

---

**User's  
Manual**

**AQ6150B, AQ6151B  
Optical Wavelength Meter  
Remote Control**

---

---

Thank you for purchasing the AQ6150 or AQ6151 Optical Wavelength Meter.

This remote control user's manual is for the instrument.

It explains the following items.

- GP-IB interface
- Ethernet interface
- Remote Commands

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.

## List of Manuals

The following manuals, including this one, are provided as manuals for the instrument. Please read all manuals.

Manual Title	Manual No.	Description
AQ6150B, AQ6151B Optical Wavelength Meter User's Manual	IM AQ6150-01EN	The manual explains all the features of the instrument other than the remote control features. The supplied CD contains the PDF file of this manual.
AQ6150B, AQ6151B Optical Wavelength Meter Getting Started Guide	IM AQ6150-02EN	Provided as a printed manual. The guide explains the handling precautions, basic operations, and specifications of the instrument. The supplied CD contains the PDF file of this manual.
AQ6150B, AQ6151B Optical Wavelength Meter Remote Control User's Manual	IM AQ6150-17EN	This manual. The manual explains the communication interface features of the instrument and how to use them. The supplied CD contains the PDF file of this manual.
AQ6150B, AQ6151B Optical Wavelength Meter	IM AQ6150B-92Z1	Document for China

The "EN" and "Z1" in the manual number are the language code.

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

Document No.	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 continuing improvements to the instrument's performance and functionality. The figures given in this manual may differ from those that actually appear 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 is strictly prohibited.

## Trademarks

- Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are either registered trademarks or trademarks of Adobe Systems Incorporated.
- In this manual, the ® and TM symbols do not accompany their respective registered trademark or trademark names.
- Other company and product names are registered trademarks or trademarks of their respective holders.

---

## Revisions

- 1st Edition: November 2018

---

# Conventions Used in This Manual

## Safety Markings

The following safety markings are used in this manual.

### **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 attention 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 information that is important for proper operation of the instrument.

## 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 the Manual

This manual contains five chapters and an appendix.

### **Chapter 1 Remote Control Feature**

Gives an overview of the various available communication interfaces.

### **Chapter 2 GP-IB Interface**

Describes the features and specifications of the GP-IB interface for controlling the instrument from a PC.

### **Chapter 3 Ethernet Interface**

Describes the features and specifications of the Ethernet interface.

### **Chapter 4 Status Registers**

Describes the status byte, various registers, and queues.

### **Chapter 5 Remote Commands**

Describes every command individually.

# Contents

List of Manuals .....	i
Conventions Used in This Manual .....	iii
How To Use This Manual .....	iv
<b>Chapter 1 Remote Control Feature</b>	
1.1 Remote Interface .....	1-1
1.2 Switching between Remote and Local Modes .....	1-2
1.3 Exchanging Remote Commands .....	1-3
1.4 Response When a Signal Is Not Detected .....	1-4
<b>Chapter 2 GP-IB Interface</b>	
2.1 Using the GP-IB Interface to Connect to a Controller .....	2-1
2.2 GP-IB Interface Features .....	2-3
2.3 GP-IB Interface Specifications .....	2-5
2.4 Setting the GP-IB Address .....	2-6
2.5 Responses to Interface Messages .....	2-8
2.6 Sample Program .....	2-10
<b>Chapter 3 Ethernet Interface</b>	
3.1 Using the Ethernet Interface to Connect the Devices .....	3-1
3.2 Ethernet Port Features .....	3-2
3.3 Ethernet Configuration .....	3-3
3.4 Sample Program (for SOCKET) .....	3-10
<b>Chapter 4 Status Registers</b>	
4.1 Status Registers .....	4-1
4.2 Status Byte Register .....	4-3
4.3 Standard Event Status Register .....	4-5
4.4 Operation Status Register .....	4-7
4.5 Questionable Status Register .....	4-12
<b>Chapter 5 Remote Commands</b>	
5.1 Command Syntax and Types .....	5-1
5.2 Table of Soft Key to Remote Command Assignments .....	5-3
5.3 Remote Command Tree .....	5-6
5.4 Common Commands .....	5-13
5.5 Device-specific Commands .....	5-15
CALCulate2 Sub System Commands .....	5-15
CALCulate3 Sub System Commands .....	5-17
CONFigure Sub System Commands .....	5-26
DISPlay Sub System Commands .....	5-28
FETCh Sub System Commands .....	5-33
FORMat Sub System Commands .....	5-35
MEASure Sub System Commands .....	5-36
MMEMory Sub System Commands .....	5-38
READ Sub System Commands .....	5-40
SENSe Sub System Commands .....	5-43
STATus Sub System Commands .....	5-43
SYSTem Sub System Commands .....	5-44

**Contents**

---

TRIGger Sub System Commands ..... 5-46

UNIT Sub System Commands ..... 5-47

**Appendix**

Appendix 1 About the IEEE 488.2-1992 Standard ..... App-1

## 1.1 Remote Interface

You can use remote commands to control the instrument.  
The remote commands conform to SCPI (Standard Commands for Programmable Instruments).  
The instrument is equipped with the following remote interfaces.

### **GP-IB (IEEE488.2; see chapter 2)**

Used to control the instrument remotely from a controller, such as a PC.  
GP-IB is used to connect a controller to the devices that it controls.

### **Ethernet (See chapter 3)**

Used to control the instrument remotely over a network from a controller, such as a PC.



---

## 1.2 Switching between Remote and Local Modes

### When Switching from Local to Remote Mode

- In the GP-IB interface, the instrument switches to remote mode when it is in local mode and receives a REN (Remote Enable) or a listen address with ATN set to "True" from the controller.
- In remote mode, the REMOTE LED illuminates.
- All panel controls except the LOCAL key and the POWER button are disabled.
- Settings entered in local mode are retained even when the instrument switches to remote mode.
- The instrument switches to local lockout mode when it receives an LLO (Local Lock Out) message from the controller.  
In local lockout mode, the instrument will not return to local mode even when you press the LOCAL key.  
To return to local mode, clear the local lockout mode first, and then press the LOCAL key. To clear the local lockout mode, set REN to false.
- If you are using the Ethernet interface, the instrument switches to remote mode when authentication is complete and you are logged in.

### When Switching from Remote to Local Mode

When the instrument is in remote mode and you press LOCAL, the instrument switches to local mode.

This will not work when the instrument is in local lockout mode.

- The REMOTE LED turns off.
- Key operations are enabled.
- Settings entered in remote mode are retained even when the instrument switches to local mode.
- In the GP-IB interface, the instrument switches to local lockout mode when it receives a GTL (Go To Local) message from the controller or when REN is set to false.

## 1.3 Exchanging Remote Commands

### Buffer

#### Input Buffer

The instrument has a one-stage input buffer. The buffer size is 4 MB.

If the instrument receives data that exceeds the buffer size, it discards the excessive portion of the data.

In such cases, the instrument also discards the command after the last command separator within the 2 MB of data it receives.

#### Output Buffer

The instrument has a one-stage output buffer. The buffer size is 4 MB.

It holds only the most recent data.

(If the instrument is holding data in the buffer and receives a command, it clears the data in the buffer and accepts the new data.)

If multiple talker commands are executed together and the buffer size is exceeded, the instrument will:

- Set the query error bit (QYE) of the standard event status register to 1.
- Clear the output buffer.
- Continue processing the commands that have already been received even after the output buffer overflows.

The instrument will not store talker data sent through new talker commands in the output buffer.

#### Error Buffer

The instrument has a 10-stage error buffer.

## 1.4 Response When a Signal Is Not Detected

The output value for when a wavelength output request is received from a PC when a signal is not detected can be set to any value between 0 nm and 300 nm. The default value is 0 nm.

**1. Press **SYSTEM**.**

A system setup menu appears.

**2. Press the **REMOTE SETTING** soft key.**

A remote setup screen appears.

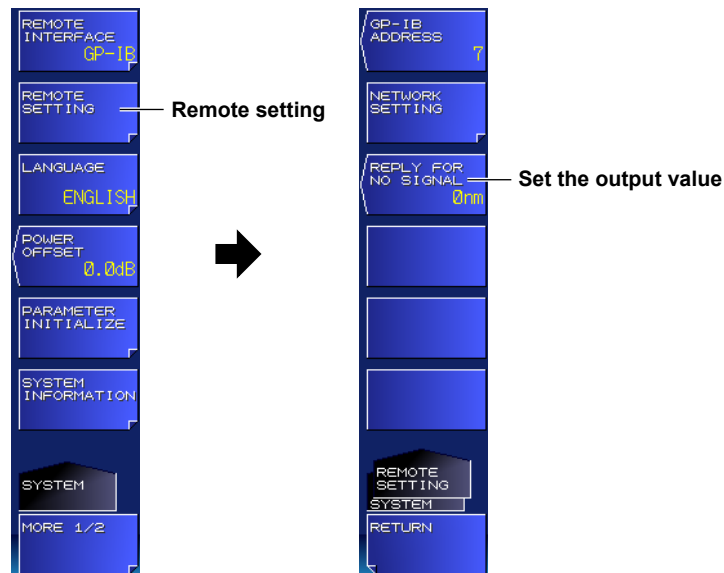
**3. Press the **REPLY FOR NO SIGNAL** soft key.**

A screen for setting the output value appears.

**4. Enter the output value using the arrow keys or numeric keypad.**

**5. Press **ENTER**.**

The output value that you enter appears on the soft key.



### Note

This function is valid for the following functions.

```
:FETCh[:SCALar]:POWer:{FREQuency|WAVelength|WNUMber}?  
:MEASure[:SCALar]:POWer:{FREQuency|WAVelength|WNUMber}?  
:READ[:SCALar]:POWer:{FREQuency|WAVelength|WNUMber}?
```

## 2.1 Using the GP-IB Interface to Connect to a Controller

### GP-IB Cable

The instrument is equipped with an IEEE St'd 488-1978 24-pin GP-IB connector. Use a GP-IB cable that conforms to this standard.

### Connection Procedure

GP-IB port: Connect this port to a PC to control the instrument from the PC.

Turn off the instrument and the device that you will connect to the instrument. Connect a GP-IB cable to the GP-IB port on the rear panel of the instrument.

#### CAUTION

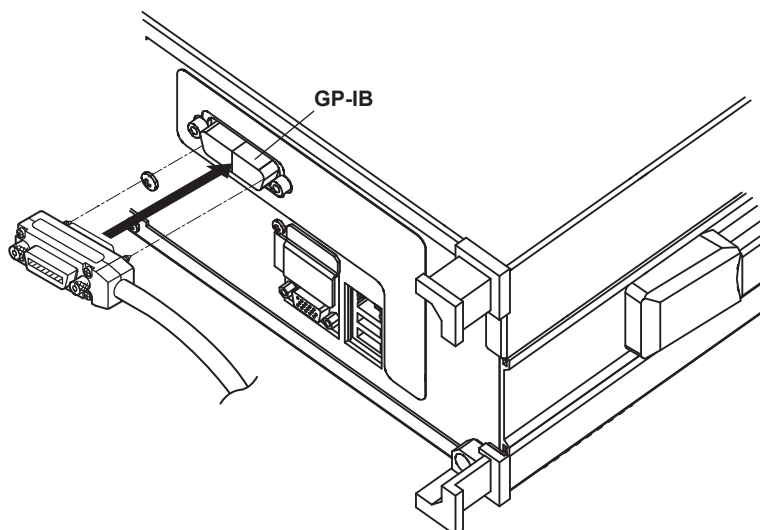
Be sure to turn off the PC and the instrument before you connect or remove communication cables.

Otherwise, erroneous operation may result, or the internal circuitry may break.

### French

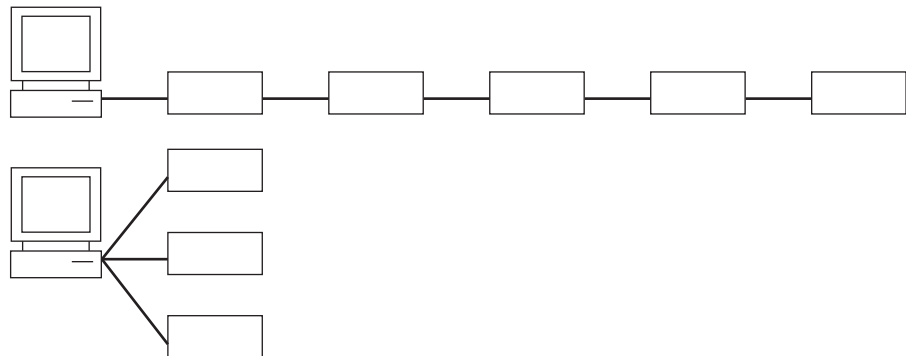
#### ATTENTION

Veiller à mettre le PC et l'instrument hors tension avant de brancher ou de débrancher les câbles de communication, pour éviter de provoquer des dysfonctionnements ou des courts-circuits internes.



### Precautions to Be Taken When Connecting the Cable

- Firmly tighten the screws on the GP-IB cable connector.
- Multiple cables can be used to connect multiple devices.  
However, no more than 15 devices, including the controller, can be connected on a single bus.
- When connecting multiple devices, you must assign a unique address to each device.
- Use cables that are no longer than 2 m in length to connect devices.
- Make sure the total length of all cables does not exceed 20 m.
- When devices are communicating, have at least two-thirds of the devices on the bus 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 Features

### GP-IB Interface Features

#### Listener Capability

- Through the GP-IB interface, you can specify the same settings that you can using the front panel keys. You cannot turn the instrument on and off or change communication settings or some other settings.
- The instrument can receive setup data, measurement data, etc., according to the output commands from the controller.
- The instrument can receive commands, such as status report commands.

#### Talker Capability

- The instrument can output setup data, measured data, etc.

#### **Note**

- Talk-only, listen-only, and controller capabilities are not available.
- You cannot use the GP-IB interface simultaneously with the Ethernet interface.

### Message Terminator

**The following message terminators can be used on the instrument.**

Program Message Terminator

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

The instrument recognizes LF as an ASCII line feed (0Ah) and CR (0Dh) in CR+LF as a white space character, so you can also use CR+LF as a message terminator.

#### Response Message Terminator

The response message terminator is LF+EOI.

#### Receiving Remote Commands

- When the instrument finishes receiving a message, it releases the GP-IB bus.
- If the instrument receives the next command while it is executing the previous command, it stores the next command in the receive buffer and then releases the GP-IB bus.
- If a command is already in the receive buffer, the instrument does not retrieve additional commands on the GP-IB bus that may be available.
- If the instrument completes the execution of the previous command, it executes the command in the buffer and clears the buffer.  
If the next command is available on the bus, the instrument stores it in the receive buffer.
- If multiple commands are included in a single output statement, the instrument retrieves all commands and executes them in the order they are written.  
In this case, the instrument cannot retrieve the next command on the bus until it starts executing the last command in the statement.

### Data Queries

- Query commands and data output requests are used to perform queries from an external controller.
- A query command has a question mark appended to the end of the command.
- For a query that has a parameter, <wsp>+<parameter> is appended to the question mark.
- When the instrument receives a query command, it places the response to the command in the output buffer.
- The data is held in the output buffer until an input statement is received from the controller or a new query command is received.
- If multiple query commands are concatenated with semicolons, the instrument places the responses to all the query commands in the output buffer.  
In this case, the instrument transmits all the data in the buffer when it receives the next data output request.

### Device Trigger Capability

When the instrument receives a GET (Group Execute Trigger) command, it performs a single measurement.

## 2.3 GP-IB Interface Specifications

### GP-IB Interface Specifications

Electrical and mechanical specifications	Conforms to IEEE St'd 488-1978
Functional specifications	See the table below.
Protocol	Conforms to IEEE St'd 488.2-1992
Code	ISO (ASCII) codes
Mode	Addressable mode
Address setting	Set the address in the range of 0 to 30 on the GP-IB setup screen of the SYSTEM menu.
Clearing remote mode	Press LOCAL to clear remote mode. This is not possible when local lockout has been activated by the controller.

### Functional Specifications

Function	Subset Name	Description
Source handshaking	SH1	Full source handshaking capability
Acceptor handshaking	AH1	Full acceptor handshaking capability
Talker	T6	Basic talker capability, serial polling, and untalk on MLA (My Listen Address). No talk-only capability
Listener	L4	Basic listener capability and unlisten on MTA (My Talk Address). No listen-only capability
Service request	SR1	Full service request capability
Remote local	RL1	Full remote/local capability
Parallel polling	PP0	No parallel polling capability
Device clear	DC1	Full device clear capability Output buffer clear capability Input buffer clear (unprocessed command clear) capability Error buffer clear capability STB and ESR clear capability
Device trigger	DT0	Device trigger capability
Controller	C0	No controller capability
Electric characteristics	E1	Open collector



## 2.4 Setting the GP-IB Address

### Procedure

#### Selecting the Communication Interface

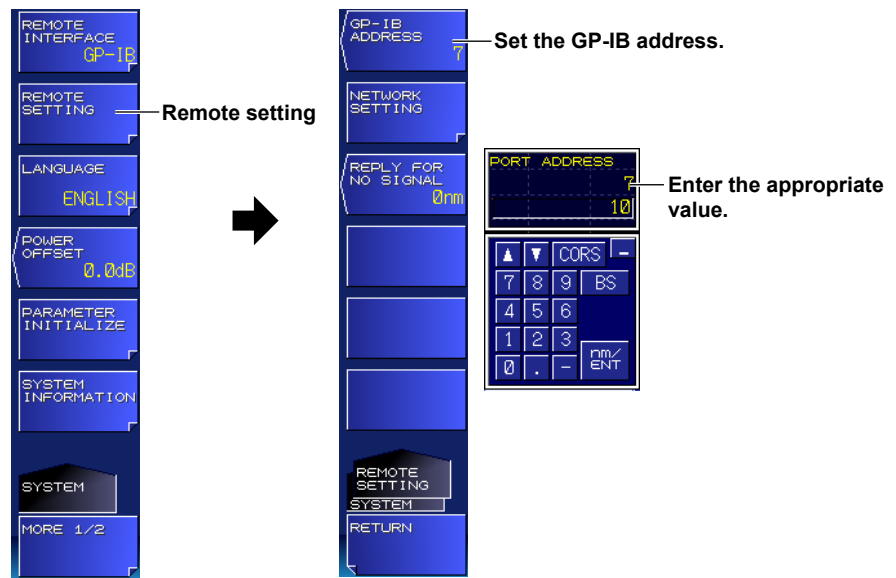
1. Press **SYSTEM**.  
A system setup menu appears.
2. Press the **REMOTE INTERFACE** soft key.  
A remote interface setup menu appears.
3. Press the **GP-IB** soft key.  
The setup menu returns to the previous display, and "GP-IB" appears on the soft key.



#### Setting the Address

4. Press the **REMOTE SETTING** soft key.  
A remote setup screen appears..
5. Press the **GP-IB ADDRESS** soft key.  
A screen for setting the GP-IB address appears.

6. Enter the address using the arrow keys or numeric keypad.



7. Press **ENTER**.  
The address that you enters appears on the soft key.

Explanation

Set the GP-IB address if you want to use the controller to configure the same settings that you can from the panel keys or have the instrument output setup data, measured data, etc., to the controller.

Setting the GP-IB Address

Set the instrument address for the addressable mode in the following range:  
0 to 30  
Each device that is connected in a GP-IB system has its own unique address. This address is used to distinguish between different devices. Therefore, you must assign a unique address to the instrument when you connect it to a PC or other device.

**Note** \_\_\_\_\_  
When the controller is using the GP-IB bus, do not change the address of any connected devices.

---

## 2.5 Responses to Interface Messages

### Responses to Interface Messages

#### Responses to Uni-Line Messages

##### IFC (Interface Clear)

- Clears the talker and listener functions.
- Stops data transmission if it is in progress.

##### REN (Remote Enable)

- Switches between the remote and local modes.

IDY (Identify) is not supported.

#### Responses to Multi-Line Messages (Address commands)

##### GTL (Go To Local)

- Switches the instrument to local mode.

##### SDC (Selected Device Clear)

- Clears the program message (command) being received and the output queue.
- Discards \*OPC and \*OPC? commands that are being executed.
- The WAI command is immediately terminated.

PPC (Parallel Poll Configure) and TCT (Take Control) are not supported.

#### Responses to Multi-Line Messages (Universal commands)

##### LLO (Local Lockout)

- Prohibits switching to local mode by disabling the LOCAL key on the front panel.

##### DCL (Device Clear)

- The same operation as the SDC message.

##### SPE (Serial Poll Enable)

- Sets the talker function on all devices on the bus to serial polling mode.
- The controller will poll each device one by one.

##### SPD (Serial Poll Disable)

- Clears the serial polling mode of the talker function on all devices on the bus.

PPU (Parallel Poll Unconfigure) is not supported.

### What Are Interface Messages?

Interface messages are also referred to as interface commands or bus commands. They are commands that are issued by the controller. They are classified as follows:

#### Uni-line Messages

A single control line is used to transmit uni-line messages. The following three types are available.

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

Multi-line Messages

Eight data lines are used to transmit multi-line messages.  
The messages are classified as follows:

Address Commands

Some address commands are valid when a device is designated as a listener, and some are valid when it is designated as a talker.  
The following five commands are available.

Commands available to a device designated as a listener

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

Commands available to a device designated as a talker

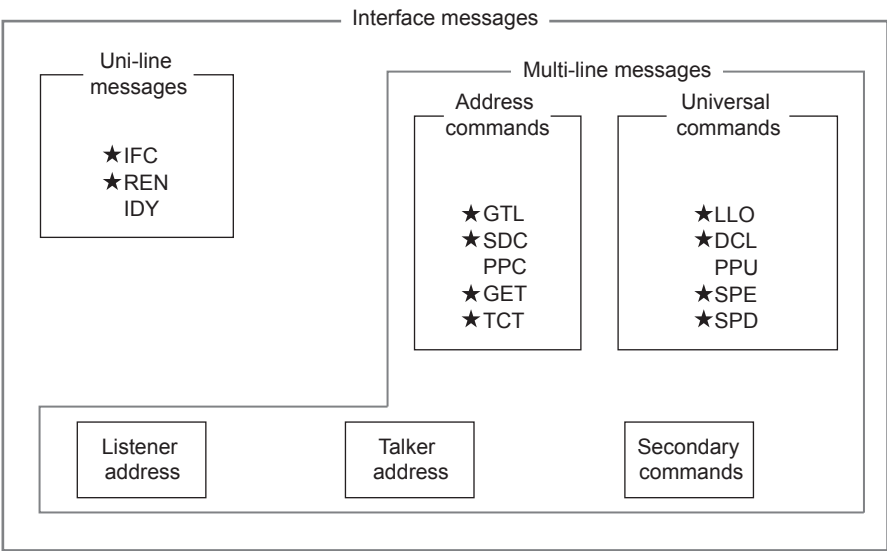
- TCT (Take Control)

Universal commands

Universal commands are available to all devices regardless of their listener or talker designation.  
The following five commands are available.

- LLO (Local Lockout)
- DCL (Device Clear)
- PPU (Parallel Poll Unconfigure)
- SPE (Serial Poll Enable)
- SPD (Serial Poll Disable)

There are other interface messages: listener-address, talk-address, and secondary commands.



The instrument supports interface messages marked with a ★.

Note

Difference between SDC and DCL

In multi-line messages, SDC messages are those that require talker or listener designation and DCL messages are those that do not require a designation. Therefore, SDC messages are directed at a particular instrument while DCL messages are directed at all instruments on the bus.

## 2.6 Sample Program

The following shows an example of controlling the instrument remotely using the GP-IB port.

Model:	PC-AT Computer
Language:	Visual Basic 2008
GP-IB board:	GP-IB board from National Instruments
Component:	NationalInstruments.Common NationalInstruments.NI4882 .NET Framework 3.5

### Sample Program 1

Sample program for making one Single measurement and displaying the results (wavelength information, power information, and FP-LD analysis results) on the screen via GP-IB. Example of the -MW (multi-wavelength) suffix code model.

