Where: x=1to 4 Y=1to 4
Preset value	"S11"

Parameter	<i>Data</i>
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. <ul style="list-style-type: none"> • <i>Data</i>(<i>n</i>*2-2) :Real part of the data (complex number) at the n-th measurement point. • <i>Data</i>(<i>n</i>*2-1) :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.

Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))* 2 when setting a corrected data array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorData As Variant
CorData = SCPI.SENSE.DATA.CORRdata("S21")
```

Related objects

```
SCPI.SENSE(Ch).DATA.RAWData
SCPI.SENSE(Ch).CORRection.DATA.CDATA
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe:DATA:CORRdata?
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44}
```

Query response

```
{array}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:DATA:CORR? S21"
20 ENTER 717;A(*)
```

SCPI.SENSE(*Ch*).DATA.RAWData(*Param*)

Object type

Property (**Read only**)

Syntax

Data = SCPI.SENSE(*Ch*).DATA.RAWData(*Param*)

Description

This command gets the raw data. Refer to Internal Data Processing.

Variable

Parameter	<i>Param</i>
Description	Measurement parameter
Data type	Character string type (String)
Range	Select the following: <ul style="list-style-type: none"> • "S<XY>" Where: x=1to 4 Y=1to 4
Preset value	"S11"

Parameter	<i>Data</i>
Description	Indicates the array data (raw data array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. <ul style="list-style-type: none"> • <i>Data</i>(<i>n</i>*2-2) :Real part of the data (complex number) at the n-th measurement point. • <i>Data</i>(<i>n</i>*2-1) :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Note

If there is no array data of NOP (number of measurement point))* 2 when setting a raw data array, an error occurs when executed and the object is ignored.

Examples

```
Dim RawData As Variant
RawData = SCPI.SENSESe.DATA.RAWData("S21")
```

Related objects

SCPI.SENSESe(Ch).DATA.CORRdata

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe:DATA:RAWData?
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44}
```

Query response

```
{array}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:DATA:RAWD? S21"
20 ENTER 717;A(*)
```

SCPI.SENSE(*Ch*).FREQuency.CENTer**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).FREQuency.CENTer = *Value**Value* = SCPI.SENSE(*Ch*).FREQuency.CENTer**Description**

This command sets/gets the center value of the sweep range of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Center value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	(Maximum Frequency - 10E5)/2
Unit	Hz (hertz)
Resolution	0.5 or 1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim Cntr As Double

SCPI.SENSE(1).FREQuency.CENTer = 2E9

Cntr = SCPI.SENSE(1).FREQuency.CENTer

Related objectsSCPI.SENSE(*Ch*).FREQuency.SPAN**Equivalent key****Center**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:FREQuency:CENTer <numeric>  
:SENSe{[1]-160}:FREQuency:CENTer?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:CEN 2E9"  
20 OUTPUT 717;":SENS1:FREQ:CEN?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).FREQUENCY.CW**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).FREQUENCY.CW = *Value**Value* = SCPI.SENSE(Ch).FREQUENCY.CW**Description**

This command sets/gets the fixed frequency (CW frequency) for the power sweep for channels 1 to 36 (*Ch*).

- This object provides the same function as the SCPI.SENSE(Ch).FREQUENCY.FIXED object.

Variable

Parameter	<i>Value</i>
Description	Fixed frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CwFreq As Double
SCPI.SENSE(1).FREQUENCY.CW = 1E9
CwFreq = SCPI.SENSE(1).FREQUENCY.CW
```

Related objects

SCPI.SENSE(Ch).FREQUENCY.FIXED

SCPI.SENSE(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Power** > **CW Freq**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency[:CW|FIXed] <numeric>

:SENSe{[1]-160}:FREQuency[:CW|FIXed]?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:FREQ 1E9"

20 OUTPUT 717;":SENS1:FREQ?"

30 ENTER 717;A

SCPI.SENSE(Ch).FREQUENCY.DATA

Object type

Property (**Read-only**)

Syntax

Data = SCPI.SENSE(Ch).FREQUENCY.DATA

Description

This command reads the frequencies at all measurement points of channels 1 to 36 (Ch).

Variable

Parameter	<i>Data</i>
Description	Indicates the array data (frequency) of NOP (number of measurement points). Where n is an integer between 1 and NOP. <ul style="list-style-type: none"><i>Data(n-1)</i>: Frequency at the n-th measurement point The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim FreqData As Variant
SCPI.SENSE(1).SWEep.POINTs = 201
FreqData = SCPI.SENSE(1).FREQUENCY.DATA
```

Related objects

SCPI.SENSE(Ch).SWEep.POINTs

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQUENCY:DATA?

Query response

{numeric 1},
{numeric NOP}<newline><^END>

Description

{numeric n}	Frequency at the n-th measurement point
----------------	-----------------------------------------

Where NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;";SENS1:FREQ:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSE(Ch).FREQUENCY.FIXed**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).FREQUENCY.FIXed = *Value**Value* = SCPI.SENSE(Ch).FREQUENCY.FIXed**Description**

This command sets/gets the fixed frequency (CW frequency) for the power sweep for channels 1 to 36 (*Ch*).

- This object provides the same function as SCPI.SENSE(Ch).FREQUENCY.CW.

Variable

Parameter	<i>Value</i>
Description	Fixed frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CwFreq As Double
SCPI.SENSE(1).FREQUENCY.FIXed = 1E9
CwFreq = SCPI.SENSE(1).FREQUENCY.FIXed
```

Related objects

SCPI.SENSE(Ch).FREQUENCY.CW

SCPI.SENSE(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Power** > **CW Freq**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:FREQuency[:CW|FIXed] <numeric>
:SENSe{[1]-160}:FREQuency[:CW|FIXed]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ 1E9"
20 OUTPUT 717;":SENS1:FREQ?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).FREQUENCY.SPAN**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).FREQUENCY.SPAN = *Value**Value* = SCPI.SENSE(*Ch*).FREQUENCY.SPAN**Description**

This command sets/gets the span value of the sweep range of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Span value
Data type	Double precision floating point type (Double)
Range	0 to (Maximum Frequency - 10E5)
Preset value	Maximum Frequency - 10E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim Span As Double

SCPI.SENSE(1).FREQUENCY.SPAN = 1E9

Span = SCPI.SENSE(1).FREQUENCY.SPAN

Related objectsSCPI.SENSE(*Ch*).FREQUENCY.CENTER**Equivalent key****Span**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:FREQuency:SPAN <numeric>
:SENSe{[1]-160}:FREQuency:SPAN?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:SPAN 1E9"
20 OUTPUT 717;":SENS1:FREQ:SPAN?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).FREQUENCY.START**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).FREQUENCY.START = *Value**Value* = SCPI.SENSE(*Ch*).FREQUENCY.START**Description**

This command sets/gets the start value of the sweep range of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Start value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Start As Double
SCPI.SENSE(1).FREQUENCY.START = 100E6
Start = SCPI.SENSE(1).FREQUENCY.START
```

Related objectsSCPI.SENSE(*Ch*).FREQUENCY.STOP**Equivalent key****Start**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:FREQuency:STARt <numeric>  
:SENSe{[1]-160}:FREQuency:STARt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:STAR 100E6"  
20 OUTPUT 717;":SENS1:FREQ:STAR?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).FREQUENCY.STOP**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).FREQUENCY.STOP = *Value**Value* = SCPI.SENSE(*Ch*).FREQUENCY.STOP**Description**

This command sets/gets the stop value of the sweep range of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Stop value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	Maximum frequency in the range
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Stp As Double
SCPI.SENSE(1).FREQUENCY.STOP = 3E9
Stp = SCPI.SENSE(1).FREQUENCY.STOP
```

Related objectsSCPI.SENSE(*Ch*).FREQUENCY.START**Equivalent key****Stop**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:FREQuency:STOP <numeric>
:SENSe{[1]-160}:FREQuency:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:STOP 100E6"
20 OUTPUT 717;":SENS1:FREQ:STOP?"
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).COUNT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SENSE(Ch).MULTiplexer(*Id*).COUNT

Description

This command reads the number of ports of the Multiport test set whose ID is 1 or 2 (*Id*).

Variable

Parameter	<i>Value</i>
Description	The number of ports
Data type	Long integer type (Long)
Range	28: E5092A 16: E5091A-016 9: E5091A-009 0: E5091A/E5092A is not connected.

Examples

```
Dim NPort As Long
NPort = SCPI.SENSE.MULTiplexer(1).COUNT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:COUNT?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:COUN?"
20 ENTER 717;A
```


SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATe = *Status**Status* = SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATe**Description**

This command turns ON/OFF or gets the status of the property display (the state of the port assignment) of the Multiport test set whose ID is *Id*.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the property display
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: Turns ON the property display. • False or OFF: Turns OFF the property display.
Preset value	False or OFF

Examples

```
Dim Prop As Boolean
SCPI.SENSE.MULTiplexer(1).DISPlay.STATe = True
Prop = SCPI.SENSE.MULTiplexer(1).DISPlay.STATe
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).CATalog
SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).SElect
```

Equivalent key**System > Multiport Test Set Setup > Test Set 1|Test Set 2 > Property****Equivalent SCPI command****Syntax**

E5071C

:SENSe:MULTiplexer{[1]|2}:DISPlay[:STATe] {ON|OFF|1|0}
:SENSe:MULTiplexer{[1]|2}:DISPlay[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":SENS:MULT1:DISP ON"
20 OUTPUT 717;":SENS:MULT1:DISP?"
30 ENTER 717;A

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.A.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.A.DATA**Description**

This command sets/gets the output port data for group A with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output port data for group A
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0

Examples

```
Dim Obj As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.A.DATA = 10
Obj = SCPI.SENSE.MULTiplexer1.OUTPut.A.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line A** > **Line [1-8]** <LOW|HIGH>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:OUTPut:A[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:A[:DATA]?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:A.DATA 10"  
20 OUTPUT 717;":SENS:MULT1:OUTP:A.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA

Description

This command sets/gets the output voltage of the group A with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group A
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

Dim Obj As Double

SCPI.SENSE.MULTiplexer(1).OUTPut.A.VOLTage.DATA =0

Obj = SCPI.SENSE.MULTiplexer1.OUTPut.A.VOLTage.DATA

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line A** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

E5071C

```
:SENSe:MULTiplexer{[1]|2}:OUTPut:A.VOLTage.[:DATA] <numeric>  
:SENSe:MULTiplexer{[1]|2}:OUTPut:A.VOLTage.[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:A.VOLT.DATA 0"  
20 OUTPUT 717;":SENS:MULT1:OUTP:A.VOLT.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.B.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.B.DATA**Description**

This command sets/gets the output port data for group B with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output port data for group B
Data type	Long integer type (Long)
Range	0 to 15
Preset value	0

Examples

```
Dim Prop As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.B.DATA = 10
Prop = SCPI.SENSE.MULTiplexer1.OUTPut.B.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line B** > **Line [1-4]** <LOW|HIGH>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:OUTPut:B[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:B[:DATA]?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:B:DATA 10"  
20 OUTPUT 717;":SENS:MULT1:OUTP:B:DATA?"  
30 ENTER 717;A
```


SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group B with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group B
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSE.MULTiplexer(1).OUTPut.B.VOLTage.DATA =0
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.B.VOLTage.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line B** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSE:MULTiplexer{[1]|2}:OUTPut:B.VOLTage[:DATA] <numeric>

E5071C

:SENSe:MULTiplexer{[1]|2}:OUTPut:B.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:B.VOLT.DATA 0"  
20 OUTPUT 717;":SENS:MULT1:OUTP:B.VOLT.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.C.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(*Id*).OUTPut.C.DATA**Description**

This command sets/gets the output port data for group C with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output port data for group C
Data type	Long integer type (Long)
Range	0 to 15
Preset value	0

Examples

```
Dim Prop As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.C.DATA = 10
Prop = SCPI.SENSE.MULTiplexer(1).OUTPut.C.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line C** > **Line [1-4]** <LOW|HIGH>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:OUTPut:C[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:C[:DATA]?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:C.DATA 10"  
20 OUTPUT 717;":SENS:MULT1:OUTP:C.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group C with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group C
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSE.MULTiplexer(1).OUTPut.C.VOLTage.DATA = 0
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.C.VOLTage.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line C** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:OUTPut:C.VOLTage[:DATA] <numeric>

E5071C

:SENSe:MULTiplexer{[1]|2}:OUTPut:C.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:C.VOLT.DATA 0"  
20 OUTPUT 717;":SENS:MULT1:OUTP:C.VOLT.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.DATA**Description**

This command sets/gets the output port data for group D with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output port data for group D
Data type	Long integer type (Long)
Range	0 to 15
Preset value	0

Examples

```
Dim Prop As Long
SCPI.SENSE.MULTiplexer(1).OUTPut.D.DATA = 10
Prop = SCPI.SENSE.MULTiplexer(1).OUTPut.D.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line D** > **Line [1-4]** <LOW|HIGH>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:OUTPut:D[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:D[:DATA]?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:D.DATA 10"  
20 OUTPUT 717;":SENS:MULT1:OUTP:D.DATA?"  
30 ENTER 717;A
```


SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA**Description**

This command sets/gets the output voltage of the group D with ID of the E5092A.

Variable

Parameter	<i>Value</i>
Description	Output voltage for group D
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSE.MULTiplexer(1).OUTPut.D.VOLTage.DATA = 0
Obj = SCPI.SENSE.MULTiplexer(1).OUTPut.D.VOLTage.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line D** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:OUTPut:D.VOLTage[:DATA] <numeric>

E5071C

:SENSe:MULTiplexer{[1]|2}:OUTPut:D.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:D.VOLT.DATA 0"  
20 OUTPUT 717;":SENS:MULT1:OUTP:D.VOLT.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.DATA**Description**

This command sets/gets or returns the status of HIGH/LOW of all the control lines of the E5091A whose ID is 1 to 2 (*Id*) when measuring selected channel (*Ch*) in the measurement using the E5091A.

- To set the control lines, use the values obtained by converting 8-bit binary values expressed HIGH (1) / LOW (0) of individual lines to decimal values, assuming line 1 as LSB and line 8 as MSB.

Variable

Parameter	<i>Value</i>
Description	Sets/gets control line value
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim OData As Long
SCPI.SENSE.MULTiplexer1.OUTPut.DATA = "0"
MultName = SCPI.SENSE.MULTiplexer1.OUTPut.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).STATE

Equivalent key**System > Multiport Test Set Setup > Test Set 1|Test Set 2 > Control Lines**

- This command is available only for E5091A multiport test set.

E5071C

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:MULTiplexer{[1]|2}:OUTPut[:DATA] <numeric>  
:SENSe{[1]-160}:MULTiplexer{[1]|2}:OUTPut[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:OUTP.DATA 5"  
20 OUTPUT 717;":SENS1:MULT1:OUTP.DATA?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).PORT(*Pt*).CATalog**Object type**Property (**Read-only**)**Syntax***String* = SCPI.SENSE.MULTiplexer(*Id*).PORT(*Pt*).CATalog**Description**

This command returns a port assigned to Port 1 to 20 (*Pt*) of the E5091A whose ID is 1 to 2 (*Id*) when measuring selected channel (*Ch*) in the measurement using the E5091A.

Examples

```
Dim PortCat As String
PortCat = SCPI.SENSE.MULTiplexer1.PORT4.CATalog
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATe
SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).SElect
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSE{[1]-
160}:MULTiplexer{[1]|2}:PORT{[1]|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
16|17|18|19|20}:CATalog?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:PORT1:CAT?"
20 ENTER 717;A$
```

SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).SElect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE.MULTiplexer(Id).PORT(Pt).SElect = *String**String* = SCPI.SENSE.MULTiplexer(Id).PORT(Pt).SElect

Description

This command sets/gets a port name assigned to Port 1 to 20 (*Pt*) of the Multiport test set whose ID is 1 to 2 (*Id*) when measuring selected channel (*Ch*) in the measurement using the multiport test set.

- The range of this command for port 3 and port 4 is R1/R2/R3/R4 (without the + or - sign). However, the firmware displays the value with - or + sign, for example R1+ or R1-. For using this command, the + or - character is not required in the value and the command should be sent without any sign (+ or -).

Variable

When the 9-port option of the E5091A (E5091A-009) is selected

Parameter	<i>String</i>			
Description	Sets a port name assigned to Port 1 to 20			
Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	A or T1	T1 or T2	R1,R2,R3	R1,R2,R3
Preset value	A	T1	R1	R1

When the 13-port configuration of the E5091A-016 (E5091_13) or E5092A (E5092_13) is selected

Parameter	<i>String</i>
Description	ON/OFF of the control of the E5091A\E5092A

Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	A,T1,T2,T3	T1,T2,T3,T4	R1,R2,R3,R4	R1,R2,R3,R4
Preset value	A	T1	R1	R1

When the 16-port configuration of the E5091A-016 (E5091_16) is selected

Parameter	<i>String</i>						
Description	ON/OFF of the control of the E5091A						
Data type	String type						
Range	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7
	A1(A),A2,A3,A4,T1,T2,T3	B1(T4),B2,B3,B4,T1,T2,T3	R1,R2,R3,R4	R1,R2,R3,R4	X1 and X2	Y1 and Y2	Z1 and Z2
Preset value	A1	B1	R1	R1	X1	Y1	Z1

When the 16-port configuration of the E5092A (E5092_16) is selected

Parameter	<i>String</i>
-----------	---------------

Parameter							
Description	ON/OFF of the control of the E5092A						
Data type	String type						
Range	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7
	A1,A2,A3,A4	B1,B2,B3,B4	R1,R2,R3,R4	R1,R2,R3,R4	X1 and X2	Y1 and Y2	Z1 and Z2
Preset value	A1	B1	R1	R1	X1	Y1	Z1

When the 22-port configuration of the E5092A (E5092_22) is selected

Parameter	<i>String</i>			
Description	ON/OFF of the control of the E5092A			
Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	A1,A2,A3,A4,A5,A6	A7,A8,A9,A10,A11	B1,B2,B3,B4,B5,B6	B7,B8,B9,B10,B11
Preset value	A1	A7	B1	B7

When the 28-port configuration of the E5092A (E5092_28) is selected

Parameter	<i>String</i>									
Description	ON/OFF of the control of the E5092A									
Data type	String type									
Range	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7	Port 8	Port 9	Port 10
	A,B,C,D	A,B,C,D	A,B,C,D	A,B,C,D	A,B	A,B	A,B	A,B	A,B	A,B
Preset value	A	A	A	A	A	A	A	A	A	A

When the 10-port full crossbar configuration of the E5092A (E5092_X10) is selected

Parameter	<i>String</i>			
Description	ON/OFF of the control of the E5092A			
Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	1,3,5,7	2,4,6,8	2,4,6,10	1,3,5,9

Preset value	1	2	2	1
---------------------	---	---	---	---

For information on the variable (*Ch*), (*Id*) and (*Pt*), see Ch, Id and Pt.

Examples

```
Dim PortSel As String
SCPI.SENSE.MULTiplexer1.PORT4.SELect = "R2"
PortSel = SCPI.SENSE.MULTiplexer1.PORT4.SELect
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATe
SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).CATalog
SCPI.SENSE(Ch).MULTiplexer(Id).STATe
```

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Port 1|Port 2|Port 3|Port 4|Port 5|Port 6|Port 7|Port 8|Port 9|Port 10**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:MULTiplexer{[1]|2}:PORT{[1]|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
16|17|18|19|20}[:SELect] <string>
:SENSe{[1]-
160}:MULTiplexer{[1]|2}:PORT{[1]|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
16|17|18|19|20}[:SELect]?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:PORT1 A"
20 OUTPUT 717;":SENS1:MULT1:PORT1?"
30 ENTER 717;A$
```

SCPI.SENSE(Ch).MULTiplexer(Id).STATE**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).STATE = *Status**Status* = SCPI.SENSE(Ch).MULTiplexer(Id).STATE**Description**

This command turns ON/OFF or return the status of the control (switching the internal switch that connects between the ports and changing control line output) of the Multiport test set whose ID is *Id*.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the control of the Multiport test set
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the control of the multiport test set. • False or OFF: Turns OFF the control of the multiport test set.
Preset value	False or OFF

Examples

```
Dim Cont As Boolean
SCPI.SENSE.MULTiplexer1.STATE = True
Cont = SCPI.SENSE.MULTiplexer1.STATE
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).OUTPut.DATA
SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).SElect
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.OUTPut.DATA
```

E5071C

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Control**

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:STATe {ON|OFF|1|0}

:SENSe:MULTiplexer{[1]|2}:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:STAT ON"  
20 OUTPUT 717;":SENS:MULT1:STAT?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.OUTPUT.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.OUTPUT.DATA = *Value**Value* = SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.OUTPUT.DATA**Description**

This command sets/gets the HIGH/LOW of all the control line of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

- To set the control lines, use values obtained by converting 8-bit binary values expressed by HIGH (1)/LOW (0) of individual lines to decimal values, assuming line 1 as LSB and line 8 as MSB.

Variable

Parameter	<i>Value</i>
Description	Setting value the control line
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim C_line As Long
SCPI.SENSE(1).MULTiplexer1.TSET9.OUTPUT.DATA = 5
C_line = SCPI.SENSE(1).MULTiplexer1.TSET9.OUTPUT.DATA
```

Related objects

SCPI.SENSE(Ch).MULTiplexer(Id).STATE

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Control Lines** >
Line 1|...|Line 8

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:OUTPut[:DATA] <numeric>  
:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:OUTPut[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:TSET9:OUTP 5"  
20 OUTPUT 717;":SENS1:MULT1:TSET9:OUTP?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT1 = *Param**Param* = SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT1

Description

This command sets/gets a port assigned to Port 1 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

- If the port assigned to Port 2 is T1 and you select T1 as the port assigned to Port 1, the port assigned to Port 2 is changed to T2 automatically.

Variable

Parameter	<i>Param</i>
Description	The port assigned to Port 1
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "A": Selects A as the port assigned to Port 1. • "T1": Selects T1 as the port assigned to Port 1.
Preset Value	A

Examples

```
Dim Port As String
SCPI.SENSE(1).MULTiplexer1.TSET9.PORT1 = "T1"
Port = SCPI.SENSE(1).MULTiplexer1.TSET9.PORT1
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).STATE
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATE
```

E5071C

SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Port 1** > **A|T1**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT1 {A|T1}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT1?

Query response

{A|T1}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT1 T1"

20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT1?"

30 ENTER 717;A\$

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT2**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT2 = *Param**Param* = SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT2**Description**

This command sets/gets a port assigned to Port 2 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

- If the port assigned to Port 1 is T1 and you select T1 as the port assigned to Port 2, the port assigned to Port 1 is changed to A automatically.

Variable

Parameter	<i>Param</i>
Description	The port assigned to Port 2
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "T1": Selects T1 as the port assigned to Port 2. • "T2": Selects T2 as the port assigned to Port 2.
Preset Value	T1

Examples

```
Dim Port As String
SCPI.SENSE(1).MULTiplexer(1).TSET9.PORT2 = "T2"
Port = SCPI.SENSE(1).MULTiplexer(1).TSET9.PORT2
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).STATE
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATE
```

E5071C

SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Port2** > **T1|T2**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT2 {T1|T2}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT2?

Query response

{A|T1}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT2 T2"

20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT2?"

30 ENTER 717;A\$

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT3**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT3 = *Param**Param* = SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT3**Description**

This command sets/gets a port assigned to Port 3 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

Variable

Parameter	<i>Param</i>
Description	The port assigned to Port 3
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "R1": Selects R1+ as the port assigned to Port 3. • "R2": Selects R2+ as the port assigned to Port 3. • "R3": Selects R3+ as the port assigned to Port 3.
Preset Value	R1

Examples

```
Dim Port As String
SCPI.SENSE(1).MULTiplexer(1).TSET9.PORT3 = "R2"
Port = SCPI.SENSE(1).MULTiplexer(1).TSET9.PORT3
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).STATE
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATE
```

E5071C

SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Port3** >
R1+|R2+|R3+

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT3 {R1|R2}R3}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT3?

Query response

{R1|R2}R3}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT3 R2"

20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT3?"

30 ENTER 717;A\$

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT4

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT3 = *Param**Param* = SCPI.SENSE(*Ch*).MULTiplexer(*Id*).TSET9.PORT3

Description

This command sets/gets a port assigned to Port 4 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

Variable

Parameter	<i>Param</i>
Description	The port assigned to Port 4
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "R1": Selects R1- as the port assigned to Port 4. • "R2": Selects R2- as the port assigned to Port 4. • "R3": Selects R3- as the port assigned to Port 4.
Preset Value	R1

Examples

```
Dim Port As String
SCPI.SENSE(1).MULTiplexer(1).TSET9.PORT4 = "R2"
Port = SCPI.SENSE(1).MULTiplexer(1).TSET9.PORT4
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).STATE
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).DISPlay.STATE
```

E5071C

SCPI.SENSE(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Port4** > **R1-|R2-|R3-**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT4 {R1|R2|R3}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT4?

Query response

{R1|R2}R3}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT4 R2"

20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT4?"

30 ENTER 717;A\$

SCPI.SENSE(*Ch*).OFFSet.LOCal.CONTRol.STATe**Type of object**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).OFFSet.LOCal.CONTRol.STATe = *Status**Status* = SCPI.SENSE(*Ch*).OFFSet.LOCal.CONTRol.STATe**Description**

This command turns ON/OFF or return the status of the external signal source control mode, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	On/off of the external signal source
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the external signal source control mode. • False or OFF: Turns off the external signal source control mode.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSE(1).OFFSet.LOCal.CONTRol.STATe = False
State = SCPI.SENSE(1).OFFSet.LOCal.CONTRol.STATe
```

Equivalent key**Sweep Setup > Frequency Offset > External Source > Control****Equivalent SCPI command****Syntax**

```
:SENSe{[1]-160}:OFFSet:LOCAl:CONTRol[:STATe] {ON|OFF|1|0}
:SENSe{[1]-160}:OFFSet:LOCAl:CONTRol[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

E5071C

Example of use

```
10 OUTPUT 717;".SENS1:OFFS:LOC:CONT ON"  
20 OUTPUT 717;".SENS1:OFFS:LOC:CONT?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.DATA**Type of object**Property (**Read-only**)**Syntax***Data* = SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.DATA**Description**

This command acquires the external signal source frequency data, for the selected channel (*Ch*).

Variable

Parameter	<i>Data</i>
Description	Reads out the frequencies of all measurement points as an array.
Data type	Variant type (Variant)

Example of use

```
Dim LoData As Variant
LoData = SCPI.SENSE(1).OFFSet.LOCal.FREQuency.DATA
```

Related objects

SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.DIVisor
 SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.MULTIplier
 SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.OFFSet
 SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.START
 SCPI.SENSE(*Ch*).OFFSet.LOCal.FREQuency.STOP

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:OFFSet:LOCAl[:FREQuency]:DATA?
```

Query response

```
{value 1},  
{value NOP}<newline><^END>
```

NOTE Where, NOP is the number of points.

Example of use

E5071C

```
10 DIM A(1:201)
20 OUTPUT 717;" :SENS1:OFFS:LOC:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor

Description

This command sets/gets the external signal source frequency by using a divisor value for the basic frequency, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a divisor value for the basic frequency
Data type	Double precision floating point type (Double)
Range	1 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Divisor As Double
SCPI.SENSE(1).OFFSet.LOCal.FREQuency.DIVisor = 50
Divisor = SCPI.SENSE(1).OFFSet.LOCal.FREQuency.DIVisor
```

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTIplier

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STARTt

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > External Source > Divisor

E5071C

- The basic frequency range is set by using `SCPI.SENSE(Ch).FREQUENCY.START` and `SCPI.SENSE(Ch).FREQUENCY.STOP`.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:DIVisor <numeric>  
:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:DIVisor?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:DIV 1E1"  
20 OUTPUT 717;":SENS1:OFFS:LOC:DIV?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTiplier

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTiplier = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTiplier

Description

This command sets/gets the external signal source frequency by using a multiplier value for the basic frequency, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a multiplier value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-100 to 100
Preset value	0
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

Dim Multiplier As Double

SCPI.SENSE(1).OFFSet.LOCal.FREQuency.MULTiplier = -10

Multiplier = SCPI.SENSE(1).OFFSet.LOCal.FREQuency.MULTiplier

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.START

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > External Source > Multiplier

E5071C

- The basic frequency range is set by using "SCPI.SENSE(Ch).FREQUENCY.START" and "SCPI.SENSE(Ch).FREQUENCY.STOP".

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:MULTiplier <numeric>
:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:MULTiplier?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:MULT -1E1"  
20 OUTPUT 717;":SENS1:OFFS:LOC:MULT?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet

Description

This command sets/gets the external signal source frequency is set by using an offset value for the basic frequency, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of an offset value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Offset As Double
SCPI.SENSE(1).OFFSet.LOCal.FREQuency.OFFSet = -10
Offset = SCPI.SENSE(1).OFFSet.LOCal.FREQuency.OFFSet
```

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor
 SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTIplier
 SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.START
 SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > External Source > Offset

- The basic frequency range is set by using "SCPI.SENSE(Ch).FREQUENCY.START" and "SCPI.SENSE(Ch).FREQUENCY.STOP".

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCAl[:FREQuency]:OFFSet <numeric>

:SENSe{[1]-160}:OFFSet:LOCAl[:FREQuency]:OFFSet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:OFFS 5E9"  
20 OUTPUT 717;":SENS1:OFFS:LOC:OFFS?"  
30 ENTER 717;A
```


SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STARt

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STARt = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STARt

Description

This command sets/gets a start value for the external signal source frequency setting, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a start value for the external signal source
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Start As Double
SCPI.SENSE(1).OFFSet.LOCal.FREQuency.STARt = 100E6
Start = SCPI.SENSE(1).OFFSet.LOCal.FREQuency.STARt
```

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTIplier

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet

E5071C

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > External Source > Start

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:LOCAl[:FREQuency]:STARt <numeric>  
:SENSe{[1]-160}:OFFSet:LOCAl[:FREQuency]:STARt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:STAR 5E9"  
20 OUTPUT 717;":SENS1:OFFS:LOC:STAR?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.STOP

Description

This command sets/gets a stop value for the external signal source frequency setting, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a stop value for the external signal source
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Stop As Double
SCPI.SENSE(1).OFFSet.LOCal.FREQuency.STOP = 100E6
Stop = SCPI.SENSE(1).OFFSet.LOCal.FREQuency.STOP
```

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.MULTIplier

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSE(Ch).OFFSet.LOCal.FREQuency.START

E5071C

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Stop**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:STOP <numeric>

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:STOP 5E9"  
20 OUTPUT 717;":SENS1:OFFS:LOC:STOP?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level value for the external signal source power setting, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a power level value for the external signal source
Data type	Double precision floating point type (Double)
Range	-150 to 30
Preset value	-10
Unit	dBm
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

Dim Power As Double

SCPI.SENSE(1).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude = -10

Power = SCPI.SENSE(1).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > Frequency Offset > External Source > Power

E5071C

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel][:IMMediate] [:AMPLitude]
<numeric>

:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel][:IMMediate]
[:AMPLitude]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:POW -12.5"  
20 OUTPUT 717;":SENS1:OFFS:LOC:POW?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA = *Value**Value* = SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Description

This command sets/gets a power slope value for the external signal source, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a power slope value for the external signal source
Data type	Double precision floating point type (Double)
Range	-2 to 2
Preset value	0
Unit	dB/GHz
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

Dim Slope As Double

SCPI.SENSE(1).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA = -1

Slope = SCPI.SENSE(1).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Related objects

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

SCPI.SENSE(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > Frequency Offset > External Source > Slope

E5071C

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel]:SLOPe[:DATA]  
<numeric>  
:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel]:SLOPe[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP 1.0"  
20 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe = *Status**Status* = SCPI.SENSE(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Description

This command turns ON/OFF or returns the power slope value setting mode for the external signal source, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	Turn the power slope setting ON/OFF, for the external signal source.
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the power slope mode. • False or OFF: Turns off the power slope mode.
Preset value	False or OFF

Example of use

Dim State As Boolean

SCPI.SENSE(1).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe = False

State = SCPI.SENSE(1).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Related objects

SCPI.SENSE(*Ch*).OFFSet.LOCal.POWer.LEVel.IMMEdiate.AMPLitudeSCPI.SENSE(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Equivalent key

Sweep Setup > Frequency Offset > External Source > Slope

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:LOCAl:POWer[:LEVel]:SLOPe:STATe
{ON|OFF|1|0}
```

```
:SENSe{[1]-160}:OFFSet:LOCAl:POWer[:LEVel]:SLOPe:STATe?
```

E5071C

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP:STAT ON"  
20 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP:STAT?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).OFFSet.LOCal.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).OFFSet.LOCal.STATe = *Status**Status* = SCPI.SENSE(*Ch*).OFFSet.LOCal.STATe

Description

This command turns ON/OFF or return the external signal source frequency setting mode regardless of status of the frequency offset, for the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	On/off of external signal source frequency setting
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the frequency setting. • False or OFF: Turns off the frequency setting.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSE(1).OFFSet.LOCal.STATe = False
State = SCPI.SENSE(1).OFFSet.LOCal.STATe
```

Equivalent key

Setup > Frequency Offset > External Source > LO Frequency

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCAl:STATe {ON|OFF|1|0}

:SENSe{[1]-160}:OFFSet:LOCAl:STATe?

