User's
Manual

AQ6374
Optical Spectrum Analyzer
Remote Control

YOKOGAWA
Yokogawa Test & Measurement Corporation

IM AQ6374-17EN
4th Edition

Thank you for purchasing the AQ6374 Optical Spectrum Analyzer. This remote control user's manual covers the AQ6374. It describes the following and.

- GP-IB Interface
- RS-232 Interface
- Ethernet Interface and Communication Commands
- Program Functions

To ensure correct use, please read this manual thoroughly before beginning operation. After reading this manual, keep it in a convenient location for quick reference in the event a question arises during operation. In addition to this manual, There are four manuals for the AQ6374 including this one. Read them along with this manual.

List of Manuals

Manual Title	Manual No.	Description
AQ6374 Optical Spectrum Analyzer User's Manual	IM AQ6374-01EN	The manual is located on the CD included in your package (pdf format). Explains all functions and operating procedures of the AQ6374 except remote control and program functions.
AQ6374 Optical Spectrum Analyzer Remote Control User's Manual	IM AQ6374-17EN	This manual. The manual is located on the CD included in your package (pdf format). Explains functions for controlling the instrument with communication commands and program functions.
AQ6374 Optical Spectrum Analyzer Getting Started Guide	IM AQ6374-02EN	Provided as a printed manual. This guide explains the handling precautions, basic operations, and specifications of the AQ6374.
Optical Spectrum Analyzer	IM AQ6360-92Z1	A document for China.

The "-EN" in the manual number is the language code.

Contact information of Yokogawa offices worldwide is provided on the following sheet.

Document Description	Description
PIM 113-01Z2	List of worldwide contacts

Notes

- The contents of this manual are subject to change without prior notice as a result of improvements in the instrument's performance and functions. Display contents illustrated in this manual may differ slightly from what actually appears on your screen.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer.
- Copying or reproducing all or any part of the contents of this manual without the permission of Yokogawa Test & Measurement Corporation is strictly prohibited.

Trademarks

- Microsoft and Windows are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are registered trademarks or trademarks of Adobe Systems incorporated.
- The company and product names used in this manual are not accompanied by the registered trademark or trademark symbols (® and TM).
- Other company and product names are registered trademarks or trademarks of their respective companies.

4th Edition: April 2021 (YMI)

Revisions

• 1st Edition	February, 2017
• 2nd Edition	October 2017
• 3rd Edition	April 2019
• 4th Edition	April 2021

Safety Precautions

This instrument is an IEC protection class I instrument (provided with terminal for protective earth grounding).
The general safety precautions described herein must be observed during all phases of operation. If the instrument is used in a manner not specified in this manual, the protection provided by the instrument may be impaired. Yokogawa Electric Corporation assumes no liability for the customer’s failure to comply with these requirements.

The following safety symbols and wording is used in this manual.

⚠ Warning: Handle with care. Refer to the user’s manual or service manual.
This symbol appears on dangerous locations on the instrument which require special instructions for proper handling or use. The same symbol appears in the corresponding place in the manual to identify those instructions.

∼ Alternating current

| ON (power)

○ OFF (power)

French

⚠ Avertissement : À manipuler délicatement.
Toujours se reporter aux manuels d’utilisation et d’entretien. Ce symbole a été apposé aux endroits dangereux de l’instrument pour lesquels des consignes spéciales d’utilisation ou de manipulation ont été émises. Le même symbole apparaît à l’endroit correspondant du manuel pour identifier les consignes qui s’y rapportent.

∼ Courant alternatif

| Marche (alimentation)

○ Arrêt (alimentation)

Conventions Used in This Manual

Safety Markings

The following markings are used in this manual.

⚠

Improper handling or use can lead to injury to the user or damage to the instrument. This symbol appears on the instrument to indicate that the user must refer to the user's manual for special instructions. The same symbol appears in the corresponding place in the user's manual to identify those instructions. In the manual, the symbol is used in conjunction with the word “WARNING” or “CAUTION.”

WARNING

Calls attention to actions or conditions that could cause serious or fatal injury to the user, and precautions that can be taken to prevent such occurrences.

CAUTION

Calls attentions to actions or conditions that could cause light injury to the user or damage to the instrument or user’s data, and precautions that can be taken to prevent such occurrences.

French

AVERTISSEMENT

Attire l’attention sur des gestes ou des conditions susceptibles de provoquer des blessures graves (voire mortelles), et sur les précautions de sécurité pouvant prévenir de tels accidents.

ATTENTION

Attire l’attention sur des gestes ou des conditions susceptibles de provoquer des blessures légères ou d’endommager l’instrument ou les données de l’utilisateur, et sur les précautions de sécurité susceptibles de prévenir de tels accidents.

Note

Calls attention to inf

Notations Used in the Procedural Explanations

On pages that describe the operating procedures in each chapter, the following notations are used to distinguish the procedure from their explanations.

Procedure This subsection contains the operating procedure used to carry out the function described in the current section. The procedures are written with inexperienced users in mind; experienced users may not need to carry out all the steps.

Explanation This subsection describes the setup parameters and the limitations on the procedures.

Terms Used in Explanations of Procedures

Panel Keys and Soft Keys
Bold characters used in the procedural explanations indicate characters that are marked on the panel keys or the characters of the soft keys displayed on the screen menu.

Units

k	Denotes 1000. Example: 12 kg, 100 kHz
K	Denotes 1024. Example: 459 KB (file size)

How To Use This Manual

Structure of This Manual

This user’s manual consists of the following eight chapters, an appendix, and an index.

Chapter 1 Remote Control Functions

This section describes the various types of communication interfaces and program functions.

Chapter 2 GP-IB Interface

Describes the functions and lists the specifications of the GP-IB1 port.

Chapter 3 Ethernet Interface

Describes the functions and lists the specifications of the Ethernet interface.

Chapter 4 Serial (RS-232) Interface

Describes the functions and lists the specifications of the RS-232 interface.

Chapter 5 Status Registers

Explains the status byte and describes the various kinds of registers, cues, and other items.

Chapter 6 Remote Commands

Describes each individual command that can be used.

Chapter 7 Program Function

Explains the program function for controlling another instrument using the AQ6374 as the controller.

Appendix

Lists commands that are compatible with the AQ6317.

Index

An alphabetical index.

Contents

Chapter 7 Program Function

Appendix AQ6317-Compatible GP-IB Commands

Index

1.1 Remote Interfaces

This instrument is equipped with the following remote interfaces.

GP-IB (IEEE 488.2, See Chapter 2)

This port is used to connect a controller such as a PC to remote control this instrument.
Connect a controller or another device controlled by the controller to this port.
This instrument is controlled using remote commands.
Two types of remote commands are provided: the instrument’s native commands complying with SCPI (Standard Commands for Programmable Instruments), and commands compatible with the conventional model AQ6317 (see the appendix).

RS-232 (See Chapter 3)

This port is used to connect a controller such as a PC to control the instrument remotely.

Ethernet (See Chapter 4)

This port is used to connect a controller such as a PC to control the instrument remotely via network.

1.2 Switching between Local and Remote

Switching from Local to Remote

When in Local mode, if a listen address is sent from the controller that sets REN (remote enable) and ATN to “True,” the instrument enters Remote mode.

- When in Remote mode, the REMOTE indicator lights.
- Keys other than the LOCAL key are disabled.
- Settings entered in Local mode are held even if switching to Remote mode.
- When an LLO (Local Lock Out) message is received from the controller, the instrument enters local lockout status. In LLO status, the LOCAL key is disabled and does not return the instrument to Local mode even when pressed. After cancelling the local lockout status, press the LOCAL key. To cancel the local lockout status, set REN to “False” from the controller.

Switching from Remote to Local

If you press the LOCAL key when in Remote mode the instrument enters Local mode. However, it does not return to Local mode if in the local lockout state.

- The REMOTE indicator turns off.
- All keys are enabled.
- Settings entered in Remote mode are held even if switching to Local mode.
- When a GTL (Go to Local) message is received from the controller, the instrument enters Local mode even if REN is set to False.

1.3 Sending/Receiving Remote Commands

Buffers

Input Buffer

The instrument’s input buffer is a single stage 3 MB buffer. When receiving data that exceeds the buffer size, the data after the 3 MB is discarded. The remote command after the last command separator of the 3 MB of data is deleted.

Output Buffer

The instrument’s output buffer is a single stage 3 MB buffer. Only the most recent data is held. (When a talker command is received while there is data in the buffer, the old data in the buffer is replaced with the incoming data.)

For commands whose talker data size is large, such as the commands for acquiring trace data (wavelength or power), send the commands one by one.

If you include multiple talker commands in a single output statement by delimiting each talker command with a semicolon, correct data cannot be acquired if talker data exceeding the buffer size is generated.

When talker commands are combined and executed resulting in generation of talker data that exceeds the buffer size, the following process is carried out.

- The query error bit (QYE) of the standard event status register is set to 1.
- The talker output buffer is cleared.
- Commands received even after the buffer overflow are processed. Note, however, that talker data by talker commands is not stored at the output buffer.

Error Buffer

This instrument’s error buffer is of a single stage and stores only the latest error information.

Message Terminators

This instrument allows the following message terminators to be used.

Program Message Terminators

- Assertion of EOI (End-Of-Identify) signal
- LF (line feed) character
- LF+EOI

Here, LF is a line feed (0Ah) in ASCII. For CR + LF, because CR (0Dh) is recognized as “wsp,” CR + LF can consequently also be used as a message terminator. Also, for waveform binary transfer, only EOI is used as a message terminator.

Response Message Terminator

LF+EOI is used as the response message terminator.

Receiving Remote Commands

- When completing receipt of a remote command, the instrument releases the GP-IB bus.
- When receiving the next command while a command action is being executed, the instrument captures that command to store it in the receive buffer, and then releases the GP-IB bus.
- When there is a remote command in the receive buffer, the instrument does not capture a successive command even if there are commands on the GP-IB bus.
- When the action of the preceding command is complete, the instrument executes the command stored in the receive buffer and clears the buffer. Then it captures the next command into the receive buffer if there is one on the bus.
- When an output statement contains multiple remote commands, this instrument captures them all and services them in the order they were written. In this case, unless the last command in the statement has started to be executed, this instrument cannot capture the next command.

Data Inquiry

- Inquiry of data by the external controller is made using a query command or a data output request from the controller.
- Query commands end with a question mark (?).
- For query commands with an argument, the argument is specified in the form of <wsp> + <argument> at the end of the “?”.
- When a query command is received, the instrument prepares a reply to the query command in the output buffer.
- Data in the output buffer will be retained until the instrument receives an input statement or a new query command from the controller.
- If multiple query commands are specified and written in succession using a semicolon “;”, the instrument prepares replies to all of them in the output buffer. In this case, the instrument will collectively output all of the prepared data when receiving the next data output request.

Setting the timeout time

A timeout time setting of 30 seconds or more is recommended.
At approximately 10 minute intervals, the instrument performs an auto offset for approximately 30 seconds. The communication timeout of the external controller should be set to 30 seconds or more so that a timeout does not occur during the execution of the offset. See the user’s manual of your remote interface card for instructions on how to set the communication timeout time.

The instrument's auto offset function is set to ON by default, and it performs offset of the analog circuits at approximately 10 minute intervals. The offset process takes about 30 seconds.

If you do not want to set the communication timeout to 30 seconds or less

To avoid remote malfunctions due to communication timeouts, offset processing can be performed manually. Turn the auto offset function OFF in advance, and perform the offset manually during a gap in measurement sequences. Wait approximately 30 seconds until the offset process is finished. After the offset is complete, restart the measurement sequence.

The remote commands are as follows.

Turn OFF the auto offset function	`:CALibration:ZERO off`
Perform a manual offset	`:CALibration:ZERO once`

Note

- An offset interval of 10 minutes is recommended.
- If the Auto Offset key is OFF, the offset can fluctuate over time, and the level axis performance can degrade. Always have it turned ON.
- When the Auto Offset key is set to ON, AUT OFS is displayed at the bottom of the screen.

Device Trigger Function

When GET (Group Execute Trigger) is received, the instrument will perform a single sweep.

2.1 Connecting via GP-IB

GP-IB Cable

This instrument is equipped with an IEEE standard 488-1978 24-pin GP-IB connector. Use a GP-IB cable that conforms to the IEEE standard 488-1978.

Connections

Can be connected to a PC for remote control of the instrument from the PC.
Turn OFF all the power switches of the AQ6374 and any devices to be connected to it.
Connect a cable to the GP-IB port on the rear panel of the instrument.

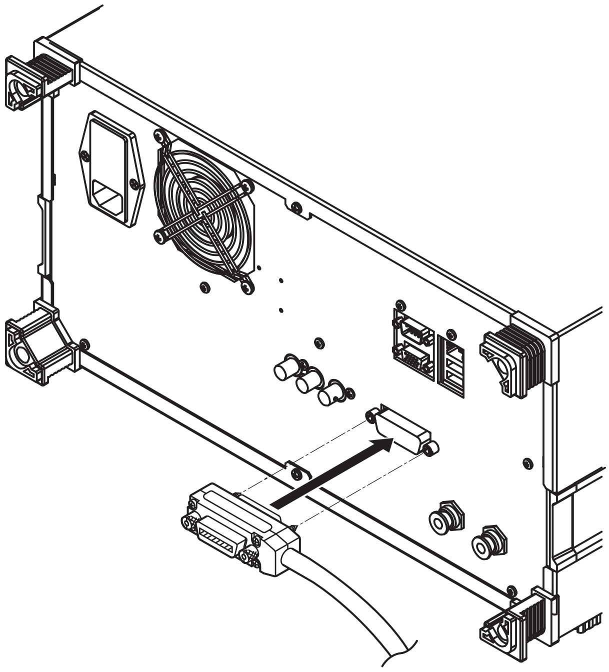
CAUTION

Always turn OFF the power to the instrument and the PC when connecting or disconnecting communication cables. Failure to turn OFF the power can result in malfunction or damage to internal circuitry.

French

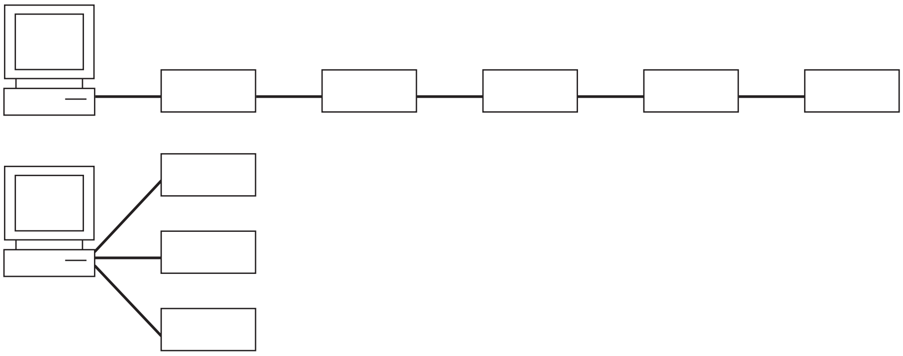
ATTENTION

Veillez à mettre le PC et l'oscilloscope AQ6374 hors tension lorsque vous branchez ou débranchez les câbles de communication, car cela risquerait de provoquer des dysfonctionnements ou des courts-circuits internes.



Precautions When Making Connections

- Securely fasten the screw that is attached to the GP-IB cable connector.
- You can connect several cables to connect to several devices. However, fifteen or more devices including the controller cannot be connected to a single bus.
- When connecting several devices, you cannot specify the same address for more than one.
- Use a cable of two meters or longer to connect between devices.
- Ensure that the total length in cables does not exceed twenty meters.
- When carrying out communications, make sure that at least two-thirds of all connected devices are turned ON.
- To connect multiple devices, wire them in a daisy-chain or star configuration as shown below. You can also mix these configurations. Loop configuration is not allowed.



2.2 GP-IB Interface Function

GP-IB Interface Function

Listener Function

- All of the same settings can be performed using the interface (except for power ON/OFF and communication settings) as when using the instrument's panel keys.
- Settings, waveforms, and other data can be received through output commands from the controller.
- Additionally, you can also receive commands regarding status reports and other data.

Talker Function

- Settings, waveforms, and other data can be output.

Note

Listen only, talk only, and controller functions are not available.

Switching between Remote and Local

Switching from Local to Remote

When in Local mode, if the instrument received a listen address from the controller that sets REN (remote enable) and ATN to "True," the instrument enters Remote mode.

- When in Remote mode, the REMOTE indicator lights.
- Keys other than the LOCAL key are disabled.
- Settings entered in Local mode are held even if switching to Remote mode.
- When an LLO (Local Lock Out) message is received from the controller, the instrument enters local lockout status. In LLO status, the LOCAL key is disabled and does not return this instrument to Local mode even when pressed. After cancelling the local lockout status, press the LOCAL key. To cancel the local lockout status, set REN to "False" from the controller.

Switching from Remote to Local

If you press the LOCAL key when in Remote mode the instrument enters Local mode. However, it does not return to Local mode if in the local lockout state.

- The REMOTE indicator turns off.
- All keys are enabled.
- Settings entered in Remote mode are held even if switching to Local mode.
- When a GTL (Go to Local) message is received from the controller, the instrument enters Local mode even if REN is set to False.

Note

The GP-IB interface cannot be used simultaneously with other communication interfaces (RS-232, USB, or Ethernet).

2.3 GP-IB Interface Specifications

GP-IB Interface Specifications

Electromechanical specifications:	Conforms to IEEE std. 488-1978
Functional specifications:	See table below
Protocols:	Conforms to IEEE std. 488.2-1992
Encoding:	ISO (ASCII)
Mode:	Addressable mode
Address setting:	Addresses 0-30 can be set in the GP-IB setting screen in the SYSTEM menu.
Remote mode cancel:	Press LOCAL to cancel Remote mode. Note that this is disabled when under Local Lockout by the controller.

Functional Specifications

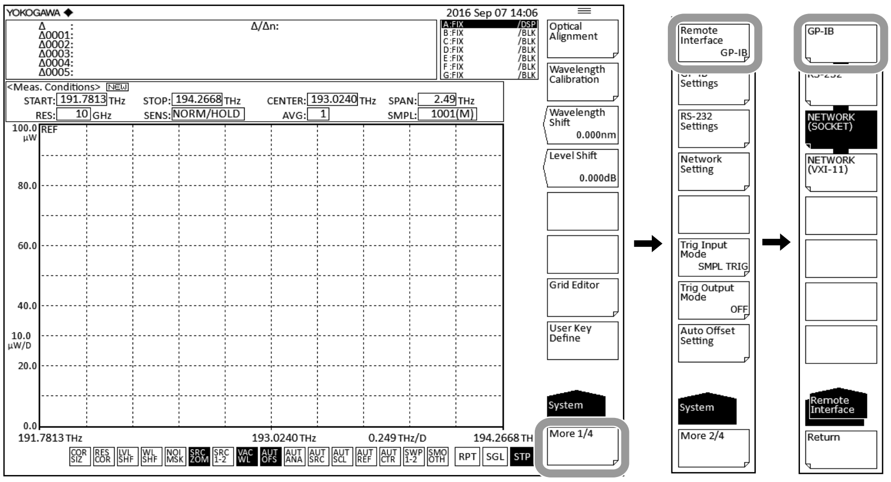
Function	Subset	Description
Source handshake	SH1	All capabilities of send handshake
Acceptor handshake	AH1	All capabilities of receive handshake
Talker	T6	Basic talker function, serial polling, and talker cancel function through MLA (my listen address). Talker only not provided.
Listener	L4	Basic listener function, serial polling, and listener cancel function through MLA (my listen address). Listener only not provided.
Service request	SR1	All service request functions
Remote local	RL1	All Remote/Local functions
Parallel port	PP0	Parallel polling function not provided
Device clear	DC1	All device clear functions Output buffer clear Input buffer clear (clearing of an unexecuted commands) Error buffer clear STB, ESR clear
Device trigger	DT0	Device trigger function
Controller	C0	Controller function not provided
Electrical characteristics	E1	Open collector

2.4 Setting the GP-IB Address

Procedure

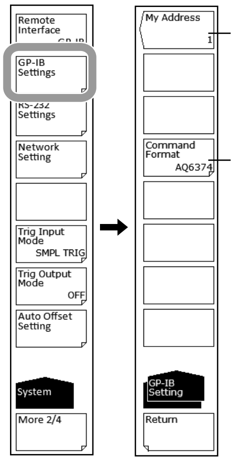
Selecting the Communication Interface

1. Press **SYSTEM**. The system setting menu is displayed.
2. Press the **More1/4** soft key. The communication interface setting menu is displayed.
3. Press the **Remote Interface** soft key. The setting menu for the interface to be used is displayed.
4. Press the **GP-IB** soft key to specify GP-IB as the communication interface.



Setting the Address

5. Press the **GP-IB Settings** soft key. The GP-IB setting menu is displayed.
6. Press the **My Address** soft key. The GP-IB address setting screen is displayed.
7. Set the GP-IB address using the **rotary knob** or the **arrow keys,** and press **ENTER**.



GP-IB address setting

Command format

Setting the Command Format

8. Perform these steps if you will use AQ6317 commands. Press the **Command Format** soft key. The command format setting menu is displayed.
9. Normally, you will enter AQ6374. If you wish to use AQ6317 commands, enter AQ6317.

Explanation

The settings below are used when entering the settings that can be entered using the instrument’s panel keys from a controller, or when outputting settings or waveform data to the controller.

GP-IB Address Settings

When in Addressable mode, set the instrument’s address within the following range.
0–30
Each device that can be connected via GP-IB has its own unique GP-IB address. This address allows each device to be distinguished from other devices. Therefore, when connecting the instrument to a PC or other device, make sure not to set the same address on the instrument as any of the other devices.

Note

Do not change an address while the controller or other devices are using GP-IB.

Command Format Settings

Normally, you will enter AQ6374 mode.
If you wish to use the commands of the AQ6317 (another product in the series), enter AQ6317. See the appendix for AQ6317 commands that are compatible with the AQ6317.

2.5 Responses to Interface Messages

Responses to Interface Messages

Responses to Uniline Messages

IFC (Interface Clear)

Clears talker and listener. Output is cancelled if outputting data.

REN (Remote Enable)

Switches between Local and Remote.

IDY (Identify) is not supported.

Responses to Multiline Messages (Address Commands)

GTL (Go To Local)

Switches to Local mode.

SDC (Selected Device Clear)

- Clears program messages (commands) being received, and the output queue.
- The *OPC and *OPC? commands are invalid during execution.
- The *WAI command closes immediately.

PPC (parallel poll configure), GET (group execute trigger), and TCT (take control) are not supported.

Responses to Multiline Messages (Universal Commands)

LLO (Local Lockout)

Disables the front panel SHIFT+CLEAR operation, and prohibits switching to Local mode.

DCL (Device Clear)

Same operation as SDC.

SPE (Serial Poll Enable)

Places the talker function of all devices on the bus in Serial poll mode. The controller polls each device in order.

SPD (Serial Poll Disable)

Cancels Serial poll mode for the talker function of all devices on the bus.

PPU (Parallel Poll Unconfigure) is not supported.

Definition of Interface Messages

Interface messages are also called *interface commands* or *bus commands*, and are commands that are issued from the controller. Interface messages come in the following categories.

Uniline Messages

A message is sent through a single command line. The following are the three types of uniline messages.

IFC (Interface Clear)
REN (Remote Enable)
IDY (Identify)

Multiline Messages

A message is sent through eight data lines. Multiline messages come in the following categories.

Address Commands

These commands are valid when the device is specified as the listener or the talker. The following are the five types of address commands.

Commands valid for devices specified as listeners

GTL (Go To Local)
SDC (Selected Device Clear)
PPC (Parallel Poll Configure)
GET (Group Execute Trigger)

Commands valid for devices specified as talkers

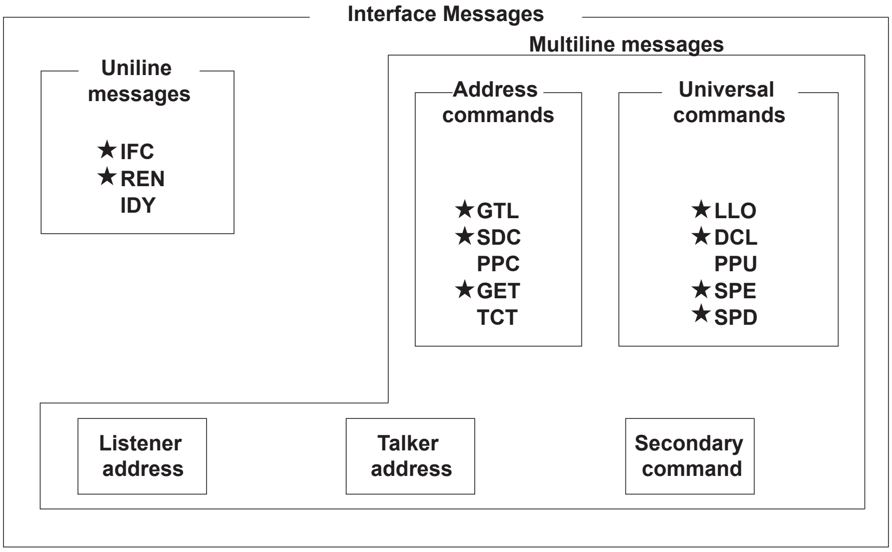
TCT (Take Control)

Universal Commands

These commands are valid for all devices regardless of whether they are specified as listeners, talkers, or neither. The following are the three types of universal commands.

LLO (Local Lockout)
DCL (Device Clear)
PPU (Parallel Poll Unconfigure)

Additionally, an interface message can consist of a listener address, talker address, or secondary command.



A star indicates an interface message supported by this instrument.

Note

Differences between SDC and DCL

Of the multiline messages, SDC is an address command requires specification of the talker or listener, and DCL is a universal command that does not require specification of the talker or listener. Therefore, SDC is applicable only to certain devices, but DCL is applicable to all devices on the bus.

2.6 Sample Program

The following shows an example of controlling the AQ6374 remotely using the GP-IB port. The sample program uses Visual Basic 6.0 as the programming language. Also, a GP-IB board by National Instruments (hereinafter, “NI”) is used as the GP-IB controller and the NI-supplied driver is used as a library.

Sample Program 1

The program sets the measurement conditions (center wavelength, span, sensitivity, and the sampling number) and then performs a sweep. After completing this sweep, the program executes a thresh-based spectrum width analysis and then outputs the results to the screen.

```
Const BOARD_ID = 0                                 ' GP-IB Interface card
                                                   Address
Const osa = 1                                      ' OSA GP-IB Address

Private Sub AQ637XTEST()
   Dim intData As Integer
   Dim dblMeanWL As Double
   Dim dblSpecWd As Double
   Dim strData As String

   ' === GP-IB Interface setting ===
   ' send IFC
   Call SendIFC(BOARD_ID)

   ' assert th REN GPIB line
   intAddrList(0) = NOADDR
   Call EnableRemote(BOARD_ID, intAddrList())

   ' GPIB time out setting
   Call ibtmo(BOARD_ID, T30s)                      ' Time out = 30sec

   ' === Set the measurement parameter ===
   Call SendGPIB(osa, "*RST")                      ' Setting initialize
   Call SendGPIB(osa, "CFORM1")                    ' Command mode
                                                   set(AQ637X mode)
   Call SendGPIB(osa, ":sens:wav:cent 1550nm")     ' sweep center wl
   Call SendGPIB(osa, ":sens:wav:span 10nm")       ' sweep span
   Call SendGPIB(osa, ":sens:sens mid")            ' sens mode = MID
   Call SendGPIB(osa, ":sens:sweep:points:auto on")
                                                   ' Sampling Point = AUTO

   ' === Sweep execute ===
   Call SendGPIB(osa, ":init:smode 1")             ' single sweep mode
   Call SendGPIB(osa, "*CLS")                      ' status clear
   Call SendGPIB(osa, ":init")                     ' sweep start

   ' === Wait for sweep complete ===
   Do
      Call SendGPIB(osa, ":stat:oper:even?")       ' get Operation Event
                                                   Register
      strData = RecieveGPIB(osa)
      intData = Val(strData)
   Loop While ((intData And 1) <> 1)               ' Bit0: Sweep status

' === Analysis ===
   Call SendGPIB(osa, ":calc:category swth")       ' Spectrum width
                                                   analysis(THRESH type)
   Call SendGPIB(osa, ":calc")                     ' Analysis Execute
   Call SendGPIB(osa, ":calc:data?")               ' get data
   strData = RecieveGPIB(osa)
```

```
    ' === Capture analytical results ===
    dblMeanWL = Val(Left(strData, 16))             ' get mean wavelegnth
    dblSpecWd = Val(Mid(strData, 18, 16))          ' get spectrum width

    ' === Output the result to the screen ===
    MsgBox ("MEAN WL: " & dblMeanWL * 1000000000# & " nm" & vbCrLf & _
            "SPEC WD: " & dblSpecWd * 1000000000# & " nm")

    ' === Disconnect ===
    Call EnableLocal(BOARD_ID, intAddrList())
End Sub


'===================================================
' Sub routine
' Send Remote Command
'===================================================
Sub SendGPIB(intAddr As Integer, strData As String)
    Call Send(BOARD_ID, intAddr, strData, NLend)
    If (ibsta And EERR) Then
        MsgBox " GP-IB device can't write"
    End If
End Sub


'===================================================
' Sub routine
' Recieve query data
'===================================================
Function RecieveGPIB(intAddr As Integer) As String
    Const READSIZE = 10000
    Dim strBuffer As String

    strBuffer = Space(READSIZE)
    RecieveGPIB = ""
    Do
        DoEvents

        Call Receive(BOARD_ID, intAddr, strBuffer, STOPend)
        If (ibsta And EERR) Then
            MsgBox " GP-IB device can't read."
            RecieveGPIB = ""
            Exit Function
        Else
            RecieveGPIB = RecieveGPIB & Left(strBuffer, ibcntl)
        End If
    Loop Until ((ibsta And EEND) = EEND)
End Function
```

Sample Program 2

Save an image of the instrument's screen to a BMP file, then use a file transfer command to load the file onto the PC. Save the image on the PC under the file name, "C:\test.bmp".

```
Const BOARD_ID = 0                     'GP-IB Interface card Address
Const osa = 1                          'OSA GP-IB Address

Private Sub Command1_Click()
   Dim intAddrList(31) As Integer
   Dim intData As Integer
   Dim lngDataSize As Long
   Dim strData As String
   Dim intI As Integer
   Dim byteData() As Byte
   Dim byteSaveData() As Byte
   Dim lngL As Long

   '----- GP-IB Interface setting
   ' send IFC
   Call SendIFC(BOARD_ID)

   ' assert th REN GPIB line
   intAddrList(0) = NOADDR
   Call EnableRemote(BOARD_ID, intAddrList())

   ' GPIB time out setting
   Call ibtmo(BOARD_ID, T30s)          'Time out = 30sec

   '----- send command to OSA
   Call SendGPIB(osa, "CFORM1")        ' Command mode set(AQ637X mode)

   Call SendGPIB(osa, ":mmem:stor:grap color,bmp,""test"",int")
                                       ' Save bmp file to internal memory
   Call SendGPIB(osa, ":mmem:data? ""test.bmp"",int")
                                       ' get file data from OSA
   lngDataSize = RecieveBinaryGPIB(osa, byteData())
                                       ' Recieve binary block data

   If byteData(0) <> Asc("#") Then     ' check first data
       MsgBox "Data format error"
       Exit Sub
   End If

   '----- calculate data size
   intData = byteData(1) - Asc("0")
   strData = ""
   For intI = 1 To intData
       strData = strData + Chr(byteData(intI + 1))
   Next intI
   lngDataSize = Val(strData)          ' data size

   '----- make save data
   ReDim byteSaveData(lngDataSize)
   For lngL = 0 To lngDataSize - 1
       byteSaveData(lngL) = byteData(lngL + intData + 2)
   Next lngL

   '----- save data to file
   Open "c:\test.bmp" For Binary As #1
       Put #1, , byteSaveData
   Close #1

   '----- Disconnect
   Call EnableLocal(BOARD_ID, intAddrList())
   MsgBox "Complete"
End Sub
```

```
'===================================================
' Sub routine
' Send Remote Command
'===================================================
Sub SendGPIB(intAddr As Integer, strData As String)
   Call Send(BOARD_ID, intAddr, strData, NLend)
   If (ibsta And EERR) Then
      MsgBox " GP-IB device can't write"
   End If
End Sub


'===================================================
' Sub routine
' Recieve Binary query data
'===================================================
Function RecieveBinaryGPIB(intAdr As Integer, byteArray() As Byte) As
Long
   Const READSIZE = 1200000           ' MAX 1.2MB
   Dim lngSize As Long
   Dim lngL As Long
   Dim lngPos As Long
   Dim ud As Integer
   Dim byteLow As Byte
   Dim byteHigh As Byte
   Dim strA As String

   Dim intDummy(READSIZE) As Integer
   lngSize = 0

'-----  open device
   ud = ildev(0, intAdr, 0, T30s, 1, 0)
   lngPos = 0

'-----  read data
   Do
      DoEvents
      Call ibrdi(ud, intDummy, READSIZE)
      If (ibsta And EERR) Then
         MsgBox "GP-IB device can't Read(GPIB:" & intAdr & ")"
         RecieveBinaryGPIB = 0
            Exit Function
      Else
         ReDim Preserve byteArray(lngPos + ibcntl + 2)
         For lngL = 0 To ibcntl / 2 - 1
            strA = Right("0000" & Hex(intDummy(lngL)), 4)
            byteHigh = Val("&H" + Left(strA, 2))
            byteLow = Val("&H" + Right(strA, 2))
            byteArray(lngPos) = byteLow
            byteArray(lngPos + 1) = byteHigh
            lngPos = lngPos + 2
         Next lngL
      End If
   Loop While (ibcntl = READSIZE)

   RecieveBinaryGPIB = lngPos
End Function
```

3.1 Connecting via Ethernet

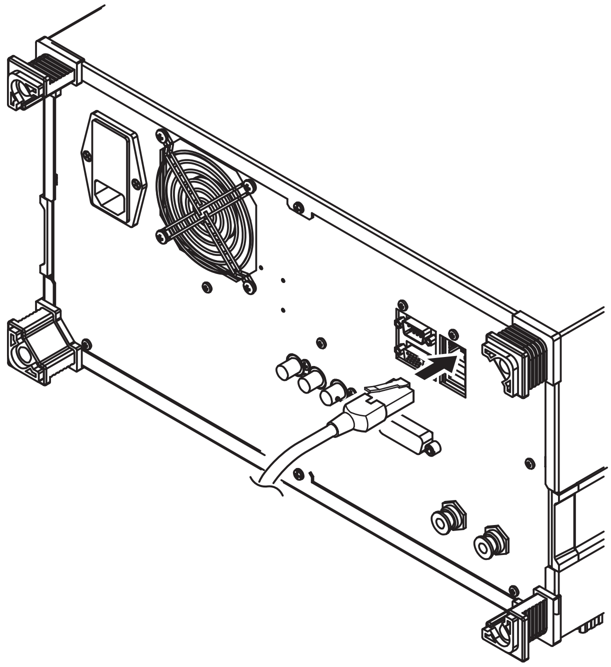
You can connect to a LAN using the Ethernet interface for control of the instrument from a PC.

Ethernet Interface Specifications

Communication ports:	1
Electromechanical specifications:	Conforms to IEEE802.3
Transmission method:	Ethernet (10BASE-T/100BASE-TX/1000BASE-T)
Transmission speed:	10 Mbps/100 Mbps/1000 Mbps
Communication protocol:	TCP/IP
Connector type:	RJ45
Port number used:	10001/tcp (default)

Connections

Connect a UTP (unshielded twisted-pair) cable or an STP (shielded twisted-pair) cable that is connected to another device to the ETHERNET port on the rear panel of the instrument.



Precautions When Making Connections

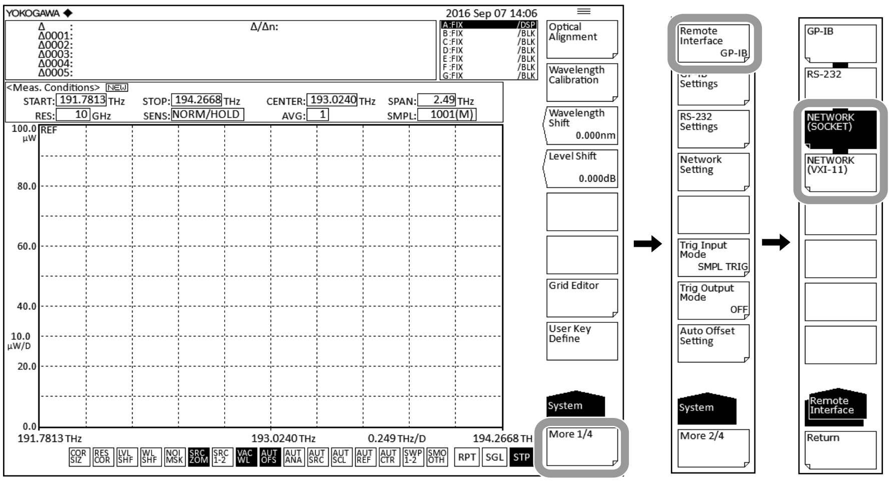
- Be sure to use a straight cable through a hub when connecting a PC to the instrument. Performance cannot be guaranteed if a 1-to-1 connection is made with a cross cable.
- When using a UTP (straight) cable, make sure that it is a category 5 cable.

3.2 Setting Up Ethernet

Procedure

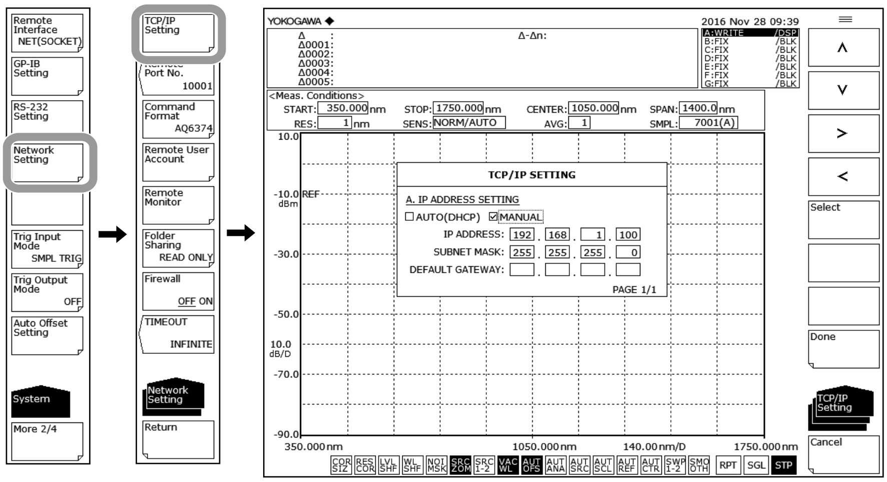
Selecting the Communication Interface

1. Press **SYSTEM**. The system setting menu is displayed.
2. Press the **More1/4** soft key. The communication interface setting menu is displayed.
3. Press the **Remote Interface** soft key. The setting menu for the interface to be used is displayed.
4. Press the **NETWORK(SOCKET)** or **NETWORK(VXI-11)** soft key to specify Ethernet as the communication interface.



Setting Up TCP/IP

5. Press the **Network Setting** soft key. The ethernet setting menu is displayed.
6. Press the **TCP/IP Setting** soft key. The TCP/IP setting menu is displayed.
7. **Using the** <, > soft keys, select AUTO (DHCP) or MANUAL.
8. Press the **Select** soft key. The item is selected.


TCP/IP SETTING

A. IP ADDRESS SETTING

□AUTO(DHCP) ☑MANUAL

IP ADDRESS: 192 . 168 . 1 . 100

SUBNET MASK: 255 . 255 . 255 . 0

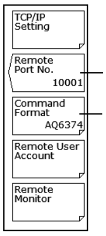
DEFAULT GATEWAY: . . .

PAGE 1/1

9. If you select MANUAL, enter the IP address, subnet mask, and default gateway. Using the arrow soft keys, select an input position, and press ENTER. If you selected AUTO, skip to step 10.
10. Enter a number using the **rotary knob** or the **<, >, ∧ , ∨ keys**, and press **ENTER**.
11. When all settings are entered, press the **DONE** soft key.

Setting the Remote Port Number (not used with the VXI-11)

12. Press the **Remote Port No.** soft key. The port number setting screen is displayed.
13. Enter a port number using the **rotary knob** or the **arrow keys**, and press **ENTER**.



Remote Port No. 10001

Command Format AQ6374

Remote port number setting

Command format setting

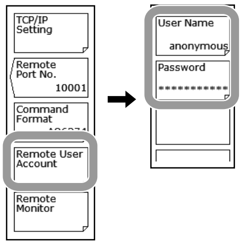
Setting the Command Format

14. Perform these steps if you will use AQ6317 commands.

    Press the **Command Format** soft key. The command format setting menu is displayed.
15. Normally, you will enter AQ6374. If you wish to use AQ6317 commands, enter AQ6317.

Setting the User Name and Password (not used with the VXI-11)

16. Press the **Remote User Account** soft key. The user name and password setting menu is displayed.



User Name anonymous

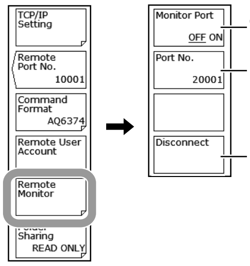
Password ***********

17. Press the **User Name** soft key. The user name setting screen appears. The default is anonymous.
18. Specify a user name using 11 alphanumeric characters or fewer.

    If the user name is set to anonymous, the password setting is not required.
19. Press the **Password** soft key. The password setting screen is displayed.
20. Specify a password using 11 alphanumeric characters or fewer.

Configuring the Remote Monitor Settings

21. Press the **Remote Monitor** soft key. The remote monitor setup menu appears.
22. Press the **Monitor Port** soft key. Each time you press the soft key, the setting toggles between ON and OFF. Remote monitoring is possible when the setting is ON.



Turns the monitor port on and off.

Set the port number (fixed).

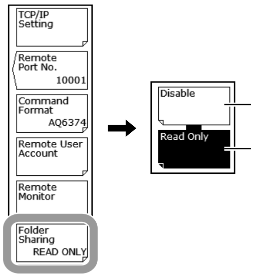
Disconnects the monitor connection

- **Disconnecting the Monitor Connection**

23. Press the **Disconnect** soft key. The monitor connection from the PC is disconnected.

Setting Directory Sharing

24. Press the **Folder Sharing** soft key. A directory sharing setup menu appears.
25. Press the **Read Only** soft key. The user area directory of the AQ6374 is shared (read only).



Disables directory sharing

Enables directory sharing

- **Disabling Directory Sharing**

26. Press the **Disable** soft key. The sharing of the user area directory is disabled.

Setting the Remote Timeout (not used with the VXI-11)

27. Press the **TIME OUT** soft key.
28. Enter the timeout period using the **rotary knob**, the **arrow keys**, or **numeric key**, and press **ENTER**.

Explanation

TCP/IP Settings

It is necessary to set up the IP address for correct use of the instrument.

If a DHCP server is provided on the network to which this instrument is connected, the IP address given to the instrument is automatically set. Thus, set the item IP ADDRESS SETTING under SYSTEM <Network Setting><TCP/IP Setting> to “AUTO.” Please ask your network administrator for details about network connections.

Note

- If you start the AQ6374 when it is connected to a network, it may take a few minutes for the start procedure to finish. (The progress of initialization is indicated at the bottom of the screen with indications from “STEP 1/9” to “STEP 9/9.”)
- When the start procedure is finished and the measurement screen appears, it may take a few more minutes before you can access the AQ6374 from a PC over the network. In addition, below the soft key may be unavailable for a certain time.
    - Remote Interface
    - Network Setting
    - System Information
    - Parameter Initialize

Remote Port No. (not used with the VXI-11)

Sets the port number for remote control via ETHERNET. (Default: 10001.)

User Authentication (not used with the VXI-11)

User authentication is required to connect to the instrument from a PC over an Ethernet network. If the user name is anonymous, a password is not required. This instrument supports plain text authentication and the MD5 Message Digest Algorithm by RSA Data Security, Inc.

Remote Monitoring

You can use the ETHERNET port to monitor the AQ6374 screen or control the AQ6374 from a PC over a network.

To use this feature, you need remote monitoring software (not included).

For information on remote monitoring software, contact your nearest YOKOGAWA dealer.

Sharing Directories

The user area directory of the AQ6374 internal memory can be shared on a PC.

When the user area directory is shared, the following files can be copied to the PC over the network.

You cannot save files to the AQ6374.

Timeout Period (not used with the VXI-11)

When a non-communication period reached the set period in a remote state, the communication is automatically disconnected to enter the local state.

The change in the timeout period resets the time elapsed.

You can set INFINITE (0 second) or 1 through 21600 seconds (six hours).

Remote Control Using Commands

The AQ6374 can be remote controlled using the LAN port.

For remote commands, use the same commands as those for control via the GP-IB interface.

The instrument also supports VXI-11 control.

Switching Interfaces

Select GP-IB, RS-232C, NETWORK (SOCKET), or NETWORK (VXI-11) as an interface to use for remote control. When you change the interface, the connection status is reset. Otherwise, the connection is kept open unless closed by the controller.

Remote Commands

As with GP-IB-based remote control, you can select the command format from the AQ6374 mode or from the AQ6317-compatible mode.

Interrupt by SRQ

An SRQ interrupt does not occur during LAN-based remote control.

Status Register

The status registers operate in the same manner as in remote control via the GP-IB interface. Using the “*SPOOL?” command dedicated for remote control using the LAN port allows you to read the status registers, as in the case with serial polling via the GP-IB interface.

*STB?:	When AQ6374 is the setting of the COMMAND FORMAT key
SPOLL?:	When AQ6317 is the setting of the COMMAND FORMAT key

Delimiter

The delimiter for LAN-based remote control is fixed to CR + LF.

Transmission of Talker Data

When the instrument receives talker data from an external PC, it sends the data to the external PC's buffer. It receives the external PC's buffer data and stores the query data.

Connection

The instrument can only be connected to one controller (an external PC or other device). If the instrument receives a connection request from a controller while already connected to another controller, the new connection is not opened and the existing connection is kept open.

Computer Name

The instrument’s computer name is as follows.

For the AQ6374

“6374@@@@@@@@@” (where “@@@@@@@@@” is the serial number)

Commands that are Necessary for Remote Control over the LAN

The authentication by OPEN command is required to remote control over the LAN.

Both the OPEN and CLOSE commands are also valid in AQ6317 mode.

OPEN

Function Sends the user name and starts user authentication.

Syntax
```
OPEN<wsp>"username"
username = the user name
```

Example
```
OPEN "yokogawa"
-> AUTHENTICATE CRAM-MD5.
```

Explanation Authentication is carried out with the OPEN command as follows.

For Plain Text Authentication

1. Send OPEN "username" to the AQ6374. The response message is received from the AQ6374.
2. Confirm that the received message is "AUTHENTICATE CRAM-MD5."
3. Send the password to the AQ6374 (anything can be input if the user name is anonymous).
4. If the message, "READY" is received from the AQ6374, authentication was successful. The AQ6374's REMOTE indicator lights, and sending of remote commands is enabled. If the user name and password are incorrect, authentication fails and the connections is closed.

For Encrypted Authentication

1. Send OPEN "username" to the AQ6374. The response message is received from the AQ6374.
2. Confirm that the received message is "AUTHENTICATE CRAM-MD5."
3. Send "AUTHENTICATE CRAM-MD5 OK" to the AQ6374. The response message (challenge string) is received from the AQ6374.
4. The received challenge string and password are processed with an MD5 hash algorithm (anything can be input if the user name is anonymous).
5. Send the returned hash data (as a 32-character hexadecimal string in lower case) to the AQ6374, and receive the response message.
6. If the message, "READY" is received from the AQ6374, authentication was successful. The AQ6374B's REMOTE indicator lights, and sending of remote commands is enabled. If the user name and password are incorrect, authentication fails and the connection is closed.

CLOSE

Function Closes the connection (turns it OFF), and switches to local mode.

Syntax
```
CLOSE
```

Example
```
CLOSE
```

3.3 Sample Program

Sample Program 1

Sending an invalid talker command to the AQ6374 and then receiving data with the instrument specified as a talker causes the GP-IB bus to stop because the instrument has no data to send. In this case, a GPIB timeout occurs, followed by recovery of the GP-IB bus.

The following shows an example of controlling the AQ6374 remotely using the Ethernet port. The sample program uses Visual Basic 6.0 as the programming language. The program sets the measurement conditions (center wavelength, span, sensitivity, and the sampling number) and then performs a sweep. After completing this sweep, the program executes a thresh-based spectrum width analysis and then outputs the results to the screen. The conditions are the same as those of the GP-IB sample program in section 2.6, “Sample Program.”

```
Private Sub AQ637XTEST()
   Dim intData As Integer
   Dim dblMeanWL As Double
   Dim dblSpecWd As Double
   Dim strData As String

   ' === Connect ===
   Winsock1.RemoteHost = "192.168.1.100"      ' OSA IP address
   Winsock1.RemotePort = 10001                ' OSA remote port num
   Winsock1.Connect

   ' === Wait to connect complete ===
   While (Winsock1.State <> sckConnected)
       DoEvents
   Wend

   ' === Authentication by OPEN Command ===
   SendLan "open ""anonymous"""
   ReceiveLan strData
   SendLan " "
   ReceiveLan strData
   If (Left(strData, 5) <> "ready") Then
       MsgBox "User authentication error."
       Exit Sub
   End If


   ' === Set the measurement parameter ===
   SendLan "*RST"                                 ' Setting initialize
   SendLan "CFORM1"                               ' Command mode set
                                                 (AQ637X mode)
   SendLan ":sens:wav:cent 1550nm"                ' sweep center wl
   SendLan ":sens:wav:span 10nm"                  ' sweep span
   SendLan ":sens:sens mid"                       ' sens mode = MID
   SendLan ":sens:sweep:points:auto on"           ' Sampling Point = AUTO

   ' === Sweep execute ===
   SendLan ":init:smode 1"                        ' single sweep mode
   SendLan "*CLS"                                 ' status clear
   SendLan ":init"                                ' sweep start
   ' === Wait for sweep complete ===
   Do
      SendLan ":stat:oper:even?"                  ' get Operation Event
                                                  Register
      ReceiveLan strData
      intData = Val(strData)
   Loop While ((intData And 1) <> 1)              ' Bit0: Sweep status

   ' === Analysis ===
   SendLan ":calc:category swth"                  ' Spectrum width
                                                  analysis(THRESH type)
   SendLan ":calc"                                ' Analysis Execute
   SendLan ":calc:data?"                          ' get data
   ReceiveLan strData

   ' === Capture analytical results ===
   dblMeanWL = Val(Left(strData, 16))             ' get mean wavelegnth
   dblSpecWd = Val(Mid(strData, 18, 16))          ' get spectrum width

   ' === Output the result to the screen ===
   MsgBox ("MEAN WL: " & dblMeanWL * 1000000000# & " nm" & vbCrLf & _
          "SPEC WD: " & dblSpecWd * 1000000000# & " nm")

   ' === Disconnect ===
   Winsock1.Close

   'Wait to disconnect complete
   While (Winsock1.State <> sckClosed)
      DoEvents
   Wend

End Sub

'===================================================
' Sub routine
' Send Remote Command
'===================================================
Sub SendLan(strData As String)
   Winsock1.SendData strData & vbCrLf
   DoEvents
End Sub

'===================================================
' Sub routine
' Receive query data
'===================================================
Sub ReceiveLan(strData As String)
   Dim strData2 As String

   strData = ""
   Do
      Winsock1.GetData strData2, vbString
      strData = strData + strData2
      DoEvents
   Loop While (Right(strData, 1) <> vbLf)
End Sub
```

Sample Program 2

Save an image of the instrument's screen to a BMP file, then use a file transfer command to load the file onto the PC. Save the image on the PC under the file name, "C:\test.bmp". The conditions are the same as the GP-IB sample program in section 2.6, "Sample Programs."

```
Const TIMEOUT = 1                                  ' time out(sec)

Private Sub cmdConnect_Click()
   Dim strData As String
   Dim byteData() As Byte
   Dim lngDataSize As Long

   '=== Connect ===
   If (ConnectLan("192.168.1.100", 10001) = False) Then
      MsgBox "Connection error"
      Winsock1.Close
      Exit Sub
   End If

   ' === Authentication by OPEN Command ===
   SendLan "open ""anonymous"""                    ' Send user name
   lngDataSize = ReceiveLan(strData)
   If (lngDataSize = -1) Then
      MsgBox "Data Receive Error"
      Winsock1.Close
      Exit Sub
   End If

   SendLan " "                                     ' Send password
   lngDataSize = ReceiveLan(strData)
   If (lngDataSize = -1) Then
      MsgBox "Data Receive Error"
      Winsock1.Close
      Exit Sub
   End If
   If (Left(strData, 5) <> "ready") Then
      MsgBox "User authentication error."
      Winsock1.Close
      Exit Sub
   End If

   '----- send command to OSA
   Call SendLan("CFORM1")                          ' Command mode
                                                   set(AQ637X mode)
   Call SendLan(":mmem:stor:grap color,bmp,""test"",int")
            ' Save bmp file to internal memory
   Call SendLan(":mmem:data? ""test.bmp"",int")   ' get file data from
                                                   OSA
      lngDataSize = ReceiveBinaryLan(byteData())        ' Recieve binary block data

   '----- save data to binary file
   Open "c:\test.bmp" For Binary As #1
      Put #1, , byteData
   Close #1

   '----- Disconnect
   Winsock1.Close

   'Wait to disconnect complete
   While (Winsock1.State <> sckClosed)
      DoEvents
   Wend
   MsgBox "Complete"
End Sub
```

```
'===================================================
' Sub routine
' Connect OSA via ETHERNET
'      in: strIP  IP Address(Ex. "192.168.1.100") or Computer Name
'                 intPort    port number (Ex. 10001)
'      out:       none
'      ret:       OK/NG  true: OK, false: NG
'===================================================
Function ConnectLan(strIP As String, intPort As Integer) As Boolean
   Dim sglStart As Single
   Dim sglEnd As Single
   Dim sglNow As Single
   Dim bConnect As Boolean

   sglStart = Timer()
   sglEnd = sglStart + TIMEOUT
   bConnect = True

   ' === Connect ===
   Winsock1.RemoteHost = strIP                   ' OSA IP address
   Winsock1.RemotePort = intPort                 ' OSA remote port num
   Winsock1.Connect

   ' === Wait to connect complete ===
   While ((Winsock1.State <> sckConnected) And (bConnect = True))
      DoEvents
      ' Timeout check
      sglNow = Timer()
      If (sglNow < sglStart) Then sglNow = sglNow + 86400
      If sglNow >= sglEnd Then bConnect = False
    Wend

   '-----  return value set
   ConnectLan = bConnect
End Function


'===================================================
' Sub routine
' Send Remote Command
'===================================================
Sub SendLan(strData As String)
   Winsock1.SendData strData & vbCrLf
   DoEvents
End Sub


'===================================================
' Sub routine
' Receive query data
'      in:    none
'      out:   strData       Receive data
'      ret:   Receive data size (Error: -1)
'===================================================
Function ReceiveLan(strData As String) As Long
   Dim strData2 As String
   Dim sglStart As Single
   Dim sglEnd As Single
   Dim sglNow As Single
   Dim bTimeout As Boolean

   sglStart = Timer()
   sglEnd = sglStart + TIMEOUT
   bTimeout = False
```

```
    strData = ""
    Do
        ' data receive
        DoEvents
        Winsock1.GetData strData2, vbString
        strData = strData + strData2

        ' Timeout check
        sglNow = Timer()
        If (sglNow < sglStart) Then sglNow = sglNow + 86400
        If sglNow >= sglEnd Then bTimeout = True
    Loop While ((Right(strData, 1) <> vbLf) And (bTimeout = False))

    ' return value set
    If bTimeout = True Then
        ReceiveLan = -1
    Else
        ReceiveLan = Len(strData)
    End If

End Function


'======================================================================
' Sub routine
' Recieve Binary query data
'      in:    none
'      out:   byteArray     Receive data (byte array)
'      ret:   Receive data size (Error: -1)
'======================================================================
Function ReceiveBinaryLan(byteArray() As Byte) As Long
    Dim lngPos As Long
    Dim lngTempPos As Long
    Dim bData As Byte
    Dim intI As Integer
    Dim intJ As Integer
    Dim strA As String
    Dim lngDataLength As Long
    Dim byteDummy() As Byte
    Dim sglStart As Single
    Dim sglEnd As Single
    Dim sglNow As Single
    Dim bTimeout As Boolean

    sglStart = Timer()
    sglEnd = sglStart + TIMEOUT
    bTimeout = False
    '---------------------------------------------
    '   Header block
    '---------------------------------------------
    Call ReadIPBin(bData)                              ' Receive 1byte

    If bData = Asc("#") Then
         Call ReadIPBin(bData)                         ' Receive 1byte

    intI = bData - Asc("0")
    strA = ""
        For intJ = 0 To intI - 1
            Call ReadIPBin(bData)                      ' Receive 1byte
            strA = strA + Chr(bData)
            Next intJ
            lngDataLength = Val(strA)                  ' block data size

        ReDim byteArray(lngDataLength)
```

```
            '------------------------------
            ' Recieve binary data block
            '------------------------------
            lngPos = 0
            lngTempPos = 0
            ReDim byteDummy(lngDataLength)
            Winsock1.GetData byteDummy, vbArray + vbByte, lngDataLength
                                                    ' Receive binary data

            Do
                DoEvents
                If (lngTempPos > UBound(byteDummy)) Then
                    Winsock1.GetData byteDummy, vbArray + vbByte, lngDataLength
                                                    ' Continue to receive
                    lngTempPos = 0
                    Else
                    byteArray(lngPos) = byteDummy(lngTempPos)
                    lngPos = lngPos + 1
                    lngTempPos = lngTempPos + 1
                End If

                'Timeout check
                sglNow = Timer()
                If (sglNow < sglStart) Then sglNow = sglNow + 86400
                If sglNow >= sglEnd Then bTimeout = True
            Loop Until ((lngPos = lngDataLength) Or (bTimeout = True))
            End If

        ' return value set
        If bTimeout = True Then
             ReceiveBinaryLan = -1
        Else
            ReceiveBinaryLan = lngDataLength
        End If
    End Function


    '=====================================================================
    '  Read binary data(1byte)
    '=====================================================================
    Sub ReadIPBin(byteData As Byte)
        Dim sglStart As Single
        Dim sglEnd As Single
        Dim sglNow As Single
        Dim bTimeout As Boolean

        sglStart = Timer()
        sglEnd = sglStart + TIMEOUT
        bTimeout = False

        '----- wait until data received or timeout
        Do
            DoEvents

            'Timeout check
            sglNow = Timer()
            If (sglNow < sglStart) Then sglNow = sglNow + 86400
            If sglNow >= sglEnd Then bTimeout = True
        Loop Until ((Winsock1.BytesReceived > 1) Or (bTimeout = True))

        Winsock1.GetData byteData, vbByte, 1           ' 1byte read
    End Sub
```

4.1 Connecting via the Serial (RS-232) Interface

Serial Interface Functions and Specifications

Receive Function

You can enter the same settings as can be entered with front panel keys.
A settings output request is received.

Send Function

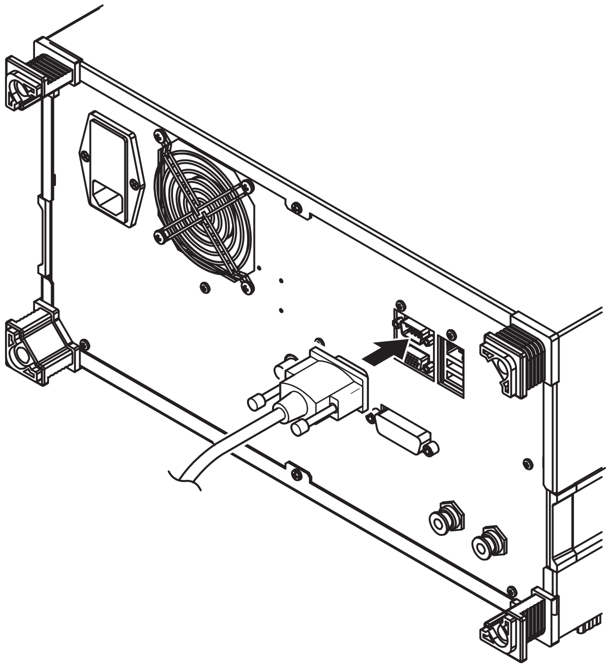
You can output settings and measured results.

Serial (RS-232) Interface Specifications

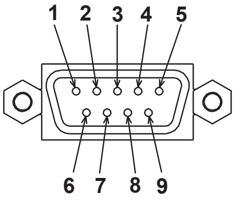
Electrical characteristics:	Conforms to the EIA-574 standard (EIA-232 (RS-232), 9-pin)
Connection type:	Point-to-point
Communication method:	Full duplex
Synchronization method:	Start-stop synchronization
Baud rate:	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Start bit:	1 bit, fixed
Data length:	8 bit, fixed
Parity:	Odd, Even, or None
Stop bit:	1 bit, fixed
Connector:	DELC-J9PAF-13L6 (JAE or equivalent)
Flow control:	Hardware handshaking using RS/CS or Non (selectable).

Connection

Make the connection as shown in the figure below.



Connector and Signal Names



DELC-J9PAF-13L6 or equivalent

2 RD (received data): Data received from the PC.
Signal direction....input

3 SD (send data): Data sent to the PC.
Signal direction....output

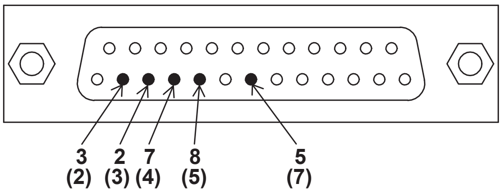
5 SG (signal ground): Ground for the signal.

7 RS (request to send): Handshaking method when receiving data from the PC.
Signal direction....output

8 CS (clear to send): Handshaking method when sending data to the PC.
Signal direction....input

* Pins 1, 4, 6, and 9 are not used.

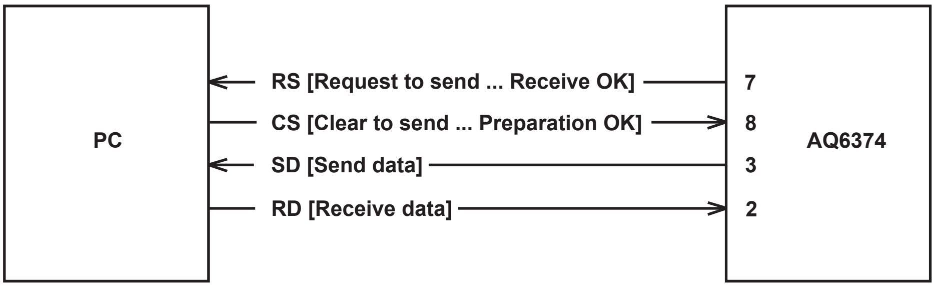
9-Pin to 25-pin Adapter and Signal Names



Numbers in parentheses are the pin numbers of the 25-pin connector.

Signal Direction

The directions of signals used by the instrument's serial interface are shown in the figure below.


PC
RS [Request to send ... Receive OK]
CS [Clear to send ... Preparation OK]
SD [Send data]
RD [Receive data]
7
8
3
2
AQ6374

List of RS-232 Standard Signals and JIS and CCITT Cable Addresses

Signal Chart

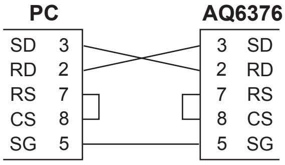
Pin Number (9-Pin Connector)	Code			Name
	RS-232	CCITT	JIS	
5	AB (GND)	102	SG	Signal ground
3	BA (TXD)	103	SD	Send data
2	BB (RXD)	104	RD	Receive data
7	CA (RTS)	105	RS	Request to send
8	CB (CTS)	106	CS	Clear to send

Signal Wire Connection Example

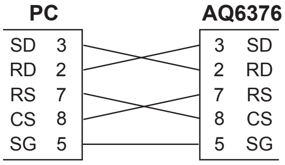
Pin numbers are for 9-pin connectors.

In most cases, use a cross cable.

• OFF-OFF/XON-XON



• Hard(CS-RS)



4.2 Remote Control Using Commands

The AQ6374 can be controlled remotely using the RS-232 port. When controlling the instrument remotely, use a cross cable to connect the instrument to the PC. Also, remote commands are the same as for remote control via GP-IB.