#### Source Code

```
Imports System
Imports System.IO
Imports NationalInstruments.NI4882

Module GpibSingleMeasure
'
' Sample Program for Making One Single Measurement and Displaying the
' Results (Wavelength Information, Power Information, and FP-LD Analysis ' Results)
' on the Screen via GP-IB
'
Sub Main()
Try
Dim GpibDevice As Device
Dim wlmAddr As Integer
Dim replyString As String
Dim wavArray As Double()
Dim powArray As Double()
Dim fwhm, ctrWl, totalPwr, sigma As Double
Dim maxPeakPower, maxPeakWl As Double

'=====
'Wavelength meter information
'=====
wlmAddr = 7
GpibDevice = New Device(0, wlmAddr)

'=====
'Set the wavelength meter's measurement conditions
'=====
Call GpibWrite("*RST", GpibDevice)
Call GpibWrite(":CALC2:PTHR:MODE REL", GpibDevice)
Call GpibWrite(":CALC2:PTHR 15", GpibDevice)
Call GpibWrite(":UNIT:WL NM", GpibDevice)
Call GpibWrite(":UNIT:POW DBM", GpibDevice)
Call GpibWrite(":DISP:WIND2:STAT ON", GpibDevice)

'=====
'Execute measurement and retrieve data
'=====
'Use the READ command to execute measurement and retrieve data.
'Use the FETC command to retrieve measured power information data.
Call GpibWrite(":READ:ARR:POW:WAV?", GpibDevice)

'Wavelength meter GP-IB address
'Open GP-IB
'Reset the instrument
'Set the peak detection threshold
'setting to relative mode
'Set the peak detection threshold
'to 15 db
'Set the wavelength unit to nm
'Set the power unit to dBm
'Enable the graph display
'Execute a Single measurement
'and retrieve wavelength data
```

```

replyString = GpibRead(GpibDevice)
Call SplitArrayData(replyString, wavArray)           'Store wavelength information in an
                                                    'array

Call GpibWrite(":FETC:ARR:POW?", GpibDevice)         'Retrieve measured power information
replyString = GpibRead(GpibDevice)
Call SplitArrayData(replyString, powArray)           'Store power information in an
                                                    'array

'=====
'Display the results (wavelength and power information)
'=====
Console.WriteLine("No. |Wavelength(m) |Power (dBm)")
For idx As Integer = 1 To wavArray.Length
    Console.WriteLine((idx).ToString + "|" + wavArray(idx - 1).ToString() + "|" + _
        powArray(idx - 1).ToString())
Next

'=====
'Retrieve maximum power peak information
'=====
Call GpibWrite(":FETC:POW? MAX", GpibDevice)         'Retrieve power information by
                                                    'specifying the peak with the
                                                    'maximum power

replyString = GpibRead(GpibDevice)
maxPeakPower = Convert.ToDouble(replyString)
Call GpibWrite(":FETC:POW:WAV?", GpibDevice)         'Retrieve the wavelength of the
                                                    'peak specified by :FETC:POW? MAX

replyString = GpibRead(GpibDevice)
maxPeakWl = Convert.ToDouble(replyString)
Console.WriteLine("Highest Peak Power      : " + maxPeakPower.ToString + " dBm")
Console.WriteLine("Highest Peak Wavelength: " + maxPeakWl.ToString + " nm")

'=====
'Retrieve the FP-LD analysis results
'=====
Call GpibWrite(":CALC3:FPER ON", GpibDevice)         'Enable FP-LD analysis
Call GpibWrite(":CALC3:FPER:FWM?", GpibDevice)       'Retrieve FWHM
replyString = GpibRead(GpibDevice)
fwhm = Convert.ToDouble(replyString)
Call GpibWrite(":CALC3:FPER:MEAN?", GpibDevice)       'Retrieve Center WL
replyString = GpibRead(GpibDevice)
ctrWl = Convert.ToDouble(replyString)
Call GpibWrite(":CALC3:FPER:POW?", GpibDevice)       'Retrieve Total Power
replyString = GpibRead(GpibDevice)
totalPwr = Convert.ToDouble(replyString)
Call GpibWrite(":CALC3:FPER:SIGM?", GpibDevice)      'Retrieve  $\sigma$ 
replyString = GpibRead(GpibDevice)
sigma = Convert.ToDouble(replyString)
Console.WriteLine("====FP-LD Analysis====")
Console.WriteLine("FWHM                : " + (fwhm * 1000000000).ToString + "nm")
Console.WriteLine("Sigma                : " + (sigma * 1000000000).ToString + "nm")
Console.WriteLine("CTR WL              : " + (ctrWl * 1000000000).ToString + "nm")
Console.WriteLine("TOTAL PWR          : " + totalPwr.ToString + "dBm")

'=====
'Save data to the internal memory
'=====
'Save the screen capture and result data to the internal memory
Call GpibWrite(":MMEM:STOR SIM2,\"\"\\WLM_IMAGE\"",INT", GpibDevice)
Call GpibWrite(":MMEM:STOR TABL,\"\"\\WLM_TABLE\"",INT", GpibDevice)

'=====
'Transfer the data saved in the internal memory to the PC
'=====
Call GpibWrite(":MMEM:DATA? \"\"\\WLM_IMAGE.BMP\"",INT", GpibDevice)
GpibReadBlockData2File(GpibDevice, "WLM_IMAGE.BMP")
Call GpibWrite(":MMEM:DATA? \"\"\\WLM_TABLE.CSV\"",INT", GpibDevice)
GpibReadBlockData2File(GpibDevice, "WLM_TABLE.CSV")

```

## 2.6 Sample Program

```
        Console.ReadLine()

Catch ex As Exception
    Console.WriteLine(ex.Message)

    Console.ReadLine()

End Try
End Sub

'=====
'Function for sending character strings to GP-IB
'=====
Sub GpibWrite(ByVal commandStr As String, ByRef gpib As Device)
    gpib.Write(commandStr)
End Sub

'=====
'Function for reading a line of data from GP-IB
'=====
Function GpibRead(ByRef gpib As Device) As String
    GpibRead = gpib.ReadString()
    Exit Function
End Function

'=====
'Function for reading block data and saving it to a file
'=====
Function GpibReadBlockData2File(ByRef gpib As Device, ByVal filename As String) As Integer
    Dim headerLen As Integer
    Dim dataLen As Integer
    Dim dataByte As Byte()
    Dim file As New FileStream(filename, FileMode.Create, FileAccess.Write)

    If String.Compare(gpib.ReadString(1), "#") <> 0 Then
        GpibReadBlockData2File = -1
        Exit Function
    End If
    headerLen = Integer.Parse(gpib.ReadString(1))
    dataLen = Integer.Parse(gpib.ReadString(headerLen))

    While dataLen > 1024
        dataByte = gpib.ReadByteArray(1024)
        file.Write(dataByte, 0, dataByte.Length)

        dataLen = dataLen - dataByte.Length
    End While

    dataByte = gpib.ReadByteArray(dataLen)
    file.Write(dataByte, 0, dataByte.Length)

    file.Close()
    GpibReadBlockData2File = 0
End Function

'=====
'Split the READ/FETC/MEAS results into an array
'=====
Sub SplitArrayData(ByVal dataString As String, ByRef dataArray As Double())
    Dim peakNum As Integer
    Dim arrayDataStr As String() = dataString.Split(",")

    peakNum = Integer.Parse(arrayDataStr(0))
End Sub
```

```
'Wait for the user to press the
'Enter key
'Error handling
'Display the error message that
'occurred
'Wait for the user to the Enter
'key

'Send data

'Receive data

'Retrieve the first character
'Error if the first character is
'not a "#"

'Size of the area containing the
'data length information
'Retrieve the data length
'information

'Read data 1024 bytes at a time
'Write the retrieved data to the
'file

'Retrieve the last piece of data
'Write the retrieved data to the
'file

'Split the data by using a comma
'as the delimiter
'Retrieve the number of data
'values
```

```

dataArray = New Double(peakNum - 1) {}
For idx As Integer = 1 To arrayDataStr.Length - 1
    dataArray(idx - 1) = Convert.ToDouble(arrayDataStr(idx)) 'Read all the data values
Next
End Sub
End Module

```

## Execution Example

```

No.|Wavelength(m) |Power(dBm)
1|1.30678822E-06|-14.3279541
2|1.30756963E-06|-9.42082105
3|1.30835228E-06|-2.23592107
4|1.30913555E-06|-3.93065804
5|1.30991986E-06|-13.5578301
Highest Peak Power :-2.23592107 dBm
Highest Peak Wavelength:1.30835228E-06 nm
====FP-LD Analysis====
FWHM           : 1.47415158nm
Sigma           : 0.625966702nm
CTR WL          : 1308.55169nm
TOTAL PWR       : 0.782282871dBm

```



### Sample Program 2

Sample program for making one Single measurement and displaying the results (wavelength information and power information) on the screen via GP-IB. Example of the -SW (single-wavelength) suffix code model.

#### Source Code

```
Imports System
Imports System.IO
Imports NationalInstruments.NI4882

Module GpibSingleMeasure
'
' Sample Program for Making One Single Measurement and Displaying the
' Results (Wavelength Information and Power Information)
' on the Screen via GP-IB
'
Sub Main()
    Try
        Dim GpibDevice As Device
        Dim wlmAddr As Integer
        Dim replyString As String
        Dim waveLength As Double()
        Dim power As Double()

        '=====
        'Wavelength meter information
        '=====
        wlmAddr = 7                                'Wavelength meter GP-IB address
        GpibDevice = New Device(0, wlmAddr)         'Open GP-IB

        '=====
        'Set the wavelength meter's measurement conditions
        '=====
        Call GpibWrite("*RST", GpibDevice)          'Reset the instrument
        Call GpibWrite(":UNIT:WL NM", GpibDevice)    'Set the wavelength unit to nm
        Call GpibWrite(":UNIT:POW DBM", GpibDevice)   'Set the power unit to dBm
        '=====
        'Execute measurement and retrieve data
        '=====
        'Use the READ command to execute measurement and retrieve data.
        'Use the FETC command to retrieve measured power information data.
        Call GpibWrite(":READ:POW:WAV?", GpibDevice) 'Execute a Single measurement
                                                    'and retrieve wavelength data

        replyString = GpibRead(GpibDevice)
        Call SplitArrayData(replyString, wavArray)  'Convert the returned string into
                                                    'values.

        Call GpibWrite(":FETC:POW?", GpibDevice)    'Retrieve measured power information
        replyString = GpibRead(GpibDevice)
        Call SplitArrayData(replyString, powArray)  'Convert the returned string into
                                                    'values.
    End Try
End Sub
```

```

'=====
'Display the results (wavelength and power information)
'=====
Console.WriteLine("Wavelength(m) : " + waveLength.ToString())
Console.WriteLine("Power(dBm)      : " + power.ToString())

'=====
'Save data to the internal memory
'=====
'Save the screen capture and result data to the internal memory
Call GpibWrite(":MMEM:STOR SIM2,\"\"\\WLM_IMAGE\"",INT", GpibDevice)
Call GpibWrite(":MMEM:STOR TABL,\"\"\\WLM_TABLE\"",INT", GpibDevice)

'=====
'Transfer the data saved in the internal memory to the PC
'=====
Call GpibWrite(":MMEM:DATA? \"\"\\WLM_IMAGE.BMP\"",INT", GpibDevice)
GpibReadBlockData2File(GpibDevice, "WLM_IMAGE.BMP")
Call GpibWrite(":MMEM:DATA? \"\"\\WLM_TABLE.CSV\"",INT", GpibDevice)
GpibReadBlockData2File(GpibDevice, "WLM_TABLE.CSV")

Console.ReadLine()                                'Wait for the user to press the
                                                    'Enter key
Catch ex As Exception                             'Error handling
    Console.WriteLine(ex.Message)                 'Display the error message that
                                                    'occurred
Console.ReadLine()                                'Wait for the user to the Enter
                                                    'key

End Try
End Sub

'=====
'Function for sending character strings to GP-IB
'=====
Sub GpibWrite(ByVal commandStr As String, ByRef gpib As Device)
    gpib.Write(commandStr)                        'Send data
End Sub

'=====
'Function for reading a line of data from GP-IB
'=====
Function GpibRead(ByRef gpib As Device) As String
    GpibRead = gpib.ReadString()                  'Receive data
    Exit Function
End Function

```

## 2.6 Sample Program

---

```
'=====
'Function for reading block data and saving it to a file
'=====
Function GpibReadBlockData2File(ByRef gpib As Device, ByVal filename As String) As Integer
    Dim headerLen As Integer
    Dim dataLen As Integer
    Dim dataByte As Byte()
    Dim file As New FileStream(filename, FileMode.Create, FileAccess.Write)

    If String.Compare(gpib.ReadString(1), "#") <> 0 Then      'Retrieve the first character
        GpibReadBlockData2File = -1                        'Error if the first character is
                                                            'not a "#"
        Exit Function
    End If
    headerLen = Integer.Parse(gpib.ReadString(1))           'Size of the area containing the
                                                            'data length information
    dataLen = Integer.Parse(gpib.ReadString(headerLen))     'Retrieve the data length
                                                            'information

    While dataLen > 1024
        dataByte = gpib.ReadByteArray(1024)               'Read data 1024 bytes at a time
        file.Write(dataByte, 0, dataByte.Length)          'Write the retrieved data to the
                                                            'file
        dataLen = dataLen - dataByte.Length
    End While

    dataByte = gpib.ReadByteArray(dataLen)                 'Retrieve the last piece of data
    file.Write(dataByte, 0, dataByte.Length)              'Write the retrieved data to the
                                                            'file

    file.Close()
    GpibReadBlockData2File = 0
End Function
End Module
```

### Execution Example

```
Wavelength(m): 1.55252398E-06
Power(dBm)    : 3.43060397
```

## Sample Program 3

Sample program for performing drift analysis via GP-IB. Example of the -MW (multi-wavelength) suffix code model.

### Source Code

```
Imports System
Imports NationalInstruments.NI4882

Module GpibDriftMeasure
'
'Sample Program for Performing Drift Analysis via GP-IB
'
  Sub Main()
    Try
      Dim GpibDevice As Device
      Dim wlmAddr As Integer
      Dim replyString As String
      Dim peakNum As Integer
      Dim refPowData, refWavData As Double()
      Dim maxPowData, maxWavData As Double()
      Dim minPowData, minWavData As Double()
      Dim dropInfo As Double()

      '=====
      'Wavelength meter information
      '=====
      wlmAddr = 7                                     'Wavelength meter GP-IB address
      GpibDevice = New Device(0, wlmAddr)             'Open GP-IB

      '=====
      ' Set the wavelength meter's measurement conditions
      '=====
      Call GpibWrite("*RST", GpibDevice)               'Reset the instrument
      Call GpibWrite(":CALC2:PTHR:MODE REL", GpibDevice) 'Set the threshold to relative mode
      Call GpibWrite(":CALC2:PTHR 15", GpibDevice)       'Set the threshold to 15 db
      Call GpibWrite(":UNIT:WL NM", GpibDevice)          'Set the wavelength unit to nm
      Call GpibWrite(":UNIT:POW DBM", GpibDevice)        'Set the power unit to dBm

      'Perform a Single measurement to obtain the reference for the drift measurement
      Call GpibWrite(":INIT;*OPC?", GpibDevice)          'Execute a Single measurement and
                                                         'wait for the measurement to
                                                         'complete
      GpibRead(GpibDevice)                             'Read the measurement complete wait
                                                         '(*OPC?) response
      Call GpibWrite(":CALC3:DRIF ON", GpibDevice)       'Turn drift analysis on
      '=====
      ' Execute a measurement
      '=====
      Call GpibWrite(":INIT:CONT ON", GpibDevice)        'Start a Repeat measurement

      For count As Integer = 1 To 60                   'Wait 1 minute
        Threading.Thread.Sleep(1000)
        Console.Write(".")
      Next
      Console.WriteLine("")
      Call GpibWrite(":INIT:CONT OFF", GpibDevice)      'Stop the Repeat measurement
```

## 2.6 Sample Program

---

```
'=====
'Retrieve the measured results
'=====
Call GpibWrite(":CALC3:POIN?", GpibDevice)           'Retrieve the number of data
                                                    'values

replyString = GpibRead(GpibDevice)
peakNum = Integer.Parse(replyString)
refPowData = New Double(peakNum - 1) {}
refWavData = New Double(peakNum - 1) {}
maxPowData = New Double(peakNum - 1) {}
maxWavData = New Double(peakNum - 1) {}
minPowData = New Double(peakNum - 1) {}
minWavData = New Double(peakNum - 1) {}
dropInfo = New Double(peakNum - 1) {}

'Retrieve the results (reference values)
Call GpibWrite(":CALC3:DRIF:REF ON", GpibDevice)
Call GpibWrite(":CALC3:DATA? POW", GpibDevice)       'Retrieve Ref Power
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, refPowData)
Call GpibWrite(":CALC3:DATA? WAV", GpibDevice)       'Retrieve Ref Wavelength
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, refWavData)

'Retrieve the results (max values)
Call GpibWrite(":CALC3:DRIF:PRES", GpibDevice)
Call GpibWrite(":CALC3:DRIF:MAX ON", GpibDevice)
Call GpibWrite(":CALC3:DATA? POW", GpibDevice)       'Retrieve MAX Power
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, maxPowData)
Call GpibWrite(":CALC3:DATA? WAV", GpibDevice)       'Retrieve MAX Wavelength
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, maxWavData)

'Retrieve the results (min values)
Call GpibWrite(":CALC3:DRIF:PRES", GpibDevice)
Call GpibWrite(":CALC3:DRIF:MIN ON", GpibDevice)
Call GpibWrite(":CALC3:DATA? POW", GpibDevice)       'Retrieve MIN Power
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, minPowData)
Call GpibWrite(":CALC3:DATA? WAV", GpibDevice)       'Retrieve MIN Wavelength
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, minWavData)

'Retrieve drop information
Call GpibWrite(":CALC3:DATA? DROP", GpibDevice)
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, dropInfo)

GpibDevice.Dispose()                               'GP-IB Close

'=====
'Display the measured results
'=====
Console.Write("No.          |")                  'Display the peak number
For idx As Integer = 0 To peakNum - 1
    Console.Write((idx + 1).ToString() + "          |")
Next
Console.WriteLine()
Console.Write("REF WL      |")                  'Display the reference wavelength
For idx As Integer = 0 To peakNum - 1
    Console.Write(refWavData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("REF POWER   |")                  'Display the reference power
For idx As Integer = 0 To peakNum - 1
    Console.Write(refPowData(idx).ToString() + " | ")
Next
```

```

Console.WriteLine()
Console.Write("MAX WL      |")
'Display the maximum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MAX POWER  |")
'Display the maximum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN WL      |")
'Display the minimum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN POWER  |")
'Display the minimum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.ReadLine()
Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.ReadLine()
End Try
End Sub

'=====
'Function for sending character strings to GP-IB
'=====
Sub GpibWrite(ByVal commandStr As String, ByRef gpib As Device)
    gpib.Write(commandStr)
    'Send data
End Sub

'=====
'Function for reading a line of data from GP-IB
'=====
Function GpibRead(ByRef gpib As Device) As String
    GpibRead = gpib.ReadString()
    'Receive data
    Exit Function
End Function

```

## 2.6 Sample Program

---

```
'=====
'Function for splitting the CALC3 results into an array
'=====
Sub SplitArrayData(ByVal dataString As String, ByRef dataArray As Double())
    Dim peakNum As Integer
    Dim arrayDataStr As String() = dataString.Split(",")    'Split the string by using a
                                                            'comma as the delimiter

    peakNum = arrayDataStr.Length
    For idx As Integer = 0 To arrayDataStr.Length - 1
        dataArray(idx) = Convert.ToDouble(arrayDataStr(idx))    'Convert the split strings into
                                                                'values
    Next
End Sub
End Module
```

### Execution Example

No.	1	2	3	4	5	
REF WL	1.30678832E-06	1.30756981E-06	1.30835238E-06	1.30913541E-06	1.30991969E-06	
REF POWER	-13.4899875	-9.04694537	-2.9512995	-3.29214313	-13.1556519	
MAX WL	-----	1.30757036E-06	1.3083528E-06	1.30913604E-06	-----	
MAX POWER	-----	-8.81158076	-0.665845116	-3.21870974	-----	
MIN WL	-----	1.30756953E-06	1.30835221E-06	1.30913538E-06	-----	
MIN POWER	-----	-10.2276251	-3.02598662	-6.67785905	-----	

## Sample Program 4

Sample program for performing drift analysis via GP-IB. Example of the -SW (single-wavelength) suffix code model.