Query response

E5071C

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:STAT ON"  
20 OUTPUT 717;":SENS1:OFFS:LOC:STAT?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.DATA

Type of object

Property (**Read-only**)

Syntax

Data = SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.DATA

Description

This command reads the frequency data for the selected port (*Pt*) when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	<i>Data</i>
Description	Reads out the frequencies of all measurement points as an array.
Data type	Variant type (Variant)

Example of use

```
Dim Freqdata As Variant
Freqdata = SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.DATA
```

Related objects

```
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.MULTIplier
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.START
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STOP
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:DATA?
```

Query response

```
{value 1},...,{value NOP}<newline><^END>
```

Where, NOP is the number of points.

Example of use

E5071C

```
10 DIM A(1:201)
20 OUTPUT 717;" :SENS1:OFFS:PORT2:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor = *Value**Value* = SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

Description

This command sets/gets the selected port (*Pt*) frequency by using a divisor value for the basic frequency when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a frequency divisor value for the basic frequency
Data type	Double precision floating point type (Double)
Range	1 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

Dim Divisor As Double

SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.DIVisor = 50

Divisor = SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.DIVisor

Related objects

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.MULTIplier

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STARt

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > Port n > Divisor

E5071C

- The basic frequency range is set by using "SCPI.SENSE(Ch).FREQUENCY.START" and "SCPI.SENSE(Ch).FREQUENCY.STOP".

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQUENCY]:DIVisor
<numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQUENCY]:DIVisor?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:DIV 1E1"  
20 OUTPUT 717;":SENS1:OFFS:PORT2:DIV?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.MULTIplier

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.MULTIplier = *Value**Value* = SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.MULTIplier

Description

This command sets/gets the selected port (*Pt*) frequencies by using a multiplier value for the basic frequency when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a frequency multiplier value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-100 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

Dim Multiplier As Double

SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.MULTIplier = -10

Multiplier = SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.MULTIplier

Related objects

SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.DIVisorSCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.OFFSetSCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.STARtSCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > Port n > Multiplier

E5071C

- The basic frequency range is set by using "SCPI.SENSE(Ch).FREQUENCY.START" and "SCPI.SENSE(Ch).FREQUENCY.STOP".

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency] :MULTiplier
<numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency] :MULTiplier?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:MULT 1E1"  
20 OUTPUT 717;":SENS1:OFFS:PORT2:MULT?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet = *Value**Value* = SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

Description

This command sets/gets the selected port (*Pt*) frequency by using an offset value for the basic frequency when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a frequency offset value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

Dim Offset As Double

SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.OFFSet = 1E9

Offset = SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.OFFSet

Related objects

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.MULTIplier

E5071C

SCPI.SENSE(Ch).OFFSET.PORT(Pt).FREQUENCY.START

SCPI.SENSE(Ch).OFFSET.PORT(Pt).FREQUENCY.STOP

Equivalent key

Sweep Setup > Frequency Offset > Port n > Offset

- The basic frequency range is set by using "SCPI.SENSE(Ch).FREQUENCY.START" and "SCPI.SENSE(Ch).FREQUENCY.STOP".

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:OFFSet
<numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:OFFSet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:OFFS 1E9"
20 OUTPUT 717;":SENS1:OFFS:PORT2:OFFS?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.STARt

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.STARt = *Value**Value* = SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.STARt

Description

This command sets/gets a frequency start value for the selected port (*Pt*) when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a frequency start value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Start As Double
SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.STARt = 100E6
Start = SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.STARt
```

Related objects

```
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.MULTIplier
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STOP
```

E5071C

Equivalent key

Sweep Setup > Frequency Offset > Port n > Start

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STARt
<numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:STAR 300000"  
20 OUTPUT 717;":SENS1:OFFS:PORT2:STAR?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STOP = *Value**Value* = SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Description

This command sets/gets the frequency stop value for the selected port (*Pt*) when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Setting of a frequency stop value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	Maximum frequency in the range
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Stop As Double
SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.STOP = 100E6
Stop = SCPI.SENSE(1).OFFSet.PORT(1).FREQuency.STOP
```

Related objects

```
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.MULTIplier
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.START
```

E5071C

Equivalent key

Sweep Setup > **Frequency Offset** > **Port n** > **Stop**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STOP  
<numeric>
```

```
:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:STOP 1e+009"  
20 OUTPUT 717;":SENS1:OFFS:PORT2:STOP?"  
30 ENTER 717;A
```


SCPI.SENSE(*Ch*).OFFSet.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).OFFSet.STATe = *Status**Status* = SCPI.SENSE(*Ch*).OFFSet.STATe

Description

This command turns ON/OFF or returns the status of the frequency offset, for the selected channel (*Ch*).

- When the frequency offset feature is ON, different frequencies can be used for measurement for each port. Frequencies set for each port are used regardless of whether the port is on the stimulus side or response side.

Variable

Parameter	<i>Status</i>
Description	On/off of the frequency offset
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> True or ON: Turns on the frequency offset. False or OFF: Turns off the frequency offset.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSE(1).OFFSet.STATe = False
State = SCPI.SENSE(1).OFFSet.STATe
```

Equivalent key

Sweep Setup > Frequency Offset > Frequency Offset

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet[:STATe] {ON|OFF|1|0}
:SENSe{[1]-160}:OFFSet[:STATe]?
```

Query response

E5071C

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS ON"  
20 OUTPUT 717;":SENS1:OFFS?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).ROSCillator.SOURce

Object type

Property (**Read-only**)

Syntax

Param = SCPI.SENSE(*Ch*).ROSCillator.SOURce

Description

This command reads whether the external reference signal is inputted to the Ref In connector on the rear panel.

Variable

Parameter	<i>Param</i>
Description	Whether the external reference signal is inputted or not.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "INTernal": The external reference signal is not inputted. • "EXTernal": The external reference signal is inputted.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Ref As String
Ref = SCPI.SENSE(1).ROSCillator.SOURce
```

Equivalent key

Displayed on the instrument status bar (at the bottom of the LCD display).

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:ROSCillator:SOURce?
```

Query response

```
{INTernal|EXTernal}<newline><^END>
```

	Description
--	--------------------

E5071C

INTernal	The external reference signal is not inputted.
EXTernal	The external reference signal is inputted.

Example of use

```
10 OUTPUT 717;":SENS1:ROSC:SOUR?"  
20 ENTER 717;A$
```

SCPI.SENSE(*Ch*).SEGMent.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).SEGMent.DATA = *Data**Data* = SCPI.SENSE(*Ch*).SEGMent.DATA

Description

This command creates the segment sweep table of selected channel (*Ch*).

Variable

Parameter	Data
Description	<p>Indicates the array data arranged in the following order (for the segment sweep table); where N is the number of segments (specified with <segm>) and n is an integer between 1 and N.</p> <p><i>Data</i> = {<buf>,<stim>,<ifbw>,<pow>,,<swp>,<time>,<segm>, <star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<swp 1>,<time 1>,... , <star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<swp n>,<time n>,... , <star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<swp N>,<time N>} </p> <p>Each parameter in the above array data is detailed below:</p> <ul style="list-style-type: none"> • <buf>: Always specify 5 or 6. You have to specify 6 if you need to set up the sweep mode setting for each segment. • <stim>: Stimulus setting mode 0: Specifies with start/stop values 1: Specifies with center/span values • <ifbw>: ON/OFF of the IF bandwidth setting for each segment 0: OFF, 1: ON • <pow>: ON/OFF of the power setting for each segment 0: OFF, 1: ON • : ON/OFF of the sweep delay time setting for

each segment

0: OFF, 1: ON

- <swp>: ON/OFF of the sweep mode setting for each segment

0: OFF, 1: ON

Not necessary when <buf> is 5.

- <time>: ON/OFF of the sweep time setting for each segment

0: OFF, 1: ON

- <segm>: Number of segments

Specify an integer ranging 1 to 201.

- <star n>: Start value/center value of the n-th segment

- <stop n>: Stop value/span value of the n-th segment

- <nop n>: Number of measurement points of the n-th segment

- <ifbw n>: IF bandwidth of the n-th segment Not necessary when the IF bandwidth setting for each segment is OFF (<ifbw>:0).

- <pow n>: Power of the n-th segment Not necessary when the power setting for each segment is OFF (<pow>:0).

- <del n>: Sweep delay time of the n-th segment Not necessary when the sweep delay time setting for each segment is OFF (:0).

- <swp n> : Sweep mode of the n-th segment

0: Stepped mode

1: Swept mode

2: Stepped mode

3: Swept mode

Not necessary when <buf> is 5 or the sweep mode setting for each segment is OFF (:0). 2 or 3 can be acceptable for compatibility with E5071C.

- <time n>

- Sweep time of the n-th segment is not necessary when the sweep time setting for each segment is OFF (<time>:0).

Data type	Variant type (Variant)
Note	<p>If the necessary amount of array data for the specified number of segments is not available while setting the segment sweep table, an error occurs when executed and the object is ignored.</p> <p>For <stim>, <ifbw>, <pow>, , <swp>, and <time>, if the specified value is not the allowable integer, an error occurs when executed.</p> <p>For <star n>, <stop n>, <nop n>, <ifbw n>, <pow n>, <del n>, and <time n> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```

Dim SegmData As Variant
SCPI.SENSE(1).SEGMENT.DATA = Array(5,0,0,1,0,0,2, _
100E6,1E9,31,0,2E9,3E9,51,-10)
SegmData = SCPI.SENSE(1).SEGMENT.DATA

Dim SegmData(14) As Variant
Dim Ref As Variant
SegmData(0) = 5
SegmData(1) = 0
SegmData(2) = 0
SegmData(3) = 1
SegmData(4) = 0
SegmData(5) = 0
SegmData(6) = 2
SegmData(7) = 100E6
SegmData(8) = 1E9
SegmData(9) = 31
SegmData(10) = 0
SegmData(11) = 2E9
SegmData(12) = 3E9
SegmData(13) = 51
SegmData(14) = -10
SCPI.SENSE(1).SEGMENT.DATA = SegmData
Ref = SCPI.SENSE(1).SEGMENT.DATA

```

Related objects

SCPI.SENSE(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > Edit Segment Table

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SEGMENT:DATA
5,<mode>,<ifbw>,<pow>,<del>,<time>,<segm>,<star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<time 1>,<?E/span> ,
<star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<time n>,<?E/span> ,
<star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<time N>
:SENSe{[1]-160}:SEGMENT:DATA
6,<mode>,<ifbw>,<pow>,<del>,<swp>,<time>,<segm>,<star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<swp 1>,<time 1>,<?E/span> ,
<star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<swp n>,<time n>,<?E/span> ,
<star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<swp N>,<time N>
:SENSe{[1]-160}:SEGMENT:DATA?
```

Where N is the number of segments (specified with <segm>) and n is an integer between 1 and N.

Query response

When the sweep mode setting for each segment is OFF:

```
5,{mode},{ifbw},{pow},{del},{time},{segm},
{star 1},{stop 1},{nop 1},{pow 1},{del 1},{time 1},?E/span> ,
{star n},{stop n},{nop n},{pow n},{del n},{time n},?E/span> ,
{star N},{stop N},{nop N},{pow N},{del N},{time N}<newline><^END>
```

When the sweep mode setting for each segment is ON:

```
6,{mode},{ifbw},{pow},{del},{swp},{time},{segm},
{star 1},{stop 1},{nop 1},{pow 1},{del 1},{swp 1},{time 1},?E/span> ,
{star n},{stop n},{nop n},{pow n},{del n},{swp n},{time n},?E/span> ,
{star N},{stop N},{nop N},{pow N},{del N},{swp N},{time N}<newline><^END>
```

Example of use

```
10 DIM H(1:3,1:4)
20 OUTPUT 717;":SENS1:SEGM:DATA 5,0,1,0,0,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,";
50 OUTPUT 717;"4E9,6E9,11,70e3"
60 OUTPUT 717;":SENS1:SEGM:DATA?"
70 ENTER 717;A,B,C,D,E,F,G,H(*)
```



```
10 DIM I(1:3,1:5)
20 OUTPUT 717;":SENS1:SEGM:DATA 6,0,1,0,0,1,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,2,";
50 OUTPUT 717;"4E9,6E9,11,70e3,3"
60 OUTPUT 717;":SENS1:SEGM:DATA?"
70 ENTER 717;A,B,C,D,E,F,G,H, I(*)
```

SCPI.SENSE(*Ch*).SEGMent.SWEep.POINts

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SENSE(*Ch*).SEGMent.SWEep.POINts

Description

This command reads the total number of the measurement points of all segments, for the segment sweep table of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Total number of measurement points of all segments
Data type	Long integer type (Long)
Preset Value	2

Examples

Dim SegmPoin As Long
SegmPoin = SCPI.SENSE(1).SEGMent.SWEep.POINts

Related objects

SCPI.SENSE(*Ch*).SEGMent.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SEGMent:SWEep:POINts?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SENS1:SEGM:SWE:POIN?"
20 ENTER 717;A

SCPI.SENSE(*Ch*).SEGMent.SWEep.TIME.DATA**Object type**Property (**Read-only**)**Syntax***Value* = SCPI.SENSE(*Ch*).SEGMent.SWEep.TIME.DATA**Description**

This command reads the total sweep time (including sweep delay time) of all segments, for the segment sweep table of the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Total sweep time of all segments
Data type	Double precision floating point type (Double)

Examples

```
Dim SegmTime As Double
SegmTime = SCPI.SENSE(1).SEGMent.SWEep.TIME.DATA
```

Related objectsSCPI.SENSE(*Ch*).SEGMent.DATA**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SENSe{[1]-160}:SEGMent:SWEep:TIME[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:TIME?"
20 ENTER 717;A
```

SCPI.SENSE(*Ch*).SEGMENT.LIST.CONTROL.DATA

Object type

Property (**Read-Write**)

Syntax

This option is only available in revision A.9.60 and above:

SCPI.SENSE(*Ch*).SEGMENT.LIST.CONTROL.DATA = *Data*

Data = SCPI.SENSE(*Ch*).SEGMENT.LIST.CONTROL.DATA

Description

This command sets/gets the state of each segment in the segment sweep table of selected channel (*Ch*).

This command is only available in revision A.9.60 and above.

Variable

Parameter	<i>Data</i>
Description	Indicates the state array data; where N is the number of segments and n is an integer between 1 and N. <i>Data</i> = {<state1>, ... ,<state n>, ... ,<state N>}
Data type	Variant type (Variant)

Examples

```
Dim SegmContrData As Variant
SCPI.SENSE(1).SEGMENT.LIST.CONTROL.DATA = Array(1, 1, 0, 0)
SegmContrData = SCPI.SENSE(1).SEGMENT.LIST.CONTROL.DATA
SCPI.SENSE(1).SEGMENT.LIST.CONTROL.DATA = SegmContrData
```

Related objects

SCPI.SENSE(*Ch*).SWEep.TYPE

Equivalent key

Sweep Setup > Edit Segment Table

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SEGMENT:LIST:CONTROL:DATA <state 1>, <state n>,
<state N>

:SENSe{[1]-160}:SEGMENT:LIST:CONTROL:DATA?

Where N is the number of segments and n is an integer between 1 and N.

Example of use

```
10 DIM H(1:3)
20 OUTPUT 717;":SENS1:SEGM:DATA 5,0,1,0,0,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,";
50 OUTPUT 717;"4E9,6E9,11,70e3"
60 OUTPUT 717;":SENS1:SEGM:LIST:CONT:DATA 1, 0, 0"
70 OUTPUT 717;":SENS1:SEGM:LIST:CONT:DATA?"
80 ENTER 717;H(*)
```

SCPI.SENSE(Ch).SEGMENT.LIST.CONTROL.STATE**Object type**Property (**Read-Write**)**Syntax**

This option is only available in revision A.9.60 and above:

SCPI.SENSE(*Ch*).SEGMENT.LIST.CONTROL.STATE = *Status*

Status= SCPI.SENSE(*Ch*).SEGMENT.LIST.CONTROL.STATE

Description

This command turns ON or OFF the individual segment state control in the segment sweep table.

This command is only available in revision A.9.60 and above.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the individual segment state control
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • ON or 1: Turns ON the individual segment state control. • OFF or 0: Turns OFF the individual segment state control.
Preset value	OFF or 0

Examples

Dim SegmState As Boolean

SCPI.SENSE(1).SEGMENT.LIST.CONTROL.STATE = True

SegmState= SCPI.SENSE(1).SEGMENT.LIST.CONTROL.STATE

Related objects

SCPI.SENSE(Ch).SEGMENT.LIST.CONTROL.DATA

Equivalent key

Sweep Setup > Edit Segment Table > List Control

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SEGMENT:LIST:CONTROL:STATE: {ON|OFF|1|0}  
:SENSe{[1]-160}:SEGMENT:LIST:CONTROL:STATE?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SEGMENT:LIST:CONTROL:STATE ON"  
20 OUTPUT 717;":SENS1:SEGMENT:LIST:CONTROL:STATE?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).SWEep.ASPurious**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).SWEep.ASPurious = *Status**Status* = SCPI.SENSE(*Ch*).SWEep.ASPurious**Description**

This command does not performs any operation. This command is provided because of its command compatibility with E5070B/E5071B.

Variable

Parameter	<i>Status</i>
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Select from the following. <ul style="list-style-type: none"> • True or ON: No meaning • False or OFF: No meaning
Preset value	True or OFF

Examples

```
Dim ASpurious As Boolean
SCPI.SENSE(1).SWEep.ASPurious = False
ASpurious = SCPI.SENSE(1).SWEep.ASPurious
```

Equivalent key

None

Equivalent SCPI command**Syntax**

:SENSe{[1]-160}:SWEep:ASPurious {ON|OFF|1|0}

:SENSe{[1]-160}:SWEep:ASPurious?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;".SENS1:SWE:ASP OFF"  
20 OUTPUT 717;".SENS1:SWE:ASP?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).SWEep.DELay

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).SWEep.DELay = *Value**Value* = SCPI.SENSE(*Ch*).SWEep.DELay

Description

This command sets/gets the sweep delay time of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Sweep delay time
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0
Unit	s (second)
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SweDel As Double
SCPI.SENSE(1).SWEep.DELay = 0.05
SweDel = SCPI.SENSE(1).SWEep.DELay
```

Equivalent key

Sweep Setup > Sweep Delay

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SWEep:DELay <numeric>  
:SENSe{[1]-160}:SWEep:DELay?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SWE:DEL 0.05"  
20 OUTPUT 717;":SENS1:SWE:DEL?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).SWEep.GENeration

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).SWEep.GENeration = *Param**Param* = SCPI.SENSE(Ch).SWEep.GENeration

Description

This command sets/gets the sweep mode of selected channel (*Ch*).

- When the sweep type is the power sweep (POW specified with the SCPI.SENSE(Ch).SWEep.TYPE object), when the power calibration is on (ON specified with the SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.STATe object), or the power slope value is other than 0 and the power slope function is on (ON specified with the SCPI.SOURce(Ch).POWER.LEVel.SLOPe.STATe object), if you execute this object to try to set the sweep mode to the swept mode, an error occurs and the sweep mode is automatically set to the step mode.

Variable

Parameter	<i>Param</i>
Description	Sweep mode
Data type	Character string type (String)
Range	<p>Select from either of the following:</p> <ul style="list-style-type: none"> "STEPped": Sets the sweep mode to the stepped mode. "ANALog": Sets the sweep mode to the swept mode. "FSTepped": Sets the sweep mode to the stepped mode. This parameter is provided because of its command compatibility with E5070B/E5071B. "FANalog": Sets the sweep mode to the swept mode. This parameter is provided because of its command compatibility with E5070B/E5071B.

Preset value	"STEPped"
---------------------	-----------

Examples

```
Dim SwptMode As String
SCPI.SENSE(1).SWEep.GENeration = "anal"
SwptMode = SCPI.SENSE(1).SWEep.GENeration
```

Related objects

```
SCPI.SENSE(Ch).SWEep.TYPE
SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.STATe
SCPI.SOURce(Ch).POWER.LEVel.SLOPe.STATe
```

Equivalent key

Sweep Setup > **Sweep Mode** > **Stepped|Swept**

Equivalent SCPI command**Syntax**

```
:SENSe{[1]-160}:SWEep:GENeration
{STEPped|ANALog|FSTepped|FANalog}
:SENSe{[1]-160}:SWEep:GENeration?
```

Query response

```
{STEP|ANAL|FST|FAN}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SWE:GEN ANAL"
20 OUTPUT 717;":SENS1:SWE:GEN?"
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).SWEp.POINTs**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).SWEp.POINTs = *Value**Value* = SCPI.SENSE(*Ch*).SWEp.POINTs**Description**

This command sets/gets the number of measurement points of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Number of measurement points
Data type	Long integer type (Long)
Range	2 to 20001
Preset value	201
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Nop As Long
SCPI.SENSE(1).SWEp.POINTs = 801
Nop = SCPI.SENSE(1).SWEp.POINTs
```

Equivalent key**Sweep Setup > Points****Equivalent SCPI command****Syntax**

:SENSe{[1]-160}:SWEp:POINTs <numeric>

:SENSe{[1]-160}:SWEp:POINTs?

Query response

2530

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:POIN 801"  
20 OUTPUT 717;":SENS1:SWE:POIN?"  
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).SWEep.TIME.AUTO**Object type**Property (**Read-Write**)**Syntax**SCPI.SENSE(*Ch*).SWEep.TIME.AUTO = *Status**Status* = SCPI.SENSE(*Ch*).SWEep.TIME.AUTO**Description**

This command sets/gets whether to automatically set the sweep time of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the auto setting of the sweep time
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the auto setting. • False or OFF: Turns OFF the auto setting.
Preset value	True or OFF

Examples

```
Dim SweAuto As Boolean
SCPI.SENSE(1).SWEep.TIME.AUTO = False
SweAuto = SCPI.SENSE(1).SWEep.TIME.AUTO
```

Related objectsSCPI.SENSE(*Ch*).SWEep.TIME.DATA**Equivalent key****Sweep Setup > Sweep Time**

- When performing the operation from the front panel, the auto setting of the sweep time is turned ON by setting the sweep time to 0 s.

Equivalent SCPI command**Syntax**

:SENSE{[1]-160}:SWEep:TIME:AUTO {ON|OFF|1|0}

:SENSe{[1]-160}:SWEep:TIME:AUTO?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TIME:AUTO ON"  
20 OUTPUT 717;":SENS1:SWE:TIME:AUTO?"  
30 ENTER 717;A
```

SCPI.SENSE(Ch).SWEep.TIME.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(Ch).SWEep.TIME.DATA = *Value**Value* = SCPI.SENSE(Ch).SWEep.TIME.DATA

Description

This command sets/gets the sweep time of selected channel (*Ch*).**NOTE**

Before using this object to set the sweep time, turns OFF the auto setting of the sweep time (specify False with the SCPI.SENSE(Ch).SWEep.TIME.AUTO object).

NOTE

When Port IFBW is turned ON, this command returns the sweep time for Port 1.

Variable

Parameter	<i>Value</i>
Description	Sweep time
Data type	Double precision floating point type (Double)
Range	Varies depending on the measurement conditions
Preset value	Varies depending on the measurement conditions
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```

Dim SweTime As Double
SCPI.SENSE(1).SWEep.TIME.AUTO = False
SCPI.SENSE(1).SWEep.TIME.DATA = 1.5
SweTime = SCPI.SENSE(1).SWEep.TIME.DATA

```

Related objects

SCPI.SENSE(Ch).SWEep.TIME.AUTO

Equivalent key

Sweep Setup > **Sweep Time**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SWEep:TIME[:DATA] <numeric>
:SENSe{[1]-160}:SWEep:TIME[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TIME 1.5"
20 OUTPUT 717;":SENS1:SWE:TIME?"
30 ENTER 717;A
```

SCPI.SENSE(*Ch*).SWEep.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).SWEep.TYPE = *Param**Param* = SCPI.SENSE(*Ch*).SWEep.TYPE

Description

This command sets/gets the sweep type of selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	Sweep type
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "LINear": Sets the sweep type to the linear sweep. • "LOGarithmic": Sets the sweep type to the log sweep. • "SEGment": Sets the sweep type to the segment sweep. • "POWER": Sets the sweep type to the power sweep.
Preset value	"LINear"

Examples

```
Dim SweType As String
SCPI.SENSE(1).SWEep.TYPE = "segm"
SweType = SCPI.SENSE(1).SWEep.TYPE
```

Equivalent key

Sweep Setup > Sweep Type > Lin Freq|Log Freq|Segment

Equivalent SCPI command

Syntax

:SENSE{[1]-160}:SWEep:TYPE {LINear|LOGarithmic|SEGment|POWER}

:SENSe{[1]-160}:SWEep:TYPE?

Query response

{LIN|LOG|SEGM|POW}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TYPE SEGM"  
20 OUTPUT 717;":SENS1:SWE:TYPE?"  
30 ENTER 717;A$
```

SCPI.SENSE(*Ch*).VOLTage(*Auxpt*).DC.RANGE.UPPer

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE(*Ch*).VOLTage(*Auxpt*).DC.RANGE.UPPer=ParamParam=SCPI.SENSE(*Ch*).VOLTage(*Auxpt*).DC.RANGE.UPPer

Description

This command sets/gets the DC voltage range of selected channel (*Ch*).

Variable

Parameter	<i>Param</i>
Description	AUX DC Input range
Data type	Double precision floating point type (Double)
Unit	Volts (V)
Range	1 or 10
Preset value	10

Parameter	<i>Auxpt</i>
Description	AUX port
Data type	Long Integer Type (Long)
Range	1 or 2
Preset value	None

Examples

Dim DcRange As variant

SCPI.SENSE(1).VOLTage(2).DC.RANGE.UPPer=1

DcRange = SCPI.SENSE(1).VOLTage(1).DC.RANGE.UPPer

Equivalent key

Meas > **Aux Input** (1 or 2) > **Range**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:VOLTage{[1]|2}:DC.RANGe:UPPer  
:SENSe{[1]-160}:VOLTage{[1]|2}:DC.RANGe:UPPer?
```

Query response

```
{1|10}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:VOLT1:DC.RANG:UPP 10"  
20 OUTPUT 717;":SENS1:VOLT1:DC.RANG:UPP?"  
30 ENTER 717;A
```

SCPI.SENSE.MULTiplexer(*Id*).INCount**Object type**Property (**Read-only**)**Syntax***Value* = SCPI.SENSE.MULTiplexer(*Id*).INCount**Description**

This command reads the number of input ports of ID 1 to 2 (*Id*), from the E5091A.

Examples

```
Dim INCount As Long
INCount = SCPI.SENSE(1).MULTiplexer(1).INCount
```

Related objects

```
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT1
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT2
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT3
SCPI.SENSE(Ch).MULTiplexer(Id).TSET9.PORT4
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SENSe:MULTiplexer{[1]|2}:INCount?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:MULT1:INC?"
20 ENTER 717;A
```


SCPI.SENSE.MULTiplexer(*Id*).NAME

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSE.MULTiplexer(*Id*).NAME = *Param**Param* = SCPI.SENSE.MULTiplexer(*Id*).NAME

Description

This command sets/gets the configuration of the multiport test set.

- The upper case and lower case are not distinguished.

Variable

Parameter	<i>Param</i>
Description	The name of the multiport test set
Data type	Character string type (String)
Range	<p>Select from either of the following:</p> <ul style="list-style-type: none"> • "E5091_9": Selects the 9-port option of the E5091A (E5091A-009). • "E5091_13": Selects the 13-port configuration of the E5091A-016. • "E5091_16": Selects the 16-port configuration of the E5091A-016. • "E5092_13": Selects the 13-port configuration of the E5092A (E5092A-020). Equivalent to E5091_13 of the E5091A-016. • "E5092_16": Selects the 16-port configuration of the E5092A (E5092A-020). Equivalent to E5091_16 of the E5091A-016. • "E5092_22": Selects the 22-port configuration of the E5092A (E5092A-020). • "E5092_28": Selects the switching independently in the E5092A (E5092A-020). • "E5092_X10": Selects the 10-port full crossbar configuration of the E5092A (E5092A-020).
Preset	E5091_9

Value

For information on the variable (*Id*), see Id.

Examples

```
Dim MultName As String
SCPI.SENSESe.MULTiplexer(1).NAME = "E5091_16"
MultName = SCPI.SENSESe.MULTiplexer(1).NAME
```

Related objects

```
SCPI.SENSESe.MULTiplexer.CATalog
SCPI.SENSESe.MULTiplexer(Id).INCount
```

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1|Test Set 2** > **Select Test Set** > **E5091_9|E5091_13|E5091_16|E5092_13|E5092_16|E5092_22|E5092_28|E5092_X10**

Equivalent SCPI command

Syntax

```
:SENSe:MULTiplexer{[1]|2}:NAME
{E5091_9|E5091_13|E5091_16|E5092_13|E5092_16|E5092_22|E5092_28|E5092_X10}
:SENSe:MULTiplexer{[1]|2}:NAME?
```

Query response

```
{E5091_9|E5091_13|E5091_16|E5092_13|E5092_16|E5092_22|E5092_28|E5092_X10}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:MULT1:NAME E5091_9"
20 OUTPUT 717;":SENS:MULT1:NAME?"
30 ENTER 717;A$
```

SCPI.SENSESe.MULTiplexer.CATalog**Object type**

Property (**Read-only**)

Syntax

Param = SCPI.SENSESe.MULTiplexer.CATalog

Description

This command reads the name of the multiport test set.

Examples

```
Dim MultCat As String
MultCat = SCPI.SENSESe.MULTiplexer.CATalog
```

Related objects

SCPI.SENSESe.MULTiplexer(Id).NAME

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SENSe:MULTiplexer{[1]|2}:CATalog?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT:CAT?"
20 ENTER 717;A$
```

SERVICE**SCPI.SERVICE.ADJust.SENSE(Ch).SWEp.RF.RANGe.PORT(Pt).AUTO****Object type**Property (**Read-Write**)**Syntax***Value* =

SCPI.SERVICE.ADJust.SENSE(Ch).SWEp.RF.RANGe.PORT(Pt).AUTO

SCPI.SERVICE.ADJust.SENSE(Ch).SWEp.RF.RANGe.PORT(Pt).AUTO =
*Value***Description**

This command sets/gets Receiver RF range mode ON/OFF status. This command is used in the 8 term Calibration using External PC.