Interrupt by SRQ

An SRQ interrupt does not occur during RS-232-based remote control.

Status Registers

The status registers operate in the same manner as in remote control via the GP-IB interface. Using the “*STB?” or “SPOLL?” command dedicated for remote control using the LAN port allows you to read the status registers, as in the case with serial polling via the GP-IB interface.

*STB?:	When AQ6374 is the setting of the Command Format key
SPOLL?:	When AQ6317 is the setting of the Command Format key

Delimiter

The delimiter for RS-232-based remote control is fixed to CR + LF.

Transmission of Talker Data

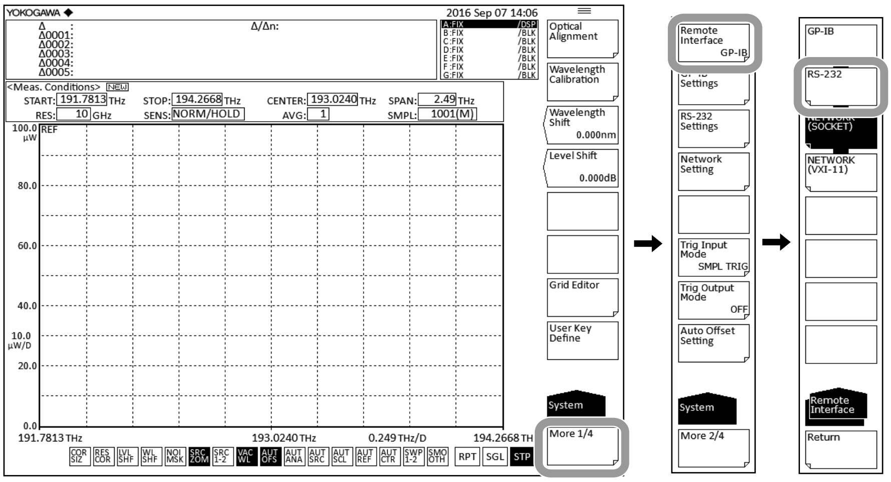
When the instrument receives talker data from an external PC, the data is sent to the external PC’s buffer. It receives the external PC's buffer data and stores the query data.

4.3 Setting Up RS-232

Procedure

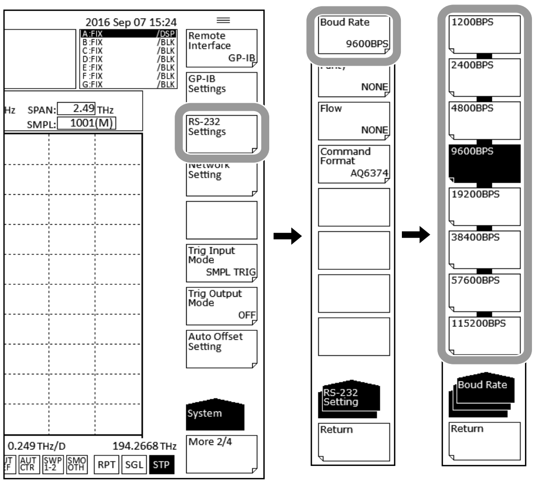
Selecting the Communication Interface

1. Press **SYSTEM**. The system setting menu is displayed.
2. Press the **More1/4** soft key. The communication interface setting menu is displayed.
3. Press the **Remote Interface** soft key. The setting menu for the interface to be used is displayed.
4. Press the **RS-232** soft key to specify **RS-232** as the communication interface.



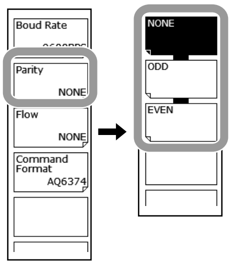
Setting the Baud Rate

5. Press the **RS-232 Settings** soft key. The RS-232 setting menu is displayed.
6. Press the **Boud Rate** soft key. The baud rate setting menu is displayed.
7. Press the soft key corresponding to the desired baud rate setting. The baud rate is set.



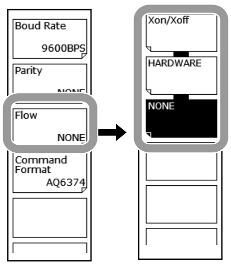
Setting the Parity

8. Press the **Parity** soft key. The parity setting menu is displayed.
9. Press the soft key corresponding to the desired parity setting. The parity is set.



Setting the Flow Control

10. Press the **Flow** soft key. The flow control setting menu is displayed.
11. Press the soft key corresponding to the desired flow control setting. The flow control is set.



Setting the Command Format

12. Perform these steps if you will use AQ6317 commands.

    Press the **Command Format** soft key. The command format setting menu is displayed.
13. Normally, you will enter AQ6374. If you wish to use AQ6317 commands, enter AQ6317.

Explanation

The settings below are used when entering the settings that can be entered using the instrument's panel keys from a controller, or when outputting settings or waveform data to the controller.

Baud Rate Setting

Select a baud rate from the following.
1200 bps, 2400 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, 57600 bps, or 115200 bps

Parity Rate Setting

Select a parity from the following.
NONE, ODD, or EVEN

Flow Control Setting

Select a Transmission data control-Receive data control from the following.
Xon/Xoff, HARDWARE, NONE

Setting the Command Format

Normally, you will enter AQ6374 mode.
If you wish to use the commands of the AQ6317 (another product in the series), enter AQ6317. See the appendix for AQ6317 commands that are compatible with the AQ6317.

5.1 Status Registers

This instrument is equipped with the status registers shown in the table below. See the next page for a diagram of all status registers.

This instrument has the following status registers defined by IEEE 488-2 and SCPI:

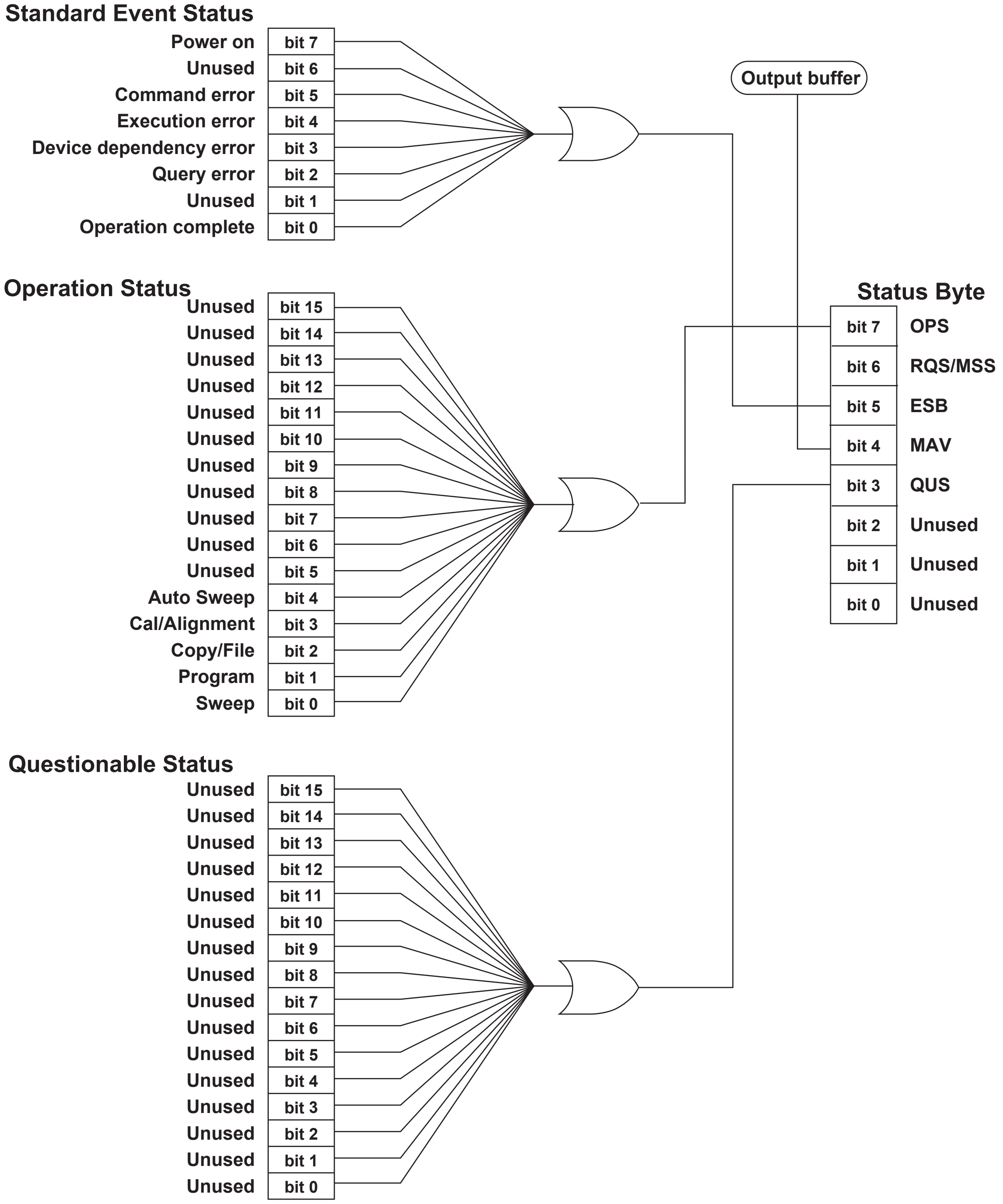
- Status byte registers
- Standard event registers
- Operation status registers
- Questionable status registers

Also, this instrument has an operation status bit (OPS) and a questionable status bit (QUS), each of which contains the summary information of each piece of register information, as the extension bits of the status byte register.

List of Status Registers

Register Name	Description
Status byte registers	Register defined by IEEE 488.2
STB: Status Byte Register	Same as the above
SRE: Service Request Enable Register	Same as the above
Standard event registers	Register defined by IEEE 488.2
ESR: Standard Event Status Register	Same as the above
ESE: Standard Event Status Register	Same as the above
Operation status registers	Provides information on operation execution (such as being swept, or under calibration).
Operation Event Register	A register indicating the presence/absence of an event. Event will be latched.
Operation Event Enable Register	A condition mask register used when the summary bit (OPS) is created.
Questionable status registers	Not assigned yet.
Questionable Event Register	A register indicating the presence/absence of an event. An event will be latched.
Questionable Event Enable Register	A condition mask register used when the summary bit (QUS) is created.

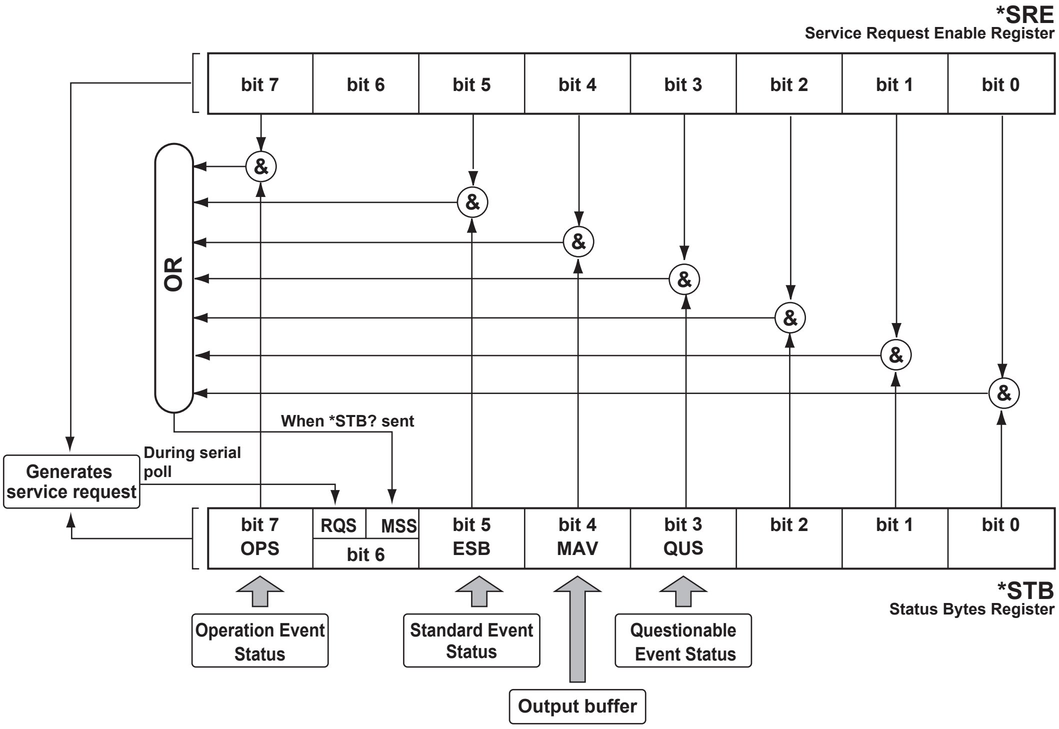
Status Register Overview Diagram



5.2 Status Byte Registers

Structure

The structure of the status byte registers is shown below. The contents and actions of these registers comply with the IEEE 488.2 standards.
Also, the AQ6374 also provides the extended OPS and QUS bits to the status byte register.



Status Byte Register Contents

Bit	Event Name	Description	Decimal Value
Bit 7	OPS	Summary bit of operation status	128
Bit 6	RQS, MSS	“1” if there is more than one service request	64
Bit 5	ESB	Summary bit of standard event status register	32
Bit 4	MAV	“1” if the output buffer contains data	16
Bit 3	QUS	Summary bit of questionable status	8
Bit 2	None	Not used (always 0)	0
Bit 1	None	Not used (always 0)	0
Bit 0	None	Not used (always 0)	0

Status Byte Register

Read

This register can be read by a serial poll or the common *STB? query. Note that the information of bit 6 changes with a different reading method.

- When read by serial polling
  An RQS message is read as bit 6 information.
  After reading, the RQS message will be cleared.
- When read by an *STB? common query
  An MSS summary message is read as bit 6 information.
  Even after reading, the MSS message will be held.

Bits other than bit 6 do not change.
The read action complies with the IEEE 488.2 standard.

Write

The contents of the register will be rewritten only when the status of an assigned status data structure has been changed. The write action complies with the IEEE 488.2 standard.

Clear

All event registers and queues, not including the output queues and MAV bit, will be cleared by the common *CLS command.
The clear action complies with the IEEE 488.2 standard.

Service Request Enable Register

Read

This register can be read by the common *SRE? query.
The value of bit 6, an unassigned bit, is always “0.” The contents of the register are not cleared even when read. The read action complies with the IEEE 488.2 standard.

Write

This register can be written by the common *SRE command.
The set value of bit 6, an unassigned bit, is always ignored. The write action complies with the IEEE 488.2 standard.

Clear

This register will be cleared under any of the following conditions.

- Data “0” is set using the common *SRE command.
- Power ON

The contents of the register are not cleared in the following cases.

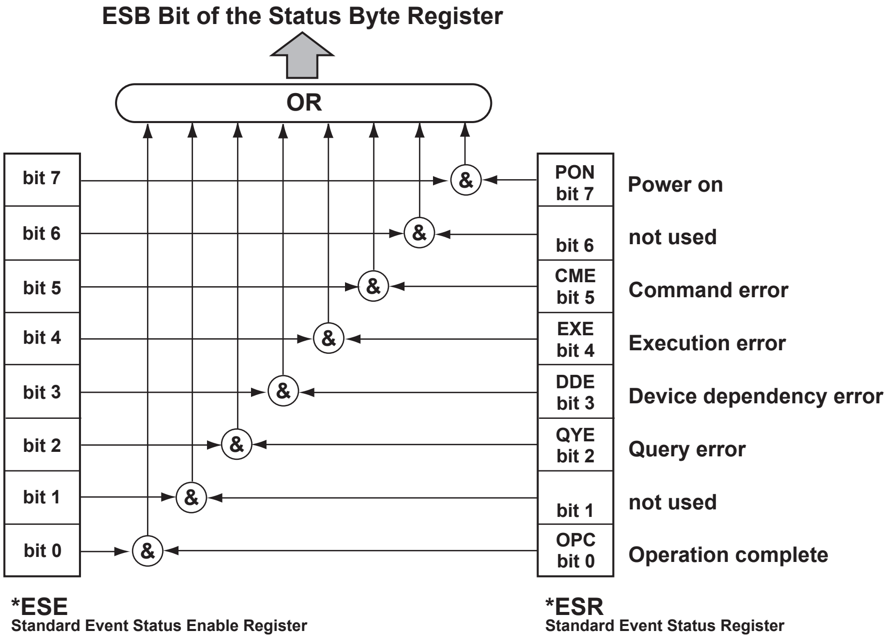
- Receipt of the *RST command
- Receipt of the *CLS command
- Device clear (DCL, SDC)

The clear action complies with the IEEE 488.2 standard.

5.3 Standard Event Status Registers

Structure

The structure of the standard event status registers is shown below. The contents and actions of the registers comply with the IEEE 488.2 standards.



Contents of the Standard Event Status Registers

Bit	Event Name	Description	Decimal Value
Bit 7	PON (Power ON)	Power is turned ON. Set to “1” at startup.	128
Bit 6	None	Not used (always 0)	0
Bit 5	CME (command error)	A syntax error or unrecognizable command is detected. GET is encountered between the 1st byte of a program message and the program message terminator.	32
Bit 4	EXE (Execution error)	Program data following the program header is out of the effective range. Receipt of a program message contradictory to device state.	16
Bit 3	DDE (Device-specific error)	Error caused by an event other than CME, EXE, or QYE.	8
Bit 2	QYE (Query error)	Access to an output queue was made with no output existing. Output queue data was lost.	4
Bit 1	None	Not used (always 0)	0
Bit 0	OPC (operation complete)	Completion of command action: Enabled only when *OPC is received Disabled if *OPC? is received	1

Standard Event Status Register

Read

This register can be read by the common *ESR? query.

Its contents will be cleared when read. The read action complies with the IEEE 488.2 standard.

Write

Contents of the register can be cleared. The register can be cleared but not written to.

Clear

This register will be cleared under any of the following conditions.

- Common *CLS command
- Common *ESR? query

The clear action complies with the IEEE 488.2 standard.

Standard Event Status Enable Register

Read

This register can be read by the common *ESE? query.

The read action complies with the IEEE 488.2 standard.

Write

This register can be written by the common *ESE command.

The write action complies with the IEEE 488.2 standard.

Clear

This register will be cleared under any of the following conditions.

- Data “0” is set using the common *ESE command.
- Power ON

The register cannot be cleared in the following cases.

- Receipt of the *RST command
- Receipt of the *CLS command
- Device clear (DCL, SDC)

The clear action complies with the IEEE 488.2 standard.

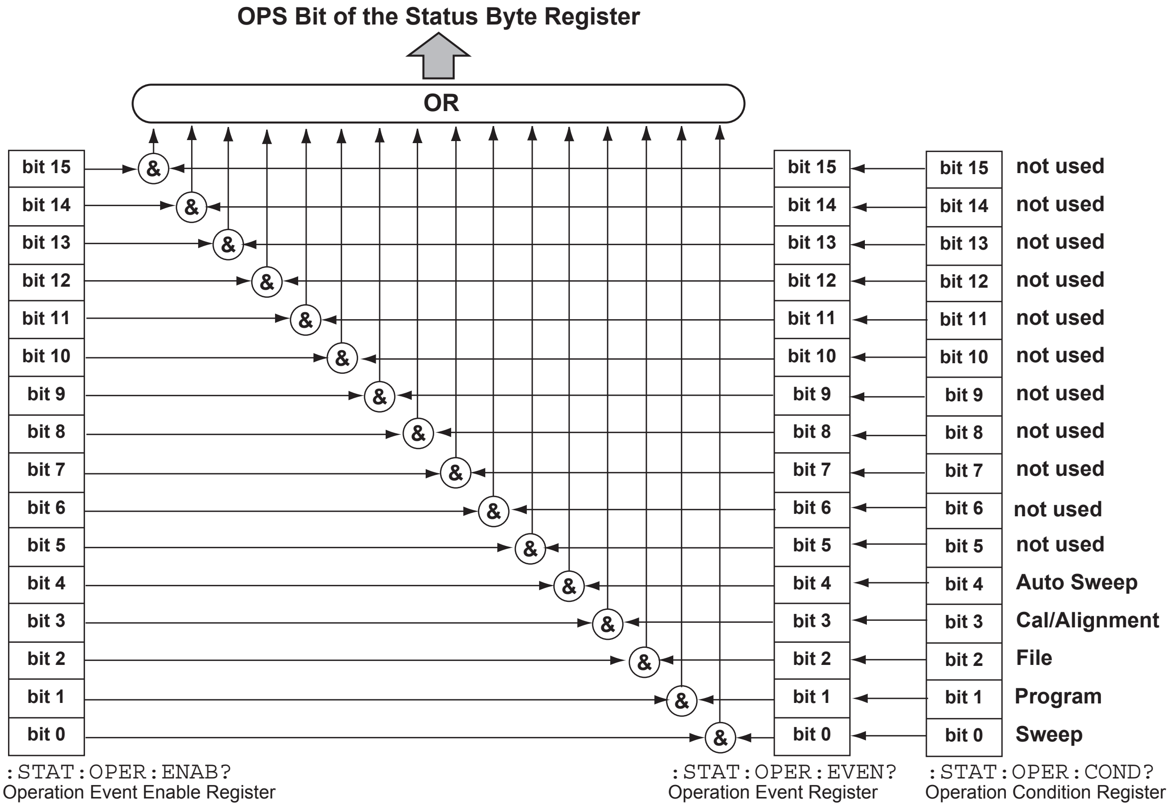
5.4 Operation Status Registers

Operation status registers report the operation status of the instrument. The operation condition registers indicate the instrument’s condition. A change in an operation condition register is latched into the operation event register. The user can refer to the operation event register to view changes in the operation status. The summary information of the instrument event register is set to the OPS bit of the status byte register. In this case, only statuses corresponding to bits specified as “1” in the operation enable register are included in the summary information.

Structure

The structure of the operation status register is shown below.

Structure of the Operation Status Register



Contents of the Operation Status Register

Bit	Event Name	Description	Decimal Value
Bit 15	Not used	Spare (always 0)	0
Bit 14	Not used	Spare (always 0)	0
Bit 13	Not used	Spare (always 0)	0
Bit 12	Not used	Spare (always 0)	0
Bit 11	Not used	Spare (always 0)	0
Bit 10	Not used	Spare (always 0)	0
Bit 9	Not used	Spare (always 0)	0
Bit 8	Not used	Spare (always 0)	0
Bit 7	Not used	Spare (always 0)	0
Bit 6	Not used	Spare (always 0)	0
Bit 5	Not used	Spare (always 0)	0
Bit 4	Auto Sweep	Completion of auto sweep running action	16
Bit 3	Cal/Alignment	Completion of wavelength calibration, alignment or resolution calibration	8
Bit 2	File	Completion of file operation	4
Bit 1	Program	Completion of execution of the program functions	2
Bit 0	Sweep	Completion of a sweep	1

Operation Condition Register

Read

This register can be read by the :STATus:OPERation:CONDition? query command.
Its contents will not be cleared even when read.

Write

The register sets or resets a bit corresponding to a change in the status of the instrument only when that change occurs. It cannot be written to.

Clear

The register cannot be cleared.

Operation Event Register

Read

This register can be read by the :STATus:OPERation[:EVENt?] query command.
Its contents will be cleared when read.

Write

Contents of the register can be cleared. The register can be cleared but not written to.
<Clear>
This register will be cleared under any of the following conditions.

- A read using the :STATus:OPERation[:EVENt?] query command
- An initialization by the :STATus:PRESet command
- The *CLS common command
- Power ON
- Operation event enable register

Read

This register can be read by the :STATus:OPERation:ENABle? query command.

Write

The register can be written by the :STATus:OPERation:ENABle command.

Clear

This register will be cleared under any of the following conditions.

- Data “0” is set by the :STATus:OPERation:ENABle command.
- Power ON

The register cannot be cleared in the following cases.

- Receipt of the *RST command
- Receipt of the *CLS command
- Device clear (DCL, SDC)

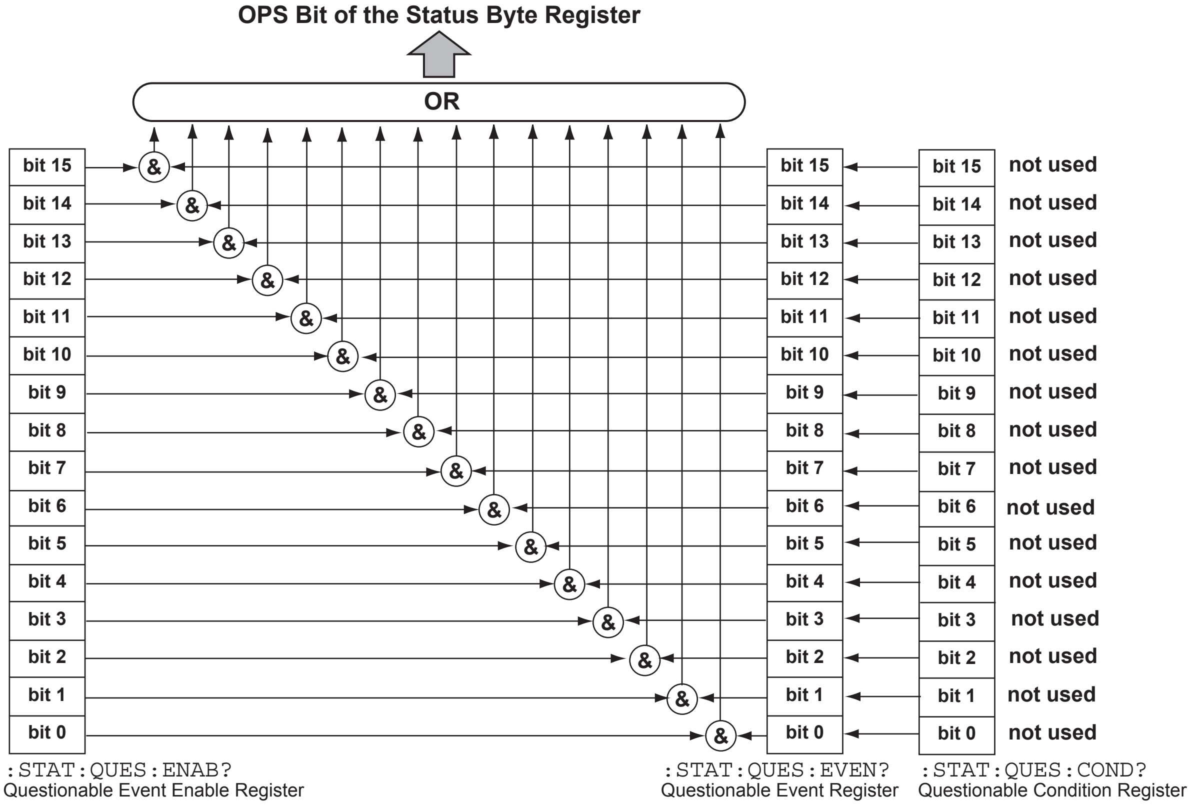
5.5 Questionable Status Registers

The questionable status registers report the questionable status of the instrument. All bits of these registers are unassigned. However, the register read/write operations are performed normally. The summary information of an event register will be set to the QUS bit of the status byte register.

Structure

The structure of the questionable status registers is shown below.

Structure of the Questionable Status Registers



Contents of the Questionable Status Registers

Bit	Event Name	Description	Decimal Value
Bit 0–15	Not used	Spare (always 0)	0

Questionable Condition Register

Read

This register can be read by the :STATus:QUEStionable:CONDition? query command. Its contents will not be cleared even when read.

Write

The register sets or resets a bit corresponding to a change in the status of the instrument only when that change occurs. It cannot be written to.

Clear

The register cannot be cleared.

Questionable Event Register

Read

This register can be read by the :STATus:QUEStionable[:EVENt?] query command.
Its contents will be cleared when read.

Write

Contents of the register can be cleared. The register can be cleared but not written to.

Clear

This register will be cleared under any of the following conditions.

- A read using the :STATus:QUEStionable[:EVENt?] query command
- Initialization by the :STATus:PRESet command
- Common *CLS command
- Power ON

Questionable Event Enable Register

Read

This register can be read by the :STATus:QUEStionable:ENABle? query command.

Write

The register can be written to by the :STATus:QUEStionable:ENABle command.

Clear

This register will be cleared under any of the following conditions.

- Data “0” is set using the :STATus:QUEStionable:ENABle command.
- Power ON

The register cannot be cleared in the following cases.

- Receipt of the *RST command
- Receipt of the *CLS command
- Device clear (DCL, SDC)

6.1 Rules of Syntax and Command Types

The following information is intended for the common commands and instrument-specific commands contained in this manual. Measured values and parameters are all sent and received using ASCII characters, not including special commands.

Description of Rules of Syntax

Rule	Description
|	Indicates that one of the elements in a list should be selected.
	E.g.: `A|B|C` = A, B, or C is used
[ ]	An item in square brackets is specified as desired.
{ }	An item in curly brackets can be specified multiple times within a command.
<wsp>[1]	Space
<integer>	Integer
<NRf>	Exponent indicating value
<"file name">	A file name can be a maximum of 56 characters, including extensions, excluding the directory part. Enclose a character string using double quotations (" ").
<trace name>	Trace name (TRA|TRB|TRC|TRD|TRE|TRF|TRG)
<marker>	Marker number (0: moving marker, 1 to 1024: fixed markers)
<"string">	Character string Enclose a character string using double quotations (" ").

1. Regarding white space (<wsp>):
White space is defined as a character corresponding to 00h to 20h (not including 0Ah (LF)) of the ASCII character sets. Aside from inserting it between a command and parameters (when specifying parameters) or using it as space in a character string such as a file name in a parameter, white space can be inserted as desired to make a program legible.

Types of Commands

This unit's commands can be classified into the following three types:

Sequential Commands

- These commands are the most general commands.
- The action of another command is not performed until the running of a sequential command is complete.
- Another action is not started until the running of the other command is complete.

Overlappable Commands

- An overlappable command allows execution of an overlapping command while it is being run.
Ex. of command: `:INITialte` Makes a sweep.

Overlapping Commands

- An overlapping command can be executed while an overlappable command is being run.
- These commands cannot be executed while a sequential command is being executed or if it has not yet been processed.
Ex. of command: `:ABORt` Stops measurement or calibration action.
`*STB?` Reads status byte.

Collective Transmission of Multiple Commands

You can create a command string using the commands described in section 6.5, “Common Commands,” and section 6.6, “Instrument-Specific Commands” and send it to the instrument. If multiple commands are written in a single output statement by using a semicolon “;” to delimit each command, the commands will be executed in the order in which they have been written.

Format of a Remote Command

Short and Long Forms

The instrument’s GP-IB commands support both short and long forms.
For the commands contained in this manual, the part written in capital letters is the short form of the command concerned. The short form of the `INITiate` command is `INIT`.

Upper- and Lower-Case Letters

The instrument does not distinguish between upper- and lower-case letters.
Return values are all in upper-case letters.

Grouping of SCPI Commands Using a Subsystem

The instrument supports the subsystem-based grouping of the SCPI commands. Commands belonging to the same sub-system and existing at the same tree of the hierarchical structure of the subsystem can be sent in combination. In this case, each command should be delimited by a semicolon.

List of GP-IB commands used in examples

```
:SENSe     :SETTing
                :ATTenuator
           :WAVelength
                :STOP
                :STARt
```

- `SENSe:WAVelength:STARt 1500NM;STOP 1600NM          (Y)`
- `SENSe:WAVelength:STARt 1500NM;ATTenuator ON        (X)`
  (Reason: They are not in the same hierarchy.)
- `SENSe:WAVelength:STARt 1500NM;:STOP 1600NM         (X)`
  (Reason: A colon “:” is unnecessary after a semicolon “;”.)

Numerics

- This instrument supports multiple notation methods when receiving a numeric(s).
- This instrument uses only the basic units when transmitting a numeric(s).

The number of digits for the real part is fixed to a one digit integer (with a sign) and eight digits for decimal places. The number of digits for the exponential part is fixed to 3.

Ex.: Receivable numerics (in case of 1550 nm)
1550 nm, 1.55 um, 1550E-9, 1.55E-6, and others

Ex.: Transmittable numerics (in case of 1550 nm)
+1.55000000E-006 only

- If a received numeric has a precision higher than the range of numerics handled inside this unit, lower decimal places will be rounded off rather than being discarded.
- This instrument can handle the following multiplier suffixes:

Multiplier	Mnemonic	Multiplier	Mnemonic
1E18	EX (exa)	1E-3	M (milli)
1E15	PE (peta)	1E-6	U (micro)
1E12	T (tera)	1E-9	N (nano)
1E9	G (giga)	1E-12	P (pico)
1E6	MA (mega)	1E-15	F (femto)
1E3	K (kilo)	1E-18	A (atto)

Specification of Parameters in a Command

To use parameters in a command, a space must be placed between the command and parameters. Each parameter is delimited by a comma “,”. A space may also be placed before and after a comma to make the command legible.

AQ6317-Compatible Commands

The instrument supports AQ6317-compatible GP-IB commands. When using AQ6317-compatible GP-IB commands, call up the **SYSTEM** menu using the SYSTEM key and place the instrument in AQ6317-compatible mode.

Differences from the AQ6370

This instrument’s remote commands differ from those of the AQ6370 in the following respects.

1. *IDN query talker data

AQ6374: "YOKOGAWA, AQ6374,----"

2. “CHOP” was eliminated from the <Chop Mode> settings.

If the AQ6374 receives a command that specifies "CHOP", it is treated as "SWITCH".

6.2 Table of Correspondence between Soft Keys and Remote Commands

The tables below list the remote commands that correspond to the soft keys used when manipulating the various settings of the instrument.

SWEEP

Function	Control Command
Auto	`:INITiate:SMODe<wsp>AUTO|3;:INITiate`
Repeat	`:INITiate:SMODe<wsp>REPeat|2;:INITiate`
Single	`:INITiate:SMODe<wsp>SINGle|1;:INITiate`
Stop	`:ABORt`
Segment Measure	`:INITiate:SMODe<wsp>SEGment|4;:INITiate`
Segment Point*****	`:SENSe:SWEep:SEGMent:POINts<wsp><integer>`
Sweep Marker L1-L2 ON/OFF	`:SENSe:WAVelength:SRANge<wsp>OFF|ON|0|1`
Sweep Interval *****sec	`:SENSe:SWEep:TIME:INTerval<wsp><integer>[SEC]`

CENTER

Function	Control Command
Center Wavelength ****.***nm	`:SENSe:WAVelength:CENTer<wsp><NRf>[M]`
Center Frequency ***.****THz	`:SENSe:WAVelength:CENTer<wsp><NRf>[HZ]`
Start Wavelength ****.***nm	`:SENSe:WAVelength:STARt<wsp><NRf>[M]`
Start Frequency ***.****THz	`:SENSe:WAVelength:STARt<wsp><NRf>[HZ]`
Stop Wavelength ****.***nm	`:SENSe:WAVelength:STOP<wsp><NRf>[M]`
Stop Frequency ***.****THz	`:SENSe:WAVelength:STOP<wsp><NRf>[HZ]`
Peak WL -> Center	`:CALCulate:MARKer:SCENter`
Auto Center ON/OFF	`:CALCulate:MARKer:MAXimum:SCENter:AUTO<wsp>OFF|ON|0|1`
View Scale -> Measure	`:DISPlay[:WINDow]:TRACe:X[:SCALe]:SMSCale`

SPAN

Function	Control Command
Span Wavelength****.*nm	`:SENSe:WAVelength:SPAN<wsp><NRf>[M]`
Span Frequency***.**THz	`:SENSe:WAVelength:SPAN<wsp><NRf>[HZ]`
Start Wavelength****.***nm	`:SENSe:WAVelength:STARt<wsp><NRf>[M]`
Start Frequency***.****THz	`:SENSe:WAVelength:STARt<wsp><NRf>[HZ]`
Stop Wavelength****.***nm	`:SENSe:WAVelength:STOP<wsp><NRf>[M]`
Stop Frequency***.****THz	`:SENSe:WAVelength:STOP<wsp><NRf>[HZ]`
0nm Sweep Time**sec	`:SENSe:SWEep:TIME:0NM<wsp><integer>[SEC]`
View Scale -> Measure	`:DISPlay[:WINDow]:TRACe:X[:SCALe]:SMSCale`

LEVEL

Function	Control Command
Reference Level	
Log Scale	`:DISPlay:[:WINDow]:Y1[:SCAle]:RLEVel<wsp><NRf>[DBM]`
Linear Scale	`:DISPlay[:WINDow]:Y1[:SCALe]:RLEVel<wsp><NRf>[NW|UM|MW]`
Log Scale**.*dB/D	`:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:PDIVision<wsp><NRf> [DB]`
Linear Scale	`:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing<wsp>LINear|1`
Linear Base Level**.*mW	`:DISPlay[:WINDow]:Y1[:SCALe]:BLEVel<wsp><NRf>[MW]`
Peak Level -> Ref Level	`:CALCulate:MARKer:MAXimum:SRLevel`
Auto Ref Level ON/OFF	`:CALCulate:MARKer:MAXimum:SRLevel:AUTO`
Level Unit dBm / dBm/nm	`:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:UNIT<wsp>DBM|DBM/NM`
Y Scale Setting	
Y Scale Division 8/10/12	`:DISPlay[:WINDow]:TRACe:Y[:SCALe]:DNUMber<wsp>8| 10|12`
Ref Level Position **DIV	`:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RPOSition<wsp><integer>[DIV]`
Sub Log**.*dB/D	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:PDIVision<wsp><NRf>[DB]`
Sub Linear*.***/D	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:PDIVision<wsp><NRf>`
Sub Scale**.*dB/km	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:PDIVision<wsp><NRf>[DB/KM]`
Sub Scale**.*%/D	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:PDIVision<wsp><NRf>[%]`
Offset Level or	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:OLEVel<wsp>`
Scale Minimum **.*dB	`<NRf>[DB]`
Length**.***km	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:LENGth<wsp><NRf>[KM]`
Auto Sub Scale ON/OFF	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:AUTO<wsp>OFF|ON|0|1`
Sub Ref Level Position **DIV	`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:RPOSition<wsp><integer>[DIV]`

SETUP

Function	Control Command
Resolution *.***nm	:SENSe:BANDwidth|:BWIDth[:RESolution]<wsp><NRf>[M|Hz]
Sensitivity/Chop Mode @@@@@@	
NORM/HOLD	:SENSe:SENSe<wsp>NHLD|0
NORM/AUTO	:SENSe:SENSe<wsp>NAUT|1
NORM	:SENSe:SENSe<wsp>NORMal|6
MID	:SENSe:SENSe<wsp>MID|2
HIGH1	:SENSe:SENSe<wsp>HIGH1|3
HIGH2	:SENSe:SENSe<wsp>HIGH2|4
HIGH3	:SENSe:SENSe<wsp>HIGH3|5
Chop Mode @@@@@	
OFF	:SENSe:CHOPper<wsp>OFF|0
SWITCH	:SENSe:CHOPper<wsp>SWITch|2
Average Times ***	:SENSe:AVERage:COUNt<wsp><integer>
Sampling Points AUTO	:SENSe:SWEep:POINts:AUTO<wsp>OFF|ON|0|1
Sampling Points *****	:SENSe:SWEep:POINts<wsp><integer>
Sampling Interval *.****nm	:SENSe:SWEep:STEP<wsp><NRf> [M]
Medium AIR/VACUUM	:SENSe:CORRection:RVELocity:MEDium <wsp>AIR|VACuum|0|1
Sweep Speed 1x/2x	:SENSe:SWEep:SPEed<wsp>1x|2x|0|1
Horizontal Scale nm/THz	:UNIT:X<wsp>WAVelength|FREQuency|0|1
Pulse Light Measure	
Peak Hold **msec	:TRIGger[:SEQuence]:STATe<wsp> OFF|ON|PHOLd|0|1|2
Ext Trigger Mode	:TRIGger[:SEQuence]:STATe<wsp> OFF|ON|PHOLd|0|1|2
Gate Mode ***.*msec	:TRIGger[:SEQuence]:GATE:TIMe
Gate Logic	:TRIGger[:SEQuence]:GATE:LOGic
Trigger Setting	
Edge RISE/FALL	:TRIGger[:SEQuence]:SLOPe<wsp>RISE|FALL|0|1
Delay ****.*μs	:TRIGger[:SEQuence]:DELay<wsp><NRf> [S]
Smoothing ON/OFF	:SENSe:SETTing:SMOothing<wsp> OFF|ON|0|1
Resolution Correction	:SENSe:SETTing:CORRection<wsp>OFF|ON|0|1
Fiber Core Size	:SENSe:SETTing:FIBer<wsp>SMALI|LARGe|0|1

ZOOM

Function	Control Command
Zoom Center Wavelength****.***nm	:DISPlay[:WINDow]:TRACe:X[:SCALe]:CENTer<wsp><NRf>[M]
Zoom Center Frequency***.****THz	:DISPlay[:WINDow]:TRACe:X[:SCALe]:CENTer<wsp><NRf>[HZ]
Zoom Span Wavelength****.*nm	:DISPlay[:WINDow]:TRACe:X[:SCALe]:SPAN<wsp><NRf>[M]
Zoom Span Frequency***.**THz	:DISPlay[:WINDow]:TRACe:X[:SCALe]:SPAN<wsp><NRf>[HZ]
Zoom Start Wavelength****.***nm	:DISPlay[:WINDow]:TRACe:X[:SCALe]:STARt<wsp><NRf>[M]
Zoom Start Frequency	:DISPlay[:WINDow]:TRACe:X[:SCALe]:
.*THz	STARt<wsp><NRf>[HZ]
Zoom Stop Wavelength ****.***nm	:DISPlay[:WINDow]:TRACe:X[:SCALe]:STOP<wsp><NRf>[M]
Zoom Stop Frequency***.****THz	:DISPlay[:WINDow]:TRACe:X[:SCALe]:STOP<wsp><NRf>[HZ]
Peak -> Zoom Ctr	:CALCulate:MARKer:MAXimum:SZCEnter
Overview Display OFF/L/R	:DISPlay[:WINDow]:OVIew:POSition<wsp>OFF|LEFT|RIGHt|0|1|2
Overview Size	:DISPlay[:WINDow]:OVIew:SIZE<wsp>
LARGE/SMALL	LARGe|SMALl|0|1
Initialize	:DISPlay[:WINDow]:TRACe:X[:SCALe]:INITialize

DISPLAY

Function	Control Command
Normal Display	:DISPlay[:WINDow]:SPLit<wsp>OFF|0
Split Display	:DISPlay[:WINDow]:SPLit<wsp>ON|1
Split Display	
Trace A UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRA,UP|LOW|0|1
Trace B UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRB,UP|LOW|0|1
Trace C UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRC,UP|LOW|0|1
Trace D UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRD,UP|LOW|0|1
Trace E UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRE,UP|LOW|0|1
Trace F UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRF,UP|LOW|0|1
Trace G UP/LOW	:DISPlay[:WINDow]:SPLit:POSition<wsp>TRG,UP|LOW|0|1
Hold	
Upper Hold ON/OFF	:DISPlay[:WINDow]:SPLit:HOLD:UPPer<wsp>OFF|ON|0|1
Lower Hold ON/OFF	:DISPlay[:WINDow]:SPLit:HOLD:LOWer<wsp>OFF|ON|0|1
Label	:DISPlay[:WINDow]:TEXT:DATA<wsp><string>
Noise Mask ***dB	:DISPlay[:WINDow]:TRACe:Y:NMASk<wsp><NRf>[DB]
Mask Line VERT / HRZN	:DISPlay[:WINDow]:TRACe:Y:NMASk:TYPE<wsp>VERTical| HORIzontal|0|1
Trace Clear	
All Trace	:DISPlay[:WINDow]:TEXT:CLEar
Display Off	:DISPlay[:WINDow]<wsp>OFF|ON|0|1

TRACE

Function	Control Command
Active Trace	
A	`:TRACe:ACTive<wsp>TRA`
B	`:TRACe:ACTive<wsp>TRB`
C	`:TRACe:ACTive<wsp>TRC`
D	`:TRACe:ACTive<wsp>TRD`
E	`:TRACe:ACTive<wsp>TRE`
F	`:TRACe:ACTive<wsp>TRF`
G	`:TRACe:ACTive<wsp>TRG`
View @ DISP/BLANK	`:TRACe:STATe:<trace name><wsp>ON|OFF|1|0`
Write @	`:TRACe:ATTRibute:<trace name><wsp>WRITe|0`
Fix @	`:TRACe:ATTRibute:<trace name><wsp>FIX|1`
Hold @	
Max Hold	`:TRACe:ATTRibute:<trace name><wsp>MAX|2`
Min Hold	`:TRACe:ATTRibute:<trace name><wsp>MIN|3`

Roll Average @ ***	`:TRACe:ATTRibute:RAVG:<trace name><wsp>`
	`<integer>`

Calculate C@@@@	
Log Math@@@@	
C = A-B(LOG)	`:CALCulate:MATH:TRC<wsp>A-B(LOG)`
C = B-A(LOG)	`:CALCulate:MATH:TRC<wsp>B-A(LOG)`
C = A+B(LOG)	`:CALCulate:MATH:TRC<wsp>A+B(LOG)`
Linear Math@@@@	
C = A+B(LIN)	`:CALCulate:MATH:TRC<wsp>A+B(LIN)`
C = A-B(LIN)	`:CALCulate:MATH:TRC<wsp>A-B(LIN)`
C = B-A(LIN)	`:CALCulate:MATH:TRC<wsp>B-A(LIN)`

C = 1-k(A/B) k: *.****	`:CALCulate:MATH:TRC:K<wsp><NRf>;`
	`:CALCulate:MATH:TRC<wsp>1-K(A/B)`

C = 1-k(B/A) k: *.****	`:CALCulate:MATH:TRC:K<wsp><NRf>;`
	`:CALCulate:MATH:TRC<wsp>1-K(B/A)`

Calculate F@@@@	
Log Math@@@@	
F = C-D(LOG)	`:CALCulate:MATH:TRF<wsp>C-D(LOG)`
F = D-C(LOG)	`:CALCulate:MATH:TRF<wsp>D-C(LOG)`
F = C+D(LOG)	`:CALCulate:MATH:TRF<wsp>C+D(LOG)`
F = D-E(LOG)	`:CALCulate:MATH:TRF<wsp>D-E(LOG)`
F = E-D(LOG)	`:CALCulate:MATH:TRF<wsp>E-D(LOG)`
F = D+E(LOG)	`:CALCulate:MATH:TRF<wsp>D+E(LOG)`
Linear Math@@@@	
F = C+D(LIN)	`:CALCulate:MATH:TRF<wsp>C+D(LIN)`
F = C-D(LIN)	`:CALCulate:MATH:TRF<wsp>C-D(LIN)`
F = D-C(LIN)	`:CALCulate:MATH:TRF<wsp>D-C(LIN)`
F = D+E(LIN)	`:CALCulate:MATH:TRF<wsp>D+E(LIN)`
F = D-E(LIN)	`:CALCulate:MATH:TRF<wsp>D-E(LIN)`
F = E-D(LIN)	`:CALCulate:MATH:TRF<wsp>E-D(LIN)`
Power/NBW@@@@@@@@@@@	
F=Pwr/NBW A	`:CALCulate:MATH:TRF<wsp>PWRNBWA`
F=Pwr/NBW B	`:CALCulate:MATH:TRF<wsp>PWRNBWB`
F=Pwr/NBW C	`:CALCulate:MATH:TRF<wsp>PWRNBWC`
F=Pwr/NBW D	`:CALCulate:MATH:TRF<wsp>PWRNBWD`
F=Pwr/NBW E	`:CALCulate:MATH:TRF<wsp>PWRNBWE`
Bandwidth	`:CALCulate:MATH:TRF:PNBW:BWIDth|BAND`

Function	Control Command
Calculate G@@@@	
Log Math@@@@	
G = C-F(LOG)	:CALCulate:MATH:TRG<wsp>C-F(LOG)
G = F-C(LOG)	:CALCulate:MATH:TRG<wsp>F-C(LOG)
G = C+F(LOG)	:CALCulate:MATH:TRG<wsp>C+F(LOG)
G = E-F(LOG)	:CALCulate:MATH:TRG<wsp>E-F(LOG)
G = F-E(LOG)	:CALCulate:MATH:TRG<wsp>F-E(LOG)
G = E+F(LOG)	:CALCulate:MATH:TRG<wsp>E+F(LOG)
Linear Math@@@@	
G = C+F(LIN)	:CALCulate:MATH:TRG<wsp>C+F(LIN)
G = C-F(LIN)	:CALCulate:MATH:TRG<wsp>C-F(LIN)
G = F-C(LIN)	:CALCulate:MATH:TRG<wsp>F-C(LIN)
G = E+F(LIN)	:CALCulate:MATH:TRG<wsp>E+F(LIN)
G = E-F(LIN)	:CALCulate:MATH:TRG<wsp>E-F(LIN)
G = F-E(LIN)	:CALCulate:MATH:TRG<wsp>F-E(LIN)
Normalize@@@@	
G = NORM A	:CALCulate:MATH:TRG<wsp>NORMA
G = NORM B	:CALCulate:MATH:TRG<wsp>NORMB
G = NORM C	:CALCulate:MATH:TRG<wsp>NORMC
Curve Fit@@@@	
G=CRV FIT A	:CALCulate:MATH:TRG<wsp>CVFTA
G=CRV FIT B	:CALCulate:MATH:TRG<wsp>CVFTB
G=CRV FIT C	:CALCulate:MATH:TRG<wsp>CVFTC
G = MKR FIT	:CALCulate:MATH:TRG<wsp>MKRFT
Threshold **dB	:CALCulate:MATH:TRG:CVFT:THResh<wsp><NRf>[DB]
Operation Area	:CALCulate:MATH:TRG:CVFT:OPARea<wsp>ALL|INL1-L2|OUTL1-L2|0|1|2
Fitting Algorhythm	:CALCulate:MATH:TRG:CVFT:FALGo<wsp>GAUSS|LORENz|3RD|4TH|5TH|0|1|2|3|4
Peak Curve Fit @@@@	
G = PKCVFIT A	:CALCulate:MATH:TRG<wsp>PKCVFTA
G = PKCVFIT B	:CALCulate:MATH:TRG<wsp>PKCVFTB
G = PKCVFIT C	:CALCulate:MATH:TRG<wsp>PKCVFTC
Threshold **dB	:CALCulate:MATH:TRG:PCVFt:THResh<wsp><NRf>[DB]
Operation Area	:CALCulate:MATH:TRG:CVFT:OPARea<wsp>ALL|INL1-L2|OUTL1-L2|0|1|2
Fitting Algorhythm	:CALCulate:MATH:TRG:CVFT:FALGo<wsp>GAUSS |LORENz|3RD|4TH|5TH|0|1|2|3|4
Trace List	–
Trace Copy	:TRACe:COPY<wsp><source trace name>, <destination trace name>
Trace Clear	:TRACe:DELete<wsp><trace name>

MARKER

Function				Control Command
Marker Active ON/OFF , Set Marker				:CALCulate:MARKer[:STATe]<wsp><marker>,|ON|1 :CALCulate:MARKer:X<wsp><marker>,<NRf>[M|HZ]
Clear Marker				:CALCulate:MARKer[:STATe]<wsp><marker>,OFF|0
Marker -> Center				:CALCulate:MARKer:SCENter
Marker -> Zoom Ctr				:CALCulate:MARKer:SZCenter
Marker -> Ref Level				:CALCulate:MARKer:SRLevel
Advanced Marker				
	Marker 1 Select @@@@@@@			:CALCulate:AMARker[1]:X<wsp><NRf>[M|Hz]
		Marker Trace		
			A	:CALCulate:AMARker[1]:TRACe<wsp>TRA
			B	:CALCulate:AMARker[1]:TRACe<wsp>TRB
			C	:CALCulate:AMARker[1]:TRACe<wsp>TRC
			D	:CALCulate:AMARker[1]:TRACe<wsp>TRD
			E	:CALCulate:AMARker[1]:TRACe<wsp>TRE
			F	:CALCulate:AMARker[1]:TRACe<wsp>TRF
			G	:CALCulate:AMARker[1]:TRACe<wsp>TRG
		Off		:CALCulate:AMARker[1][:STATe]<wsp>OFF|0
		Normal		:CALCulate:AMARker[1]:FUNCtion:PRESet
		Power Density		:CALCulate:AMARker[1]:FUNCtion:PDENsity|:NOISe [:STATe]<wsp>ON|1
		Integral Power		:CALCulate:AMARker[1]:FUNCtion:INTegral[:STATe] <wsp>ON|1
		Integral Range ***.*GHz		:CALCulate:AMARker[1]:FUNCtion:INTegral:IRANge <wsp><integer>[Hz]
	Marker 2 Select @@@@@@@			:CALCulate:AMARker2:X<wsp><NRf>[M|Hz]
		Marker Trace		
			A	:CALCulate:AMARker2:TRACe<wsp>TRA
			B	:CALCulate:AMARker2:TRACe<wsp>TRB
			C	:CALCulate:AMARker2:TRACe<wsp>TRC
			D	:CALCulate:AMARker2:TRACe<wsp>TRD
			E	:CALCulate:AMARker2:TRACe<wsp>TRE
			F	:CALCulate:AMARker2:TRACe<wsp>TRF
			G	:CALCulate:AMARker2:TRACe<wsp>TRG
		Off		:CALCulate:AMARker2[:STATe]<wsp>OFF|0
		Normal		:CALCulate:AMARker2:FUNCtion:PRESet
		Power Density		:CALCulate:AMARker2:FUNCtion:PDENsity|:NOISe [:STATe]<wsp>ON|1
		Integral Power		:CALCulate:AMARker2:FUNCtion:INTegral[:STATe] <wsp>ON|1
		Integral Range ***.*GHz		:CALCulate:AMARker2:FUNCtion:INTegral:IRANge<wsp> <integer>[Hz]
	Marker 3 Select @@@@@@@			:CALCulate:AMARker3:X<wsp><NRf>[M|Hz]
		Marker Trace		
			A	:CALCulate:AMARker3:TRACe<wsp>TRA
			B	:CALCulate:AMARker3:TRACe<wsp>TRB
			C	:CALCulate:AMARker3:TRACe<wsp>TRC
			D	:CALCulate:AMARker3:TRACe<wsp>TRD
			E	:CALCulate:AMARker3:TRACe<wsp>TRE
			F	:CALCulate:AMARker3:TRACe<wsp>TRF
			G	:CALCulate:AMARker3:TRACe<wsp>TRG
		Off		:CALCulate:AMARker3[:STATe]<wsp>OFF|0
		Normal		:CALCulate:AMARker3:FUNCtion:PRESet

Function	Control Command
Power Density	:CALCulate:AMARker3:FUNCtion:PDENsity|:NOISe [:STATe]<wsp>ON|1
Integral Power	:CALCulate:AMARker3:FUNCtion:INTegral[:STATe] <wsp>ON|1
Integral Range ***.*GHz	:CALCulate:AMARker3:FUNCtion:INTegral:IRANge<wsp> <integer>[Hz]
Marker 4 Select @@@@@@@	:CALCulate:AMARker4:X<wsp><NRf>[M|Hz]
Marker Trace	
A	:CALCulate:AMARker4:TRACe<wsp>TRA
B	:CALCulate:AMARker4:TRACe<wsp>TRB
C	:CALCulate:AMARker4:TRACe<wsp>TRC
D	:CALCulate:AMARker4:TRACe<wsp>TRD
E	:CALCulate:AMARker4:TRACe<wsp>TRE
F	:CALCulate:AMARker4:TRACe<wsp>TRF
G	:CALCulate:AMARker4:TRACe<wsp>TRG
Off	:CALCulate:AMARker4[:STATe]<wsp>OFF|0
Normal	:CALCulate:AMARker4:FUNCtion:PRESet
Power Density	:CALCulate:AMARker4:FUNCtion:PDENsity|:NOISe [:STATe]<wsp>ON|1
Integral Power	:CALCulate:AMARker4:FUNCtion:INTegral[:STATe] <wsp>ON|1
Integral Range ***.*GHz	:CALCulate:AMARker4:FUNCtion:INTegral:IRANge<wsp> <integer>[Hz]
Search	
Peak Search	:CALCulate:AMARker[1|2|3|4]:MAXimum
Bottom Search	:CALCulate:AMARker[1|2|3|4]:MINimum
Next Level Search	:CALCulate:AMARker[1|2|3|4]:MAXimum:NEXT :CALCulate:AMARker[1|2|3|4]:MINimum:NEXT
Next Search Right	:CALCulate:AMARker[1|2|3|4]:MAXimum:RIGHt :CALCulate:AMARker[1|2|3|4]:MINimum:RIGHt
Next Search Left	:CALCulate:AMARker[1|2|3|4]:MAXimum:LEFT :CALCulate:AMARker[1|2|3|4]:MINimum:LEFT
Bandwidth **.*nm	:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|: NOISe:BWIDth|:BANDwidth<wsp><NRf>[M]
All Clear	:CALCulate:AMARker[1|2|3|4]:AOFF
All Marker Clear	:CALCulate:MARKer:AOFF
Line Marker 1 ON/OFF	:CALCulate:LMARker:X<wsp>1,<NRf>[M]
Line Marker 2 ON/OFF	:CALCulate:LMARker:X<wsp>2,<NRf>[M]
Line Marker 3 ON/OFF	:CALCulate:LMARker:Y<wsp>3,<NRf>[DBM]
Line Marker 4 ON/OFF	:CALCulate:LMARker:Y<wsp>4,<NRf>[DBM]
Marker L1-L2 -> Span	:CALCulate:LMARker:SSPan
Marker L1-L2 -> Zoom Span	:CALCulate:LMARker:SZSPan
Line Marker All Clear	:CALCulate:LMARker:AOFF
Marker Display	
Offset	:CALCulate:MARKer:FUNCtion:FORMat<wsp>OFFSet|0
Spacing	:CALCulate:MARKer:FUNCtion:FORMat<wsp>SPACing|1
Marker Auto Update ON/OFF	:CALCulate:MARKer:FUNCtion:UPDateQ<wsp> OFF|ON|0|1
Marker Unit nm THz	:CALCulate:MARKer:UNIT<wsp>WAVelength|FREQuency|0|1
Search/Ana Marker L1-L2 ON/OFF	:CALCulate:LMARker:SRANge<wsp>OFF|ON|0|1
Search/Ana Zoom Area ON/OFF	:DISPlay[:WINDow]:TRACe:X[:SCALe]:QSRANge<wsp> OFF|ON|0|1
Sweep Marker L1-L2 ON/OFF	:SENSe:WAVelength:SRANge<wsp>OFF|ON|0|1

PEAK SEARCH

Function	Control Command
Peak Search	:CALCulate:MARKer:MAXimum
Bottom Search	:CALCulate:MARKer:MINimum
Next Level Search	:CALCulate:MARKer:MAXimum:NEXT or :CALCulate:MARKer:MINimum:NEXT
Next Search Right	:CALCulate:MARKer:MAXimum:RIGHt or :CALCulate:MARKer:MINimum:RIGHt
Next Search Left	:CALCulate:MARKer:MAXimum:LEFT or :CALCulate:MARKer:MINimum:LEFT
Set Marker	:CALCulate:MARKer[:STATe]<wsp><marker>,|ON|1
Clear Marker	:CALCulate:MARKer[:STATe]<wsp><marker>,OFF|0
All Marker Clear	:CALCulate:MARKer:AOFF
Auto Search ON/OFF	:CALCulate:MARKer:AUTO<wsp>OFF|ON|0|1
Mode Diff **.**dB	:CALCulate:PARameter:COMMon:MDIFf<wsp><NRf>[DB]
Search/Ana Marker L1-L2 ON/OFF	:CALCulate:LMARker:SRANge<wsp>OFF|ON|0|1
Search/Ana Zoom Area ON/OFF	:DISPlay[:WINDow]:TRACe:X[:SCALe]:SRANge<wsp>OFF|ON|0|1
Search Mode SINGL/MULTI	:CALCulate:MARKer:MSEarch<wsp>OFF|ON|0|1
Multi Search Setting	
Threshold **.**dB	:CALCulate:MARKer:MSEarch:THResh<wsp><NRf>[DB]
Sort By WL/LVL	:CALCulate:MARKer:MSEarch:SORT<wsp>WAVelength|LEVel|0|1

ANALYSIS

Function	Control Command
Spec Width@@@@	
THRESH	`:CALCulate:CATegory<wsp>SWTHresh|0`
ENVELOPE	`:CALCulate:CATegory<wsp>SWENvelope|1`
RMS	`:CALCulate:CATegory<wsp>SWRMs|2`
PEAK RMS	`:CALCulate:CATegory<wsp>SWPKrms|3`
NOTCH	`:CALCulate:CATegory<wsp>NOTCh|4`
Analysis1@@@@	
DFB-LD	`:CALCulate:CATegory<wsp>DFBLd|5`
FP-LD	`:CALCulate:CATegory<wsp>FPLD|6`
LED	`:CALCulate:CATegory<wsp>LED|7`
SMSR	`:CALCulate:CATegory<wsp>SMSR|8`
POWER	`:CALCulate:CATegory<wsp>POWer|9`
PMD	`:CALCulate:CATegory<wsp>PMD|10`
Analysis2@@@@@	
WDM	`:CALCulate:CATegory<wsp>WDM|11`
EDFA-NF	`:CALCulate:CATegory<wsp>NF|12`
FILTER-PK	`:CALCulate:CATegory<wsp>FILPk|13`
FILTER-BTM	`:CALCulate:CATegory<wsp>FILBtm|14`
WDM FIL-PK	`:CALCulate:CATegory<wsp>WFPeak|15`
WDM FIL-BTM	`:CALCulate:CATegory<wsp>WFBtm|16`
COLOR	`:CALCulate:CATegory<wsp>COLor|17`
Analysis Execute (@@@@)	`:CALCulate[:IMMediate]`
Spec Width Thresh**.*dB	`:CALCulate:PARameter[:CATegory]:SWTHresh:TH<wsp><NRf>[DB]`
Switch Display	
Trace & Table	`:CALCulate:DISPlay<wsp>0`
Table	`:CALCulate:DISPlay<wsp>1`
Trace	`:CALCulate:DISPlay<wsp>2`
Graph & Table	`:CALCulate:DISPlay<wsp>3`
Graph	`:CALCulate:DISPlay<wsp>4`
Line Marker Y1/Y2	`:CALCulate:DISPlay:GRAPh:LMARker:Y<wsp>1|2,<NRf>[DB]`
Auto Analysis ON/OFF	`:CALCulate[:IMMediate]:AUTO<wsp>OFF|ON|0|1`
Result Save	`MMEMory:STORe:ARESult<wsp><"filename">[,INTernal|EXTernal]`
Search/Ana Marker L1-L2 ON/OFF	`:CALCulate:LMARker:SRANge<wsp>OFF|ON|0|1ON/OFF`
Search/Ana Zoom Area ON/OFF	`:DISPlay[:WINDow]:TRACe:X[:SCALe]:SRANge<wsp>OFF|ON|0|1`

MEMORY

Function	Control Command
Save	
A Trace -> Memory	:MEMory:STORe<wsp><integer>,TRA
B Trace -> Memory	:MEMory:STORe<wsp><integer>,TRB
C Trace -> Memory	:MEMory:STORe<wsp><integer>,TRC
D Trace -> Memory	:MEMory:STORe<wsp><integer>,TRD
E Trace -> Memory	:MEMory:STORe<wsp><integer>,TRE
F Trace -> Memory	:MEMory:STORe<wsp><integer>,TRF
G Trace -> Memory	:MEMory:STORe<wsp><integer>,TRG
Recall	
Memory -> A Trace	:MEMory:LOAD<wsp><integer>,TRA
Memory -> B Trace	:MEMory:LOAD<wsp><integer>,TRB
Memory -> C Trace	:MEMory:LOAD<wsp><integer>,TRC
Memory -> D Trace	:MEMory:LOAD<wsp><integer>,TRD
Memory -> E Trace	:MEMory:LOAD<wsp><integer>,TRE
Memory -> F Trace	:MEMory:LOAD<wsp><integer>,TRF
Memory -> G Trace	:MEMory:LOAD<wsp><integer>,TRG
Clear	:MEMory:CLEar<wsp><integer>