### Source Code

```
Imports System
Imports NationalInstruments.NI4882

Module GpibDriftMeasure
'
'Sample Program for Performing Drift Analysis via GP-IB
'
  Sub Main()
    Try
      Dim GpibDevice As Device
      Dim wlmAddr As Integer
      Dim replyString As String
      Dim peakNum As Integer
      Dim refPowData, refWavData As Double()
      Dim maxPowData, maxWavData As Double()
      Dim minPowData, minWavData As Double()
      Dim dropInfo As Double()

      '=====
      'Wavelength meter information
      '=====
      wlmAddr = 7                                'Wavelength meter GP-IB address
      GpibDevice = New Device(0, wlmAddr)        'Open GP-IB

      '=====
      'Set the wavelength meter's measurement conditions
      '=====
      Call GpibWrite("*RST", GpibDevice)          'Reset the instrument
      Call GpibWrite(":UNIT:WL NM", GpibDevice)    'Set the wavelength unit to nm
      Call GpibWrite(":UNIT:POW DBM", GpibDevice)   'Set the power unit to dBm

      'Perform a Single measurement to obtain the reference for the drift measurement
      Call GpibWrite(":INIT;*OPC?", GpibDevice)    'Execute a Single measurement and
                                                    'wait for the measurement to
                                                    'complete
      GpibRead(GpibDevice)                        'Read the measurement complete wait
                                                    '(*OPC?) response
      Call GpibWrite(":CALC3:DRIF ON", GpibDevice) 'Turn drift analysis on

      '=====
      'Execute a measurement
      '=====
      Call GpibWrite(":INIT:CONT ON", GpibDevice)  'Start a Repeat measurement

      For count As Integer = 1 To 60              'Wait 1 minute
        Threading.Thread.Sleep(1000)
        Console.WriteLine(".")
      Next
      Console.WriteLine("")
      Call GpibWrite(":INIT:CONT OFF", GpibDevice) 'Stop the Repeat measurement
    
```



## 2.6 Sample Program

---

```
'=====
'Retrieve the measured results
'=====
Call GpibWrite(":CALC3:POIN?", GpibDevice)           'Retrieve the number of data
                                                    'values

replyString = GpibRead(GpibDevice)
peakNum = Integer.Parse(replyString)
refPowData = New Double(peakNum - 1) {}
refWavData = New Double(peakNum - 1) {}
maxPowData = New Double(peakNum - 1) {}
maxWavData = New Double(peakNum - 1) {}
minPowData = New Double(peakNum - 1) {}
minWavData = New Double(peakNum - 1) {}
dropInfo = New Double(peakNum - 1) {}

'Retrieve the results (reference values)
Call GpibWrite(":CALC3:DRIF:REF ON", GpibDevice)
Call GpibWrite(":CALC3:DATA? POW", GpibDevice)       'Retrieve Ref Power
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, refPowData)
Call GpibWrite(":CALC3:DATA? WAV", GpibDevice)       'Retrieve Ref Wavelength
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, refWavData)

'Retrieve the results (max values)
Call GpibWrite(":CALC3:DRIF:PRES", GpibDevice)
Call GpibWrite(":CALC3:DRIF:MAX ON", GpibDevice)
Call GpibWrite(":CALC3:DATA? POW", GpibDevice)       'Retrieve MAX Power
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, maxPowData)
Call GpibWrite(":CALC3:DATA? WAV", GpibDevice)       'Retrieve MAX Wavelength
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, maxWavData)

'Retrieve the results (min values)
Call GpibWrite(":CALC3:DRIF:PRES", GpibDevice)
Call GpibWrite(":CALC3:DRIF:MIN ON", GpibDevice)
Call GpibWrite(":CALC3:DATA? POW", GpibDevice)       'Retrieve MIN Power
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, minPowData)
Call GpibWrite(":CALC3:DATA? WAV", GpibDevice)       'Retrieve MIN Wavelength
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, minWavData)

'Retrieve drop information
Call GpibWrite(":CALC3:DATA? DROP", GpibDevice)
replyString = GpibRead(GpibDevice)
SplitArrayData(replyString, dropInfo)

GpibDevice.Dispose()                               'GP-IB Close
```

```

'=====
'Display the measured results
'=====
Console.Write("No.          |")                                'Display the peak number
For idx As Integer = 0 To peakNum - 1
    Console.Write((idx + 1).ToString() + "          |")
Next
Console.WriteLine()
Console.Write("REF WL          |")                                'Display the reference wavelength
For idx As Integer = 0 To peakNum - 1
    Console.Write(refWavData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("REF POWER      |")                                'Display the reference power
For idx As Integer = 0 To peakNum - 1
    Console.Write(refPowData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("MAX WL          |")                                'Display the maximum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MAX POWER      |")                                'Display the maximum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN WL          |")                                'Display the minimum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN POWER      |")                                'Display the minimum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.ReadLine()
Catch ex As Exception
    Console.WriteLine(ex.Message)                                'Error handling
                                                                'Display the error message that
                                                                'occurred
    Console.ReadLine()                                           'Wait for the user to press the
                                                                'Enter key

End Try
End Sub

```

## 2.6 Sample Program

---

```
'=====
'Function for sending character strings to GP-IB
'=====
Sub GpibWrite(ByVal commandStr As String, ByRef gpib As Device)
    gpib.Write(commandStr)           'Send data
End Sub

'=====
'Function for reading a line of data from GP-IB
'=====
Function GpibRead(ByRef gpib As Device) As String
    GpibRead = gpib.ReadString()     'Receive data

    Exit Function
End Function

'=====
'Function for splitting the CALC3 results into an array
'=====
Sub SplitArrayData(ByVal dataString As String, ByRef dataArray As Double())
    Dim peakNum As Integer
    Dim arrayDataStr As String() = dataString.Split(",") 'Split the string by using a
                                                         'comma as the delimiter

    peakNum = arrayDataStr.Length
    For idx As Integer = 0 To arrayDataStr.Length - 1
        dataArray(idx) = Convert.ToDouble(arrayDataStr(idx)) 'Convert the split strings into
                                                             'values
    Next
End Sub
End Module
```

### Execution Example

No.	1	
REF WL	1.55252293E-06	
REF POWER	3.37379314	
MAX WL	1.55252431E-06	
MAX POWER	3.43271086	
MIN WL	1.55252289E-06	
MIN POWER	3.3485737	

## 3.1 Using the Ethernet Interface to Connect the Devices

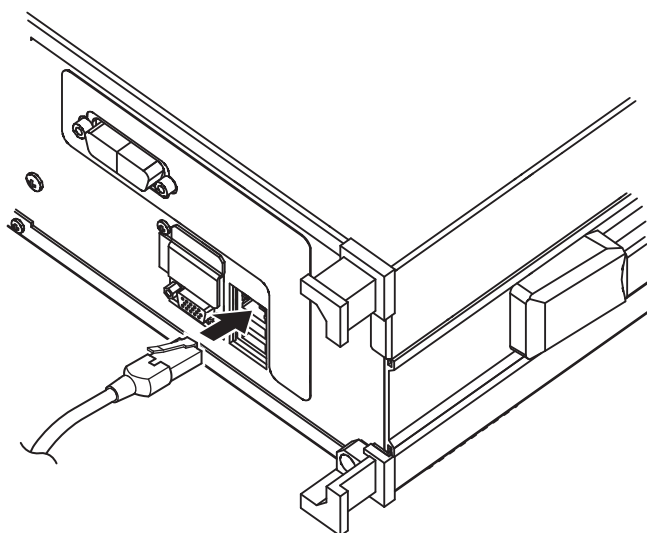
You can use the instrument's Ethernet interface to connect to a LAN and control the instrument from a PC.

### Ethernet Interface Specifications

Number of ports	1
Electrical and mechanical specifications	IEEE802.3 compliant
Transmission system	Ethernet (10BASE-T/100BASE-TX/1000BASE-T)
Data rate	10 Mbps, 100 Mbps, 1000 Mbps
Communication protocol	TCP/IP
Connector	RJ45
Port number	Any port between 1024 to 65535 except 1025 and 20001
Simultaneous connections	1

### Connection Procedure

Connect a UTP (Unshielded Twisted-Pair) or STP (Shielded Twisted-Pair) cable that is connected to a hub or other network device to the Ethernet port on the instrument rear panel.



#### Precautions to Be Taken When Connecting Cables

- To connect the instrument to a PC, be sure to use straight cables and to connect through a hub or router.
- Use straight category 5 or better UTP cables.

---

## 3.2 Ethernet Port Features

### Remote Control

You can use the ETHERNET port to control the instrument over a network.  
You can control the instrument remotely using the commands are the same as those used to control the instrument through the GP-IB interface.  
The instrument supports VXI-11 control.

### Remoto commands

#### Message Terminator

The following message terminators can be used on the instrument.

##### Program Message Terminator

LF (line feed) character

The instrument recognizes LF as an ASCII line feed (0Ah) and CR (0Dh) in CR+LF as a white space character, so you can also use CR+LF as a message terminator.

##### Response Message Terminator

The response message terminator is CR+LF.

#### Data Queries

- A query command has a question mark appended to the end of the command.
- For a query that has a parameter, <wsp>+<parameter> is appended to the question mark.
- When the instrument receives a query command, it places the response to the command in the output buffer.
- The data is held in the output buffer until an input statement is received from the controller or a new query command is received.
- If multiple query commands are concatenated with semicolons, the instrument places the responses to all the query commands in the output buffer.  
In this case, all the data are output collectively.

### Remote Monitoring

You can use the ETHERNET port to monitor the instrument screen or control the instrument 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 instrument 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 instrument.

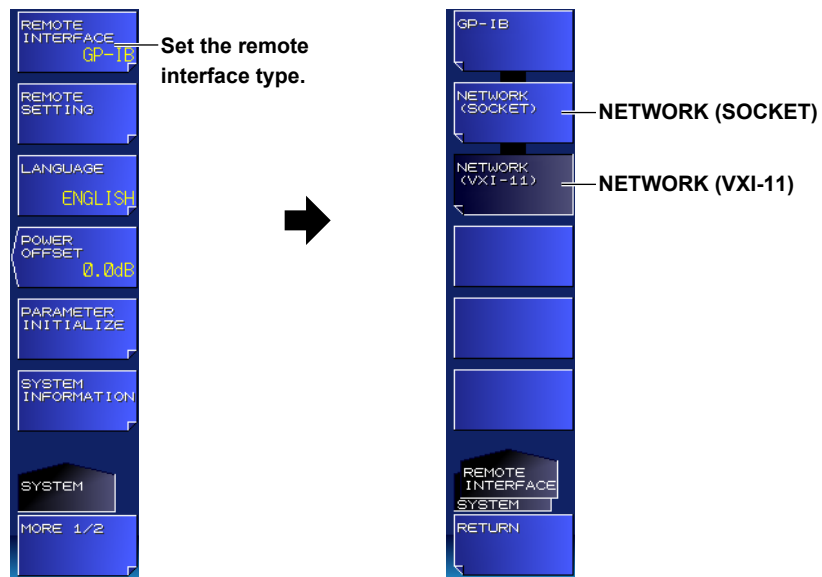
- Measurement data (CSV format)
- Setup data (WS1 format)
- Screen capture data (BMP format)
- Logging data (WG1 format)

## 3.3 Ethernet Configuration

### Procedure

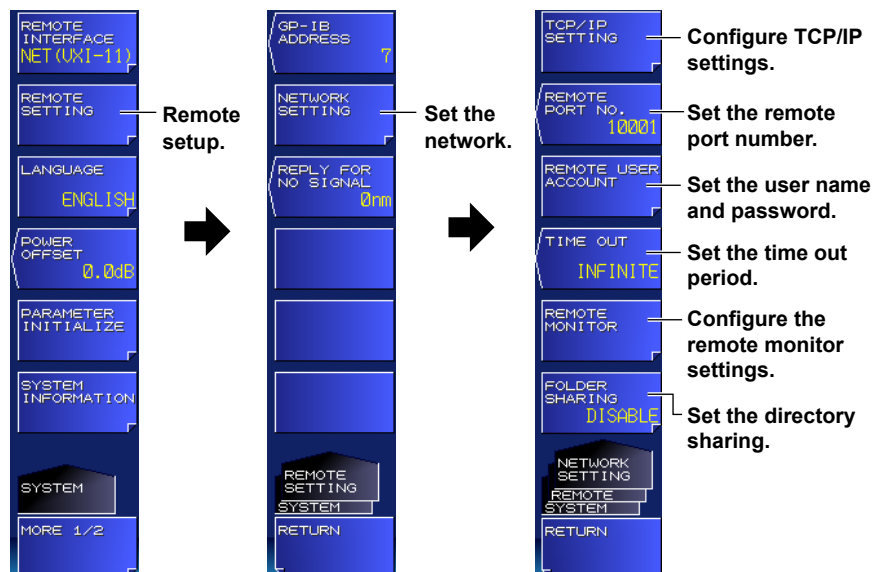
#### Selecting the Communication Interface

1. Press **SYSTEM**.  
A system setup menu appears.
2. Press the **REMOTE INTERFACE** soft key.  
A remote interface setup menu appears.
3. Press the **NETWORK (SOCKET)** or **NETWORK (VXI-11)** soft key.  
The setup menu returns to the previous display, and the specified remote interface type appears on the soft key.



#### Configuring Network Settings

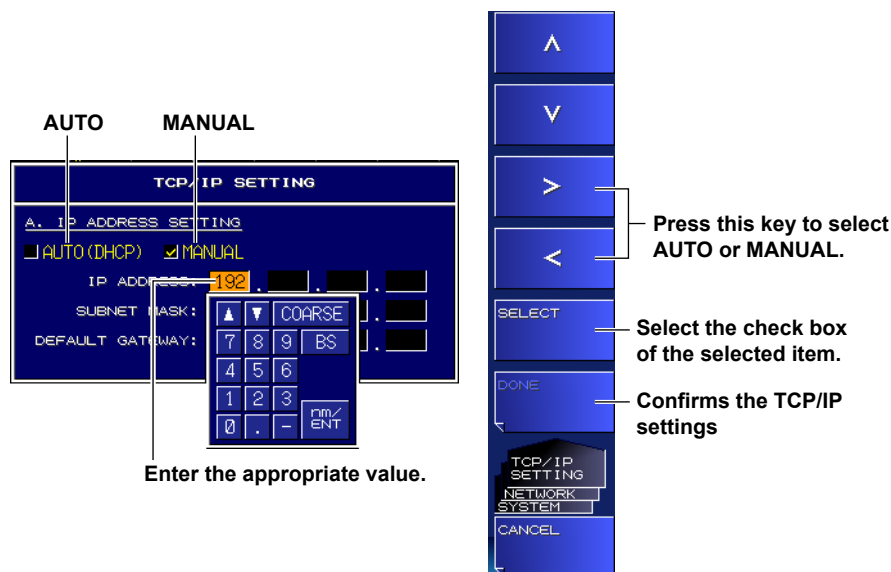
4. Press the **REMOTE SETTING** soft key.  
A remote setup screen appears..
5. Press the **NETWORK SETTING** soft key.  
The Ethernet setup menu appears.



### 3.3 Ethernet Configuration

- **Configuring the TCP/IP Settings**

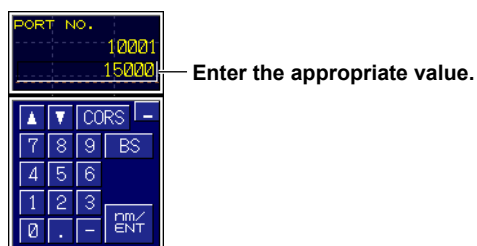
6. Press the **TCP/IP SETTING** soft key.  
The TCP/IP setup menu appears.



7. Press the **<** or **>** soft key to select **AUTO (DHCP)** or **MANUAL**.
8. Press the **SELECT** soft key to select the check box of the selected item.
9. If you select **MANUAL**, set the IP address, subnet mask, and default gateway. Use the arrow soft keys to select an item, and then press **ENTER**.  
If you select **AUTO**, proceed to step 11.
10. Use the arrow keys or numeric keypad to enter the appropriate value, and then press **ENTER**.
11. When you finish entering all the settings, press the **DONE** soft key.

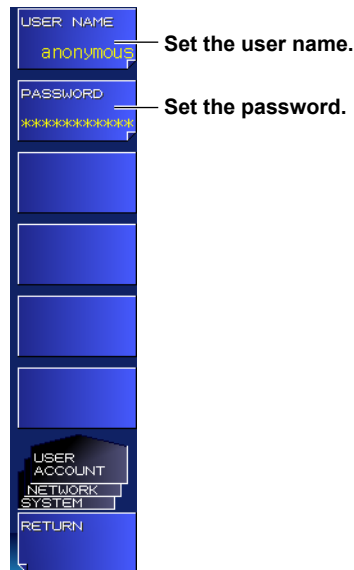
- **Setting the Remote Port Number (not used with the VXI-11)**

6. Press the **REMOTE PORT NO.** soft key.  
The port number setup screen appears.
7. Enter the port number using the arrow keys or numeric keypad.



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

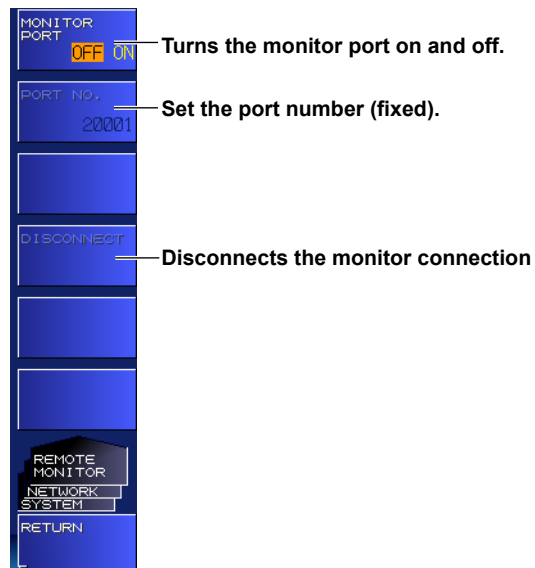
6. Press the **REMOTE USER ACCOUNT** soft key.  
A setup menu for the user name and password appears.



7. Press the **USER NAME** soft key.  
The user name setup screen appears. For instructions on how to enter text, see section 3.3 in the getting started guide, IM AQ6150-02EN.
8. Press the **PASSWORD** soft key.  
The password setup screen appears. For instructions on how to enter text, see section 3.3 in the getting started guide, IM AQ6150-02EN.

- **Configuring the Remote Monitor Settings**

6. Press the **REMOTE MONITOR** soft key.  
The remote monitor setup menu appears.



7. 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.



### 3.3 Ethernet Configuration

---

- **Disconnecting the Monitor Connection**

8. Press the **DISCONNECT** soft key.  
The monitor connection from the PC is disconnected.

- **Setting Directory Sharing**

6. Press the **FOLDER SHARING** soft key.  
A directory sharing setup menu appears.



7. Press the **READ ONLY** soft key.  
The user area directory of the instrument is shared (read only).

- **Disabling Directory Sharing**

8. Press the **DISABLE** soft key.  
The sharing of the user area directory is disabled.

- **Setting the Timeout Period (for Network (SOCKET) communication)**

6. Press the TIME OUT soft key. A parameter input window appears.
7. Enter the timeout period in the parameter input window.

**Explanation**

Configure the instrument TCP/IP settings.

**Configuring the TCP/IP Settings**

To connect the instrument to a network, you must set the instrument IP address correctly. If a DHCP server is available on the network that the instrument is connected to, an IP address is automatically assigned to the instrument.

If a DHCP server is available, set IP ADDRESS SETTING to AUTO.

For details about the network that you intend to connect the instrument to, contact your network administrator.

**REMOTE PORT NO. (not used with the VXI-11)**

Set the port number to use to control the instrument remotely over the ETHERNET port.

You can specify any port between 1024 to 65535 except 1025 and 20001.

**User Authentication (not used with the VXI-11)**

If you want to connect to the instrument from your PC over the ETHERNET port, user authentication is required.

If the user name is anonymous, you do not have to enter a password.

The instrument supports plaintext authentication and the MD5 algorithm (RSA Data Security, Inc. MD5 Message Digest Algorithm).

Set the user name and password using up to 11 characters.

The default user name is anonymous.

**Remote Control Using Commands (for SOCKET)**

**You can control the instrument through the ETHERNET port.**

The remote commands are the same as those used to control the instrument through the GP-IB interface.

**Switching the Interface**

Set the interface that you want to use for remote control to GP-IB or ETHERNET.

If you select GP-IB or press the LOCAL key, the LAN remote connection status is reset.

In all other cases, the connection is retained until the controller disconnects.

**SRQ Interrupts**

SRQ interrupts do not occur while the instrument is being remotely controlled through the ETHERNET port.

**Status Registers**

The status registers operate in the same manner as when the instrument is being controlled remotely through the GP-IB interface. You can use the \*STB? command to read the status register in the same manner as in serial polling over the GP-IB interface.

**Talker Data Transmission**

When the instrument receives talker data from the controller PC, it sends the data to the controller PC's buffer. Retrieve the data by reading the controller PC's buffer data.

**Connection**

The instrument can connect to a single controller (PC). If a connection with a controller is established and a connection request is received from another controller, the current connection will be retained (a new connection will not be established).

**Timeout Period (not used with the VXI-11)**

If no communication takes place for the specified time in remote mode, the communication is automatically disconnected, and the mode changes to local.

When you change the timeout period, the elapsed time is reset.

The time can be set to INFINITE (0 s), 1 to 21600 s (6 hours).

## Commands Necessary for Remote Control over a ETHERNET port (for SOCKET)

Authentication through an OPEN command is necessary to establish a remote connection over a ETHERNET port.

Connection cannot be established without authentication.

### OPEN

Function Sends a user name and starts user authentication.

Syntax OPEN<wsp>"username"  
username = User name

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

Description The OPEN command causes the following procedure to be executed.

#### For Plaintext Authentication

1. OPEN "username" is sent to the instrument. The response message from the instrument is received.
2. The retrieved message "AUTHENTICATE CRAM-MD5." is confirmed.
3. The password is sent to the instrument (when the user name is anonymous, the password can be anything).
4. When a "READY" message is received from the instrument, the authentication is complete. The instrument REMOTE LED illuminates, and remote command transmission becomes possible.  
If the user name or password is incorrect, authentication will fail, and the connection will be released.

#### For Encrypted Authentication

1. OPEN "username" is sent to the instrument. The response message from the instrument is received.
2. The retrieved message "AUTHENTICATE CRAM-MD5." is confirmed.
3. "AUTHENTICATE CRAM-MD5 OK" is sent to the instrument. The response message (challenge string) from the instrument is received.
4. The MD5 hash of the received challenge string and password is calculated (when the user name is anonymous, the password can be anything).
5. The resultant hash data (lowercase hexadecimal × 32 characters) is sent to the instrument, and the response message is received.
6. When a "READY" message is received from the instrument, the authentication is complete. The instrument REMOTE LED illuminates, and remote command transmission becomes possible.  
If the user name or password is incorrect, authentication will fail, and the connection will be released.

### CLOSE

Function Turns off the connection and switches to local mode.

Syntax CLOSE

Example CLOSE

---

### Note

- If you start the instrument 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 1/6" to "STEP 6/6.")
  - When the start procedure is finished and the measurement screen appears, it may take a few more minutes before you can access the instrument from a PC over the network.
-

- **Measurement Time When the Remote Monitoring Function Is in Use**

When you use the remote monitoring function to monitor this instrument's screen or control this instrument from a remote monitoring software running on a PC, the measurement time may take longer than when the remote monitoring function is not used.

## 3.4 Sample Program (for SOCKET)

The following shows an example of controlling the instrument remotely using the Ethernet port.

Model: PC-AT Computer  
Language: Visual Basic 2008  
Component: .NET Framework 3.5

### Sample Program 1

Sample program for making one Single measurement and displaying the results (wavelength information, power information, and FP-LD analysis results) on the screen via Ethernet. Example of the -MW (multi-wavelength) suffix code model.