Variable

Parameter	<i>Value</i>
Description	ON/OFF of Receiver RF Range Mode
Data type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	ON
Note	The setting for this command is set at ON by the preset, but it is not saved in the state file. Even if save command is executed after calibration with RF range Auto mode as OFF, it is not possible to measure the value correctly only by recalling. To recall the state, the status of RF range Auto should be set as OFF.

Examples

Dim Val As Boolean

SCPI.SERVICE.ADJust.SENSE(1).SWEp.RF.RANGe.PORT(2).AUTO = True

Val = SCPI.SERVICE.ADJust.SENSE(1).SWEp.RF.RANGe.PORT(2).AUTO

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SERVice:ADJust:SENSe([1]-160):SWEep:RF:RANGe:PORT([1]-4):AUTO
{ON|OFF|1|0}
```

```
:SERVice:ADJust:SENSe([1]-160):SWEep:RF:RANGe:PORT([1]-4):AUTO?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SERV:ADJ:SENS1:SWE:RF:RANG:PORT2:AUTO ON"
20 OUTPUT 717;":SERV:ADJ:SENS2:SWE:RF:RANG:PORT3:AUTO?"
30 ENTER 717;A
```

SCPI.SERVICE.ADJust.VIRTual.BRIDge.STATe**Object type**Property (**Read-Write**)**Syntax***Value* = SCPI.SERVICE.ADJust.VIRTual.BRIDge.STATeSCPI.SERVICE.ADJust.VIRTual.BRIDge.STATe = *Value***Description**

This command sets/gets virtual bridge correction ON/OFF status. This command is used in the 8 term Calibration using External PC.

- This command does not works in options 2D5, 4D5, 2K5 and 4K5.

Variable

Parameter	<i>Value</i>
Description	ON/OFF of Virtual bridge correction
Data type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	ON
Note	The setting for this command is set at ON by the preset, but it is not saved in the state file. Even if save command is executed after calibration with virtual bridge OFF, it is not possible to measure the value correctly only by recalling. To recall the state the status of virtual bridge should be set as OFF.

Examples

```
Dim Val As Boolean
SCPI.SERVICE.ADJust.VIRTual.BRIDge.STATe = True
Val = SCPI.SERVICE.ADJust.SERVICE.VIRTual.BRIDge.STATe
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SERVice:ADJust:VIRTual:BRIDge[:STATe]{ON|OFF|1|0}  
:SERVice:ADJust:VIRTual:BRIDge[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SERV:ADJ:VIRT:BRID ON"  
20 OUTPUT 717;":SERV:ADJ:VIRT:BRID?"  
30 ENTER 717;A
```

SCPI.SERVICE.CHANnel(*Ch*).TRACe.ACTive

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.CHANnel(*Ch*).TRACe.ACTive

Description

This command reads the active trace number of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Active trace number
Data type	Long integer type (Long)

Examples

```
Dim ActTrac As Long
ActTrac = SCPI.SERVICE.CHANnel(1).TRACe.ACTive
```

Related objects

SCPI.CALCulate(*Ch*).PARAmeter(*Tr*).SElect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVICE:CHANnel{[1]-36}:TRACe:ACTive?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:TRAC:ACT?"
20 ENTER 717;A
```


SCPI.SERVICE.CHANNEL.ACTive**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.CHANNEL.ACTive

Description

This command reads the active channel number.

Variable

Parameter	<i>Value</i>
Description	Active channel number
Data type	Long integer type (Long)

Examples

```
Dim ActChan As Long  
ActChan = SCPI.SERVICE.CHANNEL.ACTive
```

Related objects

SCPI.DISPLAY.WINDOW(Ch).ACTivate

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SERVICE:CHANNEL:ACTive?

Query response

{numeric}<newline><^END>

SCPI.SERVICE.CHANNEL.COUNT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.CHANNEL.COUNT

Description

This reads the upper limit of the number of channels of the E5071C.

Variable

Parameter	<i>Value</i>
Description	Upper limit of the number of channels.
Data type	Long integer type (Long)

Examples

```
Dim MaxChan As Long
MaxChan = SCPI.SERVICE.CHANNEL.COUNT
```

Equivalent key

System > Misc Setup > Channel/Trace Setup

Equivalent SCPI command

Syntax

:SERVICE:CHANNEL:COUNT?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN:COUN?"
20 ENTER 717;A
```

SCPI.SERVICE.CHANNEL.TRACE.COUNT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.CHANNEL.TRACE.COUNT

Description

This command reads the upper limit of the number of traces per channel.

Variable

Parameter	<i>Value</i>
Description	Upper limit of the number of traces.
Data type	Long integer type (Long)

Examples

```
Dim MaxTrac As Long  
MaxTrac = SCPI.SERVICE.CHANNEL.TRACE.COUNT
```

Equivalent key

System > Misc Setup > Channel/Trace Setup

Equivalent SCPI command

Syntax

`:SERVICE:CHANNEL:TRACE:COUNt?`

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN:TRAC:COUN?"  
20 ENTER 717;A
```

E5071C

SCPI.SERVICE.LOGGING.CLEAR

Object type

Method (**Write-only**)

Syntax

SCPI.SERVICE.LOGGING.CLEAR

Description

This command clears the log data of the E5071C. For more information about the log file, refer to Removing Log Data.

Examples

```
Dim ClearLog As Long  
ClearLog = SCPI.SERVICE.LOGGING.CLEAR
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVICE:LOGGING:CLEAR

Example of use

```
10 OUTPUT 717;":SERV:LOGG:CLE"
```

SCPI.SERVICE.PORT.COUNT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.PORT.COUNT

Description

This command reads the number of ports of the E5071C.

Variable

Parameter	<i>Value</i>
Description	Number of ports
Data type	Long integer type (Long)

Examples

```
Dim MaxPort As Long  
MaxPort = SCPI.SERVICE.PORT.COUNT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SERVICE:PORT:COUNT?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:PORT:COUN?"  
20 ENTER 717;A
```

SCPI.SERVICE.POWER.AC.OVERload.DETect.STATe**Object type**Method (**Read/Write**)**Syntax***Value* = SCPI.SERVICE.POWER.AC.OVERload.DETect.STATeSCPI.SERVICE.POWER:AC.OVERload.DETect.STATe= *Value***Description**

This command activates/deactivates the Overload Detection/Power Trip function. For more information, refer to Overload Detection and Power Trip Function.

- This function is valid only for E5071C and is not valid for E5070B/E5071B.

Variable

Parameter	<i>Status</i>
Description	Turns ON/OFF Overload detection/Power Trip function
Data type	Boolean type (Boolean)
Range	Select from the following: <ul style="list-style-type: none"> • True or ON: Turns ON the Overload detect • False or OFF: Turns OFF the Overload detect
Preset value	True or ON

Examples

Dim Value As Bool
Value=1

SCPI.SERVICE.POWER.AC.OVERload.DETect.STATe=Value

Value=SCPI.SERVICE.POWER.AC.OVERload.DETect.STATe

Equivalent key**System > Service Menu > Overload Detect****Equivalent SCPI command****Syntax**

```
:SERVice:POWer:AC:OVERload:DETECT:STATe {ON|OFF}  
:SERVice:POWer:AC:OVERload:DETECT:STATe?
```

Query response

```
{1|0}<newline><^END>
```

E5071C

SCPI.SERVICE.SREVISION

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.SREVISION

Description

This command reads the system spec version of the E5071C.

Variable

Parameter	<i>Value</i>
Description	1 means applying new system specifications. 0 means applying old system specifications.
Data type	Long integer type (Long)

Examples

```
Dim SystemRevision As Long  
SystemRevision = SCPI.SERVICE.PORT.COUNT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVICE:SREVISION?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SREV?"  
20 ENTER 717;A
```


SCPI.SERVICE.SWEep.FREQency.MAXimum**Object type**Property (**Read-only**)**Syntax***Value* = SCPI.SERVICE.SWEep.FREQency.MAXimum**Description**

This command reads the upper limit of measurement frequency of E5071C.

Variable

Parameter	<i>Value</i>
Description	Upper limit of measurement frequency
Data type	Double precision floating point type (Double)

Examples

```
Dim MaxFreq As Double  
MaxFreq = SCPI.SERVICE.SWEep.FREQency.MAXimum
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:SERVICE:SWEep:FREQency:MAXimum?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SERV:SWE:FREQ:MAX?"  
20 ENTER 717;A
```

E5071C

SCPI.SERVICE.SWEEp.FREQency.MINimum

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.SWEEp.FREQency.MINimum

Description

This command reads the lower limit of measurement frequency of E5071C.

Variable

Parameter	<i>Value</i>
Description	Lower limit of measurement frequency
Data type	Double precision floating point type (Double)

Examples

Dim MinFreq As Double
MinFreq = SCPI.SERVICE.SWEEp.FREQency.MINimum

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVICE:SWEEp:FREQency:MINimum?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":SERV:SWE:FREQ:MIN?"
20 ENTER 717;A

SCPI.SERVICE.SWEep.POINTs**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.SERVICE.SWEep.POINTs

Description

This command reads the upper limit of the number of measurement points at the current configuration.

Variable

Parameter	<i>Value</i>
Description	Upper limit of the number of points
Data type	Long integer type (Long)

Examples

```
Dim MaxPoin As Long
MaxPoin = SCPI.SERVICE.SWEep.POINTs
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SERVICE:SWEep:POINTs?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:POIN?"
20 ENTER 717;A
```

SOURCE**SCPI.SOURce(Ch).POWer.ATTenuation.AUTO**

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.ATTenuation.AUTO = *Status**Status* = SSCPI.SOURce(Ch).POWer.ATTenuation.AUTO

Description

This command turns on/off the Auto Power Range set function for the channels (Ch).

When the Auto Power Range set function is ON, the minimum value of step attenuation is selected according to the available source power range as described below:

Source Power Range	Step Attenuation
-25 ~ +10 dBm	0 dB
-35 ~ -25 dBm	10 dB
-45 ~ -35 dBm	20 dB
-55 ~ -45 dBm	30 dB
-65 ~ -55 dBm	40 dB
-75 ~ -65 dBm	50 dB
-85 ~ -75 dBm	60 dB

The power sweep setting is set according to the start power level and maximum value of stop power level is set according to step attenuation. For more information on power attenuation, refer to Setting Power Level with Auto Power Range.

- This command works only with the E5071C options 2D5, 4D5, 2K5 and 4K5 having a mechanical source attenuator. This command is neither required nor available in other E5071C options which are having electrical source attenuator and is provided to maintain backward compatibility.

Variable

Parameter	<i>Status</i>
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turn on the Auto Power Range • False or OFF: Turn off the Auto Power Range
Preset value	True or ON

Related objects

SCPI.SOURce(Ch).POWER.ATTenuation.DATA

Equivalent key**Sweep Setup > Power > Auto Range****Equivalent SCPI command****Syntax**

```
:SOURce{[1]-160}:POWER:ATTenuation:AUTO {ON|OFF|1|0}
:SOURce{[1]-160}:POWER:ATTenuation:AUTO?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:ATT:AUTO ON"
20 OUTPUT 717;":SOUR1:POW:ATT:AUTO?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.ATTenuation.DATA = *Value**Value* = SCPI.SOURce(Ch).POWer.ATTenuation.DATA

Description

This command sets/gets the power range for the channel (Ch).

The power range is selected depending on the setting of the attenuator. The relationship between the attenuator value and the power range is available below:

Source Power Range	Step Attenuation
-25 ~ +10 dBm	0 dB
-35 ~ 0 dBm	10 dB
-45 ~ -10 dBm	20 dB
-55 ~ -20 dBm	30 dB
-65 ~ -30 dBm	40 dB
-75 ~ -40 dBm	50 dB
-85 ~ -50 dBm	60 dB

When Auto Power Range set function is effective, this command is ignored and an attenuator level and a power range are selected automatically according to a maximum frequency and a maximum setting power. For more information on power attenuation, refer to Setting Power Level with Auto Power Range.

- This command works with E5070B/E5071B and ENA options 2D5, 4D5, 2K5 and 4K5 .

Variable

Parameter	<i>Value</i>
Description	Dummy Parameter

Data type	Long integer type (Long)
Range	0 to 60
Preset value	0
Unit	dB
Resolution	10

Examples

```
Dim Att As Long
SCPI.SOURce(1).POWer.ATTenuation.DATA = 10
Att = SCPI.SOURce(1).POWer.ATTenuation.DATA
```

Related objects

```
SCPI.SOURce(Ch).POWer.ATTenuation.AUTO
```

Equivalent key

Sweep Setup > Power > Power Ranges

Equivalent SCPI command**Syntax**

```
:SOURce{[1]-160}:POWer:ATTenuation[:DATA] <numeric>
:SOURce{[1]-160}:POWer:ATTenuation[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:ATT.DATA 20"
20 OUTPUT 717;":SOUR1:POW:ATT.DATA?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.CENTer = *Value**Value* = SCPI.SOURce(Ch).POWer.CENTer

Description

This command sets/gets the center value of the sweep range for the power sweep, for channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Center value
Data type	Double precision floating point type (Double)
Range	<p>Varies depending on the power range:</p> <p>9 kHz to 5 GHz -55 to 10 dBm</p> <p>5 to 6 GHz -55 to 9 dBm</p> <p>6 to 7 GHz -55 to 8 dBm</p> <p>7 to 8.5 GHz -55 to 7 dBm</p>
Preset value	-7.5
Unit	dBm
Resolution	0.05 or 0.025
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim Pcntr As Double

SCPI.SOURce(1).POWer.CENTer = 0

Pcntr = SCPI.SOURce(1).POWer.CENTer

Related objects

SCPI.SENSE(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.SPAN

Equivalent key

Center

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:CENTer <numeric>

:SOURce{[1]-160}:POWer:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:CENt 0"
20 OUTPUT 717;":SOUR1:POW:CENt?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWER.LEVel.IMMediate.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWER.LEVel.IMMediate.AMPLitude = *Value**Value* = SCPI.SOURce(Ch).POWER.LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level of the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Power level
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range.
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim PowLev As Double

SCPI.SOURce(1).POWER.LEVel.IMMediate.AMPLitude = -10

PowLev = SCPI.SOURce(1).POWER.LEVel.IMMediate.AMPLitude

Related objects

SCPI.SOURce(Ch).POWER.ATTenuation.DATA

SCPI.SOURce(Ch).POWER.ATTenuation.AUTO

Equivalent key

Sweep Setup > Power

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer[:LEVel][:IMMediate]
[:AMPLitude] <numeric>
:SOURce{[1]-160}:POWer[:LEVel][:IMMediate]
[:AMPLitude]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW -12.5"
20 OUTPUT 717;":SOUR1:POW?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA = *Value**Value* = SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA**Description**

This command sets/gets the correction value of the power slope feature of channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Correction value of the power slope feature
Data type	Double precision floating point type (Double)
Range	-2 to 2
Preset value	0
Unit	dB/GHz
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SlopLev As Double
SCPI.SOURce(1).POWer.LEVel.SLOPe.DATA = 0.1
SlopLev = SCPI.SOURce(1).POWer.LEVel.SLOPe.DATA
```

Related objects

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe

Equivalent key**Sweep Setup > Power > Slop [xxx dB/GHz]**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer[:LEVel]:SLOPe[:DATA] <numeric>
:SOURce{[1]-160}:POWer[:LEVel]:SLOPe[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SLOP 0.1"
20 OUTPUT 717;":SOUR1:POW:SLOP?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe = *Status**Status* = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe**Description**

This command turns ON/OFF or gets the status of the power slope feature, for the selected channel (*Ch*). This command corrects the attenuation of simple power level proportional to the frequency (attenuation due to cables etc).

Variable

Parameter	<i>Status</i>
Description	On/off of the power slope feature
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the power slop feature. • False or OFF: Turns off the power slop feature.
Preset value	False or OFF

Examples

Dim Slop As Boolean

SCPI.SOURce(1).POWer.LEVel.SLOPe.STATe = True

Slop = SCPI.SOURce(1).POWer.LEVel.SLOPe.STATe

Related objects

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA

SCPI.SENSE(Ch).SWEep.GENeration

Equivalent key**Sweep Setup > Power > Slop [ON/OFF]****Equivalent SCPI command****Syntax**

```
:SOURce{[1]-160}:POWer[:LEVel]:SLOPe:STATe {ON|OFF|1|0}  
:SOURce{[1]-160}:POWer[:LEVel]:SLOPe:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SLOP:STAT ON"  
20 OUTPUT 717;":SOUR1:POW:SLOP:STAT?"  
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWER.PORT(*Pt*).CORRection.COLLection.ACQuire

Object type

Property (**Write-only**)

Syntax

SCPI.SOURce(*Ch*).POWER.PORT(*Pt*).CORRection.COLLection.ACQuire = *Param*

Description

This command sets the power calibration data using the specified power sensor, for the selected port (*Pt*) of the selected channel (*Ch*). When the measurement is complete successfully, the power level error correction is automatically turned ON.

- If the power meter is not connected correctly, an error occurs and this command is ignored.

Variable

Parameter	<i>Param</i>
Description	Selection of the power sensor
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "ASENsor": Specifies power sensor A. • "BSENsor": Specifies power sensor B.

Examples

```
Dim Dmy As Long
SCPI.SOURce(1).POWER.PORT(1).CORRection.COLLection.ACQuire = "asen"
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > Power Calibration > Take Cal Sweep

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWER:PORT{[1]|2|3|4}:CORRection:COLLection
[:ACQuire] {ASENsor|BSENsor}
```


Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL ASEN"  
20 OUTPUT 717;":*OPC?"  
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.AVERAge.COUNT

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.AVERAge.COUNT
= *Value*

Value =

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.AVERAge.COUNT

Description

This command sets/gets the number of power calibration data measurements per measurement point (averaging factor), for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Averaging factor
Data type	Long integer type (Long)
Range	1 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim AvgCnt As Long

SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLection.AVERAge.COUNT = 6

AvgCnt = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLection.AVERAge.COUNT

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.ACQuire

Equivalent key

Cal > Power Calibration > Num of Readings

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:AVERage[:COUNT] <numeric>
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:AVERage[:COUNT]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;"SOUR1:POW:PORT1:CORR:COLL:AVER 6"
20 OUTPUT 717;"SOUR1:POW:PORT1:CORR:COLL:AVER?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.TABLE.LOSS.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.TABLE.LOSS.DAT
A = *Data*

Data =

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.TABLE.LOSS.DAT
A

Description

This command sets/gets the loss compensation table, for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (for the loss compensation table) of 1 + Num (number of set data items)X2. Where n is an integer between 1 and Num.</p> <ul style="list-style-type: none"> <i>Data(0)</i>: The number of data items you want to set. Specify an integer between 0 to 100. When you set the number of data items to 0 (to clear the loss compensation table), you specify only <i>Data(0)</i> as the Data variable. <i>Data(nX2-1)</i>: The frequency of the n-th data item (1 kHz to 500 GHz). <i>Data(nX2)</i>: The loss of the n-th data item (-100 dB to 100 dB). <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Note	<p>If the array data does not contain 1+Num (number of set data items)X2 when setting a loss compensation table, a runtime error occurs. For <i>Data(nX2-1)</i> and <i>Data(n X2)</i> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```

Dim Loss As Variant
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.DATA = Array(2,1e8,0.5,1e9,0.8)
Loss = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.DATA
""Clear Loss Table
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.DATA = Array(0)

Dim Loss(4) As Variant
Dim Ref As Variant
Loss(0) = 2
Loss(1) = 1e8
Loss(2) = 0.5
Loss(3) = 1e9
Loss(4) = 0.8
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.DATA = Loss
Ref = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.DATA
""Clear Loss Table
Dim Loss(0) As Variant
Loss(0) = 0
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.DATA = Loss

```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLeT.TABLe.LOSS.STATe

Equivalent key

Cal > Power Calibration > Loss Compens > Delete | Add | Clear Loss Table

Equivalent SCPI command

Syntax

```

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLeT
:TABLe:LOSS:DATA <numeric 1>,...,<numeric 1+(Nx2)>

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLeT
:TABLe:LOSS:DATA?

```

Query response

```
{numeric 1},...,{numeric 1+(Nx2)}<newline><^END>
```

Example of use

```

10 DIM B(1:2,1:2)
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA 2,";
30 OUTPUT 717;"1E8,0.5,1E9,0.8"
40 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA?"
50 ENTER 717;A,B(*)
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA 0" ! Clear Table

```

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLeT.TABLe.LOSS.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLeT.TABLe.LOSS.STATe = *Status*

Status =

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLeT.TABLe.LOSS.STATe

Description

This command turns ON/OFF or returns the status of the loss compensation, for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	ON/OFF of loss compensation
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the loss compensation. • False or OFF: Turns off the loss compensation.
Preset value	False or OFF

Examples

Dim LComp As Boolean

SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.STATe = True

LComp = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLeT.TABLe.LOSS.STATe

Equivalent key

Cal > Power Calibration > Loss Comp > Compensation

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLeT
:TABLe:LOSS[:STATe] {ON|OFF|1|0}
```

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:TABLe:LOSS[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:TABL:LOSS ON"  
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:TABL:LOSS?"  
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.DATA**Object type**Property (**Read-Write**)**Syntax**SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.DATA = *Data**Data* = SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.DATA**Description**

This command sets/gets the power calibration data array, for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (power calibration data array) of NOP (number of points). Where n is an integer between 1 and NOP.</p> <p><i>Data</i>(<i>n</i>-1)</p> <p>Data at the n-th measurement point</p> <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Note	If the array data does not contain NOP (number of measurement point) when setting a power calibration data array, a runtime error occurs.

Examples

```
Dim FreqData As Variant
SCPI.SENSE(1).SWEep.POINts = 201
FreqData = SCPI.SENSE(1).FREQuency.DATA

Dim CorData As Variant
SCPI.SENSE(1).SWEep.POINts = 201
CorData = SCPI.SOURce(1).POWER.PORT(1).CORRection.DATA
SCPI.SOURce(1).POWER.PORT(2).CORRection.DATA = CorData
```

Related objects

SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.STATe

SCPI.SENSE(Ch).SWEep.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:DATA <numeric
1>, ..,<numeric NOP>
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:DATA?
```

Query response

```
{numeric 1},...,{numeric NOP}<newline><^END>
```

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:DATA?"
30 ENTER 717;A(*)
```

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe = *Status**Status* = SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

Description

This command turns ON/OFF or returns the status of the power level error correction, for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	<i>Status</i>
Description	Turns ON/OFF the power level error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the power level error correction. • False or OFF: Turns off the power level error correction.
Preset value	False or OFF

Examples

Dim PowCorr As Boolean

SCPI.SOURce(1).POWer.PORT(1).CORRection.STATe = True

PowCorr = SCPI.SOURce(1).POWer.PORT(1).CORRection.STATe

Equivalent key

Cal > Power Calibration > Correction

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection[:STATe]
{ON|OFF|1|0}
```

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection[:STATe]?
```

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR ON"  
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR?"  
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWER.PORT(Pt).LEVEL.IMMediate.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWER.PORT(Pt).LEVEL.IMMediate.AMPLitude = *Value**Value* = SCPI.SOURce(Ch).POWER.PORT(Pt).LEVEL.IMMediate.AMPLitude

Description

This command sets/gets the power level, for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Power level at the specified port.
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range.
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PowLev As Double
SCPI.SOURce(1).POWER.PORT.COUPLe = False
SCPI.SOURce(1).POWER.PORT(1).LEVEL.IMMediate.AMPLitude = -12.5
PowLev = SCPI.SOURce(1).POWER.PORT(1).LEVEL.IMMediate.AMPLitude
```

Related objects

SCPI.SOURce(Ch).POWER.PORT.COUPLe

SCPI.SOURce(Ch).POWER.ATTenuation.DATA

Equivalent key

Sweep Setup > Power > Port Power > Port 1 Power | Port 2 Power | Port 3 Power | Port 4 Power

Equivalent SCPI command

Syntax

```
:SOURce{[1]-
160}:POWer:PORT{[1]|2|3|4}[:LEVel][:IMMediate][:AMPLitude]
<numeric>

:SOURce{[1]-
160}:POWer:PORT{[1]|2|3|4}[:LEVel][:IMMediate][:AMPLitude]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1 -12.5"
20 OUTPUT 717;":SOUR1:POW:PORT1?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.PORT.COUPle**Object type**Property (**Read-Write**)**Syntax**SCPI.SOURce(Ch).POWer.PORT.COUPle = *Status**Status* = SCPI.SOURce(Ch).POWer.PORT.COUPle**Description**

This command sets/gets whether to output the same power level for each port of channels 1 to 36 (*Ch*).

- When the power slope feature is ON, the same power level is always output to all ports regardless of this setting because different power levels cannot be output for each port.

Variable

Parameter	<i>Status</i>
Description	Turns ON/OFF the coupling between ports for the power level output
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> True or ON: Outputs the same power level to individual ports. False or OFF: Outputs different power levels to individual ports.
Preset value	True or ON

Examples

```
Dim OutCpl As Boolean
SCPI.SOURce(1).POWer.PORT.COUPle = False
OutCpl = SCPI.SOURce(1).POWer.PORT.COUPle
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude

Equivalent key**Sweep Setup > Power > Port Couple****Equivalent SCPI command**

Syntax

```
:SOURce{[1]-160}:POWer:PORT:COUPle {ON|OFF|1|0}
:SOURce{[1]-160}:POWer:PORT:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT:COUP OFF"
20 OUTPUT 717;":SOUR1:POW:PORT:COUP?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.SPAN**Object type**Property (**Read-Write**)**Syntax**SCPI.SOURce(Ch).POWer.SPAN = *Value**Value* = SCPI.SOURce(Ch).POWer.SPAN**Description**

This command sets/gets the span value of the sweep range for the power sweep for the channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Span value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz -55 to 10 dBm 5 to 6 GHz -55 to 9 dBm 6 to 7 GHz -55 to 8 dBm 7 to 8.5 GHz -55 to 7 dBm
Preset value	15
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pspan As Double
SCPI.SOURce(1).POWer.SPAN = 10
Pspan = SCPI.SOURce(1).POWer.SPAN
```


Related objects

SCPI.SENSE(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.CENTer

Equivalent key

Span

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:SPAN <numeric>

:SOURce{[1]-160}:POWer:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SPAN 10"
20 OUTPUT 717;":SOUR1:POW:SPAN?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.START

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.START = *Value**Value* = SCPI.SOURce(Ch).POWer.START

Description

This command sets/gets the start value of the sweep range for the power sweep for channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Start value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz -55 to 10 dBm 5 to 6 GHz -55 to 9 dBm 6 to 7 GHz -55 to 8 dBm 7 to 8.5 GHz -55 to 7 dBm
Preset value	-15
Unit	dBm
Resolution	0.05

Examples

```
Dim Pstart As Double
SCPI.SOURce(1).POWer.START = -10
Pstart = SCPI.SOURce(1).POWer.START
```

Related objects

SCPI.SENSE(Ch).SWEep.TYPE
 SCPI.SOURce(Ch).POWer.STOP

Equivalent key

Start

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:STARt <numeric>
:SOURce{[1]-160}:POWer:STARt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:STAR -10"
20 OUTPUT 717;":SOUR1:POW:STAR?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).POWer.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(Ch).POWer.STOP = *Value**Value* = SCPI.SOURce(Ch).POWer.STOP

Description

This command sets/gets the stop value of the sweep range for the power sweep for channels 1 to 36 (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Stop value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz -55 to 10 dBm 5 to 6 GHz -55 to 9 dBm 6 to 7 GHz -55 to 8 dBm 7 to 8.5 GHz -55 to 7 dBm
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pstop As Double
SCPI.SOURce(1).POWer.STOP = 10
Pstop = SCPI.SOURce(1).POWer.STOP
```

Related objects

SCPI.SENSE(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.START

Equivalent key

Stop

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:STOP <numeric>

:SOURce{[1]-160}:POWer:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:STOP 10"
20 OUTPUT 717;":SOUR1:POW:STOP?"
30 ENTER 717;A
```

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.ASENSor.RCFactor**Object type**Property (**Read-Write**)**Syntax**

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.ASENSor.RCFactor =
Value

Value =

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.ASENSor.RCFactor

Description

This command sets/gets the reference calibration coefficient (the calibration coefficient at 50 MHz) for power sensor A.

Variable

Parameter	<i>Value</i>
Description	Reference calibration coefficient
Data type	Double precision floating point type (Double)
Range	1 to 150
Preset value	100
Unit	% (percent)
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim CalRef As Double

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.ASENSor.RCFactor = 99.5

CalRef = SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.ASENSor.RCFactor

Related objects

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.BSENSor.RCFactor

Equivalent key

Cal > Power Calibration > Sensor A Settings > Ref Cal Factor

Equivalent SCPI command

Syntax

```
:SOURce:POWer:PORT:CORRection:COLLect:ASENsor:RCFactor
<numeric>
:SOURce:POWer:PORT:CORRection:COLLect:ASENsor:RCFactor?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:ASEN:RCF 99.5"
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:ASEN:RCF?"
30 ENTER 717;A
```

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.BSENSor.RCFactor

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.BSENSor.RCFactor =
Value

Value =

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.BSENSor.RCFactor

Description

This command sets/gets the reference calibration coefficient (the calibration coefficient at 50 MHz) for power sensor B.

Variable

Parameter	<i>Value</i>
Description	Reference calibration coefficient
Data type	Double precision floating point type (Double)
Range	1 to 150
Preset value	100
Unit	% (percent)
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

Dim CalRef As Double

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.BSENSor.RCFactor = 99

CalRef = SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.BSENSor.RCFactor

Related objects

SCPI.SOURce.POWER.PORT.CORRection.COLLeCt.ASENSor.RCFactor

Equivalent key

Cal > **Power Calibration** > **Sensor B Settings** > **Ref Cal Factor**

Equivalent SCPI command

Syntax

```
:SOURce:POWer:PORT:CORRection:COLLect:BSENsor:RCFactor
<numeric>
:SOURce:POWer:PORT:CORRection:COLLect:BSENsor:RCFactor?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:BSEN:RCF 99.5"
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:BSEN:RCF?"
30 ENTER 717;A
```

SCPI.SOURce.POWER.PORT.CORRection.COLlect.NTOLerance**Object type**Property (**Read-Write**)**Syntax**SCPI.SOURce.POWER.PORT.CORRection.COLlect.NTOLerance= *Value**Value* = SCPI.SOURce.POWER.PORT.CORRection.COLlect.NTOLerance**Description**

This command sets/gets the tolerance of power calibration data for each measurement point, for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Tolerance of power calibration
Data type	Double precision floating point type (Double)
Range	0 to 100
Preset value	5
Unit	dB
Resolution	0.001
Note	When the measurement result is beyond the tolerance, an error message appears and the power correction function does not turn on.