FILE

Function	Control Command
Write	
Memory INT/EXT	:MMEMory:CDRive<wsp>INTernal|EXTernal
File Name	:MMEMory:CDIRectory<wsp><directory name>
(Trace)	:MMEMory:STORe:TRACe<wsp><trace name>, BIN|CSV,<"file name">[,INTernal|EXTernal]
(All Trace)	:MMEMory:STORe:ATRace<wsp> <”file name”> [,INTernal|EXTernal]
(Memory)	:MMEMory:STORe:MEMory<wsp><integer>,BIN|CSV, <"file name">[,INTernal|EXTernal]
(Graphics)	:MMEMory:STORe:GRAPhics<wsp>B&W|COLor,BMP|TIFF, <"file name">[,INTernal|EXTernal]
(Setting)	:MMEMory:STORe:SETTing<wsp><"file name"> [,INTernal|EXTernal]
(Data)	:MMEMory:STORe:DATA<wsp><"file name"> [,INTernal|EXTernal]
Output Item Setting	
Date & Time ON/OFF	:MMEMory:STORe:DATA:TEM<wsp>DATE,OFF|ON|0|1
Label ON/OFF	:MMEMory:STORe:DATA:ITEM<wsp>LABel,OFF|ON|0|1
Data Area ON/OFF	:MMEMory:STORe:DATA:TEM<wsp>DATA,OFF|ON|0|1
Condition ON/OFF	:MMEMory:STORe:DATA:ITEM<wsp>CONDition,OFF| ON|0|1
Trace Data ON/OFF	:MMEMory:STORe:DATA:ITEM<wsp>TRACe,OFF|ON|0|1
File Type CSV/DT9	:MMEmory:STORe:DATA:TYPE<wsp>CSV|DT|0|1
Write Mode ADD/OVER	:MMEMory:STORe:DATA:MODE<wsp> ADD|OVER|0|1
(Program)	:MMEMory:STORe:PROGram<wsp><integer>, <"file name">[,INTernal|EXTernal]

Function	Control Command
(Template)	:MMEMory:STORe:TEMPlate<wsp><template>,<”file name”>[,INTernal|EXTernal]
(Logging)	:MMEMory:STORe:DLOGging<wsp><”file name”>[,INTernal|EXTernal]
<CSV Data Save>	:MMEMory:STORe:DLOGging:CSAVe<wsp>OFF|ON|0|1
<Trace Data Save>	:MMEMory:STORe:DLOGging:TSAVe<wsp>OFF|ON|0|1
Read	
Memory INT/EXT	:MMEMory:CDRive<wsp>INTernal|EXTernal
(Trace)	:MMEMory:LOAD:TRACe<wsp><trace name>,<"file name">[,INTernal|EXTernal]
(All Trace)	:MMEMory:LOAD:ATRace<wsp><”file name”>[,INTernal|EXTernal]
(Memory)	:MMEMory:LOAD:MEMory<wsp><integer>,<"file name">[,INTernal|EXTernal]
(Setting)	:MMEMory:LOAD:SETTing<wsp><"file name">[,INTernal|EXTernal]
(Data)	:MMEMory:LOAD:DATA<wsp><"file name">[,INTernal|EXTernal]
(Program)	:MMEMory:LOAD:PROGram<wsp><integer>,<"file name">[,INTernal|EXTernal]
(Template)	:MMEMory:LOAD:PROGram<wsp><template><"file name">[,INTernal|EXTernal]
(Logging)	:MMEMory:LOAD:DLOGging<wsp><"file name">[,INTernal|EXTernal]
Auto File Name	:MMEMory:ANAMe<wsp>NUMBer|DATE
Remove USB Storage	:MMEMORY:REMove
File Operation	
Memory INT/EXT	:MMEMory:CDRive<wsp>INTernal|EXTernal
Delete	:MMEMory:DELete<wsp><"file name">[,INTernal|EXTernal]
Copy	:MMEMory:COPY<wsp><"source file name">,[INTernal|EXTernal],<"destination file name>[,INTernal|EXTernal]
Rename	:MMEMory:REName<wsp><"new file name">,<"old file name">[,INTernal|EXTernal]
Make Directory	:MMEMory:MDIRectory<wsp><"directory name">[,INTernal|EXTernal]

PROGRAM

Function	Control Command
Program Execute	:PROGram:EXECute<wsp><integer>

SYSTEM

Function	Control Command
Optical Alignment	`:CALibration:ALIGn[:IMMediate]`
Wavelength Calibration	
Built-in Source	`:CALibration:WAVelength:INTernal [:IMMediate]`

External Laser ****.***nm	`:CALibration:WAVelength:EXTernal:SOURce<wsp>LASer|0;`
	`:CALibration:WAVelength:EXTernal:WAVelength<wsp><NRf>[M]`

External Gas Cell ****.***nm	`:CALibration:WAVelength:EXTernal:SOURce<wsp>GASCell|1;`
	`:CALibration:WAVelength:EXTernal:WAVelength<wsp><NRf>[M]`

Emission Line ****.***nm	`:CALibration:WAVelength:EXTernal:SOURce<wsp>EMISsion|2;`
	`:CALibration:WAVelength:EXTernal:WAVelength<wsp><NRf>[M]`

Wavelength Shift **.***nm	`:SENSe:CORRection:WAVelength:SHIFt<wsp><NRf>[M]`
Level Shift ***.***dB	`:SENSe:CORRection:LEVel:SHIFt<wsp><NRf>[DB]`
Grid Editor	
200GHz Spacing	`:SYSTem:GRID<wsp>200GHZ|4`
100GHz Spacing	`:SYSTem:GRID<wsp>100GHZ|3`
50GHz Spacing	`:SYSTem:GRID<wsp>50GHZ|2`
25GHz Spacing	`:SYSTem:GRID<wsp>25GHZ|1`
12.5GHz Spacing	`:SYSTem:GRID<wsp>12.5GHZ|0`
Custom	`:SYSTem:GRID<wsp>CUSTom|5`
Start Wavelength ****.****nm	`:SYSTem:GRID:CUSTom:STARt<wsp><NRf>[M|HZ]`
Stop Wavelength ****.****nm	`:SYSTem:GRID:CUSTom:STOP<wsp><NRf>[M|HZ]`
Spacing ***.*GHz	`:SYSTem:GRID:CUSTom:SPACing<wsp><NRf>[GHZ]`
Value Edit	–
Insert	`:SYSTem:GRID:CUSTom:INSert<wsp><NRf>[M|HZ]`
Delete	`:SYSTem:GRID:CUSTom:DELete<wsp><integer>`
Reference Wavelength ****.****nm	`:SYSTem:GRID:REFerence<wsp><NRf>[HZ]`
User Key Define	–

Function	Control Command
Command Format	`:SYSTem:COMMunicate:CFORmat<wsp>AQ6317|AQ6374|0|1`
Monitor Port ON/OFF	`:SYSTem:COMMunicate:RMONitor<wsp>OFF|ON|0|1`
Trig Input Mode	`:TRIGger[:SEQuence]:INPut<wsp>ETRigger|STRigger|SENable|0|1|2`
Trig Output Mode	`:TRIGger[:SEQuence]:OUTPut<wsp>OFF|SSTatus|0|1`
Auto Offset Setting	
Auto Offset ON/OFF	`:CALibration:ZERO[:AUTO]<wsp>OFF|ON|0|1`
Interval ***min	`:CALibration:ZERO[:AUTO]:INTerval<wsp><integer>`
Uncal Warning ON/OFF	`:SYSTem:DISPlay:UNCal<wsp>OFF|ON|0|1`
Buzzer	
Click ON/OFF	`:SYSTem:BUZZer:CLICk<wsp>OFF|ON|0|1`
Warning ON/OFF	`:SYSTem:BUZZer:WARNing<wsp>OFF|ON|0|1`
Level Display Digit	
1DIGIT	`:UNIT:POWer:DIGit<wsp>1`
2DIGIT	`:UNIT:POWer:DIGit<wsp>2`
3DIGIT	`:UNIT:POWer:DIGit<wsp>3`
Window Transparent ON/OFF	`:SYSTem:DISPlay:TRANsparent<wsp>OFF|ON|0|1`

Set Clock	`:SYSTem:DATE<wsp><year>,<month>,<day>`
	`:SYSTem:TIME<wsp><hour>,<minutes>, <seconds>`

Color Mode	
COLOR	`:DISPlay:COLor<wsp>1`
B&W	`:DISPlay:COLor<wsp>0`
Remove USB Storage	`:MMEMory:REMove`
Operation Lock	`:SYSTem:OLOCK`
System Information	
System Information	`:SYSTem:INFormation?<wsp>0|1`
Parameter Initialize	
Parameter Clear	`:SYSTem:PRESet`
Version	–
Shut Down	`:SYSTem:OPERation:SHUTdown`

ADVANCE

Function	Control Command
Template	
Go/No Go ON/OFF	:TRACe:TEMPlate:GONogo<wsp>OFF|ON|0|1
Template Display	
Upper Line Display ON/OFF	:TRACe:TEMPlate:DISPlay<wsp>UPPer,OFF|ON|0|1
Lower Line Display ON/OFF	:TRACe:TEMPlate:DISPlay<wsp>LOWer,OFF|ON|0|1
Target Line Display ON/OFF	:TRACe:TEMPlate:DISPlay<wsp>TARGet,OFF|ON|0|1
Type	
Upper	:TRACe:TEMPlate:TTYPe<wsp>UPPer
Lower	:TRACe:TEMPlate:TTYPe<wsp>LOWer
Upper & Lower	:TRACe:TEMPlate:TTYPe<wsp>U&L
Template Edit	
All Delete	:TRACe:TEMPlate:DATA:ADELete<wsp>UPPer|LOWer|TARGet
Mode ABS/REL	:TRACe:TEMPlate:DATA:MODE<wsp>UPPer|LOWer|TARGet, ABSolute|RELative
Extrapol Type	
Type A	:TRACe:TEMPlate:DATA:ETYPe<wsp>UPPer|LOWer|TARGet,A|1
Type B	:TRACe:TEMPlate:DATA:ETYPe<wsp>UPPer|LOWer|TARGet,B|2
None	:TRACe:TEMPlate:DATA:ETYPe<wsp>UPPer|LOWer|TARGet,NONE|0
Template Shift	:TRACe:TEMPlate:LEVel:SHIFt<wsp><NRf>
	:TRACe:TEMPlate:WAVelength:SHIFt<wsp><NRf>
Data Logging	
Start/Stop	:APPLication:DLOGging:STATe<wsp>STOP|STARt|0|1
Setup	
Logging Parameter	
LOGGING ITEM	:APPLication:DLOGging:LPARameter:ITEM<wsp>0|1|2|3
LOGGING MODE	:APPLication:DLOGging:LPARameter:LMODe<wsp>1|2
MINIMUM INTERVAL	:APPLication:DLOGging:LPARameter:INTerval<wsp><integer>[SEC]
TEST DURATION	:APPLication:DLOGging:LPARameter:TDURation<wsp><integer>[sec]
PEAK THRESH TYPE	:APPLication:DLOGging:LPARameter:PDETect:TTYPe<wsp>ABSolute|RELative
THRESH(ABS)	:APPLication:DLOGging:LPARameter:PDETect:ATHResh<NRf>[DBM]
THRESH(REL)	:APPLication:DLOGging:LPARameter:PDETect:RTHResh<NRf>[DB]
CH MATCHING λ THRESH	:APPLication:DLOGging:LPARameter:MTHResh<wsp><NRf>[M]
TRACE LOGGING	:APPLication:DLOGging:LPARameter:TLOGging<wsp>OFF|ON|0|1
DESTINATION MEMORY	:APPLication:DLOGging:LPARameter:MEMory<wsp>INTernal|EXTernal
Logging Data Save	:MMEMory:STORe:DLOGging<wsp><"file name">[,INTernal|EXTernal]
Logging Data Load	:MMEMory:LOAD:DLOGging<wsp><"file name">[,INTernal|EXTernal]

COPY

Function	Control Command
COPY	`:HCOPY[:IMMediate]`

PRESET

Function	Control Command
PRESET	`:SYSTem:PRESet`

6.3 ANALYSIS Setting Parameters

In setting ANALYSIS key setting parameters, the analysis parameters differ with the analysis type. Thus, the PARAMETER SETTING key commands are set independently of the regular key commands. An analysis parameter setting command is shown below.

Spec Width

ANALYSIS Parameters	Control Command
THRESH	
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:SWTHresh:TH<wsp><NRf>[DB]
K **.**	:CALCulate:PARameter[:CATegory]:SWTHresh:K<wsp><NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:SWTHresh:MFIT<wsp>OFF|ON|0|1
ENVELOPE	
THRESH LEVEL1**.**dB	:CALCulate:PARameter[:CATegory]:SWENvelope:TH1<wsp><NRf>[DB]
THRESH LEVEL2 **.**dB	:CALCulate:PARameter[:CATegory]:SWENvelope:TH2<wsp><NRf>[DB]
K **.**	:CALCulate:PARameter[:CATegory]:SWENvelope:K
PEAK RMS	
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:SWPKrms:TH<wsp><NRf>[DB]
K **.**	:CALCulate:PARameter[:CATegory]:SWPKrms:K<wsp><NRf>[DB]
NOTCH	
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:NOTCh:TH<wsp><NRf>[DB]
K **.**	:CALCulate:PARameter[:CATegory]:NOTCh:K<wsp><NRf>[DB]
Type	
PEAK	:CALCulate:PARameter[:CATegory]:NOTCh:TYPE<wsp>PEAK|0
BOTTOM	:CALCulate:PARameter[:CATegory]:NOTCh:TYPE<wsp>BOTTom|1

Analysis 1

ANALYSIS Parameters	Control Command
DFB-LD	
-XdB WIDTH	
ALGO	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,TH,<NRf>[DB]
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,TH2,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,MDIFf,<NRf>[DB]
SWIDth	
ALGO	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,TH,<NRf>[DB]

ANALYSIS Parameters	Control Command
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,TH2,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SWIDth,MDIFf,<NRf>[DB]
SMSR	
SMSR MODE	:CALCulate:PARameter[:CATegory]:DFBLd <wsp>SMSR,SMODe,SMSR1|SMSR2|SMSR3|SMSR4
SMSR MASK ±*.**nm	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SMSR,SMASk,<NRf>[M]
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>SMSR,MDIFf,<NRf>[DB]
RMS	
ALGO	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>RMS,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>RMS,TH,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>RMS,K,<NRf>
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>RMS,MDIFf,<NRf>[DB]
POWER	
SPAN **.**nm	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>POWer,SPAN,<NRf>[M]
OSNR	
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,MDIFf,<NRf>[DB]
NOISE ALGO	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,NALGo,<data>
NOISE AREA **.**nm	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,NARea,<NRf>[M]
MASK AREA *.**nm	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,MARea,<NRf>[M]
FITTING ALGO	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,FALGo,<data>
NOISE BW **.**nm	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,NBW,<NRf>[M]
SIGNAL POWER	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,SPOWer,<data>
INTEGRAL RANGE	:CALCulate:PARameter[:CATegory]:DFBLd<wsp>OSNR,IRANge,<NRf>
FP-LD	
SPECTRUM WIDTH	
ALGO	:CALCulate:PARameter[:CATegory]:FPLD<wsp>SWIDth,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>SWIDth,TH,<NRf>[DB]
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>SWIDth,TH2,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:FPLD<wsp>SWIDth,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:FPLD<wsp>SWIDth,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>SWIDth,MDIFf,<NRf>[DB]

6.3 ANALYSIS Setting Parameters

ANALYSIS Parameters	Control Command
MEAN WAVELENGTH	
ALGO	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MWAVelength,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MWAVelength,TH,<NRf>[DB]
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MWAVelength,TH2,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MWAVelength,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MWAVelength,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MWAVelength,MDIFf,<NRf>[DB]
TOTAL POWER	
OFFSET LEVEL *.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>TPOWer,OFFSet,<NRf>[DB]
MODE NO.	
ALGO	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MNUMber,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MNUMber,TH,<NRf>[DB]
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MNUMber,TH2,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MNUMber,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MNUMber,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FPLD<wsp>MNUMber,MDIFf,<NRf>[DB]
LED	
SPECTRUM WIDTH	
ALGO	:CALCulate:PARameter[:CATegory]:LED<wsp>SWIDth,ALGO,<data>
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>SWIDth,TH,<NRf>[DB]
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>SWIDth,TH2,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:LED<wsp>SWIDth,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:LED<wsp>SWIDth,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>SWIDth,MDIFf,<NRf>[DB]
MEAN WAVELENGTH	
ALGO	
THRESH **.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>MWAVelength,TH,<NRf>[DB]
THRESH2 **.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>MWAVelength,TH2,<NRf>[DB]
K	
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:LED<wsp>MWAVelength,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>MWAVelength,MDIFf,<NRf>[DB]
TOTAL POWER	
OFFSET LEVEL *.**dB	:CALCulate:PARameter[:CATegory]:LED<wsp>TPOWer,OFFSet,<NRf>[DB]
SMSR	
SMSR MODE	:CALCulate:PARameter[:CATegory]:SMSR:MODE<wsp>SMSR1|SMSR2|SMSR3|SMSR4
SMSR MASK ±*.**dB	:CALCulate:PARameter[:CATegory]:SMSR:MASK<wsp><NRf>[M]POWER
POWER	
OFFSET LEVEL *.**dB	:CALCulate:PARameter[:CATegory]:POWer:OFFSet<wsp><NRf>[DB]
PMD	
THRESH LEVEL *.**dB	:CALCulate:PARameter[:CATegory]:PMD:TH<wsp><NRf>[DB]

Analysis 2

ANALYSIS Parameters	Control Command
WDM	
CHANNEL DETECTION SETTING	
THRESH LEVEL	:CALCulate:PARameter[:CATegory]:WDM:TH<wsp><NRf>[DB]
MODE DIFF **.**dB	:CALCulate:PARameter[:CATegory]:WDM:MDIFf<wsp><NRf>[DB]
DISPLAY MASK OFF/ON *.**dB	:CALCulate:PARameter[:CATegory]:WDMASk<wsp><NRf>[DB]
INTERPOLATATION SETTING	
NOISE ALGO	
AUTO-FIX	:CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>AFIX|0
MANUAL-FIX	:CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>MFIX|1
AUTO-CTR	:CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>ACENter|2
MANUAL-CTR	:CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>MCENter|3
PIT	:CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>PIT|4
FITTING AREA	:CALCulate:PARameter[:CATegory]:WDM:NARea<wsp><NRf>[M]
MASK AREA	:CALCulate:PARameter[:CATegory]:WDM:MARea<wsp><NRf>[M]
FITTING ALGO	
LINEAR	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>LINear|0
GAUSS	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>GAUSs|1
LORENZ	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>LORenz|2
3RD POLY	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>3RD|3
4TH POLY	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>4TH|4
5TH POLY	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>5TH|5
NOISE BW *.**nm	:CALCulate:PARameter[:CATegory]:WDM:NBW<wsp><NRf>[M]
DUAL TRACE ON/OFF	:CALCulate:PARameter[:CATegory]:WDM:DUAL<wsp>OFF|ON|0|1
DISPLAY SETTING	
DISPLAY TYPE	
ABSOLUTE	:CALCulate:PARameter[:CATegory]:WDM:DTYPe<wsp>ABSolute|0
RELATIVE	:CALCulate:PARameter[:CATegory]:WDM:DTYPe<wsp>RELatibe|1
DRIFT(MEAS)	:CALCulate:PARameter[:CATegory]:WDM:DTYPe<wsp>MDRift|2
DRIFT(GRID)	:CALCulate:PARameter[:CATegory]:WDM:DTYPe<wsp>GDRift|3
CH RELATION	
OFFSET	:CALCulate:PARameter[:CATegory]:WDM:RELation<wsp>OFFSet|0
SPACING	:CALCulate:PARameter[:CATegory]:WDM:RELation<wsp>SPACing|1
REF CH	:CALCulate:PARameter[:CATegory]:WDM:RCH<wsp><integer>
MAX/MIN RESET	:CALCulate:PARameter[:CATegory]:WDM:MMReset
OUTPUT SLOPE ON/OFF	:CALCulate:PARameter[:CATegory]:WDM:OSLope<wsp>OFF|ON|0|1
POINT DISPLAY ON/OFF	:CALCulate:PARameter[:CATegory]:WDM:PDISplay<wsp>OFF|ON|0|1

<table>
<tr><th>ANALYSIS Parameters</th><th>Control Command</th></tr>
<tr><td>OTHER SETTING</td><td></td></tr>
<tr><td>SIGNAL POWER</td><td>:CALCulate:PARameter[:CATegory]:WDM:SPOWer<wsp> PEAK|INTegral|0|1</td></tr>
<tr><td>INTEGRAL RANGE</td><td>:CALCulate:PARameter[:CATegory]:WDM:IRANge<wsp> <NRf></td></tr>
<tr><td>EDFA NF</td><td></td></tr>
<tr><td>CHANNNEL DETECTION</td><td></td></tr>
<tr><td>THRESH LEVEL **.**dB</td><td>:CALCulate:PARameter[:CATegory]:NF:TH<wsp><NRf> [DB]</td></tr>
<tr><td>MODE DIFF **.**dB</td><td>:CALCulate:PARameter[:CATegory]:NF:MDIFf<wsp> <NRf>[DB]</td></tr>
<tr><td>INTERPOLATION SETTING</td><td></td></tr>
<tr><td>OFFSET(IN) **.**dB</td><td>:CALCulate:PARameter[:CATegory]:NF:IOFFset<wsp> <NRf>[DB]</td></tr>
<tr><td>OFFSET(OUT) **.**dB</td><td>:CALCulate:PARameter[:CATegory]:NF:OOFFset<wsp> <NRf>[DB]</td></tr>
<tr><td>ASE ALGO</td><td></td></tr>
<tr><td>AUTO-FIX</td><td>:CALCulate:PARameter[:CATegory]:NF:AALGo<wsp> AFIX|0</td></tr>
<tr><td>MANUAL-FIX</td><td>:CALCulate:PARameter[:CATegory]:NF:AALGo<wsp> MFIX|1</td></tr>
<tr><td>AUTO-CTR</td><td>:CALCulate:PARameter[:CATegory]:NF:AALGo<wsp> ACENter|2</td></tr>
<tr><td>MANUAL-CTR</td><td>:CALCulate:PARameter[:CATegory]:NF:AALGo<wsp> MCENter|3</td></tr>
<tr><td>FITTING AREA</td><td>:CALCulate:PARameter[:CATegory]:NF:FARea<wsp> <NRf>[M]</td></tr>
<tr><td>MASK AREA</td><td>:CALCulate:PARameter[:CATegory]:NF:MARea<wsp> <NRf>[M]</td></tr>
<tr><td>FITTING ALGO</td><td></td></tr>
<tr><td>LINEAR</td><td>:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp> LINear|0</td></tr>
<tr><td>GAUSS</td><td>:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp> GAUSs|1</td></tr>
<tr><td>LORENZ</td><td>:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp> LORenz|2</td></tr>
<tr><td>3RD POLY</td><td>:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp>3RD|3</td></tr>
<tr><td>4TH POLY</td><td>:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp>4TH|4</td></tr>
<tr><td>5TH POLY</td><td>:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp>5TH 5</td></tr>
<tr><td>POINT DISPLAY ON/OFF</td><td>:CALCulate:PARameter[:CATegory]:NF:PDISplay <wsp>OFF|ON|0|1</td></tr>
<tr><td>NF CALCULATION SETTING</td><td></td></tr>
<tr><td>RES BW</td><td>:CALCulate:PARameter[:CATegory]:NF:RBWidth<wsp> MEASured|CAL|0|1</td></tr>
<tr><td>SHOT NOISE</td><td>:CALCulate:PARameter[:CATegory]:NF:SNOise<wsp> OFF|ON|0|1</td></tr>
<tr><td>OTHER SETTING</td><td></td></tr>
<tr><td>SIGNAL POWER</td><td>:CALCulate:PARameter[:CATegory]:NF:SPOWer<wsp> PEAK|INTegral|0|1</td></tr>
<tr><td>INTEGRAL RANGE</td><td>:CALCulate:PARameter[:CATegory]:NF:IRANge<wsp> <NRf></td></tr>
</table>

ANALYSIS Parameters	Control Command
FILTER-PK	
PEAK LEVEL	
SW ON/OFF	CALCulate:PARameter[:CATegory]:FILPk<wsp>PLEVel,SW,OFF|ON|0|1
PEAK WAVELENGTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>PWAVelength,SW,OFF|ON|0|1
CENTER WAVELENGTH	
ALGO	:CALCulate:PARameter[:CATegory]:FILPk<wsp>MWAVelength,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]FILPk<wsp>MWAVelength,TH,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:FILPk<wsp>MWAVelength,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>MWAVelength,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp>MWAVelength,MDIFf,<NRf>[DB]
SPECTRUM WIDTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>SWIDth,SW,OFF|ON|0|1
ALGO	:CALCulate:PARameter[:CATegory]:FILPk<wsp>SWIDth,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp>SWIDth,TH,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]: FILPk<wsp>SWIDth,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>SWIDth,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp>SWIDth,MDIFf,<NRf>[DB]
CROSS TALK	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,SW,OFF|ON|0|1
ALGO	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,TH,<NRf>[DB]
K	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,K,<NRf>
MODE FIT ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,MFIT,OFF|ON|0|1
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,MDIFf,<NRf>[DB]
CH SPACE ±*.**nm	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,CSPace,<NRf>[M]
SEARCH AREA ±*.**nm	:CALCulate:PARameter[:CATegory]:FILPk<wsp>XTALk,SARea,<NRf>[M]
RIPPLE WIDTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILPk<wsp>RWIDth,SW,OFF|ON|0|1
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp> RWIDth,TH,<NRf>[DB]
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FILPk<wsp>RWIDth,MDIFf,<NRf>[DB]

ANALYSIS Parameters	Control Command
FILTER-BTM	
BOTTOM LEVEL	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>BLEVel,SW,OFF|ON|0|1
BOTTOM WAVELENGTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>BWAVelength,SW,OFF|ON|0|1
CENTER WAVELENGTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>CWAVelength,SW,OFF|ON|0|1
ALGO	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>CWAVelength,ALGO,<data>
THRESH LEVEL**.**dB	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>CWAVelength,TH,<NRf>[DB]
CENTER WAVELENGTH	
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>CWAVelength,MDIFf,<NRf>[DB]
NOTCH WIDTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>NWIDth,SW,OFF|ON|0|1
ALGO	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>NWIDth,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>NWIDth,TH,<NRf>[DB]
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]: FILBtm<wsp>NWIDth,MDIFf,<NRf>[DB]
CROSS TALK	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>XTALk,SW,OFF|ON|0|1
ALGO	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>XTALk,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>XTALk,TH,<NRf>[DB]
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>XTALk,MDIFf,<NRf>[DB]
CH SPACE ±*.**nm	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>XTALk,CSPace,<NRf>[M]
SEARCH AREA ±*.**nm	:CALCulate:PARameter[:CATegory]:FILBtm<wsp>XTALk,SARea,<NRf>[M]
WDM FIL-PK	
CHANNEL DETECTION/ NOMINAL WAVELENGTH	
ALGO	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>NWAVelength,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>NWAVelength,TH,<NRf>[DB]
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>NWAVelength,MDIFf,<NRf>[DB]
TEST BAND *.***nm	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>NWAVelength,TBANd<NRf>[DB]
PEAK WAVELENGTH/LEVEL	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>PWAVelength,SW,OFF|ON|0|1

ANALYSIS Parameters	Control Command
WDM FIL-PK	
XdB WIDTH/CENTER WAVELENGTH	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>CWAVelength,SW,OFF|ON|0|1
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>CWAVelength,TH,<NRf>[DB]
XdB STOP BAND	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>SBANd,SW,OFF|ON|0|1
THRESH LEVEL**.**dB	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>SBANd,TH,<NRf>[DB]
XdB PASS BAND	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>PBANd,SW,OFF|ON|0|1
THRESH LEVEL	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>PBANd,TH,<NRf>[DB]
TEST BAND *.***nm	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>PBANd,TBANd,<NRf>[DB]
RIPPLE	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>RIPPle,SW,OFF|ON|0|1
TEST BAND *.***nm	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>RIPPle,TBANd,<NRf>[DB]
CROSS TALK	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>XTALk,SW,OFF|ON|0|1
SPACING *.**nm	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>XTALk,SPACing,<NRf>[M]
TEST BAND *.***nm	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>XTALk,TBANd,<NRf>[DB]
WDM FIL-BTM	
CHANNEL DETECTION/ NOMINAL WAVELENGTH	
ALGO	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>NWAVelength,ALGO,<data>
THRESH LEVEL**.**dB	:CALCulate:PARameter[:CATegory]:WFPeak<wsp>WFBottom,TH,<NRf>[DB]
MODE DIFF *.**dB	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>NWAVelength,MDIFf,<NRf>[DB]
TEST BAND *.***nm	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>NWAVelength,TBANd<NRf>[DB]
BOTTM WAVELENGTH/LEVEL	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>BWAVelength,SW,OFF|ON|0|1
XdB NOTCH WIDTH/CENTER	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>CWAVelength,SW,OFF|ON|0|1
XdB STOP BAND	
ALGO	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>SBANd,ALGO,<data>
THRESH LEVEL **.**dB	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>SBANd,TH,<NRf>[DB]
XdB ELIMINATION BAND	
SW ON/OFF	:CALCulate:PARameter[:CATegory]:WFBottom<wsp>EBANd,SW,OFF|ON|0|1

ANALYSIS Parameters	Control Command
WDM FIL-BTM	
XdB ELIMINATION BAND	
THRESH LEVEL **.**dB	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> EBANd,TH,<NRf>[DB]`
TEST BAND *.***nm	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> EBANd,TBANd,<NRf>[DB]`
RIPPLE	
SW ON/OFF	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> RIPPle,SW,OFF|ON|0|1`
TEST BAND *.***nm	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> RIPPle,TBANd,<NRf>[DB]`
CROSS TALK	
SW ON/OFF	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> XTALk,SW,OFF|ON|0|1`
SPACING *.**nm	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> XTALk,SPACing,<NRf>[M]`
TEST BAND *.***nm	`:CALCulate:PARameter[:CATegory]:WFBottom<wsp> XTALk,TBANd,<NRf>[DB]`

Parameter Corresponding to <Analysis Parameter> of the Data Logging Function

The parameter corresponding to <Analysis Parameter> accessed through ADVANCE -> <Data Logging> -> <SETUP> varies depending on the logging item.

- **When the Logging Item Is WDM**
  The Analysis 2 parameter accessed through the ANALYSIS key in this section corresponds to <Analysis Parameter>.

- **When the Logging Item Is DFB-LD**
  The Analysis 1 parameter accessed through the ANALYSIS key in this section corresponds to <Analysis Parameter>.

6.4 Remote Command Tree

Command	Parameter	Page
COMMON command		
*CLS	none	6-38
*ESE	<integer>	6-38
*ESE?	none	6-38
*ESR?	none	6-38
*IDN?	none	6-38
*OPC	none	6-38
*OPC?	none	6-38
*RST	none	6-39
*SRE	<integer>	6-39
*SRE?	none	6-39
*STB?	none	6-39
*TRG	none	6-39
*TST?	none	6-39
*WAI	none	6-39
ABORt	none	6-40
APPLication		
:DLOGging		
:ETIMe?	none	6-40
:LPARameter		
:INTerval	<integer>	6-40
:ITEM	0|1|2|3	6-41
:LMODe	1|2	6-41
:MEMory	INTernal|EXTernal	6-41
:MTHResh	<NRf>	6-41
:PDETect		
:ATHResh	<NRf>	6-42
:RTHResh	<NRf>	6-42
:TTYPe	ABSolute|RELative	6-42
:TDURation	<integer>	6-42
:TLOGging	OFF|ON|0|1	6-42
:STATe	STOP|STARt|0|1	6-42
CALCulate		
:AMARker[1|2|3|4]		
:AOFF	none	6-43
:FUNCtion		
:INTegral		
:IRANge	<NRf>[Hz]	6-43
:RESult?	none	6-43
[:STATe]	OFF|ON|0|1	6-44
:PDENsity|NOISe		
:BWIDth|BANDwidth	<NRf>[M]	6-44
:RESult?	none	6-44
[:STATe]	OFF|ON|0|1	6-45
:PRESet	none	6-45
:MAXimum	none	6-45
:LEFT	none	6-45
:NEXT	none	6-45
:RIGHt	none	6-46
:MINimum	none	6-46
:LEFT	none	6-46
:NEXT	none	6-46
:RIGHt	none	6-46
[:STATe]	OFF|ON|0|1	6-46
:TRACe	TRA|TRB|TRC|TRD|TRE|TRF|TRG	6-47
:X	<NRf>[M|Hz]	6-47
:Y?	none	6-47
:ARESolution?	<Trace name>,[<start point>,<stop point>]	6-47
:CATegory	SWTHresh|SWENvelope|SWRMs|SWPKrms| NOTCh|DFBLd| FPLD|LED|SMSR|POWer| PMD|WDM|NF|FILPk|FILBtm|WFPeak| WFBtm|COLor	6-48

Command	Parameter	Page
:DATA?	none	6-48
:CGAin?	none	6-48
:CNF?	none	6-48
:COLor?	none	6-49
:CPOWers?	none	6-49
:CSNR?	none	6-49
:CWAVelengths	none	6-49
:DFBLd?	none	6-50
:NCHannels	none	6-50
:OSLope?	none	6-50
:DISPlay	0|1|2|3|4	6-50
:GRAPh:LMARker:Y	1|2,<NRf>[DB]	6-50
[:IMMediate]	none	6-51
:AUTO	OFF|ON|0|1	6-51
:LMARker		
:AOFF	none	6-51
:SRANge	OFF|ON|0|1	6-51
:SSPan	none	6-51
:SZSPan	none	6-51
:X	1|2,<NRf>[M|HZ]	6-51
:Y	3|4,<NRf>[DBM/DB/%]	6-51
:MARKer		
:AOFF	none	6-51
:AUTO	OFF|ON|0|1	6-52
:FUNCtion		
:FORMat	OFFSet|SPACing|0|1	6-52
:UPDate	OFF|ON|0|1	6-52
:MAXimum	none	6-52
:LEFT	none	6-52
:NEXT	none	6-52
:RIGHt	none	6-52
:SCENter	none	6-52
:AUTO	OFF|ON|0|1	6-52
:SRLevel	none	6-53
:AUTO	OFF|ON|0|1	6-53
:SZCenter	none	6-53
:MINimum	none	6-53
:LEFT	none	6-53
:NEXT	none	6-53
:RIGHt	none	6-53
:MSEarch	OFF|ON|0|1	6-53
:SORT	WAVelength|LEVel|0|1	6-53
:THResh	<NRf>[DB]	6-54
:SCENter	none	6-54
:SRLevel	none	6-54
[:STATe]	<marker>,OFF|ON|0|1	6-54
:SZCenter	none	6-54
:UNIT	WAVelength|FREQuency	6-54
:X	<marker>,<NRf> [M|HZ]	6-54
:Y?	<marker>	6-55
:MATH		
:TRC	A-B(LOG)|B-A(LOG)|A+B(LOG)|A+B(LIN)| A-B(LIN)|B-A(LIN)|1-K(A/B)|1-K(B/A)	6-55
:K	<NRf>	6-55
:TRF	C-D(LOG)|D-C(LOG)|C+D(LOG)|D-E(LOG)| E-D(LOG)|D+E(LOG)|C+D(LIN)|C-D(LIN)| D-C(LIN)|D+E(LIN)|D-E(LIN)|E-D(LIN)| PWRNBWA|PWRNBWB|PWRNBWC|PWRNBWD| PWRNBWE	6-55
:PNBW:BWIDth	<NRf>[M]	6-55
:TRG	C-F(LOG)|F-C(LOG)|C+F(LOG)|E-F(LOG)| F-E(LOG)|E+F(LOG)|C+F(LIN)|C-F(LIN)| F-C(LIN)|E+F(LIN)|E-F(LIN)|F-E(LIN)| NORMA|NORMB|NORMC|CVFTA|CVFTB|CVFTC| MKRFT|PKCVFTA|PKCVFTB|PKCVFTC	6-55

Command	Parameter	Page
:CVFT		
:FALGo	GAUSS|LORENz|3RD|4TH|5TH|0|1|2|3|4	6-56
:OPARea	ALL|INL1-L2|OUTL1-L2|0|1|2	6-56
:THResh	<integer>[DB]	6-56
:PCVFt:THResh	<integer>[DB]	6-56
:PARameter		
[:CATegory]		
:DFBLd	<item>,<paramater name>,<data>	6-56
:FILBtm	<item>,<paramater name>,<data>	6-57
:FILPk	<item>,<paramater name>,<data>	6-57
:FPLD	<item>,<paramater name>,<data>	6-58
:LED	<item>,<paramater name>,<data>	6-58
:NF		
:AALGo	AFIX|MFIX|ACENter|MCENter|0|1|2|3|	6-59
:FALGo	LINear|GAUSs|LORenz|3RD|4TH|5TH| 0|1|2|3|4|5	6-59
:FARea	<NRf>[M]	6-59
:IOFFset	<NRf>[DB]	6-59
:IRANge	<NRf>	6-59
:MARea	<NRf>[M]	6-60
:MDIFf	<NRf>[DB]	6-60
:OOFFset	<NRf>[DB]	6-60
:PDISplay	OFF|ON|0|1	6-60
:TH	<NRf>[DB]	6-60
:RBWidth	MEASURED|CAL|0|1	6-60
:SNOise	OFF|ON|0|1	6-61
:SPOWer	PEAK|INTegral|0|1	6-61
:NOTCh		
:K	<NRf>	6-61
:TH	<NRf>[DB]	6-61
:TYPE	PEAK|BOTTom|0|1	6-61
:PMD		
:TH	<NRf>[DB]	6-61
:POWer		
:OFFSet	<NRf>[DB]	6-62
:SMSR		
:MASK	<NRf>[M]	6-62
:MODE	SMSR1|SMSR2|SMSR3|SMSR4	6-62
:SWENvelope		
:K	<NRf>	6-62
:TH1	<NRf>[DB]	6-62
:TH2	<NRf>[DB]	6-62
:SWPKrms		
:K	<NRf>	6-62
:TH	<NRf>[DB]	6-63
:SWRMs		
:K	<NRf>	6-63
:TH	<NRf>[DB]	6-63
:SWTHresh		
:K	<NRf>	6-63
:MFIT	OFF|ON|0|1	6-63
:TH	<NRf>[DB]	6-63
:WDM		
:DMASk	<NRf>[DB]	6-64
:DTYPe	ABSolute|RELative|MDRIft|GDRIft| 0|1|2|3	6-64
:DUAL	OFF|ON|0|1	6-64
:FALGo	LINear|GAUSs|LORenz|3RD|4TH|5TH| 0|1|2|3|4|5	6-64
:IRANge	<NRf>	6-65
:MARea	<NRf>[M]	6-65
:MDIFf	<NRf>[DB]	6-65
:MMReset	None	6-65
:NALGo	AFIX|MFIX|ACENter|MCENter|PIT| 0|1|2|3|4	6-65
:NARea	<NRf>[M]	6-65
:NBW	<NRf>[M]	6-65
:OSLope	OFF|ON|0|1	6-66
:PDISplay	OFF|ON|0|1	6-66

Command	Parameter	Page
:RCH	<integer>	6-66
:RELation	OFFSet | SPACing|0|1	6-66
:SPOWer	PEAK|INTegral|0|1	6-66
:TH	<NRf>[DB]	6-66
:WFBottom	<item>,<paramater name>,<data>	6-67
:WFPeak	<item>,<paramater name>,<data>	6-67
:COMMON		
:MDIFf	<NRf>[DB]	6-67
CALibration		
:ALIGn		
[:IMMediate]	none	6-68
:WAVelength		
:EXTernal		
[:IMMediate]	none	6-68
:SOURce	LASEr|GASCell|EMISsion	6-68
:WAVelength	<NRf>M	6-68
:INTernal[:IMMediate]	none	6-68
:ZERO[:AUTO]	OFF|ON|0|1|ONCE	6-69
:INTerval	<integer>	6-69
:STATus?	none	6-69
DISPlay		
:COLor	0|1	6-69
[:WINDow]	OFF|ON|0|1	6-69
:OVIew		
:POSition	OFF|LEFT|RIGHt|0|1|2	6-69
:SIZE	LARGe|SMALl|0|1	6-69
:SPLIt	OFF|ON|0|1	6-70
:HOLD		
:LOWer	OFF|ON|0|1	6-70
:UPPer	OFF|ON|0|1	6-70
:POSition	<trace name>,UP|LOW|0|1	6-70
:TEXT		
:CLEar	none	6-70
:DATA	<"string">	6-70
:TRACe		
:X[:SCALe]		
:CENTer	<NRf>[M|HZ]	6-70
:INITialize none		6-70
:SMSCale	none	6-71
:SPAN	<NRf>[M|HZ]	6-71
:SRANge	OFF|ON|0|1	6-71
:STARt	<NRf>[M|HZ]	6-71
:STOP	<NRf>[M|HZ]	6-71
:Y		
:NMASk	<NRf>DB	6-71
:TYPE	VERTical|HORizontal|0|1	6-72
[:SCALe]		
:DNUMber 8|10|12		6-72
:Y1		
[:SCALe]		
:BLEVel	<NRf>[W|MW|UW|NW]	6-72
:PDIVision	<NRf>[DB]	6-72
:RLEVel	<NRf>[DBM|W|	6-72
:RPOSition	<integer>[DIV]	6-73
:SPACing LOGarithmic|LINear|0|1		6-73
:UNIT	DBM|W|DBM/NM|W/NM|0|1|2|3	6-73

Command	Parameter	Page
:Y2		
[:SCALe]		
:AUTO	OFF|ON|0|1	6-73
:LENGth	<NRf>[KM]	6-73
:OLEVel	<NRf>[DB|DB/KM]	6-73
:PDIVision	<NRf>[DB|DB|KM|%]	6-74
:RPOSition	<integer>[DIV]	6-74
:SMINimum	<NRf>[%]	6-74
:UNIT	DB|LINear|DB/KM|%|0|1|2|3	6-74
FORMat		
[:DATA]	REAL[,64|,32]|ASCii	6-75
HCOPy		
:DESTination	FILE|2	6-75
[:IMMediate]	none	6-75
INITiate		
[:IMMediate]	none	6-76
:SMODe	SINGle|REPeat|AUTO|SEGment|1|2|3|4	6-76
MEMory		
:CLEar	<integer>	6-76
:EMPty?	<integer>	6-76
:LOAD	<integer>,<trace name>	6-76
:STORe	<integer>,<trace name>	6-76
MMEMory		
:ANAMe	NUMBer|DATE	6-77
:CATalog?	[INTernal|EXTernal]	6-77
:CDIRectory	<"directory name">	6-77
:CDRive	INTernal|EXTernal	6-77
:COPY	<"source file name">, [INTernal|EXTernal], <"destination file name">[,INTernal|EXTernal]	6-77
:DATA?	<"file name">[,INTernal|EXTernal]	6-77
:DELete	<"file name">[,INTernal|EXTernal]	6-78
:LOAD		
:ATRace	<”file name”>[,INTernal|EXTernal]	6-78
:DLOGing	<”file name”>[,INTernal|EXTernal]	6-78
:MEMory	<integer>,<"filename">[,INTernal|EXTernal]	6-78
:PROGram	<integer>,<"filename">[,INTernal|EXTernal]	6-78
:SETTing	<"filename">[,INTernal|EXTernal]	6-78
:TEMPlate	<template>,<"filename">[,INTernal|EXTernal]	6-78
:TRACe	<trace name>,<"filename">[,INTernal|EXTernal]	6-78
:MDIRectory	<"directory name">[,INTernal|EXTernal]	6-79
:REMove	None	6-79
:REName	<"new file name">,<"old file name">[,INTernal|EXTernal]	6-79
:STORe		
:ARESult	<”filename”>[,INTernal|EXTernal]	6-79
:ATRace	<”file name”>[,INTernal|EXTernal]	6-79
:DATA	<"filename">,[,INTernal|EXTernal]	6-79
:ITEM	DATE|LABel|DATA|CONDition|TRACe,OFF|ON|0|1	6-79
:MODE	ADD|OVER|0|1	6-79
:TYPE	CSV|DT|0|1	6-80
:DLOGging	<”filename”>[,INTernal|EXTernal]	6-80
:CSAVe	OFF|ON|0|1	6-80
:TSAVe	OFF|ON|0|1	6-80

Command	Parameter	Page
:GRAPhics	B&W|COLor|PCOLor,BMP|TIFF,<”filename”> [,INTernal|EXTernal]	6-80
:MEMory	<integer>,BI|CSV,<”filename”> [,INTernal|EXTernal]	6-80
:PROGram	<integer>,<”filename”>[,INTernal| EXTernal]	6-80
:SETTing	<”filename”>[,INTernal|EXTernal]	6-81
:TEMPlate	<template>,<”filename”>[,INTernal| EXTernal]	6-81
:TRACe	<trace name>,BIN | CSV,<”filename”> [,INTernal|EXTernal]	6-81
PROGram		
:EXECute	<integer>	6-81
SENSe		
:AVERage:COUNt	<integer>	6-82
:BANDwidth|:BWIDth [:RESolution]	<NRf>[M|Hz]	6-82
:CHOPper	OFF|SWITch|0|2	
:CORRection		
:LEVel:SHIFt	<NRf>[DB]	6-82
:RVELocity:MEDium	AIR|VACuum|0|1	6-82
:WAVelength:SHIFt	<NRf>[M]	6-82
:SENSe	NHLD|NAUT|NORMal|MID|HIGH1| HIGH2|HIGH3	6-82
:SETTing		
:FIBer	SMALI|LARGe|0|1	6-83
:SMOothing	OFF|ON|0|1	6-83
:SWEep		
:POINts	<integer>	6-83
:AUTO	OFF|ON|0|1	6-83
:SEGMent:POINts	<integer>	6-83
:SPEed	1x|2x|0|1	6-83
:STEP	<NRf>[M]	6-84
:TIME		
:0NM	<integer>[SEC]	6-84
:INTerval	<integer>[SEC]	6-84
:WAVelength		
:CENTer	<NRf>[M|HZ]	6-84
:SPAN	<NRf>[M|HZ]	6-84
:SRANge	OFF|ON|0|1	6-84
:STARt	<NRf>[M|HZ]	6-84
:STOP	<NRf>[M|HZ]	6-85
STATus		
:OPERation		
:CONDition?	none	6-85
:ENABl	<integer>	6-85
[:EVENt]?	none	6-85
:PRESet	none	6-85
:QUEStionable		
:CONDition?	none	6-86
:ENABle	<integer>	6-86
[:EVENt]?	none	6-86

Command	Parameter	Page
SYSTem		
:BUZZer		
:CLIC	OFF|ON|0|1	6-86
:WARNing	OFF|ON|0|1	6-86
:COMMunicate		
:CFORmat	AQ6317|AQ6374|0|1	6-87
:LOCKout	OFF|ON|0|1	6-87
:RMONitor	OFF|ON|0|1	6-87
:DATE	yyyy,mm,dd	6-87
:DISPlay		
:TRANsparent	OFF|ON|0|1	6-87
:UNCal	OFF|ON|0|1	6-87
:ERRor		
[:NEXT]?	none	6-87
:GRID	12.5GHZ|25GHz|50GHZ|100GHZ|200GHZ |CUSTom|0|1|2|3|4|5	6-88
:CUSTom		
:CLEar:ALL	none	6-88
:DELete	<grid number>	6-88
:INSert	<NRf>[M|HZ]	6-88
:SPACing	<NRf>[GHZ]	6-88
:STARt	<NRf>[M|HZ]	6-88
:STOP	<NRf>[M|HZ]	6-88
:REFerence	<NRf>[M|HZ]	6-88
:INFormation?	0|1	6-89
:OLOCK	OFF|ON|0|1,<"password">	
:OPERation		
:SHUTdown	none	6-89
:REBoot	none	6-89
:PRESet	none	6-89
:TIME	hh,mm,ss	6-89
:VERSion?		6-89

Command	Parameter	Page
TRACe		
:ACTive	<trace name>	6-90
:ATTRibute[:<trace name>]	WRITe|FIX|MAX|MIN|RAVG|CALC	6-90
:RAVG[:<trace name>]	<integer>	6-90
:COPY	<source trace>,<destination trace>	6-90
[:DATA]		
:SNUMber?	<trace name>	6-90
:X?	<trace name>[,<start point>, <stop point>]	6-91
:Y?	<trace name>[,<start point>, <stop point>]	6-91
:PDENsity?	<trace name>,<NRF>[,<start point>, <stop point>	6-91
:DELete	<trace name>	6-91
:ALL		6-91
:STATe[:<trace name>]	OFF|ON|0|1	6-92
:TEMPlate		
:DATA	<template>,<wavelength>,<level>	6-92
:ADELete	<template>	6-92
:ETYPe	<template>,NONE|A|B|0|1|2	6-92
:MODE	<template>,ABSolute|RELative|0|1	6-92
:DISPlay	<template>,OFF|ON|0|1	6-93
:GONogo	OFF|ON|0|1	6-93
:LEVel:SHIFt	<NRf>[DB]	6-93
:RESult?		6-93
:TTYPe	UPPer|LOWer|U&L|0|1|2	6-93
:WAVelength:SHIFt	<NRf>[M]	6-93
TRIGger		
[:SEQuence]		
:DELay	<NRf>[S|MS|US]	6-94
:SLOPe	RISE|FALL|0|1	6-94
:STATe	OFF|ON|PHOLd|0|1|2	6-94
:INPut	ETRigger|STRigger|SENable|0|1|2	6-94
:OUTPut	OFF|SSTatus|0|1	6-94
:PHOLd:HTIMe	<NRf>[s]	6-94
UNIT		
:POWer:DIGit	1|2|3	6-95
:X	WAVelength|FREQuency|0|1	6-95

6.5 Common Commands

The instrument supports the “Required” common commands listed in the table below.

Cmd	Name	IEEE 488.2 Std.
*AAD	Accept Address Command	Option
*CAL?	Calibration Query	Option
*CLS	Clear Status Command	Required
*DDT	Define Device Trigger Command	*DT1 option
*DDT?	Define Device Trigger Query	DT1 option
*DLF	Disable Listener Function Command	Option
*DMC	Define Macro Command	Option
*EMC	Enable Macro Command	Option
*EMC?	Enable Macro Query	Option
*ESE	Standard Event Status Enable Command	Required
*ESE?	Standard Event Status Enable Query	Required
*ESR?	Standard Event Status Register Query	Required
*GMC?	Get Macro Contents Query	Option
*IDN?	Identification Query	Required
*IST?	Individual Status Query	Required for PP1
*LMC?	Learn Macro Query	Option
*LRN?	Learn Device Setup Query	Option
*OPC	Operation Complete Command	Required
*OPC?	Operation Complete Query	Required
*OPT	Option Identification Query	Option
*PCB	Pass Control Back Command	Required if not C0
*PMC	Purge Macro Command	Option
*PRE	Parallel Poll Register Enable Command	Required for PP1
*PRE?	Parallel Poll Register Enable Query	Required for PP1
*PSC	Power On Status Clear Command	Option
*PSC?	Power On Status Clear Query	Option
*PUD	Protected User Data Command	Option
*PUD?	Protected User Data Query	Option
*RCL	Recall Command	Option
*RDT	Resource DescriptionTransfer Command	Option
*RDT?	Resource Description Transfer Query	Option
*RST	Reset Command	Required
*SAV	Save Command	Option
*SRE	Service Request Enable Command	Required
*SRE?	Service Request Enable Query	Required
*STB?	Read Status Byte Query	Required
*TRG	Trigger Command	Required if DT1
*TST?	Self-Test Query	Required
*WAI	Wait-to-Continue Command	Required

*CLS(Clear Status)

Function Clears all event status registers, the summary of which is reflected in the status byte register.

Syntax `*CLS`

Example `*CLS`

Explanation
- Clears all queues, with the exception of the output queue, and all event registers, with the exception of the MAV summary message.
- After executing this command, OCIS (Operation Complete Command Idle State) and OQIS (Operation Complete Query Idle State) are brought about.
- This is a sequential command.

*ESE(Standard Event Status Enable)

Function Sets/queries the standard event enable register.

Syntax `*ESE<wsp><integer>`
`*ESE?`
<integer> = 0–255

Example `*ESE 251`
`*ESE? -> 251`

Explanation
- An item having had its bit set becomes enabled.
- Resets to the default value in the following cases:
  When power is ON
  When “0” is set
- The set value remains the same in the following cases:
  *RST
  *CLS
  Device clear (DCL, SDC)
- The default is 0.
- This is a sequential command.

*ESR?(Standard Event Status Register)

Function Queries the standard event status register and simultaneously clears it.

Syntax `*ESR?`

Example `*ESR? -> 251`

Explanation
- The return value of this query is not affected by ESE (Event Status Enable Register).
- This is an overlapping command.

*IDN? (Identification)

Function Queries the instrument type and firmware version.

Syntax `*IDN?`

Example `*IDN? ->`
`YOKOGAWA,AQ6374,aaaaaaaaa,bb.bb`
`aaaaaaaaa:` Serial number (9 digit string)
`bb.bb:` Firmware version

Explanation
- Outputs 4 field data delimited by a comma.
  Field 1: Manufacturer “YOKOGAWA”
  Field 2: Model “AQ6374”
  Field 3: Instrument serial number
  Field 4: Firmware version
- This is a sequential command.

*OPC(Operation Complete)

Function Sets/queries bit 0 (OPC) of the standard event status register (ESR) if operations waiting to be processed have all been completed.

Syntax `*OPC`
`*OPC?`

Example `*OPC`
`*OPC? -> 1`

Explanation
- At the time this command is recognized, the command changes from OCIS (Operation Complete Command Idle State) to OCAS (Operation Complete Command Active State). When the no-operation pending flag is set to “True,” it sets bit 0 (OCR) of ESR and returns to OCIS.
- If any of the following conditions are established, this command is disabled and is forced to return to OCIS.
  Power ON
  Device clear
  *CLS, *RST command
- This is an overlapping command.

*RST (Reset)

Function Executes a device reset to return the instrument to the known (default) status.

Syntax `*RST`

Example `*RST`

Explanation
- Stops operation being processed and returns the instrument to the known set value (default value) immediately.
- This unit’s parameters are cleared.
- The following items will remain the same.
  GP-IB interface status
  GP-IB address
  Output queue
  SRE
  ESE
  Calibration data affecting the instrument’s specifications
- This is an overlapping command.

*SRE(Service Request Enable)

Function Sets/queries the service request enable register.

Syntax `*SRE <wsp><integer>`
`*SRE?`
<integer> = 0–255

Example `*SRE 250`
`*SRE? -> 250`

Explanation
- An item having had its bit set becomes enabled.
- Resets to the default value in the following cases:
  When power is ON
  When “0” is set
- The set value remains the same in the following cases:
  *RST
  *CLS
  Device clear (DCL, SDC)
- The default is 0.
- This is a sequential command.

*STB?(Read Status Byte)

Function Queries the current value of the status byte register.

Syntax `*STB?`

Example `*STB? -> 251`

Explanation
- STB will not be cleared even when the contents of the register are read.
- This is an overlapable command.

*TRG(Trigger)

Function Performs a <SINGLE> sweep under the sweep conditions established immediately before receiving the command.

Syntax `*TRG`

Example `*TRG`

Explanation Performs a <SINGLE> sweep regardless of the setting condition of the `:INITiate:CONTinuous` command.
This is an overlapable command.

*TST?(Self Test)

Function Performs the instrument’s self-test and queries the status.

Syntax `*TST?`

Example `*TST? -> 0`

Explanation
- Of the initialization sequence to be run at startup, this command executes the following operations to output their results. During initialization, the screen maintains the waveform display.
  Motor’s return to origin operation
  AMP auto-offset
- Normally returns 0, or 1 for motor initialize error, or 2 for AMP offset error.
- This is a sequential command.

*WAI(Wait to Continue)

Function Prevents the instrument from executing another command until the execution of the current command is complete.

Syntax `*WAI`

Example `*WAI`

Explanation
- Becomes invalid by device clear.
- Meaningful if subsequent commands are overlapping. Meaningless with other commands.
- This is a sequential command.

6.6 Instrument-Specific Commands

ABORt Sub System Command

:ABORt

Function Stops operations such as measurements and calibration.

Syntax `ABORt`

Example `ABORt`

Explanation
- Operations to be stopped are as follows:
  ```
  :APPLication:DLOGging:STATe
  :CALibration:ALIGn[:IMMediate]
  :CALibration:ALIGn:INTernal[:IMMe
  diate]
  :CALibration:WAVelength
  :INITiate
  :PROGram:EXECute
  :HCOPy[:INITiate]
  ```
- This is an overlapping command.

APPLication Sub System Commands

Overview

- This subsystem consists of data logging commands.

:APPLication:DLOGging:ETIMe?

Function Queries the elapsed time of data logging (in seconds).

Syntax `:APPLication:DLOGging:ETIMe?`

Response <integer>

<integer> = Elapsed time [sec]

Example `:APPLICATION:DLOGGING:ETIME? -> 10220`

Description
- This is an overlap command.
- This command is invalid when data logging is paused.

:APPLication:DLOGging:LPARameter:INTerval

Function Sets or queries the measurement interval of data logging.

Syntax
```
:APPLication:DLOGging:LPARameter:IN
Terval<wsp><integer>[SEC]
:APPLication:DLOGging:LPARameter:IN
Terval?
```
<integer> = Measurement interval [sec] (0 = SWEEP TIME)

Example
```
:APPLICATION:DLOGGING:LPARAMETER:IN
TERVAL 10
:APPLICATION:DLOGGING:LPARAMETER:IN
TERVAL? -> 10
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:ITEM

Function Sets or queries the data logging source.

Syntax
```
:APPLication:DLOGging:LPARameter:ITEM<wsp>0|1|2|3
:APPLication:DLOGging:LPARameter:ITEM?
```
0|1|2|3: Data logging source
0 = WDM, 1 = PEAK, 2 = MULTI-PEAK, 3 = DFB-LD

Example
```
:APPLICATION:DLOGGING:LPARAMETER:ITEM 0
:APPLICATION:DLOGGING:LPARAMETER:ITEM? -> 0
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:LMODe

Function Sets or queries the data logging mode (maximum channel mode or maximum logging mode).

Syntax
```
:APPLication:DLOGging:LPARameter:LMODe<wsp>1|2
:APPLication:DLOGging:LPARameter:LMODe?
```
1|2: Mode
1 = Maximum channel mode (MODE1: MAX 1024ch, 2001 entries)
2 = Maximum logging mode (MODE2: MAX 256ch, 10001 entries)

Example
```
:APPLICATION:DLOGGING:LPARAMETER:LMODE 1
:APPLICATION:DLOGGING:LPARAMETER:LMODE -> 1
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:MEMory

Function Sets or queries the temporary area for saving waveform files of data logging.

Syntax
```
:APPLication:DLOGging:LPARameter:MEMory<wsp>INTernal|EXTernal
:APPLication:DLOGging:LPARameter:MEMory?
```
INTernal = Internal memory
EXTernal = USB storage media

Example
```
:APPLICATION:DLOGGING:LPARAMETER:MEMORY INTERNAL
:APPLICATION:DLOGGING:LPARAMETER:MEMORY? -> INT
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:MTHResh

Function Sets or queries the threshold of the channel-matching wavelength λ for data logging.

Syntax
```
:APPLication:DLOGging:LPARameter:MTHResh<wsp><NRf>[M]
:APPLication:DLOGging:LPARameter:MTHResh?
```
<NRf>[M] = Threshold of wavelength λ [m]

Example
```
:APPLICATION:DLOGGING:LPARAMETER:MTHResh 0.1nm
:APPLICATION:DLOGGING:LPARAMETER:MTHResh? -> +1.00000000E-010
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:PDETect:ATHResh

Function Sets or queries the threshold (absolute value) for detecting the data logging mode.

Syntax
```
:APPLication:DLOGging:LPARameter:PDETect:ATHResh<NRf>[DBM]
:APPLication:DLOGging:LPARameter:PDETect:ATHResh?
```
<NRf>[DBM] = Peak detection threshold (absolute value) [dBm]

Example
```
:APPLICATION:DLOGGING:LPARAMETER:PDETECT:ATHRESH -20.0dbm
:APPLICATION:DLOGGING:LPARAMETER:PDETECT:ATHRESH? -> -2.00000000E+001
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:PDETect:RTHResh

Function Sets or queries the threshold (relative value) for detecting the data logging mode.

Syntax
```
:APPLication:DLOGging:LPARameter:PDETect:RTHResh<NRf>[DB]
:APPLication:DLOGging:LPARameter:PDETect:RTHResh?
```
<NRf>[DB] = Peak detection threshold (relative value) [dB]

Example
```
:APPLICATION:DLOGGING:LPARAMETER:PDETECT:RTHRESH 30.0db
:APPLICATION:DLOGGING:LPARAMETER:PDETECT:RTHRESH? -> +3.00000000E+001
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:PDETect:TTYPe

Function Sets or queries how the threshold for detecting the data logging mode (peak or bottom) is specified.

Syntax
```
:APPLication:DLOGging:LPARameter:PDETect:TTYPe<wsp>ABSolute|RELative
:APPLication:DLOGging:LPARameter:PDETect:TTYPe?
```
ABSolute = Absolute value

RELative = Relative value

Example
```
:APPLICATION:DLOGGING:LPARAMETER:PDETECT:TTYPE ABSOLUTE
:APPLICATION:DLOGGING:LPARAMETER:PDETECT:TTYPE? -> ABS
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:TDURation

Function Sets or queries the measurement duration of data logging (in seconds).