#### Source Code

```
Imports System
Imports System.IO
Imports System.Net.Sockets
Imports System.Text

Module EtherSingleMeasure
'
' Sample Program for Making One Single Measurement and Displaying the Results (Wavelength
' Information, Power Information, and FP-LD Analysis Results) on the Screen via Ethernet
'
Sub Main()
    Try
        Dim wlmAddr As String
        Dim wlmPort As Integer
        Dim sockStream As NetworkStream
        Dim tcpObj As TcpClient
        Dim replyString As String
        Dim wavArray As Double()
        Dim powArray As Double()
        Dim fwhm, ctrWl, totalPwr, sigma As Double
        Dim username, passwd As String
        Dim maxPeakPower, maxPeakWl As Double

        '=====
        'Wavelength meter information
        '=====
        wlmAddr = "192.168.0.1"           'Wavelength meter IP address
        wlmPort = 10001                 'Remote port number
        username = "anonymous"          'User name
        passwd = ""                     'Password

        '=====
        'TCP connection
        '=====
        tcpObj = New TcpClient
        tcpObj.Connect(wlmAddr, wlmPort) 'TCP connection
        sockStream = tcpObj.GetStream()
        tcpObj.NoDelay = True           'Enable TCP_NODELAY

        '=====
        'Execute authentication
        '=====
        Dim recvBuffer As String
        TcpWriteLine("open "" + username + """, sockStream) 'Send an OPEN command with the
                                                            'user name

        recvBuffer = TcpReadLine(sockStream)
        If String.Compare(recvBuffer, "AUTHENTICATE CRAM-MD5") <> 0 Then
            sockStream.Dispose()
            Exit Sub 'Error if the response is not
                    'AUTHENTICATE CRAM-MD5
        End If
    End Try
End Sub
```

```

TcpWriteLine (passwd, sockStream)           'Send the password
recvBuffer = TcpReadLine(sockStream)

If String.Compare(recvBuffer, "ready") <> 0 Then
    sockStream.Dispose()
    Exit Sub                                'Authentication failure
End If

'=====
'Set the wavelength meter's measurement conditions
'=====
Call TcpWriteLine("*RST", sockStream)        'Reset the instrument
Call TcpWriteLine(":CALC2:PTHR:MODE REL", sockStream) 'Set the peak detection threshold
                                                'setting to relative mode
Call TcpWriteLine(":CALC2:PTHR 15", sockStream) 'Set the peak detection threshold
                                                'to 15 db
Call TcpWriteLine(":UNIT:WL NM", sockStream)  'Set the wavelength unit to nm
Call TcpWriteLine(":UNIT:POW DBM", sockStream) 'Set the power unit to dBm
Call TcpWriteLine(":DISP:WIND2:STAT ON", sockStream) 'Enable the graph display

'=====
'Execute measurement and retrieve data
'=====
'Use the READ command to execute measurement and retrieve data.
'Use the FETC command to retrieve measured power information data.
Call TcpWriteLine(":READ:ARR:POW:WAV?", sockStream) 'Execute a Single measurement
                                                'and retrieve wavelength data

replyString = TcpReadLine(sockStream)
Call SplitArrayData(replyString, wavArray)        'Store wavelength information in
                                                'an array

Call TcpWriteLine(":FETC:ARR:POW?", sockStream)    'Retrieve measured power
                                                'information

replyString = TcpReadLine(sockStream)
Call SplitArrayData(replyString, powArray)        'Store power information in an
                                                'array

'=====
'Display the results (wavelength and power information)
'=====
Console.WriteLine("No. |Wavelength(m) |Power(dBm) ")
For idx As Integer = 1 To wavArray.Length
    Console.WriteLine((idx).ToString + "|" + wavArray(idx - 1).ToString() + "|" + _
        powArray(idx - 1).ToString())
Next

'=====
'Retrieve maximum power peak information
'=====
Call TcpWriteLine(":FETC:POW? MAX", sockStream)    'Retrieve power information by
                                                'specifying the peak with the
                                                'maximum power

replyString = TcpReadLine(sockStream)
maxPeakPower = Convert.ToDouble(replyString)
Call TcpWriteLine(":FETC:POW:WAV?", sockStream)    'Retrieve the wavelength of the
                                                'peak specified by
                                                ':FETC:POW? MAX

replyString = TcpReadLine(sockStream)
maxPeakWl = Convert.ToDouble(replyString)
Console.WriteLine("Highest Peak Power      :" + maxPeakPower.ToString + " dBm")
Console.WriteLine("Highest Peak Wavelength:" + maxPeakWl.ToString + " nm")
'=====
'Retrieve the FP-LD analysis results
'=====
Call TcpWriteLine(":CALC3:FPER ON", sockStream)    'Enable FP-LD analysis
Call TcpWriteLine(":CALC3:FPER:FWHM?", sockStream) 'Retrieve FWHM
replyString = TcpReadLine(sockStream)
fwhm = Convert.ToDouble(replyString)
Call TcpWriteLine(":CALC3:FPER:MEAN?", sockStream) 'Retrieve Center WL
replyString = TcpReadLine(sockStream)
ctrWl = Convert.ToDouble(replyString)

```

### 3.4 Sample Program (for SOCKET)

```
Call TcpWriteLine(":CALC3:FPER:POW?", sockStream)      'Retrieve Total Power
replyString = TcpReadLine(sockStream)
totalPwr = Convert.ToDouble(replyString)
Call TcpWriteLine(":CALC3:FPER:SIGM?", sockStream)      'Retrieve  $\sigma$ 
replyString = TcpReadLine(sockStream)
sigma = Convert.ToDouble(replyString)
'Display the results (FP-LD analysis results)
Console.WriteLine("====FP-LD Analysis====")
Console.WriteLine("FWHM          : " + (fwhm * 1000000000).ToString + "nm")
Console.WriteLine("Sigma          : " + (sigma * 1000000000).ToString + "nm")
Console.WriteLine("CTR WL         : " + (ctrWl * 1000000000).ToString + "nm")
Console.WriteLine("TOTAL PWR       : " + totalPwr.ToString + "dBm")

'=====
'Save data to the internal memory
'=====
'Save the screen capture and result data to the internal memory
Call TcpWriteLine(":MMEM:STOR SIM2,\"\"\\WLM_IMAGE\"\",INT", sockStream)
Call TcpWriteLine(":MMEM:STOR TABL,\"\"\\WLM_TABLE\"\",INT", sockStream)

'=====
'Transfer the data saved in the internal memory to the PC
'=====
Call TcpWriteLine(":MMEM:DATA? \"\"\\WLM_IMAGE.BMP\"\",INT", sockStream)
TcpReadBlockData2File(sockStream, "WLM_IMAGE.BMP")
Call TcpWriteLine(":MMEM:DATA? \"\"\\WLM_TABLE.CSV\"\",INT", sockStream)
TcpReadBlockData2File(sockStream, "WLM_TABLE.CSV")
sockStream.Dispose()      'Close TCP

Console.ReadLine()
Catch ex As Exception      'Error handling
    Console.WriteLine(ex.Message)      'Display the error message that
                                      'occurred
Console.ReadLine()      'Wait for the user to press the
                                      'Enter key

End Try
End Sub

'=====
'Function for sending character strings to the TCP Socket
'=====
Sub TcpWriteLine(ByVal commandStr As String, ByRef stream As NetworkStream)
    Dim writer As StreamWriter = New StreamWriter(stream, Encoding.ASCII)
    Dim ByteLf As Byte() = New Byte() {10}
    writer.NewLine = Encoding.ASCII.GetString(ByteLf)      'The line feed code is LF.
    writer.AutoFlush = True
    writer.WriteLine(commandStr) 'Send data
End Sub

'=====
'Function for reading a line of data from the TCP Socket
'=====
Function TcpReadLine(ByRef stream As NetworkStream) As String
    Dim reader As StreamReader = New StreamReader(stream, Encoding.ASCII)
    TcpReadLine = reader.ReadLine()      'Receive data
    Exit Function
End Function

'=====
'Function for reading block data from the TCP Socket and saving the 'data to a file
'=====
Function TcpReadBlockData2File(ByRef stream As NetworkStream, ByVal filename As String) _
    As Integer
    Dim headerLen As Integer
    Dim dataLen As Integer
    Dim readLen As Integer
```

```

Dim file As New FileStream(filename, FileMode.Create, FileAccess.Write)
Dim recvBuffer As Byte() = New Byte(1024) {}
Dim ByteSharp As Byte = Asc("#")

stream.Read(recvBuffer, 0, 1) 'Retrieve the first character
If recvBuffer(0) <> ByteSharp Then 'Error if the first character is
                                   'not a "#"

    TcpReadBlockData2File = -1
    Exit Function
End If
stream.Read(recvBuffer, 0, 1)
headerLen = Integer.Parse(Encoding.ASCII.GetString(recvBuffer)) 'Size of the area
                                                                'containing the data
                                                                'length information

stream.Read(recvBuffer, 0, headerLen) 'Read the data length information
                                     'area

dataLen = Integer.Parse(Encoding.ASCII.GetString(recvBuffer)) 'Retrieve the data length
                                                                'information

While dataLen > 1024
    readLen = stream.Read(recvBuffer, 0, 1024) 'Read data 1024 bytes at a time
    file.Write(recvBuffer, 0, readLen) 'Write the retrieved data to the
                                     'file

    dataLen = dataLen - readLen
End While
readLen = stream.Read(recvBuffer, 0, recvBuffer.Length) 'Retrieve the last piece of data
file.Write(recvBuffer, 0, dataLen) 'Write the retrieved data to the
                                   'file

file.Close()
TcpReadBlockData2File = 0
End Function

'=====
'Function for splitting the READ/FETC/MEAS results into an array
'=====
Sub SplitArrayData(ByVal dataString As String, ByRef dataArray As Double())
    Dim peakNum As Integer
    Dim arrayDataStr As String() = dataString.Split(",") 'Split the data by using a comma
                                                         'as the delimiter

    peakNum = Integer.Parse(arrayDataStr(0)) 'Retrieve the number of data
                                             'values

    dataArray = New Double(peakNum - 1) {}
    For idx As Integer = 1 To arrayDataStr.Length - 1
        dataArray(idx - 1) = Convert.ToDouble(arrayDataStr(idx)) 'Read all the data values
    Next
End Sub
End Module

```

## Execution Example

```

No. | Wavelength(m) | Power(dBm)
1 | 1.30678822E-06 | -14.3279541
2 | 1.30756963E-06 | -9.42082105
3 | 1.30835228E-06 | -2.23592107
4 | 1.30913555E-06 | -3.93065804
5 | 1.30991986E-06 | -13.5578301
Highest Peak Power      :-2.23592107 dBm
Highest Peak Wavelength:1.30835228E-06 nm
====FP-LD Analysis====
FWHM                   : 1.47415158nm
Sigma                  : 0.625966702nm
CTR WL                 : 1308.55169nm
TOTAL PWR              : 0.782282871dBm

```



## Sample Program 2

Sample program for making one Single measurement and displaying the results (wavelength information and power information) on the screen via Ethernet. Example of the -SW (single-wavelength) suffix code model.

### Source Code

```
Imports System
Imports System.IO
Imports System.Net.Sockets
Imports System.Text

Module EtherSingleMeasure
'
' Sample Program for Making One Single Measurement and Displaying the Results (Wavelength
' Information and Power Information) on the Screen via Ethernet
'
Sub Main()
    Try
        Dim wlmAddr As String
        Dim wlmPort As Integer
        Dim sockStream As NetworkStream
        Dim tcpObj As TcpClient
        Dim replyString As String
        Dim waveLength As Double()
        Dim Power As Double()
        Dim username, passwd As String

        '=====
        'Wavelength meter information
        '=====
        wlmAddr = "192.168.0.1"           'Wavelength meter IP address
        wlmPort = 10001                  'Remote port number
        username = "anonymous"          'User name
        passwd = ""                     'Password

        '=====
        'TCP connection
        '=====
        tcpObj = New TcpClient
        tcpObj.Connect(wlmAddr, wlmPort) 'TCP connection
        sockStream = tcpObj.GetStream()
        tcpObj.NoDelay = True           'Enable TCP_NODELAY

        '=====
        'Execute authentication
        '=====
        Dim recvBuffer As String
        TcpWriteLine("open "" + username + """, sockStream) 'Send an OPEN command with the
                                                            'user name

        recvBuffer = TcpReadLine(sockStream)
        If String.Compare(recvBuffer, "AUTHENTICATE CRAM-MD5") <> 0 Then
            sockStream.Dispose()
            Exit Sub 'Error if the response is not
                    'AUTHENTICATE CRAM-MD5

        End If
        TcpWriteLine(passwd, sockStream) 'Send the password
        recvBuffer = TcpReadLine(sockStream)

        If String.Compare(recvBuffer, "ready") <> 0 Then
            sockStream.Dispose()
            Exit Sub 'Authentication failure
        End If
    End Try
End Sub
```

```

'=====
'Set the wavelength meter's measurement conditions
'=====
Call TcpWriteLine("*RST", sockStream)           'Reset the instrument
Call TcpWriteLine(":UNIT:WL NM", sockStream)    'Set the wavelength unit to nm
Call TcpWriteLine(":UNIT:POW DBM", sockStream)   'Set the power unit to dBm

'=====
'Execute measurement and retrieve data
'=====
'Use the READ command to execute measurement and retrieve data.
'Use the FETC command to retrieve measured power information data.
Call TcpWriteLine(":READ:POW:WAV?", sockStream)  'Execute a Single measurement
                                                'and retrieve wavelength data

replyString = TcpReadLine(sockStream)
Call SplitArrayData(replyString, wavArray)      'Convert the returned string into
                                                'values.

Call TcpWriteLine(":FETC:ARR:POW?", sockStream)  'Retrieve measured power
                                                'information

replyString = TcpReadLine(sockStream)
Call SplitArrayData(replyString, powArray)      'Convert the returned string into
                                                'values.

'=====
'Display the results (wavelength and power information)
'=====
Console.WriteLine(Wavelength(m):" + waveLength.ToString())
Console.WriteLine(Power(dBm)    : " + power. ToString())

'=====
'Save data to the internal memory
'=====
'Save the screen capture and result data to the internal memory
Call TcpWriteLine(":MMEM:STOR SIM2,""WLM_IMAGE"",INT", sockStream)
Call TcpWriteLine(":MMEM:STOR TABL,""WLM_TABLE"",INT", sockStream)

'=====
'Transfer the data saved in the internal memory to the PC
'=====
Call TcpWriteLine(":MMEM:DATA? ""WLM_IMAGE.BMP"",INT", sockStream)
TcpReadBlockData2File(sockStream, "WLM_IMAGE.BMP")
Call TcpWriteLine(":MMEM:DATA? ""WLM_TABLE.CSV"",INT", sockStream)
TcpReadBlockData2File(sockStream, "WLM_TABLE.CSV")
sockStream.Dispose()                          'Close TCP

Console.ReadLine()
Catch ex As Exception                          'Error handling
    Console.WriteLine(ex.Message)              'Display the error message that
                                                'occurred

Console.ReadLine()                            'Wait for the user to press the
                                                'Enter key

End Try
End Sub

'=====
'Function for sending character strings to the TCP Socket
'=====
Sub TcpWriteLine(ByVal commandStr As String, ByRef stream As NetworkStream)
    Dim writer As StreamWriter = New StreamWriter(stream, Encoding.ASCII)
    Dim ByteLf As Byte() = New Byte() {10}
    writer.NewLine = Encoding.ASCII.GetString(ByteLf)    'The line feed code is LF.
    writer.AutoFlush = True
    writer.WriteLine(commandStr)                        'Send data
End Sub

```

### 3.4 Sample Program (for SOCKET)

---

```
'=====
'Function for reading a line of data from the TCP Socket
'=====
Function TcpReadLine(ByRef stream As NetworkStream) As String
    Dim reader As StreamReader = New StreamReader(stream, Encoding.ASCII)
    TcpReadLine = reader.ReadLine()           'Receive data
    Exit Function
End Function

'=====
'Function for reading block data from the TCP Socket and saving the 'data to a file
'=====
Function TcpReadBlockData2File(ByRef stream As NetworkStream, ByVal filename As String) _
    As Integer
    Dim headerLen As Integer
    Dim dataLen As Integer
    Dim readLen As Integer
    Dim file As New FileStream(filename, FileMode.Create, FileAccess.Write)
    Dim recvBuffer As Byte() = New Byte(1024) {}
    Dim ByteSharp As Byte = Asc("#")

    stream.Read(recvBuffer, 0, 1)              'Retrieve the first character
    If recvBuffer(0) <> ByteSharp Then          'Error if the first character is
                                                'not a "#"

        TcpReadBlockData2File = -1
        Exit Function
    End If
    stream.Read(recvBuffer, 0, 1)
    headerLen = Integer.Parse(Encoding.ASCII.GetString(recvBuffer)) 'Size of the area
                                                                    'containing the data
                                                                    'length information

    stream.Read(recvBuffer, 0, headerLen)       'Read the data length information
                                                'area

    dataLen = Integer.Parse(Encoding.ASCII.GetString(recvBuffer)) 'Retrieve the data length
                                                                    'information

    While dataLen > 1024
        readLen = stream.Read(recvBuffer, 0, 1024) 'Read data 1024 bytes at a time
        file.Write(recvBuffer, 0, readLen)          'Write the retrieved data to the
                                                'file

        dataLen = dataLen - readLen
    End While
    readLen = stream.Read(recvBuffer, 0, recvBuffer.Length) 'Retrieve the last piece of data
    file.Write(recvBuffer, 0, dataLen)              'Write the retrieved data to the
                                                'file

    file.Close()
    TcpReadBlockData2File = 0
End Function
End Module
```

### Execution Example

Wavelength(m): 1.55252398E-06  
Power(dBm) : 3.43060397

## Sample Program 3

Sample program for performing drift analysis via Ethernet. Example of the -MW (multi-wavelength) suffix code model.

### Source Code

```
Imports System
Imports System.IO
Imports System.Net.Sockets
Imports System.Text

Module EtherDriftMeasure
    'Sample Program for Performing Drift Analysis via Ethernet
    Sub Main()
        Try
            Dim wlmAddr As String
            Dim wlmPort As Integer
            Dim sockStream As NetworkStream
            Dim tcpObj As TcpClient
            Dim replyString As String
            Dim peakNum As Integer
            Dim refPowData, refWavData As Double()
            Dim maxPowData, maxWavData As Double()
            Dim minPowData, minWavData As Double()
            Dim dropInfo As Double()
            Dim username, passwd As String

            '=====
            'Wavelength meter information
            '=====
            wlmAddr = "192.168.0.1"           'Wavelength meter IP address
            wlmPort = 10001                 'Remote port number
            username = "anonymous"          'User name
            passwd = ""                     'Password

            '=====
            ' TCP connection
            '=====
            tcpObj = New TcpClient
            tcpObj.Connect(wlmAddr, wlmPort)
            sockStream = tcpObj.GetStream()
            tcpObj.NoDelay = True           'Enable TCP_NODELAY

            '=====
            ' Execute authentication
            '=====
            Dim recvBuffer As String
            TcpWriteLine("open """" + username + """"", sockStream) 'Send an OPEN command with the
                                                                    'user name

            recvBuffer = TcpReadLine(sockStream)
            If String.Compare(recvBuffer, "AUTHENTICATE CRAM-MD5") <> 0 Then
                sockStream.Dispose()
                Exit Sub                                     'Error if the response is not
                                                                    'AUTHENTICATE CRAM-MD5

            End If
            TcpWriteLine(passwd, sockStream)                 'Send the password
            recvBuffer = TcpReadLine(sockStream)

            If String.Compare(recvBuffer, "ready") <> 0 Then
                sockStream.Dispose()
                Exit Sub                                     'Authentication failure
            End If

            '=====
            ' Set the wavelength meter's measurement conditions
            '=====
            Call TcpWriteLine("*RST", sockStream)             'Reset the instrument
            Call TcpWriteLine(":CALC2:PTHR:MODE REL", sockStream) 'Set the threshold to relative
                                                                    'mode
            Call TcpWriteLine(":CALC2:PTHR 15", sockStream)   'Set the threshold to 15 db
            Call TcpWriteLine(":UNIT:WL NM", sockStream)       'Set the wavelength unit to nm
            Call TcpWriteLine(":UNIT:POW DBM", sockStream)     'Set the power unit to dBm
        
```

### 3.4 Sample Program (for SOCKET)

---

```
'Perform a Single measurement to obtain the reference for the 'drift measurement
Call TcpWriteLine(":INIT;*OPC?", sockStream)           'Execute a Single measurement
                                                         'and wait for the measurement to
                                                         'complete
TcpReadLine(sockStream)                               'Read the measurement complete
                                                         'wait (*OPC?) response
Call TcpWriteLine(":CALC3:DRIF ON", sockStream)         'Turn drift analysis on

'=====
' Execute a measurement
'=====
Call TcpWriteLine(":INIT:CONT ON", sockStream)         'Start a Repeat measurement

For count As Integer = 1 To 60                         'Wait 1 minute
    Threading.Thread.Sleep(1000)
    Console.WriteLine(".")
Next
Console.WriteLine("")
Call TcpWriteLine(":INIT:CONT OFF", sockStream)         'Stop the Repeat measurement

'=====
'Retrieve the measured results
'=====
Call TcpWriteLine(":CALC3:POIN?", sockStream)           'Retrieve the number of data
                                                         'values

replyString = TcpReadLine(sockStream)
peakNum = Integer.Parse(replyString)
refPowData = New Double(peakNum - 1) {}
refWavData = New Double(peakNum - 1) {}
maxPowData = New Double(peakNum - 1) {}
maxWavData = New Double(peakNum - 1) {}
minPowData = New Double(peakNum - 1) {}
minWavData = New Double(peakNum - 1) {}
dropInfo = New Double(peakNum - 1) {}

'Retrieve the results (reference values)
Call TcpWriteLine(":CALC3:DRIF:REF ON", sockStream)
Call TcpWriteLine(":CALC3:DATA? POW", sockStream)       'Retrieve Ref Power
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, refPowData)
Call TcpWriteLine(":CALC3:DATA? WAV", sockStream)       'Retrieve Ref Wavelength
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, refWavData)
'Retrieve the results (max values)
Call TcpWriteLine(":CALC3:DRIF:PRES", sockStream)
Call TcpWriteLine(":CALC3:DRIF:MAX ON", sockStream)
Call TcpWriteLine(":CALC3:DATA? POW", sockStream)       'Retrieve MAX Power
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, maxPowData)
Call TcpWriteLine(":CALC3:DATA? WAV", sockStream)       'Retrieve MAX Wavelength
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, maxWavData)
'Retrieve the results (min values)
Call TcpWriteLine(":CALC3:DRIF:PRES", sockStream)
Call TcpWriteLine(":CALC3:DRIF:MIN ON", sockStream)
Call TcpWriteLine(":CALC3:DATA? POW", sockStream)       'Retrieve MIN Power
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, minPowData)
Call TcpWriteLine(":CALC3:DATA? WAV", sockStream)       'Retrieve MIN Wavelength
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, minWavData)
'Retrieve drop information
Call TcpWriteLine(":CALC3:DATA? DROP", sockStream)
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, dropInfo)

sockStream.Dispose()                                   'Close TCP
```

```

'=====
'Display the measured results
'=====
Console.Write("No.          |")                                'Display the peak number
For idx As Integer = 0 To peakNum - 1
    Console.Write((idx + 1).ToString() + " |")
Next
Console.WriteLine()
Console.Write("REF WL          |")                                'Display the reference wavelength
For idx As Integer = 0 To peakNum - 1
    Console.Write(refWavData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("REF POWER      |")                                'Display the reference power
For idx As Integer = 0 To peakNum - 1
    Console.Write(refPowData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("MAX WL          |")                                'Display the maximum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MAX POWER      |")                                'Display the maximum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN WL          |")                                'Display the minimum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN POWER      |")                                'Display the minimum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.ReadLine()
Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.ReadLine()

End Try
End Sub

'=====
'Function for sending character strings to the TCP Socket
'=====
Sub TcpWriteLine(ByVal commandStr As String, ByRef stream As NetworkStream)
    Dim writer As StreamWriter = New StreamWriter(stream, Encoding.ASCII)

```

### 3.4 Sample Program (for SOCKET)

---

```
Dim ByteLf As Byte() = New Byte() {10}
writer.NewLine = Encoding.ASCII.GetString(ByteLf)      'The line feed code is LF.
writer.AutoFlush = True
writer.WriteLine(commandStr) 'Send data
End Sub

'=====
'Function for reading a line of data from the TCP Socket
'=====
Function TcpReadLine(ByRef stream As NetworkStream) As String
    Dim reader As StreamReader = New StreamReader(stream, Encoding.ASCII)
    TcpReadLine = reader.ReadLine()                    'Receive data
Exit Function
End Function

'=====
'Function for splitting the CALC3 results into an array
'=====
Sub SplitArrayData(ByVal dataString As String, ByRef dataArray As Double())
    Dim peakNum As Integer
    Dim arrayDataStr As String() = dataString.Split(",") 'Split the string by using a
                                                         'comma as the delimiter

    peakNum = arrayDataStr.Length
    For idx As Integer = 0 To arrayDataStr.Length - 1
        dataArray(idx) = Convert.ToDouble(arrayDataStr(idx)) 'Convert the split strings into
                                                             'values
    Next
End Sub
End Module
```

#### Execution Example

No.	1	2	3	4	5	
REF WL	1.30678832E-06	1.30756981E-06	1.30835238E-06	1.30913541E-06	1.30991969E-06	
REF POWER	-13.4899875	-9.04694537	-2.9512995	-3.29214313	-13.1556519	
MAX WL	-----	1.30757036E-06	1.3083528E-06	1.30913604E-06	-----	
MAX POWER	-----	-8.81158076	-0.665845116	-3.21870974	-----	
MIN WL	-----	1.30756953E-06	1.30835221E-06	1.30913538E-06	-----	
MIN POWER	-----	-10.2276251	-3.02598662	-6.67785905	-----	

## Sample Program 4

Sample program for performing drift analysis via Ethernet. Example of the -SW (single-wavelength) suffix code model.