Examples

```
Dim TorVal As Long
SCPI.SOURce(1).POWER.PORT(1).CORRection.COLlect.NTOLerance = 10
TorVal = SCPI.SOURce(1).POWER.PORT(1).CORRection.COLlect.NTOLerance
```

Related objectsSCPI.SOURce(*Ch*).POWER.PORT(*Pt*).CORRection.COLlect.ACQUIRE**Equivalent key****Cal > Power Calibration > Tolerance****Equivalent SCPI command**

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}
:CORRection:COLLect:NTOLerance <numeric>
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}
:CORRection:COLLect:NTOLerance?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:NTOL 6"
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:NTOL?"
30 ENTER 717;A
```

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLE.ASEnSor.DATA**Object type**Property (**Read-Write**)**Syntax**

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLE.ASEnSor.DATA =
Data

Data =

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLE.ASEnSor.DATA

Description

This command sets/gets the calibration coefficient table for power sensor A.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (for the calibration coefficient table) of $1 + \text{Num}$ (number of set data items)$\times 2$. Where n is an integer between 1 and Num.</p> <ul style="list-style-type: none"> <i>Data(0)</i>: The number of data items you want to set. Specify an integer between 0 to 100. When you set the number of data items to 0 (to clear the calibration coefficient table), you specify only <i>Data(0)</i> as the Data variable. <i>Data(n\times2-1)</i>: The frequency of the n-th data item (1 kHz to 500 GHz). <i>Data(n\times2)</i>: The calibration coefficient of the n-th data item (1% to 150%). The index of the array starts from 0.
Data type	Variant type (Variant)
Note	<p>If the array data does not contain $1 + \text{Num}$ (number of set data items)$\times 2$ when setting a calibration coefficient table, a runtime error occurs. For <i>Data(n\times2-1)</i> and <i>Data(n\times2)</i> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```

Dim CalFac As Variant
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnsor.DATA = Array(2,1e7,99.8,
1e9,98.7)
CalFac = SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnsor.DATA
""Clear Cal Factor Table
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnsor.DATA = Array(0)

Dim CalFac(4) As Variant
Dim Ref As Variant
CalFac(0) = 2
CalFac(1) = 1e7
CalFac(2) = 99.8
CalFac(3) = 1e9
CalFac(4) = 98.7
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnsor.DATA = CalFac
Ref = SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnsor.DATA
""Clear Cal Factor Table
Dim CalFac(0) As Variant
CalFac(0) = 0
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnsor.DATA = CalFac

```

Related objects

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnsor.DATA

Equivalent key

Cal > Power Calibration > Sensor A Settings > Delete | Add | Clear Cal Factor Table

Equivalent SCPI command

Syntax

```

:SOURce:POWER:PORT:CORRection:COLlect:TABLe:ASEnsor:DATA
<numeric 1>,...,<numeric 1+(Nx2)>
:SOURce:POWER:PORT:CORRection:COLlect:TABLe:ASEnsor:DATA?

```

Query response

```
{numeric 1},...,{numeric 1+(Nx2)}<newline><^END>
```

Example of use

```

10 DIM B(1:3,1:2)
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA 3,";
30 OUTPUT 717;"1E6,98.5,1E7,99,1E8,99.5"
40 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA?"
50 ENTER 717;A,B(*)
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA 0" !Clear Table

```

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLE.BSENsor.DATA**Object type**Property (**Read-Write**)**Syntax**

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLE.BSENsor.DATA =
Data

Data =

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLE.BSENsor.DATA

Description

This command sets/gets the calibration coefficient table for power sensor B.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates the array data (for the calibration coefficient table) of $1 + \text{Num}$ (number of set data items)$\times 2$. Where n is an integer between 1 and Num.</p> <ul style="list-style-type: none"> <i>Data(0)</i>: The number of data items you want to set. Specify an integer between 0 to 100. When you set the number of data items to 0 (to clear the calibration coefficient table), you specify only <i>Data(0)</i> as the Data variable. <i>Data(n\times2-1)</i>: The frequency of the n-th data item (1 kHz to 500 GHz). <i>Data(n\times2)</i>: The calibration coefficient of the n-th data item (1% to 150%). <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Note	<p>If the array data does not contain $1 + \text{Num}$ (number of set data items)$\times 2$ when setting a calibration coefficient table, a runtime error occurs. For <i>Data(n\times2-1)</i> and <i>Data(n\times2)</i> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim CalFac As Variant
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnSor.DATA = Array(2,1e7,99.8,
1e9,98.7)
CalFac = SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnSor.DATA
""Clear Cal Factor Table
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnSor.DATA = Array(0)
```

```
Dim CalFac(4) As Variant
Dim Ref As Variant
CalFac(0) = 2
CalFac(1) = 1e7
CalFac(2) = 99.8
CalFac(3) = 1e9
CalFac(4) = 98.7
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnSor.DATA = CalFac
Ref = SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnSor.DATA
""Clear Cal Factor Table
Dim CalFac(0) As Variant
CalFac(0) = 0
SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.BSEnSor.DATA = CalFac
```

Related objects

SCPI.SOURce.POWER.PORT.CORRection.COLlect.TABLe.ASEnSor.DATA

Equivalent key

Cal > Power Calibration > Sensor B Settings > Delete | Add | Clear Cal Factor Table

Equivalent SCPI command

Syntax

```
:SOURce:POWer:PORT:CORRection:COLlect:TABLe:BSEnSor:DATA
<numeric 1>,...,<numeric 1+(Nx2)>
:SOURce:POWer:PORT:CORRection:COLlect:TABLe:BSEnSor:DATA?
<numeric 1> parameter.
```

Query response

```
{numeric 1},...,{numeric 1+(Nx2)}<newline><^END>
```

Example of use

```
10 DIM B(1:3,1:2)
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA 3,";
30 OUTPUT 717;"1E6,98.5,1E7,99,1E8,99.5"
40 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA?"
50 ENTER 717;A,B(*)
```

E5071C

10 OUTPUT 717,";SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA 0" !Clear Table

STATUS

SCPI.STATus.OPERation.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.OPERation.CONDition

Description

This command reads the value of the Operation Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Operation Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long  
Stat = SCPI.STATus.OPERation.CONDition
```

Related objects

SCPI.STATus.OPERation.NTRansition
SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:COND?"  
20 ENTER 717;A
```

SCPI.STATus.OPERation.ENABLE**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.OPERation.ENABLE = *Value**Value* = SCPI.STATus.OPERation.ENABLE**Description**

This command sets/gets the value of Operation Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Operation Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 3, bit 6 to13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.ENABLE = 16
Stat = SCPI.STATus.OPERation.ENABLE
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:OPERation:ENABLE <numeric>

:STATus:OPERation:ENABLE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:ENAB 16"  
20 OUTPUT 717;":STAT:OPER:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.OPERation.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.OPERation.EVENT

Description

This command reads the value of Operation Status Event Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Operation Status Event Register
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.OPERation.EVENT

Related objects

SCPI.IEEE4882.CLS
SCPI.STATus.OPERation.NTRansition
SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:OPER?"
20 ENTER 717;A

SCPI.STATus.OPERation.NTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.OPERation.NTRansition = *Value**Value* = SCPI.STATus.OPERation.NTRansition**Description**

This command sets/gets the value of negative transition filter of the Operation Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 3, bit 6 to13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.NTRansition = 16
Stat = SCPI.STATus.OPERation.NTRansition
```

Related objects

SCPI.STATus.OPERation.EVENT

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:OPERation:NTRansition <numeric>

:STATus:OPERation:NTRansition?

Query response

{numeric}<newline><^END>

E5071C

Example of use

```
10 OUTPUT 717;":STAT:OPER:NTR 16"  
20 OUTPUT 717;":STAT:OPER:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.OPERation.PTRansition**Object type**

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.PTRansition = *Value*

Value = SCPI.STATus.OPERation.PTRansition

Description

This command sets/gets the value of positive transition filter of the Operation Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	16432
Note	The bit 0 to 3, bit 6 to13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.PTRansition = 0
Stat = SCPI.STATus.OPERation.PTRansition
```

Related objects

SCPI.STATus.OPERation.EVENT

SCPI.STATus.OPERation.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:OPERation:PTRansition <numeric>

:STATus:OPERation:PTRansition?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:PTR 16"  
20 OUTPUT 717;":STAT:OPER:PTR?"  
30 ENTER 717;A
```


SCPI:STATus:PRESet**Object type**

Method (**Write-only**)

Syntax

SCPI:STATus:PRESet

Description

This command initializes all the status registers.

Examples

SCPI:STATus:PRESet

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:PRESet

Equivalent key

No equivalent key is available on the front panel.

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATUS.QUEStionable.BLIMit.CHANnel(Ch).CONDition

Description

This command reads the value of the Questionable Bandwidth Limit Channel Status Condition Register of the specified channel.

Variable

Parameter	Value
Description	The value of the Questionable Bandwidth Limit Channel Status Condition Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long
Stat = SCPI.STATUS.QUEStionable.BLIMit.CHANnel(1).CONDition

Related objects

SCPI.STATUS.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

SCPI.STATUS.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;" :STAT:QUES:BLIM:CHAN1:COND?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.CONDiTion**Object type**Property (**Read-only**)**Syntax**

Value =

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.CONDiTion

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Extra Status Condition Register" of the specified channel.

Variable

Parameter	Value
Description	The value of the Questionable Bandwidth Limit Channel Extra Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.CONDiTion

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.ENABLE**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.ENABLE = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.ENABLE

Description

This command sets/gets the value of the Questionable Bandwidth Limit Channel Extra Status Enable Register of the specified channel.

Variable

Parameter	<i>Value</i>
Description	Value of the enable register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.ENABLE = 6

Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.ENABLE

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:ENABLE
<numeric>
```

```
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:ENABLE?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:ENAB 6"  
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENT

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Extra Status Event Register" of the specified channel.

Variable

Parameter	<i>Value</i>
Description	the value of the Questionable Bandwidth Limit Channel Extra Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition**Object type**Property (**Read-Write**)**Syntax**

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition =
Value
```

```
Value =
```

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition
```

Description

This command sets/gets the value of the negative transition filter of the "Questionable Bandwidth Limit Channel Extra Status Register" of the specified channel.

Variable

Parameter	Value
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Channel Extra Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
```

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.NTRansition = 6
```

```
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.NTRansition
```

Related objects

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENT
```

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:NTRansition
<numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:NTR 6"  
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:NTR?"  
30 ENTER 717;A
```


SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition**Object type**Property (**Read-Write**)**Syntax**

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition =
Value
```

```
Value =
```

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition
```

Description

This command sets/gets the value of the positive transition filter of the "Questionable Bandwidth Limit Channel Extra Status Register" for the specified channel.

Variable

Parameter	Value
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Channel Extra Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
```

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.PTRansition = 6
```

```
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.PTRansition
```

Related objects

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENT
```

```
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:PTRansition
<numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:PTR 6"  
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABLE**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABLE = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABLE

Description

This command sets/gets the value of the "Questionable Bandwidth Limit Channel Status Enable Register" for the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Bandwidth Limit Channel Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ENABLE = 16

Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ENABLE

Related objects

SCPI.STATus.QUEStionable.BLIMit.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ENABLE <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ENABLE?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENT

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Status Event Register" of the specified channel.

Variable

Parameter	Value
Description	The value of the Questionable Bandwidth Limit Channel Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}[:EVENT]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Description

This command sets/gets the value of the negative transition filter of the "Questionable Bandwidth Limit Channel Status Register" for the specified channel.

Variable

Parameter	Value
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Channel Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).NTRansition = 16

Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).NTRansition

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENT

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:NTRansition <numeric>  
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:NTRansition?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:NTR 16"  
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Description

This command sets/gets the value of the positive transition filter of the "Questionable Bandwidth Limit Channel Status Register" for the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Channel Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).PTRansition = 0

Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).PTRansition

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENT

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax


```
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:PTRansition <numeric>  
:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:PTRansition?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:PTR 16"  
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CONDiTion

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CONDiTion

Description

This command reads the value of the Questionable Bandwidth Limit Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Bandwidth Limit Status Condition Register.
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CONDiTion

Related objects

SCPI.STATus.QUEStionable.BLIMit.NTRansition
SCPI.STATus.QUEStionable.BLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:BLIM:COND?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.BLIMit.ELIMit.CONDiTion**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.CONDiTion

Description

This command reads the value of the Questionable Bandwidth Limit Extra Status Condition Register.

Variable

Parameter	Value
Description	The value of the Questionable Bandwidth Limit Extra Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.CONDiTion
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:ELIMit:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABLE**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABLE = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABLE

Description

This command sets/gets the value of the Questionable Bandwidth Limit Extra Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Bandwidth Limit Extra Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABLE = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABLE
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:ELIMit:ENABLE <numeric>

:STATus:QUEStionable:BLIMit:ELIMit:ENABLE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:ENAB 6"  
20 OUTPUT 717;":STAT:QUES:BLIM:ELIM:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENT

Description

This command reads the value of the Questionable Bandwidth Limit Extra Status Event Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Bandwidth Limit Extra Status Event Register.
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENT

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ELIMit[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:BLIM:ELIM?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Bandwidth Limit Extra Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENT

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:ELIMit:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:ELIMit:NTRansition?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:NTR 6"  
20 OUTPUT 717;":STAT:QUES:BLIM:ELIM:NTR?"  
30 ENTER 717;A
```


SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Bandwidth Limit Extra Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENT

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:ELIMit:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:ELIMit:PTRansition?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:PTR 6"  
20 OUTPUT 717;":STAT:QUES:BLIM:ELIM:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ENABLE**Object type**

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.ENABLE = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ENABLE

Description

This command sets/gets the value of the Questionable Bandwidth Limit Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Bandwidth Limit Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ENABLE = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.ENABLE
```

Related objects

SCPI.STATus.QUEStionable.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:ENABLE <numeric>

:STATus:QUEStionable:BLIMit:ENABLE?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:BLIM:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.EVENT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.EVENT

Description

This command reads the value of the Questionable Bandwidth Limit Status Event Register.

Variable

Parameter	Value
Description	The value of the Questionable Bandwidth Limit Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long  
Stat = SCPI.STATus.QUEStionable.BLIMit.EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM?"  
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.NTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Bandwidth Limit Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.EVENT

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:NTR 16"  
20 OUTPUT 717;":STAT:QUES:BLIM:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.PTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.BLIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Bandwidth Limit Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.EVENT

SCPI.STATus.QUEStionable.BLIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:BLIMit:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:PTR 16"  
20 OUTPUT 717;":STAT:QUES:BLIM:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.CONDiTion

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.CONDiTion

Description

This command reads the value of the Questionable Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Status Condition Register
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.CONDiTion

Related objects

SCPI.STATus.QUEStionable.NTRansition
SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:COND?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.ENABLE**Object type**

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.ENABLE = *Value*

Value = SCPI.STATus.QUEStionable.ENABLE

Description

This command sets/gets the value of the Questionable Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 7 and bit 12 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.ENABLE = 6
Stat = SCPI.STATus.QUEStionable.ENABLE
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:ENABLE <numeric>

:STATus:QUEStionable:ENABLE?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.EVENT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.EVENT

Description

This command reads the value of the Questionable Status Event Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.EVENT
```

Related objects

SCPI.IEEE4882.CLS
SCPI.STATus.QUEStionable.NTRansition
SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Description

This command reads the value of the Questionable Limit Channel Status Condition Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Channel Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition
SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.CONDition**Object type**Property (**Read-only**)**Syntax***Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition**Description**

This command reads the value of the Questionable Limit Channel Extra Status Condition Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Channel Extra Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition
 SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:CONDition?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.ENABLE**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.ENABLE = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.ENABLE**Description**

This command sets/gets the value of the Questionable Limit Channel Extra Status Enable Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Channel Extra Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.ENABLE = 6
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.ENABLE
```

Related objectsSCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABLE**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:ENABLE

<numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:ENABLE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:ENAB 6"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.EVENT

Description

This command reads the value of the Questionable Limit Channel Extra Status Event Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Channel Extra Status Event Register
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.EVENT

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition =
Value

Value =

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Channel Extra Status Register the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.NTRansition = 6

Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.NTRansition

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.EVENT

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

E5071C

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:NTRansition
<numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:NTR 6"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition =
Value

Value =

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Limit Channel Extra Status Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.PTRansition = 6

Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.PTRansition

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.EVENT

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

E5071C

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:PTRansition
<numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:PTR 6"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABLE**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABLE = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABLE**Description**

This command sets/gets the value of the Questionable Limit Channel Status Enable Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Channel Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABLE = 16
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABLE
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ENABLE <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ENABLE?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ENAB?"  
30 ENTER 717;A
```


SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENT**Object type**Property (**Read-only**)**Syntax***Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENT**Description**

This command reads the value of the Questionable Limit Channel Status Event Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Channel Status Event Register of the specified channel
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Channel Status Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 15 can not be set to 1.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition = 16

Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENTSCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:NTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR 16"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition**Description**

This command sets/gets the value of the positive transition filter of the Questionable Limit Channel Status Register of the specified channel (*Ch*).

Variable

Parameter	<i>Value</i>
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition = 0

Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition

Related objectsSCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENTSCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition**Equivalent key**

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:PTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR 16"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CONDiTion

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CONDiTion

Description

This command reads the value of the Questionable Limit Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Status Condition Register
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CONDiTion

Related objects

SCPI.STATus.QUEStionable.LIMit.NTRansition
SCPI.STATus.QUEStionable.LIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:LIM:COND?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.LIMit.ELIMit.CONDition**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.CONDition

Description

This command reads the value of the Questionable Limit Extra Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Extra Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:ELIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABLE**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABLE = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABLE**Description**

This command sets/gets the value of the Questionable Limit Extra Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Extra Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABLE = 6
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABLE
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUEStionable:LIMit:ELIMit:ENABLE <numeric>
:STATus:QUEStionable:LIMit:ELIMit:ENABLE?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:ENAB 6"  
20 OUTPUT 717;":STAT:QUES:LIM:ELIM:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENT

Description

This command reads the value of the Questionable Limit Extra Status Event Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Extra Status Event Register
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENT

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ELIMit[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:LIM:ELIM?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition**Description**

This command sets/gets the value of the negative transition filter of the Questionable Limit Extra Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENT

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:ELIMit:NTRansition <numeric>

:STATus:QUEStionable:LIMit:ELIMit:NTRansition?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:NTR 6"  
20 OUTPUT 717;":STAT:QUES:LIM:ELIM:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition**Description**

This command sets/gets the value of the positive transition filter of the Questionable Limit Extra Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENT

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:ELIMit:PTRansition <numeric>

:STATus:QUEStionable:LIMit:ELIMit:PTRansition?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:PTR 6"  
20 OUTPUT 717;":STAT:QUES:LIM:ELIM:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ENABLE**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.ENABLE = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.ENABLE**Description**

This command sets/gets the value of the Questionable Limit Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ENABLE = 16
Stat = SCPI.STATus.QUEStionable.LIMit.ENABLE
```

Related objects

SCPI.STATus.QUEStionable.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:ENABLE <numeric>

:STATus:QUEStionable:LIMit:ENABLE?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:LIM:ENAB?"  
30 ENTER 717;A
```


SCPI.STATus.QUEStionable.LIMit.EVENT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.EVENT

Description

This command reads the value of the Questionable Limit Status Event Register.

Variable

Parameter	<i>Value</i>
Description	Value of the Questionable Limit Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.NTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.NTRansition = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.NTRansition**Description**

This command sets/gets the value of the negative transition filter of the Questionable Limit Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 15 can not be set to 1.

Examples

Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.NTRansition

Related objects

SCPI.STATus.QUEStionable.LIMit.EVENT

SCPI.STATus.QUEStionable.LIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:NTRansition <numeric>

:STATus:QUEStionable:LIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR 16"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.PTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.LIMit.PTRansition = *Value**Value* = SCPI.STATus.QUEStionable.LIMit.PTRansition**Description**

This command sets/gets the value of positive transition filter of the Questionable Limit Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.EVENT

SCPI.STATus.QUEStionable.LIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:LIMit:PTRansition <numeric>

:STATus:QUEStionable:LIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR 16"  
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.NTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.NTRansition = *Value**Value* = SCPI.STATus.QUEStionable.NTRansition**Description**

This command sets/gets the value of negative transition filter of the Questionable Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit (0 to 9) and bit (11 to 15) can not be set to 1.

Examples

Dim Stat As Long
SCPI.STATus.QUEStionable.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.NTRansition

Related objects

SCPI.STATus.QUEStionable.EVENT

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:NTRansition <numeric>

:STATus:QUEStionable:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:NTR 16"  
20 OUTPUT 717;":STAT:QUES:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.PTRansition**Object type**Property (**Read-Write**)**Syntax**SCPI.STATus.QUEStionable.PTRansition = *Value**Value* = SCPI.STATus.QUEStionable.PTRansition**Description**

This command sets/gets the value of the positive transition filter of the Questionable Status Register.

Variable

Parameter	<i>Value</i>
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	1024
Note	The bit 0 to 9 and bit 11 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENT

SCPI.STATus.QUEStionable.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:PTRansition <numeric>

:STATus:QUEStionable:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:PTR 16"  
20 OUTPUT 717;":STAT:QUES:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition

Description

This command reads the value of the Questionable Ripple Limit Channel Status Condition Register for the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Channel Status Condition Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).CONDition

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:COND?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.CONDition**Object type**Property (**Read-only**)**Syntax**

Value =

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.CONDition

Description

This command reads the value of the Questionable Ripple Limit Channel Extra Status Condition Register of the specified channel.

Variable

Parameter	Value
Description	The value of the Questionable Ripple Limit Channel Extra Status Condition Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.CONDition

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.NTRansition

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:COND?"
20 ENTER 717;A
```

SCPI:STATus:QUESTionable:RLIMit:CHANnel(Ch):ECHannel:ENABle**Object type**Property (**Read-Write**)**Syntax**

SCPI:STATus:QUESTionable:RLIMit:CHANnel(Ch):ECHannel:ENABle = Value

Value = SCPI:STATus:QUESTionable:RLIMit:CHANnel(Ch):ECHannel:ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Channel Extra Status Enable Register of the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Channel Extra Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI:STATus:QUESTionable:RLIMit:CHANnel(1):ECHannel:ENABle = 6

Stat = SCPI:STATus:QUESTionable:RLIMit:CHANnel(1):ECHannel:ENABle

Related objects

SCPI:STATus:QUESTionable:RLIMit:CHANnel(Ch):ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUESTionable:RLIMit:CHANnel{[1]-160}:ECHannel:ENABle
<numeric>
```

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:EChannel:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:ENAB 6"  
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENT

Description

This command reads the value of the Questionable Ripple Limit Channel Extra Status Event Register of the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Channel Extra Status Event Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.EVENT

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.NTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ECHannel.NTRansition =
Value

Value =

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ECHannel.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Channel Extra Status Register of the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the negative transition filter of the Questionable Ripple Limit Channel Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.NTRansition = 6

Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.NTRansition

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ECHannel.EVENT

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:NTRansition
<numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:NTR 6"  
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:NTR?"  
30 ENTER 717;A
```


SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition**Object type**Property (**Read-Write**)**Syntax**

```
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition =
Value
```

```
Value =
```

```
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition
```

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Channel Extra Status Register of the specified channel.

Variable

Parameter	Value
Description	The value of the positive transition filter of the Questionable Ripple Limit Channel Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
```

```
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.PTRansition = 6
```

```
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.PTRansition
```

Related objects

```
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENT
```

```
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.NTRansition
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

E5071C

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:PTRansition
<numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:PTR 6"  
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Channel Status Enable Register for the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Channel Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ENABle = 16

Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ENABle

Related objects

SCPI.STATus.QUEStionable.RLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ENABle <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ENABle?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENT

Description

This command reads the value of the Questionable Ripple Limit Channel Status Event Register of the specified channel.

Variable

Parameter	Value
Description	The value of the Questionable Ripple Limit Channel Status Event Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}[:EVENT]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Channel Status Register for the specified channel.

Variable

Parameter	<i>Value</i>
Description	The value of the negative transition filter of the Questionable Ripple Limit Channel Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).NTRansition = 16

Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).NTRansition

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENT

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

```
:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:NTRansition <numeric>  
:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:NTRansition?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:NTR 16"  
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Channel Status Register of the specified channel.

Variable

Parameter	Value
Description	The value of the positive transition filter of the Questionable Ripple Limit Channel Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

Dim Stat As Long

SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).PTRansition = 0

Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).PTRansition

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENT

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax


```
:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:PTRansition <numeric>  
:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:PTRansition?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:PTR 16"  
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CONDiTion

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CONDiTion

Description

This command reads the value of the Questionable Ripple Limit Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Status Condition Register.
Data type	Long integer type (Long)

Examples

Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CONDiTion

Related objects

SCPI.STATus.QUEStionable.RLIMit.NTRansition
SCPI.STATus.QUEStionable.RLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

10 OUTPUT 717;":STAT:QUES:RLIM:COND?"
20 ENTER 717;A

SCPI.STATus.QUEStionable.RLIMit.ELIMit.CONDiTion**Object type**Property (**Read-only**)**Syntax**

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.CONDiTion

Description

This command reads the value of the Questionable Ripple Limit Extra Status Condition Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Extra Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.CONDiTion
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:ELIMit:CONDiTion?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABLE**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABLE = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABLE

Description

This command sets/gets the value of the Questionable Ripple Limit Extra Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Extra Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABLE = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABLE
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:ELIMit:ENABLE <numeric>

:STATus:QUEStionable:RLIMit:ELIMit:ENABLE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:ENAB 6"  
20 OUTPUT 717;":STAT:QUES:RLIM:ELIM:ENAB?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENT

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENT

Description

This command reads the value of the Questionable Ripple Limit Extra Status Event Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Extra Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENT
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ELIMit[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;" :STAT:QUES:RLIM:ELIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Extra Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the negative transition filter of the Questionable Ripple Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENT

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:ELIMit:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:ELIMit:NTRansition?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:NTR 6"  
20 OUTPUT 717;":STAT:QUES:RLIM:ELIM:NTR?"  
30 ENTER 717;A
```


SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Extra Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the positive transition filter of the Questionable Ripple Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENT

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:ELIMit:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:ELIMit:PTRansition?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:PTR 6"  
20 OUTPUT 717;":STAT:QUES:RLIM:ELIM:PTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ENABLE**Object type**

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.ENABLE = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ENABLE

Description

This command sets/gets the value of the Questionable Ripple Limit Status Enable Register.

Variable

Parameter	<i>Value</i>
Description	The value of the Questionable Ripple Limit Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ENABLE = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.ENABLE
```

Related objects

SCPI.STATus.QUEStionable.ENABLE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:ENABLE <numeric>

:STATus:QUEStionable:RLIMit:ENABLE?

Query response

E5071C

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ENAB 16"  
20 OUTPUT 717;":STAT:QUES:RLIM:ENAB?"  
30 ENTER 717;A
```

SCPI:STATus:QUEStionable:RLIMit:EVENT**Object type**

Property (**Read-only**)

Syntax

Value = SCPI:STATus:QUEStionable:RLIMit:EVENT

Description

This command reads the value of the Questionable Ripple Limit Status Event Register.

Variable

Parameter	Value
Description	The value of the Questionable Ripple Limit Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI:STATus:QUEStionable:RLIMit:EVENT
```

Related objects

SCPI:IEEE4882:CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit[:EVENT]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.NTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the negative transition filter of the Questionable Ripple Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.EVENT

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:NTRansition <numeric>

:STATus:QUESTionable:RLIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:NTR 16"  
20 OUTPUT 717;":STAT:QUES:RLIM:NTR?"  
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.PTRansition**Object type**Property (**Read-Write**)**Syntax**

SCPI.STATus.QUEStionable.RLIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Status Register.

Variable

Parameter	<i>Value</i>
Description	The value of the positive transition filter of the Questionable Ripple Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.EVENT

SCPI.STATus.QUEStionable.RLIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:STATus:QUEStionable:RLIMit:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:PTR 16"  
20 OUTPUT 717;":STAT:QUES:RLIM:PTR?"  
30 ENTER 717;A
```

SYSTEM**SCPI.SYSTem.BACKlight**

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BACKlight = *Status**Status* = SCPI.SYSTem.BACKlight

Description

This command turns ON/OFF or return the status of the backlight of the LCD display.

- When the backlight is OFF, you cannot read the information on the display.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the backlight
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the backlight. • False or OFF: Turns OFF the backlight.
Preset value	True or ON

Examples

```
Dim BckLght As Boolean
SCPI.SYSTem.BACKlight = False
BckLght = SCPI.SYSTem.BACKlight
```

Equivalent key

System > Backlight

- To turn the backlight ON, press any key on the front panel.

Equivalent SCPI command

Syntax

```
:SYSTem:BACKlight {ON|OFF|1|0}
:SYSTem:BACKlight?
```

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:BACK OFF"  
20 OUTPUT 717;":SYST:BACK?"  
30 ENTER 717;A
```

SCPI.SYSTem.BEEPer.COMPLete.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.BEEPer.COMPLete.IMMediate

Description

This command generates a beep for the notification of the completion of an operation.

Examples

SCPI.SYSTem.BEEPer.COMPLete.IMMediate

Related objects

SCPI.SYSTem.BEEPer.COMPLete.STATe

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Equivalent key

System > **Misc Setup** > **Beeper** > **Test Beep Complete**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:COMPLete:IMMediate

Example of use

10 OUTPUT 717;":SYST:BEEP:COMP:IMM"

SCPI.SYSTem.BEEPer.COMPlete.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.BEEPer.COMPlete.STATe = *Status**Status* = SCPI.SYSTem.BEEPer.COMPlete.STATe**Description**

This command turns ON/OFF or returns the status of the beeper for the notification of the completion of the operation.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the beeper for the notification of the completion of the operation
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the beeper for the notification of the completion of the operation. • False or OFF: Turns OFF the beeper for the notification of the completion of the operation.
Preset value	True or ON

Examples

```
Dim BeepComp As Boolean
SCPI.SYSTem.BEEPer.COMPlete.STATe = False
BeepComp = SCPI.SYSTem.BEEPer.COMPlete.STATe
```

Related objects

SCPI.SYSTem.BEEPer.COMPlete.IMMEDIATE

SCPI.SYSTem.BEEPer.WARNIng.STATe

Equivalent key**System > Misc Setup > Beeper > Beep Complete****Equivalent SCPI command****Syntax**

E5071C

:SYSTem:BEEPer:COMPlate:STATe {ON|OFF|1|0}

:SYSTem:BEEPer:COMPlate:STATe?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":SYST:BEEP:COMP:STAT OFF"

20 OUTPUT 717;":SYST:BEEP:COMP:STAT?"

30 ENTER 717;A

SCPI.SYSTem.BEEPer.WARning.IMMediate**Object type**

Method (**Write-only**)

Syntax

SCPI.SYSTem.BEEPer.WARning.IMMediate

Description

This command generates a beep for the notification of warning/limit test results.