Syntax
```
:APPLication:DLOGging:LPARameter:TDURation<wsp><integer>[sec]
:APPLication:DLOGging:LPARameter:TDURation?
```
<integer> = Measurement duration [sec]

Example
```
:APPLICATION:DLOGGING:LPARAMETER:TDURation 3600
:APPLICATION:DLOGGING:LPARAMETER:TDURation? -> 3600
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:LPARameter:TLOGging

Function Sets or queries whether waveforms will be logged during data logging.

Syntax
```
:APPLication:DLOGging:LPARameter:TLOGging<wsp>OFF|ON|0|1
:APPLication:DLOGging:LPARameter:TLOGging?
```
OFF = Waveform data save function off

ON = Waveform data save function on

Example
```
:APPLICATION:DLOGGING:LPARAMETER:TLOGGING OFF
:APPLICATION:DLOGGING:LPARAMETER:TLOGGING? -> 0
```

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:APPLication:DLOGging:STATe

Function Starts, stops, or queries data logging.

Syntax
```
:APPLication:DLOGging:STATe<wsp>STOP|STARt|0|1
:APPLication:DLOGging:STATe?
```
START = Starts data logging

STOP = Stops data logging

Response 0 = Stopped, 1 = Running

Example
```
:APPLICATION:DLOGGING:STATE 1
:APPLICATION:DLOGGING:STATE? -> 1
```

Description
- Only the following commands are valid when data logging is in progress.
  - Common commands (excluding *TRG and *TST)
  - All query commands
  - ABORt
- This is an overlappable command.

CALCulate Sub System Command

Outline

- Commands about the following functions are summarized in this sub system.
  - Analysis function (Spectrum Width, ANALYSIS1 , ANALYSIS2)
  - Peak/Bottom search function
  - Marker function (Δ marker, line marker)
  - Calculation function of trace
  - Advanced marker function (moving marker, power spectral density marker, integrated power marker)
- The following procedure is performed in order to carry out remote control of the Analysis function.
  1. Select the analysis algorithm (`CALCulate:CATegory` command)
  2 Set the Analysis Parameter (`CALCulate:PARameter` command)
  3 Execute the analysis function (`CALCulate[:IMMediate]` command)
  4 Get the analysis results (`CALCulate:DATA?` command)
- The following command is used in order to carry out remote control of the Peak/Bottom search function. `CALCulate:MARKer:MAXimum|MINimum` command
- The following command is used to in order to carry out remote control of the Marker function.
  Δ marker: `CALCulate:MARKer` command
  Line marker: `CALCulate:LMARker` command
- The following command is used to in order to carry out remote control of the trace Calculation function. `CALCulate:MATH` command

:CALCulate:AMARker[1|2|3|4]:AOFF

Function Clears all advanced markers (moving markers, power spectral density markers, and integrated power markers) and turns off the advanced marker function.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:AOFF
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:AOFF
```

Description
- After clearing, the advanced marker function automatically turns off.
- All advanced markers are cleared regardless of which advanced marker you specify.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion:INTegral:IRANge

Function Sets or queries the integration frequency range of the specified integrated power marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:FUNCtio
n:INTegral:IRANge<wsp><NRf>[Hz]
:CALCulate:AMARker[1|2|3|4]:FUNCtio
n:INTegral:IRANge?
[1|2|3|4]: Advanced marker number
```
<NRf> = Integration frequency range [Hz]

Example
```
:CALCULATE:AMARKER:FUNCTION:INTEGRA
L:IRANge 40GHz
:CALCULATE:AMARKER:FUNCTION:INTEGRA
L:IRANge? -> 4.00000000E+010
```

Description
- An execution error will occur if the specified advanced marker has not been assigned or is not an integrated power marker.
- A query error will occur if the specified advanced marker has not been assigned or is not an integrated power marker.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion:INTegral:RESult?

Function Queries the integration value of the specified integrated power marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:FUNCtio
n:INTegral:RESult?
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:FUNCTION:INTEGRA
L:RESULT? -> -1.00000000E+001
```

Description
- The unit of the returned marker level depends on the Y-axis unit of the assigned marker trace.
- A query error will occur if the specified advanced marker has not been assigned or is not an integrated power marker.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion :INTegral[:STATe]

Function Moves the specified integrated power marker to the center of the marker trace.
Also queries the status of the specified integrated power marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:FUNCtion:INTegral[:STATe]<wsp>OFF|ON|0|1
:CALCulate:AMARker[1|2|3|4]:FUNCtion:INTegral[:STATe]?
[1|2|3|4]: Advanced marker number
```
Response 0 = Off, 1 = On

Example
```
:CALCULATE:AMARKER:FUNCTION:INTEGRAL ON
:CALCULATE:AMARKER:FUNCTION:INTEGRAL? -> 1
```

Description
- If an integrated power marker is assigned, moving markers (:CALCulate:AMARker[1|2|3|4][:STATe]) and power spectral density markers (:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|:NOISe[:STATe]) will be set to off.
- If this command with the parameter set to OFF is specified on an advanced marker that has been assigned to integrated power marker, the advanced marker will change to a moving marker.
  If the advanced marker has not been assigned, using this command with the parameter set to OFF will leave the advanced marker unassigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion :PDENsity|:NOISe:BWIDth|:BANDwidth

Function Sets or queries the normalization bandwidth of the specified power spectral density marker.

Syntax
```
:CALCulate:AMARker:FUNCtion:PDENsity|:NOISe:BWIDth|:BANDwidth<wsp><NRf>[m]
:CALCulate:AMARker:FUNCtion:PDENsity|:NOISe:BWIDth|:BANDwidth}?
```
<NRf> = Normalization bandwidth [m]

Example
```
:CALCULATE:AMARKER:FUNCTION:PDENSITY:BWIDTH 0.1nm
:CALCULATE:AMARKER:FUNCTION:PDENSITY:BWIDTH -> +1.00000000E-010
```

Description
- This command applies to advanced markers 1 to 4. The command operates in the same manner regardless of which advanced marker is specified.
- An execution error will occur if the specified advanced marker has not been assigned or is not a power spectral density marker.
- A query error will occur if the specified advanced marker has not been assigned or is not a power spectral density marker.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion :PDENsity|:NOISe:RESult?

Function Sets or queries the power spectral density value of the specified power spectral density marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|:NOISe:RESult?
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:FUNCTION:PDENSITY:RESULT? -> -1.00000000E+001
```

Description
- The unit of the returned marker level depends on the Y-axis unit of the assigned marker trace.
- A query error will occur if the specified advanced marker has not been assigned or is not a power spectral density marker.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|:NOISe[:STATe]

Function Sets the specified advanced marker to a power spectral density marker.
Also queries the status of the specified power spectral density marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|:NOISe[:STATe]<wsp>OFF|ON|0|1
:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|:NOISe[:STATe]?
[1|2|3|4]: Advanced marker number
```
Response 0 = Off, 1 = On

Example
```
:CALCULATE:AMARKER:FUNCTION:PDENSITY ON
:CALCULATE:AMARKER:FUNCTION:PDENSITY? -> 1
```

Description
- If a power spectral density marker is assigned, moving markers (:CALCulate:AMARker[1|2|3|4][:STATe]) and integrated power markers (:CALCulate:AMARker[1|2|3|4]:FUNCtion:INTegral[:STATe]) will be set to off.
- If this command with the parameter set to OFF is specified on an advanced marker that has been assigned to power spectral density marker, the advanced marker will change to a moving marker.
  If the advanced marker has not been assigned, using this command with the parameter set to OFF will leave the advanced marker unassigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:FUNCtion:PRESet

Function Changes the specified advanced marker to a moving marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:FUNCtion:PRESet
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:FUNCTION:PRESET
```

Description
- If an advanced marker is assigned to a power spectral density marker or integrated power marker, the marker can be changed directly to a moving marker.
  This does not change the marker position.
  If the advanced marker has not been assigned, using this command will leave the advanced marker unassigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MAXimum

Function Detects the peak and sets the specified advanced marker there.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:MAXimum
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:MAXIMUM
```

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MAXimum:LEFT

Function Detects the closest peak to the left of the current specified advanced marker position and sets the specified advanced marker there.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:MAXimum:LEFT
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:MAXIMUM:LEFT
```

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MAXimum:NEXT

Function Detects the highest peak whose level is less than or equal to that of the current specified advanced marker position and sets the specified advanced marker there.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:MAXimum:NEXT
[1|2|3|4]: Advanced marker number
```

Example
```
:CALCULATE:AMARKER:MAXIMUM:NEXT
```

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MAXimum:RIGHt

Function Detects the closest peak to the right of the current specified advanced marker position and sets the specified advanced marker there.

Syntax `:CALCulate:AMARker[1|2|3|4]:MAXimum:RIGHt`
`[1|2|3|4]: Advanced marker number`

Example `:CALCULATE:AMARKER:MAXIMUM:RIGHT`

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MINimum

Function Detects the bottom and sets the specified advanced marker there.

Syntax `:CALCulate:AMARker[1|2|3|4]:MINimum`
`[1|2|3|4]: Advanced marker number`

Example `:CALCULATE:AMARKER:MINIMUM`

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MINimum:LEFT

Function Detects the closest bottom to the left of the current specified advanced marker position and sets the specified advanced marker there.

Syntax `:CALCulate:AMARker[1|2|3|4]:MINimum:LEFT`
`[1|2|3|4]: Advanced marker number`

Example `:CALCULATE:AMARKER:MINIMUM:LEFT`

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MINimum:NEXT

Function Detects the lowest bottom whose level is greater than or equal to that of the current specified advanced marker position and sets the specified advanced marker there.

Syntax `:CALCulate:AMARker[1|2|3|4]:MINimum:NEXT`
`[1|2|3|4]: Advanced marker number`

Example `:CALCULATE:AMARKER:MINIMUM:NEXT`

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:MINimum:RIGHt

Function Detects the closest bottom to the right of the current specified advanced marker position and sets the specified advanced marker there.

Syntax `:CALCulate:AMARker[1|2|3|4]:MINimum:RIGHt`
`[1|2|3|4]: Advanced marker number`

Example `:CALCULATE:AMARKER:MINIMUM:RIGHT`

Description
- If the specified advanced marker has not been assigned, a moving marker will be assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4][:STATe]

Function Sets or queries whether the specified advanced marker is to be assigned.
Also queries the status of the specified moving marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]
[:STATe]<wsp>OFF|ON|0|1
:CALCulate:AMARker[1|2|3|4]
[:STATe]?
[1|2|3|4]: Advanced marker number
```
Response 0 = Off, 1 = On

Example
```
:CALCULATE:AMARKER ON
:CALCULATE:AMARKER? -> 1
```

Description
- If a moving marker is assigned, power spectral density markers (:CALCulate:AMARker[1|2|3|4]:FUNCtion:PDENsity|:NOISe[:STATe]) and integrated power markers (:CALCulate:AMARker[1|2|3|4]:FUNCtion:INTegral[:STATe]) will be set to off.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:TRACe

Function Sets or queries the trace that the specified advanced marker is assigned to.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:TRACe<wsp><trace name>
:CALCulate:AMARker[1|2|3|4]:TRACe?
[1|2|3|4]: Advanced marker number
```
<trace name> = Trace
```
TRA to TRG = Trace A to trace G
```

Example
```
:CALCULATE:AMARKER:TRACE TRA
:CALCULATE:AMARKER:TRACE? -> TRA
```

Description
- Setting and querying are possible even when the specified advanced marker has not been assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:X

Function Sets the specified advanced marker to the specified position.
Also queries the X value of the specified advanced marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:X<wsp><NRf>[M|HZ]
:CALCulate:AMARker[1|2|3|4]:X?
[1|2|3|4]: Advanced marker number
```
<NRf> = Advanced marker position

Response <NRf> = Advanced marker position

Example
```
:CALCULATE:AMARKER:X 1550.000nm
:CALCULATE:AMARKER:X? ->
+1.55000000E-006
```

Description
- The unit of the returned advanced marker X value depends on the :CALCulate:MARKer:UNIT setting.
- The unit of the returned advanced marker level depends on the :CALCulate:MARKer:UNIT setting.
- If this set command is used when the specified advanced marker has not been assigned, a moving marker will be assigned.
- If a query is made when the specified advanced marker has not been assigned, a query error will occur.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:AMARker[1|2|3|4]:Y?

Function Queries the Y value of the specified advanced marker.

Syntax
```
:CALCulate:AMARker[1|2|3|4]:Y?
[1|2|3|4]: Advanced marker number
```
Response

<NRf> = Advanced marker level

Example
```
:CALCULATE:AMARKER:X? ->
-1.00000000E+001
```

Description
- The unit of the returned advanced marker level depends on the Y-axis unit of the assigned marker trace.
- Even if the advanced marker is an integrated power marker or power spectral density marker, the moving marker level will be returned.
  The query does not return the integrated power value or the power spectral density value.
- A query error will occur if the specified advanced marker has not been assigned.
- If the advanced marker number is not specified, advanced marker 1 will be used.
- This is a sequential command.

:CALCulate:ARESolution?

Function Queries the actual resolution data of the specified trace.

Syntax
```
:CALCulate:ARESolution?<wsp><trace name>,[<start point>,<stop point>]
```

<trace name>	Target trace (TRA|TRB|TRC|TRD|TRE|TRF|TRG)
<start point>	Sample range to transfer (start point) (1 to 50001)
<stop point>	Sample range to transfer (stop point) (1 to 50001)

Example CALCULATE:ARESOLUTION?
-> +1.89759145E-009,+1.89744762E-009,+1.89730346E-009,....

Description
• The function outputs a wavelength value.
• If the <start point> and <stop point> parameters are omitted, the entire sample data of the specified trace will be output.
• The data is output in ASCII or BINARY format according to the :FORMat[:DATA] setting.
• This is a sequential command.

:CALCulate:CATegory

Function Sets/queries the type of analysis.

Syntax

```
:CALCulate:CATegory<wsp>{SWTHresh|
SWENvelope|SWRMs|SWPKrms|NOTCh|
DFBLd|FPLD|LED|SMSR|POWer|PMD|
WDM|NF|FILPk|FILBtm|WFPeak|WFBTm|
COLor|0|1|2|3|4|5|6|7|8|9|10|11|12|
13|14|15|16|17}
:CALCulate:CATegory?
```

SWTHresh|0	Spectrum width analysis (THRESH)
SWENvelope|1	Spectrum width analysis (ENVELOPE)
SWRMs|2	Spectrum width analysis (RMS)
SWPKrms|3	Spectrum width analysis (PEAK-RMS)
NOTCh|4	Notch width analysis
DFBLd|5	DFB-LD parameter analysis
FPLD|6	FP-LD parameter analysis
LED|7	LED parameter analysis
SMSR|8	SMSR analysis
POWer|9	Power analysis
PMD|10	PMD analysis
WDM|11	WDM analysis
NF|12	NF analysis
FILPk|13	Filter peak analysis
FILBtm|14	Filter bottom analysis
WFPeak|15	WDM FIL-PK analysis
WFBtm|16	WDM FIL-BTM analysis
COLor|17	COLOR analysis

Example

```
:CALCULATE:CATegory SWTHresh
:CALCULATE:CATegory? -> 0
```

Explanation

- Even when this command is executed, no analysis is performed unless the `CALCulate[:IMMediate]` command is executed.
- This is a sequential command.

:CALCulate:DATA?

Function Queries the analysis results.

Syntax `:CALCulate:DATA?`

Example `:CALCULATE:DATA?`

Explanation

- Queries the analysis results from the last time analysis was executed.
- If the analysis function has not been executed, a query error occurs.
- For a response example, see section 6.7, “Output Format of Analysis Results.”
- This is a sequential command.

:CALCulate:DATA:CGAin?

Function Queries the gain value of the EDFA-NF analysis results.

Syntax `:CALCulate:DATA:CGAin?`

Example

```
:CALCULATE:DATA:CGAin?
-> +1.00000000E+001,+1.00000000E+001
```

Explanation

- If the analysis function has not been executed, a query error occurs.
- “0” is returned if there is no relevant return value (such as if the analysis executed was not EDFA-NF analysis)
- The number of channels to be output can be acquired by the `:CALCulate:DATA:NCHannels?` command.
- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:CALCulate:DATA:CNF?

Function Queries the NF value of the EDFA-NF analysis results.

Syntax `:CALCulate:DATA:CNF?`

Example

```
:CALCULATE:DATA:CNF? ->
+1.00000000E+001,+1.00000000E+001
```

Explanation

- If `:CALCulate[:IMMediate]` has not been executed, a query error occurs.
- “0” is returned if there is no relevant return value (such as if the analysis executed was not EDFA-NF analysis)
- The number of channels to be output can be `acquired` by the `:CALCulate:DATA:NCHannels?` command.
- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:CALCulate:DATA:COLor?

Function Queries the dominant wavelength of the measured light source and the x coordinate, y coordinate, z coordinate, color temperature, and deviation of the measured light source on the xy chromaticity diagram.

Syntax `:CALCulate:DATA:COLor?`

Example
```
:CALCulate:DATA:COLor?
-> +0.58237440E-006,+4.30500000E-
001,+4.03000000E-001,+1.66500000E-
001,+3.10300000E+003,+0.0132000
0E+000
```

Explanation
- This command returns analysis results that are not returned from the ":CALCulate:DATA?" command, such as the OSNR value.
  The following items are returned. The items are indicated with symbols. For the meanings of the symbols, see section 6.7.
  <dominant wl>,<x col>,<y col>,<z col>,<color temp>,<dev>
- “0” is returned if there is no relevant return value.
- This is a sequential command.

:CALCulate:DATA:CPOWers?

Function Queries the level value of the WDM, EDFA-NF, WDM FIL-PK, or WDM FIL-BTM analysis results.

Syntax `:CALCulate:DATA:CPOWers?`

Example
```
:CALCULATE:DATA:CPOWERS? ->
+1.00000000E+001,+1.00000000E+001
```

Explanation
- If the analysis function has not been executed, a query error occurs.
- “0” is returned if there is no relevant return value.
- The number of channels to be output can be acquired by the `:CALCulate:DATA:NCHannels?` command.
- The value to be output depends on the analysis performed.

WDM :	LEVEL or MEAS LEVEL
EDFA-NF :	INPUT LEVEL
WDM FIL-PK :	PEAK LEVEL (output even if SW is OFF)
WDM FIL-BTM:	PEAK LEVEL (output even if SW is OFF)

- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:CALCulate:DATA:CSNR?

Function Queries the SNR value from the last time WDM analysis was executed.

Syntax `:CALCulate:DATA:CSNR?`

Example
```
:CALCULATE:DATA:CSNR? ->
 +4.00000000E+001,+4.00000000E+001
```

Explanation
- If the analysis function has not been executed, a query error occurs.
- “0” is returned if there is no relevant return value (for example, if analysis made is other than WDM analysis).
- The number of channels to be output can be acquired by the `:CALCulate:DATA:NCHannels?` command.
- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:CALCulate:DATA:CWAVelengths?

Function Queries the wavelength value of the WDM, EDFA-NF, WDM FIL-PK, or WDM FIL-BTM analysis results.

Syntax `:CALCulate:DATA:CWAVelengths?`

Example
```
:CALCULATE:DATA:CWAVELENGTHS? ->
+1.55000000E-006,+1.56000000E-006
```

Explanation
- If the analysis function has not been executed, a query error occurs.
- “0” is returned if there is no relevant return value.
- The number of channels to be output can be acquired by the `:CALCulate:DATA:NCHannels?` command.
- The value to be output depends on the analysis performed.

WDM :	WAVELENGTH or MEAS WL
EDFA-NF:	WAVELENGTH
WDM FIL-PK:	NOMINAL WAVELENGTH
WDM FIL-BTM:	NOMINAL WAVELENGTH

- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:CALCulate:DATA:DFBLd?

Function Queries the DFB-LD analysis results.

Syntax `:CALCulate:DATA:DFBLd?`

Example `:CALCULATE:DATA:DFBLD? ->`

Explanation
- If the :CALCulate[:IMMediate] command has not been executed, a query error occurs.
- "0" is returned if there is no relevant return value (for example, if the analysis that was executed was not a DFB-LD analysis).
- This command returns analysis results that are not returned from the ":CALCulate:DATA?" command, such as the OSNR value. The following items are returned. The items are listed here as abbreviations. For the meaning of these abbreviations, see section 6.7.
  <peak wl>,<peak lvl>,<center wl>,<spec wd>,<smsr(L)>,<smsr(R)>,<mode ofst(L)>,<mode ofst(R)>,<snr>,<power>,<rms>,<Krms>
- This is a sequential command.

:CALCulate:DATA:NCHannels?

Function Queries the number of channels of the WDM, EDFA-NF, WDM FIL-PK, or WDM FIL-BTM analysis results.

Syntax `:CALCulate:DATA:NCHannels?`

Example `:CALCULATE:DATA:NCHANNELS? -> 16`

Explanation
- If the analysis function has not been executed, a query error occurs.
- "0" is returned if there is no relevant return value.
- The value is output as ASCII data, regardless of the setting of `FORMat[:DATA]`.
- This is a sequential command.

:CALCulate:DATA:OSLope?

Function Queries the OUTPUT SLOPE value of the WDM analysis results.

Syntax `:CALCulate:DATA:OSLope?`

Response <NRf> = Output slope value [dB/nm] or [dB/THz]

Example `:CALCULATE:DATA:OSLOPE? -> +2.45352623E-001`

Explanation
- A query error will occur if the analysis function is not implemented.
- "0" is returned if there is no relevant return value (for example, if the analysis that was executed was not a WDM analysis).
- Analysis results can be queried even if the output of the OUTPUT SLOPE value is set to OFF.
- ASCII data is returned regardless of the setting specified by the :FORMat:[DATA] command.
- This is a sequential command.

:CALCulate:DISPlay

Function Sets/queries the display format of analysis results.

Syntax `:CALCulate:DISPlay<wsp>0|1|2|3|4`
`:CALCulate:DISPlay?`

0: TRACE&TABLE
1: TABLE
2: TRACE
3: GRAPH&TABLE
4: GRAPH

Example `:CALCULATE:DISPLAY 1`
`:CALCULATE:DISPLAY? -> 1`

Explanation
- This is a sequential command.

:CALCulate:DISPlay:GRAPh:LMARKer:Y

Function Sets/queries the position of line marker Y1 or Y2 on the graph display of analysis results.

Syntax `:CALCulate:DISPlay:GRAPh:LMARKer:Y<wsp>1|2,<NRf>[DB]`
`:CALCulate:DISPlay:GRAPh:LMARKer:Y?<wsp>1|2`

1: Line marker Y1. 2: Line marker Y2.
<NRf>: Line marker position

Example `:CALCULATE:DISPLAY:GRAPH:LMARKER:Y 1,3.4`
`:CALCULATE:DISPLAY:GRAPH:LMARKER:Y? 1 -> +3.40000000E+000`

Explanation
- This command is valid when the EDFA-NF analysis results are being displayed on a graph.
- This is a sequential command.

:CALCulate[:IMMediate]

Function Executes analysis. Queries the result of whether analysis has been performed.

Syntax `:CALCulate[:IMMediate]`
`:CALCulate[:IMMediate]?`
0: Not performed
1: Performed

Example `:CALCULATE`
`:CALCULATE? -> 1`

Explanation
- Analysis is performed according to the latest analysis settings.
- Analysis is performed on the following occasions:
- When `CALCulate[:IMMediate]` command is executed.
- When `CALCulatePARameter`: command is executed, or parameter settings changed
- This is a sequential command.

:CALCulate[:IMMediate]:AUTO

Function Sets/queries the automatic analysis function.

Syntax `:CALCulate[:IMMediate]:AUTO<wsp>OFF|ON|0|1`
`:CALCulate[:IMMediate]:AUTO?`
0: OFF
1: ON

Example `:CALCULATE:AUTO ON`
`:CALCULATE AUTO? -> 1`

Explanation
- When the automatic analysis function is ON, automatically activates an analysis function that is active after a sweep has ended.
- This is a sequential command.

:CALCulate:LMARker:AOFF

Function Clears all line markers.

Syntax `:CALCulate:LMARker:AOFFExample`
`:CALCULATE:LMARKER:AOFF`

Explanation This is a sequential command.

:CALCulate:LMARker:SRANge

Function Sets/queries whether to limit an analytical range to the spacing between line markers L1 and L2.

Syntax `:CALCulate:LMARker:SRANge<wsp>OFF|ON|0|1`
`:CALCulate:LMARker:SRANge?`
0: OFF
1: ON

Example `:CALCULATE:LMARKER:SRANGE ON`
`:CALCULATE:LMARKER:SRANGE? -> 1`

Explanation This is a sequential command.

:CALCulate:LMARker:SSPan

Function Sets spacing between line markers L1 and L2 for span.

Syntax `:CALCulate:LMARker:SSPan`

Example `:CALCULATE:LMARKER:SSPAN`

Explanation This is a sequential command.

:CALCulate:LMARker:SZSPan

Function Sets spacing between line markers L1 and L2 for zoom span.

Syntax `:CALCulate:LMARker:SZSPan`

Example `:CALCULATE:LMARKER:SZSPAN`

Explanation This is a sequential command.

:CALCulate:LMARker:X

Function Sets/queries the position of line markers L1 and L2.

Syntax `:CALCulate:LMARker:X<wsp>1|2,<NRf>[M|HZ]`
`:CALCulate:LMARker:X?<wsp>1|2`
1, 2 = Line marker numbers
<NRf> = Position of a line marker
Response <NRf>[m|Hz]

Example `:CALCULATE:LMARKER:X 1,1550.000nm`
`:CALCULATE:LMARKER:X? 1 -> +1.55000000E-006`

Explanation
- If the specified line marker is not located, a query error occurs.
- This is a sequential command.

:CALCulate:LMARker:Y

Function Sets/queries the position of line markers L3 and L4.

Syntax `:CALCulate:LMARker:Y<wsp>3|4,<NRf>[DBM|DB|%]`
`:CALCulate:LMARker:Y?<wsp>3|4`
3, 4 = Line marker numbers
<NRf> = Position of a line marker

Example `:CALCULATE:LMARKER:y 3,-10dBm`
`:CALCULATE:LMARKER:y? 3 -> -1.00000000E+001`

Explanation
- If the specified line marker is not located, a query error occurs.
- This is a sequential command.

:CALCulate:MARKer:AOFF

Function Clears all markers.

Syntax `:CALCulate:MARKer:AOFF`

Example `:CALCULATE:MARKER:AOFF`

Explanation This is a sequential command.

:CALCulate:MARKer:AUTO

Function Sets/queries the auto search function.

Syntax
```
:CALCulate:MARKer:AUTO<wsp>OFF|ON|0|1
:CALCulate:MARKer:AUTO?
```
0 = OFF
1 = ON

Example
```
:CALCULATE:MARKER:AUTO  ON
:CALCULATE:MARKER:AUTO? -> 1
```

Explanation
- When the auto search function is ON, this instrument automatically performs a peak/bottom search through an active trace after a sweep has ended.
- This is a sequential command.

:CALCulate:MARKer:FUNCtion:FORMat

Function Sets the format of a difference value displayed in the area marker and queries the format set.

Syntax
```
:CALCulate:MARKer:FUNCtion:FORMat<wsp>OFFSet|SPACing|0|1
:CALCulate:MARKer:FUNCtion:FORMat?
```
OFFSet = Displays the difference of each marker relative to the moving marker.
SPACing = Displays the difference of each marker relative to a neighboring marker.
Response 0 = OFFSet, 1 = SPACing

Example
```
:CALCULATE:MARKER:FUNCTION:FORMAT SPACING
:CALCULATE:MARKER:FUNCTION:FORMAT?-> 1
```

Explanation This is a sequential command.

:CALCulate:MARKer:FUNCtion:UPDate

Function Sets/queries ON/OFF of the automatic update function of fixed markers used when updating an active trace.

Syntax
```
:CALCulate:MARKer:FUNCtion:UPDate<wsp>OFF|ON|0|1
:CALCulate:MARKer:FUNCtion:UPDate?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:MARKER:FUNCTION:UPDATE ON
:CALCULATE:MARKER:FUNCTION:UPDATE? -> 1
```

Explanation
- When the automatic update function is ON and the active trace is updated, the level positions of fixed markers automatically follow the waveform.
- This is a sequential command.

:CALCulate:MARKer:MAXimum

Function Detects a peak and places the moving marker on that peak.

Syntax
```
:CALCulate:MARKer:MAXimum
```

Example
```
:CALCULATE:MARKER:MAXIMUM
```

Explanation This is a sequential command.

:CALCulate:MARKer:MAXimum:LEFT

Function Detects the nearest peak existing on the left side of the current position of the moving marker and places the moving marker on that peak.

Syntax
```
:CALCulate:MARKer:MAXimum:LEFT
```

Example
```
:CALCULATE:MARKER:MAXIMUM:LEFT
```

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:MAXimum:NEXT

Function Detects the highest peak that is below the level of the current position of the moving marker and places the moving marker on that peak.

Syntax
```
:CALCulate:MARKer:MAXimum:NEXT
```

Example
```
:CALCULATE:MARKER:MAXIMUM:NEXT
```

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:MAXimum:RIGHt

Function Detects the nearest peak existing on the right side of the current position of the moving marker and places the moving marker on that peak.

Syntax
```
:CALCulate:MARKer:MAXimum:RIGHt
```

Example
```
:CALCULATE:MARKER:MAXIMUM:RIGHT
```

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:MAXimum:SCENter

Function Detects the peak wavelength and sets it as the measurement center waveform.

Syntax
```
:CALCulate:MARKer:MAXimum:SCENter
```

Example
```
:CALCULATE:MARKER:MAXIMUM:SCENTER
```

Explanation This is a sequential command.

:CALCulate:MARKer:MAXimum:SCENter:AUTO

Function Sets/queries ON/OFF of the function to automatically detect the peak wavelength and set it as the measurement center wavelength.

Syntax
```
:CALCulate:MARKer:MAXimum:SCENter:AUTO<wsp>OFF|ON|0|1
:CALCulate:MARKer:MAXimum:SCENter:AUTO?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:MARKER:MAXIMUM:SCENTER:AUTO ON
:CALCULATE:MARKER:MAXIMUM:SCENTER:AUTO? -> 1
```

Explanation
- When this function is ON, this instrument automatically detects the peak wavelength of an active trace wavelength each time a sweep has ended, and sets it as the measurement center wavelength.
- This is a sequential command.

:CALCulate:MARKer:MAXimum:SRLevel

Function Detects the peak level and sets it for the reference level.

Syntax `:CALCulate:MARKer:MAXimum:SRLevel`

Example `:CALCULATE:MARKER:MAXIMUM:SRLEVEL`

Explanation This is a sequential command.

:CALCulate:MARKer:MAXimum:SRLevel:AUTO

Function Sets/queries ON/OFF of the function to automatically detect the peak level and sets it as the reference level.

Syntax `:CALCulate:MARKer:MAXimum:SRLevel:AUTO<wsp>OFF|ON|0|1`

Response 0 = OFF, 1 = ON

Example `:CALCULATE:MARKER:MAXIMUM:SRLEVEL:AUTO ON`
`CALCULATE:MARKER:MAXIMUM:SRLEVEL:AUTO? -> 1`

Explanation
- When this function is ON, the instrument automatically detects the peak level of an active trace wavelength each time a sweep has ended, and sets it as the reference level.
- This is a sequential command.

:CALCulate:MARKer:MAXimum:SZCenter

Function Detects the peak wavelength and sets it as the display center wavelength.

Syntax `:CALCulate:MARKer:MAXimum:SZCenter`

Example `:CALCULATE:MARKER:MAXIMUM:SZCENTER`

Explanation This is a sequential command.

:CALCulate:MARKer:MINimum

Function Detects the bottom and places the moving marker on that bottom.

Syntax `:CALCulate:MARKer:MINimum`

Example `:CALCULATE:MARKER:MINIMUM`

Explanation This is a sequential command.

:CALCulate:MARKer:MINimum:LEFT

Function Detects the nearest bottom existing on the left side of the current position of the moving marker and places the moving marker on that bottom.

Syntax `:CALCulate:MARKer:MINimum:LEFT`

Example `:CALCULATE:MARKER:MINIMUM:LEFT`

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:MINimum:NEXT

Function Detects the lowest bottom that is above the level of the current position of the moving marker and places the moving marker on that bottom.

Syntax `:CALCulate:MARKer:MINimum:NEXT`

Example `:CALCULATE:MARKER:MINIMUM:NEXT`

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:MINimum:RIGHt

Function Detects the nearest bottom existing on the right side of the current position of the moving marker and places the moving marker on that side.

Syntax `:CALCulate:MARKer:MINimum:RIGHt`

Example `:CALCULATE:MARKER:MINIMUM:RIGHT`

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:MSEarch

Function Sets/queries the type of the search function.

Syntax `:CALCulate:MARKer:MSEarch<wsp>OFF|ON|0|1`
`:CALCulate:MARKer:MSEarch?`

OFF|0: Sets the search function to single search.

ON|1: Sets the search function to multi search.

Response 0 = OFF, 1 = ON

Example `:CALCULATE:MARKER:MSEARCH on`
`:CALCULATE:MARKER:MSEARCH? -> 1`

Explanation
- The search is executed as soon as you set the search function.
- This is a sequential command.

:CALCulate:MARKer:MSEarch:SORT

Function Sets/queries the sort order of the multi search detection list.

Syntax `:CALCulate:MARKer:MSEarch:SORT<wsp>WAVelength|LEVel|0|1`
`:CALCulate:MARKer:MSEarch:SORT?`

WAVelength|0: Wavelengths are displayed in order starting from the shortest wavelength.

LEVel|1: For the peak search, levels are displayed in order starting from the highest level. For the bottom search, levels are displayed in order starting from the lowest level.

Response 0 = OFF, 1 = ON

Example `:CALCULATE:MARKER:MSEARCH:SORT WAV`
`:CALCULATE:MARKER:MSEARCH:SORT? -> 0`

Explanation
- This is a sequential command.

:CALCulate:MARKer:MSEarch:THResh

Function Sets/queries the multi search threshold.

Syntax
```
:CALCulate:MARKer:MSEarch:THResh<wsp>
<NRf>[DB]
:CALCulate:MARKer:MSEarch:THResh?
```
<NRf>: Threshold (dB)

Example
```
:CALCULATE:MARKER:MSEARCH:THRESH 50DB
:CALCULATE:MARKER:MSEARCH? ->
+5.00000000E+001
```

Explanation • This is a sequential command.

:CALCulate:MARKer:SCENter

Function Sets the wavelength of the current moving marker as the measurement center waveform.

Syntax `:CALCulate:MARKer:SCENter`

Example `:CALCULATE:MARKER:MINIMUM:SCENTER`

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:SRLevel

Function Sets the current level of the moving marker for the reference level.

Syntax `:CALCulate:MARKer:SRLevel`

Example `:CALCULATE:MARKER:MINIMUM:SRLEVEL`

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer[:STATe]

Function Specified marker is positioned or deleted in the position of the moving marker. Also, queries the status of the specified marker.

Syntax
```
:CALCulate:MARKer[:STATe]<wsp>
<marker>,OFF|ON|0|1:CALCulate:MARKer[:STATe]?<wsp><marker>
```
<marker>: Marker number (0: moving marker)

Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:MARKER:STATE 1,ON
:CALCULATE:MARKER:STATE 1 -> 1
```

Explanation
- When the moving marker is not active and an attempt is made to set a fixed marker, an execution error occurs.
- If moving marker is specified, it is placed in the center of measurment display.
- This is a sequential command.

:CALCulate:MARKer:SZCenter

Function Sets the current wavelength of the moving marker for the display center wavelength.

Syntax `:CALCulate:MARKer:SZCenter`

Example `:CALCULATE:MARKER:SZCENTER`

Explanation
- If the moving marker is OFF, an execution error occurs.
- This is a sequential command.

:CALCulate:MARKer:UNIT

Function Sets/queries the units of display for the marker values.

Syntax
```
:CALCulate:MARKer:UNIT<wsp>WAVelength|FREQuency|0|1
:CALCulate:MARKer:UNIT?
```
WAVelength|0

FREQuency|1

Response 0=WAVelength, 1= FREQuency

Example
```
:CALCULATE:MARKER:UNIT FREQUENCY
:CALCULATE:MARKER:UNIT? -> 1
```

Explanation • This is a sequential command.

:CALCulate:MARKer:X

Function Places a specified marker in a specified position. Queries the X value of the specified marker.

Syntax
```
:CALCulate:MARKer:X<wsp><marker>,<NRf>[M|HZ]
:CALCulate:MARKer:X?<wsp><marker>|ALL
```
<marker> = Marker number (0: moveing marker)

ALL : All assigned markers

<NRf>= Marker position

Response

<NRf>`[m|Hz]`

If <marker> is specified

<integer>, <NRf>, <NRf>, ..., <NRf>

(If ALL is specified)

Example
```
:CALCULATE:MARKER:X 0,1550.000nm
:CALCULATE:MARKER:X? 0 ->
+1.55000000E-006
```

Explanation
- If an already located marker is specified, that marker will be moved to a specified position.
- If the specified marker is not located, a query error occurs.
- If ALL is specified (e.g., :CALC:MARK:Y? ALL), the Y values of all assigned markers will be returned.
  The number of assigned markers will be returned as an integer, and then all the marker values will follow.
- This is a sequential command.

:CALCulate:MARKer:Y?

Function Queries the Y value of the specified marker.

Syntax `:CALCulate:MARKer:Y?<wsp><marker>|ALL`

<marker> : Marker number (0: moveing marker)

ALL = All assigned markers

If <marker> is specified

<NRf>= Marker level

If ALL is specified

<integer>, <NRf>, <NRf>, ..., <NRf>

Example `:CALCULATE:MARKER:Y? 0 -> -1.00000000E+001`

Explanation
- This unit of the marker level to be queried is dependent on the Y-axis unit of the active trace.
- If the specified marker is not located, a query error occurs.
- If ALL is specified (e.g., :CALC:MARK:Y? ALL), the Y values of all assigned markers will be returned.
- This is a sequential command.

:CALCulate:MATH:TRC

Function Sets/queries the TRACE C calculation function.

Syntax
```
:CALCulate:MATH:TRC<wsp>A-B(LOG)|
B-A(LOG)|A+B(LOG)|A+B(LIN)|
A-B(LIN)|B-A(LIN)|1-K(A/B)|
1-K(B/A)|
:CALCulate:MATH:TRC?
```

Example
```
:CALCULATE:MATH:TRC A-B(LOG)
:CALCULATE:MATH:TRC? -> A-B(LOG)
```

Explanation
- When the calculation function of trace C is set using this command, the attribute of trace C automatically becomes attribute “CALC”.
- If trace C is not a calculation trace, “NONE” is returned.
- This is a sequential command.

:CALCulate:MATH:TRC:K

Function Sets/queries parameter K of the TRACE C calculation function.

Syntax
```
:CALCulate:MATH:TRC:K<wsp><NRf>
:CALCulate:MATH:TRC:K?
```

<NRf> = Parameter K

Example
```
:CALCULATE:MATH:TRC:K 0.1
:CALCULATE:MATH:TRC:K? ->
+1.00000000E-001
```

Explanation This is a sequential command.

:CALCulate:MATH:TRF

Function Sets/queries the TRACE F calculation function.

Syntax
```
:CALCulate:MATH:TRF<wsp>C-D(LOG)|
D-C(LOG)|C+D(LOG)|D-E(LOG)|
E-D(LOG)|D+E(LOG)|C+D(LIN)|
C-D(LIN)|D-C(LIN)|D+E(LIN)|
D-E(LIN)|E-D(LIN)|PWRNBWA|PWRNBWB|
PWRNBWC|PWRNBWD|PWRNBWE
:CALCulate:MATH:TRF?
```

Example
```
:CALCULATE:MATH:TRF C-D(LOG)
:CALCULATE:MATH:TRF? -> C-D(LOG)
```

Explanation
- When the calculation function of trace F is set using this command, the attribute of trace F automatically becomes attribute “CALC”.
- If trace F is not a calculation trace, “NONE” is returned.

  Example `calc:math:trf c-d(log)`
  `calc:math:trf? -> C-D(LOG)`
- This is a sequential command.

:CALCulate:MATH:TRF:PNBW:BWIDth|BANDwidth

Function Sets/queries the normalization bandwidth of the power spectral density trace.

Syntax
```
:CALCulate:MATH:TRF:PNBW:BWIDth|
BANDwidth<wsp><NRf>[m]
:CALCulate:MATH:TRF:PNBW:BWIDth|
BANDwidth?
```

<NRf>=Normalization bandwidth[mm]

Example
```
:CALCULATE:MATH:TRF:PNBW:BAND 0.1nm
:CALCULATE:MATH:TRF:PNBW:BAND? ->
1.00000000E-010
```

Explanation
- This is a sequential command.

:CALCulate:MATH:TRG

Function Sets/queries the TRACE G calculation function.

Syntax
```
:CALCulate:MATH:TRG<wsp>C-F(LOG)|
F-C(LOG)|C+F(LOG)|E-F(LOG)|
F-E(LOG)|E+F(LOG)|C+F(LIN)|
C-F(LIN)|F-C(LIN)|E+F(LIN)|
E-F(LIN)|FLIN)|NORMA|NORMB|NORMC|
CVFTA|CVFTB|CVFTC|MKRFT|PKCVFTA|
PKCVFTB|PKCVFTC
:CALCulate:MATH:TRG?
```

Example
```
:CALCULATE:MATH:TRG C-F(LOG)
:CALCULATE:MATH:TRG? -> C-F(LOG)
```

Explanation
- When the calculation function of trace G is set using this command, the attribute of trace G automatically becomes attribute “CALC”.
- If trace G is not a calculation trace, “NONE” is returned.
- This is a sequential command.

:CALCulate:MATH:TRG:CVFT:FALGo

Function Sets/queries the fitting curve function of the TRACE G fitting curve function.

Syntax
```
:CALCulate:MATH:TRG:CVFT:FALGo
<wsp><algorhythm>
:CALCulate:MATH:TRG:CVFT:FALGo?
```
<algorhythm>
GAUSs = GAUSS
LORENz = LORENZ
3RD = 3RD POLY
4TH = 4TH POLY
5TH = 5TH POLY

Response
0 = GAUSS 1 = LORENZ,
2 = 3RD POLY 3 = 4TH POLY
4 = 5TH POLY

Example
```
:CALCULATE:MATH:TRG:CVFT:
FALGO GAUSS
:CALCULATE:MATH:TRG:CVFT:FALG? -> 1
```

Explanation
- Setting of calculation area is common to curve fit and peak curve fit.
- This is a sequential command.

:CALCulate:MATH:TRG:CVFT:OPARea

Function Sets/queries a calculation area during curve fit and peak curve fit.

Syntax
```
:CALCulate:MATH:TRG:CVFT:OPARea
<wsp>ALL|INL1-L2|OUTL1-L2|0|1|2
:CALCulate:MATH:TRG:CVFT:OPARea?
```
ALL = all of the set wavelength range
INL1-L2 = range surrounding line marker 1 and 2
OUTL1-L2 = range outisde line markers 1 and 2
Response 0 = ALL, 1 = INL1-L2, 2 = OUTL1-L2

Example
```
:CALCULATE:MATH:TRG:CVFT:
OPAREA inl1-l2
:CALCULATE:MATH:TRG:CVFT:OPAREA?->
1
```

Explanation
- Setting of calculation area is common to curve fit and peak curve fit.
- This is a sequential command.

:CALCulate:MATH:TRG:CVFT:THResh

Function Sets/queries the threshold value for curve fitting.

Syntax
```
:CALCulate:MATH:TRG:CVFT:THResh
<wsp><integer>[DB]
:CALCulate:MATH:TRG:CVFT:THResh?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:MATH:TRG:CVFT:THRESH
10db
:CALCULATE:MATH:TRG:CVFT:THRESH?->
10
```

Explanation This is a sequential command.

:CALCulate:MATH:TRG:PCVFt:THResh

Function Sets/queries the threshold value for peak curve fitting.

Syntax
```
:CALCulate:MATH:TRG:PCVFt:THResh
<wsp><integer>[DB]
:CALCulate:MATH:TRG:PCVFt:THResh?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:MATH:TRG:PCVFT:
THRESH 10db
:CALCULATE:MATH:TRG:PCVFT:THRESH?->
10
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:DFBLd

Function Sets/queries parameters for the DFB-LD analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:DFB
Ld<wsp><item>,<paramater>,<data>
:CALCulate:PARameter[:CATegory]:DFB
Ld?<wsp><item>,<paramater>
```
<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Setting data

<item>	<parameter>	<data>
SWIDth	ALGO	ENVelope|THResh|RMS|PKRMs
	TH	<NRf>[DB]
	TH2	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
SMSR	SMODe	SMSR1|SMSR2|SMSR3|SMSR4
	SMASk	<NRf>[M]
	MDIFf	<NRf>[DB]
RMS	ALGO	RMS|PKRMs
	TH	<NRf>[DB]
	K	<NRf>
	MDIFf	<NRf> [DB]
POWer	SPAN	<NRf>[M]
OSNR	MDIFf	<NRf>[DB]
	NALGo	AFIX|MFIX|ACENter|MCENter|PIT|0|1|2|3|4
	NARea	<NRf>[M]
	MARea	<NRf>[M]
	FALGo	LINear|GAUSs|LORenz|3RD|4TH|5TH|0|1|2|3|4|5
	NBW	<NRf>[M]
	SPOWer	PEAK|INTegral|0|1
	IRANge	<NRf>

Example
```
:CALCULATE:PARAMETER:
DFBLD SWIDTH,ALGO,THRESH
:CALCULATE:PARAMETER:DFBLD? SWIDTH,
ALGO -> THR
:CALCULATE:PARAMETER:DFBLD
SMSR,SMASK,0.5NM
:CALCULATE:PARAMETER:DFBLD?
SMSR,SMASK -> +5.00000000E-010
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs. (such as combinations of SWIDth and SMODe)
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:FILBtm

Function Sets/queries parameters for the FILTER-BTM analysis function.

Syntax

```
:CALCulate:PARameter[:CATegory]:FIL
Btm<wsp><item>,<paramater>,
<data>
:CALCulate:PARameter[:CATegory]:FIL
Btm?<wsp><item>,<paramater>
```

<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Data to be set

<item>	<parameter>	<data>
BLEVel	SW	OFF|ON|0|1
BWAVelength	SW	OFF|ON|0|1
CWAVelength	SW	OFF|ON|0|1
	ALGO	PEAK|BOTTom
	TH	<NRf>[DB]
	MDIFf	<NRf>[DB]
NWIDth	SW	OFF|ON|0|1
	ALGO	PEAK|BOTTom
	TH	<NRf>[DB]
	MDIFf	<NRf>[DB]
XTALk	SW	OFF|ON|0|1
	ALGO	PEAK|BOTTom|BLEVel|GRID
	TH	<NRf>[DB]
	MDIFf	<NRf>[DB]
	CSPace	<NRf>[M]
	SARea	<NRf>[M]

Example

```
:CALCULATE:PARAMETER:FILBTM
CWAVELENGTH,ALGO,BOTTOM
:CALCULATE:PARAMETER:FILBTM
CWAVELENGTH,ALGO -> BOTT
:CALCULATE:PARAMETER:FILBTM
XTALK,CSPACE,0.2NM
:CALCULATE:PARAMETER:FILBTM?
XTALK,CSPACEe -> +2.00000000E-010
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs (a combination of CWAVelength and SARea, etc.).
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:FILPk

Function Sets/queries parameters for the FILTER PEAK analysis function.

Syntax

```
:CALCulate:PARameter[:CATegory]:FIL
Pk<wsp><item>,<paramater>,<data>
:CALCulate:PARameter[:CATegory]:FIL
Pk?<wsp><item>,<paramater>
```

<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Data to be set

<item>	<parameter>	<data>
PLEVel	SW	OFF|ON|0|1
PWAVelength	SW	OFF|ON|0|1
MWAVelength	SW	OFF|ON|0|1
	ALGO	THResh|RMS
	TH	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
SWIDth	SW	OFF|ON|0|1
	ALGO	THResh|RMS
	TH	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
XTALk	SW	OFF|ON|0|1
	ALGO	THResh|PLEVel|GRID
	TH	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
	CSPace	<NRf>[M]
	SARea	<NRf>[M]
RWIDth	SW	OFF|ON|0|1
	TH	<NRf>[DB]
	MDIFf	<NRf>[DB

Example

```
:CALCULATE:PARAMETER:FILPK
SWIDTH,ALGO,THRESH
:CALCULATE:PARAMETER:FILPK?
SWIDTH,ALGO -> THR
:CALCULATE:PARAMETER:FILPK XTALK,
CSPACE,0.5NM :CALCULATE:PARAMETER:
FILPK? XTALK,CSPACE ->
+5.00000000E-010
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs (a combination of SWIDth and CSPace, etc.).
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:FPLD

Function Sets/queries parameters for the FP-LD analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:FPLD<wsp><item>,<paramater>,<data>
:CALCulate:PARameter[:CATegory]:FPLD?<wsp><item>,<paramater>
```
<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Setting data

<item>	<parameter>	<data>
SWIDth	ALGO	ENVelope|THResh|RMS|PKRMs
	TH	<NRf>[DB]
	TH2	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
MWAVelength	ALGO	ENVelope|THResh|RMS|PKRMs
	TH	<NRf>[DB]
	TH2	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
TPOWer	OFFSet	<NRf>[DB]
MNUMber	ALGO	ENVelope|THResh|RMS|PKRMs
	TH	<NRf>[DB]
	TH2	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|11
	MDIFf	<NRf>[DB]

Example
```
:CALCULATE:PARAMETER:FPLD
SWIDTH,ALGO,THRESH
:CALCULATE:PARAMETER:FPLD?
SWIDTH,ALGO -> THR
:CALCULATE:PARAMETER:FPLD TPOWER,
OFFSET,1.0DB :CALCULATE:PARAMETER:
FPLD? TPOWER,OFFSET ->
+1.00000000E+000
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs. (a combination of SWIDth and OFFSET, etc.)
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:LED

Function Sets/queries parameters for the LED analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:LED<wsp><item>,<paramater>,<data>
:CALCulate:PARameter[:CATegory]:LED?<wsp><item>,<paramater>
```
<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Setting data

<item>	<parameter>	<data>
SWIDth	ALGO	ENVelope|THResh|RMS|PKRMs
	TH	<NRf>[DB]
	TH2	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
MWAVelength	ALGO	ENVelope|THResh|RMS|PKRMs
	TH	<NRf>[DB]
	TH2	<NRf>[DB]
	K	<NRf>
	MFIT	OFF|ON|0|1
	MDIFf	<NRf>[DB]
TPOWer	OFFSet	<NRf>[DB]

Example
```
:CALCULATE:PARAMETER:LED
SWIDTH,ALGO,THRESHh
:CALCULATE:PARAMETER:LED?
SWIDTH,ALGO -> THR
:CALCULATE:PARAMETER:LED TPOWER,
OFFSET,1.0DB :CALCULATE:PARAMETER:
LED? TPOWER,OFFSET ->
+1.00000000E+000
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs (a combination of SWIDth and OFFSet, etc.).
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:AALGo

Function Sets/queries the measurement algorithm applied to ASE level measurements made by the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:
AALGo<wsp><algorhythm>
:CALCulate:PARameter[:CATegory]:NF:
AALGo?
```
<algorhythm> = Measurement algorithm
AFIX: AUTO FIX
MFIX: MANUAL FIX
ACENter: AUTO CENTER
MCENter: MANUAL CENTER

Response 0 = AUTO FIX
1 = MANUAL FIX
2 = AUTO CENTER
3 = MANUAL CENTER

Example
```
:CALCULATE:PARAMETER:NF:AALGO MFIX
:CALCULATE:PARAMETER:NF:AALGO? -> 1
```
Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:FALGo

Function Sets/queries the fitting function during level measurement applied to ASE level measurements made by the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:
FALGo<wsp><algorhythm>
:CALCulate:PARameter[:CATegory]:NF:
FALGo?
```
<algorhythm> = Fitting function
LINear: LINEAR
GAUSs: GAUSS
LORenz: LORENZ
3RD: 3RD POLY
4TH: 4YH POLY
5TH: 5TH POLY

Response 0 =LINEAR
1 = GAUSS
2 = LORENZ
3 = 3RD POLY
4 = 4YH POLY
5 = 5TH POLY

Example
```
:CALCULATE:PARAMETER:NF:FALGO GAUSS
:CALCULATE:PARAMETER:NF:FALGO? -> 1
```
Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:FARea

Function Sets/queries the fitting range for level measurement applied to ASE level measurements made by the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:
FARea<wsp><NRf>[M]
:CALCulate:PARameter[:CATegory]:NF:
FARea?
```
<NRf>= fitting range [m]

Example
```
:CALCULATE:PARAMETER:NF:
FAREA 0.80NM
:CALCULATE:PARAMETER:NF:FAREA? ->
+8.00000000E-10
```
Explanation
- When the fitting range is set to “Between CH” (and ASE measurement algorithm is set to “AUTO-CTR” or “MANUAL-CTR”), then the command returns 0.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:IOFFset

Function Sets/queries level offset values (signal light) for the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:
IOFFset<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:NF:
IOFFset?
```
<NRf> = Level offset value of signal light [dB]

Example
```
:CALCULATE:PARAMETER:NF:
IOFFSET 10.00
:CALCULATE:PARAMETER:NF:IOFFSET? ->
+1.00000000E+001
```
Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:IRANge

Function Sets or queries the integration frequency range for when the EDFA-NF analysis feature calculates the signal optical power.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:
IRANge<wsp><NRf>
:CALCulate:PARameter[:CATegory]:NF:
IRANge?
```
<NRf> = Integration range [GHz]

Example
```
:CALCulate:PARameter:NF:IRANGE 40
:CALCulate:PARameter:NF:IRANGE?
-> +4.00000000E+001
```
Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:MARea

Function Sets/queries the mask range for level measurement applied to ASE level measurements made by the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:MARea<wsp><NRf>[M]
:CALCulate:PARameter[:CATegory]:NF:MARea?
```
<NRf> = mask range [m]

Example
```
:CALCULATE:PARAMETER:NF:MAREA 0.40NM
:CALCULATE:PARAMETER:NF:MAREA? ->
+4.00000000E-10
```

Explanation • When the mask range is set to “---” (and ASE level measurement function is set to “LINEAR”), the command returns 0.

• This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:MDIFf

Function Sets/queries the peak bottom difference of channel detection for the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:MDIFf<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:NF:MDIFf?
```
<NRf> = Peak bottom difference [dB]

Example
```
:CALCULATE:PARAMETER:NF:MDIFF 3.00DB
:CALCULATE:PARAMETER:NF:MDIFF? ->
+3.00000000E+000
```

Explanation • This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:OOFFset

Function Sets/queries level offset values (output light) for the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:OOFFset<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:NF:OOFFset?
```
<NRf> = Level offset value of output light [dB]

Example
```
:CALCULATE:PARAMETER:NF:OOFFSET 10.00
:CALCULATE:PARAMETER:NF:OOFFSET? ->
+1.00000000E+001
```

Explanation • This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:PDISplay

Function Sets/queries whether to display data used for fitting of the NF analysis function on the waveform screen.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:PDISplay<wsp>OFF|ON|0|1
:CALCulate:PARameter[:CATegory]:NF:PDISplay?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:PARAMETER:NF:PDISPLAY ON
:CALCULATE:PARAMETER:NF:PDISPLAY?->
1
```

Explanation • When this set value is 1 (ON), data used for fitting is displayed on the waveform screen.

• This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:TH

Function Sets/queries the threshold level of channel detection for the NF analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:NF:TH?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:PARAMETER:NF:TH 20.00DB
:CALCULATE:PARAMETER:NF:TH->
+2.00000000E+001
```

Explanation • This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:RBWidth

Function Sets/queries the method for calculating the resolution value of the NF computation.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:RBWidth<wsp>MEASured|CAL|0|1
:CALCulate:PARameter[:CATegory]:NF:RBWidth?
```

MEASured | 0	Use the value determined from the waveform using THRESH 3dB analysis.
CAL | 1	Use the actual resolution value stored in the instrument .

Response 0=MEASURED, 1=CAL

Example
```
:CALCULATE:PARAMETER:NF:RBWIDTH MEASURED
:CALCULATE:PARAMETER:NF:RBWIDTH?
-> 0
```

Explanation • This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:SNOise

Function Sets/queries whether Shot Noise is included in the NF computation

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:SNOise<wsp>OFF|ON|0|1
:CALCulate:PARameter[:CATegory]:NF:SNOise?
```

OFF | 0	Shot Noise not included in the NF computation
ON | 1	Shot Noise included in the NF computation
Response	0=OFF, 1=ON

Example
```
:CALCULATE:PARAMETER:NF:SNOISE OFF
:CALCULATE:PARAMETER:NF:SNOISE?-> 0
```

Explanation • This is a sequential command.

:CALCulate:PARameter[:CATegory]:NF:SPOWer

Function Sets or queries the signal optical power calculation method of the EDFA-NF analysis feature.

Syntax
```
:CALCulate:PARameter[:CATegory]:NF:SPOWer<wsp>PEAK | INTegral | 0 | 1
:CALCulate:PARameter[:CATegory]:NF:SPOWer?
```

PEAK|0:	The signal optical power is set to the level of the mode peak.
INTegral|1:	The signal optical power is set to the power obtained by integrating the spectrum.

Example
```
:CALCulate:PARameter:NF:SPOWer PEAK
:CALCulate:PARameter:NF:SPOWer?
-> 0
```

Explanation • This is a sequential command.

:CALCulate:PARameter[:CATegory]:NOTCh:K

Function Sets/queries the magnification of the notch width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NOTCh:K<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:NOTCh:K?
```
<NRf> = Magnification

Example
```
:CALCULATE:PARAMETER:NOTCH:K 2.00
:CALCULATE:PARAMETER:NOTCH:K?->
+2.00000000E+000
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:NOTCh:TH

Function Sets/queries the threshold value for the notch width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NOTCh:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:NOTCh:TH?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:PARAMETER:NOTCH:
TH 3.00DB
:CALCULATE:PARAMETER:NOTCH:TH?->
+3.00000000E+000
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:NOTCh:TYPE

Function Sets/queries the analysis direction of the notch width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:NOTCh:TYPE<wsp>PEAK|BOTTom|0|1
:CALCulate:PARameter[:CATegory]:NOTCh:TYPE?
```

PEAK:	Performs analysis using the peak level of a waveform as a reference.
BOTTom:	Performs analysis using the bottom level of a waveform as a reference.
Response	0 = PEAK, 1 = BOTTom

Example
```
:CALCULATE:PARAMETER:NOTCH:
TYPE BOTTOM
:CALCULATE:PARAMETER:NOTCH:TYPE? ->
1
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:PMD:TH

Function Sets/queries the threshold value for the PMD analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:PMD:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:PMD:TH?
```
<NRf> = Threshold level [dB]

Explanation
```
:CALCULATE:PARAMETER:PMD:TH 10.00DB
:CALCULATE:PARAMETER:PMD:TH?->
+1.00000000E+001
```

:CALCulate:PARameter[:CATegory]:POWer:OFFSet

Function Sets/queries the offset value for the POWER analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:POWer:OFFSet<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:POWer:OFFSet?
```
<NRf> = Offset value [dB]

Example
```
:CALCULATE:PARAMETER:POWER:
OFFSET 1.00DB
:CALCULATE:PARAMETER:POWER:OFFSET?-
> +1.00000000E+000
```

:CALCulate:PARameter[:CATegory]:SMSR:MASK

Function Sets/queries the mask value for the SMSR analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SMSR:MASK<wsp><NRf>[M]
:CALCulate:PARameter[:CATegory]:SMSR:MASK?
```
<NRf> = Mask value [m]

Example
```
:CALCULATE:PARAMETER:SMSR:
MASK 2.0nm
:CALCULATE:PARAMETER:SMSR:MASK ?->
+2.00000000E-009
```

:CALCulate:PARameter[:CATegory]:SMSR:MODE

Function Sets/queries the analysis mode for the SMSR analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SMSR:MODE<wsp>SMSR1|SMSR2|SMSR3|SMSR4
:CALCulate:PARameter[:CATegory]:SMSR:MODE?
```

Example
```
:CALCULATE:PARAMETER:SMSR:
MODE SMSR1
:CALCULATE:PARAMETER:SMSR:MODE?->
SMSR1
```

:CALCulate:PARameter[:CATegory]:SWENvelope:K

Function Sets/queries the magnification of the ENVELOPE method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWENvelope:K
:CALCulate:PARameter[:CATegory]:SWENvelope:K
```
<NRf> = Magnification

Example
```
:CALCULATE:PARAMETER:SWENVELOPE:
K 2.00
:CALCULATE:PARAMETER:SWENVELOPE:K?
-> +2.00000000E+000
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWENvelope:TH1

Function Sets/queries the search threshold level of the ENVELOPE method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWENvelope:TH1<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWENvelope:TH1?
```
<NRf> = Search threshold level [dB]

Example
```
:CALCULATE:PARAMETER:SWENVELOPE:
TH1 3.00
:CALCULATE:PARAMETER:SWENVELOPE:
TH1?-> +3.00000000E+000
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWENvelope:TH2

Function Sets/queries the threshold level of the ENVELOPE method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWENvelope:TH2<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWENvelope:TH2?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:PARAMETER:SWENVELOPE:
TH2 10.00db
:CALCULATE:PARAMETER:SWENVELOPE:
TH2?-> +1.00000000E+001
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWPKrms:K

Function Sets/queries the magnification of the PEAK-RMS method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWPKrms:K<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWPKrms:K?
```
<NRf> = Magnification

Example
```
:CALCULATE:PARAMETER:SWPKRMS:K 2.00
:CALCULATE:PARAMETER:SWPKRMS:K?->
+2.00000000E+000
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWPKrms:TH

Function Sets/queries the threshold level of the PEAK-RMS method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWPKrms:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWPKrms:TH?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:PARAMETER:SWPKRMS:TH 3.00db
:CALCULATE:PARAMETER:SWPKRMS:TH?->
+3.00000000E+000
```
Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWRMs:K

Function Sets/queries the magnification of the RMS method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWRMS:K<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWRMS:K?
```
<NRf> = Magnification

Explanation
```
:CALCULATE:PARAMETER:SWRMS:K2.00
:CALCULATE:PARAMETER:SWRMS;K? ->
+2.00000000E+000
```
Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWRMs:TH

Function Sets/queries the threshold level of the RMS method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWRMS:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWRMS:TH?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:PARAMETER:SWRMS:TH 3.00db
:CALCULATE:PARAMETER:SWRMS:TH?->
+3.00000000E+000
```
Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWTHResh:K

Function Sets/queries the magnification of the THRESH method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWTHResh:K<wsp><NRf>
:CALCulate:PARameter[:CATegory]:SWTHResh:K?
```
<NRf> = Magnification

Example
```
:CALCULATE:PARAMETER:SWTHRESH:K 2.00
:CALCULATE:PARAMETER:SWTHRESH:K?->
+2.00000000E+000
```
Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWTHresh:MFIT

Function Sets/queries whether to enable the mode fit of the THRESH method-based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWTHresh:MFIT<wsp>OFF|ON|0|1
:CALCulate:PARameter[:CATegory]:SWTHresh:MFIT?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:PARAMETER:SWTHRESH:MFIT ON
:CALCULATE:PARAMETER:SWTHRESH:MFIT?-> 1
```
Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:SWTHresh:TH

Function Sets/queries the threshold level of the THRESH method–based spectrum width analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:SWTHresh:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:SWTHresh:TH?
```
<NRf> = Threshold level [dB]

Response ex. Same as above

Explanation
```
:CALCULATE:PARAMETER:SWTHRESH:TH 3.00DB
:CALCULATE:PARAMETER:SWTHRESH:TH?->
+3.00000000E+000
```
Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:DMASk

Function Sets/queries the channel mask threshold level for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:DMASk<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:WDM
:DMASk?
```
<NRf> = Threshold level [dB] (–999: Mask OFF)

Example
```
:CALCULATE:PARAMETER:WDM:DMASK -999
:CALCULATE:PARAMETER:WDM:DMASK? ->
-9.99000000E+002
```

Explanation
- Channels the level of which are below this parameter will not be detected as a channel.
- To turn off the channel mask function, set the threshold level to –999.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:DTYPe

Function Sets/queries the displayed waveforms of the analysis results for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:DTYPe<wsp><display type>
:CALCulate:PARameter[:CATegory]:WDM
:DTYPe?
```
`<display type>`=Type of display

ABSolute = Absolute value display

RELative = Relative value display

MDRift = Drift value display based on the past measurement wavelength

GDRift = Drift value display based on the grid wavelength

Response 0 = Absolute value display

1 = Relative value display

2 = Display drift value using previously measured waveforms as a reference

3 = Display drift value using grid wavelength as a reference

Example
```
:CALCULATE:PARAMETER:WDM:DTYPE:ABSO
LUTE
:CALCULATE:PARAMETER:WDM:DTYPE:ABSO
LUTE? -> 0
```

Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:DUAL

Function Sets/queries the SNR calculation mode for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:DUAL<wsp>OFF|ON|0|1
:CALCulate:PARameter[:CATegory]:WDM
:DUAL?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:PARAMETER:WDM:DUAL ON
:CALCULATE:PARAMETER:WDM:DUAL ON?
-> 1
```

Explanation
- When this set value is 1 (ON), SNR calculation uses both traces A and B data.
- When this set value is 0 (OFF), SNR calculation uses active trace data.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:FALGo

Function Sets/queries the fitting function during level measurement applied to noise level measurements made by the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:FALGo<wsp><algorhythm>
:CALCulate:PARameter[:CATegory]:WDM
:FALGo?
```
LINear = LINEAR

GAUSs = GAUSS

LORenz = LORENZ

3RD = 3RD POLY

4TH = 4YH POLY

5TH = 5TH POLY

Response 0 = LINEAR

1 = GAUSS

2 = LORENZ

3 = 3RD POLY

4 = 4YH POLY

5 = 5TH POLY

Example:
```
CALCULATE:PARAMETER:WDM:FALGO GAUSS
:CALCULATE:PARAMETER:WDM:FALGO? ->
1
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:IRANge

Function Sets/queries the integral frequency range during signal light power calculation by the WDM analysis function

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:IRANge<wsp><NRf>
:CALCulate:PARameter[:CATegory]:WDM
:IRANge?
```
<NRf> Integral frequency range [GHz]

Example
```
calc:par:wdm:iran 40
calc:par:wdm:iran? ->
+4.00000000E+001
```

Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:MARea

Function Sets/queries the mask range during level measurement applied to noise level measurements made by the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:MARea<wsp><NRf>[M]
:CALCulate:PARameter[:CATegory]:WDM
:MARea?
```

Example
```
:CALCULATE:PARAMETER:WDM:
MAREA 0.40NM
:CALCULATE:PARAMETER:WDM:MAREA? ->
+4.00000000E-10
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:MDIFf

Function Sets/queries the peak bottom difference of channel detection for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:MDIFf<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:WDM
:MDIFf?
```
<NRf> = Peak bottom difference [dB]

Example
```
:CALCULATE:PARAMETER:WDM:
MDIFF 3.00DB
:CALCULATE:PARAMETER:WDM:MDIFF
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:MMReset