### Source Code

```
Imports System
Imports System.IO
Imports System.Net.Sockets
Imports System.Text

Module EtherDriftMeasure
'Sample Program for Performing Drift Analysis via Ethernet
Sub Main()
    Try
        Dim wlmAddr As String
        Dim wlmPort As Integer
        Dim sockStream As NetworkStream
        Dim tcpObj As TcpClient
        Dim replyString As String
        Dim peakNum As Integer
        Dim refPowData, refWavData As Double()
        Dim maxPowData, maxWavData As Double()
        Dim minPowData, minWavData As Double()
        Dim dropInfo As Double()
        Dim username, passwd As String

        '=====
        'Wavelength meter information
        '=====
        wlmAddr = "192.168.0.1"
        wlmPort = 10001
        username = "anonymous"
        passwd = ""

        '=====
        ' TCP connection
        '=====
        tcpObj = New TcpClient
        tcpObj.Connect(wlmAddr, wlmPort)
        sockStream = tcpObj.GetStream()
        tcpObj.NoDelay = True

        '=====
        ' Execute authentication
        '=====
        Dim recvBuffer As String
        TcpWriteLine("open "" + username + """, sockStream)
        recvBuffer = TcpReadLine(sockStream)
        If String.Compare(recvBuffer, "AUTHENTICATE CRAM-MD5") <> 0 Then
            sockStream.Dispose()
            Exit Sub

        End If
        TcpWriteLine(passwd, sockStream)
        recvBuffer = TcpReadLine(sockStream)
        If String.Compare(recvBuffer, "ready") <> 0 Then
            sockStream.Dispose()
            Exit Sub

        End If

        'Wavelength meter IP address
        'Remote port number
        'User name
        'Password

        'Enable TCP_NODELAY

        'Send an OPEN command with the
        'user name

        'Error if the response is not
        'AUTHENTICATE CRAM-MD5

        'Send the password

        'Authentication failure
```



### 3.4 Sample Program (for SOCKET)

---

```
'=====
' Set the wavelength meter's measurement conditions
'=====
Call TcpWriteLine("*RST", sockStream)           'Reset the instrument
Call TcpWriteLine(":UNIT:WL NM", sockStream)     'Set the wavelength unit to nm
Call TcpWriteLine(":UNIT:POW DBM", sockStream)    'Set the power unit to dBm
'Perform a Single measurement to obtain the reference for the 'drift measurement
Call TcpWriteLine(":INIT;*OPC?", sockStream)      'Execute a Single measurement
                                                    'and wait for the measurement to
                                                    'complete

TcpReadLine(sockStream)                         'Read the measurement complete
                                                    'wait (*OPC?) response

Call TcpWriteLine(":CALC3:DRIF ON", sockStream)   'Turn drift analysis on

'=====
' Execute a measurement
'=====
Call TcpWriteLine(":INIT:CONT ON", sockStream)    'Start a Repeat measurement
For count As Integer = 1 To 60                  'Wait 1 minute
    Threading.Thread.Sleep(1000)
    Console.WriteLine(".")
Next
Console.WriteLine("")
Call TcpWriteLine(":INIT:CONT OFF", sockStream)   'Stop the Repeat measurement

'=====
'Retrieve the measured results
'=====
Call TcpWriteLine(":CALC3:POIN?", sockStream)      'Retrieve the number of data
                                                    'values

replyString = TcpReadLine(sockStream)
peakNum = Integer.Parse(replyString)
refPowData = New Double(peakNum - 1) {}
refWavData = New Double(peakNum - 1) {}
maxPowData = New Double(peakNum - 1) {}
maxWavData = New Double(peakNum - 1) {}
minPowData = New Double(peakNum - 1) {}
minWavData = New Double(peakNum - 1) {}
dropInfo = New Double(peakNum - 1) {}

'Retrieve the results (reference values)
Call TcpWriteLine(":CALC3:DRIF:REF ON", sockStream)
Call TcpWriteLine(":CALC3:DATA? POW", sockStream) 'Retrieve Ref Power
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, refPowData)
Call TcpWriteLine(":CALC3:DATA? WAV", sockStream) 'Retrieve Ref Wavelength
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, refWavData)

'Retrieve the results (max values)
Call TcpWriteLine(":CALC3:DRIF:PRES", sockStream)
Call TcpWriteLine(":CALC3:DRIF:MAX ON", sockStream)
Call TcpWriteLine(":CALC3:DATA? POW", sockStream) 'Retrieve MAX Power
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, maxPowData)
Call TcpWriteLine(":CALC3:DATA? WAV", sockStream) 'Retrieve MAX Wavelength
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, maxWavData)

'Retrieve the results (min values)
Call TcpWriteLine(":CALC3:DRIF:PRES", sockStream)
Call TcpWriteLine(":CALC3:DRIF:MIN ON", sockStream)
Call TcpWriteLine(":CALC3:DATA? POW", sockStream) 'Retrieve MIN Power
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, minPowData)
Call TcpWriteLine(":CALC3:DATA? WAV", sockStream) 'Retrieve MIN Wavelength
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, minWavData)

'Retrieve drop information
Call TcpWriteLine(":CALC3:DATA? DROP", sockStream)
replyString = TcpReadLine(sockStream)
SplitArrayData(replyString, dropInfo)
sockStream.Dispose()                               'Close TCP
```

```

'=====
'Display the measured results
'=====
Console.Write("No.          |")                                'Display the peak number
For idx As Integer = 0 To peakNum - 1
    Console.Write((idx + 1).ToString() + " |")
Next
Console.WriteLine()
Console.Write("REF WL          |")                                'Display the reference wavelength
For idx As Integer = 0 To peakNum - 1
    Console.Write(refWavData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("REF POWER      |")                                'Display the reference power
For idx As Integer = 0 To peakNum - 1
    Console.Write(refPowData(idx).ToString() + " | ")
Next
Console.WriteLine()
Console.Write("MAX WL          |")                                'Display the maximum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MAX POWER      |")                                'Display the maximum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(maxPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN WL          |")                                'Display the minimum wavelength
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minWavData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.Write("MIN POWER      |")                                'Display the minimum power
For idx As Integer = 0 To peakNum - 1
    If dropInfo(idx) <> 0 Then
        Console.Write("----- | ")
    Else
        Console.Write(minPowData(idx).ToString() + " | ")
    End If
Next
Console.WriteLine()
Console.ReadLine()
Catch ex As Exception
    Console.WriteLine(ex.Message)
    Console.ReadLine()

    'Error handling
    'Display the error message that
    'occurred
    'Wait for the user to press the
    'Enter key

End Try
End Sub

```

### 3.4 Sample Program (for SOCKET)

---

```
'=====
'Function for sending character strings to the TCP Socket
'=====
Sub TcpWriteLine(ByVal commandStr As String, ByRef stream As NetworkStream)
    Dim writer As StreamWriter = New StreamWriter(stream, Encoding.ASCII)
    Dim ByteLf As Byte() = New Byte() {10}
    writer.NewLine = Encoding.ASCII.GetString(ByteLf)          'The line feed code is LF.
    writer.AutoFlush = True
    writer.WriteLine(commandStr)                                'Send data
End Sub

'=====
'Function for reading a line of data from the TCP Socket
'=====
Function TcpReadLine(ByRef stream As NetworkStream) As String
    Dim reader As StreamReader = New StreamReader(stream, Encoding.ASCII)
    TcpReadLine = reader.ReadLine()                             'Receive data
Exit Function
End Function

'=====
'Function for splitting the CALC3 results into an array
'=====
Sub SplitArrayData(ByVal dataString As String, ByRef dataArray As Double())
    Dim peakNum As Integer
    Dim arrayDataStr As String() = dataString.Split(",")        'Split the string by using a
                                                                'comma as the delimiter

    peakNum = arrayDataStr.Length
    For idx As Integer = 0 To arrayDataStr.Length - 1
        dataArray(idx) = Convert.ToDouble(arrayDataStr(idx))    'Convert the split strings into
                                                                'values
    Next
End Sub
End Module
```

#### Execution Example

No.	1	
REF WL	1.55252293E-06	
REF POWER	3.37379314	
MAX WL	1.55252431E-06	
MAX POWER	3.43271086	
MIN WL	1.55252289E-06	
MIN POWER	3.3485737	

## 4.1 Status Registers

The instrument has the status registers shown in the table below.

A complete status register diagram is provided on the next page.

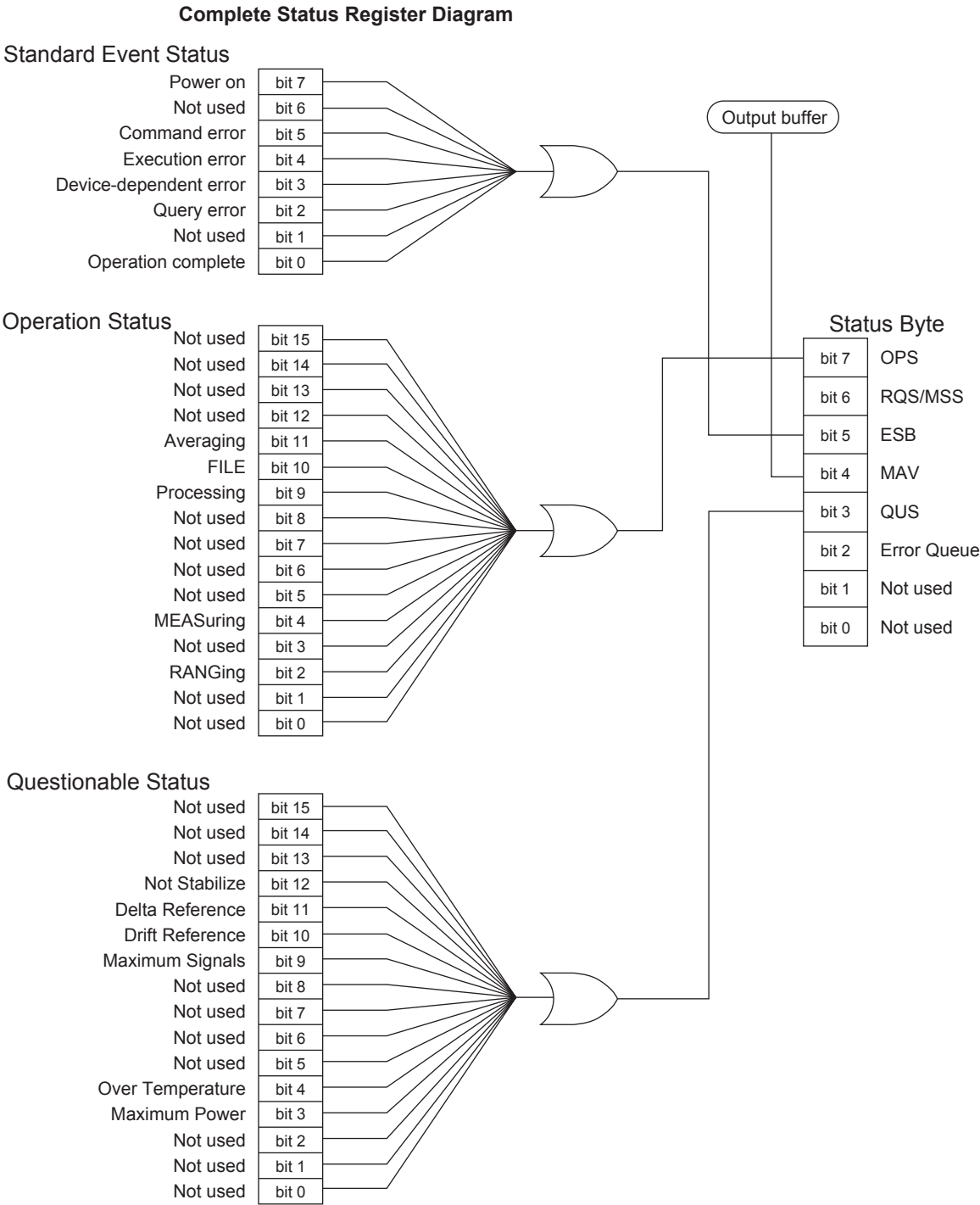
The instrument has the following registers, which are defined in IEEE488.2 and SCPI.

- Status Byte Register
- Standard Event Status Register
- Operation Status Register
- Questionable Status Register

As a summary of each register, an operation status bit (OPS) and questionable status bit (QUS) are assigned to the expansion bits of the Status Byte Register.

### Status Registers

Register Name	Description
Status Byte Register	Register defined in IEEE488.2
STB: Status Byte Register	Same as above
SRE: Service Request Enable Register	Same as above
Standard Event Status Register	Register defined in IEEE488.2
ESR: Standard Event Status Register	Same as above
ESE: Standard Event Status Register	Same as above
Operation Status Register	Provides execution information about operations (measuring, averaging, etc.)
Operation Event Register	Indicates the presence or absence of events. Events are latched.
Operation Event Enable Register	Conditional masking register for generating summary bit OPS
Questionable Status Register	Provides information about the operation status of the instrument
Questionable Event Register	Indicates the presence or absence of events. Events are latched.
Questionable Event Enable Register	Conditional masking register for generating summary bit QUS



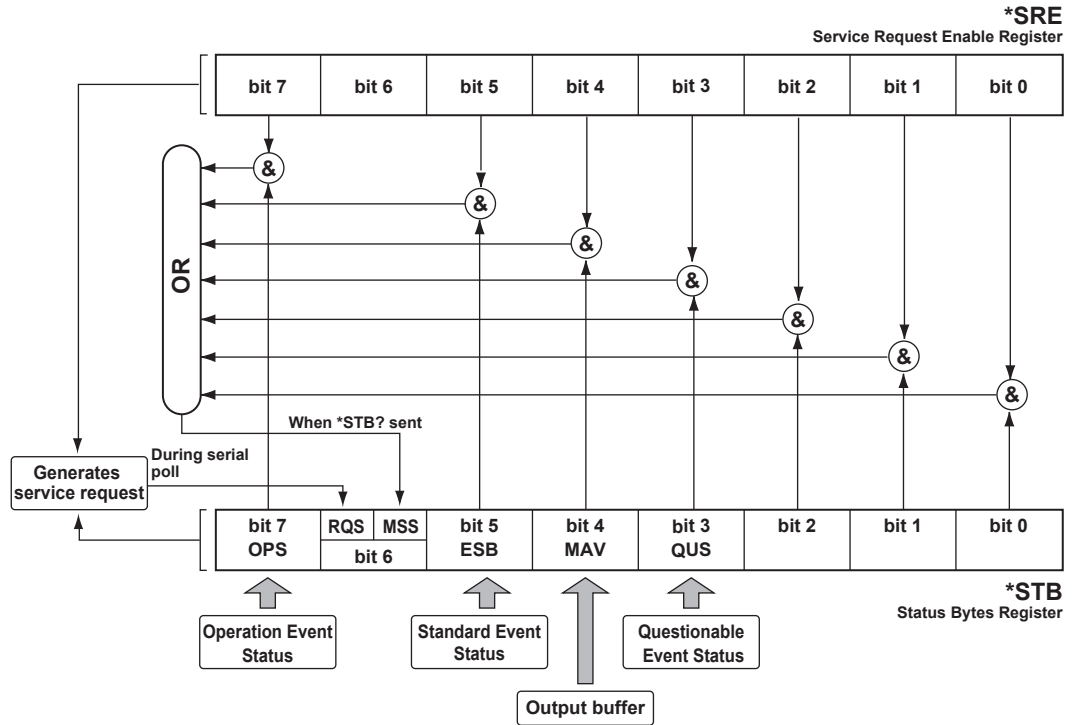
## 4.2 Status Byte Register

### Configuration

The following figure shows the configuration of the status byte register.

The register details and operation comply with IEEE488.2.

The OPS and QUS bits are expansions used by the instrument.



### Status Byte Register Description

Bit	Event	Description	Decimal value
Bit 7	OPS	Operation status summary bit	128
Bit 6	RQS, MSS	Set to 1 when there are one or more service requests	64
Bit 5	ESB	Standard Event Status Register summary bit	32
Bit 4	MAV	Set to 1 when data exists in the output buffer	16
Bit 3	QUS	Questionable status summary bit	8
Bit 2	Error Queue	Set to 1 when there is an error	4
Bit 1	None	Not used (always 0)	0
Bit 0	None	Not used (always 0)	0

### Status Byte Register

#### Reading

You can read the Status Byte Register through serial polling or the \*STB? common query. The bit 6 information varies

depending on which method you use to read the register.

- Serial polling  
The RQS message is read as bit 6 information.  
After the register is read, the RQS message is cleared.
- \*STB? common query  
MSS summary message is read as bit 6 information.  
After the register is read, the MSS message does not change.

All other bits are the same.

The read operation complies with IEEE488.2.

#### Writing

The register is written only when the status of the assigned status data structure changes.

The write operation complies with IEEE488.2.

#### Clearing

The \*CLS common command clears all event registers and queues except for the output queue and MAV bit.

The clear operation complies with IEEE488.2.

### Service Request Enable Register

#### Reading

You can use the \*SRE? common query to read the register.

The register is not cleared when it is read.

The read operation complies with IEEE488.2.

#### Writing

You can use the \*SRE common command to write to the register.

The value for the unused bit, bit 6, is always ignored.

The write operation complies with IEEE488.2.

#### Clearing

The register is cleared in the following circumstances.

- When data 0 is set with the \*SRE common command
- When the power is turned on

The register is not cleared in the following circumstances.

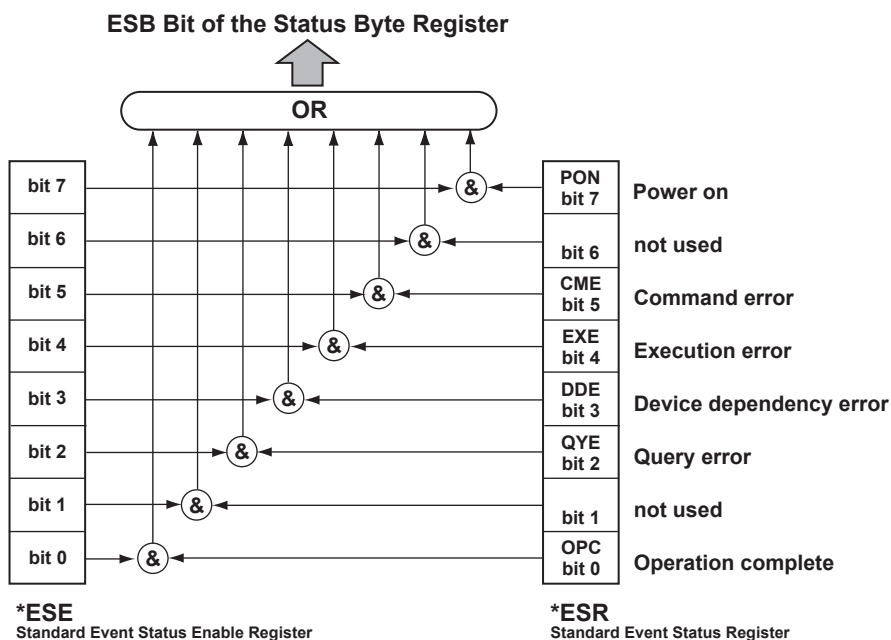
- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

The clear operation complies with IEEE488.2.

## 4.3 Standard Event Status Register

### Configuration

The following figure shows the configuration of the Standard Event Status Register. The register details and operation comply with IEEE488.2.



#### Standard Event Status Register Description

Bit	Event	Description	Decimal value	Note
Bit 7	PON (power ON)	Set to 1 when the power changes from off to on	128	
Bit 6	None	Not used (always 0)	0	
Bit 5	CME (command error)	Set to 1 when a syntax error is detected or when an unknown command is detected. Set to 1 when a GET command is detected between the first byte of the program message and the program message terminator.	32	
Bit 4	EXE (execution error)	Set to 1 when the program data following the program header is outside the valid range. Set to 1 when a program message that conflicts with the current device state is received.	16	Set to 1 when message number 200 occurs.
Bit 3	DDE	Set to 1 when an error other than CME, EXE, or QYE (a device-specific error) occurs	8	Set to 1 when a message number in the 70s (measurement processing errors) or 80s (hardware operation error) occurs.
Bit 2	QYE (query error)	Set to 1 when the response to a query command is not in the output queue.	4	Set to 1 when message number 410 or 440 occurs.
Bit 1	None	Not used (always 0)	0	
Bit 0	OPC (operation complete)	Command operation complete. Valid only for *OPC. Not valid for *OPC?.	1	For details on the timing of command operation completion, see page 4-8.

#### Note

For details on messages, see section 4.1 in the User's Manual, IM AQ6150B-02EN.



### Standard Event Status Register

#### Reading

You can use the \*ESR? common query to read the register.  
The contents of the register are cleared when it is read.  
The read operation complies with IEEE488.2.

#### Writing

You can clear the contents of the register. Only clearing is allowed. Writing is not allowed.

#### Clearing

The register is cleared in the following circumstances.

- When a \*CLS common command is received
- When a \*ESR? common query is received

The clear operation complies with IEEE488.2.

### Standard Event Status Enable Register

#### Reading

You can use the \*ESE? common query to read the register.  
The read operation complies with IEEE488.2.

#### Writing

You can use the \*ESE common command to write to the register.  
The write operation complies with IEEE488.2.

#### Clearing

The register is cleared in the following circumstances.

- When data 0 is set with the \*ESE common command
- When the power is turned on

The register is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

The clear operation complies with IEEE488.2.

## 4.4 Operation Status Register

The Operation Status Register indicates the operation status of the instrument.

The instrument status is indicated by the Operation Condition Register.

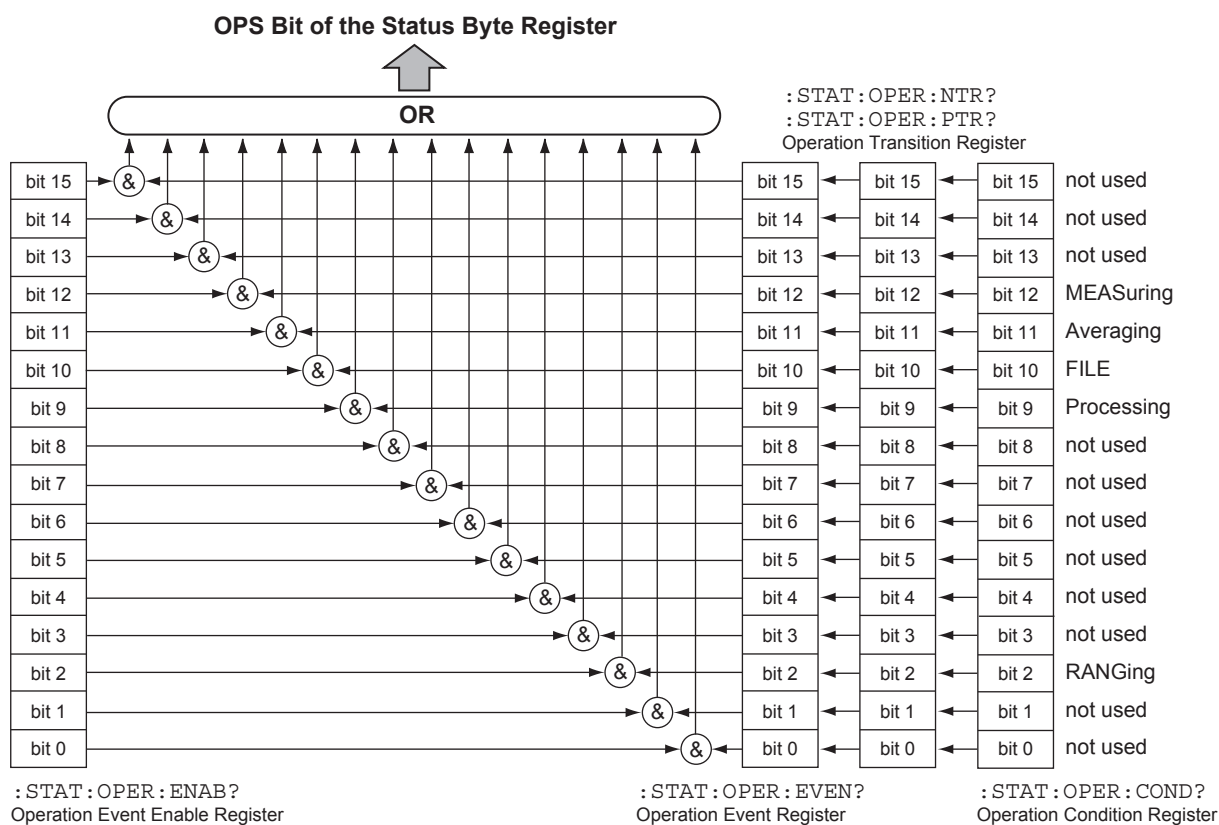
The changes in the Operation Condition Register are reflected in the Operation Event Register. You can identify changes in the operation status by referring to the Operation Status Register.