Examples

SCPI.SYSTem.BEEPer.WARning.IMMediate

Related objects

SCPI.SYSTem.BEEPer.WARning.STATe

SCPI.SYSTem.BEEPer.COMPLete.IMMediate

Equivalent key

System > Misc Setup > Beeper > Test Beep Warning

Equivalent SCPI command**Syntax**

:SYSTem:BEEPer:WARning:IMMediate

Example of use

10 OUTPUT 717;":SYST:BEEP:WARN:IMM"

SCPI.SYSTem.BEEPer.WARning.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.BEEPer.WARning.STATe = *Status**Status* = SCPI.SYSTem.BEEPer.WARning.STATe**Description**

This command turns ON/OFF or return the status of the beeper for the notification of warning/limit test results.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the beeper for the notification of warning/limit test result
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the beeper for the notification of warning/limit test result. • False or OFF: Turns OFF the beeper for the notification of warning/limit test result.
Preset value	True or ON

Examples

```
Dim BeepWarn As Boolean
SCPI.SYSTem.BEEPer.WARning.STATe = False
BeepWarn = SCPI.SYSTem.BEEPer.WARning.STATe
```

Related objects

SCPI.SYSTem.BEEPer.WARning.IMMediate

SCPI.SYSTem.BEEPer.COMplete.STATe

Equivalent key**System > Misc Setup > Beeper > Beep Warning****Equivalent SCPI command****Syntax**


```
:SYSTem:BEEPer:WARNing:STATe {ON|OFF|1|0}  
:SYSTem:BEEPer:WARNing:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:BEEP:WARN:STAT OFF"  
20 OUTPUT 717;":SYST:BEEP:WARN:STAT?"  
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.ECAL.CATalog

Object type

Property (**Read-only**)

Syntax

Ctlg = SCPI.SYSTem.COMMunicate.ECAL.CATalog

Description

This command reads the ID string of ECal that are connected to the ENA through USB. Use the list to select an ECal.

Variable

Parameter	<i>Ctlg</i>
Description	Returns a string containing the following parameters for all the connected ECals: <ol style="list-style-type: none">1. Model Number2. Serial Number
Data type	Character string type (String)

Examples

```
Dim Ctlg As String
Ctlg = SCPI.SYSTem.COMMunicate.ECAL.CATalog
```

Related objects

SCPI.SYSTem.COMMunicate.ECAL.DEFine

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:ECAL:CATalog

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYSTem:COMMunicate:ECAL:CATalog?"
20 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.ECAL.DEFine**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.COMMunicate.ECAL.DEFine = *ECaId**ECaId* = SCPI.SYSTem.COMMunicate.ECAL.DEFine**Description**

This command sets/gets the ID string of the ECAL being used at that time.

Variable

Parameter	<i>ECaId</i>
Description	Returns ECaId string containing the following parameters: <ol style="list-style-type: none"> 1. Model Number 2. Serial Number
Data type	Character string type (String)

Examples

```
Dim ECal As String, ECalReturn As String
SCPI.SYSTem.COMMunicate.ECAL.DEFine = "N4431B 03605"
ECalReturn = SCPI.SYSTem.COMMunicate.ECAL.DEFine
```

Related objects

SCPI.SYSTem.COMMunicate.ECAL.CATalog

Equivalent key**Cal > ECal > ECal****Equivalent SCPI command****Syntax**

```
:SYSTem:COMMunicate:ECAL:DEFine <string>
:SYSTem:COMMunicate:ECAL:DEFine?
```

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYSTem:COMMunicate:ECAL:DEFine ""N4431B 03605""
10 OUTPUT 717;":SYSTem:COMMunicate:ECAL:DEFine?"
20 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDress**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDress = *Value**Value* = SCPI.COMMunicate.GPIB.PMETer.ADDress**Description**

This command sets/gets the GPIB address of the power meter in use.

Variable

Parameter	<i>Value</i>
Description	GPIB address of the power meter
Data type	Long integer type (Long)
Range	0 to 30
Preset value	13
Note	If the specified parameter is out of the allowable setting range, a runtime error occurs.

Examples

```
Dim Paddr As Long
SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDress = 15
Paddr = SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDress
```

Equivalent key**System > Misc Setup > GPIB Setup > Power Meter Address****Equivalent SCPI command****Syntax**

```
:SYSTem:COMMunicate:GPIB:PMETer:ADDRes <numeric>
:SYSTem:COMMunicate:GPIB:PMETer:ADDRes?
```

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:PIB:PMET:ADDR 10"  
20 OUTPUT 717;":SYST:COMM:PIB:PMET:ADDR?"  
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDress

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDress = *Value**Value* = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDress

Description

This command sets/gets the GPIB address of the external signal source in use.

Variable

Parameter	<i>Value</i>
Description	The GPIB address of the external signal source
Data type	Long integer type (Long)
Range	0 to 30
Preset value	19
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGaddr As Long
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDress = 20
SGaddr = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDress
```

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:ADDress <numeric>
:SYSTem:COMMunicate:GPIB:SGENerator:ADDress?
```

Query response

{value}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:PIB:SGEN:ADDR 19"  
20 OUTPUT 717;":SYST:COMM:PIB:SGEN:ADDR?"  
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency =
Param

Param = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

Description

This command sets/gets the output frequency setting command for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	Value
Description	Frequency setting command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	"FR %f% HZ"
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed.</p> <p>The output frequency in the command string is written as "%f%." The frequency of the external signal source, which changes for each measurement point, is set and the setting command is sent to the external signal source.</p>

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency = "FREQ %f%HZ"
SGcomm = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address > Custom Commands > Set Frequency

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:FREQuency
<string>
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:FREQuency?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:FREQ ""FREQ %f%HZ""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:FREQ?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer = *Param**Param* = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

Description

This command sets/gets the output level setting command for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	Value
Description	Level setting command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	"AP %p% DM"
Note	<p>If the specified variable is out of the allowable setup range, an error occurs when executed.</p> <p>The output level in the command string is written as "%p%." The output level of the external signal source, which changes for each measurement point, is set and the setting command is sent to the external signal source.</p>

Example of use

Dim SGcomm As String

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer = "AMPL %p%DBM"

SGcomm = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Signal Generator Address** > **Custom Commands** > **Set Power Level**

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:POWer <string>

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:POWer?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:POW ""AMPL %p%DBM""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:POW?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet = *Param**Param* = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.POWER

Description

This command sets/gets the preset command for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	Value
Description	Preset command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	""
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet =
"*RST"
SGcomm =
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet
```

Related objects

```
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWER
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE
```

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Signal Generator Address** > **Custom Commands** > **Preset**

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:PRESet <string>
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:PRESet?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:PRES ""*RST""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:PRES?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.RFON

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.RFON = *Param**Param* = SCPI.COMMunicate.GPIB.SGEnerator.CCOMmand.RFON

Description

This command sets/gets the command to turn on RF for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	Value
Description	RF:on setting command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	"R3"
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.RFON = "AMPL:STATE ON"
SGcomm = SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.RFON
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.DWELI

SCPI.SYSTem.COMMunicate.GPIB.SGEnerator.TYPE

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address > Custom Commands > Turn RF Out On

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:RFON <string>
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:RFON?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:RFON ""AMPL:STATE ON""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:RFON?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI = *Value**Value* = SCPI.COMMunicate.GPIB.SGENerator.DWELI

Description

This command sets/gets the wait time after setting the frequency or output level for the external signal source in use.

Variable

Parameter	<i>Value</i>
Description	Wait time setting for the user-defined external signal source
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0.1
Unit	s (second)
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

Dim SGdelay As Double

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI = 0.2

SGdelay = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Signal Generator Address** > **Switching Time**

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:DWELI <numeric>
:SYSTem:COMMunicate:GPIB:SGENerator:DWELI?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:DWEL 0.2"
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:DWEL?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE = *Value**Value* = SCPI.COMMunicate.GPIB.SGENerator.TYPE

Description

This command sets/gets the type of the external signal source in use.

Variable

Parameter	<i>Value</i>
Description	Type setting of the external signal source 1: User-defined 2: 8643A, 8644B, 8664A, 8665A/B 3: 8648A/B/C/D, ESG Series, PSG Series
Data type	Long integer type (long)
Range	1 to 3
Preset value	3
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGtype As long
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE = 2
SGtype = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON
 SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELI

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address > Custom Commands | 8643A, 8644B, 8664A, 8665A/B | 8648A/B/C/D, ESG Series, PSG Series

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:TYPE <numeric>
:SYSTem:COMMunicate:GPIB:SGENerator:TYPE?
```

Query response

```
{value}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:TYPE 2"
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:TYPE?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.PSEnSor**Object type**Property (**Read-Write**)**Syntax**

SCPI.SYSTem.COMMunicate.PSEnSor = <char>,<string>

Param = SCPI.SYSTem.COMMunicate.PSEnSor**Description**

This command specifies the type and location of the power meter to be used in a source power calibration

Variable

Parameter	<i>char</i>
Description	Power Meter Type
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "GPIB": GPIB Power Meter. • "USB": USB Power Meter.

Parameter	<i>string</i>
Description	Address/ID string of Power Meter
Data type	Character string type (String)
Range	For char=GPIB, address of the power meter. Choose any integer between 0 and 30. For char=USB, the ID string of the power meter or power sensor. Use SYST:COMM:USB:PMET:CAT? to see a list of ID strings of connected power meters and sensors.

Examples

```
Dim PsenModel As String
PsenModel = SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog
SCPI.SYSTem.COMMunicate.PSEnSor = Array("USB", PsenModel)
```

Related objects

SYSTem.COMMunicate.USB.PMETer.CATalog

Equivalent key

System > Misc Setup > Power Meter Setup > Select Type > GPIB|USB

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:PSEnSor {GPIB|USB},{string (address/ID string)}
:SYSTem:COMMunicate:PSEnSor?
```

Query response

```
{GPIB|USB},{string (address/ID string)}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:USB:PMET:CAT?"
20 ENTER 717;A$
30 OUTPUT 717;":SYST:COMM:PSEN USB,"&A$
```

SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog

Object type

Property (**Read-only**)

Syntax

Ctlg = SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog

Description

This command reads the ID string of power meters / sensors that are connected to the ENA through USB. Use the list to select a power sensor for a source power cal.

Variable

Parameter	<i>Ctlg</i>
Description	Returns a string containing the following parameters for all the connected power meters/sensors: <ul style="list-style-type: none">1. Agilent Technologies2. Model Number3. Serial Number4. Firmware Revision
Data type	Character string type (String)

Examples

```
Dim Ctlg As String
Ctlg = SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog
```

Related objects

SCPI.SYSTem.COMMunicate.PSEnSor

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:USB:PMETer:CATalog

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;";SYSTem:COMMunicate:USB:PMETer:CATalog?"  
20 ENTER 717;A$
```

SCPI.SYSTem.CORRection.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.CORRection.STATe = *Status**Status* = SCPI.SYSTem.CORRection.STATe

Description

This command turns ON/OFF or returns the status of the system correction.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of the system correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none">• True or ON: Turns ON the system correction.• False or OFF: Turns OFF the system correction.
Preset value	True or ON

Examples

```
Dim SysCal As Boolean
SCPI.SYSTem.CORRection.STATe = False
SysCal = SCPI.SYSTem.CORRection.STATe
```

Equivalent key

System > Misc Setup > Service Menu > System Correction

Equivalent SCPI command

Syntax

:SYSTem:CORRection[:STATe] {ON|OFF|1|0}

:SYSTem:CORRection[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:CORR OFF"  
20 OUTPUT 717;":SYST:CORR?"  
30 ENTER 717;A
```

SCPI.SYSTem.DATE**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.DATE = *Data**Data* = SCPI.SYSTem.DATE**Description**

This command sets/gets the date of the clock built in the E5071C.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 3-element array data (date of the built-in clock).</p> <ul style="list-style-type: none"> • <i>Data(0)</i> Sets year. • <i>Data(1)</i> Sets month. • <i>Data(2)</i> Sets day. <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<ul style="list-style-type: none"> • <i>Data(0)</i> 2002 to 2037 • <i>Data(1)</i> 1 to 12 • <i>Data(2)</i> 1 to 31
Resolution	1
Note	<p>If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.</p>

Examples

```
Dim Day As Variant
SCPI.SYSTem.DATE = Array(2005,12,10)
Day = SCPI.SYSTem.DATE
```

```
Dim Day(2) As Variant
Dim Ref As Variant
Day(0) = 2005
Day(1) = 12
Day(2) = 10
SCPI.SYSTem.DATE = Day
Ref = SCPI.SYSTem.DATE
```

Related objects

```
SCPI.SYSTem.TIME
SCPI.DISPlay.CLOCK
```

Equivalent key

System > Misc Setup > Clock Setup > Set Date and Time

Equivalent SCPI command

Syntax

```
:SYSTem:DATE <numeric 1 (year)>,<numeric 2 (month)>,<numeric 3
(day)>
:SYSTem:DATE?
```

Query response

```
{numeric 1 (year)},{numeric 2 (month)},{numeric 3
(day)}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:DATE 2005,12,10"
20 OUTPUT 717;":SYST:DATE?"
30 ENTER 717;A,B,C
```

SCPI.SYSTem.ERRor**Object type**Property (**Read-only**)**Syntax***Err* = SCPI.SYSTem.ERRor**Description**

This command reads the oldest error from the list of errors stored in the error queue of the E5071C. The read-out error is deleted from the error queue. The size of the error queue is 100.

Executing SCPI.IEEE4882.CLS command clears the errors stored in the error queue.

- This object can not return an error that occurs by the manual operation or the SCPI command used in controlling the E5071C from the external con-troller.

Variable

Parameter	<i>Err</i>
Description	<p>Indicates 2-element array data (for error).</p> <ul style="list-style-type: none"> • <i>Err(0)</i> :Error number • <i>Err(1)</i> :Error message <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Note	If no error is stored in the error queue, 0 and "No error" are read out as the error number and the error message.

Examples

```
Dim Err As Variant
Err = SCPI.SYSTem.ERRor
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:SYSTem:ERRor?

Query response

{numeric},{string}<newline><^END>

{numeric}:

Error number

{string}:

Error message (a character string with double quotation marks (""))

If no error is stored in the error queue, 0 and "No error" are read out as the error number and the error message.

Example of use

10 OUTPUT 717;":SYST:ERR?"

20 ENTER 717;A,B\$

SCPI.SYSTem.ISPControl.PORT**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.ISPControl.PORT = *Value**Status* = SCPI.SYSTem.ISPControl.PORT**Description**

This command sets/gets a test port to be selected for stimulus destination when the Initial Source Port Control feature is on.

Variable

Parameter	<i>Value</i>
Description	Selected Port
Data type	Long integer type (Long)
Range	1to 4
Preset value	1

- For option 2D5, 4D5, 2K5 and 4K5, the preset value is 1 and cannot be changed. When this command is send to the option 2D5, 4D5, 2K5 and 4K5, an error of "Option Not Installed" is displayed.

Examples

```
SCPI.INIT(ch).CONT = False
SCPI.SYST.ISPC.STAT = True
SCPI.SYST.ISPC.PORT = 1
```

Related objects

SCPI.SYSTem.ISPControl.STATe

Equivalent key**System** > **Service** > **Init Src Port [1|2|3|4]**

- For option 2D5, 4D5, 2K5 and 4K5, the key is not displayed.

Equivalent SCPI command**Syntax**

```
:SYSTem:ISPControl:PORT <numeric>
:SYSTem:ISPControl:PORT?
```

Query response

{1-4}<newline><^END>

Example of use

```
10 :INIT:CONT OFF
20 :SYSTem:ISPControl[:STATe] ON
30 :SYSTem:ISPControl:PORT 1
```

SCPI.SYSTem.ISPControl.STATe**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.ISPControl.STATe = *Status**Status* = SCPI.SYSTem.ISPControl.STATe**Description**

This command turns ON/OFF or returns the status of the Initial Source Port Control feature (to switch the stimulus output in the trigger hold state to a test port).

Variable

Parameter	<i>Status</i>
Description	ON/OFF Initial Source Port Control feature
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: ON Control feature. • False or OFF: OFF Control feature.
Preset value	True or ON

- The default value is ON for all options.

Examples

```
SCPI.INIT(ch).CONT = False
SCPI.SYST.ISPC.STAT = True
SCPI.SYST.ISPC.PORT = 1
```

Related objects

SCPI.SYSTem.ISPC.PORT

Equivalent key**System > Service > Init Src Ctrl [ON]****Equivalent SCPI command****Syntax**


```
:SYSTem:ISPControl [:STATe] {ON|OFF|1|0}  
:SYSTem:ISPControl [:STATe] ?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 :INIT:CONT OFF  
20 :SYSTem:ISPControl[:STATe] ON  
30 :SYSTem:ISPControl:PORT 1
```

SCPI.SYSTem.KLOCK.KBD**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.KLOCK.KBD = *Status**Status* = SCPI.SYSTem.KLOCK.KBD**Description**

This command sets/gets whether to lock the operation of the front panel (key and rotary knob) and keyboard.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of lock
Data type	Boolean type (Boolean)
Range	Select from either of the following. <ul style="list-style-type: none">• True or ON: Specifies lock.• False or OFF: Specifies unlock.
Preset value	False or OFF

Examples

```
Dim FKLock As Boolean
SCPI.SYSTem.KLOCK.KBD = True
FKLock = SCPI.SYSTem.KLOCK.KBD
```

Related objects

SCPI.SYSTem.KLOCK.MOUSE

Equivalent key**System > Misc Setup > Front Panel & Keyboard Lock****Equivalent SCPI command****Syntax**

```
:SYSTem:KLOCK:KBD {ON|OFF|1|0}
:SYSTem:KLOCK:KBD?
```

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:KLOC:KBD ON"  
20 OUTPUT 717;":SYST:KLOC:KBD?"  
30 ENTER 717;A
```

SCPI.SYSTem.KLOCK.MOUSe**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.KLOCK.MOUSe = *Status**Status* = SCPI.SYSTem.KLOCK.MOUSe**Description**

This command sets/gets whether to lock the operation of the mouse and touch screen.

Variable

Parameter	<i>Status</i>
Description	ON/OFF of lock
Data type	Boolean type (Boolean)
Range	Select from either of the following. <ul style="list-style-type: none">• True or ON: Specifies lock.• False or OFF: Specifies unlock.
Preset value	False or OFF

Examples

```
Dim MTLock As Boolean
SCPI.SYSTem.KLOCK.MOUSe = True
MTLock = SCPI.SYSTem.KLOCK.MOUSe
```

Related objects

SCPI.SYSTem.KLOCK.KBD

Equivalent key**System > Key Lock > Mouse Lock****Equivalent SCPI command****Syntax**

:SYSTem:KLOCK:MOUSe {ON|OFF|1|0}

:SYSTem:KLOCK:MOUSe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:KLOC:MOUS ON"  
20 OUTPUT 717;":SYST:KLOC:MOUS?"  
30 ENTER 717;A
```

E5071C

SCPI.SYSTem.POFF

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.POFF

Description

This command turns OFF the E5071C.

Examples

SCPI.SYSTem.POFF

Equivalent key

Standby switch

Equivalent SCPI command

Syntax

:SYSTem:POFF

Example of use

10 OUTPUT 717;":SYST:POFF"

SCPI.SYSTem.PRESet**Object type**

Method (**Write-only**)

Syntax

SCPI.SYSTem.PRESet

Description

This command presets the setting state of the E5071C to the original factory setting (Default Conditions). This command is different from SCPI.IEEE4882.RSTas the continuous startup mode (see SCPI.INITiate(Ch).CONTInuous) of channel 1 is set to ON.

Examples

SCPI.SYSTem.PRESet

Related objects

SCPI.IEEE4882.RST

SCPI.SYSTem.UPReset

Equivalent key

Preset > OK

Equivalent SCPI command**Syntax**

:SYSTem:PRESet

Example of use

10 OUTPUT 717;":SYST:PRES"

SCPI.SYSTem.SECurity.LEVel

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.SECurity.LEVel = Param

Param = SCPI.SYSTem.SECurity.LEVel

Description

This command sets/gets the security level of the E5071C.

Variable

Parameter	<i>Param</i>
Description	The security level.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "NONE": Specifies OFF to the security level. • "LOW": Specifies LOW level to the security level. • "HIGH": Specifies HIGH level to the security level.
Preset value	"NONE"
Note	<p>When the setting is LOW, it is able to change to NONE or HIGH. But when this setting is HIGH, it is not able to change NONE or LOW.</p> <p>The setting can be turned NONE by executing the preset or recalling when the setting of frequency blank function is HIGH.</p> <p>Even if the setting is LOW and HIGH, the command that reads out the frequency is not influenced.</p>

Examples

```
Dim SecLev As String
SCPI.SYSTem.SECurity.LEVel = "LOW"
SecLev = SCPI.SYSTem.SECurity.LEVel
```

Equivalent key

System > Service Menu > Security Level > None|Low|High

Equivalent SCPI command

Syntax

```
:SYSTem:SECurity:LEVel {NONE|LOW|HIGH}
:SYSTem:SECurity:LEVel?
```

Query response

```
{NONE|LOW|HIGH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:SEC:LEV LOW"
20 OUTPUT 717;":SYST:SEC:LEV?"
30 ENTER 717;A$
```

E5071C

SCPI.SYSTem.SERVice

Object type

Property (**Read-only**)

Syntax

Status = SCPI.SYSTem.SERVice

Description

This command reads whether the E5071C is in the service mode or not.

Variable

Parameter	<i>Status</i>
Description	Whether to be in the service mode
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none">• True or ON: In the service mode.• False or OFF : Not in the service mode.
Preset Value	False or OFF

Examples

Dim SvMode As Boolean
SvMode = SCPI.SYSTem.SERVice

Equivalent key

Displayed on the instrument status bar (at the bottom of the LCD display).

Equivalent SCPI command

Syntax

:SYSTem:SERVice?

Query response

{1|0}<newline><^END>

Example of use

10 OUTPUT 717;":SYST:SERV?"
30 ENTER 717;A

SCPI.SYSTem.TEMPerature.HIGH**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.TEMPerature.HIGH = *Status**Status* = SCPI.SYSTem.TEMPerature.HIGH**Description**

This command does not works in E5071C. This command is provided because of its command compatibility with E5070B/E5071B. On execution, this command does not works and generates no error.

Variable

Parameter	<i>Status</i>
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: No Meaning. • False or OFF: No meaning.
Preset value	False or OFF

Examples

```
Dim TempMode As Boolean
SCPI.SYSTem.TEMPerature.HIGH = True
TempMode = SCPI.SYSTem.TEMPerature.HIGH
```

Equivalent key

None

Equivalent SCPI command**Syntax**

```
:SYSTem:TEMPerature:HIGH {ON|OFF|1|0}
:SYSTem:TEMPerature:HIGH?
```

Query response

{1|0}<newline><^END>

Example of use

E5071C

```
10 OUTPUT 717;":SYST:TEMP:HIGH ON"  
20 OUTPUT 717;":SYST:TEMP:HIGH?"  
30 ENTER 717;A
```

SCPI.SYSTem.TEMPerature.STATe**Object type**

Property (**Read-only**)

Syntax

Status = SCPI.SYSTem.TEMPerature.STATe

Description

This command does not works in E5071C. This command is provided because of its command compatibility with E5070B/E5071B. On execution, this command does not works and generates no error.

Variable

Parameter	<i>Status</i>
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Always True

Examples

Dim WarmUp As Boolean
WarmUp = SCPI.SYSTem.TEMPerature.STATe

Equivalent key

None

Equivalent SCPI command**Syntax**

:SYSTem:TEMPerature[:STATe]?

Query response

{1}<newline><^END>

Example of use

10 OUTPUT 717;":SYST:TEMP?"
30 ENTER 717;A

SCPI.SYSTem.TIME**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.TIME = *Data**Data* = SCPI.SYSTem.TIME**Description**

This command sets/gets the time of the clock built in the E5071C.

Variable

Parameter	<i>Data</i>
Description	<p>Indicates 3-element array data (time of the built-in clock).</p> <p><i>Data(0)</i>: Sets hour (24-hour basis)</p> <p><i>Data(1)</i> :Sets minute.</p> <p><i>Data(2)</i> :Sets second.</p> <p>The index of the array starts from 0.</p>
Data type	Variant type (Variant)
Range	<p><i>Data(0)</i> :0 to 23</p> <p><i>Data(1)</i> :0 to 59</p> <p><i>Data(2)</i> :0 to 59</p>
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Time As Variant
SCPI.SYSTem.TIME = Array(21,30,0)
Time = SCPI.SYSTem.TIME
```

```
Dim Time(2) As Variant
Dim Ref As Variant
Time(0) = 21
Time(1) = 30
Time(2) = 0
SCPI.SYSTem.TIME = Time
Ref = SCPI.SYSTem.TIME
```

Related objects

SCPI.SYSTem.DATE
SCPI.DISPlay.CLOCK

Equivalent key

System > Misc Setup > Clock Setup > Set Date and Time

Equivalent SCPI command

Syntax

```
:SYSTem:TIME <numeric 1>,<numeric 2>,<numeric 3>
:SYSTem:TIME?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:TIME 17,30,0"
20 OUTPUT 717;":SYST:TIME?"
30 ENTER 717;A,B,C
```

SCPI.SYSTem.TSET.EXTernal**Object type**Property (**Read-Write**)**Syntax**SCPI.SYSTem.TSET.EXTernal = *Param**Param* = SCPI.SYSTem.TSET.EXTernal**Description**

This command turns on/off the external connection test set, and selects the setting of the test set mode. When the external connection test set is on (MODE1/MODE2), only the S11, S21, or absolute value measurements are available. The fixture simulator's balance measurement function and frequency offset function are not available. When the external connection test set is changed, the calibration coefficients are cleared, and the error correction function is turned off.

Variable

Parameter	<i>Param</i>
Description	External test set mode
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "OFF": Turns off the external test set. • "MODE1": Turns on the external test set, and sets the test set mode to MODE1. • "MODE2": Turns on the external test set, and sets the test set mode to MODE2.
Preset value	"OFF"

Examples

Dim ExtTset As String

SCPI.SYSTem.TSET.EXTernal = "MODE1"

ExtTset = SCPI.SYSTem.TSET.EXTernal

Equivalent key**System > Service Menu > External Test Set****Equivalent SCPI command**

Syntax

```
:SYSTem:TSET:[EXTernal] {OFF|MODE1|MODE2}
:SYSTem:TSET:[EXTernal]?
```

Query response

```
{OFF|MODE1|MODE2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:TSET MODE1"
20 OUTPUT 717;":SYST:TSET?"
30 ENTER 717;A
```

E5071C

SCPI.SYSTem.UPReset

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.UPReset

Description

This command presets the E5071C with the user settings. The command is executed regardless of the operation mode in preset state.

- If you try to specify a file for a preset (D:\UserPreset.sta) that does not exist, a warning message will be displayed and SCPI.SYSTem.PRESet will be executed.

Examples

CPI.SYSTem.UPReset

Related objects

SCPI.IEEE4882.RST

SCPI.SYSTem.PRESet

Equivalent key

Preset > OK

Equivalent SCPI command

Syntax

:SYSTem:UPReset

Example of use

10 OUTPUT 717;":SYST:UPR"

TRIGGER**SCPI.TRIGger.OUTPUT.POLarity**

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPUT.POLarity=*param**param*=SCPI.TRIGger.OUTPUT.POLarity

Description

This command sets/gets the polarity of the pulse generated by the External Trigger Output Port.

Variable

Parameter	<i>Param</i>
Description	Sets/Gets polarity of the pulse generated by the output trigger.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "POSitive": Generates a Positive pulse. • "NEGative": Generates a Negative pulse.
Preset value	"POSitive"

Examples

Dim TrigPol As String

TrigPol="NEGative"

SCPI.TRIGger.OUTPUT.POLarity=TrigPol

Related objects

SCPI.TRIGger.OUTPUT.POSition

SCPI.TRIGger.OUTPUT.STATe

Equivalent key

Trigger > Polarity

Equivalent SCPI command

Syntax

E5071C

:TRIGger:OUTPut:POLarity {POSitive|NEGative}
:TRIGger:OUTPut:POLarity?

Query response

{POSitive|NEGative}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:POL POS"  
20 OUTPUT 717;":TRIG:OUTP:POL?"  
30 ENTER 717;A$
```

SCPI.TRIGger.OUTPUT.POSition**Object type**Property (**Read-Write**)**Syntax**

SCPI.TRIGger.OUTPUT.POSition=Param

Param=SCPI.TRIGger.OUTPUT.POSition

Description

This command sets/gets the position of the External Trigger Output Port.

Variable

Parameter	<i>Param</i>
Description	Sets/Gets (after or before measurement point) position of the output trigger.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "AFTer": Generates a Pulse (trigger) after the measurement points. • "BEFore": Generates a Pulse (trigger) before the measurement points.
Preset value	"AFTer"

Examples

Dim TrigPos As String

TrigPos="BEFore"

SCPI.TRIGger.OUTPUT.POSition=TrigPos

Related objects

SCPI.TRIGger.OUTPUT.POLarity

SCPI.TRIGger.OUTPUT.STATe

Equivalent key**Trigger > Position****Equivalent SCPI command**

E5071C

Syntax

:TRIGger:OUTPut:POSition {AFTer|BEFore}

:TRIGger:OUTPut:POSition?

Query response

{AFTer|BEFore}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:POS AFT"  
20 OUTPUT 717;":TRIG:OUTP:POS?"  
30 ENTER 717;A$
```

SCPI.TRIGger.OUTPUT.STATE**Object type**Property (**Read-Write**)**Syntax**

SCPI.TRIGger.OUTPUT.STATE=Param

Param=SCPI.TRIGger.OUTPUT.STATE

Description

This command sets/gets the External Trigger Output Port state.

Variable

Parameter	<i>Param</i>
Description	Sets/Gets external trigger output state.
Data type	Boolean
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns ON the External Trigger Output • False or OFF : Turns OFF the External Trigger Output
Preset value	"OFF"

Examples

Dim TrigStat As boolean

TrigStat=1

SCPI.TRIGger.OUTPUT.STATE=TrigStat

Related objects

SCPI.TRIGger.OUTPUT.POLarity

SCPI.TRIGger.OUTPUT.POSition

Equivalent key**Trigger > Ext Trig Output****Equivalent SCPI command****Syntax**

:TRIGger:OUTPUT[:STATE] {ON|1|OFF|0}

E5071C

:TRIGger:OUTPut[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:STAT ON"  
20 OUTPUT 717;":TRIG:OUTP:STAT?"  
30 ENTER 717;A$
```


SCPI.TRIGger.SEQuence.AVERage

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.AVERage = *Status**Status* = SCPI.TRIGger.SEQuence.AVERage

Description

This command turns ON/OFF or gets the status of the averaging trigger function.

- The sweep averaging feature must be set to *ON* when turning on the averaging trigger feature.

Variable

Parameter	<i>Status</i>
Description	On/off status of the averaging trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following. <ul style="list-style-type: none"> • True or ON: Turns on the averaging trigger. • False or OFF: Turns off the averaging trigger.
Preset value	False or OFF

Example of use

Dim Avetrig as Boolean

SCPI.TRIGger.SEQuence.AVERage = True

Avetrig = TRIGger.SEQuence.AVERage

Related objects

SCPI.SENSE(Ch).AVERage.STATe

Equivalent key

Ave > **Ave Trigger**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:AVERage {ON|OFF|1|0}

:TRIGger[:SEQuence]:AVERage?

E5071C

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:AVER ON"  
20 OUTPUT 717;":TRIG:AVER?"  
30 ENTER 717;A
```

For a complete sample program using VBA, please refer to Averaging Sample Program.

SCPI.TRIGger.SEQuence.EXTErnal.DELEay

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.EXTErnal.DELEay = *Value**Value* = SCPI.TRIGger.SEQuence.EXTErnal.DELEay

Description

This command sets/gets the time that it takes from receiving the trigger to starting measurement when the trigger source is external.

Variable

Parameter	<i>Value</i>
Description	External trigger delay time
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0
Unit	s (second)
Resolution	10 μ

Example of use

```
Dim ExtDel As Double
SCPI.TRIGger.SEQuence.EXTErnal.DELEay = 0.05
ExtDel = SCPI.TRIGger.SEQuence.EXTErnal.DELEay
```

Related objects

SCPI.TRIGger.SEQuence.EXTErnal.LLATency.STATe

SCPI.TRIGger.SEQuence.POINt

Equivalent key

Trigger > **Ext Trigger Delay**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:EXTErnal:DELEay <numeric>

:TRIGger[:SEQuence]:EXTErnal:DELEay?

E5071C

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:EXT:DEL 0.05"  
20 OUTPUT 717;":TRIG:EXT:DEL?"  
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTeRnal.LLATency.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.EXTeRnal.LLATency.STATe = *Status**Status* = SCPI.TRIGger.SEQuence.EXTeRnal.LLATency.STATe

Description

This command turns ON/OFF or returns the status of the low-latency external trigger feature.

When turning on the low-latency external trigger feature, the point trigger feature must be set to on and the trigger source must be set to external trigger.

- When the point trigger feature is set to off, or when the trigger source is set to one other than the external trigger, the change is ignored.

Variable

Parameter	<i>Status</i>
Description	On/off of the low-latency external trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following: <ul style="list-style-type: none"> • True or ON: Turns on the low-latency external trigger. • False or OFF: Turns off the low-latency external trigger.
Preset value	False or OFF

Example of use

```
Dim Llat As Boolean
SCPI.TRIGger.SEQuence.EXTeRnal.LLATency.STATe = True
Llat = SCPI.TRIGger.SEQuence.EXTeRnal.LLATency.STATe
```

Related objects

SCPI.TRIGger.SEQuence.EXTeRnal.DELaY

SCPI.TRIGger.SEQuence.POINt

E5071C

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

Trigger > **Low Latency**

Equivalent SCPI command

Syntax

```
:TRIGger[:SEQuence]:EXTernal:LLATency[:STATe] {ON|OFF|1|0}  
:TRIGger[:SEQuence]:EXTernal:LLATency[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:EXT:LLAT ON"  
20 OUTPUT 717;":TRIG:EXT:LLAT?"  
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTeRnal.SLOPe**Object type**

Property

Syntax

SCPI.TRIGger.SEQuence.EXTeRnal.SLOPe

Description

This command sets/gets the polarity of the External Trigger Input Port.