Function Resets the maximum and minimum of the drift values of the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:MMReset
```

Example
```
:CALCULATE:PARAMETER:WDM:MMRESET
```

Explanation
- When "`DISPLAY TYPE`" (set by the `:CALCulate:PARameter[:CATegory]:WDM:DTYPe` command is set to other than “DRIFT”, an execution error occurs.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:NALGo

Function Sets/queries the measurement algorithm applied to noise level measurements made by the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:NALGo<wsp><algorhythm>
:CALCulate:PARameter[:CATegory]:WDM
:NALGo?
```
AFIX|0 = AUTO FIX
MFIX|1 = MANUAL FIX
ACENter|2 = AUTO CENTER
MCENter|3 = MANUAL CENTER
PIT|4 = PIT

Response 0 = AUTO FIX
1 = MANUAL FIX
2 = AUTO CENTER
3 = MANUAL CENTER
4 = PIT

Example
```
:CALCULATE:PARAMETER:WDM:
NALGO ACENTER
:CALCULATE:PARAMETER:WDM:NALGO?-> 2
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:NARea

Function Sets/queries the measuring range applied to noise level measurements made by the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:NARea<wsp><NRf>[M]
:CALCulate:PARameter[:CATegory]:WDM
:NARea?
```
<NRf> = NOISE AREA [m]

Example
```
:CALCULATE:PARAMETER:WDM:
NAREA 0.80NM
:CALCULATE:PARAMETER:WDM:NAREA? ->
+8.00000000E-10
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:NBW

Function Sets/queries the noise bandwidth for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:NBW<wsp><NRf>[M]
:CALCulate:PARameter[:CATegory]:WDM
:NBW?
```
<NRf> = Noise bandwidth [m]

Example
```
:CALCULATE:PARAMETER:WDM:NBW 0.10NM
:CALCULATE:PARAMETER:WDM:NBW?->
+1.00000000E-010
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:OSLope

Function Sets/queries whether to enable the function of obtaining the least square approximation line in the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:OSLope<wsp>OFF|ON|0|1
:CALCulate:PARameter[:CATegory]:WDM
:OSLope?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:PARAMETER:WDM:OSLOP ON
:CALCULATE:PARAMETER:WDM:OSLOP? ->
1
```

Explanation
- When this set value is 1 (ON), this instrument calculates the least square approximation line of the peak of each channel and draws it on the waveform screen.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:PDISplay

Function Sets/queries whether to display data used for fitting of the WDM analysis function on the waveform screen.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:PDISplay<wsp>OFF|ON|0|1
:CALCulate:PARameter[:CATegory]:WDM
:PDISplay?
```
Response 0 = OFF, 1 = ON

Example
```
:CALCULATE:PARAMETER:WDM:
PDISPLAY ON
:CALCULATE:PARAMETER:WDM:PDISPLAY?-
> 1
```

Explanation
- When this set value is 1 (ON), data used for fitting is displayed on the waveform screen.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:RCH

Function Sets/queries the reference channel used in calculating the offset wavelength/level of the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:RCH<wsp><integer>
:CALCulate:PARameter[:CATegory]:WDM
:RCH?
```
<integer> = Reference channel number
(0: channel with the highest level)

Example
```
:CALCULATE:PARAMETER:RCH 10
:CALCULATE:PARAMETER:RCH? -> 10
```

Explanation
- When this set value is “0,” the channel with the highest level is regarded as the reference channel.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:RELation

Function Sets/queries the display format of the wavelength/level relative values for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:RELation<wsp>OFFSet|SPACing|0|1
:CALCulate:PARameter[:CATegory]:WDM
:RELation?
```
OFFSet|0 = Displays an offset value based on any channel.

SPACing|1 = Displays an offset value relative to a neighboring channel.

Response 0 = OFFSET, 1 = SPACING

Example
```
:CALCULATE:PARAMETER:WDM:
RELATION SPACING
:CALCULATE:PARAMETER:WDM:RELATION?-
> 1
```

Explanation
- When "DISPLAY TYPE" (set by the :CALCulate:PARameter[:CATegory]:WDM:DTYPe command is set to other than “ABSOLUTE”, an execution error occurs.
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:SPOWer

Function Sets/queries the signal light power calculation method of the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:SPOWer<wsp>PEAK|INTegral|0|1
:CALCulate:PARameter[:CATegory]:WDM
:SPOWer?
```
PEAK|0 = Sets the mode peak to the signal light power

INTegral|1 = Sets the power that integrates the spectrum to the signal light power

Example
```
:CALCULATE:PARAMETER:WDM:SPOwer PEAK
:CALCULATE:PARAMETER:WDM:SPOwer?
-> 0
```

Explanation
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WDM:TH

Function Sets/queries the threshold level of channel detection for the WDM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WDM
:TH<wsp><NRf>[DB]
:CALCulate:PARameter[:CATegory]:WDM
:TH?
```
<NRf> = Threshold level [dB]

Example
```
:CALCULATE:PARAMETER:WDM:TH 20.00db
:CALCULATE:PARAMETER:WDM:TH->
+2.00000000E+001
```

Explanation This is a sequential command.

:CALCulate:PARameter[:CATegory]:WFBottom

Function Sets/queries parameters for the WDM FILTER-BTM analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WFBottom<wsp><item>,<paramater>,<data>
:CALCulate:PARameter[:CATegory]:WFBottom?<wsp><item>,<paramater>
```
<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Data to be set

<item>	<parameter>	<data>
NWAVelength	ALGO	BOTtom|NPEak|NBOTtom|GFIT|GRID
	MDIFf	<NRf>[DB]
	TH	<NRf>[DB]
	TBANd	<NRf>[M]
BWAVelength	SW	OFF|ON|0|1
CWAVelength	SW	OFF|ON|0|1
	ALGO	NPEak|NBOTtom
	TH	<NRf>[DB]
SBANd	SW	OFF|ON|0|1
	TH	<NRf>[DB]
EBANd	SW	OFF|ON|0|1
	TH	<NRf>[DB]
	TBANd	<NRf>[M]
RIPPle	SW	OFF|ON|0|1
	TBANd	<NRf>[M]
XTALk	SW	OFF|ON|0|1
	SPACing	<NRf>[M]
	TBANd	<NRf>[M]

Example
```
:CALCULATE:PARAMETER:WFBOTTOM
NWAY,ALGO,NPEAK
:CALCULATE:PARAMETER:WFBOTTOM?
NWAY,ALGO -> NPE:CALCULATE:
PARAMETER:WFBOTTOM BWAVELENGTH,SW,
OFF
:CALCULATE:PARAMETER:WFBOTTOM?
BWAVELENGTH,SW -> 0
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs (a combination of NWAVelength and SPACing, etc.).
- This is a sequential command.

:CALCulate:PARameter[:CATegory]:WFPeak

Function Sets/queries parameters for the WDM FILTER-PEAK analysis function.

Syntax
```
:CALCulate:PARameter[:CATegory]:WFPeak<wsp><item>,<paramater>,<data>
:CALCulate:PARameter[:CATegory]:WFPeak?<wsp><item>,<paramater>
```
<item> = Analytical item that sets parameter(s)
<parameter> = Parameter to be set
<data> = Data to be set

<item>	<parameter>	<data>
NWAVelength	ALGO	PEAK|MEAN|GFIT|GRID
	MDIFf	<NRf>[DB]
	TH	<NRf>[DB]
	TBANd	<NRf>[M]
PWAVelength	SW	OFF|ON|0|1
CWAVelength	SW	OFF|ON|0|1
	TH	<NRf>[DB]
SBANd	SW	OFF|ON|0|1
	TH	<NRf>[DB]
PBANd	SW	OFF|ON|0|1
	TH	<NRf>[DB]
	TBANd	<NRf>[M]
RIPPle	SW	OFF|ON|0|1
	TBANd	<NRf>[M]
XTALk	SW	OFF|ON|0|1
	SPACing	<NRf>[M]
	TBANd	<NRf>[M]

Example
```
:CALCULATE:PARAMETER:WFPEAK
NWAY,ALGO,PEAK
:CALCULATE:PARAMETER:WFPEAK?
NWAY,ALGO -> PEAK
:CALCULATE:PARAMETER:WFPEAK
BWAVELENGTH,SW,OFF
:CALCULATE:PARAMETERWFPEAK?
BWAVELENGTH,S -> 0
```

Explanation
- If a non-existing parameter is used for a combination, an execution error occurs (a combination of NWAVelength and SPACing, etc.).
- This is a sequential command.

:CALCulate:PARameter:COMMon:MDIFf

Function Sets/queries the peak-bottom difference parameter of channel detection used in the analysis function.

Syntax
```
:CALCulate:PARameter:COMMon:MDIFf<wsp><NRf>[DB]
:CALCulate:PARameter:COMMon:MDIFf?
```

Example
```
:CALCULATE:PARAMETER:COMMON:
MDIFF 3.00DB
:CALCULATE:PARAMETER:COMMON:MDIFF->
+3.00000000E+000
```

Explanation This is a sequential command.

6

Remote Commands

CALibration Sub System Command

:CALibration:ALIGn[:IMMediate]

Function Executes optical axis adjustment of the monochromator using the internal reference light source.

Syntax `:CALibration:ALIGn[:IMMediate]`

Example `:CALIBRATION:ALIGN`

Explanation This is an overlapable command.

:CALibration:WAVelength:EXTernal[:IMMediate]

Function Performs wavelength calibration using an external reference light source.

Syntax `:CALibration:WAVelength:EXTernal[:IMMediate]`

Example `:CALIBRATION:WAVELENGTH:EXTERNAL1`

Explanation
- The type of the external reference light source to be used for calibration is set using the `CALibration:WAVelength:EXTernal:SOURce` command.
- The wavelength of the external reference light source to be used for calibration is set using the `CALibration:WAVelength:EXTernal:WAVelenght` command.
- This is an overlapable command.

:CALibration:WAVelength:EXTernal:SOURce

Function Sets/queries the type of the light source used for external reference light source-based wavelength calibration.

Syntax `:CALibration:WAVelength:EXTernal:SOURce<wsp>LASer|GASCell|EMISsion|0|1|2`

`:CALibration:WAVelength:EXTernal:SOURce?`

LASer = An external reference light source is used for the laser

GASCell = A gas cell is used as the external reference light source.

EMISsion = An emission light is used as the external reference light source.

Response 0 = Laser, 1 = Gas cell, 2 = Emission light

Example `:CALIBRATION:WAVELENGTH:EXTERNAL1:SOURCE LASER`

`:CALIBRATION:WAVELENGTH:EXTERNAL1:SOURCEe? -> 0`

Explanation
- Of the level offset table, the command sets or queries the offset value of a wavelength specified by <integer>.
- This is a sequential command.

:CALibration:WAVelength:EXTernal:WAVelength

Function Sets/queries the wavelength of the light source used for external reference light source-based wavelength calibration.

Syntax `:CALibration:WAVelength:EXTernal:WAVelength<wsp><NRf>[M]`

`:CALibration:WAVelength:EXTernal:WAVelength?`

<NRf> = Wavelength of the external reference light source [nm]

Example `:CALIBRATION:WAVELENGTH:EXTERNAL1:WAVELENGTH 1550.000NM`

`:CALIBRATION:WAVELENGTH:EXTERNAL1:WAVELENGTH? -> +1.55000000E-006`

Explanation This is a sequential command.

:CALibration:WAVelength:INTernal[:IMMediate]

Function Performs wavelength calibration using an internal reference light source.

Syntax `:CALibration:WAVelength:INTernal[:IMMediate]`

Example `:CALIBRATION:WAVELENGTH:INTERNAL1`

Explanation
- This is an overlapable command.

:CALibration:ZERO[:AUTO]

Function Sets/queries whether to enable the auto offset function of the level.

Syntax
```
:CALibration:ZERO[:AUTO]<wsp>OFF|ON
|0|1|ONCE
:CALibration:ZERO[:AUTO]?
```
Response 0 = OFF, 1 = ON

Example
```
:CALIBRATION:ZERO ONCE
:CALIBRATION:ZERO? -> 1
```

Explanation
- If you send this command with the parameter “ONCE” when the sweep is stopped, offset adjustment is performed once. In this case, ON/OFF of this setting does not change.
- The operation of this command is complete at the instant the offset adjustment starts. Therefore, the AQ6374 can execute the next command even while offset adjustment is being performed. You can use :CALibration:ZERO[:AUTO]:STATus? to query the execution status of the offset adjustment.
- This is a sequential command.

:CALibration:ZERO[:AUTO]:INTerval

Function Sets/queries the time interval for executing the Auto Offset function for the level.

Syntax
```
:CALibration:ZERO[:AUTO]:INTerval<wsp><integer>
:CALibration:ZERO[:AUTO]:INTerval?
```
<integer>= Interval of execution (specified in units of minutes)

Example
```
:CALIBRATION:ZERO:INTERVAL 20
:CALIBRATION:ZERO:INTERVAL? -> 20
```

Explanation
- When a time is set for this parameter, the auto offset adjustment is performed at the specified time interval starting from the moment of execution.
- This is a sequential command.

:CALibration:ZERO[:AUTO]:STATus?

Function Queries the offset adjustment status.

Syntax
```
:CALibration:ZERO[:AUTO]:STATus?
```
0: The offset adjustment is not being executed.

1: The offset adjustment is being executed.

Example
```
:CALIBRATION:ZERO:STATUS? -> 1
```

Explanation
- This is a sequential command.

DISPlay Sub System Command

:DISPlay:COLor

Function Sets/queries the screen color mode.

Syntax
```
:DISPlay:COLor<wsp><mode>
:DISPlay:COLor?
```
0 = Black and white mode

1 = Color mode

Example
```
:DISPLAY:COLOR 1
:DISPLAY:COLOR? -> 1
```

Explanation This is a sequential command.

:DISPlay[:WINDow]

Function Sets/queries whether the display is enabled.

Syntax
```
:DISPlay[:WINDow]<wsp>OFF|ON|0|1
:DISPlay[:WINDow]?
```
Response 0 = OFF, 1 = ON

Example
```
:DISPLAY OFF
:DISPLAY? -> 0
```

Explanation
- This is a sequential command.

:DISPlay[:WINDow]:OVIew:POSition

Function Sets/queries the ON/OFF and position of the OVERVIEW display shown during zoom operation.

Syntax
```
:DISPlay[:WINDow]:OVIew:POSition<wsp>OFF|LEFT|RIGHt|0|1|2
:DISPlay[:WINDow]:OVIew:POSition?
```
OFF = Display OFF

LEFT = The overview display is on the left of the screen.

RIGHt = The overview display is on the right of the screen.

Response 0 = OFF, 1 = LEFT, 2 = RIGHt

Example
```
:DISPLAY:OVIEW:POSITION RIGHT
:DISPLAY:OVIEW:POSITION? -> 2
```

Explanation This is a sequential command.

:DISPlay[:WINDow]:OVIew:SIZE

Function Sets/queries the size of the OVERVIEW display shown during zoom operation.

Syntax
```
:DISPlay[:WINDow]:OVIew:SIZE<wsp>LARGe|SMALl|0|1
:DISPlay[:WINDow]:OVIew:SIZE?
```
LARGe = Larger OVERVIEW size

SMALl = Smaller OVERVIEW size

Response 0 = LARGe, 1 = SMALl

Example
```
:DISPLAY:OVIEW:SIZE LARGE
:DISPLAY:OVIEW:SIZE? -> 0
```

Explanation This is a sequential command.

:DISPlay[:WINDow]:SPLit

Function Sets/queries whether to split the screen display into two parts.

Syntax `:DISPlay[:WINDow]:SPLit<wsp>OFF|ON|0|1`
`:DISPlay[:WINDow]:SPLit?`
Response 0 = OFF, 1 = ON

Example `:DISPLAY:SPLIT ON`
`:DISPLAY:SPLIT? -> 1`

Explanation This is a sequential command.

:DISPlay[:WINDow]:SPLit:HOLD:LOWer

Function Sets/queries whether to fix a trace assigned to the lower area when the screen is in the upper/lower 2-split display mode.

Syntax `:DISPlay[:WINDow]:SPLit:HOLD:LOWer<wsp>OFF|ON|0|1`
`:DISPlay[:WINDow]:SPLit:HOLD:LOWer?`
Response 0 = OFF, 1 = ON

Example `:DISPLAY:SPLIT:HOLD:LOWER ON`
`:DISPLAY:SPLIT:HOLD:LOWER? -> 1`

Explanation If not in 2-split screen display mode, an execution error occurs.

:DISPlay[:WINDow]:SPLit:HOLD:UPPer

Function Sets/queries whether to fix a trace assigned to the upper area when the screen is in the upper/lower 2-split display mode.

Syntax `:DISPlay[:WINDow]:SPLit:HOLD:UPPer<wsp>OFF|ON|0|1`
`:DISPlay[:WINDow]:SPLit:HOLD:UPPer?`
Response 0 = OFF, 1 = ON

Example `:DISPLAY:SPLIT:HOLD:UPPER ON`
`:DISPLAY:SPLIT:HOLD:UPPER? -> 1`

Explanation
- If not in 2-split screen display mode, an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:SPLit:POSition

Function Sets/queries whichever display area, upper or lower, is used to display a trace when the screen is in the upper/lower 2-split display mode.

Syntax `:DISPlay[:WINDow]:SPLit:POSition<wsp><trace name>,UP|LOW|0|1`
`:DISPlay[:WINDow]:SPLit:POSition?<wsp><trace name>`
<trace name> = trace name (TRA,TRB,TRC,TRD,TRE,TRF,TRG)
UP = Trace is displayed in the upper area.
LOW = Trace is displayed on the lower area.
Response 0 = UP, 1 = LOW

Example `:DISPLAY:SPLIT:POSITION TRA,UP`
`:DISPLAY:SPLIT:POSITION? TRA -> 0`

Explanation This is a sequential command.

:DISPlay[:WINDow]:TEXT:CLEar

Function Clears labels.

Syntax `:DISPlay[:WINDow]:TEXT:CLEar`

Example `:DISPLAY:TEXT:CLEAR`

Explanation This is a sequential command.

:DISPlay[:WINDow]:TEXT:DATA

Function Sets/queries the labels.

Syntax `:DISPlay[:WINDow]:TEXT:DATA<wsp><string>`
`:DISPlay[:WINDow]:TEXT:DATA?`
<string> = Label character string (56 characters max.)

Example `:DISPLAY:TEXT:DATA "Optical Spectrum Analyzer"`
`:DISPLAY:TEXT:DATA?->`
`Optical Spectrum Analyzer`

Explanation
- A label character string has a maximum length of 56 characters.If a label of more than 56 characters is specified, characters from and exceeding the 57th will be ignored.
- If there is no label, one space character is returned.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:CENTer

Function Sets/queries the center wavelength of the X-axis of the display scale.

Syntax `:DISPlay[:WINDow]:TRACe:X[:SCALe]:CENTer<wsp><NRf>[M|HZ]`
`:DISPlay[:WINDow]:TRACe:X[:SCALe]:CENTer?`
<NRf> = Center wavelength [m|Hz]
Response
<NRf>`[m|Hz]`

Example `:DISPLAY:TRACE:X:CENTER 1550.000NM`
`:DISPLAY:TRACE:X:CENTER?->`
`+1.55000000E-006`

Explanation
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:INITialize

Function Initializes the X-axis parameters of the display scale.

Syntax `:DISPlay[:WINDow]:TRACe:X[:SCALe]:INITialize`

Example `:DISPLAY:TRACE:X:INITIALIZE`

Explanation
- The following parameters are initialized based on the measurement scale after this command has been executed.
  ZOOM CENTER, ZOOM SPAN, ZOOM START, ZOOM STOP
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:SMSCale

Function Sets parameters of the current display scale to the measurement scale.

Syntax
```
:DISPlay[:WINDow]:TRACe:X[:SCALe]:SMSCale
```

Example
```
:DISPLAY:TRACE:X:SMSCALE
```

Explanation
- The following parameters are initialized based on the display scale after this command has been executed.
  CENTER, SPAN, START, STOP
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:SPAN

Function Sets/queries the span of the X-axis of the display scale.

Syntax
```
:DISPlay[:WINDow]:TRACe:X[:SCALe]:SPAN<wsp><NRf>[M|HZ]
:DISPlay[:WINDow]:TRACe:X[:SCALe]:SPAN?
```
<NRf> = Span [m|Hz]

Response

<NRf>[m|Hz]

Example
```
:DISPLAY:TRACE:X:SPAN 20.0NM
:DISPLAY:TRACE:X:SPAN? ->
+2.00000000E-008
```

Explanation
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:SRANge

Function Sets/queries whether to limit an analytical range to the display scale range.

Syntax
```
:DISPlay[:WINDow]:TRACe:X[:SCALe]:SRANge<wsp>OFF|ON|0|1
:DISPlay[:WINDow]:TRACe:X[:SCALe]:SRANge?
```
Response 0 = OFF, 1 = ON

Example
```
:DISPLAY:TRACE:X:SRANGE on
:DISPLAY:TRACE:X:SRANGE? -> 1
```

Explanation This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:STARt

Function Sets/queries the start wavelength of the X-axis of the display scale.

Syntax
```
:DISPlay[:WINDow]:TRACe:X[:SCALe]:STARt<wsp><NRf>[M|HZ]
:DISPlay[:WINDow]:TRACe:X[:SCALe]:STARt?
```
<NRf> = Start wavelength [m|Hz]

Response

<NRf>[m|Hz]

Example
```
:DISPLAY:TRACE:X:START 1540.000NM
:DISPLAY:TRACE:X:START?->
+1.54000000E-006
```

Explanation
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:X[:SCALe]:STOP

Function Sets/queries the stop wavelength of the X-axis of the display scale.

Syntax
```
:DISPlay[:WINDow]:TRACe:X[:SCALe]:STOP<wsp><NRf>[M|HZ]
:DISPlay[:WINDow]:TRACe:X[:SCALe]:STOP?
```
<NRf> = Stop wavelength [m|Hz]

Response

<NRf>[m|Hz]

Example
```
:DISPLAY:TRACE:X:STOP 1560.000NM
:DISPLAY:TRACE:X:STOP?->
+1.56000000E-006
```

Explanation
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y:NMASk

Function Sets whether to mask the display of waveforms the level of which is at or below a set threshold level or queries the condition of whether the relevant waveform display is masked.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y:NMASk<wsp><NRf>[DB]
:DISPlay[:WINDow]:TRACe:Y:NMASk?
```
<NRf> = Threshold level [dB] (–999: Masking function OFF)

Example
```
:DISPLAY:TRACE:Y:MASK -999
:DISPLAY:TRACE:Y:MASK? ->
-9.99000000E+002
```

Explanation
- The display of waveforms the level of which is at or below this parameter will be masked.
  To turn off the mask function, set the threshold level to –999.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y:NMASk:TYPE

Function Sets/queries the display method when a waveform display at or below a threshold level is masked.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y:NMASk:TYPE<wsp>VERTical|HORIzontal|0|1
:DISPlay[:WINDow]:TRACe:Y:NMASk:TYPE?
```
VERTical = Waveform display with zero as the mask value or lower

HORizontal = Waveform display with the mask value as the mask value or lower

Response 0 = VERTical, 1 = HORizontal

Example
```
:DISPLAY:TRACE:Y:MASK:TYPE VERTICAL
:DISPLAY:TRACE:Y:MASK:TYPE? -> 0
```
Explanation This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y[:SCALe]:DNUMber

Function Sets/queries the number of display divisions of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y[:SCALe]:DNUMber<wsp>8|10|12
:DISPlay[:WINDow]:TRACe:Y[:SCALe]:DNUMber?
```
8, 10, 12 = Number of display divisions

Example
```
:DISPLAY:TRACE:Y:DNUMBER 10
:DISPLAY:TRACE:Y:DNUMBER? -> 10
```
Explanation This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:BLEVel

Function Sets/queries the base level applied when the main scale of the level axis is linear.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:BLEVel<wsp><NRf>[W]
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:BLEVel?
```
<NRf> = Base level value [W]

Example
```
:DISPLAY:TRACE:Y1:BLEVEL 1.0MW
:DISPLAY:TRACE:Y1:BLEVEL?->
+1.00000000E-003
```
Explanation
- If a instrument other than W is specified, an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:PDIVision

Function Sets/queries the main scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:PDIVision<wsp><NRf>[DB]
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:PDIVision?
```
<NRf> = Level scale [dB]

Example
```
:DISPLAY:TRACE:Y1:PDIV 5.0DB
:DISPLAY:TRACE:Y1:PDIV?->
+5.00000000E+000
```
Explanation
- If a instrument other than dB is specified, an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RLEVel

Function Sets/queries the reference level of the main scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RLEVel<wsp><NRf>[DBM|W]
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RLEVel?
```
<NRf> = Reference level [dB|W]

Example
```
:DISPLAY:TRACE:Y1:RLEVEL -30dbm
:DISPLAY:TRACE:Y1:RLEVEL?->
-3.00000000E+001
```
Explanation
- When the unit is omitted in the parameter, the reference level is set in dBm if the main scale of the level axis is in the LOG mode or is set in W if it is in the linear mode.
- If the setting condition of the LOG/linear mode of the level axis' main scale does not match the unit specified in the parameter of the command, the parameter of this command is translated matching the LOG/linear mode of the main scale. For example, when the main scale is LOG and you set the reference level to 1m with this command, the reference level is set to 0 dB.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RPOSition

Function Sets/queries the position of the reference level of the main scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:
RPOSition<wsp><integer>[DIV]
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:
RPOSition?
```
<integer> = Position of the reference level

Example
```
:DISPLAY:TRACE:Y1:RPOSITION 10DIV
:DISPLAY:TRACE:Y1:RPOSITION? -> 10
```

Explanation
- If a value greater than the number of display divisions of the level axis is specified for the position of the reference level, the position of this level is treated as the top of the scale.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing

Function Sets/queries the scale mode of the main scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:
SPACing<wsp>LOGarighmic|LINear|0|1
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:
SPACing?
```
LOGarithmic = LOG scale
LINear = Linear scale
Response 0 = LOGarithmic, 1 = LINear

Example
```
:DISPLAY:TRACE:Y1:SPACING LINIER
:DISPLAY:TRACE:Y1:SPACING? -> 1
```

Explanation This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:UNIT

Function Sets/queries the units of the main scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:
UNIT<wsp><unit>
:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:
UNIT?
```
DBM = dBm
W = W
DBM/NM = dBm/nm or dBm/THz
W/NM = W/nm or W/THz
Response 0 = dBm
1 = W
2 = DBM/NM
3 = W/NM

Example
```
:DISPLAY:TRACE:Y1:UNIT DBM/NM
:DISPLAY:TRACE:Y1:UNIT? -> 2
```

Explanation
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:AUTO

Function Sets/queries the automatic setting function of the sub scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
AUTO<wsp>OFF|ON|0|1
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
AUTO?
```
Response 0 = OFF, 1 = ON

Example
```
:DISPLAY:TRACE:Y2:AUTO ON
:DISPLAY:TRACE:Y2:AUTO? -> 1
```

Explanation This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:LENGth

Function Sets/queries the parameter of the optical fiber length used when the unit of the subscale of the level axis is dB/km.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
LENGth<wsp><NRf>[KM]
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
LENGth?
```
<NRf> = Length of optical fiber [km]

Example
```
:DISPLAY:TRACE:Y2:LENGTH 99.999KM
:DISPLAY:TRACE:Y2:LENGTH?->
+9.99990000E+001
```

Explanation
- When the unit of the subscale is set to other than "dB/km", an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:OLEVel

Function Sets/queries the offset level of the sub scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
OLEVel<wsp><NRf>[DB|DB/KM]
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
OLEVel?
```
<NRf> = Offset level [dB|dB/km]

Example
```
:DISPLAY:TRACE:Y2:OLEVEL 10DB/KM
:DISPLAY:TRACE:Y2:OLEVEL? ->
+1.00000000E+001
```

Explanation
- When the unit of the subscale is set to other than "dB" or "dB/km", an execution error occurs.
- If the unit is not specified in the parameter, dB is set if the subscale of the level axis is in the dB mode or dB/km is set if it is in the dB/km mode.
- If a unit different from the current set unit (`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:UNIT`) of the subscale is specified, an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:PDIVision

Function Sets/queries the sub scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
PDIVision<wsp><NRf>[DB|DB/KM | %]
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
PDIVision?
```
<NRf> = Level scale [dB | dB/km | %]

Example
```
:DISPLAY:TRACE:Y2:PDIVISION 5.0%
:DISPLAY:TRACE:Y2:PDIVISION? ->
+5.00000000E+000
```

Explanation
- If the unit is not specified in the parameter, the set unit of the subscale of the level axis is used as the set unit of this parameter.
- If a unit different from the current set unit (`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:UNIT`) of the subscale is specified, an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:RPOSition

Function Sets/queries the position of the reference level of the sub scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
RPOSition<wsp><integer>[DIV]
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
RPOSition?
```
<integer> = Position of the reference level

Example
```
:DISPLAY:TRACE:Y2:RPOSITION 10DIV
:DISPLAY:TRACE:Y2:RPOSITION? -> 10
```

Explanation
- If a value greater than the number of display divisions of the level axis is specified for the position of the reference level, the position of this level is treated as the top of the scale.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:SMINimum

Function Sets/queries the value of the bottom of the scale applied when the subscale of the level axis is set to the linear or % mode.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
SMINimum<wsp><NRf>[%]
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
SMINimum?
```
<NRf> = Value of the bottom of the scale [%]

Example
```
:DISPLAY:TRACE:Y2:SMINIMUM 0%
:DISPLAY:TRACE:Y2:SMINIMUM? -> 0
```

Explanation
- If the unit is not specified in the parameter, the set unit of the subscale of the level axis is used as the set unit of this parameter.
- If a unit different from the current set unit (`:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:UNIT`) of the subscale is specified, an execution error occurs.
- This is a sequential command.

:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:UNIT

Function Sets/queries the units of the sub scale of the level axis.

Syntax
```
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
UNIT<wsp><unit>
:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:
UNIT?
```
<unit> = Units
DB = dB display
LINear = Linear display
DB/KM = dB/km display
% = % display

Response 0 = DB
1 = LINear
2 = DB/KM
3 = %

Example
```
:DISPLAY:TRACE:Y2:UNIT DB/KM
:DISPLAY:TRACE:Y2:UNIT? -> 2
```

Explanation This is a sequential command.

FORMat Sub System Command

:FORMat[:DATA]

Function Sets/queries the format used for data transfer via GP-IB.

Syntax `:FORMat[:DATA]<wsp>REAL[,64|,32]|ASCii`

`:FORMat[:DATA]?`

ASCii = ASCII format (default)

REAL[,64] = REAL format (64bits)

REAL,32 = REAL format (32bits)

Example
```
FORMAT:DATA REAL,64
FORMAT:DATA? -> REAL,64
FORMAT:DATA REAL,32 FORMAT:DATA? ->
REAL,32
FORMAT:DATA ASCII
FORMAT:DATA? -> ASCII
```

Explanation
- When the format is set to REAL (binary) using this command, the output data of the following commands are produced in the REAL format.
  ```
  :CALCulate:DATA:CGAin?
  :CALCulate:DATA:CNF?
  :CALCulate:DATA:CPOWers?
  :CALCulate:DATA:CSNR?
  :CALCulate:DATA:CWAVelengths?
  :TRACe[:DATA]:X?
  :TRACe[:DATA]:Y?
  ```
- The default is ASCII mode.
- When the *RST command is executed, the format is reset to the ASCII mode.
- The ASCII format outputs a list of numerics each of which is delimited by a comma (,). Example: 12345,12345,....
- By default, the REAL format outputs data in fixed length blocks of 64 bits, floating-point binary numerics.
- If “REAL,32” is specified in the parameter, data is output in the 32-bit, floating-point binary form.
- The fixed length block is defined by IEEE 488.2 and consists of “#” (ASCII), one numeric (ASCII) indicating the number of bytes that specifies the length after #, length designation (ASCII), and binary data of a specified length in this order. Binary data consists of a floating-point data string of 8 bytes (64 bits) or 4 bytes (32 bits). Floating-point data consists of lower-order bytes to higher-order bytes.
  E.g.: #18 [eight <byte data>]
  #280[80 <byte data>]
  #48008[8008 <byte data>]
- For data output in the 32-bit floating-point binary form, cancellation of significant digits is more likely to occur in comparison with transfer of data in the 64-bit, floating-point binary form.
- This is a sequential command.

HCOPY Sub System Command

:HCOPY:DESTination

Function Sets/queries the data output destination.

Syntax `:HCOPY:DESTination<wsp>|FILE|2`

`:HCOPY:DESTination?`

FILE = File

Response 2 = FILE

Example
```
:HCOPY:DESTINATION FILE
:HCOPY:DESTINATION? -> 2
```

Explanation • This is a sequential command.

:HCOPY[:IMMediate]

Function Makes a hard copy of the screen display.

Syntax `:HCOPY[:IMMediate]`

Example `:HCOPY`

Explanation This is an overlapable command.

INITiate Sub System Command

:INITiate[:IMMediate]

Function Makes a sweep.

Syntax `:INITiate[:IMMediate]`

Example `:INITIATE`

Explanation
- You can stop sweep with the `:ABORt` command.
- The sweep mode (AUTO, SINGLE, REPEAT, or SEGMENT MEASURE) is set using the `:INITiate:SMODe` command.
- If this command is executed while the sweep mode is in REPEAT (`:INITiate:SMODe REPeat`), the operation of the command is complete at the instant a sweep starts. In this case, this command is regarded as a sequential command.
- If this command is executed while the sweep mode is one of AUTO, SINGLE, and SEGMENT MEASURE, the operation of the command is complete at the instant a sweep ends. In this case, this command is regarded as a command subject to overlapping.

:INITiate:SMODe

Function Sets/queries the sweep mode.

Syntax `:INITiate:SMODe<wsp><sweep mode>`
`:INITiate:SMODe?`
<sweep mode> = Sweep mode
SINGle = SINGLE sweep mode
REPeat = REPEAT sweep mode
AUTO = AUTO sweep mode
SEGMent = SEGMENT
Response 1 = SINGle
2 = REPeat
3 = AUTO
4 = SEGMent

Example `:INITIATE:SMODE REPEAT`
`:INITIATE:SMODE? -> 2`

Explanation This is a sequential command.

MEMory Sub System Command

:MEMory:CLEar

Function Clears the contents of a specified waveform memory.

Syntax `:MEMory:CLEar<wsp><integer>`
<integer> = Memory number

Example `:MEMORY:CLEAR 10`

Explanation
- No execution error occurs even if a specified waveform memory has already been cleared.
- This is a sequential command.

:MEMory:EMPTy?

Function Queries the condition of whether a waveform has been specified in a specified waveform memory.

Syntax `:MEMory:EMPTy?<wsp><integer>`
<integer> = Memory number

Example `:MEMORY:EMPTY? 10 -> 1`

Explanation This is a sequential command.

:MEMory:LOAD

Function Loads a waveform from a specified waveform memory into a specified trace.

Syntax `:MEMory:LOAD<wsp><integer>,<trace name>`
<integer> = Memory number
<trace name> = trace
(TRA,TRB,TRC,TRD,TRE,TRF,TRG)

Example `:MEMORY:LOAD 10,TRA`

Explanation
- When a waveform is not registered in the specified waveform memory, a warning message appears.
- This is a sequential command.

:MEMory:STORe

Function Stores the waveform of a specified trace into a specified waveform memory.

Syntax `:MEMory:STORe<wsp><integer>,<trace name>`
<integer> = Memory number
<trace name>= trace
(TRA,TRB,TRC,TRD,TRE,TRF,TRG)

Example `:MEMORY:STORE 10,TRA`

Explanation
- When waveform data do not exist in the specified trace, a warning message appears.
- This is a sequential command.

MMEMory Sub System Command

Common Items

- To include a directory name in <"filename">, specify the path in the following manner.
  - Specification of an absolute path
    When the head of <"file name"> is character "\", specify the absolute path.
  - Relative path specification
    When the head of <"file name"> is any character other than "\", specify the the relative path from the current directory. The current directory is specified using the `:MMEMory:CDIRectory` command.
- If INTernal|EXTernal is not specified, access is made to the current drive.
  The current drive is specified using the `:MMEMory:CDRive` command.
- If a file name extension is omitted when storing a file, an extension corresponding to the data type will be appended to the file name.
- When loading a file, the file name extension can be omitted.

:MMEMory:ANAMe

Function Sets or queries the naming rule for automatic file names.

Syntax
```
:MMEMory:ANAMe<wsp>NUMBer|DATE|0|1
:MMEMory:ANAMe?
NUMBer|0     Number
DATE|1       Timestamp
```

Example
```
:MMEMORY:ANAME DATE
:MMEMORY:ANAME? -> 1
```

Description
- This is a sequential command.

:MMEMory:CATalog?

Function Queries a list of all files in the current directory.

Syntax
```
:MMEMory:CATalog?<wsp>[INTernal|
EXTernal][,<directory name>]
```

INTernal = Acquires a file list in the current directory of the internal memory.

EXTernal = Acquires a file list in the current directory of the external USB storage.

directory name = Default name

Response

<free size>,<file number>,<file name>,<file name>, … ,<file name>

<free size> = <NRf> Disk's free size [KB] (1KB=1024 bytes))

<file number>= <integer> number of files

<file name> = File name

Example
```
:MMEMORY:CATALOG? INTERNAL,"\TEST\
SAMPLE"
 -> +1.91176800E+006,2,
test0001.WX9,test0002.WX9
```

Explanation This is a sequential command.

:MMEMory:CDIRectory

Function Sets/queries the current directory.

Syntax
```
:MMEMory:CDIRectory<wsp><directory
name>
:MMEMory:CDIRectory?
```

<directory name> = Directory name to be changed

Example
```
:MMEMORY:CDIRECTORY "\test\sample"
:MMEMORY:CDIRECTORY? ->
\test\sample
```

Explanation This is a sequential command.

:MMEMory:CDRive

Function Sets/queries the current drive.

Syntax
```
:MMEMory:CDRive<wsp>INTernal|EXTernal
:MMEMory:CDRive?
```

INTernal = Makes the current drive the internal memory.

EXTernal = Makes the current drive the external USB storage.

Example
```
:MMEMORY:CDRIVE INTERNAL
:MMEMORY:CDRIVE -> INT
```

Explanation This is a sequential command.

:MMEMory:COPY

Function Copies a specified file.

Syntax
```
:MMEMory:COPY<wsp>
<"source file name">,[INTernal|
EXTernal],
<"destination file name">[,INTernal|
EXTernal]
```

<"source file name"> = File name at the copy source

<"destination file name"> = File name at the copy destination

Example
```
:MMEMORY:COPY "test001.wv7",,
"test002.wv7"
```

Explanation This is a sequential command.

:MMEMory:DATA?

Function Queries the data in the specified file.

Syntax
```
:MMEMory:DATA?<wsp><"file name">
[,INTernal| EXTernal]
```

<"file name">= Name of the file to be read

Response

The data that was read (binary block data of fixed length starting with "#")

Example
```
:MMEMORY:DATA? "test.csv",internal
-> #18ABCDEFGH
```

Explanation
- Maximum file size that can be sent is 3 MB.
- For the data format of the fixed length blocks, see `:FORMat` Command.
- This is a sequential command.

:MMEMory:DELete

Function Deletes a specified file.

Syntax `:MMEMory:DELete<wsp><"file name">[,INTernal|EXTernal]`
<"file name"> = Name of a file to be deleted

Example `:MMEMORY:DELETE "test002.WX9", internal`

Explanation This is a sequential command.

:MMEMory:LOAD:ATRace

Function Loads the specified waveform files (all traces) into traces.

Syntax `:MMEMory:LOAD:ATRace<wsp>`
<”file name”> [,INTernal|EXTernal]
<"file name"> = Name of file to load
INTernal|EXTernal = Source drive for loading

Example: `MMEMORY:LOAD:ATRACE "test001.csv",internal`

Explanation This is a sequential command.

:MMEMory:LOAD:DLOGging

Function Loads the specified data logging file.

Syntax `:MMEMory:LOAD:DLOGging<wsp><"filename">[,INTernal|EXTernal]`
<"filename"> = Name of the file to load
INTernal|EXTernal = Source drive to load from

Example `:MMEMORY:LOAD:DLOGGING "test001.LX9",INTERNAL`

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:MMEMory:LOAD:MEMory

Function Loads a specified waveform file into a specified memory.

Syntax `:MMEMory:LOAD:MEMory<wsp><integer>,<"file name">[,INTernal|EXTernal]`
<integer> = Number of the memory into which a file is loaded
<"file name"> = Name of file to be loaded
INTernal| EXTernal = Drive of source file to load

Example `:MMEMORY:LOAD:MEMORY 1,"test001.WX9"INTERNAL`

Explanation This is a sequential command.

:MMEMory:LOAD:PROGram

Function Loads a specified program file into a specified program number.

Syntax `:MMEMory:LOAD:PROGram<wsp><integer>,<"file name">[,INTernal|EXTernal]`
<trace name> = Number of the program into which a file is loaded
<"file name"> = Name of a file to be loaded
INTernal|EXTernal = Drive of source file to be loaded

Example `MMEMORY:LOAD:PROGRAM 1,"test001.PG9",INTERNAL`

Explanation This is a sequential command.

:MMEMory:LOAD:SETTing

Function Loads a specified setting file.

Syntax `:MMEMory:LOAD:SETTing<wsp><"file name">[,INTernal|EXTernal]`
<"file name"> = Name of a file to be loaded
INTernal|LOPpy = Drive of source file to be loaded

Example `MMEMORY:LOAD:SETTING "test001.SX9",INTERNAL`

Explanation This is a sequential command.

:MMEMory:LOAD:TEMPlate

Function Loads a specified template file.

Syntax `:MMEMory:LOAD:TEMPlate<wsp><template>,<"file name">[,INTernal|EXTernal]`
<template> = Template at the loading destination (UPPER|LOWER|TARGET)
<"file name"> = Name of a file to be loaded
INTernal|EXTernal = Drive at the loading source

Example `:MMEMORY:LOAD:SETTING UPPER,"test001.csv",INTERNAL`

Explanation This is a sequential command.

:MMEMory:LOAD:TRACe

Function Loads a specified waveform file into a specified trace.

Syntax `:MMEMory:LOAD:TRACe<wsp><trace name>,<"file name">[,INTernal|EXTernal]`
<trace name> = Trace to be loaded
<"file name"> = Name of file to be loaded
INTernal|EXTernal = Drive of source file to load

Example `:MMEMORY:LOAD:TRACE TRA,"test001.WX9",INTERNAL`

Explanation This is a sequential command.

:MMEMory:MDIRectory

Function Creates a new directory.

Syntax `:MMEMory:MDIRectory<wsp><"directory name">[,INTernal|EXTernal]`
<directory name> = Directory name to be created
INTernal|EXTernal = Destination drive for created directory

Example `:MMEMORY:MDIRECTORY "sample2",INTERNAL`

Explanation This is a sequential command.

:MMEMory:REMove

Function Readies the USB storage media for removal or queries the readiness status.

Syntax `:MMEMory:REMove`
`:MMEMory:REMove?`
Response 0 = Ready for removal
1 = Not ready

Example `:MMEMORY:REMOVE`
`:MMEMORY:REMOVE -> 1`

:MMEMory:REName

Function Renames a specified file.

Syntax `:MMEMory:REName<wsp><"new file name">,<"old file name">[,INTernal|EXTernal]`
<"new file name">= Name of new file
<"old file name">= Name of old file
INTernal|EXTernal = Target drive

Example `:MMEMORY:RENAME "test001.WX9", "test002.WX9",INTERNAL`

Explanation This is a sequential command.

:MMEMory:STORe:ARESult

Function Stores a variety of analysis results to a specified file.

Syntax `:MMEMory:STORe:ARESult<wsp><"file name">[,INTernal|EXTernal]`
<"file name"> = Name of a file to be saved
INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:ARESULT "test001",INTERNAL`

Explanation This is a sequential command.

:MMEMory:STORe:ATRace

Function Stores the specified waveform files (all traces) into traces.

Syntax `:MMEMory:STORe:ATRace<wsp>`
<”file name”> [,INTernal|EXTernal]
<"file name"> = Name of file be saved
INTernal|EXTernal = Save destination drive

Example: `MMEMORY:STORE:ATRACE "test001.csv",internal`

Explanation This is a sequential command.

:MMEMory:STORe:DATA

Function Stores a variety of data to a specified file.

Syntax `:MMEMory:STORe:DATA<wsp><"file name">[,INTernal|EXTernal]`
<"file name"> = Name of a file to be saved
INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:DATA "test001",INTERNAL`

Explanation
- The type of data to be stored is specified using the `:MMEMory:STORe:DATA:ITEM` command.
- Whether to insert data into or overwrite the file with it when storing it is specified using the `:MMEMory:STORe:DATA:MODE` command.
- This is a sequential command.

:MMEMory:STORe:DATA:ITEM

Function Sets/queries an item to be used when storing data.

Syntax `:MMEMory:STORe:DATA:ITEM<wsp><item>,OFF|ON|0|1`
`:MMEMory:STORe:DATA:ITEM?<wsp><item>`
<item> DATE = Date/time at the time of storage
LABel = Label
DATA = DATA area data
CONDition = Setting conditions
OWINdow= OUTPUT WINDOW
TRACe = Waveform data
OFF = Do not save
ON = Save
Response 0 = OFF, 1 = ON

Example `:MMEMORY:STORE:DATA:ITEM TRACE,OFF`
`:MMEMORY:STORE:DATA:ITEM? TRACE -> 0`

Explanation This is a sequential command.

:MMEMory:STORe:DATA:MODE

Function Sets whether to insert data into or overwrite an existing file with the data when storing it or queries the condition of whether data is inserted or overwritten.

Syntax `:MMEMory:STORe:DATA:MODE<wsp>ADD|OVER|0|1`
`:MMEMory:STORe:DATA:MODE?`
ADD = Insert mode
OVER = Overwrite mode
Response 0 = ADD, 1 = OVER

Example `:MMEMORY:STORE:DATA:MODE OVER`
`:MMEMORY:STORE:DATA:MODE? -> 1`

Explanation This is a sequential command.

:MMEMory:STORe:DATA:TYPE

Function Sets/queries a file format to be used when storing data.

Syntax `:MMEMory:STORe:DATA:TYPE<wsp>CSV|DT|0|1`
`:MMEMory:STORe:DATA:TYPE?`
CSV = CSV storage format
DT = DT9 storage format
Response 0 = CSV, 1 = DT9

Example `:MMEMORY:STORE:DATA:TYPE DT9`
`:MMEMORY:STORE:DATA:TYPE? -> 1`

Explanation This is a sequential command.

:MMEMory:STORe:DLOGging

Function Saves the data logging results to a specified file.

Syntax :MMEMory:STORe:DLOGging<wsp><"file name">[,INTernal|EXTernal]
<"file name"> = Name of the file to save to
INTernal|EXTernal = Drive to save to

Example :MMEMORY:STORE:DLOGGING "test001",INTERNAL

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:MMEMory:STORe:DLOGging:CSAVe

Function Sets or queries whether data logging results will be saved to a file in CSV format.

Syntax `:MMEMory:STORe:DLOGging:CSAVe<wsp>OFF|ON|0|1`
`:MMEMory:STORe:DLOGging:CSAVe?`
OFF = Data will not be saved to CSV format.
ON = Data will be saved to CSV format.
Response 0 = Off, 1 = On

Example `:MMEMORY:STORE:DLOGGING:CSAVE ON`
`:MMEMORY:STORE:DLOGGING:CSAVE? -> 1`

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:MMEMory:STORe:DLOGging:TSAVe

Function Sets or queries whether temporary saved waveform files will be saved when data logging results is saved.

Syntax `:MMEMory:STORe:DLOGging:TSAVe<wsp>OFF|ON|0|1`
`:MMEMory:STORe:DLOGging:TSAVe?`
OFF: Will not be saved
ON: Will be saved
Response 0 = Off, 1 = On

Example `:MMEMORY:STORE:DLOGGING:TSAVE ON`
`:MMEMORY:STORE:DLOGGING:TSAVE? -> 1`

Description
- This command is invalid when data logging is in progress.
- This is a sequential command.

:MMEMory:STORe:GRAPhics

Function Stores a waveform screen to a specified graphic file.

Syntax `:MMEMory:STORe:GRAPhics<wsp>B&W|COLor|PCOLor,BMP|TIFF,<"file name">[,INTernal| EXTernal]`
B&W|COLor PCOLor = Color mode when saving
B&W = Black and white mode
COLor = Color mode
PCOLor = Preset color (waveforms in color, background in black & white)
BMP|TIFF = Saved format
BMP = BMP format
TIFF = TIFF format
<"file name"> = Name of a file to be saved
INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:GRAPHICS COLOR,BMP,"test001",INTERNAL`

Explanation
- This is a sequential command.

:MMEMory:STORe:MEMory

Function Stores a specified memory to a specified waveform file.

Syntax `:MMEMory:STORe:MEMory<wsp><integer>,BIN|CSV,<"file name">[,INTernal|EXTernal]`
<integer> = Number of a memory whose contents are stored
BIN|CSV = Sav format
BIN = Binary format
CSV = Text format
<"file name"> = Name of file to be saved
INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:MEMORY 1,CSV,"test001",INTERNAL`

Explanation This is a sequential command.

:MMEMory:STORe:PROGram

Function Stores a specified program to a specified file.

Syntax `:MMEMory:STORe:PROGram<wsp><integer>,<"file name">[,INTernal|EXTernal]`
<integer> = Number of a program whose contents are stored
<"file name"> = Name of a file to be saved
NTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:PRORAM 1,"test001",INTERNAL`

Explanation This is a sequential command.

:MMEMory:STORe:SETTing

Function Stores setting information to a specified file.

Syntax `:MMEMory:STORe:SETTing<wsp><"file name">[,INTernal|EXTernal]`

<"file name"> = Name of a file to be saved

INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:SETTING "test001", INTERNAL`

Explanation This is a sequential command.

:MMEMory:STORe:TEMPlate

Function Stores specified template data to a specified file

Syntax `:MMEMory:STORe:TEMPlate <wsp><template>,<"file name">[,INTernal|EXTernal]`

<template> = Template to be saved. (UPPER|LOWER|TARGET)

<"file name"> = Name of a file to be saved

INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:TEMPLATE UPPER, "test001",INTERNAL`

Explanation This is a sequential command.

:MMEMory:STORe:TRACe

Function Stores a specified trace to a specified waveform file.

Syntax `:MMEMory:STORe:TRACe<wsp><trace name>,BIN|CSV,<"file name">[,INTernal|EXTernal]`

<trace name> = Trace to be saved

BIN|CSV = Save format

BIN = Binary format

CSV = Text format)

<"file name"> = Name of file to be saved

INTernal|EXTernal = Save destination drive

Example `:MMEMORY:STORE:TRACE TRA,CSV, "test001",INTERNAL`

Explanation This is a sequential command.

PROGram Sub System Command

:PROGram:EXECute

Function This key is used to execute a program that has been specified.

Syntax `:PROGram:EXECute<wsp><integer>`

<integer> = Number of a program to execute

Example `:PROGRAM:EXECUTE 1`

Explanation This is an overlapable command.

SENSe Sub System Command

:SENSe:AVERage:COUNt

Function: Sets/queries the number of times averaging for each measured point.

Syntax:
```
:SENSe:AVERage:COUNt<wsp><integer>
:SENSe:AVERage:COUNt?
```
<integer> = Number of times averaging

Example:
```
:SENSE:AVERAGE:COUNT 100
:SENSE:AVERAGE:COUNT? -> 100
```

Explanation: This is a sequential command.

:SENSe:BANDwidth|:BWIDth[:RESolution]

Function: Sets/queries the measurment resolution.

Syntax:
```
:SENSe:BANDwidth|:BWIDth[:RESolution]<wsp><NRf>[M|Hz]
:SENSe:BANDwidth|:BWIDth
[:RESolution]?
```
<NRf> = Measurement resolution [m|Hz]

Response
<NRf>[m|Hz]

Example:
```
:SENSE:BANDWIDTH:RESOLUTION 100PM
:SENSE:BANDWIDTH? -> +1.00000000E-010
```

Explanation:
- This is a sequential command.

:SENSe:CHOPper

Function: Sets/queries chopper mode.

Syntax:
```
:SENSe:CHOPper<wsp>OFF|SWITch|0|2
:SENSe:CHOPper?
```
Response 0 = OFF,
2 = SWITCH

Example:
```
:SENSE:CHOPPER SWITCH
:SENSE:CHOPPER? -> 2
```

Explanation:
- When the measurement sensitivity setting (`:SENSe:SENSe` command) is NORMAL HOLD, NORMAL AUTO or NORMAL, Chopper does not function even if chopper mode is turned on with this command.
- This is a sequential command.

:SENSe:CORRection:LEVel:SHIFt

Function: Sets/queries the offset value for the level.

Syntax:
```
:SENSe:CORRection:LEVel:SHIFt<wsp><NRf>[DB]
:SENSe:CORRection:LEVel:SHIFt?
```
<NRf> = Level offset value [dB]

Example:
```
:SENSE:CORRECTION:LEVEL:SHIFT 0.2DB
:SENSE:CORRECTION:LEVEL:SHIFT?->
+2.00000000E-001
```

Explanation: This is a sequential command.

:SENSe:CORRection:RVELocity:MEDium

Function: Sets/queries whether air or vacuum is used as the wavelength reference.

Syntax:
```
:SENSe:CORRection:RVELocity:MEDium
<wsp>AIR|VACuum|0|1
:SENSe:CORRection:RVELocity:MEDium?
```
AIR = Air is assumed to be the reference.
VACuum = Vacuum is assumed to be the reference.

Response 0 = AIR
1 = VACuum

Example:
```
:SENSE:CORRECTION:RVELOCITY:
MEDIUM VACUUM
:SENSE:CORRECTION:RVELOCITY:MEDIUM?-> 1
```

Explanation: This is a sequential command.

:SENSe:CORRection:WAVelength:SHIFt

Function: Sets/queries the offset value for the levelwavelength.

Syntax:
```
:SENSe:CORRection:WAVelength:SHIFt<wsp><NRf>[M]
:SENSe:CORRection:WAVelength:SHIFt?
```
<NRf>= Wavelength offset value [m]

Example:
```
:SENSE:CORRECTION:WANELENGTH:
SHIFT 0.05NM
:SENSE:CORRECTION:WANELENGTH:SHIFT?-> +5.00000000E-011
```

Explanation: This is a sequential command.

:SENSe:SENSe

Function: Sets/queries the measurement sensitivity.

Syntax:
```
:SENSe:SENSe<wsp><sense>
:SENSe:SENSe?
```
<sense>= Sensitivity setting parameters
NHLD = NORMAL HOLD
NAUT = NORMAL AUTO
NORMal = NORMAL
MID = MID
HIGH1 = HIGH1
HIGH2 = HIGH2
HIGH3 = HIGH3

Response 0 = NHLD
1 = NAUT
2 = MID
3 = HIGH1
4 = HIGH2
5 = HIGH3
6 = NORMAL

Example:
```
:SENSE:SENSE MID
:SENSE:SENSE? -> 2
```

Explanation: This is a sequential command.

:SENSe:SETTing:FIBer

Function Sets/queries the fiber core size mode.

Syntax `:SENSe:SETTing:FIBer<wsp>SMALl|LARGe|0|1`
`:SENSe:SETTing:FIBer?`
SMALl= Standard mode
LARGe= Large core size fiber mode
Response 0 = SMALl, 1 = LARGe

Example `:SENSE:SETTING:FIBER LARGE`
`:SENSE:SETTING:FIBER? -> 1`

Explanation • This is a sequential command.

:SENSe:SETTing:SMOothing

Function Sets/queries the Smoothing function.

Syntax `:SENSe:SETTing:SMOothing<wsp>OFF|ON|0|1`
`:SENSe:SETTing:SMOothing?`
Response 0 = OFF, 1 = ON

Example `:SENSE:SETTING:SMOothing ON`
`:SENSE:SETTING:SMOothing? -> 1`

Explanation • This is a sequential command.

:SENSe:SWEep:POINts

Function Sets/queries the number of samples measured.

Syntax `:SENSe:SWEep:POINts<wsp><integer>`
`:SENSe:SWEep:POINts?`
<integer> = The number of samples to be measured

Example `:SENSE:SWEEP:POINTS 20001`
`:SENSE:SWEEP:POINTS? -> 20001`

Explanation
- When the function of automatically setting the sampling number to be measured (`SENSe:SWEep:POINts:AUTO` command) is ON, the sampling number to be measured that has been set can be queried.
- When the function of automatically setting the sampling number to be measured (`SENSe:SWEep:POINts:AUTO` command) is ON, this command will be automatically set to OFF.
- When the sampling number to be measured is set using this command, the sampling intervals for measurements (`SENSe:SWEep:STEP`) will be automatically set.
- This is a sequential command.

:SENSe:SWEep:POINts:AUTO

Function Sets/queries the function of automatically setting the sampling number to be measured.

Syntax `:SENSe:SWEep:POINts:AUTO<wsp>OFF|ON|0|1`
`:SENSe:SWEep:POINts:AUTO?`
Response 0 = OFF, 1 = ON

Example `:SENSE:SWEEP:POINTS:AUTO ON`
`:SENSE:SWEEP:POINTS:AUTO? -> 1`

Explanation
- When the capability to automatically set the sampling number to be measured is set to ON using this command, the sampling number to be measured and the sampling intervals for measurements (`SENSe:SWEep:STEP`) will be automatically set.
- This is a sequential command.

:SENSe:SWEep:SEGMent:POINts

Function Sets/queries the number of sampling points to be measured at one time when performing SEGMENT MEASURE.

Syntax `:SENSe:SWEep:SEGMent:POINts<wsp><integer>`
`:SENSe:SWEep:SEGMent:POINts?`
<integer> = The number of samples measured

Example `:SENSE:SWEEP:SEGMENT:POINTS 100`
`:SENSE:SWEEP:SEGMENT:POINTS? -> 100`

Explanation This is a sequential command.

:SENSe:SWEep:SPEed

Function Sets/queries the sweep speed.

Syntax `:SENSe:SWEep:SPEed<wsp>1x|2x|0|1`
`:SENSe:SETTing:FCONnector?`
1x|0: Standard
2x|1: Twice as fast as standard
Response 0 = 1x, 1 = 2x

Example `:SENSE:SWEEP:SPEED 2x`
`:SENSE:SWEEP:SPEED? -> 1`

Explanation • This is a sequential command.

:SENSe:SWEep:STEP

Function Sets/queries the sampling interval for measurements.

Syntax `:SENSe:SWEep:STEP<wsp><NRf>[M]`
`:SENSe:SWEep:STEP?`
<NRf> = The sampling interval for measurement [m]

Example `:SENSE:SWEEP:STEP 1PM`
`:SENSE:SWEEP:STEP?-> +1.00000000E-012`

Explanation
- When the function of automatically setting the sampling interval for measurement (`SENSe:SWEep:POINts:AUTO` command) is ON, the sampling number to be measured that has been set can be queried.
- When the function of automatically setting the sampling number to be measured (`SENSe:SWEep:POINts:AUTO` command) is ON, this command will be automatically set to OFF.
- When the sampling interval for measurement is set using this command, the sampling intervals for measurements (`SENSe:SWEep:POINts`) will be automatically set.
- This is a sequential command.

:SENSe:SWEep:TIME:0NM

Function Sets/queries the time taken from the start to the end of measurements when measurement is made in the 0-nm sweep mode.

Syntax `:SENSe:SWEep:TIME:0NM<wsp><integer>[SEC]`
`:SENSe:SWEep:TIME:0NM?`
<integer> = Measurement time [sec] (0 = MINIMUM)

Example `:SENSE:SWEEP:TIME:0NM 10SEC`
`:SENSE:SWEEP:TIME:0NM? -> 10`

Explanation This is a sequential command.

:SENSe:SWEep:TIME:INTerval

Function Sets/queries the time taken from the start of a sweep to that of the next sweep when repeat sweeps are made.

Syntax `:SENSe:SWEep:TIME:INTerval<wsp><integer>[SEC]`
`:SENSe:SWEep:TIME:INTerval?`
<integer> = Measurement time [sec] (0 = MINIMUM)

Example `:SENSE:SWEEP:TIME:INTERVAL 100sec`
`:SENSE:SWEEP:TIME:INTERVAL? -> 100`

Explanation This is a sequential command.

:SENSe:WAVelength:CENTer

Function Sets/queries the measurement condition center wavelength.

Syntax `:SENSe:WAVelength:CENTer<wsp><NRf>[M|HZ]`
`:SENSe:WAVelength:CENTer?`
<NRf> = Measurement center wavelength [m]
Response
<NRf>`[m|Hz]`

Example `:SENSE:WAVELENGTH:CENTER 1550.000NM`
`:SENSE:WAVELENGTH:CENTER?-> +1.55000000E-006`

Explanation
- This is a sequential command.

:SENSe:WAVelength:SPAN

Function Sets/queries the measurement condition measurement span.

Syntax `:SENSe:WAVelengthSPAN<wsp><NRf>[M|HZ]`
`:SENSe:WAVelength:SPAN?`
<NRf> = Measurement span [m]
Response
<NRf>`[m|Hz]`

Example `:SENSE:WAVELENGTH:SPAN 20.0NM`
`:SENSE:WAVELENGTH:SPAN?-> +2.00000000E-008`

Explanation
- This is a sequential command.

:SENSe:WAVelength:SRANge

Function Sets/queries whether to limit a sweep range to the spacing between line markers L1 and L2.

Syntax `:SENSe:WAVelength:SRANge<wsp>OFF|ON|0|1`
`:SENSe:WAVelength:SRANge?`
Response 0 = OFF, 1 = ON

Example `:SENSE:WAVELENGTH:SRANGE ON`
`:SENSE:WAVELENGTH:SRANGE? -> 1`

Explanation This is a sequential command.

:SENSe:WAVelength:STARt

Function Sets/queries the measurement condition measurement start wavelength.

Syntax `:SENSe:WAVelength:STARt<wsp><NRf>[M|HZ]`
`:SENSe:WAVelength:STARt?`
<NRf>=Measurement center wavelength [m]
Response
<NRf>`[m|Hz]`

Example `:SENSE:WAVELENGTH:START 1540.000NM`
`:SENSE:WAVELENGTH:START?-> +1.54000000E-006`

Explanation
- This is a sequential command.

:SENSe:WAVelength:STOP

Function Sets/queries the measurement condition measurement stop wavelength.

Syntax `:SENSe:WAVelengthSTOP<wsp><NRf>[M|HZ]`
`:SENSe:WAVelength:STOP?`
<NRf> = Measurement stop wavelength [m]
Response
<NRf>`[m|Hz]`

Example `:SENSE:WAVELENGTH:STOP 1560.000NM`
`:SENSE:WAVELENGTH:STOP?-> +1.56000000E-006`

Explanation
- This is a sequential command.

STATus Sub System Command

:STATus:OPERation:CONDition?

Function Queries the contents of the operation status condition register.

Syntax `:STATus:OPERation:CONDiton?`

Example `:STATUS:OPERATION:CONDITION? -> 1`

Explanation This is a sequential command.

:STATus:OPERation:ENABle

Function Queries the contents of the operation status Enable register.

Syntax `:STATus:OPERation:ENABle<wsp><integer>`
`:STATus:OPERation:ENABle?`
<integer> = Contents of the operation status enable register

Example `:STATUS:OPERATION:ENABLE 8`
`:STATUS:OPERATION:ENABLE? -> 8`

Explanation This is a sequential command.

:STATus:OPERation[:EVENt]?

Function Queries the contents of the operation status Event register.

Syntax `:STATus:OPERation[:EVENt]?`

Example `:STATUS:OPERATION? -> 1`

Explanation This is a sequential command.

:STATus:PRESet

Function Clears the event register and sets all bits of the enable register.

Syntax `:STATus:PRESet`

Example `:STATUS:PRESET`

Explanation
- When this command is executed, the registers will be affected as follows.
- The operation status event register is cleared to “0.”
- All bits of the operation status enable register are set to “0.”
- The questionable status event register is cleared to “0.”
- All bits of the questionable status enable register are set to “0.”
- Even when this command is executed, the standard event status register and standard event status enable register do not change.
- This is a sequential command.

:STATus:QUEStionable:CONDition?

Function Queries the contents of the qestionable status condition register.

Syntax `:STATus:QUEStionable:CONDiton?`

Example `:STATUS:QUESTIONABLE:CONDITION? -> 1`

Explanation This is a sequential command.