The summary of the Operation Event Register is set in the OPS bit of the Status Byte Register. This bit includes only the statuses that correspond to the bits that are set to 1 in the Operation Event Enable Register.

### Configuration

The following figure shows the configuration of the Operation Status Register.

#### Operation Status Register Structure



## 4.4 Operation Status Register

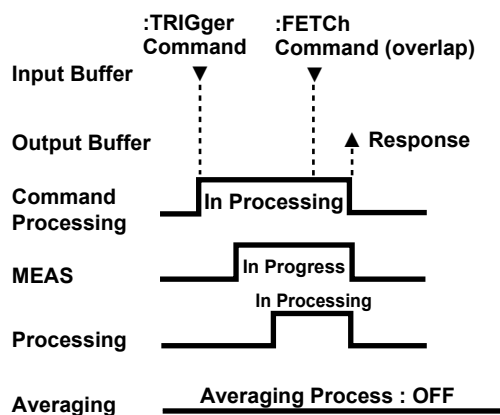
### Operation Status Register Description

Bit	Event	Description	Decimal value	Note
Bit 15	Not used	Reserved (always 0)	0	
Bit 14	Not used	Reserved (always 0)	0	
Bit 13	Not used	Reserved (always 0)	0	
Bit 12	Not used	Reserved (always 0)	0	
Bit 11	Averaging	Averaging	2048	Set to 1 when the average count is set to a value larger than 1
Bit 10	FILE	File being accessed	1024	Set to 1 when the file is being accessed (read, write, copy, delete, or rename).
Bit 9	Processing	Computing	512	Set to 1 when computation is in progress
Bit 8	Not used	Reserved (always 0)	0	
Bit 7	Not used	Reserved (always 0)	0	
Bit 6	Not used	Reserved (always 0)	0	
Bit 5	Not used	Reserved (always 0)	0	
Bit 4	MEASuring	Measuring	16	Set to 1 when measurement is in progress
Bit 3	Not used	Reserved (always 0)	0	
Bit 2	RANGing	Range switching	4	Set to 1 when "under range" or "over range" occurs
Bit 1	Not used	Reserved (always 0)	0	
Bit 0	Not used	Reserved (always 0)	0	

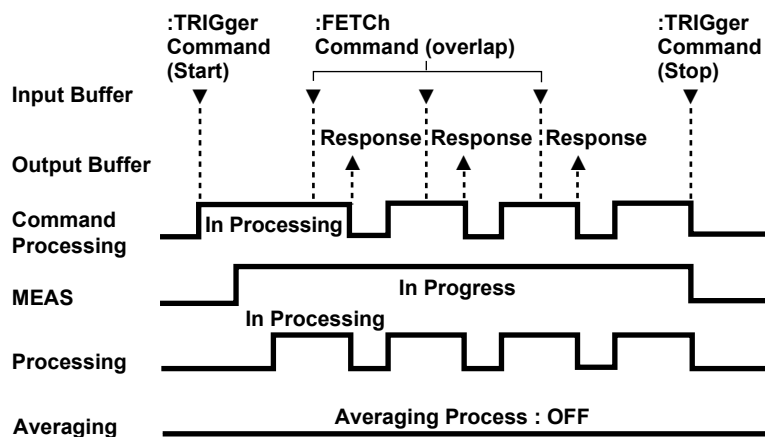
### Example of Operation Status Change

The figure below shows how each status bit changes when the instrument receives a measurement start command, and the timing for the instrument to return a response when it receives a query for the measurement results when it is measuring.

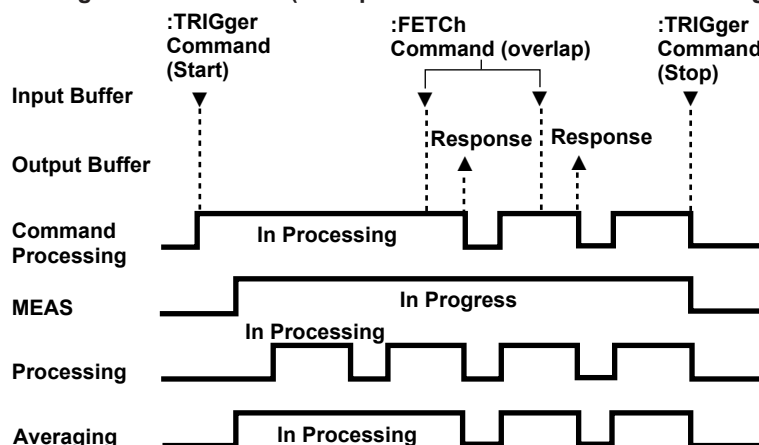
#### • Single measurement



#### • Repeat measurement



- Averaged measurement (for repeat measurement when the average count is 2)



When the instrument receives a command in its input buffer, it enters a command processing state. The command processing state continues until all processing (measurement, computation, and averaging) is complete. In this state, only overlap commands can be executed.

In the example in the figure, the command that queries the measured results is processed as an overlap command. The instrument returns a response (measurement results) when the processing of the command is finished.

In repeat measurement and averaged measurement, MEAS remains at 1 while measurement is in progress.

Processing is set to 1 only when computation is being performed.

In the first averaging process, Averaging is set to 1 while averaging is performed for the average count. In the subsequent averaging processes, the instrument averages the new measured value with the already measured value, so Averaging will be synchronous to Processing.

The \*OPC and \*OPC? commands query the command processing status.

## Operation Condition Register

### Reading

You can use the :STATus:OPERation:CONDition? query command to read the register. The contents of the register are not cleared when it is read.

### Writing

The bits in this register are set or reset only when the corresponding statuses of the instrument change.

Writing is not allowed.

### Clearing

Clearing is not allowed.

### Operation Event Register

#### Reading

You can use the :STATus:OPERation[:EVENT]? query command to read the register.  
The contents of the register are cleared when it is read.

#### Writing

You can clear the contents of the register. Only clearing is allowed. Writing is not allowed.  
Clearing

The register is cleared in the following circumstances.

- When the register is read with the :STATus:OPERation[:EVENT]? query command
- When the instrument is initialized with the :STATus:PRESet command
- When a \*CLS common command is received
- When the power is turned on

### Operation Event Enable Register

#### Reading

You can use the :STATus:OPERation:ENABLE? query command to read the register.

#### Writing

You can use the :STATus:OPERation:ENABLE command to write to the register.

#### Clearing

The register is cleared in the following circumstances.

- When data 0 is set with the :STATus:OPERation:ENABLE command
- When the power is turned on

The register is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

### Operation Positive Transition Filter

#### Reading

You can use the :STATus:OPERation:PTRansition? query command to read the filter.

#### Writing

You can use the :STATus:OPERation:PTRansition command to write to the filter.

#### Clearing

The filter is cleared in the following circumstances.

- When data 0 is set with the :STATus:OPERation:PTRansition command
- When the power is turned on

The filter is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

## Operation Negative Transition Filter

### Reading

You can use the :STATus:OPERation:NTRansition? query command to read the filter.

### Writing

You can use the :STATus:OPERation:NTRansition command to write to the filter.

### Clearing

The filter is cleared in the following circumstances.

- When data 0 is set with the :STATus:OPERation:NTRansition command
- When the power is turned on

The filter is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

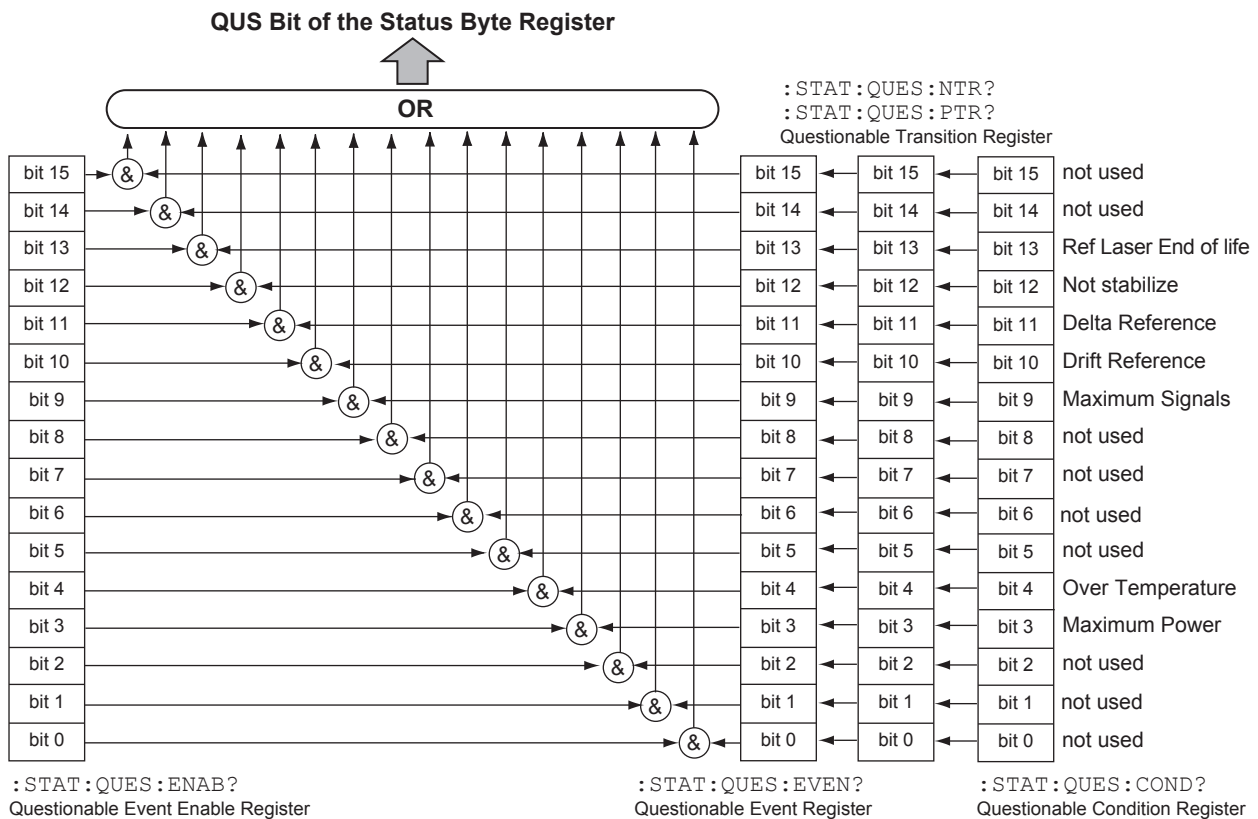
## 4.5 Questionable Status Register

The Questionable Status Register indicates the questionable status of the instrument. The summary of the Questionable Event Register is set in the QUS bit of the Status Byte Register.

### Configuration

The following figure shows the configuration of the Questionable Status Register.

Questionable Status Register Configuration



**Questionable Status Register Description**

Bit	Event	Description	Decimal value	Note
Bit 15	Not used	Reserved (always 0)	0	
Bit 14	Not used	Reserved (always 0)	0	
Bit 13	Ref Laser End of Life	The end of reference light source service life is reached	8192	Set to 1 when the end of the reference light source service life is detected
Bit 12	Not stabilize	Reference light source status	0	Set to 1 when the reference light source is not stable.
Bit 11	Delta Reference	The reference is lost	2048	Set to 1 during delta measurement when the reference peak is lost. This is always zero on the -SW (single-wavelength) suffix code type.
Bit 10	Drift Reference	The number of peaks is different	1024	Set to 1 during drift measurement when the number of reference peaks differs from the number of measured peaks
Bit 9	Maximum Signals	The maximum number of detected peaks (1024) is exceeded	512	Set to 1 when the maximum number of detected peaks is exceeded. This is always zero on the -SW (single-wavelength) suffix code type.
Bit 8	Not used	Reserved (always 0)	0	
Bit 7	Not used	Reserved (always 0)	0	
Bit 6	Not used	Reserved (always 0)	0	
Bit 5	Not used	Reserved (always 0)	0	
Bit 4	Over Temperature	An overheating error occurs	16	Set to 1 when the internal temperature of the instrument rises to an abnormal level
Bit 3	Maximum Power	The maximum input power is exceeded	8	Set to 1 when the optical input power exceeds the allowed power
Bit 2	Not used	Reserved (always 0)	0	
Bit 1	Not used	Reserved (always 0)	0	
Bit 0	Not used	Reserved (always 0)	0	

**Questionable Condition Register****Reading**

You can use the :STATus:QUESTionable:CONDition? query command to read the register. The contents of the register are not cleared when it is read.

**Writing**

The bits in this register are set or reset only when the corresponding statuses of the instrument change.

Writing is not allowed.

**Clearing**

Clearing is not allowed.

**Questionable Event Register****Reading**

You can use the :STATus:QUESTionable[:EVENT]? query command to read the register. The contents of the register are cleared when it is read.

**Writing**

You can clear the contents of the register. Only clearing is allowed. Writing is not allowed.

**Clearing**

The register is cleared in the following circumstances.

- When the register is read with the :STATus:QUESTionable[:EVENT]? query command
- When the instrument is initialized with the :STATus:PRESet command
- When a \*CLS common command is received
- When the power is turned on



### Questionable Event Enable Register

#### Reading

You can use the :STATus:QUESTionable:ENABle? query command to read the register.

#### Writing

You can use the :STATus:QUESTionable:ENABle command to write to the register.

#### Clearing

The register is cleared in the following circumstances.

- When data 0 is set with the :STATus:QUESTionable:ENABle command
- When the power is turned on

The register is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

### Questionable Positive Transition Filter

#### Reading

You can use the :STATus:QUESTionable:PTRansition? query command to read the register.

#### Writing

You can use the :STATus:QUESTionable:PTRansition command to write to the filter.

#### Clearing

The filter is cleared in the following circumstances.

- When data 0 is set with the :STATus:QUESTionable:PTRansition command
- When the power is turned on

The filter is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

### Questionable Negative Transition Filter

#### Reading

You can use the :STATus:QUESTionable:NTRansition? query command to read the register.

#### Writing

You can use the :STATus:QUESTionable:NTRansition command to write to the filter.

#### Clearing

The filter is cleared in the following circumstances.

- When data 0 is set with the :STATus:QUESTionable:NTRansition command
- When the power is turned on

The filter is not cleared in the following circumstances.

- When a \*RST command is received
- When a \*CLS command is received
- When device clear (DCL or SDC) is specified

## 5.1 Command Syntax and Types

The information covered in this section applies to the common commands and device-specific commands in this manual.

Excluding special commands, all measured values and parameters are exchanged using ASCII character strings.

### Syntax

Convention	Description
	Indicates that an element in the list is to be selected. Example: A B C Select A, B, or C.
[]	Items in brackets are optional.
{ }	Items in braces can be specified multiple times in a command.
<wsp> <sup>1</sup>	White space
<integer>	Integer
<NRf>	Value in exponential, integer, or floating-point notation
<"file name">	The length of a file name excluding its path but including its extension is up to 56 characters. Enclose the character string in double quotation marks (").
<"string">	Character string Enclose the character string in double quotation marks (").

- 1 Regarding white spaces (<wsp>)  
Characters that correspond to ASCII 00h to 20h (excluding 0Ah(LF)) are defined as white spaces.  
With the exception of the white space that is necessary between a command and its parameters and white spaces within a character string, such as a file name parameter, you can enter (or omit) white spaces as you like in order to make your program easier to read.

### Command Type

The instrument has the following three types of commands.

Overlappable commands and overlap commands are indicated in the explanations in sections 5.4 and 5.5.

#### Sequential Commands

- The execution of other commands does not start until a sequential command is finished.
- The execution of a sequential command does not start until other commands are finished.

#### Overlappable Commands

- The execution of other overlap commands starts before an overlappable command is finished.
- The execution of sequential commands does not start until an overlappable command is finished.
- The execution of an overlappable command does not start until other commands are finished.

#### Overlap Commands

- The execution of overlap commands can start before an overlappable command is finished.
- The execution of other commands does not start until an overlap command is finished.
- The execution of an overlap command does not start until a sequential command is finished.

### Sending Multiple Commands at Once

You can create command strings using the commands described in section 5.4, “Common Commands” and section 5.5, “Device-Specific Commands” and send them to the instrument.

You can concatenate multiple commands in a single output statement by separating each command with a semicolon. These commands will be executed in the order they are written.

### Remote Command Syntax

#### Short Form and Long Form

The instrument’s remote commands can be written in the short form or long form.

In the command description in this manual, the section of the command written in uppercase corresponds to the short form of the command.

For example, the short form of the INITiate command is INIT; the long form is INITIATE.

#### Uppercase and Lowercase

The instrument commands are not case sensitive.

Read values are written in all uppercase.

#### Values

- This instrument can receive values that are written in a variety of formats.
- Values that the instrument sends are in the basic form.

The mantissa is fixed to one integer digit (with sign) and eight fractional digits.

The exponent is fixed to three digits.

Example:     Acceptable values for 1550 nm  
                  1550nm, 1.55um, 1550E-9, 1.55E-6, etc.

Example:     Transmitted value for 1550 nm  
                  +1.55000000E-006 only

- If the received value is higher in resolution than that used in the instrument, the value is rounded (not truncated).
- The instrument supports 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)

#### Parameter Specification in Commands

To include a parameter in a command, you must insert a space between the command and parameter.

Parameters are separated by commas.

To make a command easier to read, you can insert spaces before or after each comma.

## 5.2 Table of Soft Key to Remote Command Assignments

The following tables list the remote commands that correspond to the soft keys for operating the instrument.

For details on the parameters of each command, see section 5.3 or 5.5.

For detailed descriptions of parameters, see section 5.5.

### SETUP

Soft Key	Remote Command	Note
DEVICE TYPE	[ :SENSe]:CORRection:DEVIce	
PEAK THRESH TYPE <sup>1</sup>	:CALCulate2:PTHReshold:MODE	
PEAK THRESH VALUE <sup>1</sup>	:CALCulate2:PTHReshold[:RELative]	Relative value
	:CALCulate2:PTHReshold:ABSolute	Absolute value
PEAK EXCURSION	:CALCulate2:PEXCursion	
WAVELENGTH LIMIT		
LIMITING MODE	:CALCulate2:WLIMit[:STATe]	
LIMIT START WL	:CALCulate2:WLIMit:START:FREQuency	Frequency
	:CALCulate2:WLIMit:START[:WAVelength]	Wavelength
	:CALCulate2:WLIMit:START:WNUMber	Wavenumber
LIMIT STOP WL	:CALCulate2:WLIMit:STOP:FREQuency	Frequency
	:CALCulate2:WLIMit:STOP[:WAVelength]	Wavelength
	:CALCulate2:WLIMit:STOP:WNUMber	Wavenumber
SET PRESET LIMITS	—	
AVERAGE TIMES	:CALCulate2:COUNT	
WAVELENGTH UNIT	:UNIT:WL	
POWER UNIT	:UNIT[:POWER]	
MEAS WL	[ :SENSe]:CORRection:MEDIum	
CH MATCHING THRESH FREQ <sup>1</sup>	:CALCulate2:MTHResh	
UPDATE RATE	[ :SENSe]:URATe	

1 For the -MW (multi-wavelength) suffix code type

### SYSTEM

Soft Key	Remote Command	Note
REMOTE SETTING		
REPLY FOR NO SIGNAL	:FORMat:NDATa[:WAVelength]	
LANGUAGE	:SYSTem:LANGuage	
POWER OFFSET	[ :SENSe]:CORRection:OFFSet[:MAGNitude]	
PARAMETER INITIALIZE		
MEAS PARAM CLEAR	:SYSTem:PRESet	
BUZZER		
CLICK	:SYSTem:BUZZer[:CLICk]	
WARNING	:SYSTem:BUZZer:WARning	
SET CLOCK	:SYSTem:DATE	Date
	:SYSTem:TIME	Time
COLOR MODE	:DISPlay:COLor	

## 5.2 Table of Soft Key to Remote Command Assignments

### DISPLAY

Soft Key	Remote Command	Note
VIEW MODE	:CONFigure[:SCALar]:POWer	SINGLE-WL
	:CONFigure:ARRay:POWer <sup>1</sup>	MULTI-WL
	:CALCulate3:DELta:WPOWer[:STATe] <sup>1</sup>	DELTA-WL
	:CALCulate3:GRID[:STATe]	GRID
GRID PARAMETER		
START WL	:CALCulate3:GRID:START[:WAVelength]	
START FREQ	:CALCulate3:GRID:START:FREQuency	
START WNUM	:CALCulate3:GRID:START:WNUMber	
STOP WL	:CALCulate3:GRID:STOP[:WAVelength]	
STOP FREQ	:CALCulate3:GRID:STOP:FREQuency	
STOP WNUM	:CALCulate3:GRID:STOP:WNUMber	
SEARCH AREA	:CALCulate3:GRID:SARea:FREQuency	
REF FREQ	:CALCulate3:GRID:REFerence:FREQuency	
SPACING	:CALCulate3:GRID:SPACing:FREQuency	
SHOW ALL <sup>1</sup>	:CALCulate3:GRID:DISPlay:ALL	
PREV PEAK <sup>1</sup>	-	
NEXT PEAK <sup>1</sup>	-	
LIST ONLY <sup>1</sup>	:DISPlay[:WINDow]:STATe	
SPECTRUM DISPLAY <sup>1</sup>	:DISPlay:WINDow2:STATe	
AUTO SCALE <sup>1</sup>	:DISPlay:WINDow2:TRACe[:SCALe]:ASCale	
SCALE <sup>1</sup>		
CENTER WL	:DISPlay:WINDow2:TRACe[:SCALe]:CENTer[:WAVeleng th]	Center wavelength
CENTER FREQ	:DISPlay:WINDow2:TRACe[:SCALe]:CENTer:FREQuency	Center frequency
CENTER WNUM	:DISPlay:WINDow2:TRACe[:SCALe]:CENTer:WNUMber	Center wavenumber
SPAN WL	:DISPlay:WINDow2:TRACe[:SCALe]:SPAN[:WAVelength]	Wavelength span
SPAN FREQ	:DISPlay:WINDow2:TRACe[:SCALe]:SPAN:FREQuency	Frequency span
SPAN WNUM	:DISPlay:WINDow2:TRACe[:SCALe]:SPAN:WNUMber	Wavenumber span
START WL	:DISPlay:WINDow2:TRACe[:SCALe]:LEFT[:WAVelength]	Start wavelength
START FREQ	:DISPlay:WINDow2:TRACe[:SCALe]:LEFT:FREQuency	Start frequency
START WNUM	:DISPlay:WINDow2:TRACe[:SCALe]:LEFT:WNUMber	Start wavenumber
STOP WL	:DISPlay:WINDow2:TRACe[:SCALe]:RIGHT[:WAVeleng th]	Stop wavelength
STOP FREQ	:DISPlay:WINDow2:TRACe[:SCALe]:RIGHT:FREQuency	Stop frequency
STOP WNUM	:DISPlay:WINDow2:TRACe[:SCALe]:RIGHT:WNUMber	Stop wavenumber
PEAK CENTER	:DISPlay:WINDow2:TRACe[:SCALe]:CENTer:PEAK	
INITIAL	:DISPlay:WINDow2:TRACe[:SCALe]:INITialize	
LIST BY <sup>1</sup>	:CONFigure:ARRay:POWer	Sort by power
	:CONFigure:ARRay:POWer:WAVelength	Sort by wavelength
DIGIT	:DISPlay:RESolution[:WAVelength]	
	:DISPlay:RESolution:FREQuency	
	:DISPlay:RESolution:WNUMber	
OVER VIEW DISPLAY <sup>1</sup>	—	
LABEL	:DISPlay[:WINDow]:TEXT:DATA	
DISPLAY OFF	:DISPlay[:WINDow]	