Variable

Parameter	<i>Param</i>
Description	Sets/Gets polarity of the external input trigger.
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "POSitive": Sets/Gets Positive polarity. • "NEGative": Sets/Gets Negative polarity.
Preset value	"NEGative"

Examples

```
Dim InputTrigPol As String
InputTrigPol="NEGative"
SCPI.TRIGger.SEQuence.EXTeRnal.SLOPe=InputTrigPol
```

Related objects

SCPI.TRIGger.SEQuence.EXTeRnal.DELay
 SCPI.TRIGger.SEQuence.EXTeRnal.LLATency.STATe

Equivalent key**Trigger > Ext Trig Input****Equivalent SCPI command****Syntax**

```
:TRIGger[:SEQuence]:EXTeRnal:SLOPe {POSitive|NEGative}
:TRIGger[:SEQuence]:EXTeRnal:SLOPe?
```

Query response

E5071C

{POSitive|NEGative}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:SEQ:EXT:SLOP POS"  
20 OUTPUT 717;":TRIG:SEQ:EXT:SLOP?"  
30 ENTER 717;A$
```


SCPI.TRIGger.SEQuence.IMMEDIATE**Object type**Method (**Write-only**)**Syntax**

SCPI.TRIGger.SEQuence.IMMEDIATE

Description

This command generates a trigger immediately and executes a measurement, regardless of the setting of the trigger mode.

This command is different from SCPI.TRIGger.SEQuence.SINGLE as the execution of the object finishes at the time of a trigger.

- If you execute this object when the trigger system is not in the trigger wait state (trigger event detection state), an error occurs when executed and the object is ignored.

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMEDIATE
```

Related objects

SCPI.TRIGger.SEQuence.IMMEDIATE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:TRIGger[:SEQuence][:IMMEDIATE]

Example of use

```
10 OUTPUT 717;":TRIG:IMM"
20 OUTPUT 717;":*OPC?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.POINt

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.POINt = *Status**Status* = SCPI.TRIGger.SEQuence.POINt

Description

This command turns ON/OFF or returns the status of the point trigger feature.

- When the point trigger feature is turned on, if the sweep mode is Swept, it changes to Stepped.
When the trigger source is set to the internal trigger (Internal), the setting is ignored.

Variable

Parameter	<i>Status</i>
Description	Turns ON/OFF the point trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following: True or ON: Turns on the point trigger. False or OFF: Turns off the point trigger.
Preset value	False or OFF
Note	

Example of use

```
Dim Ptrig As Boolean
SCPI.TRIGger.SEQuence.POINt = True
Ptrig = TRIGger.SEQuence.POINt
```

Related objects

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

Trigger > Trigger Event

Equivalent SCPI command

Syntax

```
:TRIGger[:SEQuence]:POINt {ON|OFF|1|0}
:TRIGger[:SEQuence]:POINt?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:POIN ON"
20 OUTPUT 717;":TRIG:POIN?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.SCOPE

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.SCOPE = *Param**Param* = SCPI.TRIGger.SEQuence.SCOPE

Description

This command sets/gets the effective scope of triggering. When this function is enabled with a value of "ACTive", only active channel is triggered. When this function is enabled with a value of "ALL", all channels of the E5071C are triggered.

For example, if TRIGger.SCOPE value is "ACTive" when :INIT:CONT command is turned on for all channels, a measurement channel will automatically be changed by switching over the active channel.

Variable

Parameter	<i>Param</i>
Description	Trigger source
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> "ALL": Specifies trigger to all channels. "ACTive": Specifies trigger to active channel.
Preset value	"ALL"

Examples

```
Dim TrigScope As Enum
SCPI.TRIGger.SEQuence.SCOPE = "ACTive"
TrigScope = SCPI.TRIGger.SEQuence.SCOPE
```

Equivalent key

Trigger > Trigger Scope > All Channel|Active Channel

Equivalent SCPI command

Syntax

```
:TRIGger[:SEQuence]:SCOPE {ALL|ACTive}
:TRIGger[:SEQuence]:SCOPE?
```

Query response

{ALL|ACTive}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:SEQ:SCOP ACT"  
20 OUTPUT 717;":TRIG:SEQ:SCOP?"  
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.SINGle**Object type**Method (**Write-only**)**Syntax**

SCPI.TRIGger.SEQuence.SINGle

Description

This command generates a trigger immediately and executes a measurement, regardless of the setting of the trigger mode.

This command is different from SCPI.TRIGger.SEQuence.IMMediate as the execution of the object finishes when the measurement (all of the sweep) initiated with this object is complete. In other words, you can wait for the end of the measurement using the SCPI.IEEE4882.OPC object.

If you execute this object when the trigger system is not in the trigger wait state (trigger event detection state), an error occurs when executed and the object is ignored.

For details about the trigger system, see Trigger System.

Examples

```
Dim Dmy As Long
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.SINGle
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.TRIGger.SEQuence.IMMediate

SCPI.IEEE4882.OPC

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command**Syntax**

:TRIGger[:SEQuence]:SINGle

Example of use

```
10 OUTPUT 717;":TRIG:SING"
20 OUTPUT 717; "**OPC?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.SOURce**Object type**Property (**Read-Write**)**Syntax**SCPI.TRIGger.SEQuence.SOURce = *Param**Param* = SCPI.TRIGger.SEQuence.SOURce**Description**

This command sets/gets the trigger source from the following 4 types:

1. Internal Trigger
Uses the internal trigger to generate continuous triggers automatically.
2. External Trigger
Generates a trigger when the trigger signal is inputted externally via the Ext Trig connector or the handler interface.
3. Manual Trigger
Generates a trigger when the key operation of **Trigger** > **Trigger** is executed from the front panel.
4. Bus Trigger
Generates a trigger when the SCPI.IEEE4882.TRG object is executed.
5. When you change the trigger source during sweep, the sweep is aborted.

Variable

Parameter	<i>Param</i>
Description	Trigger source
Data type	Character string type (String)
Range	Select from either of the following: <ul style="list-style-type: none"> • "INTernal": Specifies internal trigger. • "EXTernal": Specifies external trigger. • "MANual": Specifies manual trigger. • "BUS": Specifies bus trigger.
Preset value	"INTernal"

Examples

E5071C

```
Dim TrigSour As String
SCPI.TRIGger.SEQuence.SOURce = "bus"
TrigSour = SCPI.TRIGger.SEQuence.SOURce
```

Equivalent key

Trigger > **Trigger Source** > **Internal|External|Manual|Bus**

Equivalent SCPI command

Syntax

```
:TRIGger[:SEQuence]:SOURce {INTernal|EXTernal|MANual|BUS}
:TRIGger[:SEQuence]:SOURce?
```

Query response

```
{BUS|EXT|INT|MAN}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:SOUR BUS"
20 OUTPUT 717;":TRIG:SOUR?"
30 ENTER 717;A$
```


TDR Command Reference

Option TDR Command Reference

The commands for TDR measurement (Option TDR) should be sent via SICT-LAN through TCPIP0::localhost::inst0::INSTR address. The TDR application must be installed and executed on your PC before you send the TDR commands. These commands can not be used in the built-in VBA.

The Command Finder helps you to find the command.

The TDR Notation Convention helps you understand the commands.

Other topics about Programming

Notational Conventions

This section describes the notational conventions used for the description of TDR commands reference.

Command Type

Object type describes different types of E5071C- SCPI commands. The E5071C provides properties and methods as SCPI commands. SCPI commands which set (send)/read (return) the state of the E5071C using variables are defined as property and SCPI commands which does other processing are defined as method.

SCPI commands used to read the state of the E5071C are indicated with "**Query**" and ones used only to set the state of the E5071C are indicated by "**Command**". SCPI commands that can both read and write data to the E5071C are indicated by '**Command/Query**'.

Syntax

Syntax describes the syntax for sending an SCPI command from the controller (PC) to the E5071C. The syntax consists of two parts: the object part and the set part, with a space " " inserted between them. As for query, the syntax consists of object part followed by a question mark, with no space in between them.

Description

Description describes how to use the SCPI commands or the operation when executed.

Parameters

Parameters provide description about different parameters that can be used with the SCPI commands. It gives the description, data type, allowable range, preset value, unit, resolution, and notes (if any).

The data types of the E5071C SCPI commands include 5 types as shown in the following table.

Data type	Name	Consumed memory	Range
Long	Long integer type	4 bytes	-2,147,483,648 to 2,147,483,647
Double	Double precision floating point type	8 bytes	For a negative value: -1.79769313486232E+308 to -4.94065645841247E-324

			For a positive value: -1.79769313486232E+308 to -4.94065645841247E-324
Boolean	Boolean type	2 bytes	1 (ON) or 0 (OFF)
String	Character string type	1 byte / alphanumeric character	Up to approximately 2 billion characters
Variant	Variant type	16 bytes	No limitation

Examples

Examples provides a sample example of using the object through coding with the Basic language.

Related Commands

Related commands provide information about other commands that are similar/related with the command.

Equivalent GUI

Equivalent GUI shows the operational procedure by using the E5071C-TDR graphic user interface that has the same effect as the object.

CALCULATE**:CALCulate:ALLocate****Type**

Command Only

Syntax**:CALCulate:ALLocate {SPARameters|TPARameters|MIXed}****Description**

This command sets type of the parameter & format allocation for each traces.

Parameter

Description	Type of the parameter & format allocation
Data Type	Character string type (String)
Range	SPARameters: All S TPARameters: All T MIXed: Mixed
Preset Value	MIXed
Unit	-
Resolution	-

Examples

10 OUTPUT 717;":CALC:ALL SPAR"

Equivalent GUI

TDR/TDT > Trace Control > Mixed OR All T OR All S button (under Allocation)

:CALCulate:ATRaces:ACTive**Type**

Command/Query

Syntax

:CALCulate:ATRaces:ACTive <numeric>

:CALCulate:ATRaces:ACTive?

Description

This command sets active trace number.

Parameter

Description	Active trace number
Data Type	Long integer type (Long)
Range	1 to 16
Preset Value	1
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:ATR:ACT 7"

20 OUTPUT 717;":CALC:ATR:ACT?"

30 ENTER 717;A

Related Commands

:CALCulate:ATRaces:COUNT

Equivalent GUI**Trace** > Select the trace number OR**Trace Next** or **Trace Prev**

:CALCulate:ATRaces:COUNT

Type

Query Only

Syntax

:CALCulate:ATRaces:ACTive?

Description

This command returns the number of trace.

Parameter

Description	Number of Trace.
Data Type	Long integer type (Long)
Range	1 to 16
Preset Value	Depending on DUT topology setting
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:ATR:ACT?"
20 ENTER 717;A

Related Commands

:CALCulate:ATRaces:ACTive

Equivalent GUI

No equivalent GUI is available.

:CALCulate:ATRaces:MARKer:COUPle**Type**

Command/Query

Syntax**:CALCulate:ATRaces:MARKer:COUPle {ON|OFF|1|0}****:CALCulate:ATRaces:MARKer:COUPle?****Description**

This command sets state for the marker couple mode.

Parameter

Description	State of marker coupling
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:ATR:MARK:COUP ON"  
20 OUTPUT 717;":CALC:ATR:MARK:COUP?"  
30 ENTER 717;A
```

Related Commands**:CALCulate:ATRaces:TIME:COUPle****Equivalent GUI****TDR/TDT > Trace Control > Marker** (Under **Coupling**)

:CALCulate:ATRaces:TIME:COUPle

Type

Command/Query

Syntax

:CALCulate:ATRaces:TIME:COUPle {ON|OFF|1|0}

:CALCulate:ATRaces:TIME:COUPle?

Description

This command sets state for the transform couple mode.

Parameter

Description	State of the time coupling.
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:ATR:TIME:COUP ON"
20 OUTPUT 717;":CALC:ATR:TIME:COUP?"
30 ENTER 717;A

Related Commands

:CALCulate:ATRaces:MARKer:COUPle

Equivalent Key

TDR/TDT > Trace Control > Time (Under **Coupling**)

:CALCulate:ATRaces:TIME:STEP:AMPLitude**Type**

Command/Query

Syntax

:CALCulate:ATRaces:TIME:STEP:AMPLitude <numeric>

:CALCulate:ATRaces:TIME:STEP:AMPLitude?

Description

This command sets step amplitude value for the transform function.

Parameter

Description	Stimulus amplitude level
Data Type	Double precision floating point type (Double)
Range	1m to 5
Preset Value	200m
Unit	volt
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:ATR:TIME:STEP:AMPL 100e-3"

20 OUTPUT 717;":CALC:ATR:TIME:STEP:AMPL?"

30 ENTER 717;A

Equivalent GUI**Setup > Stim. Ampl.** (under **Basic**)

:CALCulate:DEvice

Type

Command/Query

Syntax

:CALCulate:DEvice {SEND1|SEND2|DIF1|SEND4|DIF2}

:CALCulate:DEvice?

Description

This command sets DUT topology.

Parameter

Description	DUT topology
Data Type	Character string type (String)
Range	SEND1: Single-Ended 1-Port SEND2: Single-Ended 2-Port DIF1: Deferential 1-Port SEND4: Single-Ended 4-Port DIF2: Deferential 2-Port
Preset Value	SEND1
Unit	-
Resolution	-

Query Response

{SEND1|SEND2|DIF1|SEND4|DIF2}<newline><^END>

Examples

10 OUTPUT 717;":CALC:DEV SEND1"
20 OUTPUT 717;":CALC:DEV?"
30 ENTER 717;A\$

Equivalent GUI

Setup > DUT Topology (under **Basic**)

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILeName**Type**

Command/Query

Syntax

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILeName <File>

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILeName?

Description

This command sets the filename of the S2P de-embedding user file. This file is saved as a 2-port touchstone file with the .s2p extension.

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Parameter

Description	2-port touchstone file name (extension: .s2p) for the de-embedding user file
Data Type	Character string type (String)
Range	254 characters or less
Preset Value	""

Query Response

{string}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMB:S2P:PORT1:DEEM:FILe ""D:\folder\User.s2p""

20 OUTPUT 717;":CALC:EMB:S2P:PORT1:DEEM:FILe?"

30 ENTER 717;A\$

Related Commands

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILeName

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe

:CALCulate:EMBed:S4P:DIFF[1-2]:STATe

:CALCulate:EMBed:STATe

Equivalent GUI**Setup > Adv Waveform > De-embedding > Load****Eye/Mask > Advanced Waveform (Under **Stimulus**) > De-embedding > Load**

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe**Type**

Command/Query

Syntax

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe {ON|OFF|1|0}

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe?

Description

This command sets the S2P de-embedding function state ON/OFF. To turn ON, it is necessary to load the S2P file in advance.

Parameter

Description	S2P de-embedding function ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMB:S2P:PORT2:DEEM:FILE "D:\test.s2p"

20 OUTPUT 717;":CALC:EMB:S2P:PORT2:DEEM:STAT ON"

30 OUTPUT 717;":CALC:EMB:S2P:PORT2:DEEM:STAT?"

40 ENTER 717;A

Related Commands

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILENAME

:CALCulate:EMBed:S4P:DIFF[1-2]:FILENAME

:CALCulate:EMBed:S4P:DIFF[1-2]:STATe

:CALCulate:EMBed:STATe

Equivalent GUI**Setup > Adv Waveform > De-embedding > S2P File (Port 1|2|3|4)**

Eye/Mask > Advanced Waveform (Under **Stimulus) > De-embedding > S2P
File (Port 1|2|3|4)**

:CALCulate:EMBed:S4P:DIFF[1-2]:FILename**Type**

Command/Query

Syntax

:CALCulate:EMBed:S4P:DIFF[1-2]:FILename <File>

:CALCulate:EMBed:S4P:DIFF[1-2]:FILename?

Description

This command sets the filename of the S4P de-embedding user file. This file is saved as a 4-port touchstone file with the .s4p extension.

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Parameter

Description	4-port touchstone file name (extension: .s4p) for the de-embedding user file
Data Type	Character string type (String)
Range	254 characters or less
Preset Value	""

Query Response

{string}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMB:S2P:PORT1:DEEM:FIL ""D:\folder\User.s4p""

20 OUTPUT 717;":CALC:EMB:S2P:PORT1:DEEM:FIL?"

30 ENTER 717;A\$

Related Commands

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILename

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe

:CALCulate:EMBed:S4P:DIFF[1-2]:STATe

:CALCulate:EMBed:STATe

Equivalent GUI**Setup > Adv Waveform > De-embedding > Load****Eye/Mask > Advanced Waveform (Under **Stimulus**) > De-embedding > Load**

:CALCulate:EMBed:S4P:DIFF[1-2]:STATe**Type**

Command/Query

Syntax

:CALCulate:EMBed:S4P:DIFF[1-2]:DEEMbed:STATe {ON|OFF|1|0}

:CALCulate:EMBed:S4P:DIFF[1-2]:DEEMbed:STATe?

Description

This command sets the S4P de-embedding function state on/off.

Parameter

Description	S4P de-embedding function ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMB:S4P:DIFF1:DEEM ON"

20 OUTPUT 717;":CALC:EMB:S4P:DIFF1:DEEM:STAT?"

30 ENTER 717;A

Related Commands

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILeName

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe

:CALCulate:EMBed:S4P:DIFF[1-2]:FILeName

:CALCulate:EMBed:STATe

Equivalent GUI**Setup > Adv Waveform > De-embedding > S4P File (Port 1&2|3&4)****Eye/Mask > Advanced Waveform (Under **Stimulus**) > De-embedding > S4P File (Port 1&2|3&4)**

:CALCulate:EMBed:STATe**Type**

Command/Query

Syntax

:CALCulate:EMBed:STATe {ON|OFF|1|0}

:CALCulate:EMBed:STATe?

Description

This command sets the embed function state on/off.

Parameter

Description	Embed function ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMB:STAT ON"

20 OUTPUT 717;":CALC:EMB:STAT?"

30 ENTER 717;A

Related Commands

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:FILEname

:CALCulate:EMBed:S2P:PORT[1-4]:DEEMbed:STATe

:CALCulate:EMBed:S4P:DIFF[1-2]:FILEname

:CALCulate:EMBed:S4P:DIFF[1-2]:STATe

Equivalent GUI**Setup > Adv Waveform > De-embedding > Enable****Eye/Mask > Advanced Waveform (Under **Stimulus**) > De-embedding > Enable**

:CALCulate:EMPHasis:CURSor:POST2**Type**

Command/Query

Syntax

:CALCulate:EMPHasis:CURSor:POST2 <numeric>

:CALCulate:EMPHasis:CURSor:POST2?

Description

This command sets the emphasis post2 level.

Parameter

Description	Emphasis post2 level
Data Type	Double precision floating point type (Double)
Range	-20 to 20
Preset Value	-3
Unit	dB
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMP:CURS:POST2 3"

20 OUTPUT 717;":CALC:EMP:CURS:POST2?"

30 ENTER 717;A

Related Commands

:CALCulate:EMPHasis:CURSor:POST1

:CALCulate:EMPHasis:CURSor:PRE1

:CALCulate:EMPHasis:STaTe

Equivalent GUI**Setup > Adv Waveform > Emphasis > Post 2 Cursor****Eye/Mask > Advanced Waveform (Under Stimulus) > Emphasis > Post 2 Cursor**

:CALCulate:EMPHasis:CURSor:POST1**Type**

Command/Query

Syntax

:CALCulate:EMPHasis:CURSor:POST1 <numeric>

:CALCulate:EMPHasis:CURSor:POST1?

Description

This command sets the emphasis post1 level.

Parameter

Description	Emphasis post1 level
Data Type	Double precision floating point type (Double)
Range	-20 to 20
Preset Value	0
Unit	dB
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMP:CURS:POST1 3"

20 OUTPUT 717;":CALC:EMP:CURS:POST1?"

30 ENTER 717;A

Related Commands

:CALCulate:EMPHasis:CURSor:POST2

:CALCulate:EMPHasis:CURSor:PRE1

:CALCulate:EMPHasis:STATE

Equivalent GUI**Setup > Adv Waveform > Emphasis > Post 1 Cursor****Eye/Mask > Advanced Waveform (Under **Stimulus**) > Emphasis > Post 1 Cursor**

:CALCulate:EMPHasis:CURSor:PRE1**Type**

Command/Query

Syntax

:CALCulate:EMPHasis:CURSor:PRE1 <numeric>

:CALCulate:EMPHasis:CURSor:PRE1?

Description

This command sets the emphasis pre1 level.

Parameter

Description	Emphasis pre1 level
Data Type	Double precision floating point type (Double)
Range	-20 to 20
Preset Value	0
Unit	dB
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMP:CURS:PRE1 2

20 OUTPUT 717;":CALC:EMP:CURS:PRE1?"

30 ENTER 717;A

Related Commands

:CALCulate:EMPHasis:CURSor:POST2

:CALCulate:EMPHasis:CURSor:POST1

:CALCulate:EMPHasis:STATE

Equivalent GUI**Setup > Adv Waveform > Emphasis > Pre Cursor****Eye/Mask > Advanced Waveform (Under Stimulus) > Emphasis > Pre Cursor**

:CALCulate:EMPHasis:STATe**Type**

Command/Query

Syntax

:CALCulate:EMPHasis:STATe {ON|OFF|1|0}

:CALCulate:EMPHasis:STATe?

Description

This command sets the emphasis function state on/off.

Parameter

Description	Emphasis function ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EMP:STAT ON"

20 OUTPUT 717;":CALC:EMP:STAT?"

30 ENTER 717;A

Related Commands

:CALCulate:EMPHasis:CURSor:POST2

:CALCulate:EMPHasis:CURSor:POST1

:CALCulate:EMPHasis:CURSor:PRE1

Equivalent GUI**Setup > Adv Waveform > Emphasis > Enable****Eye/Mask > Advanced Waveform (Under **Stimulus**) > Emphasis > Enable**

:CALCulate:EQUalization:CTLE:DC**Type**

Command/Query

Syntax

:CALCulate:EQUalization:CTLE:DC <numeric>

:CALCulate:EQUalization:CTLE:DC?

Description

This command sets the equalization CTLE (Continuous Time Linear Equalization) DC gain parameter.

Parameter

Description	CTLE DC gain parameter
Data Type	Double precision floating point type (Double)
Range	0 - 10
Preset Value	0.667
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EQU:CTLE:DC 0.077"

20 OUTPUT 717;":CALC:EQU:CTLE:DC?"

30 ENTER 717;A\$

Related Commands

:CALCulate:EQUalization:CTLE:POLE1

:CALCulate:EQUalization:CTLE:POLE2

:CALCulate:EQUalization:CTLE:ZERO1

:CALCulate:EQUalization:FILEname

:CALCulate:EQUalization:STATe

:CALCulate:EQUalization:TYPE

Equivalent GUI**Setup > Adv Waveform > Equalization > DC Gain**

E5071C

Eye/Mask > Advanced Waveform (Under **Stimulus) > Equalization > DC Gain**

:CALCulate:EQUalization:CTLE:POLE1**Type**

Command/Query

Syntax

:CALCulate:EQUalization:CTLE:POLE1 <numeric>

:CALCulate:EQUalization:CTLE:POLE1?

Description

This command sets the equalization CTLE (Continuous Time Linear Equalization) Pole1 parameter.

Parameter

Description	CTLE DC pole1 parameter
Data Type	Double precision floating point type (Double)
Range	0 - 2E10
Preset Value	1.95E9
Unit	Hz
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EQU:CTLE:POLE1 2.05E9"
20 OUTPUT 717;":CALC:EQU:CTLE:POLE1?"
30 ENTER 717;A$
```

Related Commands

```
:CALCulate:EQUalization:CTLE:DC
:CALCulate:EQUalization:CTLE:POLE2
:CALCulate:EQUalization:CTLE:ZERO1
:CALCulate:EQUalization:FILEname
:CALCulate:EQUalization:STATe
:CALCulate:EQUalization:TYPE
```

Equivalent GUI**Setup > Adv Waveform > Equalization > Pole1 Freq**

E5071C

Eye/Mask > Advanced Waveform (Under Stimulus) > Equalization > Pole1 Freq

:CALCulate:EQUalization:CTLE:POLE2**Type**

Command/Query

Syntax

:CALCulate:EQUalization:CTLE:POLE2 <numeric>

:CALCulate:EQUalization:CTLE:POLE2?

Description

This command sets the equalization CTLE (Continuous Time Linear Equalization) Pole2 parameter.

Parameter

Description	CTLE DC pole2 parameter
Data Type	Double precision floating point type (Double)
Range	0 - 2E10
Preset Value	5E9
Unit	Hz
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EQU:CTLE:POLE2 5.05E9"
20 OUTPUT 717;":CALC:EQU:CTLE:POLE2?"
30 ENTER 717;A$
```

Related Commands

```
:CALCulate:EQUalization:CTLE:DC
:CALCulate:EQUalization:CTLE:POLE1
:CALCulate:EQUalization:CTLE:ZERO1
:CALCulate:EQUalization:FILEname
:CALCulate:EQUalization:STATe
:CALCulate:EQUalization:TYPE
```

Equivalent GUI**Setup > Adv Waveform > Equalization > Pole2 Freq**

E5071C

Eye/Mask > Advanced Waveform (Under **Stimulus) > Equalization > Pole2 Freq**

:CALCulate:EQUalization:CTLE:ZERO1**Type**

Command/Query

Syntax

:CALCulate:EQUalization:CTLE:ZERO1 <numeric>

:CALCulate:EQUalization:CTLE:ZERO1?

Description

This command sets the equalization CTLE (Continuous Time Linear Equalization) zero parameter.

Parameter

Description	CTLE DC zero parameter
Data Type	Double precision floating point type (Double)
Range	0 - 2E10
Preset Value	650E6
Unit	Hz
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EQU:CTLE:ZERO1 0.000650"
20 OUTPUT 717;":CALC:EQU:CTLE:ZERO1?"
30 ENTER 717;A$
```

Related Commands

```
:CALCulate:EQUalization:CTLE:DC
:CALCulate:EQUalization:CTLE:POLE1
:CALCulate:EQUalization:CTLE:POLE2
:CALCulate:EQUalization:FILEname
:CALCulate:EQUalization:STATe
:CALCulate:EQUalization:TYPE
```

Equivalent GUI**Setup > Adv Waveform > Equalization > Zero Freq**

E5071C

Eye/Mask > Advanced Waveform (Under **Stimulus) > Equalization > Zero Freq**

:CALCulate:EQUalization:FILename**Type**

Command/Query

Syntax

:CALCulate:EQUalization:FILename <File>

:CALCulate:EQUalization:FILename?

Description

This command sets the filename of the equalization equation user file. It is necessary to select the file by :CALCulate:EQUalization:TYPE and turn ON the :CALCulate:EQUalization:STATe. This file is saved with .csv extension.

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Parameter

Description	Equalization equation file name (extension: .csv)
Data Type	Character string type (String)
Range	254 characters or less
Preset Value	""

Query Response

{string}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EQU:FIL ""D:\folder\User.csv""

20 OUTPUT 717;":CALC:EQU:FIL?"

30 ENTER 717;A\$

Related Commands

:CALCulate:EQUalization:CTLE:DC

:CALCulate:EQUalization:CTLE:POLE1

:CALCulate:EQUalization:CTLE:POLE2

:CALCulate:EQUalization:CTLE:ZERO1

:CALCulate:EQUalization:STATe

:CALCulate:EQUalization:TYPE

Equivalent GUI**Setup > Adv Waveform > Equalization > Load**

E5071C

Eye/Mask > Advanced Waveform (Under **Stimulus) > Equalization > Load**

:CALCulate:EQUalization:STATe**Type**

Command/Query

Syntax

:CALCulate:EQUalization:STATe {ON|OFF|1|0}

:CALCulate:EQUalization:STATe?

Description

This command sets the equalization function state on/off.

Parameter

Description	Equalization function ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EQU:STAT ON"

20 OUTPUT 717;":CALC:EQU:STAT?"

30 ENTER 717;A\$

Related Commands

:CALCulate:EQUalization:CTLE:DC

:CALCulate:EQUalization:CTLE:POLE1

:CALCulate:EQUalization:CTLE:POLE2

:CALCulate:EQUalization:CTLE:ZERO1

:CALCulate:EQUalization:FILENAME

:CALCulate:EQUalization:TYPE

Equivalent GUI

E5071C

Setup > Adv Waveform > Equalization > Enable

Eye/Mask > Advanced Waveform (Under **Stimulus) > Equalization > Enable**

:CALCulate:EQUalization:TYPE**Type**

Command/Query

Syntax

:CALCulate:EQUalization:TYPE {EQUation|USER}

:CALCulate:EQUalization:TYPE?

Description

This command sets the equalization type.

Parameter

Description	Equalization type
Data Type	Double precision floating point type (Double)
Range	EQUation: Option to enter values for equation calculation USER: Load user's file
Preset Value	EQUation
Unit	-
Resolution	-

Query Response

{EQU|USER}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EQU:TYPE EQU"

20 OUTPUT 717;":CALC:EQU:TYPE?"

30 ENTER 717;A\$

Related Commands

:CALCulate:EQUalization:CTLE:DC

:CALCulate:EQUalization:CTLE:POLE1

:CALCulate:EQUalization:CTLE:POLE2

:CALCulate:EQUalization:CTLE:ZERO1

:CALCulate:EQUalization:FILENAME

:CALCulate:EQUalization:STATE

E5071C

Equivalent GUI

Setup > Adv Waveform > Equalization > Type:Equation|File

Eye/Mask > Advanced Waveform (Under Stimulus) > Equalization > Type:Equation|File

:CALCulate:EYE:ABORt

Type

Command Only

Syntax

:CALCulate:EYE:ABORt

Description

This command aborts calculations for the simulated eye diagram.

From A.01.50, this function is deleted.

Examples

10 OUTPUT 717;":CALC:EYE:ABOR"

Related Commands

:CALCulate:EYE:EXECute

Equivalent Key

Eye/Mask > Abort (Under **Eye Control**) (Only A.01.00)

:CALCulate:EYE:EXECute

Type

Command Only

Syntax

:CALCulate:EYE:EXECute

Description

This command performs the calculation for the simulated eye diagram for the active trace.