:STATus:QUEStionable:ENABle

Function Reads the contents of the questionable status enable register or writes data to this register.

Syntax `:STATus:QUEStionable:ENABle<wsp><integer>`
`:STATus:QUEStionable:ENABle?`
<integer> = Contents of the questionable status enable register

Example `:STATUS:QUESTIONABLE:ENABLE 8`
`:STATUS:QUESTIONABLE:ENABLE? -> 8`

Explanation This is a sequential command.

:STATus:QUEStionable[:EVENt]?

Function Reads the contents of the questionable status event register.

Syntax `:STATus:QUEStionable[:EVENt]?`

Example `:STATUS:QUESTIONABLE:? -> 1`

Explanation This is a sequential command.

SYStem Sub System Command

:SYSTem:BUZZer:CLICk

Function Sets/queries whether to sound the buzzer when clicked the key.

Syntax `:SYSTem:BUZZer:CLICk<wsp>OFF|ON|0|1`
`:SYSTem:BUZZer:CLICk?`
Response 0 = OFF, 1 = ON

Example `:SYSTEM:BUZZER:CLICK ONn`
`:SYSTEM:BUZZER:CLICK? -> 1`

Explanation This is a sequential command.

:SYSTem:BUZZer:WARNing

Function Sets/queries whether to sound the buzzer during an alarm.

Syntax `:SYSTem:BUZZer:WARNing<wsp>OFF|ON|0|1`
`:SYSTem:BUZZer:WARNing?`
Response 0 = OFF, 1 = ON

Example `:SYSTEM:BUZZER:WARNING ON`
`:SYSTEM:BUZZER:WARNING? -> 1`

Explanation This is a sequential command.

:SYSTem:COMMunicate:CFORmat

Function Sets/queries the GP-IB command format of this unit.

Syntax
```
:SYSTem:COMMunicate:CFORmat<wsp>
<mode>
:SYSTem: COMMunicate:CFORmat?
```
<mode> = GP-IB command format
AQ6317 = AQ6317 compatible mode
AQ6374 = AQ6374 mode
Response 0 = AQ6317, 1 = AQ6374

Example
```
:SYSTEM:COMMUNICATE:CFORMAT AQ6370C
syst:comm:cformat? -> 1
```

Explanation
- This command is valid when in AQ6374 mode. This command results in an error when in AQ6317 compatible mode.
- To set the GP-IB command format while this unit is in the AQ6317-compatible mode, use the following commands.
  Control command
  CFORM* (*: 0 = AQ6317 compatible mode, 1 = AQ6374 mode)
  Query command
  CFORM? (return value: 0 = AQ6317-compatible mode, 1 = AQ6374 mode)
- To use a GP-IB command to place this unit into the AQ6317-compatible mode, regardless of the status during execution of the command, execute the following command. Note that if this unit has already been in the AQ6317-compatible mode at the time of executing this command, a command error occurs, but you can ignore it.
  ```
  :SYSTem:COMMunicate:CFORmat<wsp>
  AQ6317
  ```
- To use a GP-IB command to place this unit into the AQ6374 mode, regardless of the status during execution of the command, execute the following command. Note that if this unit has already been in the AQ6374 mode at the time of executing this command, a command error occurs, but you can ignore it.
  ```
  CFORM1
  ```
- This is a sequential command.

:SYSTem:COMMunicate:LOCKout

Function Sets/cancels local lockout.

Syntax
```
:SYSTem:COMMunicate:LOCKout<wsp>
OFF|ON|0|1
:SYSTem:COMMunicate:LOCKout?
```
OFF|0: Cancels local lockout
ON|1: Sets local lockout

Example
```
:SYSTEM:COMMUNICATE:LOCKOUT OFF
:SYSTEM:COMMUNICATE:LOCKOUT? -> 0
```

Explanation
- This command is valid when the remote interface is the Ethernet interface. An interface message is available for the GP-IB interface.
- During local lockout, if the Ethernet connection is lost, the instrument switches to local mode, regardless of the local lockout status.
- This is a sequential command.

:SYSTem:COMMunicate:RMONitor

Function Sets/queries whether the remote monitor function is enabled.

Syntax
```
:SYSTem:COMMunicate:RMONitor<wsp>
OFF|ON|0|1
:SYSTem:COMMunicate:RMONitor?
```
OFF|0: Disables the remote monitor function
ON|1: Enables the remote monitor function

Example
```
:SYSTEM:COMMUNICATE:RMONITOR OFF
:SYSTEM:COMMUNICATE:RMONITOR? -> 0
```

Explanation
- This is a sequential command.

:SYSTem:DATE

Function Sets/queries the system data.

Syntax
```
:SYSTem:DATE<wsp><year>,<month>,<d
ay>
:SYSTem:DATE?
```
<year> = Year
<month> = Month
<day> = Day

Example
```
:SYSTEM:DATE 2017,01,01
:SYSTEM:DATE? -> 2017,01,01
```

Explanation This is a sequential command.

:SYSTem:DISPlay:TRANsparent

Function Sets/queries whether to make the Interrupt Window and OVERVIEW Window of the measurement screen semi-transparent.

Syntax
```
:SYSTem:DISPlay:TRANsparent<wsp>OFF
|ON|0|1
:SYSTem:DISPlay:TRANsparent?
```
Response 0 = OFF, 1 = ON

Example
```
:SYSTEM:DISPLAY:TRANSPARENT OFF
:SYSTEM:DISPLAY:TRANSPARENT? -> 0
```

Explanation This is a sequential command.

:SYSTem:DISPlay:UNCal

Function Sets/queries whether to display an alarm message in the event of UNCAL.

Syntax
```
:SYSTem:DISPlay:UNCal<wsp>OFF|
ON|0|1
:SYSTem:DISPlay:UNCal?
```
Response 0 = OFF, 1 = ON

Example
```
:SYSTEM:DISPLAY:UNCAL OFF
:SYSTEM:DISPLAY:UNCAL? -> 0
```

Explanation This is a sequential command.

:SYSTem:ERRor[:NEXT]?

Function Queries data in an error queue and deletes it from the queue.

Syntax
```
:SYSTem:ERRor[:NEXT]?
```
<integer> = Error number

Example
```
:SYSTEM:ERROR? -> 100
```

Explanation This is a sequential command.

:SYSTem:GRID

Function Sets/queries the instrument's grid setting.

Syntax `:SYSTem:GRID<wsp><grid>`
`:SYSTem:GRID?`
<grid> = Grid setting
12.5 GHZ = 12.5 GHz Spacing
25 GHZ = 25 GHz Spacing
50 GHZ = 50 GHz Spacing
100 GHZ = 100 GHz Spacing
200 GHZ = 200 GHz Spacing
CUSTom = User setting
Response 0 = 12.5GHz
1 = 25GHz
2 = 50GHz
3 = 100GHz
4 = 200GHz
5 = CUSTom

Example `:SYSTEM:GRID 50GHZ`
`:SYSTEM:GRID? -> 2`

Explanation
- This is a sequential command.

:SYSTem:GRID:CUSTom:CLEar:ALL

Function Clears the user-specified custom grid and returns it to the default value.

Syntax `:SYSTem:GRID:CUSTom:CLEar:ALL`

Example `:SYSTem:GRID:CUSTOM:CLEAR:ALL`

Explanation
- This is a sequential command.

:SYSTem:GRID:CUSTom:DELete

Function Deletes the specified grid of the custom grid.

Syntax `:SYSTem:GRID:CUSTom:DELete<wsp><integer>`
<integer> = Number of a grid to be deleted

Example `:SYSTem:GRID:CUSTOM:DELETE 10`

Explanation
- This is a sequential command.

:SYSTem:GRID:CUSTom:INSert

Function Inserts a new grid when the grid setting is in the custom grid.

Syntax `:SYSTem:GRID:CUSTom:INSert<wsp><NRf>[M|HZ]`
<NRf> = Grid wavelength/frequency to be inserted [m| Hz]

Example `:SYSTem:GRID:CUSTOM:INSERT 1550.123NM`

Explanation
- When `:SYSTem:GRID` is CUSTom, an execution error occurs.
- This is a sequential command.

:SYSTem:GRID:CUSTom:SPACing

Function Sets/queries the grid spacing of the custom grid.

Syntax `:SYSTem:GRID:CUSTom:SPACing<wsp><NRf>[GHZ]`
`:SYSTem:GRID:CUSTom:SPACing?`
<NRf> = Grid spacing [GHz]

Example `:SYSTem:GRID:CUSTOM:SPACING 12.5`
`:SYSTem:GRID:CUSTOM:SPACING?->`
`+1.25000000E+001`

Explanation
- When `:SYSTem:GRID` is CUSTom, an execution error occurs.
- This is a sequential command.

:SYSTem:GRID:CUSTom:STARt

Function Sets/queries the custom grid start wavelength.

Syntax `:SYSTem:GRID:CUSTom:STARt<wsp><NRf>[M|HZ]`
`:SYSTem:GRID:CUSTom:STARt?`
<NRf> = Grid start wavelength [m|Hz]

Example `:SYSTem:GRID:CUSTOM:START 1550.000NM`
`:SYSTem:GRID:CUSTOM:START?->`
`+1.55000000E-006`

Explanation
- When `:SYSTem:GRID` is CUSTom, an execution error occurs.
- This is a sequential command.

:SYSTem:GRID:CUSTom:STOP

Function Sets/queries the custom grid stop wavelength.

Syntax `:SYSTem:GRID:CUSTom:STOP<wsp><NRf>[M|HZ]`
`:SYSTem:GRID:CUSTom:STOP?`
<NRf> = Grid stop wavelength [m|Hz]

Example `:SYSTEM:GRID:CUSTOM:STOP 1560.000NM`
`:SYSTEM:GRID:CUSTOM:STOP?->`
`+1.56000000E-006`

Explanation
- When `:SYSTem:GRID` is something other than CUSTom, an execution error occurs.
- This is a sequential command.

:SYSTem:GRID:REFerence

Function Sets/queries the reference frequency of the instrument's grid setting.

Syntax `:SYSTem:GRID:REFerence<wsp><NRf>[HZ]`
`:SYSTem:GRID:REFerence?`
<NRf> = Grid's reference frequency [Hz]

Example `:SYSTEM:GRID:REFERENCE 193.1000HZ`
`:SYSTEM:GRID:REFERENCE ?->`
`+1.93000000E+014`

Explanation
- This is a sequential command.

:SYSTem:INFormation?

Function Queries model-specific information (the model code and special code)

Syntax `:SYSTem:INFormation?<wsp><integer>`
<integer> = The type of instrument specific information to obtain
0 = MODEL code
1 = SPECIAL code

Example `:SYSTEM:INFORMATION? 0 -> AQ6374-D/FC/RFC`
`Example:SYSTEM:INFORMATION? 1 -> AQ6370C-M/`

Explanation
- Outputs model-specific information (the model code and special code)
- If no SPECIAL mode is present, "NONE" is returned.
- This is a sequential command.

:SYSTem:OLOCk

Function Sets or queries whether keys are locked.

Syntax `:SYSTem:OLOCk<wsp>OFF|ON|0|1,<"password">`
`:SYSTem:OLOCk?`
OFF = Not locked (release the lock)
ON = Locked
<"password"> = 4-digit password string
The characters that can be used are numbers from 0 to 9.
Response 0=OFF, 1=ON

Example `:SYST:OLOC ON,"1234"`
`:SYST:OLOC? -> 1`

Description
- This is a sequential command.

:SYSTem:OPERation:SHUTdown

Function Shuts down the AQ6374.

Syntax `:SYSTem:OPERation:SHUTdown`

Example `:SYSTEM:OPERATION:SHUTDOWN`

Description
- This is a sequential command.

:SYSTem:OPERation:REBoot

Function Restarts the AQ6374.

Syntax `:SYSTem:OPERation:REBoot`

Example `:SYSTEM:OPERATION:REBOOT`

Description
- This is a sequential command.

:SYSTem:PRESet

Function Initializes the unit status.

Syntax `:SYSTem:PRESet`

Example `:SYSTEM:PRESET`

Explanation This is a sequential command.

:SYSTem:TIME

Function Sets/queries the system time.

Syntax `:SYSTem:TIME<wsp><hour>,<minute>,<second>`
`:SYSTem:TIME?`
<hour> = Hour
<minute> = Minute
<second> = Second

Example `:SYSTEM:TIME 22,10,01`
`:SYSTEM:TIME? -> 22,10,1`

Explanation This is a sequential command.

:SYSTem:VERSion?

Function Queries the SCPI compatibility version of this unit.

Syntax `:SYSTem:VERSion?`

Example `:SYSTEM:VERSION? -> 1999.0`

Explanation This is a sequential command.

TRACe Sub System Command

:TRACe:ACTive

Function Sets/queries the active trace.

Syntax
```
:TRACe:ACTive<wsp><trace name>
:TRACe:ACTive?
```
<trace name> = Active trace (TRA|TRB|TRC|TRD|TRE|TRF|TRG)

Example
```
:TRACE:ACTIVE TRA
:TRACE:ACTIVE? -> TRA
```

Explanation This is a sequential command.

:TRACe:ATTRibute[:<trace name>]

Function Sets/queries the attributes of the specified trace.

Syntax
```
:TRACe:ATTRibute[:<trace name>]
<wsp><attribute>
:TRACe:ATTRibute[:<trace name>]?
```
<trace name> = trace (TRA|TRB|TRC|TRD|TRE|TRF|TRG)

<attribute> = Attribute
WRITe = WRITE
FIX = FIX
MAX = MAX HOLD
MIN = MIN HOLD
RAVG = ROLL AVG
CALC = CALC

Response 0 = WRITe
1 = FIX
2 = MAX
3 = MIN
4 = RAVG
5 = CALC

Example
```
:TRACE:ATTRIBUTE:TRA WRITE
:TRACE:ATTRIBUTE:TRA? -> 0
```

Explanation
- If <trace name> is omitted, the command is executed with respect to the active trace.
- If <trace name> is specified, the specified trace is set as the active trace after the command is exeucted.
- When the attribute is set to a CALC trace, the expression is set using the `:CALCulate:MATH` command.
- This is a sequential command.

:TRACe:ATTRibute:RAVG[:<trace name>]

Function Sets/queries the number of times for averaging of the specified trace.

Syntax
```
:TRACe:ATTRibute:RAVG[:<trace
name>]<wsp><integer>
:TRACe:ATTRibute:RAVG[:<trace
name>]?
```
<trace name> = trace (TRA|TRB|TRC|TRD|TRE|TRF|TRG)

<integer> = Number of times averaging of ROLL AVG

Example
```
:TRACE:ATTRIBUTE:RAVG:TRA 10
:TRACE:ATTRIBUTE:RAVG:TRA? -> 10
```

Explanation
- When this command is executed, the attribute of the set trace goes to ROLL AVG.
- If <trace name> is omitted, the command is executed with respect to the active trace.
- If <trace name> is specified, the specified trace is set as the active trace after the command is exeucted.
- This is a sequential command.

:TRACe:COPY

Function Copies the data of a specified trace to another trace.

Syntax
```
:TRACe:COPY<wsp><source trace
name>,<destination trace name>
```
<source trace name> = Copy source trace

<destination trace name> = Copy trace destination

Example
```
:TRACE:COPY TRA,TRB
```

Explanation This is a sequential command.

:TRACe[:DATA]:SNUMber?

Function Sets/queries the number of number of data sampled of the specified trace.

Syntax
```
:TRACe[:DATA]:SNUMber?<wsp><trace
name>
```
<trace name> = Trace from which to acquire data

Example
```
:TRACE:DATA:SNUMBER? -> 50001
```

Explanation
- If a specified trace has no data, “0” is returned.
- This is a sequential command.

:TRACe[:DATA]:X?

Function Queries the wavelength axis data of the specified trace.

Syntax `:TRACe[:DATA]:X?<wsp><trace name> [,<start point>,<stop point>]`

<trace name>= Trace to be transferred (TRA|TRB|TRC|TRD|TRE|TRF|TRG)

<start point>= A range of samples to be transferred (starting point) (1 to 100001)

<stop point> = A range of samples to be transferred (stopping point) (1 to 100001)

Example `:TRACE:X? TRA -> +1.55000000E-006,+1.55001000E-006,+1.55002000E-006,.....`

Explanation
- Data is output in the unit of wavelength value (m), regardless of whether this unit is in the wavelength mode or in the frequency mode.
- If the parameter <start point> or <stop point> is omitted, all sampling data of a specified trace will be output.
  The number of output data can be acquired by executing `:TRACe[:DATA]:SNUMber?`.
- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:TRACe[:DATA]:Y?

Function Queries the level axis data of specified trace.

Syntax `:TRACe[:DATA]:Y?<wsp><trace name> [,<start point>,<stop point>]`

<trace name> = Trace to be transferred (TRA|TRB|TRC|TRD|TRE|TRF|TRG)

<start point> = A range of samples to be transferred (starting point) (1 to 100001)

<stop point> = A range of samples to be transferred (stopping point) (1 to 100001)

Response For ASCII data: <NRf>,<NRf>,.....<NRf>

For BINARY data: '#'<integer><byte num><data byte>

Example `:TRACE:Y? TRA -> -1.00000000E+001, -1.00000000E+001, -1.00000000E+001,....`

Explanation
- The data is output in order of its wavelength from the shortest level to the longest, irrespective of the wavelength/frequency mode.
- When the level scale is LOG, data is output in LOG values.
- When the level scale is Linear, data is output in linear values.
- If the parameter <start point> or <stop point> is omitted, all sampling data of a specified trace will be output.
  The number of output data can be acquired by executing `:TRACe[:DATA]:SNUMber?`.
- Data is output in either ASCII or binary form, depending on the setting of `:FORMat[:DATA]`.
- This is a sequential command.

:TRACe[:DATA]:Y:PDENsity?

Function Queries the power spectral density trace data.

Syntax `:TRACe[:DATA]:Y:PDENsity? <wsp> <trace name>,<NRf>[m][, <start point>,<stop point>`

<trace name> = Computation source trace

<NRF> = Normalization bandwidth [m]

<start point> = Sample range to transfer (start point) (1 to 100001)

<stop point> = Sample range to transfer (stop point) (1 to 100001)

Example `:trac:y:pden? tra,0.1nm -> -5.36017335E+001, -5.36143380E+001, -5.34441639E+001,....`

Description
- When the level scale is set to LOG, LOG values will be output.
- When the level scale is set to linear, linear values will be output.
- If the <start point> and <stop point> parameters are omitted, the entire sample data of the specified trace will be output.
- The data is output in ASCII or BINARY format according to the :FORMat[:DATA] setting.
- This is a sequential command.

:TRACe:DELete

Function Deletes the data of a specified trace.

Syntax `:TRACe:DELete<wsp><trace name>`

<trace name> = Trace to be transferred (TRA|TRB|TRC|TRD|TRE|TRF|TRG)

Example `:TRACE:DELETE TRA`

Explanation This is a sequential command.

:TRACe:DELete:ALL

Function Clears the data for all traces.

Syntax `:TRACe:DELete:ALL`

Example `:TRACE:DELETE:ALL`

Explanation This is a sequential command.

:TRACe:STATe[:<trace name>]

Function Sets/queries the display status of the specified trace.

Syntax `:TRACe:STATe[:<trace name>]<wsp>OFF|ON|0|1`
`:TRACe:ACTive?`
<trace name> = Trace to be transferred (TRA|TRB|TRC|TRD|TRE|TRF|TRG)
OFF = Hide trace (BLANK)
ON = Makes trace visible (DISP).
Response 0 = OFF, 1 = ON

Example `:TRACE:STATE OFF`
`:TRACE:STATE OFF? -> 0`

Explanation
- If <trace name> is omitted, the command is executed with respect to the active trace.
- If <trace name> is specified, the specified trace is set as the active trace after the command is exeucted.
- This is a sequential command.

TRACe:TEMPlate:DATA

Function Adds data to the specified template or queries the data.

Syntax `:TRACe:TEMPlate:DATA<wsp><template>,<wavelength>[M],<level>[DB]`
`:TRACe:TEMPlate:DATA?<wsp><template>`
<template> = Template (UPPer|LOWer|TARGet)
<wavelength> = Wavelength of template data to be added [nm]
<level> = Lvl. of template data added [dB]
Response <integer>,<wavelengh>,<level>,<wavelength>,<level>, ... ,<level>
<integer> = Number of data points
<wavelength> = wavelength value [m]
<level> = Level value [dB]

Example `:TRACE:TEMPlATE:DATA TARGET,1550NM,-10dbm`
`:TRACE:TEMPlATE:DATA? TARGET -> 3,+1.54000000E-006,-1.00000000E+001,+1.54500000E-006,-5.00000000E+000,+1.55000000E-006,-1.00000000E+001`

Explanation
- Adds data to a specified template.
- After data has been added, it will be sorted by wavelength.
- If data exceeding the maximum number of template data is added, an execution error occurs.
- This is a sequential command.

:TRACe:TEMPlate:DATA:ADELete

Function Deletes all data of a specified template.

Syntax `:TRACe:TEMPlate:DATA:ADELete<wsp><template>`
<template> = Template (UPPer|LOWer|TARGet)

Example `:TRACE:TEMPlATE:DATA:ADELETE TARGET`

Explanation
- Deletes all data of a specified template.
- This is a sequential command.

:TRACe:TEMPlate:DATA:ETYPe

Function Sets/queries the extrapolation mode of the specified template.

Syntax `:TRACe:TEMPlate:DATA:ETYPe<wsp><template>, <type>`
<template> = Template (UPPer | LOWer | TARGet)
<type> = Extrapolation type
A = Extrapolation type A
B = Extrapolation type B
NONE = No extrapolation
Response 0 = NONE, 1 = A, 2 = B

Example `:TRACE:TEMPlATE:DATA:ETYPE TARGET,NONE`
`:TRACE:TEMPlATE:DATA:ETYPE? target -> NONE`

Explanation This is a sequential command.
Parameter: Response ex. Same as the above

:TRACe:TEMPlate:DATA:MODE

Function Sets/queries the absolute value mode/relative value mode of the specified template.

Syntax `:TRACe:TEMPlate:DATA:MODE<wsp><template>,<mode>`
<template> = Template (UPPer|LOWer|TARGet)
<mode> = Mode (ABSolute | RELative)
ABSolute = Absolute value mode
RELative = Relative value mode
Response 0 = ABSolute, 1 = RELative

Example `:TRACE:TEMPlATE:DATA:MODE TARGET,RELATIVE`
`:TRACE:TEMPlATE:DATA:MODE? TARGET -> REL`

Explanation This is a sequential command.

:TRACe:TEMPlate:DISPlay

Function Sets/queries display ON/OFF for the specified template.

Syntax `:TRACe:TEMPlate:DISPlay<wsp><template>,OFF|ON|0|1`
`:TRACe:TEMPlate:DISPlay?<wsp><template>`
<template>= Template (UPPer|LOWer|TARGet)
OFF = Display OFF
ON = Display ON
Response 0 = OFF, 1 = ON

Example `:TRACE:TEMPlATE:DISPLAY TARGET,OFF`
`:TRACE:TEMPlATE:DISPLAY? TARGET-> 0`

Explanation This is a sequential command.

:TRACe:TEMPlate:GONogo

Function Sets or acquires ON/OFF of the go/no-go decision function of the template function.

Syntax `:TRACe:TEMPlate:GONogo<wsp>OFF|ON|0|1`
`:TRACe:TEMPlate:GONogo?`
OFF = Judgement function OFF
ON = Judgment function ON
Response 0 = OFF, 1 = ON

Example `:TRACE:TEMPlATE:GONOGO OFF`
`:TRACE:TEMPlATE:GONOGO? -> 0`

Explanation This is a sequential command.

:TRACe:TEMPlate:LEVel:SHIFt

Function Sets/queries the amount of level shift for the template.

Syntax `:TRACe:TEMPlate:LEVel:SHIFt<wsp><NRf>[DB]`
`:TRACe:TEMPlate:LEVel:SHIFt?`
<NRf> = Level shift amount [dB]

Example `:TRACE:TEMPlATE:LEVEL:SHIFT -1db`
`:TRACE:TEMPlATE:LEVEL:SHIFT?-> -1.00000000E+000`

Explanation This is a sequential command.

:TRACe:TEMPlate:RESult?

Function Queries the results of go/no-go decision of the template function.

Syntax `:TRACe:TEMPlate:RESult?`
Response 0= No go, 1= Go

Example `:TRACE:TEMPlATE:RESULT? -> 1`

Explanation This is a sequential command.

:TRACe:TEMPlate:TTYPe

Function Sets/queries judgement type of the go/no-go decision function of the template function.

Syntax `:TRACe:TEMPlate:TTYPe<wsp><type>`
`:TRACe:TEMPlate:TTYPe?`
<type>=Judgement type
UPPer = Judge Upper line only
LOWer= Judge Lower line only
U&L = Judge both Upper and LOWer line
Response 0 = UPPer, 1 = LOWer, 2 = U&L

Example `:TRACE:TEMPlATE:TTYPE U&L`
`:TRACE:TEMPlATE:TTYPE? -> 2`

Explanation This is a sequential command.

:TRACe:TEMPlate:WAVelength:SHIFt

Function Sets/queries the amount of wavelength shift for the template.

Syntax `:TRACe:TEMPlate:WAVelength:SHIFt<wsp><NRf>[M]`
`:TRACe:TEMPlate:WAVelength:SHIFt?`
<NRf> = Amount of a wavelength shift [m]

Example `:TRACE:TEMPlATE:WAVELENGTH:SHIFT -5NM`
`:TRACE:TEMPlATE:WAVELENGTH:SHIFT? -> -5.00000000E-009`

Explanation This is a sequential command.

TRIGger Sub System Command

:TRIGger[:SEQuence]:DELay

Function Sets/queries the trigger delay.

Syntax
```
:TRIGger[:SEQuence]:DELay<wsp><NRf>[S]
:TRIGger[:SEQuence]:DELay?
```
<NRf> = delay [sec]

Example
```
:TRIGER:DELAY 100.0US
:TRIGER:DELAY? -> +1.00000000E-004
```

Explanation
- When this command is executed, the external trigger mode becomes enabled.

(`TRIGger[:SEQuence]:STATe ON`)
- This is a sequential command.

:TRIGger[:SEQuence]:GATE:ITIMe

Function Sets or queries sampling interval for gate sampling.

Syntax
```
:TRIGger[:SEQuence]:GATE:ITIMe<wsp><NRf>[S]
:TRIGger[:SEQuence]:GATE:ITIMe?
```
<NRf> = Sampling interval

Response <NRf> = Sampling interval[S]

Example
```
:TRIGGER:SEQUENCE:GATE:ITIME 100ms
:TRIGGER:SEQUENCE:GATE:ITIME? ->
+1.00000000E-001
```

Description
- This is a sequential command.

:TRIGger[:SEQuence]:GATE:LOGic

Function Sets or queries the gate signal logic of gate sampling.

Syntax
```
:TRIGger[:SEQuence]:GATE:LOGic<wsp>POSI|NEGA|0|1
:TRIGger[:SEQuence]:GATE:LOGic?
```
POSI = Sampling is performed when the gate signal is at high level

NEGA = Sampling is performed when the gate signal is at low level

Response 0 = POSI, 1 = NEGA

Example
```
:TRIGGER:SEQUENCE:GATE:LOGIC POSI
:TRIGGER:SEQUENCE:GATE:LOGIC? -> 0
```

Description
- This is a sequential command.

:TRIGger[:SEQuence]:SLOPe

Function Sets/queries the trigger edge.

Syntax
```
:TRIGger[:SEQuence]:SLOPe<wsp>RISE|FALL|0|1
:TRIGger[:SEQuence]:SLOPe?
```
RISE = RISE

FALL = FALL

Response 0 = RISE, 1 = FALL

Example
```
:TRIGER:SLOPE RISE
:TRIGER:SLOPE? -> 0
```

Explanation
- When this command is executed, the external trigger mode becomes enabled.
- This is a sequential command.

:TRIGger[:SEQuence]:STATe

Function Sets/queries the external trigger mode.

Syntax
```
:TRIGger[:SEQuence]:STATe<wsp>OFF|ON|PHOLd|0|1|2
:TRIGger[:SEQuence]:STATe?
```
OFF: External Trigger OFF

ON: External trigger mode

PHOLd: Peak hold mode

GATE: Gate sampling

Response 0 = OFF, 1 = ON, 2 = PHOLd, 3 = GATE

Example
```
:TRIGER:STATE ON
:TRIGER:STATE? -> 1
```

Explanation
- This is a sequential command.

:TRIGger[:SEQuence]:INPut

Function Sets/queries the signal of the input trigger.

Syntax
```
:TRIGger[:SEQuence]:INPut<wsp>ETRigger|STRigger|SENable|0|1|2
:TRIGger[:SEQuence]:INPut?
```
ETRigger|0: Sampling trigger

STRigger|1: Sweep trigger

SENable|2: Sample enable

Example
```
:TRIGER:INPUT STRIGGER
:TRIGER:INPUT? -> 1
```

Explanation This is a sequential command.

:TRIGger[:SEQuence]:OUTPut

Function Sets/queries the signal of the output trigger.

Syntax
```
:TRIGger[:SEQuence]:OUTPut<wsp>OFF|SSTatus|0|1
:TRIGger[:SEQuence]:OUTPut?
```
OFF: OFF

SSTatus: Sweep status

Response 0 = OFF, 1 = SSTatus

Example
```
:TRIGER:OUTPUT SSTATUS
:TRIGER:OUTPUT? -> 1
```

Explanation This is a sequential command.

:TRIGger[:SEQuence]:PHOLd:HTIMe

Function Sets/queries the hold time of peak hold mode.

Syntax
```
:TRIGger[:SEQuence]:PHOLd:HTIMe<wsp><NRf>[s]
:TRIGger[:SEQuence]:PHOLd:HTIMe?
```
<NRf> = Hold time [s]

Example
```
:TRIGER:PHOLD:HTIME 100MS
:TRIGER:PHOLD:HTIME? ->
+1.00000000E-1
```

Explanation This is a sequential command.

UNIT Sub System Command

:UNIT:POWer:DIGit

Function Sets/queries the number of decimal places displayed for the level value.

Syntax
```
:UNIT:POWer:DIGit<wsp>1|2|3
:UNIT:POWer:DIGit?
```
1, 2, 3 = Number of displayed digits

Example
```
:UNIT:POWER:DIGIT 3
:UNIT:POWER:DIGIT? -> 3
```
Explanation This is a sequential command.

:UNIT:X

Function Sets/queries the units for the X axis.

Syntax
```
:UNIT:X<wsp>WAVelength|FREQuency|
0|1
:UNIT:X?
```
WAVelength|0 = Wavelength

FREQuency|1 = Frequency

Response 0 = WAVelength, 1 = FREQuency,

Example
```
:UNIT:X FREQUENCY
:UNIT:X? -> 1
```
Explanation This is a sequential command.

6.7 Output Format for Analysis Results

Output of Analysis Results

The analysis results of analysis functions are collectively output using the CALCulate:DATA? command. If analysis has been not performed, a query error occurs.

Output Data Format for Each Analysis Function

The output data format of each analysis function is as shown below.
For information on abbreviations such as <center wl>, see “List of Abbreviations of Data Output using the CALCulate:DATA? Command.”

THRESH , ENVELOPE, PK-RMS

<center wl>,<spec wd>,<mode num>

RMS

<center wl>,<spec wd>

NOTCH

<center wl>,<notch wd>

SMSR

- **SMSR1, SMSR2**
  <peak wl>,<peak lvl>,<2nd peak wl>,<2nd peak lvl>,<delta wl>,<delta lvl>
- **SMSR3, SMSR4**
  <peak wl>,<peak lvl>,<2nd peak wl(L)>,<2nd peak lvl(L)>,<delta wl(L)>,<delta lvl(L)>, <2nd peak wl(R)>,<2nd peak lvl(R)>,<delta wl(R)>,<delta lvl(R)>

POWER

<total pow>

DFB-LD

<spec wd>,<peak wl>,<peak lvl>,<mode ofst>,<smsr>

FP-LD

<spec wd>,<peak wl>,<peak lvl>,<center wl>,<total pow>,<mode num>

LED

<spec wd>,<peak wl>,<peak lvl>,<center wl>,<total pow>

PMD

<left mode peak>,<right mode peak>,<pmd>

WDM

- **ABSOLUTE, CH RELATION = OFFSET**
  <ch num>,<center wl>,<peak lvl>,<offset wl>, <offset lvl>, <noise>, <snr>,...

- **ABSOLUTE, CH RELATION = SPACING**
  <ch num>,<center wl>,<peak lvl>,<spacing>,<lvl diff>,<noise>,<snr>,...

- **RELATIVE**
  <ch num>,<grid wl>,<center wl>,<rel wl>,<peak lvl>,<noise>,<snr>,...

- **DRIFT (MEAS)**
  <ch num>,<grid wl>,<center wl>,<wl diff max>,<wl diff min>,<ref lvl>, <peak lvl>, <lvl diff max>, <lvl diff min>,...

- **DRIFT (GRID)**
  <ch num>,<ref wl>,<center wl>,<wl diff max>,<wl diff min>,<ref lvl>, <peak lvl>, <lvl diff max>, <lvl diff min>,...

EDFA-NF

<ch num>,<center wl>,<input lvl>,<output lvl>,<ase lvl>, <resoln>, <gain>, <nf>,...

FILTER-PK

<peak wl>,<peak lvl>,<center wl>,<spec wd>,<l-xtalk>,<r-xtalk>, <ripple>

FILTER-BTM

<btm wl>, <btm lvl>, <center wl>, <notch wd>, <l-xtalk>, <r-xtalk>

WDM FIL-PK

<ch num>, <nominal wl>, <peak wl>, <peak lvl>, <xdb wd>, <center wl>, <xdb sb>, <xdb pb>, <ripple>, <l-xtalk>, <r-xtalk>,...

* Items with SW set to OFF are also output.

WDM FIL-BTM

<ch num>, <nominal wl>, <btm wl>, <btm lvl>, <xdb ntwd>, <center wl>, <xdb sb>, <xdb eb>, <ripple>, <l-xtalk>, <r-xtalk>,...

* Items with SW set to OFF are also output.

COLOR

<dominant wl>, <x col>, <y col>, <z col>

List of Abbreviations of Data Output Using the CALCulate:DATA? Command

Abbreviation	Description	Format	Output Unit
<center wl>	Center wavelength	<NRf>	m / Hz
<spec wd>	Spectrum width	<NRf>	m / Hz
<mode num>	Mode number	<integer>	
<notch wd>	Notch width	<NRf>	m / Hz
<peak wl>	Peak wavelength	<NRf>	m / Hz
<peak lvl>	Peak level	<NRf>	dBm
<2nd peak wl>	2nd peak wavelength	<NRf>	m / Hz
<2nd peak lvl>	2nd peak level	<NRf>	dB
<delta wl>	Wavelength difference	<NRf>	m / Hz
<delta lvl>	Level difference	<NRf>	dB
<mode ofst>	Mode offset	<NRf>	m / Hz
<smsr>	SMSR value	<NRf>	dB
<smsr(L)>	SMSR value (shorter wavelength side)	<NRf>	dB
<smsr(R)>	SMSR value (longer wavelength side)	<NRf>	dB
<mode ofst(L)>	Mode offset (shorter wavelength side)	<NRf>	m/Hz
<mode ofst(R)>	Mode offset (longer wavelength side)	<NRf>	m/Hz
<2nd peak wl(L)>	Second peak wavelength (shorter wavelength side)	<NRf>	m/Hz
<2nd peak wl(R)>	Second peak wavelength (longer wavelength side)	<NRf>	m/Hz
<2nd peak lvl(L)>	Second peak level (shorter wavelength side)	<NRf>	dB
<2nd peak lvl(R)>	Second peak level (longer wavelength side)	<NRf>	dB
<delta wl(L)>	Wavelength difference (shorter wavelength side)	<NRf>	m/Hz
<delta wl(R)>	Wavelength difference (longer wavelength side)	<NRf>	m/Hz
<delta lvl(L)>	Level difference (shorter wavelength side)	<NRf>	dB
<delta lvl(R)>	Level difference (longer wavelength side)	<NRf>	dB
<power>	Power value	<NRf>	dB/W
<total pow>	Total power value	<NRf>	dB / W
<mode num>	Mode number	<integer>	
<left mode peak>	Mode peak frequency (left)	<NRf>	Hz
<right mode peak>	Mode peak frequency (right)	<NRf>	Hz
<pmd>	PMD value	<NRf>	ps
<ch num>	Channel number	<integer>	
<offset wl>	Offset wavelength	<NRf>	m / Hz
<offset lvl>	Offset level	<NRf>	dB
<noise>	Noise level	<NRf>	dBm / NBW
<snr>	SNR value	<NRf>	dB
<grid wl>	Grid wavelength	<NRf>	m / Hz
<rel wl>	Relative wavelength	<NRf>	m / Hz
<wl diff max>	Wavelength difference (max.)	<NRf>	m / Hz
<wl diff min>	Wavelength difference (min.)	<NRf>	m / Hz
<ref lvl>	Relative level	<NRf>	dB
<lvl diff max>	Level difference (max.)	<NRf>	dB
<lvl diff min>	Level difference (min.)	<NRf>	dB
<input lvl>	Input level	<NRf>	dBm
<output lvl>	Output level	<NRf>	dBm
<ase lvl>	ASE level	<NRf>	dBm / RES
<resoln>	Measurement resolution	<NRf>	m
<gain>	Gain	<NRf>	dB
<nf>	NF value	<NRf>	dB
<l-xtalk>	Crosstalk value (left)	<NRf>	dB
<r-xtalk>	Crosstalk value (right)	<NRf>	dB
<ripple>	Ripple width	<NRf>	m / Hz

Abbreviation	Description	Format	Output Unit
<nominal wl>	Reference wavelength	<NRf>	m / Hz
<xdb wd>	Xdb width	<NRf>	m / Hz
<xdb sb>	XdB stop-band	<NRf>	m / Hz
<xdb pb>	XdB passband	<NRf>	m / Hz
<xdb eb>	XdB elimination band	<NRf>	m / Hz
<dominant wl>	Dominant Wavelength	<NRf>	m/Hz
<x col>	Chromaticity coordinates (x)	<NRf>	
<y col>	Chromaticity coordinates (y)	<NRf>	
<z col>	Chromaticity coordinates (z)	<NRf>	
<color temp>	Color temperature	<NRf>	K
<dev>	Deviation	<NRf>	

7.1 Editing a Program

To use the program functions, a program must be pre-registered in the instrument.

Procedure

1. Press **PROGRAM**.

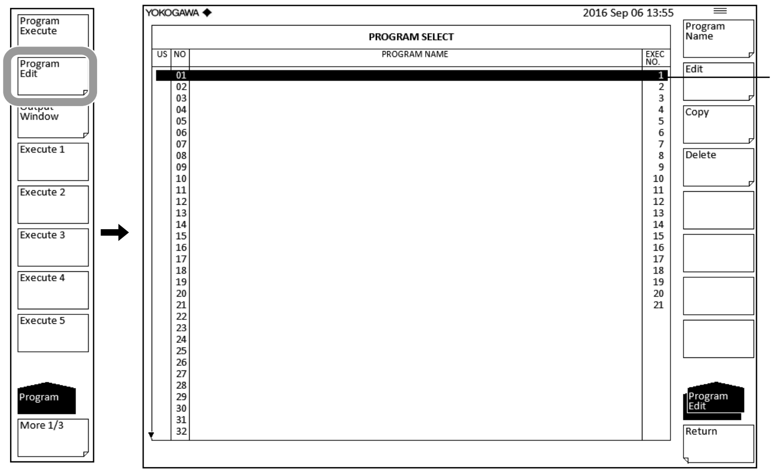
   The program menu is displayed.

2. Press the **Program Edit** soft key. The program registration screen appears.

Note

- Thirty-two program names are displayed on a single screen.
- The US column includes an asterisk (*) if a program has already been registered in the corresponding program number.
- The EXECUTE NO. column shows the registered program numbers for programs that have been registered to the <Execute 1> to <Execute 21> keys.
  See section 7.2, "Executing Programs" for information on registering programs to the Execute 1–Execute 21 soft keys.

3. Select a registration number using the **rotary knob** or the **arrow keys**.



YOKOGAWA ◆ 2016 Sep 06 13:55

PROGRAM SELECT

US	NO	PROGRAM NAME	EXEC NO.
	01		1
	02		2
	03		3
	04		4
	05		5
	06		6
	07		7
	08		8
	09		9
	10		10
	11		11
	12		12
	13		13
	14		14
	15		15
	16		16
	17		17
	18		18
	19		19
	20		20
	21		21
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		

Selected program number

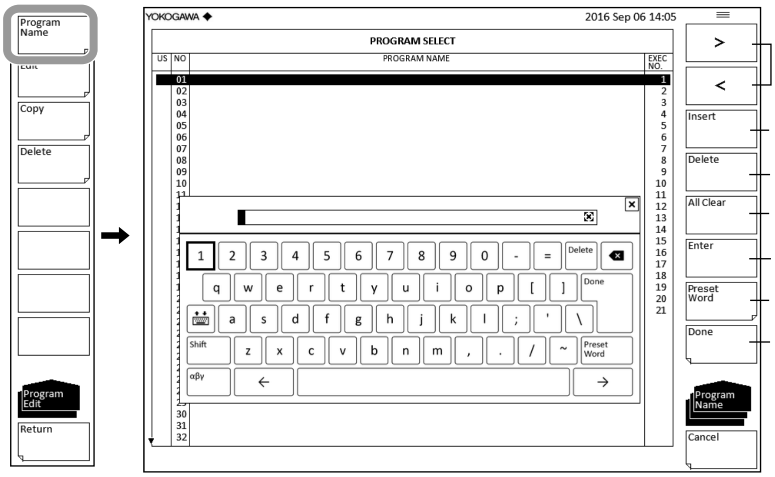
Entering a Program Name

4. Press the **Program Name** soft key.

The program name input screen appears.

5. Enter a program name using the **rotary knob** and soft keys.

6. After entering a name, press the **Done** soft key. The program name is finalized, the instrument returns to the program registration screen. The entered program name is displayed in the program registration screen.



Move input position

Insert characters

Delete characters

All clear

Enter characters

Register/set entry characters

Finalize program name

Note

To register and reuse an entered string, or to use a previously entered string, press the Preset Word soft key.

Registering Strings

After a string has been entered in the program name entry screen, press the Preset Word soft key.

Select a registration number and press the Save soft key. The entered string is registered in the program name input screen.

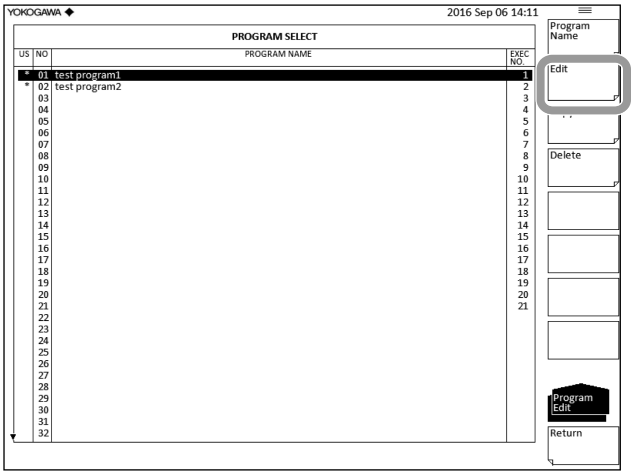
Using Registered Strings

Press the Preset Word soft key.

Select the number of the string you wish to use and press the Recall soft key. The selected string is entered as a program name.

Editing a Program

7. Select a program to edit in the program registration screen and press the **Edit** soft key. The program edit screen appears.



YOKOGAWA ◆ 2016 Sep 06 14:11

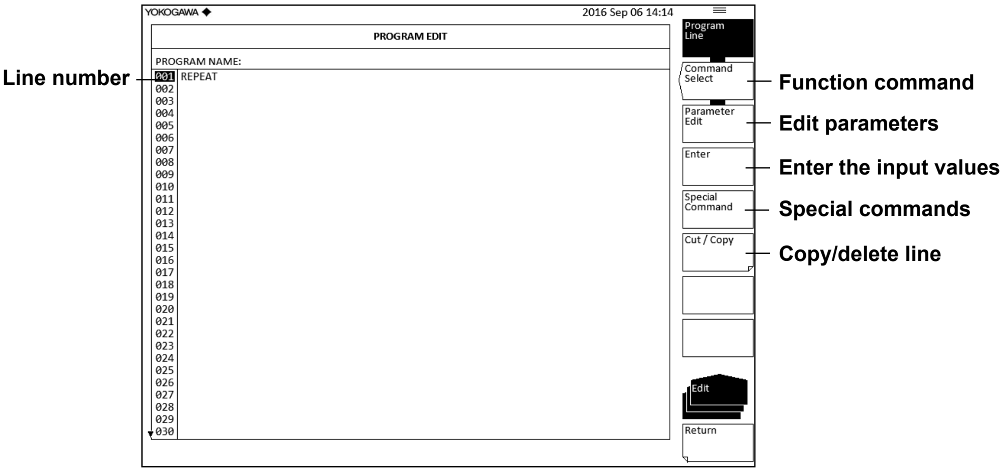
PROGRAM SELECT

US	NO	PROGRAM NAME	EXEC NO.
*	01	test program1	1
*	02	test program2	2
	03		3
	04		4
	05		5
	06		6
	07		7
	08		8
	09		9
	10		10
	11		11
	12		12
	13		13
	14		14
	15		15
	16		16
	17		17
	18		18
	19		19
	20		20
	21		21
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		

8. Select a line to edit using the **rotary knob** or the **arrow keys**. When a line of a specified command parameter is selected, the **Parameter Edit** soft key is enabled.

9. Edit the program using the soft keys. For the settings associated with each soft key, see pages 7-6 and 7-7.

10. When finished editing the program, press the **Return** soft key.



YOKOGAWA ◆ 2016 Sep 06 14:14

PROGRAM EDIT

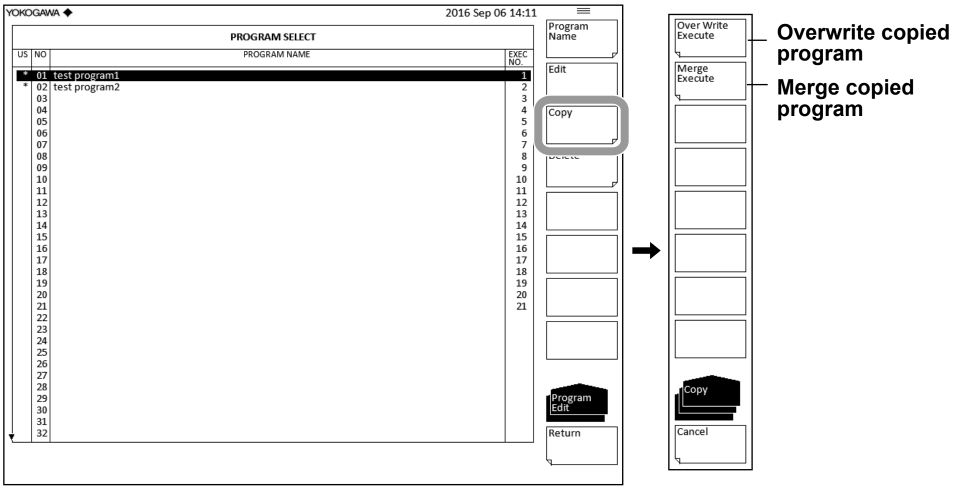
PROGRAM NAME:

```
001 REPEAT
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
```

Copying/Merging (Combining) Programs

11. Select the program to copy in the program registration screen in step 2.

12. Press the **Copy** soft key.



PROGRAM SELECT

US	NO	PROGRAM NAME	EXEC NO.
*	01	test program1	1
*	02	test program2	2
	03		3
	04		4
	05		5
	06		6
	07		7
	08		8
	09		9
	10		10
	11		11
	12		12
	13		13
	14		14
	15		15
	16		16
	17		17
	18		18
	19		19
	20		20
	21		21
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		

Overwrite copied program

Merge copied program

Overwriting a Copied Program

13. Select a copy destination program and press the **Over Write Execute** soft key. The copied program overwrites the selected destination program.

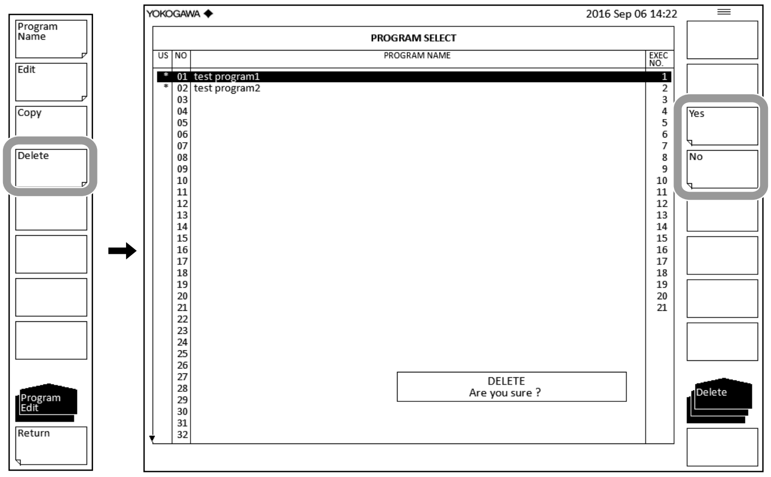
Merging a Copied Program

14. After performing step 12, select a copy destination program and press the **Merge Execute** soft key. The contents of the copied program are pasted onto the end of the copy destination program (making one large program).

Deleting a Program

15. Select the program to delete in the program registration screen in step 2.

16. Press the **Delete** soft key. A confirmation message is displayed.



PROGRAM SELECT

US	NO	PROGRAM NAME	EXEC NO.
*	01	test program1	1
*	02	test program2	2
	03		3
	04		4
	05		5
	06		6
	07		7
	08		8
	09		9
	10		10
	11		11
	12		12
	13		13
	14		14
	15		15
	16		16
	17		17
	18		18
	19		19
	20		20
	21		21
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		

DELETE
Are you sure ?

17. Press the **Yes** or **No** soft key to delete the program or cancel.

Program Editing Operations

The following describes the operation of the various soft keys when editing programs. Each description assumes that the program editing screen is open (by pressing **PROGRAM**, followed by the **Program Edit** > **Edit** soft keys).

Selecting Commands

The following two types of commands are available.

Function Commands

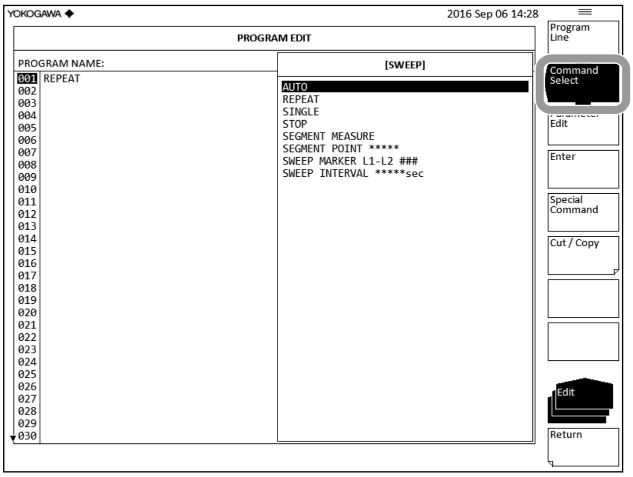
These commands execute the same function as a function switch (including the contents of a soft key).
(Commands corresponding to the soft keys such as Single and Span)

Special Commands

These commands include jump commands, program control commands for conditional decision, etc., control commands to an external device, and data output commands.

1. To select function commands or special commands, press the **Command Select** or **Special Command** soft keys, respectively. The function command or special command selection screen is displayed.

Function commands



YOKOGAWA ◆ 2016 Sep 06 14:28

PROGRAM EDIT

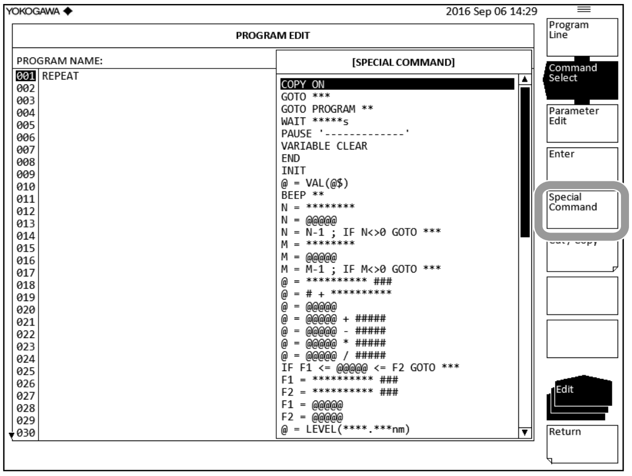
PROGRAM NAME:

```
001 REPEAT
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
```

[SWEEP]

```
AUTO
REPEAT
SINGLE
STOP
SEGMENT MEASURE
SEGMENT POINT *****
SWEEP MARKER L1-L2 ###
SWEEP INTERVAL *****sec
```

Special commands



YOKOGAWA ◆ 2016 Sep 06 14:29

PROGRAM EDIT

PROGRAM NAME:

```
001 REPEAT
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
```

[SPECIAL COMMAND]

```
COPY ON
GOTO ***
GOTO PROGRAM **
WAIT *****s
PAUSE '-------------'
VARIABLE CLEAR
END
INIT
@ = VAL(@$)
BEEP **
N = ********
N = @@@@@
N = N-1 ; IF N<>0 GOTO ***
M = ********
M = @@@@@
M = M-1 ; IF M<>0 GOTO ***
@ = ********** ###
@ = # + **********
@ = @@@@@
@ = @@@@@ + #####
@ = @@@@@ - #####
@ = @@@@@ * #####
@ = @@@@@ / #####
IF F1 <= @@@@@ <= F2 GOTO ***
F1 = ********** ###
F2 = ********** ###
F1 = @@@@@
F2 = @@@@@
@ = LEVEL(****.***nm)
```

2. Select a command using the **rotary knob** or the **arrow keys**, and press the **Enter** soft key. The selected command is entered. When entering commands that require parameter settings, the parameter setting screen is displayed.

3. Enter the parameter and press the **Enter** soft key. The parameter is set.

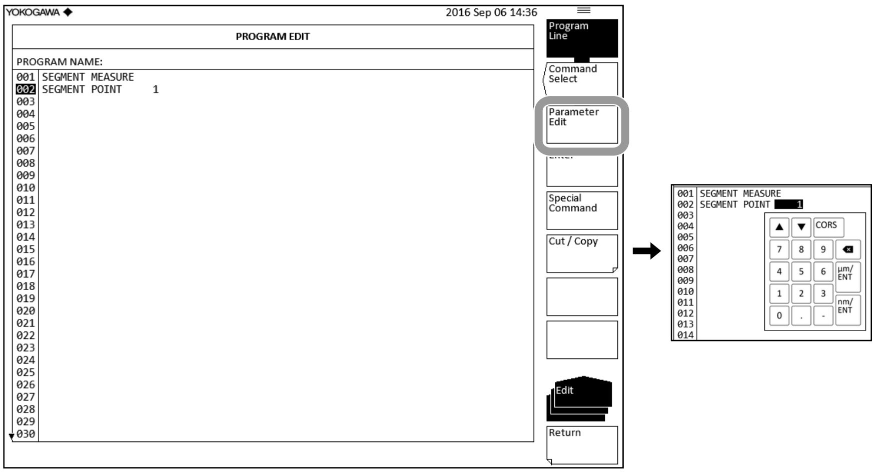
Note

- The ***** portion of commands are numbers, the ### portion is the selected parameter, and - - - - - - is text input.
- Function commands can also be set using the mouse. Right-click the mouse to display a shortcut list of panel keys. Left-clicking enters the function command corresponding to the selected panel key.

Editing Parameters

Modifying Parameters of an Entered Command

1. Select the line of the command whose parameter you wish to modify using the **rotary knob** or the **arrow keys**. The Parameter Edit soft key becomes enabled.
2. Press the **Parameter Edit** soft key. The parameter setting screen is displayed.



```
YOKOGAWA ◆                                   2016 Sep 06 14:36
PROGRAM EDIT
PROGRAM NAME:
001 SEGMENT MEASURE
002 SEGMENT POINT     1
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
```

```
001 SEGMENT MEASURE
002 SEGMENT POINT    1
003
004
005
006
007
008
009
010
011
012
013
014
```

3. Enter the parameter and press the **Enter** soft key. The parameter is set.

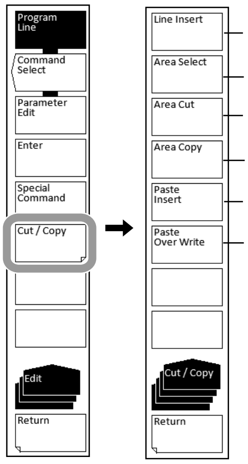
Note

The parameter setting screen displayed differs depending on the type of parameter.

Inserting, Copying, or Deleting a Line

You can copy or delete the contents of a line.

1. Press the **Cut/Copy** soft key. The Cut/Copy screen is displayed.



Insert line

Set range of lines for copy/delete

Delete line

Copy line

Paste (insert)

Paste (overwrite)

Inserting a Line

2. Select a line number on which to insert a line using the **rotary knob** or the **arrow keys**.
3. Press the **Line Insert** soft key. One line is inserted above the selected line number.

Note

If commands have been entered in all 200 lines, a new line cannot be inserted.

Deleting a Line

2. To delete one line, select the line to delete using the **rotary knob**.

   To delete multiple lines, select the first or last line to delete and press the **Area Select** soft key.

   Select the range of lines to delete using the **rotary knob** or the **arrow keys**.

3. Press the **Area Cut** soft key. The specified range of lines is deleted.

   To restore the deleted line, press UNDO/LOCAL.

Copying a Line

2. To copy one line, select the line to copy using the **rotary knob** or the **arrow keys**.

   To copy multiple lines, select the first or last line to copy and press the **Area Select** soft key.

   Select the range of lines to copy using the **rotary knob** or the **arrow keys**.

3. Press the **Area Copy** soft key. The specified range of lines is copied.
4. Select a copy destination line using the **rotary knob** or the **arrow keys**.
5. To insert the copied lines, press the **Paste Insert** soft key.

   To overwrite with the copied lines, press the **Paste Over Write** soft key.

   The copied lines are pasted, starting from the line selected as the copy destination. To restore the pasted contents, press **UNDO/LOCAL**.

Explanation

Programs

Up to 64 programs can be registered.

A program key can be assigned to each program allowing you to execute the program simply by pressing its soft key.

Commands

There are two types of executable commands.

Function Commands

(Commands corresponding to the soft keys such as Single and Span)

Special Commands

These commands include jump commands, program control commands for conditional decision, etc., control commands to an external device, and data output commands.

For detailed information on commands, see section 7.3, "Program Function Commands."

Merging a Program

You can combine two different programs into one program.

The copied program is pasted onto the end of another specified program.

7.2 Executing a Program

There are two methods for executing a program: specifying then executing the program, and assigning the program to a soft key and executing it directly with that key.

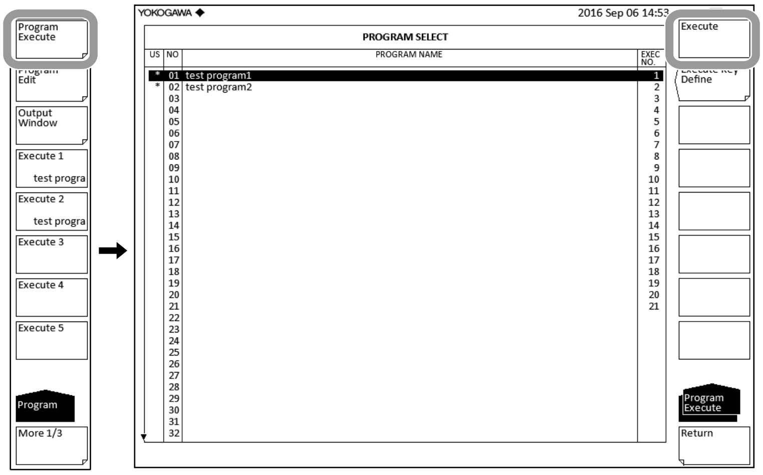
Procedure

Specifying and Executing a Program

1. Press **PROGRAM**.

   The program menu is displayed.

2. Press the **Program Execute** soft key. The program selection screen appears.



PROGRAM SELECT

US	NO	PROGRAM NAME	EXEC NO.
*	01	test program1	1
*	02	test program2	2
	03		3
	04		4
	05		5
	06		6
	07		7
	08		8
	09		9
	10		10
	11		11
	12		12
	13		13
	14		14
	15		15
	16		16
	17		17
	18		18
	19		19
	20		20
	21		21
	22		
	23		
	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		
	32		

3. Select a program to execute using the **rotary knob** or the **arrow keys**.
4. Press the **Execute** soft key. The program executes.

Note

To stop the program during execution, press the Program Exit soft key.

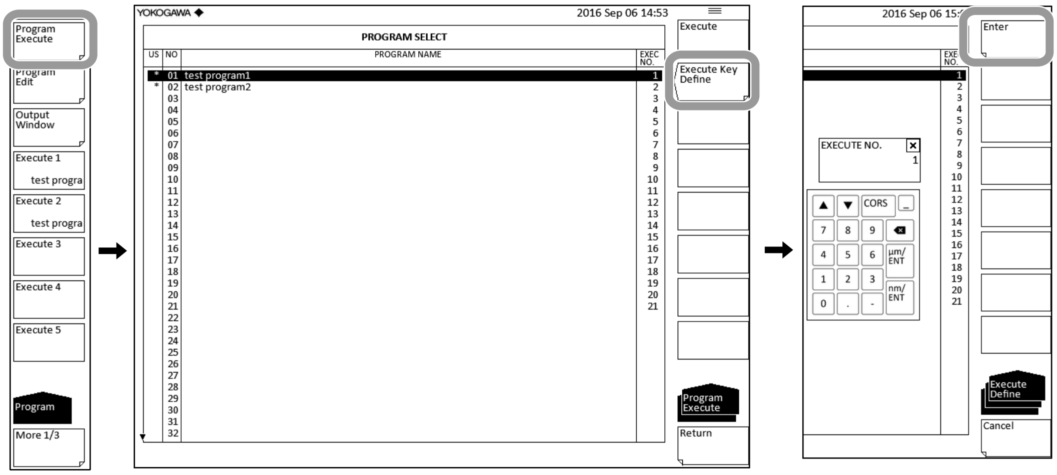
Assigning a Program to a Soft Key and Executing

Assigning to a Soft Key

1. Press **PROGRAM**.

   The program menu is displayed.

2. Press the **Program Execute** soft key. The program selection screen appears.



3. Select a program to assign using the **rotary knob** or the **arrow keys**.

4. Press the **Execute Key Define** soft key. A screen for assigning soft keys is displayed.

5. Enter a soft key number between 1 and 21 and press the **Enter** soft key. If a program is already assigned to that number, the existing program is overwritten.

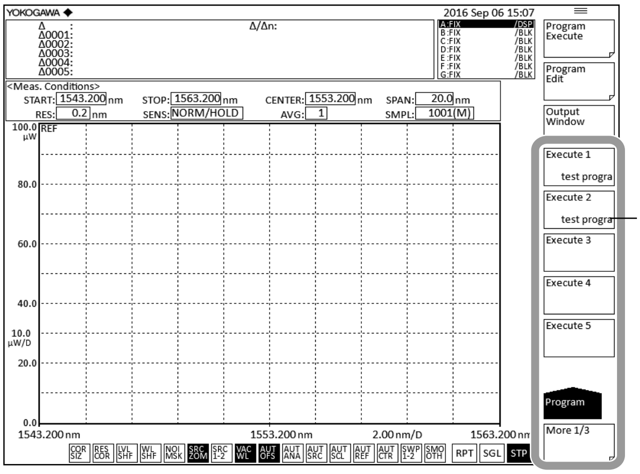
Note

A single program cannot be assigned to multiple soft keys.

Executing the Program

1. Press **PROGRAM**.

   The program menu is displayed.