<sup>1</sup> For the -MW (multi-wavelength) suffix code type

### SEARCH(For the -MW (multi-wavelength) suffix code type)

Soft Key	Remote Command	Note
PEAK	:DISPlay:MARKer:MAXimum	
NEXT POWER	:DISPlay:MARKer:MAXimum:NEXT	
PREV POWER	:DISPlay:MARKer:MAXimum:PREVIOUS	
NEXT WL	:DISPlay:MARKer:MAXimum:RIGHT	
PREV WL	:DISPlay:MARKer:MAXimum:LEFT	
LIST BY	—	
AUTO PEAK SEARCH	:CALCulate2:ASEarch	

## ANALYSIS

Soft Key	Remote Command	Note
FABRY-PEROT LASER <sup>1</sup>	:CALCulate3:FPERot[:STATe]	
DRIFT MEASUREMENT/ PARAMETER SETTING	:CALCulate3:DRIFT[:STATe]	
REF SET	:CALCulate3:DRIFT:REfERENCE:RESEt	
DISPLAY MODE		
DELTA	:CALCulate3:DRIFT:PRESet	
MAX	:CALCulate3:DRIFT:MAXimum[:STATe]	
MIN	:CALCulate3:DRIFT:MINimum[:STATe]	
MAX-MIN	:CALCulate3:DRIFT:DIFFerence[:STATe]	
WAVELENGTH	:CALCulate3:DRIFT:WAVelength[:STATe]	
POWER	:CALCulate3:DRIFT:POWer[:STATe]	
DATA LOGGING	:CALCulate3:DLOGging:STATe	
LOGGING	:CALCulate3:DLOGging:MEASure:STATe	
CURSOR/SCALE	—	
SETUP		
LOGGING	:CALCulate3:DLOGging:LPARAMeter:ITEM	Logging item
PARAMETER	:CALCulate3:DLOGging:LPARAMeter:LMODe	Logging mode
	:CALCulate3:DLOGging:LPARAMeter:INTerval	Logging interval
	:CALCulate3:DLOGging:LPARAMeter:TDURation	Logging duration
	:CALCulate3:DLOGging:LPARAMeter:ASAVe[:STATe]	Auto data saving
GRAPH ITEM	—	
GRAPH CHANNEL	—	
CURSOR DATA	—	
FORMAT	—	
DATA DISPLAY	—	
DATA VIEW	—	
LOGGING DATA CLEAR	—	
SPECTRUM DISPLAY	—	
FILE	—	
WDM(OSNR)	:CALCulate3:SNR[:STATe]	
NOISE ALGO		
AUTO-CTR	:CALCulate3:SNR:AUTO	ON
MANUAL-FIX	:CALCulate3:SNR:AUTO	OFF
NOISE AREA	:CALCulate3:SNR:REfERENCE[:WAVelength]:RELative	
NOISE BW	:CALCulate3:SNR:REfERENCE:BWIDth	

1 For the -MW (multi-wavelength) suffix code type

## FILE

Soft Key	Remote Command	Note
WRITE		
MEMORY	:MMEMory:CDRive	
MAKE DIRECTORY	:MMEMory:MDIRectory	
FILE SORT	—	
EXECUTE	:MMEMory:STORe	
READ		
MEMORY	:MMEMory:CDRive	
FILE SORT	—	
EXECUTE	:MMEMory:LOAD	
ITEM SELECT	—	Specified when writing is executed
REMOVE USB STORAGE	:MMEMory:REMOVe	
FILE OPERATION		
MEMORY	:MMEMory:CDRive	
DELETE	:MMEMory:DELeTe	
COPY	:MMEMory:COpy	
RENAME	:MMEMory:REName	
MAKE DIRECTORY	:MMEMory:MDIRectory	
FILE SELECT	—	Other commands have a parameter for specifying the file name.

## 5.3 Remote Command Tree

The following tables lists the parameters of each command.

For information on which soft key each command corresponds to, see section 5.2.

For detailed descriptions of parameters, see sections 5.4 and 5.5.

### Common Commands

Command	Parameters	Reference Page
*CLS	none	5-13
*ESE	<integer>	5-13
*ESR?	none	5-13
*IDN?	none	5-13
*OPC	none	5-13
*RCL	1 2 3 4	5-13
*RST	none	5-13
*SAV	1 2 3 4	5-13
*SRE	<integer>	5-13
*STB?	none	5-14
*TRG	none	5-14
*TST?	none	5-14
*WAI	none	5-14

### CALCulate2

Command	Parameters	Reference Page
:CALCulate2		
:ASEarch <sup>1</sup>	ON OFF 0 1	5-15
:COUNT	<integer> MINimum MAXimum	5-15
:DATA?	FREQuency POWer WAVelength WNUMber  {ALL[,WAVelength FREQuency WNUMber]}	5-15
:MTHResh <sup>1</sup>	<thresh>	5-15
:PEXCursion	MINimum MAXimum DEFault <integer>	5-15
:POINTs?	none	5-15
:PTHReshold <sup>1</sup>		
:ABSolute	<NRf> MINimum MAXimum DEFault	5-15
[[:RELative]]	MINimum MAXimum DEFault <integer>	5-16
:MODE	RELative ABSolute	5-16
:WLIMit		
:START		
:FREQuency	<NRf> MINimum MAXimum	5-16
[[:WAVelength]]	<NRf> MINimum MAXimum	5-16
:WNUMber	<NRf> MINimum MAXimum	5-16
[[:STATe]]	ON OFF 0 1	5-16
:STOP		
:FREQuency	<NRf> MINimum MAXimum	5-17
[[:WAVelength]]	<NRf> MINimum MAXimum	5-17
:WNUMber	<NRf> MINimum MAXimum	5-17

<sup>1</sup> For the -MW (multi-wavelength) suffix code type

## CALCulate3

Command	Parameters	Reference Page
:CALCulate3		
:ASNR <sup>1</sup>		
:COUNT	<integer> MINmum MAXimum	5-17
:DATA?	<b>Drift</b> POWer FREQuency WAVelength WNUMber DROPPed  {ALL[,WAVelength FREQuency WNUMber]} <b>Delta</b> POWer FREQuency WAVelength WNUMber <b>GRID</b> STAtus {GRID[,WAVelength FREQuency WNUMber]}  {PEAK[,WAVelength FREQuency WNUMber POWer]}  {DEViation[,WAVelength FREQuency WNUMber]} {ALL[,WAVelength FREQuency WNUMber]} <b>WDM (OSNR)</b> POWer SIGNal NOISe {ALL[,WAVelength  FREQuency WNUMber]}	5-17
:DELTA <sup>1</sup>		
:POWer[:STATe]	0 OFF 1 ON	5-19
:PRESet	none	5-19
:REFeRence		
:FREQuency	<NRf> MINimum MAXimum	5-19
:POWer?	none	5-19
[:WAVelength]	<NRf> MINimum MAXimum	5-20
:WNUMber	<NRf> MINimum MAXimum	5-20
:WAVelength[:STATe]	0 OFF 1 ON	5-20
:WPOWer[:STATe]	0 OFF 1 ON	5-20
:DLOGging		
:ETIme?	none	5-20
:LPARameter		
:ASAVe		
:FNAMe?	none	5-20
[:STATe]	OFF INTernAl EXternAl	5-21
:INTerval	MINimum <NRf>	5-21
:ITEM	PEAK FPLD	5-21
:LMODE	MODE1 MODE2 MODE3	5-21
:TDURation	<integer>	5-21
:MEASure		
:STATe	0 OFF 1 ON	5-21
:STATe	0 OFF 1 ON	5-21
:DRIFT		
:DIFFerence[:STATe]	0 OFF 1 ON	5-22
:ETIme	none	5-22
:MAXimum[:STATe]	0 OFF 1 ON	5-22
:MINimum[:STATe]	0 OFF 1 ON	5-22
:POWer[:STATe]	0 OFF 1 ON	5-22
:WAVelength[:STATe]	0 OFF 1 ON	5-22
:PRESet	none	5-22
[:STATe]	0 OFF 1 ON	5-22
:REFeRence		
:RESet	none	5-23
[:STATe]	0 OFF 1 ON	5-23
:FPERot <sup>1</sup>		
[:STATe]	0 OFF 1 ON	5-23
:FWHM		
[:WAVelength]?	none	5-23
:FREQuency?	none	5-23
:WNUMber?	none	5-23
:MEAN		
[:WAVelength]?	none	5-23
:FREQuency?	none	5-23
:WNUMber?	none	5-23



### 5.3 Remote Command Tree

Command	Parameters	Reference Page
:MODE:SPACing		
[:WAVelength]?	none	5-23
:FREQuency?	none	5-23
:WNUMber?	none	5-23
:PEAK		
[:WAVelength]?	none	5-23
:FREQuency?	none	5-23
:WNUMber?	none	5-23
:POWer		
[:DBM]?	none	5-23
:WATTs?	none	5-23
:POWer		
[:DBM]?	none	5-23
:WATTs?	none	5-23
:SIGMa		
[:WAVelength]?	none	5-24
:FREQuency?	none	5-24
:WNUMber?	none	5-24
:GRID		
:DISPlay		
:ALL	0 OFF 1 ON	5-24
:REFErence		
:FREQuency	DEFAult <NRf>	5-24
:STARt		
[:WAVelength]	<NRf>	5-24
:FREQuency	<NRf>	5-24
:WNUMber	<NRf>	5-24
[:STATe]	0 OFF 1 ON	5-24
:STOP		
[:WAVelength]	<NRf>	5-24
:FREQuency	<NRf>	5-24
:WNUMber	<NRf>	5-25
:SPACing		
:FREQuency	<NRf>	5-25
:SARea		
:FREQuency	<NRf>	5-25
:POINTs?	none	5-25
:PRESet	none	5-25
:SNR		
:AUTO	0 OFF 1 ON	5-25
:REFErence		
[:WAVelength]		
:RELative	<NRf>	5-25
:BWIDth	<NRf>	5-25
[:STATe]	0 OFF 1 ON	5-25

1 For the -MW (multi-wavelength) suffix code type

### CONFigure

Command	Parameters	Reference Page
:CONFigure?	none	5-26
[:SCALar]		
:POWer	MAXimum MINimum DEFAult <NRf>	5-26
:FREQuency	MAXimum MINimum DEFAult <NRf>	5-26
:WAVelength	MAXimum MINimum DEFAult <NRf>	5-26
:WNUMber	MAXimum MINimum DEFAult <NRf>	5-26
:ARRay <sup>1</sup>		
:POWer	MAXimum MINimum DEFAult <NRf>	5-27
:FREQuency	MAXimum MINimum DEFAult <NRf>	5-27
:WAVelength	MAXimum MINimum DEFAult <NRf>	5-27
:WNUMber	MAXimum MINimum DEFAult <NRf>	5-27

1 For the -MW (multi-wavelength) suffix code type

## DISPlay

Command	Parameters	Reference Page
:DISPlay		
:COLor	0 1	5-28
[ :WINDow]	0 OFF 1 ON	5-28
:MARKer <sup>1</sup>		
:MAXimum	none	5-28
:LEFT	none	5-28
:NEXT	none	5-28
:PREVious	none	5-28
:RIGHT	none	5-28
:RESolution		
[:WAVelength]	R0.0001 R0.001 R0.01 R0.1 MAXimum MINimum	5-28
:FREQuency	R0.00001 R0.0001 R0.001 R0.01 MAXimum MINimum	5-28
:WNUMber	R0.001 R0.01 R0.1 R1 MAXimum MINimum	5-29
:UNIT		
:WAVelength	NM THZ ICM	5-29
[ :WINDow]		
:TEXT		
:DATA	<"string">	5-29
:STATe	0 OFF 1 ON	5-29
:WINDow2 <sup>1</sup>		
:STATe	0 OFF 1 ON	5-29
:TRACe		
[:SCALe]		
:AUTOmeasure	none	5-29
:ASCAle	none	5-29
:INITialize	none	5-30
:LEFT		
[:WAVelength]	<NRf> MINimum MAXimum	5-30
:FREQuency	<NRf> MINimum MAXimum	5-30
:WNUMber	<NRf> MINimum MAXimum	5-30
:RIGHT		
[:WAVelength]	<NRf> MINimum MAXimum	5-31
:FREQuency	<NRf> MINimum MAXimum	5-31
:WNUMber	<NRf> MINimum MAXimum	5-31
:CENTer		
[:WAVelength]	<NRf>	5-31
:FREQuency	<NRf>	5-31
:WNUMber	<NRf>	5-32
:PEAK	none	5-32
:SPAN		
[:WAVelength]	<NRf> MAXimum	5-32
:FREQuency	<NRf> MAXimum	5-32
:WNUMber	<NRf> MAXimum	5-32

<sup>1</sup> For the -MW (multi-wavelength) suffix code type

### 5.3 Remote Command Tree

## FEtCh

Command	Parameters	Reference Page
:FEtCh?	none	5-33
:ARRay		
:POWer?	MAximum MINimum DEFault <NRf>	5-33
:FREQuency?	MAximum MINimum DEFault <NRf>	5-33
:WAVelength?	MAximum MINimum DEFault <NRf>	5-34
:WNUmber?	MAximum MINimum DEFault <NRf>	5-34
[:SCALar]		
:POWer?	MAximum MINimum DEFault <NRf>	5-34
:FREQuency?	MAximum MINimum DEFault <NRf>	5-35
:WAVelength?	MAximum MINimum DEFault <NRf>	5-35
:WNUmber?	MAximum MINimum DEFault <NRf>	5-35

## FORMat

Command	Parameters	Reference Page
:FORMa		
:NDAa		
[:WAVelength]	<NRf>	5-35

## MEASure

Command	Parameters	Reference Page
:MEASure		
:ARRay		
:POWer?	MAximum MINimum DEFault <NRf>	5-36
:FREQuency?	MAximum MINimum DEFault <NRf>	5-36
:WAVelength?	MAximum MINimum DEFault <NRf>	5-36
:WNUmber?	MAximum MINimum DEFault <NRf>	5-37
[:SCALar]		
:POWer?	MAximum MINimum DEFault <NRf>	5-37
:FREQuency?	MAximum MINimum DEFault <NRf>	5-37
:WAVelength?	MAximum MINimum DEFault <NRf>	5-38
:WNUmber?	MAximum MINimum DEFault <NRf>	5-38

## MMEMory

Command	Parameters	Reference Page
:MMEMory		
:CATalog?	[<"directory"> ROOT[,INTernal EXTernal]]	5-38
:CDIRectory	<"directory"> ROOT[,INTernal EXTernal]	5-39
:CDRive	INTernal EXTernal	5-39
:COpy	<"source_file_name">[,INTernal EXTernal], <"dest_file_name">[,INTernal EXTernal]	5-39
:DATA?	<"filename">[,INTernal EXTernal]	5-39
:DELeTe	<"filename">[,INTernal EXTernal]	5-39
:INformation?	<"filename">[,INTernal EXTernal]	5-39
:LOAD	<"filename">[,INTernal EXTernal]	5-39
:MDIRectory	<"directory_name">[,INTernal EXTernal]	5-39
:PWDirectory?	none	5-40
:REMove	none	5-40
:REName	<"new_file_name">,<"old_file_name">[,INTernal EXTernal]	5-40
:STORe	TABLE SETup SIMage1 SIMage2 SIMage3 DLOGging1 DLOGging2,<"filename">[,INTernal EXTernal]	5-40

## READ

Command	Parameters	Reference Page
:READ?	none	5-40
:ARRay		
:POWer?	MAXimum MINimum DEFault <NRf>	5-41
:FREQuency?	MAXimum MINimum DEFault <NRf>	5-41
:WAVelength?	MAXimum MINimum DEFault <NRf>	5-41
:WNUmber?	MAXimum MINimum DEFault <NRf>	5-41
[ :SCALar]		
:POWer?	MAXimum MINimum DEFault <NRf>	5-42
:FREQuency?	MAXimum MINimum DEFault <NRf>	5-42
:WAVelength?	MAXimum MINimum DEFault <NRf>	5-42
:WNUmber?	MAXimum MINimum DEFault <NRf>	5-42

## SENSe

Command	Parameters	Reference Page
[ :SENSe]		
:CORRection		
:DEVice	NARRow BROad	5-43
:MEDium	AIR VACuum	5-43
:OFFSet		
[ :MAGNitude]	<NRf> MINimum MAXimum	5-43
:URATe	NORMal FAST	5-43

## STATus

Command	Parameters	Reference Page
:STATus		
:OPERation		
:CONDition?	none	5-43
:ENABle	<integer>	5-43
[ :EVENT]?	none	5-43
:NTRansition	<integer>	5-43
:PTRansition	<integer>	5-43
:PRESet	none	5-44
:QUEStionable		
:CONDition?	none	5-44
:ENABle	<integer>	5-44
[ :EVENT]?	none	5-44
:NTRansition	<integer>	5-44
:PTRansition	<integer>	5-44

### 5.3 Remote Command Tree

## SYSTem

Command	Parameters	Reference Page
:SYSTem		
:BUZZer		
[:CLICk]	0 OFF 1 ON	5-44
:WARNing	0 OFF 1 ON	5-44
:CAPability		
:WAVelength?	none	5-44
:DATE	<year>, <month>, <day>	5-44
:ENVironment?	none	5-45
:ERRor?	none	5-45
:INFormation?	0 1	5-45
:LANGUage	ENGLISH CHINEse JAPanese	5-45
:PRESet	none	5-45
:REFLaser		
:CONDition?	none	5-45
:COUNter?	none	5-45
:OTIME?	none	5-45
:TIME	<hour>, <minute>, <second>	5-46
:VERSion?	none	5-46

## TRIGger

Command	Parameters	Reference Page
[:TRIGger]		
:ABORT	none	5-46
:INITiate		
:CONTinuous	0 OFF 1 ON	5-46
[:IMMediate]	none	5-46

## UNIT

Command	Parameters	Reference Page
:UNIT		
[:POWER]	W DBM	5-47
:WL	THZ NM ICM	5-47

## 5.4 Common Commands

The commands in this group are defined in IEEE 488.2-1991 and are independent from the instrument's individual functions.

There are no front panel keys that correspond to the commands in this group.

### \*CLS (Clear Status)

**Function** Clears the error queue, Standard Event Register, and Status Byte Register.

**Syntax** \*CLS

**Example** \*CLS

**Description** This is an overlapping command.

### \*ESE (Standard Event Status Enable)

**Function** Sets or queries the Standard Event Enable Register.

**Syntax** \*ESE<wsp><integer>  
\*ESE?

**Example** \*ESE 255  
\*ESE? -> +255<END>

**Description** This is an overlapping command.

### \*ESR? (Standard Event Status Register)

**Function** Queries the Standard Event Status Register value.

**Syntax** \*ESR?

**Example** \*ESR? -> +128<END>

**Description**

- A query using \*ESR? will clear the contents of the Standard Event Register.
- This is an overlapping command.

### \*IDN? (Identification)

**Function** Queries the device model, serial number, and firmware version.

**Syntax** \*IDN?

**Response**  
YOKOGAWA,AQ615xB,<SerialNo>,<Version>  
AQ615xB: Model  
<SerialNo>: Serial number  
<Version>: Firmware version

**Example** \*IDN? -> YOKOGAWA,AQ6151B,012345678,01.00<END>

**Description** This is an overlapping command.

### \*OPC (Operation Complete)

**Function** Sets or queries bit 0 (the OPC bit) of the Standard Event Register upon the completion of an overlap operation.

**Syntax** \*OPC  
\*OPC?

**Example** \*OPC  
\*OPC? -> 1<END>

**Description**

- To set bit 0, specify 1. When 1 is returned in response to a query, the overlap operation is finished.
- This is an overlapping command.

For details on the timing of command operation completion, see page 4-8.

### \*RCL (Recall Command)

**Function** Returns the instrument settings to the contents saved with the \*SAV command.

**Syntax** \*RCL<wsp>1|2|3|4  
1|2|3|4: Preset number

**Example** \*RCL 1

**Description** Select the settings that you want to return the instrument to (1 to 4).

### \*RST (Reset)

**Function** Initializes settings.

**Syntax** \*RST

**Example** \*RST

### \*SAV (Save Command)

**Function** Saves the current instrument settings.

**Syntax** \*SAV<wsp>1|2|3|4  
1|2|3|4: Preset number

**Example** \*SAV 1

**Description** You can save up to four sets of settings.

### \*SRE (Service Request Enable)

**Function** Sets or queries the Service Request Enable Register.

**Syntax** \*SRE<wsp><integer>  
\*SRE?

**Example** \*SRE 255  
\*SRE? -> +255<END>

**Description** This is an overlapping command.

## 5.4 Common Commands

### **\*STB? (Read Status Byte)**

Function     Queries the Status Byte Register value.  
Syntax       \*STB?  
Example      \*STB? -> +12<END>  
Description   This is an overlapping command.

### **\*TRG (Trigger)**

Function     Starts a single measurement.  
Syntax       \*TRG  
Example      \*TRG  
Description   This is an overlappable command.

### **\*TST? (Self Test)**

Function     Executes a self-test and queries the result.  
Syntax       \*TST?  
              Response        0: No error  
                              Not 0: Error (error code)  
Example      \*TST? -> 0<END>  
Description   • The instrument always returns 0.  
              • This is an overlapping command.

### **\*WAI (Wait to Continue)**

Function     Sets the instrument so that it will not execute other commands until the execution of the current command is finished.  
Syntax       \*WAI  
Example      \*WAI  
Description   This is an overlapping command.  
              For details on the timing of command operation completion, see page 4-8.

## 5.5 Device-specific Commands

This section provides parameters and syntax examples of commands used to operate the functions of the instrument. For information on which soft key each command corresponds to, see section 5.2.

### CALCulate2 Sub System Commands

#### :CALCulate2:ASEarch

**Function** Sets or queries the on/off state of auto peak (wavelength/power) detection.

**Syntax** :CALCulate2:ASEarch<wsp>ON|OFF|0|1  
:CALCulate2:ASEarch?

ON|1: Auto peak detection on  
OFF|0: Auto peak detection off

**Example** :CALC2:ASE ON  
:CALC2:ASE? -> 1<END>

**Description** This command is valid on the -MW (multi-wavelength) suffix code type.

#### :CALCulate2:COUNT

**Function** Sets or queries the peak detection average count.

**Syntax** :CALCulate2:COUNT<wsp><average\_times>  
:CALCulate2:COUNT?

<average\_times> (average count):  
<integer>|MINimum|MAXimum  
MINimum:1  
MAXimum:100

**Example** :CALC2:COUNT 10  
:CALC2:COUNT? -> +10<END>

#### :CALCulate2:DATA?

**Function** Queries the measured values of all detected peaks.

**Syntax** :CALCulate2:DATA?<wsp>FREQUENCY|POWER|WAVELENGTH|WNUMBER|{ALL[,WAVELENGTH|FREQUENCY|WNUMBER]}

FREQUENCY: Queries the wavelength in unit of frequency.  
WAVELENGTH: Queries the wavelength in unit of wavelength.  
WNUMBER: Queries the wavelength in unit of wavenumber.  
POWER: Queries the power value.  
All: Queries all power values and wavelengths (frequency, wavelength, or wavenumber).