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"  
20 OUTPUT 717 ":CALC:ATR:ACT 3"  
30 OUTPUT 717;":CALC:EYE:EXEC"
```

Related Commands

:CALCulate:EYE:STATe

:CALCulate:ATRaces:ACTive

:CALCulate:EYE:ABORt

Equivalent Key

Eye/Mask > Draw Eye (Under **Eye Control**)

:CALCulate:EYE:INPut:BPATtern:LENGth**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:BPATtern:LENGth <numeric>

:CALCulate:EYE:INPut:BPATtern:LENGth?

Description

This command sets bits' power of 2 for PRBS pattern. This value is used only when the bit pattern type is selected at PRBS.

Parameter

Description	Bit length
Data Type	Long integer type (Long)
Range	3 to 15
Preset Value	7
Unit	-
Resolution	2

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE PRBS"

20 OUTPUT 717;":CALC:EYE:INP:BPAT:LENG 3"

Related Commands

:CALCulate:EYE:INPut:BPATtern:TYPE

Equivalent GUI**Eye Mask > Length** (under **Bit Pattern**)

:CALCulate:EYE:INPut:BPATtern:TYPE**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:BPATtern:TYPE {PRBS|K285|USER|STATistical}

:CALCulate:EYE:INPut:BPATtern:TYPE?

Description

This command sets the bit pattern type for the simulated eye function.

Parameter

Description	Bit pattern type
Data Type	Character string type (String)
Range	PRBS: Pseudo-Random Bit Sequence K285: K 28.5 USER: User Custom STAT: Statistical calculation
Preset Value	PRBS
Unit	-
Resolution	-

Query Response

{PRBS|K285|USER|STAT}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE K285"
20 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:EYE:INPut:BPATtern:LENGth

:MMEMory:LOAD:EYE:BPATtern

Equivalent GUI**Eye Mask > Type** (under **Stimulus**)

:CALCulate:EYE:INPut:DRATe**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:DRATe <numeric>

:CALCulate:EYE:INPut:DRATe?

Description

This command sets the bit rate for the simulated eye function.

Parameter

Description	Bit Rate
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	1G
Unit	bits/sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:DRAT 1.1e9"

20 OUTPUT 717;":CALC:EYE:INP:DRAT?"

30 ENTER 717;A

Related Commands

:CALCulate:EYE:INPut:OLEVel

:CALCulate:EYE:INPut:ZLEVel

:CALCulate:EYE:INPut:RTIME:DATA

:CALCulate:EYE:INPut:RTIME:THReshold

Equivalent GUI**Eye/Mask > Data Rate** (Under **Bit Pattern**)

:CALCulate:EYE:INPut:JITTer:DLIMit**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:JITTer:DLIMit <numeric>

:CALCulate:EYE:INPut:JITTer:DLIMit?

Description

This command sets the display limit value.

Parameter

Description	Display Limit
Data Type	Double precision floating point type (Double)
Range	0 to 1
Preset Value	10E-8
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:JTtT:DLIM 10E-10"

Related Commands

:CALCulate:EYE:INPut:JITTer:STATe

:CALCulate:EYE:INPut:JITTer:TYPE

:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency

:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude

:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude

Equivalent GUI**Eye/Mask > Advanced Waveform (Under **Stimulus**) > Jitter > Display Limit**

:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency <numeric>

:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency?

Description

This command sets the periodic jitter frequency. This value is used only when periodic jitter function type is selected.

Parameter

Description	Periodic jitter frequency
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	500E3
Unit	Hz
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:JTT:TYPE PER"

20 OUTPUT 717;":CALC:EYE:INP:JITT:PER:FREQ 10E3"

Related Commands

:CALCulate:EYE:INPut:JITTer:DLIMit

:CALCulate:EYE:INPut:JITTer:STATe

:CALCulate:EYE:INPut:JITTer:TYPE

:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude

:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude

Equivalent GUI

Eye/Mask > Advanced Waveform (Under **Stimulus**) > **Jitter > Frequency**
(Under **Periodic Jitter**)

:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude <numeric>

:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude?

Description

This command sets the periodic jitter magnitude in rms. This value is used only when periodic jitter function type is selected.

Parameter

Description	Periodic jitter magnitude
Data Type	Double precision floating point type (Double)
Range	0 to 1
Preset Value	0
Unit	UI
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:JTT:TYPE PER"

20 OUTPUT 717;":CALC:EYE:INP:JITT:PER:MAGN 0.5"

Related Commands

:CALCulate:EYE:INPut:JITTer:DLIMit

:CALCulate:EYE:INPut:JITTer:STATe

:CALCulate:EYE:INPut:JITTer:TYPE

:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude

:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency

Equivalent GUI

Eye/Mask > Advanced Waveform (Under **Stimulus**) > **Jitter > Magnitude**
(Under **Periodic Jitter**)

:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude <numeric>

:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude?

Description

This command sets the random jitter magnitude in rms. This value is used only when random jitter function type is selected.

Parameter

Description	Random jitter magnitude
Data Type	Double precision floating point type (Double)
Range	0 to 0.25
Preset Value	0
Unit	UI
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:JTT:TYPE RAND"

20 OUTPUT 717;":CALC:EYE:INP:JITT:RAND:MAGN 0.05"

Related Commands

:CALCulate:EYE:INPut:JITTer:DLIMit

:CALCulate:EYE:INPut:JITTer:STATe

:CALCulate:EYE:INPut:JITTer:TYPE

:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency

:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude

Equivalent GUI

Eye/Mask > Advanced Waveform (Under **Stimulus**) > **Jitter > Magnitude (RMS)** (Under **Random Jitter**)

:CALCulate:EYE:INPut:JITTer:STATe

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:JITTer:STATe {ON|OFF|1|0}
:CALCulate:EYE:INPut:JITTer:STATe?

Description

This command sets the jitter function state with simulated eye on/off.

Parameter

Description	Jitter function ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":MMEM:LOAD:EYE:MASK ""D:\TDR\FC0133.msk""
40 OUTPUT 717;":CALC:EYE:INP:JITT:STAT ON"
50 OUTPUT 717;":CALC:EYE:EXEC"

Related Commands

:CALCulate:EYE:INPut:JITTer:DLIMit
:CALCulate:EYE:INPut:JITTer:TYPE
:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude
:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency
:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude

Equivalent GUI

Eye/Mask > **Advanced Waveform** (Under **Stimulus**) > **Jitter** > **Enable**
checkbox

:CALCulate:EYE:INPut:JITTer:TYPE**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:JITTer:TYPE {RANDom|PERiodic}

:CALCulate:EYE:INPut:JITTer:TYPE?

Description

This command sets the jitter function type for the simulated eye function.

Parameter

Description	Jitter function type
Data Type	Character string type (String)
Range	RANDom: Random jitter function PERiodic: Periodic jitter function
Preset Value	PERiodic
Unit	-
Resolution	-

Query Response

{RAND|PER}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:JTT:TYPE RAND"
20 OUTPUT 717;":CALC:EYE:INP:JITT:TYPE?"
30 ENTER 717;A$
```

Related Commands

```
:CALCulate:EYE:INPut:JITTer:DLIMit
:CALCulate:EYE:INPut:JITTer:STATe
:CALCulate:EYE:INPut:JITTer:RANDom:MAGNitude
:CALCulate:EYE:INPut:JITTer:PERiodic:FREQuency
:CALCulate:EYE:INPut:JITTer:PERiodic:MAGNitude
```

Equivalent GUI

Eye/Mask > **Advanced Waveform** (Under **Stimulus**) > **Random|Periodic** radio dial (Under **Type**)

:CALCulate:EYE:INPut:OLEVel**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:OLEVel <numeric>

:CALCulate:EYE:INPut:OLEVel?

Description

This command sets the voltage level for bit "1" for the simulated eye function.

Parameter

Description	Level for bit "1"
Data Type	Double precision floating point type (Double)
Range	-5 to +5
Preset Value	200m
Unit	volt
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:OLEV 100e-3"
20 OUTPUT 717;":CALC:EYE:INP:ZLEV -100e-3"
30 OUTPUT 717;":CALC:EYE:INP:OLEV?"
40 ENTER 717;A
50 OUTPUT 717;":CALC:EYE:INP:ZLEV?"
60 ENTER 717;B
```

Related Commands

:CALCulate:EYE:INPut:ZLEVel

:CALCulate:EYE:INPut:OFFSet

Equivalent GUI**Eye Mask > One Lv. (Under Bit Pattern)**

:CALCulate:EYE:INPut:RTIMe:DATA

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:RTIMe:DATA <numeric>

:CALCulate:EYE:INPut:RTIMe:DATA?

Description

This command sets the rise time value for the simulated eye function.

Parameter

Description	Rise time
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	35p
Unit	second
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:INP:RTIM:DATA 90e-12"
20 OUTPUT 717;":CALC:EYE:INP:RTIM:DATA?"
30 ENTER 717;A

Related Commands

:CALCulate:EYE:INPut:RTIMe:THReshold

Equivalent GUI

Eye Mask > Rise Time (Under **Bit Pattern**)

:CALCulate:EYE:INPut:RTIME:THReshold**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:RTIME:THReshold {T1_9|T2_8}

:CALCulate:EYE:INPut:RTIME:THReshold?

Description

This command sets the rise time threshold for the simulated eye.

Parameter

Description	Threshold levels define rise time
Data Type	Character string type (String)
Range	T1_9: 10%-90% T2_8: 20%-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:RTIM:THR T1_9"
20 OUTPUT 717;":CALC:EYE:INP:RTIM:THR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:EYE:INPut:RTIME:DATA

Equivalent GUI

Eye Mask > Rise Time (Under **Bit Pattern**)

:CALCulate:EYE:INPut:ZLEVel**Type**

Command/Query

Syntax

:CALCulate:EYE:INPut:ZLEVel <numeric>

:CALCulate:EYE:INPut:ZLEVel?

Description

This command sets the voltage level for bit "0" for the simulated eye function.

Parameter

Description	level for bit "0"
Data Type	Double precision floating point type (Double)
Range	-5 to 5
Preset Value	0
Unit	volt
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:OLEV 100e-3"
20 OUTPUT 717;":CALC:EYE:INP:ZLEV -100e-3"
30 OUTPUT 717;":CALC:EYE:INP:OLEV?"
40 ENTER 717;A
50 OUTPUT 717;":CALC:EYE:INP:ZLEV?"
60 ENTER 717;B
```

Related Commands

:CALCulate:EYE:INPut:OLEVel

:CALCulate:EYE:INPut:OFFSet

Equivalent GUI**Eye Mask > Zero Lv. (Under Bit Pattern)**

:CALCulate:EYE:MASK:FAIL**Type**

Query Only

Syntax

:CALCulate:EYE:MASK:FAIL?

Description

This command returns the mask test result.

Parameter

Description	Mask test result
Data Type	Boolean type (Boolean)
Range	1: Mask Test Fail 0: Mask Test Pass
Preset Value	-
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"  
20 OUTPUT 717 ":CALC:ATR:ACT 3"  
30 OUTPUT 717;":MMEM:LOAD:EYE:MASK ""D:\TDR\FC0133.msk""  
40 OUTPUT 717;":CALC:EYE:MASK:STAT ON"  
50 OUTPUT 717;":CALC:EYE:EXEC"  
60 OUTPUT 717;":CALC:EYE:MASK:FAIL?"  
70 ENTER 717;A
```

Related Commands

:CALCulate:EYE:MASK:STATe

Equivalent GUI**Eye/Mask** > Mask Test Result in windows

:CALCulate:EYE:MASK:STATe

Type

Command/Query

Syntax

:CALCulate:EYE:MASK:STATe {ON|OFF|1|0}

:CALCulate:EYE:MASK:STATe?

Description

This command sets mask test with simulated eye on/off.

Parameter

Description	Mask Test ON/OFF
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":MMEM:LOAD:EYE:MASK ""D:\TDR\FC0133.msk""
40 OUTPUT 717;":CALC:EYE:MASK:STAT ON"
50 OUTPUT 717;":CALC:EYE:EXEC"

Related Commands

:CALCulate:EYE:MASK:STATe
:CALCulate:EYE:EXECute
:CALCulate:EYE:MASK:FAIL
:MMEMory:LOAD:EYE:MASK

Equivalent GUI

Eye Mask > Mask Test (Under Result)

:CALCulate:EYE:RESults:DATA**Type**

Query Only

Syntax

:CALCulate:EYE:RESults:DATA?

Description

This command returns the results of eye measurement. The 18 values are returned. The minimum and maximum values are returned in addition of the displayed results (16 values) on the TDR application GUI.

Parameter

Description	Results of eye measurement
Data Type	Variant type Array (Range)
Range	-
Preset Value	-
Unit	-
Resolution	-

Query Response

```
{numeric (Minimum Value)},{numeric (Maximum Value)},{numeric (Level
Zero)},{numeric (Level One)}, {numeric (Level Mean)},{numeric
(Amplitude)},{numeric (Height)},{numeric (always 0)}, {numeric
(Width)},{numeric (Opening Factor)}, {numeric (Signal/Noise Ratio)},
{numeric (Duty Cycle Distortion)},{numeric (Duty Cycle Distortion-
%)},{numeric (Rise Time)},{numeric (Fall Time)},{numeric (Jitter peak to
peak)},{numeric (Jitter RMS)},{numeric (Crossing
Percentage)}<newline><^END>
```

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":CALC:EYE:RES:THR T2_8"
30 OUTPUT 717;":CALC:EYE:EXEC"
40 OUTPUT 717;":CALCulate:EYE:RESults:DATA?"
50 ENTER 717;A(*)
```

Related Commands

:CALCulate:EYE:RESults:THReshold

E5071C

:CALCulate:EYE:EXECute

Equivalent GUI

Eye Mask > Result Window (Under **Result**)

Eye Mask > **Export to CSV** (Under **Result**)

(Not exactly equivalent)

:CALCulate:EYE:RESults:DISPlay:STATe**Type**

Command/Query

Syntax**:CALCulate:EYE:RESults:DISPlay:STATe {ON|OFF|1|0}****:CALCulate:EYE:RESults:DISPlay:STATe?****Description**

This command turns the overlay on/off.

Parameter

Description	Overlay on/off
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:STAT ON"

20 OUTPUT 717 ":CALC:ATR:ACT 3"

30 OUTPUT 717;":CALC:EYE:EXEC"

40 OUTPUT 717;":CALC:EYE:RES:DISP:STAT OFF"

Related Commands**:CALCulate:EYE:RESults:DATA****Equivalent GUI****Eye Mask > Overlay (Under Result)**

:CALCulate:EYE:RESults:THReshold

Type

Command/Query

Syntax

:CALCulate:EYE:RESults:THReshold {T1_9|T2_8}
:CALCulate:EYE:RESults:THReshold?

Description

This command sets the rise time threshold level for the results of eye measurement.

Parameter

Description	Rise time threshold level
Data Type	Character string type (String)
Range	T1_9: 10-90% T2_8: 20-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

10 OUTPUT 717;":CALC:EYE:RES:THR T1_9"
20 OUTPUT 717;":CALC:EYE:RES:THR?"
30 ENTER 717;A\$

Related Commands

:CALCulate:EYE:RESults:DATA

Equivalent GUI

Eye Mask > Rise Time Def. (Under **Result**)

:CALCulate:EYE:STATe**Type**

Command/Query

Syntax

:CALCulate:EYE:STATe {ON|OFF|1|0}

:CALCulate:EYE:STATe?

Description

This command displays the EYE/MASK window.

Parameter

Description	Status of Eye/Mask window
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

See the example in :CALCulate:EYE:EXECute

Related Commands

:CALCulate:EYE:EXECute

Equivalent GUI**Eye Mask**

:CALCulate:TRACe{Tr}:AMARkers:ACTive**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:AMARkers:ACTive <numeric>

:CALCulate:TRACe{Tr}:AMARkers:ACTive?

Description

This command sets active marker number.

Parameter

Description	Marker number to active
Data Type	Long integer type (Long)
Range	0 to 10
Preset Value	0
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:AMAR:ACT 1"

20 OUTPUT 717;":CALC:TRAC1:AMAR:ACT?"

30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCTion:EXECute

Equivalent GUI**Marker** > Select the marker number OR**Marker** > Select the marker number

:CALCulate:TRACe{Tr}:CONVersion:PEELing:STATe**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:CONVersion:PEELing:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:CONVersion:PEELing:STATe?

Description

This command sets state for the peeling function.

Parameter

Description	Peeling function on/off.
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:CONV:PEEL:STAT ON"  
20 OUTPUT 717;":CALC:TRAC1:CONV:PEEL:STAT?"  
30 ENTER 717;A
```

Equivalent GUI**TDR/TDT > Parameters > Peeling**

:CALCulate:TRACe{Tr}:DTIME:DATA

Type

Query Only

Syntax

:CALCulate:TRACe{Tr}:DTIME:DATA?

Description

This command gets delta time result value. You can get the result even if :CALCulate:TRACe{Tr}:DTIME:STATE is off.

Parameter

Description	Value of delta time result.
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:DTIM:TARG 5"
20 OUTPUT 717;":CALC:TRAC1:DTIM:POS 50"
30 OUTPUT 717;":CALC:TRAC1:DTIM:DATA?"
40 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:DTIME:POSition
:CALCulate:TRACe{Tr}:DTIME:STATe
:CALCulate:TRACe{Tr}:DTIME:TARGet

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:DTIME:POSition**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:DTIME:POSition <numeric>

:CALCulate:TRACe{Tr}:DTIME:POSition?

Description

This command sets delta time reference position.

Parameter

Description	Delta time reference position.
Data Type	Double precision floating point type (Double)
Range	0 to 100
Preset Value	50
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:DTIM:POS 0"

20 OUTPUT 717;":CALC:TRAC1:DTIM:POS?"

30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:DTIME:DATA

:CALCulate:TRACe{Tr}:DTIME:STATe

:CALCulate:TRACe{Tr}:DTIME:TARGet

Equivalent GUI**Marker Search > Δ Time > Position(%)** (In Δ Time Dialog box)

:CALCulate:TRACe{Tr}:DTIME:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:DTIME:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:DTIME:STATe?

Description

This command displays the delta time marker in the marker search..

Parameter

Description	Status of display of delta time marker
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0} <newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:DTIM:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:DTIM:STAT?"
30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:DTIME:DATA
:CALCulate:TRACe{Tr}:DTIME:POSition
:CALCulate:TRACe{Tr}:DTIME:TARGet

Equivalent GUI

Marker Search > Δ Time > Δ Time (Checkbox)

:CALCulate:TRACe{Tr}:DTIMe:TARGet**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:DTIMe:TARGet <numeric>

:CALCulate:TRACe{Tr}:DTIMe:TARGet?

Description

This command sets target trace number for the delta time function. The {Tr} is the trace number starting point for delta time. The <numeric> is the trace number stopping point for delta time.

Parameter

Description	Target trace number for the delta time function.
Data Type	Long integer type (Long)
Range	1 to 16
Preset Value	1
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:DTIM:TARG 5"
20 OUTPUT 717;":CALC:TRAC1:DTIM:TARG?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:DTIMe:DATA

:CALCulate:TRACe{Tr}:DTIMe:POSition

:CALCulate:TRACe{Tr}:DTIMe:STATe

Equivalent GUI**Marker Search > Δ Time > trace selection (In Δ Time Dialog box)**

:CALCulate:TRACe{Tr}:FORMat**Type**

Command/Query

Syntax

```
:CALCulate:TRACe{Tr}:FORMat
{IMPedance|VOLT|MLOGarithmic|MLINear|REAL|IMAGinary|GDElay|SWR|
PHASe|UPHase|PPHase|SLINear|SLOGarithmic|SCOMplex|SMITH|SADMittance|PLINear|PLOGarithmic|POLar}

:CALCulate:TRACe{Tr}:FORMat?
```

Description

This command sets trace format.

Parameter

Description	Format for the trace.
Data Type	Character string type (String)
Range	<p>For S-Parameter Measurement: IMPedance MLOGarithmic MLINear REAL IMAGinary GDElay SWR PHASe UPHase PPHase SLINear SLOGarithmic SCOMplex SMITH SADMittance PLINear PLOGarithmic POLar</p> <p>For Time Domain measurement: IMPedance VOLT MLOGarithmic MLINear REAL</p>
Preset Value	Depending on trace.
Unit	-
Resolution	-

Query Response

{IMPedance|VOLT|MLOGarithmic|MLINear|REAL|IMAGinary|GDElay|SWR|PHASe|UPHase|PPHase|SLINear|SLOGarithmic|SCOMplex|SMITH|SADMittance|PLINear|PLOGarithmic|POLar}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:FORM IMP"
20 OUTPUT 717;":CALC:TRAC1:FORM?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:PARAmeter

Equivalent Key

TDR/TDT > Parameters

:CALCulate:TRACe{Tr}:GATE:START**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:START <numeric>

:CALCulate:TRACe{Tr}:GATE:START?

Description

This command sets start value for the gating function.

Parameter

Description	Starting point of the gate
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
 20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
 30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
 40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"

50 OUTPUT 717;":CALC:TRAC1:GATE:STAR?"
 60 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:GATE:STOP
 :CALCulate:TRACe{Tr}:GATE:TYPE
 :CALCulate:TRACe{Tr}:GATE:STATE

Equivalent GUITDR/TDT > **Gating** > **Start** virtual knob OR

TDR/TDT > Gating > type in the text box under the **Start** virtual knob OR
Right-click on the time domain graph plot and select **Set Gating Range**

:CALCulate:TRACe{Tr}:GATE:STATe**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:GATE:STATe?

Description

This command sets state for the gating function.

Parameter

Description	Activating the gating function
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"

20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"

30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"

40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"

50 OUTPUT 717;":CALC:TRAC1:GATE:STAT?"

60 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:GATE:START

:CALCulate:TRACe{Tr}:GATE:STOP

:CALCulate:TRACe{Tr}:GATE:TYPE

Equivalent GUI**TDR/TDT > Gating > select the **Gating** check box OR**

Right-click on the time domain active trace and select **Switch Gating State**

:CALCulate:TRACe{Tr}:GATE:STOP**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:STOP <numeric>

:CALCulate:TRACe{Tr}:GATE:STOP?

Description

This command sets stop value for the gating function.

Parameter

Description	Stopping point of the gate
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```

10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"
50 OUTPUT 717;":CALC:TRAC1:GATE:STOP?"
60 ENTER 717;A

```

Related Commands

:CALCulate:TRACe{Tr}:GATE:STARt

:CALCulate:TRACe{Tr}:GATE:TYPE

:CALCulate:TRACe{Tr}:GATE:STATe

Equivalent GUI**TDR/TDT** > **Gating** > **Stop** virtual knob OR**TDR/TDT** > **Gating** > type in the text box under the **Stop** virtual knob ORRight-click on the time domain graph plot and select **Set Gating Range**

:CALCulate:TRACe{Tr}:GATE:TYPE**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:TYPE {BPASs|NOTCh}

:CALCulate:TRACe{Tr}:GATE:TYPE?

Description

This command sets gate type for the gating function

Parameter

Description	Gate type for the gating function
Data Type	Character string type (String)
Range	BPASs: Band Pass NOTCh: Notch
Preset Value	BPASs
Unit	-
Resolution	-

Query Response

{BPASs|NOTCh}<newline><^END>

Examples

```

10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"
50 OUTPUT 717;":CALC:TRAC1:GATE:TYPE?"
60 ENTER 717;A

```

Related Commands

:CALCulate:TRACe{Tr}:GATE:START

:CALCulate:TRACe{Tr}:GATE:STOP

:CALCulate:TRACe{Tr}:GATE:STATE

Equivalent GUI**TDR/TDT > Gating > Notch OR Bandpass (under Type)**

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:EXECute

Type

Command Only

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:EXECute

Description

This command executes marker Max or Min for a selected marker function. This is different than tracking mode because moving the marker to another point, the marker does not point back to Max or Min that is previously set.

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE MAX"  
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:EXEC"
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TRACking

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TYPE

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TRACking**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TRACking {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TRACking?

Description

This command sets state for the marker tracking mode. Using this command makes the marker constantly pointing to the point whereby the marker is either set to Max or Min and is not able to move to other desire point.

Parameter

Description	Status for the marker tracking mode
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE MAX"
20 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TRAC ON"
30 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TRAC?
40 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:EXECute

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TYPE

Equivalent GUI**Marker Search > Max or Min**

(:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TYPE is used together when equivalent GUI is clicked)

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCTion:TYPE

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCTion:TYPE
{MAXimum|MINimum}

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCTion:TYPE?

Description

This command sets type for the marker function. The equivalent GUI use the tracking mode command. Alternatively, execute command can be used.

Parameter

Description	Marker function type
Data Type	Character string type (String)
Range	MAX: The marker moves to the maximum value of the trace. MIN: The marker moves to the minimum value of the trace.
Preset Value	MAXimum
Unit	-
Resolution	-

Query Response

{MAXimum|MINimum}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE MAX"
20 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TRAC ON"
30 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE?"
40 ENTER 717;A\$

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCTion:EXECute
:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCTion:TRACking

Equivalent GUI

Marker Search > Max or Min

(:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNction:TRACking is used together when equivalent GUI is clicked)

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe?

Description

This command sets state for the reference marker display.

Parameter

Description	Status of the reference marker display
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:MARK1:REF:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:MARK1:REF:STAT?"
30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe

Equivalent GUI

Marker > Ref OR

Setup > Advanced Mode > Marker > Marker Ref

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe?

Description

This command sets state for the marker display.

Parameter

Description	Status for the marker display
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:MARK1:STAT ON"

20 OUTPUT 717;":CALC:TRAC1:MARK1:STAT?"

30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe

Equivalent GUI**Marker OR****Setup > Advanced Mode > Marker**

:CALCulate:TRACe{Tr}:MARKer{Mk}:X

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:X <numeric>

:CALCulate:TRACe{Tr}:MARKer{Mk}:X?

Description

This command sets stimulus value of the marker.

Parameter

Description	Marker stimulus value (X-Axis)
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:MARK1:X 50"
20 OUTPUT 717;":CALC:TRAC1:MARK1:X?"
30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:Y

Equivalent GUI

Setup > Advanced Mode > Marker > Select Marker (Entry box above the graph plot)

:CALCulate:TRACe{Tr}:MARKer{Mk}:Y**Type**

Query Only

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:Y?

Description

This command gets response values of the marker.

Parameter

Description	Marker response value (Y-axis)
Data Type	Double precision floating point type (Double)
Range	- -
Preset Value	- -
Unit	- -
Resolution	- -

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:Y?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:X

Equivalent GUI

No equivalent GUI is available.

E5071C

:CALCulate:TRACe{Tr}:MATH:MEMorize

Type

Command Only

Syntax

:CALCulate:TRACe{Tr}:MATH:MEMorize

Description

This command copies measurement data to memory trace.

Examples

10 OUTPUT 717;":CALC:TRAC1:MATH:MEM"

Related Commands

:DISPlay:TRACe{Tr}:DMEMory:TYPE

Equivalent GUI

Data Mem > Data -> Mem

:CALCulate:TRACe{Tr}:PARAmeter**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:PARAmeter {String}

:CALCulate:TRACe{Tr}:PARAmeter?

Description

This command sets measurement parameter. See Measurement Parameters of Balanced Devices for balanced parameters.

Parameter

Description	Measurement parameter
Data Type	Character string type (String)
Range	<p>For Time Domain</p> <ul style="list-style-type: none"> • Txy • Tddxy • Tdcxy • Tcdxy • Tccxy <p>For S-parameter</p> <ul style="list-style-type: none"> • Sxy • Sddxy • Sdcxy • Scdxy • Sccxy <p>x: 1 to 4 y: 1 to 4</p>
Preset Value	Depending on Trace and Selected DUT topology
Unit	-
Resolution	-

Query Response

E5071C

{String}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:PAR T11"  
20 OUTPUT 717;":CALC:TRAC1:PAR?"  
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:FORMat

Equivalent GUI

TDR/TDT > Parameter

:CALCulate:TRACe{Tr}:SMOothing:STATe**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:SMOothing:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:SMOothing:STATe?

Description

This command sets state for smoothing.

Parameter

Description	Status for smoothing
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:SMO:STAT ON"

20 OUTPUT 717;":CALC:TRAC1:SMO:STAT?"

30 ENTER 717;A

Equivalent GUI**TDR/TDT > Parameters > Smoothing**

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh <numeric>

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh?

Description

This command sets impulse width value for the transform function.

Parameter

Description	Transform function impulse width value
Data Type	Double precision floating point type (Double)
Range	30.2p to 20.8n
Preset Value	35p
Unit	sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:TIME:IMP:WIDT 17e-12"
20 OUTPUT 717;":CALC:TRAC1:TIME:IMP:WIDT?"
30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:TIME:STIMulus

Equivalent GUI

TDR/TDT > Parameters > Lowpass Impulse (In Stimulus)

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:DATA**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:DATA <numeric>

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:DATA?

Description

This command sets rise time value for the transform function.

Parameter

Description	Transform function rise time value
Data Type	Double precision floating point type (Double)
Range	15.9p to 20.8n
Preset Value	35p
Unit	sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:DATA 0"

20 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:DATA?"

30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:THReshold

:CALCulate:TRACe{Tr}:TIME:STIMulus

Equivalent GUI**TDR/TDT > Parameters > Entry box (In Rise Time)**

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:THReshold**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:THReshold {T1_9|T2_8}

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:THReshold?

Description

This command sets rise time threshold for the transform function.

Parameter

Description	Transform function rise time threshold
Data Type	Character string type (String)
Range	T1_9 : 10-90% T2_8 : 20-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:THR T1_9"
20 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:THR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:DATA

:CALCulate:TRACe{Tr}:TIME:STIMulus

Equivalent GUI**TDR/TDT > Parameters > 10-90% or 20-80% (In Stimulus)**

:CALCulate:TRACe{Tr}:TIME:STIMulus**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:STIMulus {LPSTep|LPIMpulse}

:CALCulate:TRACe{Tr}:TIME:STIMulus?

Description

This command sets stimulus type for the transform function.

Parameter

Description	Transform function stimulus type
Data Type	Character string type (String)
Range	LPSTep: Lowpass Step LPIMpulse: Lowpass Impulse
Preset Value	LPSTep
Unit	-
Resolution	-

Query Response

{LPSTep|LPIMpulse}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TIME:STIM LPSTep"
20 OUTPUT 717;":CALC:TRAC1:TIME:STIM?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:DATA

:CALCulate:TRACe{Tr}:TIME:STEP:RTIME:THReshold

Equivalent GUI**TDR/TDT > Parameters > Lowpass Step or Lowpass Impulse (In Stimulus)**

:CALCulate:TRACe{Tr}:TTIMe:DATA

Type

Query Only

Syntax

:CALCulate:TRACe{Tr}:TTIMe:DATA?

Description

This command returns the rise time result value for marker search. You can get the data even if CALCulate:TRACe{Tr}:TTIMe:STATe is off.

Parameter

Description	Value of transition time result.
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:TTIM:DATA?"
20 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:TTIMe:STATe
:CALCulate:TRACe{Tr}:TTIMe:THReshold

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:TTIMe:STATe**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TTIMe:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:TTIMe:STATe?

Description

This command displays the rise time marker.

Parameter

Description	Status of display of rise time marker.
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0} <newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:TTIM:STAT ON"

20 OUTPUT 717;":CALC:TRAC1:TTIM:STAT?"

30 ENTER 717;A

Related Commands

:CALCulate:TRACe{Tr}:TTIMe:DATA

:CALCulate:TRACe{Tr}:TTIMe:THReshold

Equivalent GUI**Marker Search > Rise Time (10-90%) or Rise Time (20-80%)**

:CALCulate:TRACe{Tr}:TTIMe:THReshold**Type**

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TTIMe:THReshold {T1_9|T2_8}

:CALCulate:TRACe{Tr}:TTIMe:THReshold?

Description

This command sets the rise time threshold for the rise time in the marker search function.

Parameter

Description	Transition time function rise time threshold
Data Type	Character string type (String)
Range	T1_9 : 10-90% T2_8 : 20-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

10 OUTPUT 717;":CALC:TRAC1:TTIM:THR T1_9"

20 OUTPUT 717;":CALC:TRAC1:TTIM:THR?"

30 ENTER 717;A\$

Related Commands

:CALCulate:TRACe{Tr}:TTIMe:DATA

:CALCulate:TRACe{Tr}:TTIMe:STATe

Equivalent GUI**Marker Search > Rise Time (10-90%) or Rise Time (20-80%)**

DISPLAY**:DISPlay:ATRaces:SCALe:AUTO****Type**

Command Only

Syntax**:DISPlay:ATRaces:SCALe:AUTO****Description**

This command executes auto scale for all traces.