Registered program name

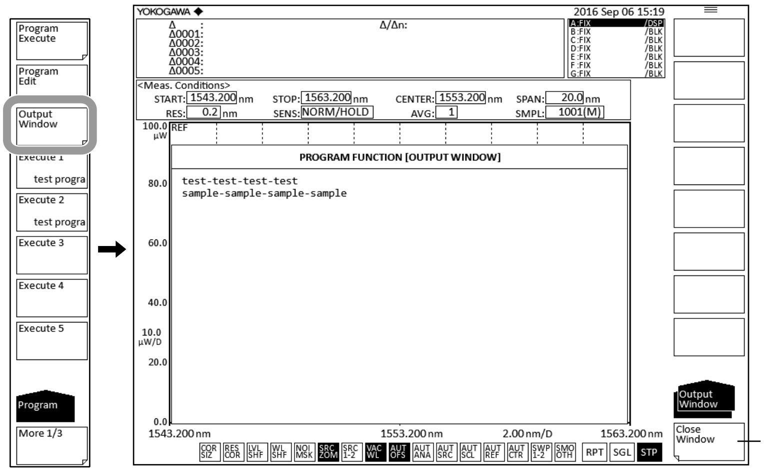
2. Press a soft key from **Execute 1** to **Execute 21**. The program assigned to the soft key executes.

Displaying the OUTPUT WINDOW

1. Press **PROGRAM**.

   The program menu is displayed.

2. Press the **Output Window** soft key. The OUTPUT WINDOW is displayed.



PROGRAM FUNCTION [OUTPUT WINDOW]

```
test-test-test-test
sample-sample-sample-sample
```

Close OUTPUT WINDOW

Note

If there is no data to display in the OUTPUT WINDOW, the Output Window soft key is disabled.
Data and characters output by the DATA OUTPUT command are displayed.

3. To close the OUTPUT WINDOW, press the **Close Window** soft key.

Note

- The contents of the OUTPUT WINDOW are held until execution of the OUTPUT WINDOW CLEAR special command.
- The contents of the OUTPUT WINDOW can be stored in a file. See the main unit user’s manual (IM AQ6374-01EN) for details.
- If the contents of the OUTPUT WINDOW exceeds 200 lines, data will be erased beginning from the first line, in turn.
- Turning off the power switch on the instrument erases data in the OUTPUT WINDOW.

Explanation

Using Special Commands

During program execution, you can perform unique operations with commands.

When Executing a Program Including “PAUSE ’----------‘”

The program pauses.

The message included in the “PAU2SE ’----------‘” line is displayed, and the program pauses. To resume execution of the program, press the Continue soft key.

If a program is executed via remote control, the “PAUSE ’------------‘ command is ignored.

When Executing a Program Including the “DATA INPUT -------------‘;@” Command

After the program executes, a data entry window is displayed.

In this case, one of two types of windows will appear depending on the @ variable.

String variables: Enter a file name using the same procedure as that of label input and press the DONE soft key.

Numerical variables: A data entry window is displayed. Enter an arbitrary number using the rotary knob, arrow keys, or ten key. If a program is executed via remote control, the “DATA INPUT ’------------ ‘;@ command is ignored.

Outputting Data Using “DATA OUTPUT @@@@@”

When executing a program, the OUTPUT WINDOW for displaying output data is displayed.

The contents of the variables specified by “@@@@@” appear in the OUTPUT WINDOW. Up to 200 lines can be displayed in the OUTPUT WINDOW. Only 20 lines can be displayed at once. To display lines other than the first 20, use the rotary knob or arrow keys to scroll.

The OUTPUT WINDOW can be displayed during execution of a program. To do so, use the “OUTPUT WINDOW ###” special command. Note that the OUTPUT WINDOW disappears if the program ends.

Error Encountered upon Execution of a Program

If an error occurs during the running of a program, an error number indicating the details of the error is displayed in a window, and execution of the program is stopped.

Classification of Error Numbers

300–307	Errors caused by attempting a setting in manual operation which is disabled
320–326	Special command-related errors
340–347	Input/output-related errors
360–369	External memory-related errors
380, 381	Other errors

The above numbers can be read out using the SYSTem:ERRor[:NEXT]? command (see section 6.6, "Instrument-Specific Commands").

No.	Message	Cause
300	Parameter out of range	A variable value is out of range or is not defined for a command that sets a parameter using variables.
302	Scale unit mismatch	There is a difference between the Y-axis scale of the active trace and the unit of a parameter in the "LINE MKR 3 or 4" command.
303	No data in active trace	Setting of the moving marker, a peak (or bottom) search, or activation of the analysis function was made with no data in the active trace.
304	Marker value out of range	Specified wavelength was out of the sweep range in the moving marker or line wavelength marker setting command.
305	No data in traces A or B	No waveform data in traces A or B when executing the "EDFA NF" command
306	Invalid data	Trace had no data when attempting to save it to memory or to write it to FD/INT.
307	Unsuitable Write item	All data items were OFF at execution of "WRITE DATA".
320	Undefined variable	A command containing an undefined variable was executed.
321	Variable unit mismatch	The unit of each variable does not agree within a command containing two or more variables.
322	Overflow	An overflow occurred in an arithmetic operation.
323	Undefined marker variable	A command containing a marker-value variable was executed when no marker had been displayed.
324	Invalid marker variable	A command containing the corresponding variable was executed at a time other than immedia tely after execution of a spectrum width search, peak search, etc.
325	Undefined line number	GOTO command's jumping destination is a number other than 1 to 200.
326	F1 greater than F2	F1>F2 when the "IF F1 @@@@@ F2" command was executed.
345	Option does not respond	No response from an external device.
346	Option is not connected	No external device is connected.
360	Disk full	No file can be created due to insufficient free space in the external memory.
361	Disk not inserted	No external memory is connected.
362	Disk is write protected	The external memory is write protected.
363	Disk not initialized	External memory is not initialized. Or, it has been formatted in a format not supported by this instrument.
364	Directory full	Directory is full, therefore no file can be created.
365	File not found	The specified file cannot be read because it has not been found. Or, the file does not exist on the disk.
366	File is write protected	The file is specified to be read only, so that it cannot be rewritten or deleted.
367	No data	No data to store.
368	File is not a trace file	A file cannot be read because it is not a trace file.
369	Illegal file name	A file cannot be saved due to an incorrect file name.
371	Directory already exist	Unable to make the directory because a directory with the same name already exists.

No.	Message	Cause
380	Undefined program	An attempt was made to run a program that is not defined.
381	Syntax error	Command incorrect. (a program has been rewritten for some reason)
382	Program nesting over	Nesting is not possible because the program nesting is already too deep.
383	Program reentrant error	The destination of the GOTO command is set to its own program number. This will create an infinite loop.

7.3 Program Function Commands

There are two types of commands used in a program: function commands which are input using a panel switch, and other special commands.

Variables

Variables that can be used in a program are shown in the table below.

Type	Variable Name	Description
Generalized variables	E, G–K, O–Z	Contains a generalized numeric variable.
	A$, B$, C$, D$	Contains a generalized string variable.
Special characters variables	FILE$	Contains the name of the last file accessed.
	TIME$	Contains the date and time. (Ex. 2016 Sep 08 20:45:37)
Marker variables	WM	Contains the wavelength value of the moving marker.
	W1	Contains the wavelength value of fixed marker 1.
	W2	Contains the wavelength value of fixed marker 2.
	W2-W1	Contains the wavelength difference between fixed markers 1 and 2.
	W(CH)	Contains the level values of fixed markers (CH: 1 to 1024).
	LM	Contains the level value of the moving marker.
	L1	Contains the level value of fixed marker 1.
	L2	Contains the level value of fixed marker 2.
	L2-L1	Contains the level difference between fixed markers 1 and 2.
	L(CH)	Contains the level values of fixed markers (CH: 1 to 1024).
Analysis variables	SPWD	Contains spectrum width applied in making a spectrum width search.
	PKWL	Contains a peak (or bottom) wavelength value applied in making a peak (or bottom) search or spectrum width search.
	MEANWL	Contains center wavelength applied in making spectrum width search.
	PKLVL	Contains a peak (or bottom) level value applied in making a peak (or bottom) search or spectrum width search.
	MODN	Contains the number of modes applied in making a spectrum width search.
	SMSR	Contains the side mode suppression ratio (level difference) applied in making SMSR measurements.
	SMSR2	Contains the longer wavelength side's value of the side mode suppression ratio (level difference) when an SMSR3 or SMSR4 measurement is executed.
	WDMCHN	Contains the number of channels detected in performing WDM analysis.
	WDMWL(CH)	Contains the center wavelength of channel CH used in performing WDM analysis.
	WDMLVL(CH)	Contains the level of channel CH used in performing WDM analysis.
	WDMSNR(CH)	Contains SNR of channel CH used in performing WDM analysis.
	MKPWR	Contains power obtained in making between line-markers power measurements.
	PMD	Contains the PMD value obtained in PMD analysis.

Type	Variable Name	Description
Analysis variables		
	NFCHN	Contains the number of channels detected in performing EDFA-NF analysis.
	NFWL(CH)	Contains the center wavelength of channel CH used in performing EDFA-NF analysis.
	NFLVLI(CH)	Contains the input signal level of channel CH used in performing EDFA-NF analysis.
	NFLVLO(CH)	Contains the output signal level of channel CH used in performing EDFA-NF analysis.
	NFASELV(CH)	Contains the ASE level of channel CH used in performing EDFA-NF analysis.
	NFGAIN(CH)	Contains the gain of channel CH used in performing EDFA-NF analysis.
	NFNF(CH)	Contains NF of channel CH used in performing EDFA-NF analysis.
	DOMWL	Dominant wavelength value is entered when measuring the dominant wavelength.
	XCOL	The chromaticity coordinate value X is entered when perfoming COLOR analysis
	YCOL	The chromaticity coordinate value Y is entered when perfoming COLOR analysis
	ZCOL	The chromaticity coordinate value Z is entered when perfoming COLOR analysis
Program control variables	M	Contains loop counter data.
	N	Contains loop counter data.
	F1	Contains a conditional judgment variable.
	F2	Contains a conditional judgment variable.
	CH	Contains an element number variable used in accessing an array variable (1–1024).
Temprate variables	GONO	Contains GONO judgment results

Principles of Variable-based Arithmetic Operations

For assignment of units after arithmetic operations when a variable with a unit is used in the operation, see below.

Expression	Results
(With a unit) × (Without unit)	With a unit
(With a unit)/(Without unit)	With a unit
(Without unit) + (Without unit)	Without unit
(Without unit) – (Without unit)	Without unit
(Without unit) × (Without unit)	Without unit
(Without unit) / (Without unit)	Without unit
(nm) + (nm)	(nm)
(nm) – (nm)	(nm)
(nm) / (nm)	Without unit
(dB) + (dB)	(dB)
(dB) – (dB)	(dB)
(dB) + (dB)	(dBm)
(dBm) – (dB)	(dBm)
(dBm) – (dBm)	(dB)
(#W) + (#W)	(#W)
(#W) – (#W)	(#W)
(#W) / (#W)	Without unit

Note

- For the units of dBm/nm, W/nm, dB/km, and %, dBm, W, dB, and without unit apply respectively in terms of variables.
- Arithmetic operations are made as noted above according to the unit of a variable, and the unit is appended to the result obtained after operation.
- If an arithmetic operation is made in any combination other than the above (addition, subtraction, multiplication, or division of variables with different units), the result of the operation has no units.
- The units of #W are treated as follows:

  1 mW=1
  1 mW=0.001
  1 nW=0.000001
  1 pW=0.000000001

Specifications of "@=VAL(@$)" Command

A character string other than the numerics located before a value (starting with a sign or number) in @$ character string will be ignored, and are converted as follows.

- " , "
- Numbers up to the next string or delimiter

If no numeric exists in @$ character string, "0" is substituted for variable @.

List of Function Commands

A description is given of the program commands of each function command. The optical spectrum analyzers on which the program command is valid is indicated along with the parameter range and variables supported.

SWEEP

Program Command	Description	Parameter ranges and supported variables.
AUTO	Auto sweep	
REPEAT	Repeat sweep	
SINGLE	Single sweep	
STOP	Sweep stop	
SEGMENT MEASURE	Makes measurements only by a specified number of points start- ing at the position of the wave- length being stopped.	
SEGMENT POINT *****	Specifies the number of points to be measured with the SEGMENT MEASURE key	1–100001 (1 step)
SWEEP MKR L1-L2 ###	Selects ON/OFF of sweep function between markers	###: ON or OFF
SWEEP INTVL *****sec	Sets the interval time for repeat sweep	MINIMUM, 1 to 99999sec (1 step) (MINIMUM when set to 0.)

CENTER

Program Command	Description	Parameter ranges and supported variables.
CENTER WL ****.***nm	Sets measurement center wavelength.	350.000 to 1750.000nm (0.001 step)
CENTER WL @@@@@	Sets the value of variable @@@@@ to measurement center wavelength	@@@@@: E, G, H, I, J, K, O, P, Q, R, S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), PKWL, MEANWL, WDMWL(CH), NFWL(CH) WAM1, WAM2, WAM3, WAM4
CENTER FREQ ***.**** THz	Sets measurement center frequency.	171.0000 to 857.0000THz (0.0001 step)
CENTER FREQ @@@@@	Sets the value of variable @@@@@ to measurement center frequency	@@@@@: E, G, H, I, J, K, O, P, Q, R, S, T, U, V, X, Y, Z, WM, W1, W2,WAM1, WAM2, WAM3, WAM4, W(CH), PKWL, MEANWL, WDMWL(CH), NFWL(CH)

Program Command	Description	Parameter ranges and supported variables.
START WL ***.**** nm	Sets measurement-starting wavelength.	1.000 to 1750.000 nm (0.001 step)
START FREQ ***.**** THz	Sets measurement-starting frequency.	10.0000 to 857.0000 THz (0.0001 step)
STOP WL ****.***nm	Sets measurement-ending wavelength.	350.000 to 2450.000 nm (0.001 step)
STOP FREQ ***.****THz	Sets measurement-ending frequency.	171.0000 to 999.9000 THz (0.0001 step)
PEAK->CENTER	Sets the center frequency of the waveform on the active trace	
MEAN WL->CENTER	Performs a spectrum width search on the active trace, and sets the results of center wavelength to the measurement center wavelength.	
AUTO CENTER ###	Executes every time a sweep finishes. <PEAK →CENTER> Function ON/Selects OFF	###: ON or OFF
VIEW SCALE->MEAS SCALE	Sets the current display conditions to measuring conditions.	

SPAN

Program Command	Description	Parameter ranges and supported variables.
SPAN WL ****.*nm	Sets the measuring span.	0.5 to 1400.0 nm (0.1 step)
SPAN WL @@@@@	Sets the value of variable @@@@@ to the measuring spa	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD, WAM2-WAM1, WAM4-WAM3
SPAN FREQ ***.**THz	Sets the measuring span.	0.05 to 686.00 THz (0.01 step)
SPAN FREQ @@@@@	Sets the value of variable @@@@@ to the measuring span	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD, WAM2-WAM1, WAM4-WAM3
START WL ****.***nm	Sets measurement-starting wavelength.	1.000 to 1750.000 nm (0.001 step)
START FREQ ***.**** THz	Sets measurement-starting . frequency.	10.0000 to 857.0000 THz (0.0001 step)
STOP WL ****.***nm	Sets measurement-ending wavelength.	350.000 to 2450.000 nm (0.001 step)
STOP FREQ ***.**** THz	Sets measurement-ending frequency.	171.0000 to 999.9000 THz (0.0001 step)
Δλ->SPAN	Performs a spectrum width search on the active trace, and sets the results to the measuring span.	
0nm SWEEP TIME ** sec	Sets sweep time used when a sweep is made in a 0 nm measuring span.	0(MINIMUM), 1 to 50 (1step)
VIEW SCALE-> MEAS SCALE	Sets the current display conditions to measuring conditions.	

LEVEL

Program Command	Description	Parameter ranges and supported variables.
REFERENCE LEVEL ***.*dBm	Sets the reference level valueused for LOG scaling.	–90.0 to 30.0dBm (0.1 step)
REFERENCE LEVEL ***.*##	Sets the reference level value used for LIN scaling	1.00pW to 1000mW (1.00 to 9.99 [pW, nW, μW, mW] :0.01 step 10.0 to 99.9(100)[pW, nW, μW, (mW)] :0.1 step 100 to 999 [pW, nW, μW, mW] : 1 step) ## is , pW, nW, μW, mW (select one of the above)
REFERENCE LEVEL @@@@@	Sets the value of variable @@@@@ to the reference level value	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, LM, L1, L2, L(CH), PKLVL, WDMLVL(CH), FLVI(CH), NFLVO(CH), NFASELV(CH), MKPWR, LAM1, LAM2, LAM3, LAM4
LEVEL SCALE **.*dB/D	Sets a level scale value.	0 (LINEAR), 0.1 to 10.0dB/DIV (0.1 step)
LEVEL SCALE @@@@@	Sets the value of variable @@@@@ to the level scale	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, L2-L1, SMSR, SMSR2, WDMSNR(CH), NFNF(CH), LAM2-LAM1, LAM4-LAM3
BASE LEVEL ****	Lower value for linear scale setting. Use units set under REF LEVEL. If exceeds 90% of upper units of scale, execution error results	0 to 900 (0.1 step)
PEAK->REF LEVEL	Sets peak level of the waveform on the active trace to the reference level value	
AUTO REF LEVEL ###	Executes after each sweep finishes. Selects ON/OFF for the <PEAK → REF LEVEL> function.	###: ON/OFF
LEVEL UNIT ######	Sets the unit of a level scale.	###: dBm, dBm/nm
Y SCALE DIVISION ##DIV	Sets the level scale division.	##: 8, 10, 12
REF LEVEL POSITION **DIV	Sets the position of the reference level on the level scale	0 to 12 (1 step)
SUB SCALE LOG **.*dB/D	Sets the sub scale value used for LOG scaling.	0.1 to 10.0dB/DIV (0.1 step)
SUB SCALELIN *.***/D	Sets the sub scale value used for LIN scaling.	0.005 to 1.250 (0.005 step)
SUB SCALE **.*dB/km	Sets the sub scale value used for dB/km scaling.	0.1 to 10.0 (0.1 step)
SUB SCALE **.*%/D	Sets the sub scale value used for %D scaling.	0.5 to 125.0 (0.1 step)
OFFSET LEVEL **.*dB	Sets the sub scale offset value used for LOG scaling	–99.9 to 99.9 (0.1 step)
OFFSET LEVEL ***.*dB/km	Sets the sub scale offset value used for dB/km scaling	–99.9 to 99.9 (0.1 step)
SCALE MINIMUM **.**	Sets the lower sub scale value used for linear scaling.	000 to 12.50 (0.01 step)
SCALE MINIMUM ***.*%	Sets the lower sub scale value used for &D scaling.	0.0 to 1250.0 (0.1 step)
LENGTH **.***km	Sets fiber length.	0.001 to 99.999 (0.001 step)
AUTO SUB SCALE ###	Automatically sets the sub scale from the calculated trac e waveform	###: ON/OFF
SUB REF LEVEL POSITION **DIV	Sets the position of the reference level on the sub level scale	0 to 12 (1 step)

SETUP

Program Command	Description	Parameter ranges and supported variables.
RESOLUTION WL *.****nm	Sets the wavelength resolution.	0.05 to 2.000 (1-2-5 step)
RESOLUTION WL @@@@@	Sets the value of variable @@@@@ to the wavelength resolution	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD
RESOLUTION FREQ ***GHz	Sets the frequency resolution.	10 to 400 (1-2-4 step)
RESOLUTION FREQ@@@@@	Sets the value of variable @@@@@ to the frequency resolution	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD
SENS NORMAL/HOLD	Sets the measuring sensitivity to NORMAL/HOLD	
SENS NORMAL/AUTO	Sets the measuring sensitivity to NORMAL/AUTO	
SENS NORMAL	Sets measuring sensitivity to NORMAL	
SENS MID	Sets measuring sensitivity to MID.	
SENS HIGH1	Sets measuring sensitivity to HIGH1	
SENS HIGH2	Sets measuring sensitivity to HIGH2	
SENS HIGH3	Sets measuring sensitivity to HIGH3	
CHOPPER #####	Switches chopper mode.	#####: OFF/SWITCH
AVERAGE TIMES ***	Sets the number of averaging times.	1 to 999 (1 step)
AVERAGE TIMES @	Sets the number of averaging times to the value of variable	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, M, N
SAMPLING POINT AUTO ###	Sets sampling points per sweep automatically.	###: ON/OFF
SAMPLING POINT *****	Sets sampling points per sweep.	101 to 100001 (1 step)
SAMPLING POINT @	Sets the sampling points to the variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, M, N
SAMPLING INTERVAL *.****nm	Sets the measurement sampling interval per sweep	0.002 to SPAN/101 (0.001 step)
SAMPLING INTERVAL @	Sets the sampline interval per sweep to the value of variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD, WAM2-WAM1, WAM4-WAM3
MEASURE WAVELENGTH AIR	Sets the measurement wavelength to an air wavelength	
MEASURE WAVELENGTH VACUUM	Sets measurement wavelength to a vacuum wavelength.	
X SCALE UNIT WAVELENGTH	Sets axis X to wavelength display mode.	

Program Command	Description	Parameter ranges and supported variables
X SCALE UNIT FREQUENCY	Sets axis X to frequency display mode.	
SWEEP SPEED ##	Sets the sweep speed	##: 1x/2x
PLS LIGHT MEASURE OFF	Turns OFF pulse light measurement mode	
PEAK HOLD****msec	Sets the HOLD time for PEAK HOLD pulse light measurement	****: 1 to 9999
EXTERNAL TRIGGER MODE	Sets external trigger mode	
EXTERNAL TRIGGER EDGE RISE	Detects the falling edge of an external trigger signal	
EXTERNAL TRIGGER EDGE FALL	Detects the rising edge of an external trigger signal	
EXTERNAL TRIGGER DELAY ****.*µs	After detection of an external trigger signal, and sets the delay time until data acquisition	0.0 to 1000.0 (0.1 step)
GATE MODE ****.*msec	Sets the sampling interval on the gate sampling mode	****.* : 0.1 to 1000.0 (0.1 step)
GATE LOGIC POSI	Sets the gate signal logic of gate sampling to the positive logic	
GATE LOGIC NEGA	Sets the gate signal logic of gate sampling to the negative logic	
SMOOTHING ###	Turns the smoothing function ON/OFF	###: ON/OFF
FIBER CORE SIZE #####	Switches the fiber core size mode.	#####: SMALL/LARGE
TRIGGER INPUT SAMPLING TRIGGER	Sets the trigger input mode tosampling trigger	
TRIGGER INPUT SWEEP TRIGGER	Sets the trigger input mode to sweeptrigger	
TRIGGER INPUT SAMPLING ENABLE	Sets the trigger input mode to sampling enable	
TRIGGER OUTPUT SWEEP STATUS	Sets the trigger output mode to sweep status	
TRIGGER OUTPUT OFF	Turns OFF the trigger output mode	
RESOLUTION CORRECTION ###	Turns the wavelength resolution correction function ON/OFF	###: ON/OFF

TRACE

Program Command	Description	Parameter ranges and supported variables.
ACTIVE TRACE #	Sets trace # to active trace.	#: A to G
DISPLAY #	Sets trace # to display mode.	#: A to G
BLANK #	Sets trace # to invisible mode.	#: A to G
WRITE #	Sets trace # to write mode.	#: A to G
FIX #	Sets trace # to data-fixing mode.	#: A to G
MAX HOLD #	Sets trace # to max. value detection mode.	#: A to G
MIN HOLD #	Sets trace # to min. value detection mode.	#: A to G
ROLL AVG # ***	Sets trace # to sequential addition averaging mode.	#: A to G, 2 to 100 (1 step)
C=A-B(LOG)	Sets trace C to TRACE A-B computation mode (LOG)	
C=B-A(LOG)	Sets trace C to TRACE B-A computation mode (LOG)	
C=A+B(LOG)	Sets trace C to TRACE A+B computation mode (LOG)	
C=A+B(LIN)	Sets trace C to TRACE A+B computation mode (LIN)	
C=A-B (LIN)	Sets trace C to TRACE A-B computation mode (LIN)	
C=B-A(LIN)	Sets trace C to TRACE B-A computation mode (LIN)	
C=1-k(A/B) k=*.****	Sets trace C to 1-k (TRACE A/B) computation mode	1.0000 to 20000.0000 (0.0001 step)
C=1-k(B/A) k=*.****	Sets trace C to 1-k (TRACE B/A) computation mode	1.0000 to 20000.0000 (0.0001 step))
F=C-D(LOG)	Sets trace F to TRACE C-D computation mode (LOG)	
F=D-C(LOG)	Sets trace F to TRACE D-C computation mode (LOG)	
F=C+D(LOG)	Sets trace F to TRACE C+D computation mode (LOG)	
F=D-E(LOG)	Sets trace F to TRACE D-E computation mode (LOG)	
F=E-D(LOG)	Sets trace F to TRACE E-D computation mode (LOG)	
F=D+E(LOG)	Sets trace F to TRACE D+E computation mode (LOG)	
F=C+D(LIN)	Sets trace F to TRACE C+D computation mode (LIN)	
F=C-D(LIN)	Sets trace F to TRACE C-D computation mode (LIN)	
F=D-C(LIN)	Sets trace F to TRACE D-C computation mode (LIN)	
F=D+E(LIN)	Sets trace F to TRACE D+E computation mode (LIN)	
F=D-E(LIN)	Sets trace F to TRACE D-E computation mode (LIN)	
F=E-D(LIN)	Sets trace F to TRACE E-D computation mode (LIN)	
F=POWER/NBW A **.*nm	Sets the power spectral density of trace A to be displayed on trace F	
F=POWER/NBW B **.*nm	Sets the power spectral density of trace B to be displayed on trace F	

Program Command	Description	Parameter ranges and supported variables.
F=POWER/NBW C **.*nm	Sets the power spectral density of trace C to be displayed on trace F	
F=POWER/NBW D **.*nm	Sets the power spectral density of trace D to be displayed on trace F	
F=POWER/NBW E **.*nm	Sets the power spectral density of trace E to be displayed on trace F	
G=C-F(LOG)	Sets trace G to TRACE C-F computation mode (LOG)	
G=F-C(LOG)	Sets trace G to TRACE F-C computation mode (LOG)	
G=C+F(LOG)	Sets trace G to TRACE C+F computation mode (LOG)	
G=E-F(LOG)	Sets trace G to TRACE E-F computation mode (LOG)	
G=F-E(LOG)	Sets trace G to TRACE F-E computation mode (LOG)	
G=E+F(LOG)	Sets trace G to TRACE E+F computation mode (LOG)	
G=C+F(LIN)	Sets trace G to TRACE C+F computation mode (LIN)	
G=C-F(LIN)	Sets trace G to TRACE C-F computation mode (LIN)	
G=F-C(LIN)	Sets trace G to TRACE F-C computation mode (LIN)	
G=E+F(LIN)	Sets trace G to TRACE E+F computation mode (LIN)	
G=E-F(LIN)	Sets trace G to TRACE E-F computation mode (LIN)	
G=F-E(LIN)	Sets trace G to TRACE F-E computation mode (LIN)	
G=NORM A	Sets the normalizd data of trace A to be displayed on trace G.	
G=NORM B	Sets the normalizd data of trace B to be displayed on trace G.	
G=NORM C	Sets the normalizd data of trace C to be displayed on trace G.	
G=CURVE FIT A **dB	Sets curve fit processed data from TRACE A to be displayed on trace G.	0 to 99 (1 step)
G=CURVE FIT B **dB	Sets curve fit processed data from TRACE B to be displayed on trace G.	0 to 99 (1 step)
G=CURVE FIT C **dB	Sets curve fit processed data from TRACE C to be displayed on trace G.	0 to 99 (1 step)
G=CURVE FIT PEAK A **dB	Sets peak fit processed data from TRACE A to be displayed on trace G.	0 to 99 (1 step)
G=CURVE FIT PEAK B **dB	Sets peak curve fit processed data from race B to be displayed on trace G.	0 to 99 (1 step)
G=CURVE FIT PEAK C **dB	Sets peak curve fit processed data from trace C to be displayed on trace G.	0 to 99 (1 step)
G=MARKER FIT **dB	Sets curve fit processed data from the placed delta marker to be displayed on trace G.	0 to 99 (1 step)
CVFIT OPERATION AREA####	Sets the target range for calculation when creating curve fit processed data.	####: ALL/IN L1-L2/OUT L1-L2
CURVE FIT/CURVE FIT PEAK ALGO ####	Sets the fitting function when creating a fitting function.	####:GAUSS/LORENZ/ 3RD POLY/4TH POLY/5TH POLY
TRACE #->＃	Copies data from TRACE of the variable @ to TRACE of the variable @	#: A to G
TRACE # CLEAR	Clears trace # data.	#: A to G
ALL TRACE CLEAR	Clears all trace data.	

ZOOM

Program Command	Description	Parameter ranges and supported variables.
ZOOM CENTER WL ****.***nm	Sets the display scale's center wavelength.	350.000 to 1750.000 (0.001 step)
ZOOM CENTER @@@@@	Sets the value of variable @@@@@ to the display scale center wavelength	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), PKWL, MEANWL, WDMWL(CH), NFWL(CH), WAM1, WAM2, WAM3, WAM4
ZOOM CENTER FREQ ***.****THz	Sets the display scale's center frequency.	171.0000 to 857.0000 THz (0.0001 step)
ZOOM CENTER FREQ @@@@@	Sets the value of variable @@@@@ to the display scale center frequency	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, PKWL, MEANWL, WDMWL(CH), NFWL(CH) , WAM1, WAM2, WAM3, WAM4
ZOOM SPAN WL ****.*nm	Sets the display scale's span.	0.5 to 1400.0 nm (0.1 step)
ZOOM SPAN WL @@@@@	Sets the value of variable @@@@@ to the display scale span	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD, WAM2-WAM1, WAM4-WAM3
ZOOM SPAN FREQ ***.**THz	Sets the display scale's span.	0.05 to 686.00 THz (0.01 step)
ZOOM SPAN FREQ @@@@@	Sets the value of variable @@@@@ to the display scale span	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, W2-W1, SPWD, WAM2-WAM1, WAM4-WAM3
ZOOM START WL ****.***nm	Sets the starting wavelength of the display scale.	1.000 to 1749.950 nm (0.001 step)
ZOOM START FREQ ***.****THz	Sets the starting frequency of the display scale.	10.0000 to 856.9950 THz (0.0001 step)
ZOOM STOP WL ****.***nm	Sets the ending wavelength of the display scale.	350.050 to 2450.000 nm (0.001 step)
ZOOM STOP FREQ ***.****THz	Sets the ending frequency of the display scale.	171.0050 to 999.9000 (0.0001 step)
PEAK->ZOOM CENTER	Sets the peak wavelength of the waveform on the active trace. Sets the wavelength to the display scale's center wavelength.	
OVERVIEW DISPLAY OFF	Sets OVERVIEW display during ZOOM toOFF	
OVERVIEW DISPLAY LEFT	Sets OVERVIEW display during ZOOM to the left side of the waveform screen	
OVERVIEW DISPLAY RIGHT	Sets OVERVIEW display during ZOOM to the right side of the waveform screen	
OVERVIEW SIZE LARGE	Sets OVERVIEW display during ZOOM to a large display	
OVERVIEW SIZE SMALL	Sets OVERVIEW display during ZOOM to a small display	
ZOOM INITIALIZE	Resets the display scale to the initial state.	

DISPLAY

Program Command	Description	Parameter ranges and supported variables.
NORMAL DISPLAY	Sets the screen into normal display mode.	
SPLIT DISPLAY	Sets the screen into split display mode.	
TRACE # UPPER	Sets trace # to the top screen of the split display.	#: A to G
TRACE # LOWER	Sets trace # to the bottom screen of the split display.	#: A to G
UPPER HOLD ###	Holds the top screen of the of split display.	###: ON/OFF
LOWER HOLD ###	Holds the bottom screen of the split display	###: ON/OFF
LABEL '---56 chars---'	Displays a label comment in the label area. If a semicolon (;) is added to the end, the comment (variable value) specified by the next LABEL command is displayed.	
LABEL @@@@@	Sets the contents of variable @@@@@ to the label area	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), W2-W1, LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, A$, B$, C$, D$, FILE$, TIME$, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
LABEL @@@@@;	Sets the contents of variable @@@@@ to the label display. The comment (variable value) specified by the next LABEL command is displayed.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), W2-W1, LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, A$, B$, C$, D$, FILE$, TIME$, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
LABEL CLEAR	Clears the LABEL command in the label area.	
NOISE MASK ***dB	Displays waveform data with the data at or below the set level masked	OFF (–999), –100 to 0 (1 step)
MASK LINE VERTICAL	Sets the mask value in the noise mask function or lower to zero.	
MASK LINE HORIZONTAL	Sets the mask value in the noise mask function or lower to the mask value.	
TRACE # CLEAR	Clears trace # data.	#: A to G
ALL TRACE CLEAR	Clears all trace data.	
DISPLAY OFF	Turns the display OFF	
DISPLAY ON	Turns the display ON	

MARKER

Program Command	Description	Parameter ranges and supported variables.
MARKER ****.***nm	Sets the marker to the specified wavelength position on the active trace (according to the wavelength value)	350.000 to 1750.000 (0.001 step)
MARKER ***.****THz	Sets the marker to the specified wavelength position on the active trace (according to the frequency value)	171.0000 to 857.0000 (0.0001 step)
MARKER @@@@@	Sets a marker to the wavelength position of variable @@@@@	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), MEANWL, PKWL, WDMWL(CH), NFWL(CH), WAM1, WAM2, WAM3, WAM4
SET MARKER ****	Sets fixed marker **** to the moving marker position	1 to 1024 (1 step)
SET MARKER @	Sets the fixed marker of variable @ to the moving marker position	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
CLEAR MARKER ****	Clears fixed marker ****.	1 to 1024 (1 step)
CLEAR MERKER @	Clears the fixed marker of variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, N, M
MARKER->CENTER	Sets the wavelength value of a marker to the measurement center wavelength.	
MARKER->ZOOMCENTER	Sets the wavelength value of a marker to the display scale’s center wavelength	
MARKER->REF LEVEL	Sets the marker level value to thereference level.	
ADV MARKER ##,****.***nm	Sets the advanced marker to the specified wavelength position (according to the wavelength value)	##:M1,M2,M3,M4 ****.***: 350.000 to 1750.000 (0.001 step)
ADV MARKER ##,***.****THz	Sets the advanced marker to the specified frequency position (according to the frequency value)	##:M1,M2,M3,M4 ****.***: 171.0000 to 857.0000 (0.001step)
ADV MARKER ##,@@@@@	Sets an advanced marker to the wavelength position of variable @@@@@	##:M1,M2,M3,M4, @@@@@: G,H,I,J,K,P,Q,R,S,X,Y,Z,WM,W1,W2,W(CH),WAM1,WAM2,WAM3,WAM4,MEANWL,PKWL,WDMWL(CH),NFWL(CH)
ADV MARKER TRACE ##,###	Sets the trace of advanced marker to ###	##:M1,M2,M3,M4 ###: TRA/TRB/TRC/TRD/TRE/TRF/TRG)
ADV MARKER SELECT ##,########	Sets the type of advanced marker to ###	##:M1,M2,M3,M4 ########: OFF/NORMAL/DENSITY/INTEGRAL
ADV MARKER INTEGRAL RANGE ##,***.*GHz	Sets the integration range of the advanced marker	##:M1,M2,M3,M4 ***.* : 1.0 to 999.9 (0.1 step)
ADV MARKER PEAK SEARCH ##	Detects the peak and sets the advanced marker	##:M1,M2,M3,M4
ADV MARKER BOTTOM SEARCH ##	Detects the bottom and sets the advanced marker	##:M1,M2,M3,M4
ADV MARKER NEXT SEARCH ##	Detects the next peak whose level is the current advanced marker position and sets the advanced marker	##:M1,M2,M3,M4
ADV MARKER SEARCH RIGHT ##	Detects the closest peak to the right of the current advanced marker position and sets the advanced marker	##:M1,M2,M3,M4
ADV MARKER SEARCH LEFT ##	Detects the closest peak to the left of the current advanced marker position and sets the advanced marker	##:M1,M2,M3,M4

Program Command	Description	Parameter ranges and supported variables.
ADV MARKER BANDWIDTH **.*nm	Sets the normalization bandwidth of the power spectral density marker	**.* : 0.1 to 10.0 (0.1 step)
ADV MARKER ALL CLEAR	Clears all advanced markers on the screen	
ALL MARKER CLEAR	Clears all markers from the screen.	
LINE MARKER1****.***nm	Sets line marker 1 to a specified wavelength position (according to a wavelength value).	350.000 to 1750.000 (0.001 step)
LINE MARKER1 ***.****THz	Sets line marker 1 to a specified frequency position (according to a frequency value).	171.0000 to 857.0000 (0.0001 step)
LINE MARKER1@@@@@	Sets line marker 1 to the wavelength position of vaiable @@@@@	@@@@@:E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), MEANWL, PKWL, WDMWL(CH), NFWL(CH), WAM1, WAM2, WAM3, WAM4
LINE MARKER2****.***nm	Sets line marker 2 to a specified wavelength position (according to a wavelength value).	350.000 to 1750.000 (0.001 step)
LINE MARKER2 ***.****THz	Sets line marker 2 to a specified frequency position (according to a frequency value).	171.0000 to 857.0000 (0.0001 step)
LINE MARKER2@@@@@	Sets line marker 2 to the wavelength position of variable @@@@@	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W(CH), MEANWL, PKWL, WDMWL(CH), NFWL(CH), WAM1, WAM2, WAM3, WAM4
LINE MARKER3****.***dB	Sets line marker 3 to a specified level.	–139.90 to 159.90 (0.01 step)
LINE MARKER3 ****.***dBm	Sets line marker 3 to a specified level.	–210.00 to 50.00 (0.01 step)
LINE MARKER3 *.***##	Sets line marker 3 to a specified level.	1.00pW to 1000mW (1.00 to 9.99[pW, nW, mW, mW]: 0.01 step 10.0 to 99.9(100) [pW, nW, mW, (mW)]: 0.1 step 100 to 999 [pW, nW, mW, mW]: 1 step) ## is , pW, nW, mW, mW (Select one of the above)
LINE MARKER3 **.***	Sets line marker 3 to a specified level.	0.00 to 2500.00 (0.01 step)
LINE MARKER3@@@@@	Sets line marker 3 to the level position of variable @@@@@	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, LM, L1, L2, L(CH), PKLVL, WDMLVL(CH), NFLVI(CH), NFLVO(CH), NFASELV(CH), MKPWR, LAM1, LAM2, LAM3, LAM4

Program Command	Description	Parameter ranges and supported variables.
LINE MARKER4****.***dB	Sets line marker 4 to a specified level.	–139.90 to 159.90 (0.01 step)
LINE MARKER4 ****.***dBm	Sets line marker 4 to a specified level.	–210.00 to 50.00 (0.01 step)
LINE MARKER4 *.***##	Sets line marker 4 to a specified level.	1.00pW to 1000mW (1.00 to 9.99[pW, nW, μW, mW]: 0.01 step 10.0 to 99.9(100) [pW, nW, μW, (mW)]: 0.1 step 100 to 999 [pW, nW, μW, mW]: 1 step) ## is , pW, nW, μW, mW(Select one of the above)
LINE MARKER4 **.***	Sets line marker 4 to a specified level.	0.00 to 2500.00 (0.01 step)
LINE MARKER4@@@@@	Sets line marker 4 to the level position of variable @@@@@	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, LM, L1, L2, L(CH), PKLVL, WDMLVL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), MKPWR
MARKER L1-L2->SPAN	Sets the range surrounded by line markers 1 and 2 to the measuring span.	
MARKER L1-L2->ZOOM SPAN	Sets the range surrounded by line markers 1 and 2 to the display scale span.	
LINE MARKER CLEAR	Clears line markers on the screen.	
MARKER OFFSET LIST	Displays the difference from the moving marker.	
MARKER SPACING LIST	Displays a difference to a neighboring marker.	
MARKER AUTO UPDATE ###	Makes the level position of a fixed marker follow the active trace waveform.	###: ON/OFF
MARKER UNIT nm	Sets a wavelength marker value to the wavelength display.	
MARKER UNIT THz	Sets a wavelength marker value to the frequency display.	
SEARCH/ANA L1-L2 ###	Selects ON/OFF for the analysis function in the range surrounded by line markers 1 and 2	###: ON/OFF
SEARCH/ANAZOOM AREA ###	Selects ON/OFF for the analysis function of the display scale range	###: ON/OFF

PEAK SEARCH

Program Command	Description	Parameter ranges and supported variables.
PEAK SEARCH	Performs a peak search on the active trace waveform	
BOTTOM SEARCH	Performs a bottom search on the active trace waveform	
NEXT SRCH	Searches for the next peak/bottom after the peak/bottom level of the active trace waveform	
NEXT SRCH RIGHT	Searches for the peak/bottom to the right of the peak/bottom marker of the active trace waveform	
NEXT SRCH LEFT	Searches for the peak/bottom to the left of the peak/bottom marker of the active trace waveform	
SET MARKER ****	Sets fixed marker to the moving marker **** position	1–1024 (1 step)
SET MARKER @	Sets the fixed marker of variable @ to the moving marker position	@: G, H, I, J, K, P, Q, R, S, X, Y, Z, S, N, M
CLEAR MARKER ****	Clears fixed marker ****.	1–1024 (1 step)
CLEAR MERKER @	Clears the fixed marker of variable @.	@: G, H, I, J, K, P, Q, R, S, X, Y, Z, S, N, M
ALL MARKER CLEAR	Clears all markers from the screen.	
AUTO SEARCH ###	Selects ON/OFF of the peak/bottom search function conducted each sweep	###: ON/OFF
MODE DIFF **.**dB	Sets the level difference of the mode judgment criteria used for peak search or waveform analysis.	0.01–50.00 (0.01 step)
SEARCH/ANA L1-L2 ###	Selects ON/OFF for the analysis function in the range surrounded by line markers 1 and 2	###: ON/OFF
SEARCH/ANAZOOM AREA ###	Selects ON/OFF for the analysisFunction of the display scale range	###: ON/OFF
SEARCH MODE ######	Sets the search mode	######: SINGLE/MULTI
MULTI SEARCH THRESH **.**dB	Sets the multi search threshold	0.01: 99.99 (0.01 step)
MULTI SEARCH SORT BY #####	Sets the multi search detection list sort order	#####: WL/LEVEL

ANALYSIS

Program Command	Description	Parameter ranges and supported variables.
SPEC WD THRESH **.**dB	Performs a THRESH-based spectrum width search according to the specified threshold value	0.01 to 50.00 (0.01 step)
PARAM THRESHK **.**	Sets the magnification for the THRESH based spectrum width search	1.00 to 10.00 (0.01 step)
PARAM THRESH MODE FIT ###	Turns ON/OFF the function that sets the marker to the peak of the mode when performing a THRESH-based spectrum width search.	###: ON/OFF
SPEC WD ENV **.**dB	Performs an envelope-based spectrum width search using the specified threshold value	0.01 to 50.00 (0.01 step)
PARAM ENV TH2 **.**dB	Sets the cutoff value for the envelope-based spectrum width search.	0.01 to 50.00 (0.01 step)
PARAM ENV K **.**dB	Sets the cutoff value for the envelope-based using the THRESH method.	1.00 to 10.00 (0.01 step)
SPEC WD RMS **.**dB	Performs an RMS-based spectrum width search according to a specified threshold.	0.01 to 50.00 (0.01 step)
PARAM RMS K **.**	Sets the magnification for an RMS-based spectrum width search	1.00 to 10.00 (0.01 step)
SPEC WD PEAK RMS **.**dB	Performs an RMS-based spectrum width search according to a specified threshold value	0.01 to 50.00 (0.01 step)
PARAM PEAK RMS K**.**	Sets the magnification for a PEAK-RMS-based spectrum width search	1.00 to 10.00 (0.01 step)
SPEC WD NOTCH **.**dB	Measures the NOTCH width using a specified threshold value	0.01 to 50.00 (0.01 step)
PARAM NOTCH K **.**	Sets a magnification based on notch width measurement	1.00 to 10.00 (0.01 step)
NOTCH FROM ######	Sets the reference for making notch width measurements.	#####: PEAK/BOTTOM
SMSR *	Sets the execution mode applied in SMR measurement	1, 2, 3, 4
SMSR MASK ±**.**nm	Sets a mask range close to the peak during SMSR1 measurements	0.00 to 99.99 (0.01 step)
POWER	Performs power analysis	
POWER OFFSET ***.**dB	Sets a correction value in power measurements	–10.00 to 10.00 (0.01 step)
DFB-LD ANALYSIS	Performs analysis necessary for DFB-LD.	
FP-LD ANALYSIS	Performs analysis necessary for FP-LD.	
LED ANALYSIS	Performs analysis necessary for LED.	
PMD ANALYSIS	Performs analysis necessary for PMD.	
PMD THRESH **.**dB	Sets a threshold value for PMD .analysis	0.01 to 50.00 (0.01 step)
WDM ANALYSIS	Performs analysis necessary for WMD.	
WMD THRESH **.**dB	Sets a threshold value for WDM analysis	0.1 to 99.9 (0.1 step)
WDM MODE DIFF **.**dB	Sets the minimum peak/bottom difference for channel detection during WDM analysis.	0.01 to 50.00 (0.01 step)

Program Command	Description	Parameter ranges and supported variables.
WDM DISPLAY MASK OFF	Cancels level threshold value settingwhen masking display channels	
WDM DISPLAY MASK ****.**dB	Sets the level threshold value when masking display channels	–100.00 to 0.00 (0.01 step)
WDM NOISE ALGO AUTO-FIXFIX	Sets noise level measuring algorithm to AUTO	
WDM NOISE ALGO MANUAL FIX	Sets noise level measuring algorithmto MANUAL FIX	
WDM NOISE ALGO AUTO CTR	Sets noise level measuring algorithmto AUTO CTR	
WDM NOISE ALGO MANUAL CTR	Sets noise level measuring algorithmto MANUAL CTR	
WDM NOISE ALGO PIT	Sets noise level measuring algorithmto PIT	
WDM NOISE AREA **.**nm	Sets an area used for noise level analysis in a range centered on channel wavelength.	0.01 to 10.00 (0.01 step)
WDM NOISE AREA @	Sets an area used for noise level analysis in a range of variable @ centered on channel channel wavelength.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
WDM MASK AREA **.**nm	Sets the signal light spectrum range to mask as centered on channel wavelength	0.01 to 10.00 (0.01 step)
WDM MASK AREA @	Sets the signal light spectrum range to mask as centered on channel wavelength, to the range of variable @	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
WDM FITTING ALGO LINEAR	Sets the fitting algorithm for finding noise level to linear interpolation mode	
WDM FITTING ALGO GAUSS	Sets the fitting algorithm for finding noise level to normal distribution curve mode	
WDM FITTING ALGO LORENZ	Sets the fitting algorithm for finding noise level to Lorenz curve mode	
WDM FITTING ALGO3RD POLY	Sets the fitting algorithm for finding noise level in 3rd polynomial mode.	
WDM FITTING ALGO4TH POLY	Sets the fitting algorithm for finding noise level in 4th polynomial mode	
WDM FITTING ALGO5TH POLY	Sets the fitting algorithm for finding noise level in 5th polynomial mode	
WDM NOISE BANDWIDTH *.**nm	Sets bandwidth applied in measuring noise	0.01 to 1.00 (0.01 step)
WDM DUAL TRACE ###	Makes setting so that both TRACEs A and B are used in analyzing WDM.	###: ON/OFF
WDM DISPLAY ABSOLUTE	Sets the display of WDM analysis results to absolute value display.	
WDM DISPLAY RELATIVE	Sets the display of WDM analysis results to relative value display.	
WDM DISPLAY DRIFT MEAS	Sets the display of WDM analysis results to drift value display (drift display using past measurement wavelength as a reference)	
WDM DISPLAY DRIFT GRID	Sets the display of WDM analysis results to drift value display (using grid wavelength as a reference)	

Program Command	Description	Parameter ranges and supported variables.
WDM CH RELATION #######:	Sets the display format of an inter-channel level absolute value when WDM analysis display is in absolute value display	####### OFFSET/SPACING
WDM REF CHANNELHIGHEST	Sets the reference channel when the CH RELATION is OFFSET to the channel with the highest level	
WDM CHANNEL NO.****	Sets the reference channel when the CH RELATION is OFFSET	1 to 1024 (1 step)
WDM MAX/MIN RESET	Resets MAX/MIN data during DRIFT (MEAS, GRID) measurement	
WDM OUTPUT SLOPE ###	Displays the least square approximation line of a channel peak.	###: ON/OFF
WDM POINT DISPLAY ###	Displays the range of data used in fitting on the waveform screen.	###: ON/OFF
WDM SIGNAL POWER########	Sets the signal optical power calculation method.	########: PEAK / INTEGRAL
WDM INTEGRAL RANGE ***GHz	Sets the signal optical power integral range.	1.0 to 999.9 (0.1step)
COLOR ANALYSIS	Performs a color analysis	
EDFA NF ANALYSIS	Performs analysis necessary for EDFA-NF measurements.	
EDFA NF THRESH **.**dB	Sets an EDFA-NF analysis threshold.	0.1 to 99.9 (0.1 step)
EDFA NF MODE DIFF **.**dB	Sets the minimum peak/bottom difference for channel detection during EDFA-NF analysis.	0.01 to 50.00 (0.01 step)
EDFA NF OFFSET(IN) ***.**dB	Sets a signal light offset value used for NF and Gain calculation	–99.99 to 99.99 (0.01 step)
EDFA NF OFFSET(IN)@@@@@	Sets the offset value of the signal used for NF and Gain calculation to the variable @@@@@	@@@@@:E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
EDFA NF OFFSET(OUT) ***.**dB	Sets an output light offset value used for NF and Gain calculation	–99.99 to 99.99 (0.01 step)
EDFA NF OFFSET(OUT) @@@@@	Sets an output light offset value used for NF and Gain calculation to the variable @@@@@	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
EDFA NF ASE ALGOAUTO FIX	Sets the ASE level measuring algorithm to ATUO FIX	
EDFA NF ASE ALGO MANUAL FIX	Sets the ASE level measuring algorithm to MANUAL FIX	
EDFA NF ASE ALGOAUTO CTR	Sets the ASE level measuring algorithm to AUTO CTR	
EDFA NF ASE ALGOMANUAL CTR	Sets the ASE level measuring algorithm to MANUAL CTR	
EDFA NF ASE AREA **.**nm	Sets an area used for ASE level analysis in a range centered on channel wavelength	0.01 to 10.00 (0.01 step)
EDFA NF ASE AREA @	Sets an area used for ASE level analysis in a range centered on variable @@@@@	@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
EDFA NF MASK AREA **.**nm	Sets the signal light spectrum range to mask as centered on channel wavelength	0.01 to 10.00 (0.01 step)
EDFA NF MASK AREA @	Sets the signal light spectrum range to mask as centered on variable @@@@@	@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
EDFA NF FITTING ALGO LINEAR	Sets the fitting algorithm for finding ASE level to linear interpolation mode	
EDFA NF FITTING ALGO GAUSS	Sets the fitting algorithm for finding ASE level to normal distribution curve mode	

Program Command	Description	Parameter ranges and supported variables.
EDFA NF FITTING ALGO LORENZ	Sets the fitting algorithm for finding ASE level to Lorenz curve mode	
EDFA NF FITTING ALGO 3RD POLY	Sets the fitting algorithm for finding ASE level in 3rd polynomial mode	
EDFA NF FITTING ALGO 4TH POLY	Sets the fitting algorithm for finding ASE level in 4th polynomial mode	
EDFA NF FITTING ALGO 5TH POLY	Sets the fitting algorithm for finding ASE level in 5th polynomial mode	
EDFA NF POINTDISPLAY ###	Displays the range of data used in fitting on the waveform screen.	###: ON/OFF
EDFA NF RES BWMEASURED	For the resolution, use the value determined from the waveform using THRESH 3dB analysis.	
EDFA NF RES BWCAL DATA	For the resolution, use the actual resolution value stored in the instrument.	
EDFA NF SHOT NOISE ###	Set whether to include/not include Shot Noise in the NF computation	###: ON/OFF
EDFA NF SIGNAL POWER ########	Sets the signal optical power calculation method.	########: PEAK / INTEGRAL
EDFA NF INTEGRAL RANGE ***.*GHz	Sets the signal optical power integral range.	1.0 to 999.9 (0.1step)
FILTER(PEAK) ANALYSIS	Performs optical filter (PEAK) analysis.	
FILTER(BOTTOM) ANALYSIS	Performs optical filter (BOTTOM) analysis.	
WDMFILTER(PEAK) ANALYSIS	Performs multi-channel type optical filter (PEAK) analysis	
WDM FILTER(BOTTOM) ANALYSIS	Performs multi-channel type optical filter (BOTTOM) analysis.	
SWITCH DISPLAY TO TRACE&TABLE	Displays both waveforms and tables in the display of analysis results.	
SWITCH DISPLAY TO TABLE	Displays only tables in the display of analysis results.	
SWITCH DISPLAY TO TRACE	Displays only traces in the display of analysis results.	
SWITCH DISPLAY TO GRAPH&TABLE	Displays both graphs and tables in the display of analysis results	
SWITCH DISPLAY TO GRAPH	Displays only graphs in the displayof analysis results	
AUTO ANALYSIS ###	Selects ON/OFF of the waveform analysis function activated each time a sweep is made	###: ON/OFF
ANALYSIS RESULT PRINT	Prints out analysis results.	
RESULT WRITE INT:'########.***'	Specifies a filter name and savesanalysis results to internal memory.	'########.***': File name
RESULT WRITE EXT:'########.***'	Specifies a file name and savesanalysis results to internal memory.	'########.***':File name
RESULT WRITE INT	Saves analysis results in internalmemory. File names are assignedautomatically.	
RESULT WRITE EXT	Saves analysis results in external memory. File names are assignedautomatically.	
RESULT WRITE INT @@	Specifies a file name and saves analysis results to internal memory under the file name specified in the variable @@.	@@: A$, B$, C$, D$
RESULT WRITE EXT @@	Specifies a file name and saves analysis results to floppy disk under the file name specified in the variable @@.	@@: A$, B$, C$, D$

Program Command	Description	Parameter ranges and supported variables.
SEARCH/ANA L1-L2 ###	Sets ON/OFF for the analysis function in the range surrounded by line markers 1 and 2.	###: ON/OFF
SEARCH/ANAZOOM AREA ###	Selects ON/OFF for the analysis function of the display scale range	###: ON/OFF

MEMORY

Program Command	Description	Parameter ranges and supported variables.
SAVE #->MEMORY **	Writes the contents of the selected TRACE from the specified memory number.	0 to 63 (1 step) #: A, B, C, D, E, F, G
SAVE #->MEMORY @	Writes the contents of the selected TRACE from the memory number in variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z #: A, B, C, D, E, F, G
RECALL MEMORY **->#	Reads the contents of the selected TRACE from the specified memory number.	0 to 63 (1 step) #: A, B, C, D, E, F, G
RECALL MEMORY @->#	Reads the contents of the selected TRACE from the memory number in variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z #: A, B, C, D, E, F, G
CLEAR MEMORY **	Clears trace data in the memory	0 to 63 (1 step)
CLEAR MEMORY @	Clears the trace data in the memoryspecified by the variable @/	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z

FILE

Program Command	Description	Parameter ranges and supported variables.
WRITE TRACE # INT:’ ########.***’	Assign a file name to specified TRACE data and save itto internal memory	#: A, B, C, D, E, F, G ’########.***’: file name
WRITE TRACE # EXT: ’ ########.***’	Assign a file name to specified TRACE data and save it in external memory	#: A, B, C, D, E, F, G ’########.***’: file name
WRITE TRACE # INT	Saves specified TRACE data in internal memory. File names are assigned automatically	#:A, B, C, D, E, F, G
WRITE TRACE # EXT	Saves specified TRACE data inexternal memory. File names are assigned automatically	#: A, B, C, D, E, F, G
WRITE TRACE # INT@@	Saves specified TRACE data in internal memory under the file namespecified in the variable @@.	#: A, B, C, D, E, F, G @@: A$, B$, C$, D$
WRITE TRACE # EXT @@	Saves specified TRACE data in external memory under the file namespecified in the variable @@.	#: A, B, C, D, E, F, G @@: A$, B$, C$, D$
TRACE WRITE:BINARY	Sets the data storage format to BINARY	
TRACE WRITE:CSV	Sets the data storage format to CSV	
WRITE ALL TRACE INT:’ ########.CSV’	Specify a file name for all TRACE data and save to internal memory.	’########.CSV’: File name
WRITE ALL TRACE EXT:’ ########.CSV’	Specify a file name for all TRACE data and save to external memory.	’########.CSV’: File name
WRITE ALL TRACE INT:	Save all TRACE data to internal memory. A file name is automatically assigned	
WRITE ALL TRACE EXT:	Save all TRACE data to external memory. A file name is automatically assigned	
WRITE ALL TRACE INT @@	Save all TRACE data under file names specified by the @@ variable to internal memory	@@: A$, B$, C$, D$
WRITE ALL TRACE EXT @@	Save all TRACE data under file names specified by the @@ variable to external memory	@@: A$, B$, C$, D$
WRITE MEMORY ** INT:’ ########.***’	Specifies a file name and savesthe memory data in internal memory	**: 0 to 63 (1 step) ’########.***’: file name
WRITE MEMORY ** EXT:’ ########.***’	Specifies a file name and savesthe memory data in external memory	**: 0 to 63 (1 step) ’########.***’: file name
WRITE MEMORY**INT	Saves memory data in internal memoryFile names are assigned automatically	**: 0 to 63 (1 step)
WRITE MEMORY ** EXT	Saves memory data in external memoryFile names are assigned automatically	**: 0 to 63 (1 step)
WRITE MEMORY ** INT @@	Saves memory data under the file name specified in the variable @@ in internal memory	**: 0 to 63 (1 step) @@:A$, B$, C$, D$
WRITE MEMORY ** EXT @@	Saves memory data under the file name specified in the variable @@ in external memory	**: 0 to 63 (1 step) @@:A$, B$, C$, D$
WRITE GRAPH INT:’ ########.***’	Specifies a file name and saves graphic data in internal memory.	’########.***’:File name
WRITE GRAPH EXT:’########.***’	Specifies a file name and saves graphic data in external memory.	’########.***’: file name
WRITE GRAPH INT	Saves graphic data in internal memoryFile names are assigned automatically	

Program Command	Description	Parameter ranges and supported variables.
WRITE GRAPH EXT	Saves graphic data in external memoryFile names are assigned automatically	
WRITE GRAPH INT @@	Saves graphic data under the file name specified by the variable @@ in internal memory.	@@: A$, B$, C$, D$
WRITE GRAPH EXT @@	Saves graphic data under the file name specified by the variable @@ in external memory.	@@: A$, B$, C$, D$
GRAPH COLOR MODE:	Sets the graphic color mode to black & white.	
GRAPH COLOR MODE:COLOR	Sets the graphic color mode to screencolor mode	
GRAPH COLOR MODE: PRESET COLOR	Sets the graphic color mode to PRESET COLOR (waveform in color, background in black & white)	
GRAPH TYPE:BMP	Sets the graphic file type to BMP	
GRAPH TYPE:TIFF	Sets the graphic file type to TIFF	
WRITE SETTING INT:'########.SX9'	Specifies a file name and savessetting data to internal memory.	'########.SX9': file name
WRITE SETTING EXT: '########.SX9'	Specifies a file name and savessetting data to external memory.	'########.SX9': file name
WRITE SETTING INT	Saves setting data to internal memoryFile names are assigned automatically	
WRITE SETTING EXT	Saves setting data to external memoryFile names are assigned automatically	
WRITE SETTING INT @@	Saves setting data under the file name specified in the variable @@ to internal memory.	@@: A$, B$, C$, D$
WRITE SETTINGEXT @@	Saves setting data under the file name specified in the variable @@ to external memory	@@: A$, B$, C$, D$
DATA:ADD WRITE	Writes an added data file	
DATA:OVER WRITE	Overwrites a data file	
DATA WRITE:CSV	Sets the data storage format to CSV	
DATA WRITE:DT9	Sets the data storage format to DT9	
WRITE DATA INT:' ########.***'	Specifies a file name and saves data to internal memory	'########.***': file name
WRITE DATA EXT:' ########.***'	Specifies a file name and saves datato external memory	'########.***': file name
WRITE DATA INT	Specifies a file name and saves datato internal memory. File names are assigned automatically.	
WRITE DATA EXT	Specifies a file name and saves data to external memory. File names are assigned automatically.	
WRITE DATA INT @@	Specifies a file name and saves dataunder the file name specified by the variable @@ in internal memory.	@@: A$, B$, C$, D$
WRITE DATA EXT @@	Specifies a file name and saves dataunder the file name specified by the variable @@ in external memory.	@@: A$, B$, C$, D$
DATA DATE&TIME ###	Selects ON/OFF of date and time output.	###: ON/OFF
DATA LABEL ###	Selects ON/OFF of label output.	###: ON/OFF
DATA DATA AREA ###	Selects ON/OFF of data area output.	###: ON/OFF
DATA CONDITION ###	Selects ON/OFF of measuring conditions output.	###: ON/OFF
DATA TRACE DATA ###	Selects ON/OFF of waveform data output.	###: ON/OFF
DATA OUTPUT WINDOW ###	Selects ON/OFF of contents output of the OUTPUT WINDOW PROGRAM function .	###: ON/OFF

Program Command	Description	Parameter ranges and supported variables.
READ TRACE # INT:’ ########.$$$’	Assigns a file name to specified TRACE data and reads it frominternal memory	’########.$$$’: file name #: A, B, C, D, E, F, G
READ TRACE # EXT:’ ########.$$$’	Assigns a file name to specified TRACE data and reads it from external memory	’########.$$$’: file name #: A, B, C, D, E, F, G
READ TRACE # INT @@	Reads TRACE data in the file namespecified by the variable @@ frominternal memory	#: A, B, C, D, E, F, G @@:A$, B$, C$, D$
READ TRACE # EXT @@	Reads TRACE data in the file namespecified by the variable @@ fromexternal memory	#: A, B, C, D, E, F, G @@:A$, B$, C$, D$
READ MEMORY **INT:’########.$$$’	Specifies a file name and readsmemory data from internal memory	’########.$$$’: file name 0 to 63 (1 step)
READ MEMORY ** EXT:’########.$$$’	Specifies a file name and readsmemory data from external memory	’########.$$$’: file name 0 to 63 (1 step)
READ MEMORY ** INT @@	Reads memory data in the file namespecified by the variable @@ frominternal memory	**: 0 to 63 (1 step) @@:A$, B$, C$, D$
READ MEMORY **	Reads memory data in the file namespecified by the variable @@ fromexternal memory	**: 0 to 63 (1 step) @@:A$, B$, C$, D$
READ SETTING INT:’ ########.$$$’	Specifies a file name and reads setting data from internal memory	’########.$$$’: file name
READ SETTING EXT:’ ########.$$$’	Specifies a file name and saves setting data from external memory	’########.$$$’: file name
READ SETTING INT @@	Reads setting data of the file name specified by the variable @@ from internal memory	@@: A$, B$, C$, D$
READ SETTING EXT @@	Reads setting data of the file name specified by the variable @@ from external memory	@@: A$, B$, C$, D$
READ DATA INT:’########.$$$’	Specifies a file name and reads datafrom internal memory	’########.$$$’: file name
READ DATA EXT:’########.$$$’	Specifies a file name and reads datafrom external memory	’########.$$$’: file name
READ DATA INT @@	Reads data in the file namespecified by thevariable @@ frominternal memory	@@: A$, B$, C$, D$
READ DATA EXT @@	Reads data in the file name specifiedby the variable @@ from externalmemory	@@: A$, B$, C$, D$
READ ALL TRACE INT:’########.CSV’	Specify a file name for all TRACE data and reads from internal memory.	’########.CSV’: File name
READ ALL TRACE EXT:’########.CSV’	Specify a file name for all TRACE data and reads from external memory.	’########.CSV’: File name
READ ALL TRACE INT @@	Reads all TRACE data under file names specified by the @@ variable from internal memory	@@: A$, B$, C$, D$
READ ALL TRACE EXT @@	Save all TRACE data under file names specified by the @@ variable from external memory	@@: A$, B$, C$, D$
READ TEMPLATE EXT:’########.$$$’	Specifies a file name and reads a templatefrom external memory	’########.$$$’: file name
READ TARGET LINE EXT:’########.$$$’	Specifies a file name and reads target line data from external memory	’########.$$$’: file name
DELETE INT:’########.$$$’	Deletes files in internal memory	’########.$$$’: file name
DELETE EXT:’########.$$$’	Deletes files in external memory	’########.$$$’: file name
DELETE INT @@	Deletes files specified by the variable @@ from internal memory	@@: A$, B$, C$, D$