Examples

10 OUTPUT 717;":DISP:ATR:SCAL:AUTO"

Related Commands**:DISPlay:TRACe{Tr}:X:SCALe:AUTO****:DISPlay:TRACe{Tr}:Y:SCALe:AUTO****Equivalent GUI****Auto Scale > All Traces** ORRight-click on the graph plot > **Auto Scale All**

:DISPlay:ATRaces:SCALe:RPOSition

Type

Command/Query

Syntax



:DISPlay:ATRaces:SCALe:RPOSition {LEFT|CENTer}

:DISPlay:ATRaces:SCALe:RPOSition?

Description

This command sets x-axis reference position for the time domain measurement.

Parameter

Description	Sets x-axis reference position for the time domain measurement
Data Type	Character string type (String)
Range	LEFT: The reference position is left edge  CENTer: The reference position is center 
Preset Value	LEFT
Unit	-
Resolution	-

Query Response

{LEFT|CENT}<newline><^END>

Examples

10 OUTPUT 717;":DISP:ATR:SCAL:RPOS LEFT"
20 OUTPUT 717;":DISP:ATR:SCAL:RPOS?"
30 ENTER 717;A\$

Equivalent GUI

TDR/TDT >  or  (Under **Horizontal**)

:DISPlay:ATRaces:VIEW**Type**

Command/Query

Syntax

:DISPlay:ATRaces:VIEW STIMulus

:DISPlay:ATRaces:VIEW?

Description

This command selects the view point for waveform analysis either before or after DUT.

Parameter

Description	Selects waveform analysis view
Data Type	Character string type (String)
Range	STIMulus: Stimulus view, observation point before DUT RESPonse: Response view, observation point after DUT
Preset Value	RESPonse
Unit	-
Resolution	-

Query Response

{STIM|RESP}<newline><^END>

Examples

10 OUTPUT 717;":DISP:ATR:VIEW STIM"

20 OUTPUT 717;":DISP:ATR:VIEW?"

30 ENTER 717;A\$

Equivalent GUI**Eye/Mask > Advanced Waveform (Under Stimulus)**

E5071C

:DISPlay:EYE:Y:SCALE:AUTO

Type

Command Only

Syntax

:DISPlay:EYE:Y:SCALE:AUTO

Description

This command executes Y axis auto scaling for eye diagram.

Examples

10 OUTPUT 717;":DISP:EYE:Y:SCAL:AUTO"

Related Commands

:DISPlay:EYE:Y:SCALE:PDIVision

:DISPlay:EYE:Y:SCALE:RLEVel

Equivalent GUI

Eye/Mask > Scale/Mask > Auto Scale radio button

:DISPlay:EYE:Y:SCALE:PDIVision**Type**

Command/Query

Syntax

:DISPlay:EYE:Y:SCALE:PDIVision <numeric>

:DISPlay:EYE:Y:SCALE:PDIVision?

Description

This command sets value of y-axis scale per division for eye diagram. The scale mode should be set at manual by the :DISPlay:EYE:Y:SCALE:MANual before this command is executed.

Parameter

Description	Value of eye diagram y-axis scale per division
Data Type	Double precision floating point type (Double)
Range	1E-18 to 5
Preset Value	200m
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:EYE:Y:SCAL:MAN"
20 OUTPUT 717;":DISP:EYE:Y:SCAL:PDIV 300E-03"
30 OUTPUT 717;":DISP:EYE:Y:SCAL:PDIV?"
40 ENTER 717;A$
```

Related Commands

:DISPlay:EYE:Y:SCALE:MANual

:DISPlay:EYE:Y:SCALE:RLEVel

:DISPlay:EYE:Y:SCALE:AUTO

Equivalent GUI**Eye/Mask > Scale/Mask > Manual radio button > Scale / Div**

:DISPlay:EYE:Y:SCALe:RLEVel**Type**

Command/Query

Syntax

:DISPlay:EYE:Y:SCALe:RLEVel <numeric>

:DISPlay:EYE:Y:SCALe:RLEVel?

Description

This command sets value of eye diagram y-axis reference line. The scale mode should be set at manual by the :DISPlay:EYE:Y:SCALe:MANual before this command is executed.

Parameter

Description	Value of eye diagram y-axis reference line
Data Type	Double precision floating point type (Double)
Range	-5 to 5
Preset Value	0
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:EYE:Y:SCAL:MAN"
20 OUTPUT 717;":DISP:EYE:Y:SCAL:RLEV 0.01"
30 OUTPUT 717;":DISP:EYE:Y:SCAL:RLEV?"
40 ENTER 717;A
```

Related Commands

:DISPlay:EYE:Y:SCALe:MANual

:DISPlay:EYE:Y:SCALe:PDIVision

:DISPlay:EYE:Y:SCALe:AUTO

Equivalent GUI**Eye/Mask > Scale/Mask > Manual radio button > Offset**

:DISPlay:IMAGe**Type**

Command/Query

Syntax

:DISPlay:IMAGe {NORMal|INVert}

:DISPlay:IMAGe?

Description

This command changes the back ground color of screen.

Parameter

Description	Back ground color
Data Type	Character string type (String)
Range	NORMal: Black background color INVert: While background color
Preset Value	NORMal
Unit	-
Resolution	-

Query Response

{NORMal|INVert}<newline><^END>

Examples

10 OUTPUT 717;":DISPlay:IMAGe NORMal"

20 OUTPUT 717;":DISPlay:IMAGe?"

30 ENTER 717;A\$

Related Commands

:MMEMory:STORe:IMAGe

Equivalent GUI**File > Invert Color**

:DISPlay:TRACe{Tr}:DMEMory:TYPE**Type**

Command/Query

Syntax

:DISPlay:TRACe{Tr}:DMEMory:TYPE {OFF|DATA|MEMory|DMEMory}

:DISPlay:TRACe{Tr}:DMEMory:TYPE?

Description

This command sets type of data/memory display.

Parameter

Description	Sets the display to off, data type, memory type or data and memory type
Data Type	Character string type (String)
Range	OFF: Nothing is displayed on the graph plot. DATA: Data only is displayed on the graph plot. MEMory: Memory only is displayed on the graph plot. DMEMory: Data and Memory are displayed on the graph plot.
Preset Value	DATA
Unit	-
Resolution	-

Query Response

{OFF|DATA|MEMory|DMEMory}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:TRAC1:DMEM:TYPE OFF"
20 OUTPUT 717;":DISP:TRAC1:DMEM:TYPE?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:MATH:MEMorize

Equivalent GUI**Data Mem** > Select **Off**, **Data**, **Memory** or **Data & Memory**.

:DISPlay:TRACe{Tr}:X:SCALe:AUTO**Type**

Command Only

Syntax

:DISPlay:TRACe{Tr}:X:SCALe:AUTO

Description

This command executes x-axis auto scaling.

Examples

10 OUTPUT 717;":DISP:TRAC1:X:SCAL:AUTO"

Related Commands

:DISPlay:ATRAces:SCALe:AUTO

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Equivalent GUI

Auto Scale > X OR

Right-click on the graph plot > **Auto Scale X**

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision**Type**

Command/Query

Syntax

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision <numeric>

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision?

Description

This command sets value of x-axis scale per division.

Parameter

Description	X-axis scale per division
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	2n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":DISP:TRAC1:X:SCAL:PDIV 1e-9"

20 OUTPUT 717;":DISP:TRAC1:X:SCAL:PDIV?"

30 ENTER 717;A

Related Commands

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision

Equivalent GUI**TDR/TDT** > input box under the left knob (Under **Horizontal**)

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel**Type**

Command/Query

Syntax

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel <numeric>

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel?

Description

This command sets value of x-axis reference line.

Parameter

Description	Value at X-axis reference line
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":DISP:TRAC1:X:SCAL:RLEV 20e-9"

20 OUTPUT 717;":DISP:TRAC1:X:SCAL:RLEV?"

30 ENTER 717;A

Related Commands

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel

Equivalent GUI**TDR/TDT** > input box under the right knob (Under **Horizontal**)

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Type

Command Only

Syntax

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Description

This command executes y-axis auto scaling.

Examples

10 OUTPUT 717;":DISP:TRAC1:Y:SCAL:AUTO"

Related Commands

:DISPlay:ATRaces:SCALe:AUTO

:DISPlay:TRACe{Tr}:X:SCALe:AUTO

Equivalent GUI

Auto Scale > Y OR

Right-click on the graph plot > **Auto Scale Y**

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision**Type**

Command/Query

Syntax

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision <numeric>

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision?

Description

This command sets value of Y-axis scale per division.

Parameter

Description	Y-axis scale per division value.
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	20
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":DISP:TRAC1:Y:SCAL:PDIV 20"
 20 OUTPUT 717;":DISP:TRAC1:Y:SCAL:PDIV?"
 30 ENTER 717;A

Related Commands

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision

Equivalent GUI**TDR/TDT** > input box under the left knob (Under **Vertical**)

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel

Type

Command/Query

Syntax

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel <numeric>

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel?

Description

This command sets value of y-axis reference line.

Parameter

Description	Value at Y-axis reference line
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	100
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":DISP:TRAC1:Y:SCAL:RLEV 0.01"

20 OUTPUT 717;":DISP:TRAC1:Y:SCAL:RLEV?"

30 ENTER 717;A

Related Commands

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel

Equivalent GUI

TDR/TDT > input box under the right knob (Under **Vertical**)

IEEE

*CLS

Type

Command Only

Syntax

*CLS

Description

This command clears the status.

Examples

```
10 OUTPUT 717;"*CLS"
```

Equivalent GUI

No equivalent GUI is available.

E5071C

*IDN

Type

Query Only

Syntax

*IDN?

Description

This command returns the product information and revision number of E5071C-TDR application.

Parameter

Description	Gets model information
Data Type	Character string type (String)
Range	-
Preset Value	"Agilent Technologies,ENA-TDR,-----,<revision number>"
Unit	-
Resolution	-

Query Response

{String}<newline><^END>

Examples

```
10 OUTPUT 717;"*IDN?"  
20 ENTER 717;A$
```

Equivalent GUI

E5071C OPT. TDR logo at the upper left corner on GUI.

OPC*Type**

Query Only

Syntax

*OPC?

Description

This command detects the operation complete.

Parameter

Description	Operation complete
Data Type	Long integer type (Long)
Range	1 is returned when all pending operations are complete
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;"*OPC?"  
20 ENTER 717;A
```

Equivalent GUI

No equivalent GUI is available.

MMEMORY**:MMEMory:LOAD:EYE:BPATtern****Type**

Command Only

Syntax**:MMEMory:LOAD:EYE:BPATtern <string>****Description**

This command loads the specified user bit pattern file. The extension of file should be .txt. The bit pattern editing is not available through the command.

Parameter

Description	File name of user bit pattern (.txt)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":MMEM:LOAD:EYE:BPAT ""D:\TDR\userbit.txt""
20 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE USER"
```

Related Commands**:CALCulate:EYE:INPut:BPATtern:TYPE****Equivalent GUI****Eye/Mask > User Pattern (Under **Bit Pattern**) > Load**

:MMEMory:LOAD:EYE:MASK**Type**

Command Only

Syntax

:MMEMory:LOAD:EYE:MASK <string>

Description

This command loads eye-mask file. The format of eye mask file should be the same as ones of Infiniium DCA (86100C). The extension of file should be .msk. The MASK pattern editing is not available through the command.

Parameter

Description	File name of eye mask (.msk)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

See the example in :CALCulate:EYE:MASK:STATe

Related Commands

:CALCulate:EYE:MASK:STATe

:CALCulate:EYE:EXECute

:CALCulate:EYE:STATe

Equivalent GUI

Eye Mask > Mask Pattern (under **Eye Control**) > **File Tab > Load**

:MMEMory:STORe:FDATa**Type**

Command Only

Syntax

:MMEMory:STORe:FDATa <string>

Description

This command stores measurement trace data.

Parameter

Description	File name of data file (.csv)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

10 OUTPUT 717;":MMEMory:STORe:FDATa ""D:\TDR\dataresult.csv""

Related Commands

:MMEMory:STORe:SNP

Equivalent GUI**File > Save Trace Data**

:MMEMory:STORe:IMAGe**Type**

Command Only

Syntax

:MMEMory:STORe:IMAGe <string>

Description

This command stores a display image into a bmp/png format file.

Parameter

Description	File name of image (.bmp or .png)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

10 OUTPUT 717;":MMEM:STOR:IMAG ""D:\TDR\figure1.png""

Related Commands

:DISPlay:IMAGe

Equivalent GUI

File > Save Image

:MMEMory:STORe:SNP**Type**

Command Only

Syntax**:MMEMory:STORe:SNP <string>****Description**

This command stores the touchstone format file. The file extension should be changed according to the setting of DUT topology.

File Extension	DUT Topology
.s1p	Single Ended 1
.s2p	Single Ended 2, Differential 1
.s4p	Single Ended 4, Differential 2

Parameter

Description	File name of touchstone file (.snp)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":MMEM:STOR:SNP ""D:\TDR\touchstone.s2p""
```

Related Commands**:MMEMory:STORe:FDATa****Equivalent GUI****File > Save Touchstone**

SENSE

:SENSe:AVERage:CLEar

Type

Command Only

Syntax

:SENSe:AVERage:CLEar

Description

This command clears average count.

Examples

10 OUTPUT 717;":SENS:AVER:CLE"

Related Commands

:SENSe:AVERage:COUNT

:SENSe:AVERage:STATe

Equivalent GUI

No equivalent GUI is available.

:SENSe:AVERage:COUNT**Type**

Command/Query

Syntax

:SENSe:AVERage:COUNT <numeric>

:SENSe:AVERage:COUNT?

Description

This command sets averaging factor value.

Parameter

Description	Number of Averaging factor.
Data Type	Long integer type (Long)
Range	1 to 999
Preset Value	16
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":SENS:AVER:COUN 16"

20 OUTPUT 717;":SENS:AVER:STAT ON"

Related Commands

:SENSe:AVERage:STATe

:SENSe:AVERage:CLEar

Equivalent GUI**Setup** > **Averaging** > Type in the average factor in to entry box beside **Averaging**

:SENSe:AVERage:STATe**Type**

Command/Query

Syntax

:SENSe:AVERage:STATe {ON|OFF|1|0}

:SENSe:AVERage:STATe?

Description

This command activates the averaging.

Parameter

Description	Status of averaging on or off
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":SENS:AVER:COUN 16"

20 OUTPUT 717;":SENS:AVER:STAT ON"

Related Commands

:SENSe:AVERage:CLEar

:SENSe:AVERage:COUNT

:TRIGger:AVERage

Equivalent GUI**Setup > Averaging**

:SENSe:BWIDth

Type

Command/Query

Syntax

:SENSe:BWIDth <numeric>

:SENSe:BWIDth?

Description

This command sets the IF bandwidth value.

Parameter

Description	IF Bandwidth value
Data Type	Double precision floating point type (Double)
Range	10 to 500E3
Preset Value	70E3
Unit	Hz
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":SENS:BWID 100E3"
20 OUTPUT 717;":SENS:BWID?"
30 ENTER 717;A

Related Commands

:SENSe:SPURious:AVOid:IMMediate

:SENSe:SPURious:AVOid:STATe

:SENSe:SPURious:INPut:DRATe

:SENSe:SPURious:STATe

Equivalent GUI

Setup> IF Bandwidth (Under **Average**)

**:SENSe:CORRection:COLLection:ADVanced:IMMediate
Type**

Command Only

Syntax

:SENSe:CORRection:COLLection:ADVanced:IMMediate

Description

This command clears the calibration in the basic mode and allows you to use the Advance Calibration Mode. Before you perform the calibration, such as TRL calibration, for channel 1 (TDR), this command should be executed.

Examples

10 OUTPUT 717;":SENS:CORR:COLL:ADV:IMM "

Equivalent GUI

Advanced Mode > check box (on) of **Advanced Calibration Methods**

:SENSe:CORRection:COLLection:DLCComp:LOAD**Type**

Command Only

Syntax

:SENSe:CORRection:COLLection:DLCComp:LOAD

Description

This command executes load measurement, as a part of Loss Compensation sequence.

Parameter

Description	Port number where the load measurement is executed.
Data Type	Long integer type (Long)
Range	1 to 4
Preset Value	-
Unit	-
Resolution	-

Examples

10 OUTPUT 717; ":SENS:CORR:COLL:DLC:LOAD 1 "

20 OUTPUT 717; ":SENS:CORR:COLL:DLC:SAVE"

Related Commands

:SENSe:CORRection:COLLection:DLCComp:OPEN

:SENSe:CORRection:COLLection:DLCComp:THRU

:SENSe:CORRection:COLLection:DLCComp:SAVE

Equivalent GUI

Setup > Deskew & Loss Compensation (under **Basic**) > **Measure > Port 1**
and/or **Port 2** and/or **Port 3** and/or **Port 4**

:SENSe:CORRection:COLLection:DLComp:OPEN**Type**

Command Only

Syntax**:SENSe:CORRection:COLLection:DLComp:OPEN****Description**

This command executes open measurement, as a part of Loss Compensation sequence.

Parameter

Description	Port number where the open measurement is executed.
Data Type	Long integer type (Long)
Range	1 to 4
Preset Value	-
Unit	-
Resolution	-

Examples

10 OUTPUT 717;":SENS:CORR:COLL:DLC:OPEN 1"

20 OUTPUT 717;":SENS:CORR:COLL:DLC:SAVE"

Related Commands**:SENSe:CORRection:COLLection:DLComp:LOAD****:SENSe:CORRection:COLLection:DLComp:THRU****:SENSe:CORRection:COLLection:DLComp:SAVE****Equivalent GUI****Setup > Deskew & Loss Compensation (under **Basic**) > Measure****NOTE**

This applies to Single-Ended 1-Port topology only.

:SENSe:CORRection:COLLection:DLCComp:SAVE

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:DLCComp:SAVE

Description

This command saves the result of Loss Compensation sequence.

Examples

10 OUTPUT 717;":SENS:CORR:COLL:DLC:SAVE"

Related Commands

:SENSe:CORRection:COLLection:DLCComp:OPEN

:SENSe:CORRection:COLLection:DLCComp:LOAD

:SENSe:CORRection:COLLection:DLCComp:THRU

Equivalent GUI

Setup > Deskew & Loss Compensation (under **Basic**) > **Measure** > **Port 1**
and/or **Port 2** and/or **Port 3** and/or **Port 4** > **Apply**

:SENSe:CORRection:COLLection:DLCComp:THRU**Type**

Command Only

Syntax**:SENSe:CORRection:COLLection:DLCComp:THRU****Description**

This command executes a thru measurement, as a part of Loss Compensation sequence.

Parameter

Description	Port number where the thru measurement is executed.
Data Type	Character string type (String)
Range	TH12: Thru between ports 1 and 2. TH13: Thru between ports 1 and 3. TH24: Thru between ports 2 and 4.
Preset Value	-
Unit	-
Resolution	-

Examples

10 OUTPUT 717; ":SENS:CORR:COLL:DLC:THRU TH12"

20 OUTPUT 717; ":SENS:CORR:COLL:DLC:SAVE"

Related Commands**:SENSe:CORRection:COLLection:DLCComp:OPEN****:SENSe:CORRection:COLLection:DLCComp:LOAD****:SENSe:CORRection:COLLection:DLCComp:SAVE****Equivalent GUI****Setup > Deskew & Loss Compensation (under **Basic**) > Measure****NOTE**

This applies to all other topologies except for Single-Ended 1-Port.

E5071C

:SENSe:CORRection:COLLection:ECAL:FCOMp:IMMediate
Type

Command Only

Syntax

:SENSe:CORRection:COLLection:ECAL:FCOMp:IMMediate

Description

This command executes fixture compensation after ECal

Examples

10 OUTPUT 717;":SENS:CORR:COLL:ECAL:FCOM:IMM"

Equivalent GUI

Setup > ECal (under Basic) > Calibrate > Fixture Comp

:SENSe:CORRection:COLLection:ECAL:IMMediate

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:ECAL:IMMediate

Description

This command executes full calibration using the ECal module

Examples

10 OUTPUT 717;":SENS:CORR:COLL:ECAL:IMM"

Equivalent GUI

Setup > **ECal** (under **Basic**) > **Calibrate**

:SENSe:CORRection:DCONstant**Type**

Command/Query

Syntax

:SENSe:CORRection:DCONstant <numeric>

:SENSe:CORRection:DCONstant?

Description

This command sets dielectric constant value.

Parameter

Description	Dielectric constant value
Data Type	Double precision floating point type (Double)
Range	10m to 100
Preset Value	1000m
Unit	-
Resolution	-

Query Response

{numeric} <newline><^END>

Examples

```
10 OUTPUT 717;":SENS:CORR:DCON 0.01"  
20 OUTPUT 717;":SENS:CORR:DCON?"  
30 ENTER 717;A
```

Related Commands

:SENSe:CORRection:RIMPedance

Equivalent GUI**Setup > Dielectric Const.** (under **More Functions**)

:SENSe:CORRection:EXTension:AUTO:IMMEDIATE

Type

Command Only

Syntax

:SENSe:CORRection:EXTension:AUTO:IMMEDIATE

Description

This command executes deskew (auto port extension).

Examples

10 OUTPUT 717;":SENS:CORR:EXT:AUTO:IMM"

Related Commands

:SENSe:CORRection:EXTension:AUTO:PORT{pt}

:SENSe:CORRection:EXTension:AUTO:STANdard

Equivalent GUI

Setup > DeskeW (under **Basic**).

:SENSe:CORRection:EXTension:AUTO:PORT{pt}**Type**

Command/Query

Syntax

:SENSe:CORRection:EXTension:AUTO:PORT{[1]-4} {ON|OFF|1|0}

:SENSe:CORRection:EXTension:AUTO:PORT{[1]-4}?

Description

This command sets ON/OFF or gets the status of auto port extension.

Parameter

Description	Status of auto port extension
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:CORR:EXT:AUTO:PORT1 ON"
20 OUTPUT 717;":SENS:CORR:EXT:AUTO:PORT1?"
30 ENTER 717;A
```

Related Commands

:SENSe:CORRection:EXTension:AUTO:IMMediate

:SENSe:CORRection:EXTension:AUTO:STANdard

Equivalent GUI

Setup > Deskew (under **Basic**) > **Option > Port1** and/or **Port2** and/or **Port3** and/or **Port4** OR

Setup > Deskew & Loss Compensation (under **Basic**) > **Option > Port1** and/or **Port2** and/or **Port3** and/or **Port4**

:SENSe:CORRection:EXTension:AUTO:STANdard**Type**

Command/Query

Syntax

:SENSe:CORRection:EXTension:AUTO:STANdard {OPEN|SHORT}

:SENSe:CORRection:EXTension:AUTO:STANdard?

Description

This command sets the standard for auto port extension.

Parameter

Description	Standard for Deskew
Data Type	Character string type (String)
Range	OPEN: Open termination SHORT: Short termination
Preset Value	OPEN
Unit	-
Resolution	-

Query Response

{OPEN|SHORT}<newline><^END>

Examples

10 OUTPUT 717;":SENS:CORR:EXT:AUTO:STAN OPEN"

20 OUTPUT 717;":SENS:CORR:EXT:AUTO:STAN?"

30 ENTER 717;A\$

Related Commands

:SENSe:CORRection:EXTension:AUTO:PORT{pt}

:SENSe:CORRection:EXTension:AUTO:IMMediate

Equivalent GUI**Setup > Deskew** (under **Basic**) > **Option** > select **Standard Type****Setup > Deskew & Loss Compensation** (under **Basic**) > **Option** > select **Standard Type**

E5071C

:SENSe:CORRection:RIMPedance**Type**

Command/Query

Syntax

:SENSe:CORRection:RIMPedance <numeric>

:SENSe:CORRection:RIMPedance?

Description

This command sets reference impedance value.

Parameter

Description	Reference impedance value
Data Type	Double precision floating point type (Double)
Range	1m to 10M
Preset Value	50
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":SENS:CORR:RIMP 0.001"

20 OUTPUT 717;":SENS:CORR:RIMP?"

30 ENTER 717;A

Related Commands

:SENSe:CORRection:DCONstant

Equivalent GUI**Setup > Ref. Z** (under **More Functions**)

E5071C

:SENSe:DLENgth:AUTO:IMMediate

Type

Command Only

Syntax

:SENSe:DLENgth:AUTO:IMMediate

Description

This command executes auto DUT length setting

Examples

10 OUTPUT 717;":SENS:DLEN:AUTO:IMM"

Related Commands

:SENSe:DLENgth:DATA

Equivalent GUI

TDR/TDT > DUT Length: Auto (under Basic) > Measure

:SENSe:DLENgth:DATA**Type**

Command/Query

Syntax

:SENSe:DLENgth:DATA <numeric>

:SENSe:DLENgth:DATA?

Description

This command sets DUT length value.

Parameter

Description	DUT length value
Data Type	Double precision floating point type (Double)
Range	6.26n to 416n
Preset Value	6.26n
Unit	sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:DLEN:DATA 6.26e-9"  
20 OUTPUT 717;":SENS:DLEN:DATA?"  
30 ENTER 717;A
```

Related Commands

:SENSe:DLENgth:AUTO:IMMediate

Equivalent GUI**TDR/TDT** > **DUT Length** (under **Basic**) > Enter the DUT length

:SENSe:SPURious:AVOid:IMMediate

Type

Command only

Syntax

:SENSe:SPURious:AVOid:IMMediate

Description

This command executes avoid spurious.

Examples

10 OUTPUT 717;":SENS:SPUR:AVO:IMM"

Related Commands

:SENSe:BWIDth

:SENSe:SPURious:AVOid:STATe

:SENSe:SPURious:INPut:DRATe

:SENSe:SPURious:STATe

Equivalent GUI

Setup> Avoid Spurious (Under **Hot TDR**)

NOTE

A green check mark is observed next to the **Avoid Spurious** button when this command is successful.

:SENSe:SPURious:AVOid:STATe**Type**

Query only

Syntax

:SENSe:SPURious:AVOid:STATe?

Description

This command queries the avoid spurious state. This command is used only with Hot TDR mode.

NOTE

This command is ON when
:SENS:SPURious:AVOid:IMMediate command succeeds.

NOTE

This command is OFF when
:SENS:SPURious:AVOid:IMMediate command fails to find
spurious.

Parameter

Description	Status of avoid spurious
Data Type	Boolean type (Boolean)
Range	1 (ON) or 0 (OFF)
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0} <newline><^END>

Examples

10 OUTPUT 717;":SENS:SPUR:AVO:STAT?"

Related Commands

:SENSe:BWIDth

:SENSe:SPURious:AVOid:IMMediate

:SENSe:SPURious:INPut:DRATe

:SENSe:SPURious:STATe

Equivalent GUI

Setup> Avoid Spurious (Under **Hot TDR**)

NOTE

A green check mark is observed next to the **Avoid Spurious** button when Avoid Spurious state is successfully turned ON.

:SENSe:SPURious:INPut:DRATe**Type**

Command/Query

Syntax

:SENSe:SPURious:INPut:DRATe <numeric>

:SENSe:SPURious:INPut:DRATe?

Description

This command sets the value of input bit rate for avoid spurious.

Parameter

Description	Value of input bit rate
Data Type	Double precision floating point type (Double)
Range	1.21E6 to 16E9
Preset Value	1E9
Unit	bps
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":SENS:SPUR:INP:DRAT 1.5E9"

20 OUTPUT 717;":SENS:SPUR:INP:DRAT?"

30 ENTER 717;A

Related Commands

:SENSe:BWIDth

:SENSe:SPURious:AVOid:IMMediate

:SENSe:SPURious:AVOid:STATe

:SENSe:SPURious:STATe

Equivalent GUI**Setup> Data Rate** (Under **Hot TDR**)

:SENSe:SPURious:STATe

Type

Query only

Syntax

:SENSe:SPURious:STATe?

Description

This command queries the Hot TDR mode status.

NOTE To turn ON Hot TDR mode, use
:SENSe:SPURious:AVOid:IMMediate

NOTE To turn OFF Hot TDR mode, use :SYSTem:PRESet

Parameter

Description	Status of Hot TDR mode
Data Type	Boolean type (Boolean)
Range	1 (ON) or 0 (OFF)
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":SENS:SPUR:STAT?"

Related Commands

:SENSe:BWIDth
:SENSe:SPURious:AVOid:IMMediate
:SENSe:SPURious:AVOid:STATe
:SENSe:SPURious:INPut:DRATe

Equivalent GUI

"TDR ?" is indicated at the channel window. The blue SVC indicator is also turned ON.

SOURCE**:SOURce:POWer**

Type

Command/Query

Syntax

:SOURce:POWer <numeric>

:SOURce:POWer?

Description

This command sets the value for source power.

Parameter

Description	Value for source power
Data Type	Double precision floating point type (Double)
Range	Depends on frequency option (See specification)
Preset Value	-20
Unit	dBm
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

10 OUTPUT 717;":SOURce:POWer 20"

20 OUTPUT 717;":SOURce:POWer?"

30 ENTER 717;A\$

Equivalent GUI

Setup> Source Power (Under **More Functions**)

SYSTEM

:SYSTem:ERRor

Type

Query Only

Syntax

:SYSTem:ERRor?

Description

This command gets error code & description

Parameter

Description	Error code
Data Type	Double precision floating point type (Double)
Range	- -
Preset Value	0 "No error"
Unit	- -
Resolution	- -

Query Response

{ - - } <newline><^END>

Examples

10 OUTPUT 717;":SYST:ERR?"
30 ENTER 717;A

Equivalent GUI

No equivalent GUI is available.

:SYSTem:PRESet

Type

Command Only

Syntax

:SYSTem:PRESet

Description

This command presets the E5071C setting. Calibration data will be lost.

Examples

10 OUTPUT 717;":SYST:PRES"

Equivalent GUI

Setup > Preset (under **Basic**)

TRIGGER

:TRIGger:AVERage

Type

Command/Query

Syntax

:TRIGger:AVERage {ON|OFF|1|0}

:TRIGger:AVERage?

Description

This command sets averaging trigger on/off. When averaging trigger is on, one trigger makes one averaging measurement. For example, if the averaging factor is set at 16, one trigger makes a measurement for 16 times.

Parameter

Description	Status of Averaging trigger
Data Type	Boolean type (Boolean)
Range	ON OFF 1 0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

10 OUTPUT 717;":TRIG:MODE HOLD"
20 OUTPUT 717;":SENS:AVER:COUN 16"
30 OUTPUT 717;":SENS:AVER:STAT ON"
40 OUTPUT 717;":TRIG:AVER ON"
50 OUTPUT 717;":TRIG:SING"

Related Commands

:SENSe:AVERage:COUNt
:SENSe:AVERage:STATe
:TRIGger:SINGLE

Equivalent GUI

No equivalent GUI is available.

:TRIGger:MODE**Type**

Command/Query

Syntax

:TRIGger:MODE {HOLD|SINGle|RUN}

:TRIGger:MODE?

Description

This command sets trigger mode.

Parameter

Description	Trigger mode
Data Type	Character string type (String)
Range	HOLD: Trigger is on hold until the conditions are meet, the trigger event starts. SINGle: Trigger event is run once. RUN: Trigger event is running continuous.
Preset Value	RUN
Unit	-
Resolution	-

Query Response

{HOLD|SINGle|RUN}<newline><^END>

Examples

```
10 OUTPUT 717;":TRIG:MODE HOLD"
20 OUTPUT 717;":TRIG:MODE?"
30 ENTER 717;A$
```

Related Commands

:TRIGger:SINGle

Equivalent GUI**Stop Single or Run**

:TRIGger:SINGle

Type

Command Only

Syntax

:TRIGger:SINGle

Description

This command executes single trigger.

Examples

```
10 OUTPUT 717;":TRIG:SING"
```

Related Commands

:TRIGger:MODE

Equivalent GUI

No equivalent GUI is available.