Program Command	Description	Parameter ranges and supported variables.
DELETE EXT @@	Deletes files specified by the variable @@ from external memory	@@: A$, B$, C$, D$
RENAME INT:## @@	Changes the names of files in internal memory specified by the variable ## to the file name specified by the variable @@	##, @@: A$, B$, C$, D$
RENAME EXT:## @@	Changes the names of files in external memory specified by the variable ## to the file names specified by the variable @@	##, @@: A$, B$, C$, D$
REMOVE USB STORAGE	Brings USB storage media online	
WRITE LOGGING INT:#######.LX9	Specifies a file name and saves the logging data in internal memory	#######.LX9: file name
WRITE LOGGING EXT:#######.LX9	Specifies a file name and saves the logging data in external memory	#######.LX9: file name
WRITE LOGGING INT	Automatically assignes a file name and saves the logging data in internal memory	
WRITE LOGGING EXT	Automatically assignes a file name and saves the logging data in external memory	
WRITE LOGGING INT@@	Save logging data under file names specified by the @@ variable to internal memory	@@: A$: B$: C$: D$
WRITE LOGGING EXT@@	Save logging data under file names specified by the @@ variable to external memor	@@: A$: B$: C$: D$
LOGGING SAVE CSV###	Sets whether data logging results will be saved to a file in CSV format.	###: ON/OFF
LOGGING SAVE TRACE###	Sets whether temporary saved waveform files will be saved when data logging results is saved.	###: ON/OFF
READ LOGGING INT:#######.$$$	Specifies a file name and reads data from internal memory	#######.$$$: file name
READ LOGGING EXT:#######.$$$	Specifies a file name and reads data from external memory	#######.$$$: file name
READ LOGGING INT@@	Reads data in the file name specified by the variable @@ from internal memory	@@: A$: B$: C$: D$
READ LOGGING EXT@@	Reads data in the file name specified by the variable @@ from external memory	@@: A$: B$: C$: D$

ADVANCE

Program Command	Description	Parameter ranges and supported variables.
TEMPLATE GO/NO GO ##	Sets whether GO/NO-GO judgment is made	###: ON/OFF
TEMPLATE DISPLAY ###	Turns the template data display ON/OFF. ON: UPPER LINE=ONLOWER LINE=ONTARGET LINE=ON	###: ON/OFF
TEMPLATE DISPLAYUPPER ###	Sets ON/OFF of upper line display.	###: ON/OFF
TEMPLATE DISPLAYLOWER ###	Sets ON/OFF of lower line display.	###: ON/OFF
TEMPLATE DISPLAYTARGET ###	Sets ON/OFF of target line display.	###: ON/OFF
TMPLATE TEST TYPE UPPER	Sets if GO/NO-GO judgment at the upper line is made.	
TMPLATE TEST TYPE LOWER	Sets if GO/NO-GO judgment at the lower line is made.	
TMPLATE TEST TYPE UP & LOW	Sets if GO/NO-GO judgment at the upper and lower lines is made.	
TMPLATE WL SHIFT****.***nm	Sets the amount of wavelength shift of the template.	–999.999 to 999.999 (0.001 step)
TEMPLATE LEVEL SHIFT ***.**dB	Sets the amount of level shift of the template.	–99.99 to 99.99 (0.01 step)
DATA LOGGING START	Starts data logging	
DATA LOGGING ITEM########	Sets the data logging source	########: WDM, PEAK, MULTI-PEAK, DFBLD
DATA LOGGING MODE####	Sets the data logging mode (maximum channel mode or maximum logging mode)	# # # # : MODE1(MAX 1024 ch, 2001 times), MODE2(MAX 256 ch, 10001 times)
DATA LOGGING INTERVAL #######	Sets the measurement interval of data logging	####: SWEEP TIME, 1sec, 2sec, 5sec, 10sec, 30 sec, 1 min, 2 min, 5 min , 10min
DATA LOGGING TEST DURATION *******s	Sets the measurement duration of data logging (in seconds)	*******: 1 to 8639999(1 step)
DATA LOGGING PEAK TH TYPE###	Sets how the threshold for detecting the data logging mode (peak or bottom) is specified	###: ABS, REL
DATA LOGGING PEAK TH (ABS)****.**dBm	Sets the threshold (absolute value) for detecting the data logging mode	****.**: 20.00 to -100.00
DATA LOGGING PEAK TH (REL)**.**dB	Sets the threshold (relative value) for detecting the data logging mode	**.**: 0.01 to 99.99
DATA LOGGING CH MATCHING TH ± *.**nm	Sets the threshold of the channel-matching wavelength λ for data logging	*.**: 0.01 to 1.00
DATA LOGGING MEMORY ########	Sets the temporary area for saving waveform files of data logging	###: INTERNAL, EXTERNAL
DATA LOGGING TRACE LOGGING ###	Sets whether waveforms will be logged during data logging	###: ON/OFF

SYSTEM

Program Command	Description	Parameter ranges and supported variables.
OPTICAL ALIGNMENT	Aligns the optical axis of a monochromator optical system.	
SELF WL CALIBRATION	Sets the light source to be wavelength calibrated for the internal light source.	
EXT WL CALIBRATION ****.***nm	Sets the light source to be wavelength calibrated for the external light source (laser type)	350.000 to 1750.000 (0.001 step)
EXT-GAS WLCALIBRATION ****.***nm	Sets the light source to be wavelength calibrated for the external light source (gas cell type)	350.000 to 1750.000 (0.001 step)
EMIS LINE WL CALIBRATION****.***nm	Set the light source to be wavelength calibrated for the external light source(emission line light source type)	350.000 to 1750.000nm (0.001 step)
WL SHIFT **.***nm	Sets the amount of wavelength shift.	–5.000 to 5,000 (0.001 step)
LEVEL SHIFT ***.***dB SYSTEM GRID 200GHz	Sets the amount of level shift. Sets system grid to a 200 GHz spacing grid table.	–60.000 to 60,000 (0.001 step)
SYSTEM GRID 100GHz	Sets system grid to a 100 GHz spacing grid table.	
SYSTEM GRID 50GHz	Sets system grid to a 50 GHz spacing grid table.	
SYSTEM GRID 25GHz	Sets system grid to a 25 GHz spacing grid table.	
SYSTEM GRID 12.5GHz	Sets system grid to a 12.5 GHz spacing grid table.	
CUSTOM GRID STARTWL ****.****nm	Inputs the user grid table startwavelength.	1000.0000 to 1700.0000 (0.0001 step)
CUSTOM GRID STARTFREQ ***.****THz	Inputs the user grid table startfrequency.	176.3486 to 299.7924 (0.0001 step)
CUSTOM GRID STOP WL ****.****nm	Inputs the user grid table stopwavelength.	1000.0000 to 1700.0000 (0.0001 step)
CUSTOM GRID STOPFREQ ***.****THz	Inputs the user grid table stopfrequency.	176.3486 to 299.7924 (0.0001 step)
CUSTOM GRID SPACING ***.*GHz	Inputs the user grid table gridspacing.	0.1 to 999.9 (0.1 step)

Program Command	Description	Parameter ranges and supported variables.
GRID REFERENCE WL ****.****nm	Inputs the reference wavelength of the grid table.	350.0000 to 1750.0000 (0.0001 step)
GRID REFERENCE FREQ ***.*****THz	Inputs the reference frequency of the grid table.	171.0000 to 857.0000 (0.0001 step)
REMOTE INTERFACE: GP-IB	Sets the remote interface to GP-IB	
REMOTE INTERFACE: RS-232	Sets the remote interface to RS-232	
REMOTE INTERFACE: NETWORK(SOCKET)	Sets the remote interface to Ethernet (SOCKET)	
REMOTE INTERFACE: NETWORK(VXI-11)	Sets the remote interface to Ethernet (VXI-11)	
SELECT COLOR *	Selects the display color of the screen.	0 to 1 (1 step)
UNCAL WARNING DISPLAY ###	Displays UNCAL and warning.	###: ON/OFF
BUZZER CLICK ###	Turns the key press click soundON/ OFF	###: ON/OFF
BUZZER WARNING ###	Turns the warning/error buzzerON/ OFF	###: ON/OFF
LEVEL DISPLAY DIGIT *	Sets the number of displayed digits (decimal place) of the level data displayed under the marker area and ANALYSIS results.	1 to 3 (1 step)
WINDOW TRANSPARENT ###	Selects ON/OFF of the transparent display function for the split display and OVERVIEW window	###: ON/OFF
AUTO OFFSET ###	Turns auto offset ON/OFF.	###: ON/OFF
AUTO OFFSET INTERVAL *** min	Sets the time interval for executing the auto offset function	***: integer
TRIGGER INPUT SAMPLING TRIGGER	Sets the trigger input mode tosampling trigger	
TRIGGER INPUT SWEEP TRIGGER	Sets the trigger input mode to sweeptrigger	
TRIGGER INPUT SAMPLING ENABLE	Sets the trigger input mode to sampling enable	
TRIGGER OUTPUT SWEEP STATUS	Sets the trigger output mode to sweep status	
TRIGGER OUTPUT OFF	Turns OFF the trigger output mode	
REMOVE USB STORAGE	Brings USB storage media online	

Lists of Special Commands

General Commands

Program Command	Description	Parameter ranges and supported variables.
COPY ON	Produces a hard copy of the screen to file.	
GOTO ***	Makes a jump to line ***.	1 to 200 (1 step)
GOTO PROGRAM **	Makes a jump to program ** to run it from the first line. After completing running of program **, control returns to the original program. However, if there is an END command in program **, return to the jump source is not performed and the program ends. When a program is executed using this command, variables are not initialized.	
WAIT *****S	Makes a wait of **** seconds.	1 to 99999 (1 step)
PAUSE '---56 chars---'	Pauses execution of a program and causes a message window to appear. This window displays a message and an explanation of the CONTINUE key. Pressing the CONTINUE soft key closes the window and executes the program. If a program is started via GP-IB, no pause is made.	
VARIABLE CLEAR	Initializes all variables used in a program.	
END	Ends a program.	
INIT	Initializes all parameters, but does not clear variables.	
@=VAL(@$)	Converts the string in variable @ to a numerical value an substitutes the value into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z @$: A$, B$, C$, D$
BEEP **	Buzzer sounds for ** x 100 msec.	1 to 10 (1 step)
REM '---56 chars---'	Defines a comment in the program list. This command is not processed, and the program proceeds to the next line.	

Loop Control

Program Command	Description	Parameter ranges and supported variables.
N=********	Substitutes a value into variable N.	1 to 99999999 (1 step)
N=@@@@@@	Copies the contents of variable @@@@@ to variable N.	@@@@@: MODN, WDMCHN, NFCHN, GONO, M
N-N-1;IF N<>0GOTO ***	Subtracts "1" from variable N and, if the result is not "0," makes a jump toline ***.	1 to 200 (1 step)
M=********	Substitutes a value into variable M.	1 to 99999999 (1 step)
M=@@@@@@	Copies the contents of variable @@@@@ to variable M.	@@@@@: MODN, WDMCHN, NFCHN, GONO, N
M-M-1;IF M<>0 GOTO ***	Subtracts "1" from variable N and, if the result is not "0," makes a jump to line ***.	1 to 200 (1 step)

Variable Calculations

Program Command	Description	Parameter ranges and supported variables.
@ = ********** ###	Substitutes a value into variable @.For **********, a real number of 10 or fewer digits can be specified, including a sign and the decimal point.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, CH **********: -999999999 to 9999999999 (1 step) ###: nm, dB, dBm, pW, nW, mW, mW, W, THz, ' '(without units)
@ = # + **********	Adds value ********** to variable # and substitutes the value into variable @.********* can be specified with a real number of 10 or fewerdigits, including a sign and the decimal point. By specifying a negative value, you can cause subtraction to be made from variable #.	@, #: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, CH **********: –999999999 to 9999999999 (1 step)
@ = @@@@@	Copies the contents of variable @@@@@ to variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, CH @@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
@ = @@@@@ + ##### @ = @@@@@ - ##### @ = @@@@@ * ##### @ = @@@@@ / #####	Performs addition, subtraction, multiplication, and/or division between variables.	####: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, CH @@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
@$ = @$	Copies string variable @$ to stringvariable @.	@$: A$, B$, C$, D$
@$ = MID (@$, @, @)	Substitutes @'s worth of characters in the string that is distant from the start of character variable @$ by the number of characters in the numerical variable @ into character variable @$.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z @$: A$, B$, C$, D$
@$ = '---56 chars---'	Substitutes string to character variable @$. (56 chars max)	@$: A$, B$, C$, D$

Program Command	Description	Parameter ranges and supported variables.
@$ =@$+@$	Substitutes the character string obtained by concatenating character variable @$ and character variable @$ into character variable @$.	@$: A$: B$: C$: D$
@$ =STR(@)	Converts variable @ into a character string and substitutes it into character variable @$	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z @$: A$: B$: C$: D$
@$ =DATEINFO(###)	Substitutes the date and time into character variable @$.	@$: A$: B$: C$: D$ ####: DATE&TIME, DATE, TIME

Condition Judgement

Program Command	Description	Parameter ranges and supported variables.
IF F1 <= @@@@@ <= F2 GOTO ***	Value of variable @@@@@ is F1 or greaterIf less than F2, jumps to line ***	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), FNCHN, NFWL(CH), NFLVL(CH), NFLVI(CH), NFLVO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, A$, B$, C$, D$, FILE$, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3 ***: 1 to 200 (1 step)
F1 = ********** ###	Substitutes a value into variable F1. For **********, a real number of 10 or fewer digits can be specified including a sign and the decimal point.	###: nm, dB, dBm, pW, nW, mW, mW, W, THz, ' '(without units) **********: –999999999 to 9999999999 (1 step)
F2 = **********	Substitutes a value into ### variable F2. For **********, a real number of 10 or fewer digits can be specified, including a sign and the decimal point.	###: nm, dB, dBm, pW, nW, mW, mW, W, THz, ' '(without units) **********: –999999999 to 9999999999 (1 step)
F1 = @@@@@	Copies the contents of variable @@@@@ to the variable F1.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
F2 = @@@@@	Copies the contents of variable @@@@@ to the variable F2.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3

Program Command	Description	Parameter ranges and supported variables.
@ = LEVEL (****.***nm)	Substitutes the level of the point of wavelength ****.*** nm on an active trace into variable @.	@: G, H, I, J, K, P, Q, R, S, X, Y, Z ****.***: 350.000 to 1750.000 (0.001 step)
@ = LEVEL (@@@@@)	Substitutes the level of the point of the wavelength @@@@@ (variable) on the active trace into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z @@@@@: G, H, I, J, K, P, Q, R, S, X, Y, Z, WM, W1, W2, W(CH), MEANWL, PKWL, WDMLVL(CH), WDMWL(CH), NFWL(CH)
IF @@@@@ < @@@@@ GOTO *** IF @@@@@ =< @@@@@ GOTO *** IF @@@@@ = @@@@@ GOTO *** IF @@@@@ <> @@@@@ GOTO ***	Compares the large and small relationship of two variables and if the conditions are met, makes a jump to line ***.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), \ WDMSNR(CH), FNCHN, NFWL(CH), -NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3 ***: 1 to 200 (1 step),

External Control

Program Command	Description	Parameter ranges and supported variables.
SEND RS232 ’---56 chars---’	Sets the external instrumentthat is connected to the RS-232 connector as the listener, and sends the command in single quotes (’ ’).The delimiter is the set value of SET DELIMITER.	
SEND RS232 ’---56 chars---’;@	Sets the external instrument that isconnected to the RS-232 connector as the listener, and following the commnd in single quotes (’ ’), sends the value of variable @. The delimiter is the setting value of SET DELIMITER.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
SEND RS232 ’---20 chars---’; @;’---20 chars ---’	Sets the external instrument that isconnected to the RS-232 connector as the listener, and following the cmd. in single quotes (’ ’), sends the value of variable @, and also sends the cmd in single quotes. The delimiter is the setting value of SET DELIMITER.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
SEND LAN @$, ‘*****’ ; ‘---56 chars---’ *@$: computer name or IP address ****: Port number	Specifies the external instrument that is connected to the LAN connector and that is specified by the computer name, IP address, and port number as the listener, and sends the command and sends the command in single quotes (’ ’). Delimiter is value of SET DELIMITER	Port Numeber: 1024 to 65535 @ $: A$, B$, C$, D$

Program Command	Description	Parameter ranges and supported variables.
SEND LAN @$, '*****' ; '---56 chars---' ;@ *@$: computer name or IP address ****: Port number	Specifies the external instrument that is connected to the LAN connector and that is specified by the computer name, IP address,and port number as the listener, and sends the command and following the commnd in single quotes (' '), sends the value of the variable @. Delimiter is value of SET DELIMITER.	Port Number: 1024 to 65535 @: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z @ $: A$, B$, C$, D$
SEND LAN @$, '*****' , '--20 chars?';@ "?20 chars?'*@$: computer name or IP address ****: Port Number	Specifies the external instrument that isconnected to the LAN connector and that is specified by the computer name, IP address, and port number as the listener, and following the commnd in single quotes (' '), sends the value of variable @, as well as the command in single quotes. The delimiter is the setting value of SET DELIMITER.	Port Number: 1024 to 65535 @: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z @ $: A$, B$, C$, D$
SENDR RS-232 '---56 chars---' ; @$	Sends a query command to the external instrument connected to the RS-232 connector, and substitutes the message received from the external instrument into character variable @$. Up to 512 characters can be received. Delimiter is value of SET DELIMITER	@ $: A$, B$, C$, D$
SENDR LAN @$, ' ***** ' , '---56 chars---' ; @$ *@$: computer name or IP address ****: Port number	Sends a query command to the external instrument that is connected to the LAN connector and which is specified by the computer name, IP address, and port number stored in variable @$. Substitutes the message received from the external instrument into character variable @$. Up to 512 characters can be received. The delimiter is the setting value of SET DELIMITER.	Port Number: 1024 to 65535 @ $: A$, B$, C$, D$
SET DELIMITER ###	On the external instrument beingremotely controlled with the RS-232 or LAN port, sets the delimiter that is sent/received by the instrument.	###:CR, LF, CR+LF

Substitution of Measuring Conditions

Program Command	Description	Parameter Rng, Avail Variables
@ = CENTER	Substitutes the current measurement center wavelength into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
@ = SPAN	Substitutes the current sweep width into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
@ = REF LEVEL	Substitutes the current reference levelinto variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
@ = RESOLUTION	Substitutes the current measurement resolution into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
@ = SAMPLING POINT	Substitutes the current number of samples into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
@ = ZOOM CENTER	Substitutes the current display center wavelength into variable @.	@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z
@ = ZOOM SPAN	Substitutes the current display width into variable @.	E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z

User I/O

Program Command	Description	Parameter ranges and supported variables.
DATA INPUT '---56 chars---';@	Pauses program execution, and gets the value/string input into variable @by the user. The Input Window appears on screen displaying a character string in ' '. When variable @ is numerical it accepts numerical input and when it is a string variable it accepts string input.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, A$, B$, C$, D$
DATA OUTPUT	The string in single quotes (' ') is output to the OUTPUT WINDOW. If a semicolon is added to the end of the string, no line feed is made after output of the string, but a character string or the variable values specified by the next DATA OUTPUT command are output successively.	
DATA OUTPUT @@@@	The value of variable @@@@ is output to the OUTPUT WINDOWwith units added.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, A$, B$, C$, D$, FILE$, TIME$, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
DATA OUTPUT @@@@@;	Outputs the value of variable @@@@@ to the OUTPUT WINDOW with units added. After a string is output, no line feed is sent, but the value of the string or variable of the next DATA OUTPUT command is output.	@@@@@: E, G, H, I, J, K, O, P, Q, R,S, T, U, V, X, Y, Z, WM, W1, W2, W2-W1, W(CH), LM, L1, L2, L2-L1, L(CH), SPWD, MEANWL, PKWL, PKLVL, MODN, GONO, SMSR, SMSR2, WDMCHN, WDMWL(CH), WDMLVL(CH), WDMSNR(CH), NFCHN, NFWL(CH), NFLVLI(CH), NFLVLO(CH), NFASELV(CH), NFGAIN(CH), NFNF(CH), MKPWR, PMD, M, N, CH, A$, B$, C$, D$, FILE$, TIME$, WAM1, WAM2, WAM3, WAM4, WAM2-WAM1, WAM4-WAM3, LAM1, LAM2, LAM3, LAM4, LAM2-LAM1, LAM4-LAM3
DATA OUTPUT DATA AREA	Outputs the contents of the data area to the OUTPUT WINDOW.	
OUTPUT WINDOWCLEAR	Clears the contents of the OUTPUTWINDOW.	
OUTPUT WINDOW ###	Sets whether to display or hide the OUTPUT WINDOW on the screen.	###: ON or OFF

7.4 Controlling an External Instrument with the Program Function

Using the program function, the instrument can remote control the external devices which are connected by various interfaces. In addition, it is possible to remote control the multiple external devices by one program source.

Remote Control Using the RS-232 Port

Using the program function, the unit can send remote commands, receive talker data, and perform serial polling on the external device which is connected to the RS-232 port. Connect a cross cable to the RS-232 interface at the back side of the instrument. See chapter 4 for the various serial communication settings. If you want to receive query data from the external device, use the send/receive command. Query data is stored in the specified string variable @$.

Send Commands

```
SEND RS-232 ‘control command (56 chars)’
SEND RS-232 ’control command (56 chars)’ ;@
SEND RS-232 ’control command (20 chars)’ :@:’ control command (20 chars)’
```

Send/Receive Command

```
SENDRCV RS-232 ‘query command (56 chars)’;@$
```

Note

- Depending on the external device connected, there are times when it is necessary to change the delimiter setting of the send command. If the setting for the delimiter must be changed, use the SET SEND DELIMITER special command and make the setting match that of the instrument on the receiving end. (Default: CR+LF)
- Using a command such as SEND RS232 ’control command (56 characters)’;@, if you insert <wsp> between the command string and the variable @, add "␣" to the end of the command.

Remote Control of an External Instrument Using the LAN Port

Using the program function, specify the “Computer Name” or “IP address” and “Port Number” of the external device connected to the LAN connector to perform remote control. “Computer Name” or “IP address” must be entered it in the character variable @$ of the program command. If you want to receive query data from the external device, use the send/receive command. Query data is stored in the specified string variable @$.

Send Commands

SEND LAN @$ **** ‘control command (56 chars)’
SEND LAN @$ **** ’control command (56 chars)’ ;@
SEND LAN @$ **** ’control command (20 chars)’ :@:’control command (20 chars)
@$: Computer name or IP address
****: Port Number’

Send/Receive Command

SENDRCV LAN @$ **** ‘query command (56 characters)’
@$: computer name or IP address
****: Port number

Note

- Be sure to set the instrument’s IP address correctly.
- When using DHCP, the instrument’s IP address is automatically set. Set ADDRESS SETTING under TCP/IP SETTING to AUTO (DHCP).
- Please ask your network administrator for details about network connections.
- Using a command such as SEND LAN ’control command (56 characters)’;@, if you insert <wsp> between the command string and the variable @, add "␣" to the end of the command.

7.5 Sample Program

Here, an example is given of performing the operation below.

Conditions

After the measuring conditions have been set, the program performs a single sweep. Then it searches for a spectrum width and peak wavelength, and outputs the results to the label area and OUTPUT WINDOW. It repeats these operations ten times with a wait of three seconds between repetitions.

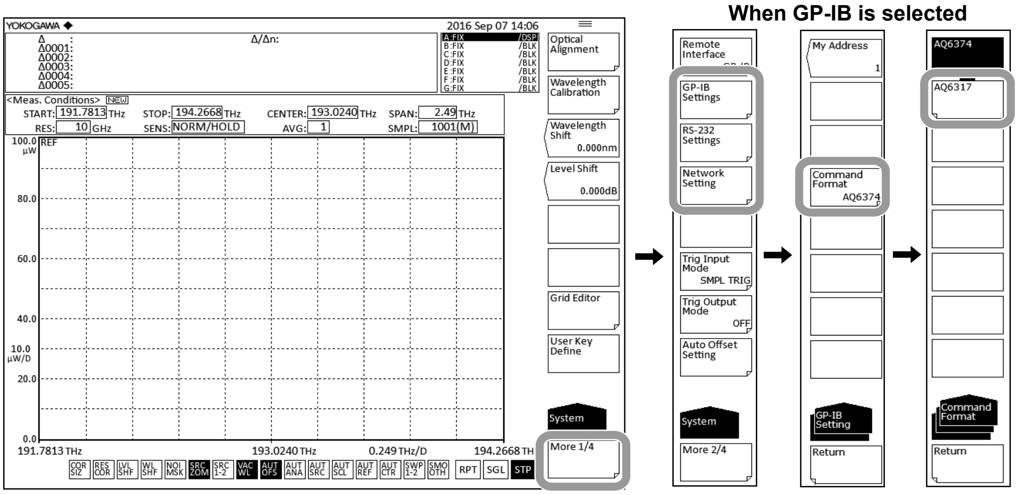
```
001  CENTER WL 1555.00nm                            :Set measurement conditions
002  SPAN 10.0nm
003  REFERENCE LEVEL -10.0dBm
004  RESOLUTION 0.1nm
005  AVERAGE TIMES 1
006  SENS NORMAL/HOLD
007  OUTPUT WINDOW CLEAR                            :Clear the OUTPUT WINDOW
                                                    data.
008  OUTPUT WINDOW ON                               :Display the OUTPUT
                                                    WINDOW.
009  N=10 :Set loop counter N to 10
010  SINGLE                                         :Set loop, counter N to 10
                                                    Perform a single sweep.
011  SPEC WD THRESH 20.0dB                          :Perform a spectrum width
                                                    search
012  DATA OUTPUT ‘Wd = ;                            :Output spectrum width
                                                    to OUTPUT WINDOW and the
                                                    label area.
013  LABEL ‘Wd = ;‘
014  DATA OUTPUT SPWD;
015  LABEL SPWD ;
016  PEAK SEARCH                                    :Perform a peak search
017  DATA OUTPUT ‘Pk = ;                            :Output the peak wavelength
                                                    value to OUTPUT WINDOW and
                                                    the label area.
018  LABEL ‘Pk = ;‘
019  DATA OUTPUT PKWL
020  LABEL PKWL
021  WAIT 3S                                        :Wait three second.
022  N=N-1 ; IF N <> 0 GOTO 10                      :Subtract 1 from loop
                                                    counter N and if the
                                                    result is not 0, make a
                                                    jump to line 010.
023  END                                            :Exit the Program.
```

Switching Command Modes

To use AQ6317-compatible commands, you must place the instrument in AQ6317 command mode.

Procedure

1. Press **SYSTEM**. The system setting menu is displayed.
2. Press the **More1/4** soft key. The communication interface setting menu is displayed.
3. Press the **GP-IB Settings**, **RS-232 Settings** or **Network Setting** soft key to specify each communication interface.
4. Press the **Command Format** soft key. The command format setting menu is displayed.
5. Press the **AQ6317** soft key.



Explanation

Because remote control via the GP-IB interface of the /AQ6374 complies with the IEEE 488.2 standard, it is not compatible with the conventional model AQ6317 (complying with the IEEE 488.1 standard) as to the remote commands and internal actions.
However, by placing the instrument in AQ6317-compatible command mode, you can use some of the AQ6317 commands. Status register operation also has compatibility with the AQ6317. When you switch the command mode, it causes all the contents of the status registers and queues and receive buffer and talker output buffer to be initialized.

Operation in AQ6317-Compatible Mode

The instrument operates as follows when it is remote controlled in the AQ6317-compatible mode.

- The majority of AQ6317 control commands and talker commands are available.
- Talker data is output in the AQ6317-compatible format.
- To send multiple commands at one time, use a comma “,” as a separator.
- If receiving multiple query commands in a single line, the instrument outputs only data relative to the last query command.

Switching Command Modes with Commands

The command mode can also be switched using the following GP-IB commands.
Commands to use when in AQ6374 mode (invalid in the AQ6317-compatible mode)

```
:SYSTem:COMMunicate:CFORmat<wsp><mode>
```

<mode> = AQ6317|AQ6374
AQ6317 = Switch to AQ6317-compatible mode
AQ6374 = Switch to AQ6374 mode

```
:SYSTem:COMMunicate:CFORmat?
```

0 = AQ6317
1 = AQ6374

Commands to use when in AQ6317-compatible mode (result in errors when in AQ6374 mode)

Control commands

```
CFORM*
```

*: 0 = AQ6317-compatible mode, 1 = AQ6374 mode

```
CFORM?
```

0 = AQ6317-compatible mode, 1 = AQ6374 mode

AQ6317 Status Byte

The status byte of AQ6317-compatible mode operates like the status byte in the AQ6317. Refer to the manuals for the AQ6317 series for the details of GP-IB.

Bit	Function and Setting Condition	Clear Timing
Bit 7	0	
Bit 6	Send an SRQ signal.	• Upon execution of serial polling • Upon receipt of DCL or SDC
Bit 5	When receiving data exceeding the receive buffer capacity of 512 byte “1” is set.	• Upon execution of serial polling • Upon receipt of DCL or SDC • At a start of measurement
Bit 4	0	
Bit 3	When a command data error occurs, set “1”.	• Upon receipt of DCL or SDC • Upon execution of serial polling • At a start of measurement
Bit 2	Warning error (including errors upon execution of a Program) occurs, set “1”. For the contents of the warning its number can be output	• When the warning error display disappears • Upon execution of serial polling • Upon receipt of DCL or SDC • At a start of measurement
Bit 1	When the execution of a copy or program terminates, set “1”.	• Upon execution of serial polling • Upon receipt of DCL or SDC • At a start of measurement
Bit 0	After sweep finishes, “1” is set.	• Upon execution of serial polling • Upon receipt of DCL or SDC • At a start of measurement

List of the AQ6317-Compatible Commands

For compatibility with the AQ6374, see the following table, AQ6317-Compatible Commands.

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
3D	×	-	
3DRCL	×	-	
A+BCL	○	:CALCulate:MATH:TRC<wsp>A+B(LIN)	
A=B	○	:TRACe:COPY<wsp>TRB,TRA	
A=C	○	:TRACe:COPY<wsp>TRC,TRA	
A-BC	○	:CALCulate:MATH:TRC<wsp>A-B(LOG)	
A-BCL	○	:CALCulate:MATH:TRC<wsp>A-B(LIN)	
ACTV*	○	:TRACe:ACTive<wsp><trace name> <trace name>=TRA|TRB|TRC	
ANA?	○	:CALCulate:DATA?	Diff. talker format
ANGL***	×	-	
AREA*	×	-	
ARES?	×	-	
ARESDSP*	×	-	
ATANA*	○	:CALCulate[:IMMediate]:AUTO<wsp> OFF|ON|0|1	
ATCTR*	○	:CALCulate:MARKer:MAXimum:SCENter:AUTO<wsp> OFF|ON|0|1	
ATOFS*	○	:CALibration:ZERO[:AUTO]<wsp> OFF|ON|0|1|	
ATREF*	○	:CALCulate:MARKer:MAXimum:SRLevel:AUTO	
ATSCL*	○	:DISPlay[:WINDow]:TRACe:Y2[:SCALe]:AUTO<wsp>OFF|ON|0|1	
ATSR*	○	:CALCulate:MARKer:AUTO<wsp> OFF|ON|0|1	
AUTO	○	:INITIate:SMODe<wsp>AUTO|3; INITiate	
AVG****	▲	:SENSe:AVERage:COUNt<wsp> <integer>	Diff. parameter range
B=A	○	:TRACe:COPY<wsp>TRA,TRB	
B=C	○	:TRACe:COPY<wsp>TRC,TRB	
B-AC	○	:CALCulate:MATH:TRC<wsp> B-A(LOG)	
B-ACL	○	:CALCulate:MATH:TRC<wsp>B-A(LIN)	
BASL***.*	○	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing<wsp>LINear|1; :DISPlay[:WINDow]:TRACe:Y1[:SCALe]:BLEVel<wsp><NRf>[MW]	
BD*	○	-	
BLKA	○	:TRACe:STATe:TRA<wsp>OFF|0	
BLKB	○	:TRACe:STATe:TRB<wsp>OFF|0	
BLKC	○	:TRACe:STATe:TRC<wsp>OFF|0	
BTSR	○	:CALCulate:MARKer:MINimum	

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
BZCLK*	○	:SYSTem:BUZZer:CLICk<wsp>OFF |ON|0|1	
BZWRN*	○	:SYSTem:BUZZer:WARNing<wsp> OFF|ON|0|1	
C=A	○	:TRACe:COPY<wsp>TRA,TRC	
C=B	○	:TRACe:COPY<wsp>TRC,TRB	
CLMES	○	-	

CLR	○	:TRACe:DELete<wsp>TRA;
		:TRACe:DELete<wsp>TRB;
		:TRACe:DELete<wsp>TRC	

CNDDT*	○	:MMEMory:STORe:DATA:ITEM<wsp> CONDition,OFF|ON|0|1	
COPY*	○	:HCOPY[:IMMediate]	
CRS*	○	-	
CTR=M	○	:CALCulate:MARKer:SCENter	
CTR=P	○	:CALCulate:MARKer:MAXimum:SCENter	
CTRF***.***	▲	:SENSe:WAVelength:CENTer<wsp> <NRf>[HZ]	Diff.parameter range
CTRWL****.**	▲	:SENSe:WAVelength:CENTer<wsp> <NRf>[M]	Diff. parameter range
CVFTC**	×	-	Same cmd for TRACE G
CVPKC**	×	-	Same cmd for TRACE G
CWPLS?	▲	-	Diff. query data 0: Except CW 1: CW
D&TDT*	○	:MMEMory:STORe:DATA:ITEM<wsp> DATE,OFF|ON|0|1	
DATE?	○	:SYSTem:DATE?	Diff. talker format
DATE YR.MO.DY	○	:SYSTem:DATE<wsp><year>,<month>,<day>	
TIME HH:MM	○	:SYSTem:TIME<wsp><hour>,<minute>, <second>	
DEFCL*	▲	:DISPlay:COLor<wsp><mode> <mode>=0: B&W, 1-5: mode 1 - mode 5	Diff. display color
DEL'@@@@.***'	○	:MMEMory:DELete<wsp><"file name">, EXTernal	
DFBAN	○	:CALCulate:CATegory<wsp>DFBLd|4	
DFBLD○;□;▲;****	▲	-	
DIR?	×	-	
DISP?	○	-	
DSPA	○	:TRACe:STATe:TRA<wsp>ON|1	
DSPB	○	:TRACe:STATe:TRB<wsp>ON|1	
DSPA?	○	:TRACe:STATe:TRA?	
DSPB?	○	:TRACe:STATe:TRB?	
DSPC	○	:TRACe:STATe:TRC<wsp>ON|1	
DSPC?	○	:TRACe:STATe:TRC?	
DTAD*	○	:MMEMory:STORe:DATA:MODE<wsp> ADD|OVER|0|1	
DTARA*	○	:MMEMory:STORe:DATA:ITEM<wsp> DATA,OFF|ON|0|1	
DUTCH***; ####.##	×	-	

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
DUTCHF***; ###.###	×	-	
DUTLEV**.**	×	-	
DUTSNR**.**	×	-	
EDFCVF*	×	-	
EDFTH**.*	×	-	
EDNF	×	-	
ENVK**.**	○	:CALCulate:PARameter[:CATegory]:SWENvelope:K<wsp><NRf>	
ENVT1**.**	○	:CALCulate:PARameter[:CATegory]:SWENvelope:TH1<wsp><NRf>[DB]	
ENVT2**.**	○	:CALCulate:PARameter[:CATegory]:SWENvelope:TH2<wsp><NRf>[DB]	
EXEC**	○	:PROGram:EXECute<wsp><integer>	
EXTRG	○	:TRIGger[:SEQuence]:STATe<wsp>OFF|ON|0|1	
FIG*	○	:UNIT:POWer:DIGit<wsp>1|2|3	
FILBTM○;□;▲;***	▲	:CALCulate:PARameter[:CATegory]:FILBtm<wsp><item>,<paramater>,<data>	
FILBTMAN	○	:CALCulate:CATegory<wsp>FILBtm|14	
FILPK○;□;▲;***	▲	:CALCulate:PARameter[:CATegory]:FILPk<wsp><item>,<paramater>,<data>	
FILPKAN	○	:CALCulate:CATegory<wsp>FILPk|13	
FIXA	○	:TRACe:ATTRibute:TRA<wsp>FIX|1	
FIXB	○	:TRACe:ATTRibute:TRB<wsp>FIX|1	
FIXC	○	:TRACe:ATTRibute:TRC<wsp>FIX|1	
FMKR***.****	▲	:CALCulate:MARKer:X<wsp>0,<NRf>[HZ]	Diff. parameter range
FPAN	○	:CALCulate:CATegory<wsp>FPLD|5	
FPLD;○;□;▲;****	▲	:CALCulate:PARameter[:CATegory]:FPLD<wsp><item>,<paramemter>,<data>	
GP2ADR**	○	:SYSTem:COMMunication:GP-IB2:ADDRess<wsp><integer>	
GRCOL*	▲	-	Valid only when the parameter is 0 or 1
GRFMT*	○	-	
HD*	○	-	
HELP*	×	-	
*IDN?	○	*IDN?	
INIT	○	:SYSTem:PRESet	
KABC	○	:CALCulate:MATH:TRC<wsp>1-K(A/B)	
KABCK*****.****	○	:CALCulate:MATH:TRC:K<wsp><NRf>	
KBAC	○	:CALCulate:MATH:TRC<wsp>1-K(B/A)	
KYDNE	×	-	
L1FMK***.****	▲	:CALCulate:LMARker:X<wsp>1,<NRf>[HZ]	Diff. parameter range
L1MK****.*	▲	:CALCulate:LMARker:X<wsp>1,<NRf>[M]	Diff. parameter range

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
L1MK?	▲	:CALCulate:LMARker:X?<wsp>1	Diff. parameter range
L2FMK****.****	▲	:CALCulate:LMARker:X<wsp>2,<NRf> [HZ]	Diff. parameter range
L2MK****.***	▲	:CALCulate:LMARker:X<wsp>2,<NRf> [M]	Diff. parameter range
L2MK?	▲	:CALCulate:LMARker:X?<wsp>2	Diff. parameter range
L3DB****.**	▲	:CALCulate:LMARker:Y<wsp>3,<NRf> [DB]	Diff. parameter range
L3DBM****.**	▲	:CALCulate:LMARker:Y<wsp>3,<NRf> [DBM]	Diff. parameter range
L3LN*.***E±**	▲	:CALCulate:LMARker:Y<wsp>3,<NRf>	Diff. parameter range
L3MK?	▲	:CALCulate:LMARker:Y?<wsp>3	Diff. parameter range
L4DB****.**	▲	:CALCulate:LMARker:Y<wsp>4,<NRf> [DB]	Diff. parameter range
L4DBM****.**	▲	:CALCulate:LMARker:Y<wsp>4,<NRf> [DBM]	Diff. parameter range
L4LN*.***E±**	▲	:CALCulate:LMARker:Y<wsp>4,<NRf>	Diff. parameter range
L4MK?	▲	:CALCulate:LMARker:Y?<wsp>4	Diff. parameter range
LBL '*********'	▲	:DISPlay[:WINDow]:TEXT:DATA<wsp> <string>	Diff. no. of chars
LBLCL	○	:DISPlay[:WINDow]:TEXT:CLEar	
LBLDT*	○	:MMEMory:STORe:DATA:ITEM<wsp> LABel,OFF|ON|0|1	
LCALT****;#.###	▲	:CALibration:POWer:OFFSet:TABLe <wsp><integer>,<NRf> [DB]	Diff. parameter range
LDATAR****-R**** LDATBR****-R**** LDATCR****-R**** WDATAR****-R**** WDATBR****-R**** WDATCR****-R**** DTNUM A DTNUM B DTNUM C	○	:TRACe[:DATA]:X?<wsp><trace name> [,<start point>,<stop point>] :TRACe[:DATA]:Y?<wsp><trace name> [,<start point>,<stop point>] :TRACe[:DATA]:SNUMber?<wsp> <trace name>	
LMEM$$R****-R**** WMEM$$R****-R**** DTNUM **	○		

List of the AQ6317-Compatible Commands

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
LDTDIG*	○	-	
LEDo;□;▲;****	▲	:CALCulate:PARameter[:CATegory]: LED<wsp><item>,<paramater>,<data>	
LEDAN	○	:CALCulate:CATegory<wsp>LED|6	
LHLD*	○	:DISPlay[:WINDow]:SPLit<wsp>ON|1; :DISPlay[:WINDow]:SPLit:HOLD: LOWer<wsp>OFF|ON|0|1	
LMKCL	○	:CALCulate:LMARker:AOFF	
LNGT**.***	○	:DISPlay[:WINDow]:TRACe:Y2[:SCALe]: LENGth<wsp><NRf>[KM]	
LOFSKM***.*	○	:DISPlay[:WINDow]:TRACe:Y2[:SCALe]: OLEVel<wsp><NRf>[DB/KM]	
LOFST***.*	○	:DISPlay[:WINDow]:TRACe:Y2[:SCALe]: OLEVel<wsp><NRf>[DB]	
LOGLMT***	×	-	
LPF	×	-	
LSCL**.*	○	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]: SPACing<wsp>LOGarighmic|0; :DISPlay[:WINDow]:TRACe:Y1[:SCALe]: PDIVision<wsp><integer>[DIV]	
LSUNT*	○	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]: UNIT<wsp>DBM|DBM/NM	
LTABS	×	-	
LTALM?	×	-	
LTALMDT?	×	-	
LTATSCL*	×	-	
LTATSET	×	-	
LTCH***	×	-	
LTCHCUR***	×	-	
LTINTVL****.*	×	-	
LTL	×	-	
LTLHI***.**	×	-	
LTLLOW***.**	×	-	
LTLVLCTR***.**	×	-	
LTLVLSCL**.*	×	-	
LTREFINI	×	-	
LTREFSET	×	-	
LTREL	×	-	
LTSNR	×	-	
LTSNRCTR***.**	×	-	
LTSNRLIM**.**	×	-	
LTSNRSCL**.*	×	-	
LTSWP	×	-	
LTTIME****	×	-	
LTTMCUR****	×	-	
LTWL	×	-	
LTWLCTR****.**	×	-	
LTWLLIM**.**	×	-	
LTWLSPN****.*	×	-	
LVSFT***.**	○	:SENSe:CORRection:LEVel:SHIFt<wsp> <NRf>[DB]	

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
MAXA	○	:TRACe:ATTRibute:TRA<wsp>MAX|2	
MCLR***	▲	:CALCulate:MARKer[:STATe]<wsp><marker>,OFF|0	Diff. parameter range
MEM*	×	-	
MESWL*	○	:SENSe:CORRection:RVELocity:MEDium<wsp>AIR|VACuum|0|1	
MIMSK**.**	×	-	
MINB	○	:TRACe:ATTRibute:TRB<wsp>MIN|3	
MKCL	○	:CALCulate:MARKer:AOFF	
MKR***	▲	:CALCulate:MARKer[:STATe]<wsp><marker>,|ON|1	Diff. parameter range
MKR?	○	:CALCulate:MARKer:X?<wsp>0	
MKR?****	○	:CALCulate:MARKer:X?<wsp><marker>	Diff. parameter range
MKR1	○	:CALCulate:MARKer[:STATe]<wsp>1,|ON|1	
MKR1?	○	:CALCulate:MARKer:X?<wsp>1	
MKR2	○	:CALCulate:MARKer[:STATe]<wsp>2,|ON|1	
MKR2?	○	:CALCulate:MARKer:X?<wsp>2	
MKROS*	○	:CALCulate:MARKer:FUNCtion:FORMat<wsp>OFFSet|SPACing|0|1	
MKRPRT	○	:HCOPY[:IMMediate]:FUNCtion:MARKer:LIST	
MKRUP*	○	:CALCulate:MARKer:FUNCtion:UPDate<wsp>OFF|ON|0|1	
MKUNT*	○	:CALCulate:MARKer:UNIT<wsp>WAVelength|FREQuency|0|1	
MLTMKR*	×	-	
MODFT*	○	:CALCulate:PARameter[:CATegory]:SWTHresh:MFIT<wsp>OFF|ON|0|1	
MODIF**.**	○	:CALCulate:PARameter:COMMon:MDIFf<wsp><NRf>[DB]	
MSKL*	○	:DISPlay[:WINDow]:TRACe:Y:NMASk:TYPE<wsp>VERTical|HORIzontal|0|1	
NCHMOD*	○	:CALCulate:PARameter[:CATegory]:NOTCh:TYPE<wsp>PEAK|BOTTom|0|1	
NCHTH**.*	○	:CALCulate:PARameter[:CATegory]:NOTCh:TH<wsp><NRf>[DB]	
NMSK****	▲	:DISPlay[:WINDow]:TRACe:Y:NMASk<wsp><NRf>[DB]	Diff. parameter range
NORMC	×	-	Same cmd for TRACE G
GNORMD	○	:DISPlay[:WINDow]:SPLit<wsp>OFF|0	
NSR	○	:CALCulate:MARKer:MAXimum:NEXT or :CALCulate:MARKer:MINimum:	
NEXTNSRL	○	:CALCulate:MARKer:MAXimum:LEFT or :CALCulate:MARKer:MINimum:LEFT	
NSRR	○	:CALCulate:MARKer:MAXimum:RIGHt or :CALCulate:MARKer:MINimum:RIGHt	

List of the AQ6317-Compatible Commands

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
OFIN***.**	×	-	
OFOUT***.**	×	-	
OPALIGN	○	:CALibration:ALIGn[:IMMediate]	
PKHLD****	○	-	
PKSR	○	:CALCulate:MARKer:MAXimum	
PKSR?	○	-	
PLMES	×	-	• PKHLD**** when PEAK HOLD MODE • EXTRG when EXT TRIGGER MODE
PLMOD?	○	-	
PLMSK**.**	×	-	
PMD	○	:CALCulate:CATegory<wsp>PMD|9	
PMDTH**.**	○	:CALCulate:PARameter[:CATegory]:PMD:TH<wsp><NRf>[DB]	
PMRPT	×	-	
PMRST	×	-	
PMSGL	×	-	
PMSTP	×	-	
PMST?	×	-	
PMUNT*	×	-	
POFS**.**	○	:CALCulate:PARameter[:CATegory]:POWer:OFFSet<wsp><NRf>[DB]	
PRDEL**	○	-	
PREXT	○	-	
PRFED**	×	-	
PRMK**.**	○	:CALCulate:PARameter[:CATegory]:SWPKrms:K<wsp><NRf>	
PRMTH**.*	○	:CALCulate:PARameter[:CATegory]:SWPKrms:TH<wsp><NRf>[DB]	
PWR	○	:CALCulate:CATegory<wsp>POWer|8	
RAVA***	○	:TRACe:ATTRibute:RAVG[:TRA]<wsp><integer>	
RAVB***	○	:TRACe:ATTRibute:RAVG:TRB<wsp><integer>	
RCLA**	▲	:MEMory:LOAD<wsp><integer>,TRA	Diff. parameter range
RCLB**	▲	:MEMory:LOAD<wsp><integer>,TRB	Diff. parameter range
RCLC**	▲	:MEMory:LOAD<wsp><integer>,TRC	Diff. parameter range
RD*'@@@@'	○	:MMEMory:LOAD:TRACe<wsp><trace name>,<"file name">,EXTernal <trace name>=TRA|TRB|TRC	Loads external memory
RD3D*'@@@@'	×	-	
RDDT'@@@@'	○	:MMEMory:LOAD:DATA<wsp><"file name">,EXTernal	Loads external memory
RDLT'@@@@'	×	-	
RDMEM** '@@@@'	○	:MMEMory:LOAD:MEMory<wsp><integer>,<"file name">,EXTernal	Loads external memory
RDPRG** '@@@@'	○	:MMEMory:LOAD:PROGram<wsp><program number>,<"file name">,EXTernal	Loads external memory
RDSET'@@@@'	○	:MMEMory:LOAD:SETTing<wsp><"file ame">,EXTernal	Loads external memory

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
RDTMP'@@@@'	▲	:MMEMory:LOAD:TEMPlate<wsp><template>,<"file name">,EXTernal <template> = UPPer|LOWer|TARGet	Loads external memory
REF = M	○	:CALCulate:MARKer:SRLevel	
REF = P	○	:CALCulate:MARKer:MAXimum:SRLevel	
REFL***.*	▲	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing<wsp>LOGarighmic|0; :DISPlay:[:WINDow]:TRACeY1[:SCAle]:RLEVel<wsp><NRf>[DBM]	Diff. parameter range
REFLM*.**	○	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing<wsp>LINear|1; :DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RLEVel<wsp><NRf>[MW]	
REFLN*.**	○	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing<wsp>LINear|1; :DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RLEVel<wsp><NRf>[NW]	
REFLP*.**	×	-	
REFLU*.**	○	:DISPlay[:WINDow]:TRACe:Y1[:SCALe]:SPACing<wsp>LINear|1; :DISPlay[:WINDow]:TRACe:Y1[:SCALe]:RLEVel<wsp><NRf>[UW]	
REFL?	▲	:DISPlay:[:WINDow]:Y1[:SCAle]:RLEVel?	Diff. parameter range
REL*	×	-	
RESCOR*	○	-	
RESLN*.**	▲	:SENSe:BANDwidth|:BWIDth[:RESolution]<wsp><NRf>[M]	Diff. parameter range
RESLNF***	×	-	
RMSK**.**	○	:CALCulate:PARameter[:CATegory]:RMS:K<wsp><NRf>	
RMSTH**.*	○	:CALCulate:PARameter[:CATegory]:RMS:TH<wsp><NRf>[DB]	
RPT	○	:INITIate:SMODe<wsp>REPeat|2; INITiate	
*RST	▲	*RST	Diff. operation
SAVEA**	▲	:MEMory:STORe<wsp><integer>,TRA	Diff. parameter range
SAVEB**	▲	:MEMory:STORe<wsp><integer>,TRB	Diff. parameter range
SAVEC**	▲	:MEMory:STORe<wsp><integer>,TRC	Diff. parameter range
SENS?	○	:SENSe:SENSe?	0 if SENS is set to NORMAL
SD*	○	-	
SEGP****	▲	:SENSe:SWEep:SEGMent:POINts<wsp><integer>	Diff. parameter range
SGL	○	:INITIate:SMODe<wsp>SINGle|1	

List of the AQ6317-Compatible Commands

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
SHI1	▲	:SENSe:SENSe<wsp>HIGH1|3; :SENSe:CHOPer<wsp>OFF|0	Chopper Unused
SHI2	▲	:SENSe:SENSe<wsp>HIGH2|4; :SENSe:CHOPer<wsp>OFF|0	Chopper Unused
SHI3	▲	:SENSe:SENSe<wsp>HIGH3|5; :SENSe:CHOPer<wsp>OFF|0	Chopper Unused
SKM**.*	○	DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:UNIT<wsp>DB/KM|2 DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:PDIVision<wsp><NRf>[DB/KM]	
SLIN*.***	○	DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:UNIT<wsp>LINear|1 DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:PDIVision<wsp><NRf>	
SLOG**.*	○	DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:UNIT<wsp>DB|0 DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:PDIVision<wsp><NRf>[DB]	
SMEAS	○	:INITIate:SMODe<wsp>SEGment|4	
SMID	○	:SENSe:SENSe<wsp>MID|2	
SMIN***.*	○	:DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:SMINimum<wsp><NRf>	
SMINP***.*	○	:DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:SMINimum<wsp><NRf>[%]	
SMPL****	▲	:SENSe:SWEep:POINts<wsp><integer>	Diff.parameter range
SMSR*	○	:CALCulate:PARameter[:CATegory] :SMSR:MODE<wsp>SMSR1|SMSR2	
SNAT	○	:SENSe:SENSe<wsp>NAUT|1	
SNHD	○	:SENSe:SENSe<wsp>NHLD|0	
SP = LM	○	:CALCulate:LMARker:SSPan	
SPAN****.*	▲	:SENSe:WAVelength:SPAN<wsp><NRf>[M]	Diff. parameter range
SPANF***.***	▲	-	Diff. parameter range
SPLIT	○	:DISPlay[:WINDow]:SPLit<wsp>ON|1	
SPN = W	○	-	
SPS***.*	○	DISPlay[:WINDow]:TRACe:Y2[:SCALe]:UNIT<wsp>%|3 DISPlay[:WINDow]:TRACe:Y2[:SCALe]:PDIVision<wsp><NRf>[%]	
SRLMK*	○	:CALCulate:LMARker:SRANge<wsp>OFF|ON|0|1	
SRMSK***	○	-	
SRQ*	○	*SRE<wsp><integer>	
SSE*	×	-	
SSMSK**.**	○	:CALCulate:PARameter[:CATegory]:SMSR:MASK<wsp><NRf>[M]	
SSUNT?	○	:DISPlay[:WINDow]:TRACe:Y2 [:SCALe]:UNIT?	

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
STAF***.***	▲	:SENSe:WAVelength:STARt<wsp><NRf>[HZ]	Diff. parameter range
STAWL****.**	▲	:SENSe:WAVelength:STARt<wsp><NRf>[M]	Diff. parameter range
STP	○	:ABORt	
STPF***.***	▲	:SENSe:WAVelength:STOP<wsp><NRf>[HZ]	Diff. parameter range
STPWL****.**	▲	:SENSe:WAVelength:STOP<wsp><NRf>[M]	Diff. parameter range
SW*	○	:CALCulate:CATegory<wsp>SWTHresh|0	
SWDSP*	×	-	
SWENV**.**	○	:CALCulate:PARameter[:CATegory]:SWENvelope:TH1<wsp><NRf>[DB]	
SWEEP?	○	-	
SWPI*****	○	:SENSe:SWEep:TIME:INTerval<wsp><integer>[SEC]	
SWPM*	○	:SENSe:WAVelength:SRANge<wsp>OFF|ON|0|1	
SWPRM**.**	○	:CALCulate:PARameter[:CATegory]:SWPKrms:TH<wsp><NRf>[DB]	
SWRMS**.**	○	:CALCulate:PARameter[:CATegory]:RMS:TH<wsp><NRf>[DB]	
SWTHR**.**	○	:CALCulate:PARameter[:CATegory]:SWTHresh:TH<wsp><NRf>[DB]	
THRK**.**	○	:CALCulate:PARameter[:CATegory]:SWTHresh:K<wsp><NRf>	
THRTH**.**	○	:CALCulate:PARameter[:CATegory]:SWTHresh:TH<wsp><NRf>[DB]	
TIME?	○	-	
TLDAT*****.**; ***.**;***.**	×	:TRACe:TEMPlate:DATA<wsp><template>,<wavelength>[M],<level>[DB]	
TLDATCLR	▲	:TRACe:TEMPlate:DATA:ADELete<wsp><template> <template>=UPPer|LOWer|TARGet	An active template (UPPER/LOWER/TARGET)
TLDISP*	○	:TRACe:TEMPlate:DISPlay	
TLEXTRA*	▲	:TRACe:TEMPlate:EDIT:ETYPe	An active template (UPPER/LOWER/TARGET)
TLGONO*	○	:TRACe:TEMPlate:GONOgo	
TLSADR**	○	-	
TLSSYNC*	○	-	
TLLVSFT***.**	○	:TRACe:TEMPlate:WAVelength:SHIFt	
TLRESLT?	○	:TRACe:TEMPlate:RESult?	
TLTYPE*	○	:TRACe:TEMPlate:TTYPe	
TLWLSFT****.**	○	:TRACe:TEMPlate:WAVelength:SHIFt	
TRA?	▲	:TRACe:ATTRibute:TRA?	Diff.talker format 2: MAX HOLD / MIN HOLD

List of the AQ6317-Compatible Commands

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
TRB?	▲	:TRACe:ATTRibute:TRB?	Diff. talker format 2: MAX HOLD / MIN HOLD
TRC?	×	:TRACe:ATTRibute:TRC?	
TRFMT*	○	-	
UCWRN*	○	:SYSTem:DISPlay:UNCal<wsp> OFF|ON|0|1	
UHLD*	○	:DISPlay[:WINDow]:SPLit<wsp>ON|1; :DISPlay[:WINDow]:SPLit:HOLD: UPPer<wsp>OFF|ON|0|1	
ULTRA*	○	:DISPlay[:WINDow]:SPLit<wsp>ON|1; :DISPlay[:WINDow]:SPLit:POSition<wsp> TRA,UP|LOW|0|1	
ULTRB*	○	:DISPlay[:WINDow]:SPLit<wsp>ON|1; :DISPlay[:WINDow]:SPLit:POSition <wsp>TRB,UP|LOW|0|1	
ULTRC*	○	:DISPlay[:WINDow]:SPLit<wsp>ON|1; :DISPlay[:WINDow]:SPLit:POSition <wsp>TRC,UP|LOW|0|1	
WARN?	▲	:SYSTem:ERRor[:NEXT]?	
WCAL****.***	▲	:CALibration:WAVelength:EXTernal: SOURce<wsp>LASer|0; CALibration:WAVelength:EXTernal: WAVelength<wsp><NRf>[M]	Diff. parameter range
WCALG****.***	▲	:CALibration:WAVelength:EXTernal: SOURce<wsp>GASCell|1; CALibration:WAVelength:EXTernal: WAVelength<wsp><NRf>[M]	Diff. parameter range
WCALS	○	:CALibration:WAVelength:INTernal [:IMMediate]	
WCALT****;#.###	▲	:CALibration:WAVelength:OFFSet: TABLe<wsp><integer>,<NRf>[DB]	Diff. parameter range
WDMAN	○	:CALCulate:CATegory<wsp>WDM|10	
WDMCHAUT*	×	-	No parameter
WDMCHSW***;#	×	-	
WDMDIF**.**	▲	:CALCulate:PARameter[:CATegory] :WDM:MDIFf<wsp><NRf>[DB]	Set only in WDM Analysis, not in NF Analysis
WDMDISP*	▲	:CALCulate:PARameter[:CATegory] :WDM:DTYPe<wsp><display type> <display type>=ABSolute|0, RELatibe|1,MDRift|2,GDRift|3	Diff. parameter 0: ABSOLUTE 1: RELATIVE 3: DRIFT(MEAS) 4: DRIFT(GRID)
WDMDSPMSK***	▲	:CALCulate:PARameter[:CATegory] :WDM:DMASk<wsp><NRf>[DB]	Diff. parameter range
WDMDUAL*	○	:CALCulate:PARameter[:CATegory] :WDM:DUAL<wsp>OFF|ON|0|1	
WDMMAX***	×	-	No parameter
WDMMR	○	:CALCulate:PARameter[:CATegory] :WDM:MMReset	

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
WDMNOI*	▲	[NOISE_ALGO is Auto Center] (NOISE POI=CTR) :CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>ACENter|2 [NOISE_ALGO is MANUAL Fix] (NOISE POI=CTR) :CALCulate:PARameter[:CATegory]:WDM:NALGo<wsp>MFIX|1; :CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>LINear|0;	Diff. set value 0: AUTO-FIX 1: AUTO-CTR Set only in WDM Analysis, not in NF Analysis
WDMNOIBW****	○	:CALCulate:PARameter[:CATegory]:WDM:NBW<wsp><NRf>[M|HZ]	
WDMNOIP**.**	▲	:CALCulate:PARameter[:CATegory]:WDM:FALGo<wsp>LINear|0; :CALCulate:PARameter[:CATegory]:WDM:NBW<wsp><NRf>[M]	Valid only when NOISE ALGO is set to MANUAL FIX
WDMOS*	○	:CALCulate:PARameter[:CATegory]:WDM:RELation<wsp>OFFSet|SPACing|0|1	
WDMREF*	×	-	
WDMREFDAT*	×	-	
WDMRH	○	:CALCulate:PARameter[:CATegory]:WDM:RCH<wsp>0	
WDMRN***	○	:CALCulate:PARameter[:CATegory]:WDM:RCH<wsp><integer>	
WDMSLOPE*	○	:CALCulate:PARameter[:CATegory]:WDM:OSLope<wsp>OFF|ON|0|1	
WDMTCOPY	○	:HCOPY[:IMMediate]:FUNCtion :CALCulate:LIST	
WDMTH**.*	▲	:CALCulate:PARameter[:CATegory]:WDM:TH<wsp><NRf>[DB]	Set only in WDM Analysis, not in NF Analysis
WDMUNT*	×	:CALCulate:MARKer:UNIT<wsp>WAVelength|FREQuency|0|1	
WLSFT**.***	○	:SENSe:CORRection:WAVelength:SHIFt<wsp><NRf>[M]	
WMKR****.***	▲	:CALCulate:MARKer:X<wsp>0,<NRf>[M]	Diff. parameter range
WNFAN	○	:CALCulate:CATegory<wsp>NF|11	
WNFCVF*	▲	:CALCulate:PARameter[:CATegory]:NF:FALGo<wsp><algorhythm> <algorhythm>=AFIX|0,MFIX|1,ACENter|2,MCENter|3	Valid only when ASE ALGO is set to MANUAL FIX or MANUAL CTR
WNFFA**.**	▲	:CALCulate:PARameter[:CATegory]:NF:FARea<wsp><NRf>[M|HZ]	Valid only when ASE ALGO is set to MANUAL FIX
WNFNP**.**	▲	:CALCulate:PARameter[:CATegory]:NF:MARea<wsp><NRf>[M|HZ]	Valid only when all the following conditions are satisfied 1. ASE ALGO is set to MANUAL FIX or MANUAL CTR 2. FITTING ALGO is set besides LINEAR
WNFOFI***.**	○	:CALCulate:PARameter[:CATegory]:NF:IOFFset<wsp><NRf>[DB]	
WNFOFO***.**	○	:CALCulate:PARameter[:CATegory]:NF:OOFFset<wsp><NRf>[DB]	
WNFSSE*	×	-	No parameter

List of the AQ6317-Compatible Commands

AQ6317 Series Control Command	Operates in AQ6317-Comp Mode	AQ6374 Control Command Corresponding to AQ6317 Command	Remarks
WR*'@@@@'	○	:MMEMory:STORe:TRACe<wsp> <trace name>,BIN|CSV, <"file name">,EXTernal <trace name>=TRA|TRB|TRC	Saving to the external memory
WR3D*'@@@@'	×	-	
WRDT '@@@@'	○	:MMEMory:STORe:DATA<wsp> <"file name">,EXTernal	Saving to the external memory
WRGR'@@@@'	○	:MMEMory:STORe:GRAPhics<wsp> B&W|COLor,BMP|TIFF,<"file name">, EXTernal	Saving to the external memory
WRMEM** '@@@@'	○	:MMEMory:STORe:MEMory<wsp> <integer>,BIN|CSV,<"file name">, EXTernal	Saving to the external memory
WRPRG** '@@@@'	○	:MMEMory:STORe:PROGram<wsp> <integer>,<"file name">,EXTernal	Saving to the external memory
WRSET'@@@@'	○	:MMEMory:STORe:SETTing<wsp> <"file name">,EXTernal	Saving to the external memory
WRTA	○	:TRACe:ATTRibute:TRA<wsp>WRITe|0	
WRTB	○	:TRACe:ATTRibute:TRB<wsp>WRITe|0	
WRTC	○	:TRACe:ATTRibute:TRC<wsp>WRITe|0	
WRTLT'@@@@'	×	-	
XUNT*	○	:UNIT:X<wsp>WAVelength|FREQuency|	
		|0|1	
ZSCL**	×	-	
ZSWPT**	○	:SENSe:SWEep:TIME:0NM<wsp> <integer>[SEC]	

HIGH1, HIGH2, HIGH3 of Measurement Sensitivity

Even when the measurement sensitivity of the instrument is set to HIGH1 or HIGH2 or HIGH3, the chopper cannot operate unless the Chop Mode setting of the Sensitivity/ Chop Mode soft key is set to SWITCH.
However, with AQ6317 series instruments, if the measurement sensitivity is set to HIGH1, HIGH2, or HIGH3, a chopper that removes monochrometer stray light is activated.
The instrument includes the following AQ6317-compatible mode commands that allow you to edit the settings of the chopper operation.

AQ6317 command to use to set the chopper operation
Control command
    CHOP*
        *: 0 = Chopper OFF, 2 = SWITCH mode
Query command
    CHOP?
        A return value: Same as the above

Index