**Example** :CALC2:DATA? FREQ  
-> +1.93596570E+014,+1.93738272E+014,+1.93880006E+014<END>

**Description**

- Returns all detected peaks in floating-point numbers, each peak separated by a comma.
- The power value is returned in the specified unit.
- If no peaks have been detected (no signal), the following value is returned.  
Wavelength, power (mW, µW):  
0.000000E+000  
Power (dBm): -2.000000E+002
- This is an overlapping command.

#### :CALCulate2:MTHResh

**Function** Sets or queries the frequency threshold for judging channel matching.

**Syntax** :CALCulate2:MTHResh<wsp><thresh>  
:CALCulate2:MTHResh?

<thresh>: <NRf> Specify the channel matching threshold in unit of Hz (1 GHz to 99 GHz).

**Example** :CALC2:MTHR 2GHZ  
:CALC2:MTHR?  
-> +2.00000000E+009<END>

**Description** This command is valid on the -MW (multi-wavelength) suffix code type.

#### :CALCulate2:PEXCursion

**Function** Sets or queries the peak difference to use to detect peaks.

**Syntax** :CALCulate2:PEXCursion<wsp>  
<pexcursion\_value>  
:CALCulate2:PEXCursion?

<pexcursion\_value> (power difference):  
MINimum|MAXimum|DEFAULT|<integer>  
MINimum: 1 dB  
MAXimum: 30 dB  
DEFAULT: 15 dB

**Example** :CALC2:PEXC 10  
:CALC2:PEXC? -> +10<END>

#### :CALCulate2:POINTS?

**Function** Queries the number of detected peaks.

**Syntax** :CALCulate2:POINTS?

**Example** :CALC2:POIN? -> +3<END>

**Description**

- The maximum number of detected peaks is 1024.
- This is an overlapping command.

#### :CALCulate2:PTHReshold:ABSolute

**Function** Sets or queries the peak detection threshold value.

**Syntax** :CALCulate2:PTHReshold:ABSolute  
<wsp><thresh>  
:CALCulate2:PTHReshold:ABSolute?

<thresh> (threshold value):  
<NRf>|MINimum|MAXimum|DEFAULT  
MINimum: -40 dBm  
MAXimum: 10 dBm  
DEFAULT: -20 dBm

**Example** :CALC2:PTHR:ABS -20  
:CALC2:PTHR:ABS?  
-> -2.00000000E+001<END>

**Description** This command is valid on the -MW (multi-wavelength) suffix code type.



## 5.5 Device-specific Commands

### **:CALCulate2:PTHReshold[:RELative]**

Function	Sets or queries the peak detection threshold value as a relative value in reference to the maximum power peak.
Syntax	<pre>:CALCulate2:PTHReshold[:RELative] &lt;wsp&gt;&lt;thresh&gt; :CALCulate2:PTHReshold[:RELative]? &lt;thresh&gt; (threshold value):     MINimum/MAXimum/DEFault/&lt;integer&gt;     MINimum: 0 dB     MAXimum: 40 dB     DEFault: 10 dB</pre>
Example	<pre>:CALC2:PTHR 9 :CALC2:PTHR? -&gt; +9&lt;END&gt;</pre>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

### **:CALCulate2:PTHReshold:MODE**

Function	Sets or queries the definition of the peak detection threshold value.
Syntax	<pre>:CALCulate2:PTHReshold:MODE&lt;wsp&gt; RELative ABSolute :CALCulate2:PTHReshold:MODE? RELative: Threshold value defined as a relative value ABSolute: Threshold value defined as an absolute value</pre>
Example	<pre>:CALC2:PTHR:MOD REL :CALC2:PTHR:MODE? -&gt; REL&lt;END&gt;</pre>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

### **:CALCulate2:WLIMit:START:FREQuency**

Function	Sets or queries the start frequency of the measurement range limit of peak detection.
Syntax	<pre>:CALCulate2:WLIMit:START :FREQuency&lt;wsp&gt;&lt;freq&gt; :CALCulate2:WLIMit:START:FREQuency? &lt;freq&gt;: (start frequency)     &lt;NRf&gt; MINimum MAXimum     MINimum: 181.69THz(-10 suffix code type)     176.35THz(-20 suffix code type)     176.35THz(-30 suffix code type)     MAXimum: Stop frequency - 0.1 THz</pre>
Example	<pre>:CALC2:WLIM:STAR:FREQ 191THZ :CALC2:WLIM:STAR:FREQ? -&gt; +1.91000000E+014&lt;END&gt;</pre>
Description	Query results are returned in Hz.

### **:CALCulate2:WLIMit:START[:WAVelength]**

Function	Sets or queries the start wavelength of the measurement range limit of peak detection.
Syntax	<pre>:CALCulate2:WLIMit:START[:WAVelength] &lt;wsp&gt;&lt;wavelength&gt; :CALCulate2:WLIMit:START [:WAVelength]? &lt;wavelength&gt; (start wavelength):     &lt;NRf&gt; MINimum MAXimum     MINimum: 1270 nm (-10 suffix code type)     1200 nm (-20 suffix code type)     900 nm (-30 suffix code type)     MAXimum: Stop wavelength - 1 nm</pre>
Example	<pre>:CALC2:WLIM:STAR 1500NM :CALC2:WLIM:STAR? -&gt; +1.50000000E-006&lt;END&gt;</pre>
Description	Query results are returned in m.

### **:CALCulate2:WLIMit:START:WNUMber**

Function	Sets or queries the start wavenumber of the measurement range limit of peak detection.
Syntax	<pre>:CALCulate2:WLIMit:START: WNUMber&lt;wsp&gt;&lt;wnumber&gt; :CALCulate2:WLIMit:START:WNUMber? &lt;wnumber&gt; (wavenumber):     &lt;NRf&gt; MINimum MAXimum     MINimum: 6060.0 cm<sup>-1</sup> (-10 suffix code type)     5882.4 cm<sup>-1</sup> (-20 suffix code type)     5882.4 cm<sup>-1</sup> (-30 suffix code type)     MINimum: 6060 cm<sup>-1</sup>     MAXimum: Stop wavenumber - 1 cm<sup>-1</sup></pre>
Example	<pre>:CALC2:WLIM:STAR:WNUM 6400ICM :CALC2:WLIM:STAR:WNUM? -&gt; +6.40000000E+005&lt;END&gt;</pre>
Description	Query results are returned in m <sup>-1</sup> .

### **:CALCulate2:WLIMit[:STATe]**

Function	Sets or queries the on/off state of the measurement range limit of peak detection.
Syntax	<pre>:CALCulate2:WLIMit[:STATe]&lt;wsp&gt; 0 OFF 1 ON :CALCulate2:WLIMit[:STATe]? 0 OFF: Measurement range limit off 1 ON: Measurement range limit on</pre>
Example	<pre>:CALC2:WLIM ON :CALC2:WLIM? -&gt; 1&lt;END&gt;</pre>

**:CALCulate2:WLIMit:STOP:FREQuency**

Function	Sets or queries the stop frequency of the measurement range limit of peak detection.
Syntax	:CALCulate2:WLIMit:STOP:FREQuency <wsp><frequency> :CALCulate2:WLIMit:STOP:FREQuency? <frequency> (stop frequency): <NRf> MINimum MAXimum MINimum: Start frequency + 0.1 THz MAXimum: 230.06 THz (-10 suffix code type) 249.83 THz (-20 suffix code type) 333.11 THz (-30 suffix code type)
Example	:CALC2:WLIM:STOP:FREQ 195THZ :CALC2:WLIM:STOP:FREQ? -> +1.95000000E+014<END>
Description	Query results are returned in Hz.

**:CALCulate2:WLIMit:STOP[:WAVelength]**

Function	Sets or queries the stop wavelength of the measurement range limit of peak detection.
Syntax	:CALCulate2:WLIMit:STOP[:WAVelength] <wsp><wavelength> :CALCulate2:WLIMit:STOP[:WAVelength]? <wavelength> (stop wavelength): <NRf> MINimum MAXimum MINimum: Start wavelength + 1 nm MAXimum: 1650 nm (-10 suffix code type) 1700 nm (-20 suffix code type) 1700 nm (-30 suffix code type)
Example	:CALC2:WLIM:STOP 1640NM :CALC2:WLIM:STOP? -> +1.64000000E-006<END>
Description	Query results are returned in m.

**:CALCulate2:WLIMit:STOP:WNUMber**

Function	Sets or queries the stop wavenumber of the measurement range limit of peak detection.
Syntax	:CALCulate2:WLIMit:STOP: WNUMber<wsp><wnumber> :CALCulate2:WLIMit:STOP:WNUMber? <wnumber> (wavenumber): <NRf> MINimum MAXimum MINimum: Start wavenumber + 1 cm <sup>-1</sup> MAXimum: 7875.0 cm <sup>-1</sup> (-10 suffix code type) 8333.3 cm <sup>-1</sup> (-20 suffix code type) 11111.1 cm <sup>-1</sup> (-30 suffix code type)
Example	:CALC2:WLIM:STOP:WNUM 7800ICM :CALC2:WLIM:STOP:WNUM? -> +7.80000000E+005<END>
Description	Query results are returned in m <sup>-1</sup> .

**CALCulate3 Sub System Commands****:CALCulate3:ASNR:COUNT**

Function	Set or queries the averaging number of OSNR calculation.
Syntax	:CALCulate3:ASNR:COUNT<wsp><count>  MINimum MAXimum :CALCulate3:ASNR:COUNT? <count>: <integer> type, 1 to 100 (averaging number) MINimum: 1 MAXimum: 100
Example	:CALC3:ASNR:COUN 3 :CALC3:ASNR:COUN? -> 3<END>
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>This setting and averaging count setting (:CALC2:COUNT) are common.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:CALCulate3:DATA?**

Function	Queries the results of a drift measurement, delta measurement, grid measurement, or WDM(OSNR). The delta measurement is valid on the -MW (multi-wavelength) suffix code type.
Syntax	For drift measurement: :CALCulate3:DATA?<wsp>POWER FREQuency WAVelength WNUMber DROPPed {ALL[,WAVelength FREQuency WNUMber]} For delta measurement: :CALCulate3:DATA?<wsp>POWER FREQuency WAVelength WNUMber For grid measurement: :CALCulate3:DATA?<wsp>STATUS {GRID[,WAVelength FREQuency WNUMber]} {PEAK[,WAVelength FREQuency WNUMber POWER]} {DEVIation[,WAVelength FREQuency WNUMber]} {ALL[,WAVelength FREQuency WNUMber]} For WDM(OSNR): :CALCulate3:DATA?<wsp>POWER SIGNal NOISe {ALL[,WAVelength FREQuency WNUMber]}
Example	:CALC3:DATA? POW -> 4.80000000E-001, -3.60000000E-001, +5.70000000E-001<END>
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>The results of the drift, delta, grid or WDM(OSNR) measurement, whichever is turned on, are returned.</li> <li>For details on turning on or off drift measurement, see the :CALCulate3:DRIFT[:STATe] command.</li> <li>For details on turning on or off delta measurement, see the commands below. :CALCulate3:DELTA:POWER[:STATe] :CALCulate3:DELTA:WAVelength[:STATe] :CALCulate3:DELTA:WPOWER[:STATe]</li> </ul>

## 5.5 Device-specific Commands

- For details on turning on or off grid measurement, see the :CALCulate3:GRID[:STATe] command.
- For details on turning on or off WDM(OSNR) measurement, see the :CALCulate3:SNR[:STATe] command.

For drift measurement

- Of the items listed below, the response will contain the items that are turned on (specified to be measured). The number of values of each item will be equal to the number of detected peaks.  
MAX, MIN, MAX-MIN, DELTA, Wavelength, Power, Ref  
Example, if MAX is turned on, and three peaks have been detected, three MAX values will be returned.  
You can turn on or off each item with commands. For details, see the descriptions of the relevant commands.
- If you specify the ALL parameter, the response will contain the following values separated by commas in this order: DROPPed, MAX POWER, MIN POWER, MAX-MIN POWER, REF POWER, POWER, MAX WAVElength, MIN WAVElength, MAX-MIN WAVElength, REF WAVElength, and WAVElength.
- Specifying the ALL,WAVElength parameter is the same as specifying the ALL parameter.
- If you specify the ALL,FREQUENCY parameter, the response will contain the following values separated by commas in this order: DROPPed, MAX POWER, MIN POWER, MAX-MIN POWER, REF POWER, POWER, MAX FREQUENCY, MIN FREQUENCY, MAX-MIN FREQUENCY, REF FREQUENCY, and FREQUENCY.
- If you specify the ALL,WNUMBER parameter, the response will contain the following values separated by commas in this order: DROPPed, MAX POWER, MIN POWER, MAX-MIN POWER, REF POWER, POWER, MAX WNUMBER, MIN WNUMBER, MAX-MIN WNUMBER, REF WNUMBER, and WNUMBER.
- The response data format is shown below.  
DROPPed:  
0: Normal data, 1: Drop data  
POWER, WAVElength, FREQUENCY, WNUMBER:  
Floating-point number

For delta measurement, This command is valid on the -MW (multi-wavelength) suffix code type.

- Returns the values of items specified with parameters in floating-point numbers for all detected peaks. Each value separated by a comma.  
Power, Wavelength, Frequency, Wnumber

For grid measurement

- Returns the measured values of items specified with parameters in order by grid. Each value is separated by a comma.  
STATus Peak presence  
0: Not present  
1: Present  
2: Multiple peaks present  
GRID Grid wavelength (the unit depends on the wavelength unit)  
GRID,FREQUENCY Grid's frequency  
GRID,WAVElength Grid's wavelength  
GRID,WNUMBER Grid's wavenumber  
DEViation Deviation between the peak closest to the grid and the grid (the unit depends on the wavelength unit)  
DEViation,FREQUENCY Deviation between the peak closest to the grid and the grid (frequency)  
DEViation,WAVElength Deviation between the peak closest to the grid and the grid (wavelength)  
DEViation,WNUMBER Deviation between the peak closest to the grid and the grid (wavenumber)  
PEAK Wavelength of the peak in the grid (the unit depends on the wavelength unit)  
PEAK,FREQUENCY Frequency of the peak in the grid  
PEAK,WAVElength Wavelength of the peak in the grid  
PEAK,WNUMBER: Wavenumber of the peak in the grid  
PEAK,POWER Power of the peak in the grid  
ALL Grid number, status, grid wavelength, deviation, peak wavelength(the unit depends on the wavelength unit), peak power  
ALL,FREQUENCY Grid number, status, grid frequency, deviation, peak frequency, peak power  
ALL,WAVElength Grid number, status, grid wavelength, deviation, peak wavelength, peak power  
ALL,WNUMBER Grid number, status, grid wavenumber, deviation, peak wavenumber, peak power
- If SHOW ALL is ON, the measured values of all grids are returned.  
If SHOW ALL is OFF, the measured values of grids with peaks are returned.

For WDM (OSNR) measurement

- Returns the measured values of items specified with parameters. Each value is separated by a comma.  
**POWer** OSNR array  
**SIGNal** Signal power array  
**NOISe** Noise power array  
**ALL** Array of wavelengths, OSNR, signal power, noise power  
**ALL, FREQuency** Array of frequencies, OSNR, signal power, noise power  
**ALL, WAVelength** Wavelength, OSNR, signal power, noise power  
**ALL, WNUMber** Wavenumber, OSNR, signal power, noise power
- To acquire peak, power, and other measured values, use the **:CALC2:DATA?** command.

#### **:CALCulate3:DELTA:POWer[:STATE]**

Function	Sets or queries the on/off state of power delta measurement.
Syntax	<b>:CALCulate3:DELTA:POWer[:STATE]</b> <wsp>0 OFF 1 ON <b>:CALCulate2:DELTA:POWer[:STATE]?</b> 0 OFF: Delta is not measured. 1 ON: Delta is measured.
Example	<b>:CALC3:DELTA:POW ON</b> <b>:CALC3:DELTA:POW? -&gt; 1&lt;END&gt;</b>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

#### **:CALCulate3:DELTA:PRESet**

Function	Aborts delta measurements.
Syntax	<b>:CALCulate3:DELTA:PRESet</b>
Example	<b>:CALC3:DELTA:PRES</b>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

#### **:CALCulate3:DELTA:REference:FREQuency**

Function	Sets or queries the reference peak of delta measurement in frequency.
Syntax	<b>:CALCulate3:DELTA:REference:FREQuency</b> <wsp><frequency> <b>:CALCulate3:DELTA:REference:FREQuency?</b> <frequency> (frequency): <NRf>/MINimum/MAXimum MINimum: 181.69 THz (-10 suffix code type) 176.35 THz (-20 suffix code type) 176.35 THz (-30 suffix code type) MAXimum: 230.06 THz (-10 suffix code type) 249.83 THz (-20 suffix code type) 333.11 THz (-30 suffix code type)
Example	<b>:CALC3:DELTA:REF:FREQ 193.8THZ</b> <b>:CALC3:DELTA:REF:FREQ?</b> -> +1.93878971E+014<END>
Description	<ul style="list-style-type: none"> <li>• Query results are returned in Hz. The peak closest to the frequency specified by this command becomes the reference. Therefore, the specified frequency and the query result may differ.</li> <li>• This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

#### **:CALCulate3:DELTA:REference:POWer?**

Function	Queries the power of the reference peak of delta measurement.
Syntax	<b>:CALCulate3:DELTA:REference:POWer?</b>
Example	<b>:CALC3:DELTA:REF:POW?</b> -> -1.02600000E+001<END>
Description	<ul style="list-style-type: none"> <li>• Query results are returned in dBm or W according to the setting.</li> <li>• This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

## 5.5 Device-specific Commands

### **:CALCulate3:DELTA:REference[:WAVelen gth]**

Function	Sets or queries the reference peak of delta measurement in wavelength.
Syntax	:CALCulate3:DELTA:REference [:WAVlength]<wsp><wavelength> :CALCulate3:DELTA:REference [:WAVlength]? <wavelength> (wavelength): <NRf>/MINimum/MAXimum MINimum: 1270 nm (-10 suffix code type) 1200 nm (-20 suffix code type) 900 nm (-30 suffix code type) MAXimum: 1650 nm (-10 suffix code type) 1700 nm (-20 suffix code type) 1700 nm (-30 suffix code type)
Example	:CALC3:DELT:REF 1547.4NM :CALC3:DELT:REF? -> +1.54741791E-006<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m. The peak closest to the wavelength specified by this command becomes the reference. Therefore, the specified wavelength and the query result may differ.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### **:CALCulate3:DELTA:REference:WNUMBER**

Function	Sets or queries the reference peak of delta measurement in wavenumber.
Syntax	:CALCulate3:DELTA:REference: WNUMBER<wsp><wnumber> :CALCulate3:DELTA:REference: WNUMBER? <wnumber> (wavenumber): <NRf>/MINimum/MAXimum MINimum: 6060.0 cm <sup>-1</sup> (-10 suffix code type) 5882.4 cm <sup>-1</sup> (-20 suffix code type) 5882.4 cm <sup>-1</sup> (-30 suffix code type) MAXimum: 7875.0 cm <sup>-1</sup> (-10 suffix code type) 8333.3 cm <sup>-1</sup> (-20 suffix code type) 11111.1 cm <sup>-1</sup> (-30 suffix code type)
Example	:CALC3:DELT:REF:WNUM 646700 :CALC3:DELT:REF:WNUM? -> +6.46710630E+005<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m<sup>-1</sup>. The peak closest to the wavenumber specified by this command becomes the reference. Therefore, the specified wavenumber and the query result may differ.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### **:CALCulate3:DELTA:WAVlength[:STATE]**

Function	Sets or queries the on/off state of delta measurement.
Syntax	:CALCulate3:DELTA:WAVlength [:STATE]<wsp>0 OFF 1 ON :CALCulate3:DELTA:WAVlength[: STATE]? 0 OFF: Delta is not measured. 1 ON: Delta is measured.
Example	:CALC3:DELT:WAV ON :CALC3:DELT:WAV? -> 1<END>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

### **:CALCulate3:DELTA:WPOwer[:STATE]**

Function	Sets or queries the on/off state of delta measurements.
Syntax	:CALCulate3:DELTA:WPOwer [:STATE]<wsp>0 OFF 1 ON :CALCulate3:DELTA:WPOwer[:STATE]? 0 OFF: Delta is measured. 1 ON: Delta is measured.
Example	:CALC3:DELT:WPOW ON :CALC3:DELT:WPOW? -> 1<END>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

### **:CALCulate3:DLOGging:ETIME?**

Function	Queries the elapsed time of data logging (in seconds).
Syntax	:CALCulate3:DLOGging:ETIME?
Example	:CALC3:DLOG:ETIM? -> 30<END>
Description	<ul style="list-style-type: none"> <li>This is an overlap command.</li> <li>This command is invalid when data logging is stopped.</li> </ul>

### **:CALCulate3:DLOGging:LPAParameter:**

#### **ASAVE:FNAME?**

Function	Queries the name of the file saved last with the auto save function of data logging.
Syntax	:CALCulate3:DLOGging:LPAParameter: ASAVE:FNAME?
Example	:CALC3:DLOG:LPAR:ASAV:FNAM? -> L0001.WG1,EXT<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned as &lt;filename&gt;,INT EXT. &lt;filename&gt;: File name INT: Internal memory EXT: USB memory</li> <li>This includes the file that is currently being saved with the auto save function if such file is present.</li> <li>If there are no files that have been saved with the auto save function, ",", is returned.</li> </ul>

**:CALCulate3:DLOGging:LPAParameter:****ASAVE [:STATe]**

**Function** Sets or queries whether the auto save function of data logging is on or off and the auto save destination medium.

**Syntax** :CALCulate3:DLOGging:LPAParameter:  
ASAVE [:STATe] <wsp> <mode>  
:CALCulate3:DLOGging:LPAParameter:  
ASAVE [:STATe] ?  
<mode>: OFF|INTERNAL|EXTERNAL  
OFF: Will not be saved automatically  
INTERNAL: Will be saved automatically to the internal memory  
EXTERNAL: Will be saved automatically to USB memory

**Example** :CALC3:DLOG:LPA:ASAV EXT  
:CALC3:DLOG:LPA:ASAV? -> EXT<END>

**:CALCulate3:DLOGging:LPAParameter:****INTERval**

**Function** Sets or queries the measurement interval of data logging.

**Syntax** :CALCulate3:DLOGging:LPAParameter:  
INTERval <wsp> MINimum | <NRF> [S|MS]  
:CALCulate3:DLOGging:LPAParameter:  
INTERval ?

**Example** :CALC3:DLOG:LPA:INT 5S  
:CALC3:DLOG:LPA:INT?  
-> +5.0000000E+000

**Description** • You can enter any value of your choice, but it will be set to the nearest value among 200 ms, 500 ms, 1 s, 2 s, 5 s, 10 s, 30 s, 1 m, 2 m, 5 m, 10 m. Entering 200 ms is the same as specifying the MINimum parameter.  
• Query results are returned in seconds in floating-point format. If you specify 200 ms or MINimum, "MIN" is returned.  
• This command is invalid when data logging is in progress.

**:CALCulate3:DLOGging:LPAParameter:ITEM**

**Function** Sets or queries the data logging source.

**Syntax** :CALCulate3:DLOGging:LPAParameter:  
ITEM <wsp> PEAK | FPLD  
:CALCulate3:DLOGging:LPAParameter:  
ITEM ?

PEAK: Wavelength and power of each peak will be logged.

FPLD: The result of performing an FP-LD analysis on the peak will be logged.

**Example** :CALC3:DLOG:LPA:ITEM PEAK  
:CALC3:DLOG:LPA:ITEM? -> PEAK<END>

**Description** • This command is invalid when data logging is in progress.  
• This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:DLOGging:LPAParameter:****LMODE**

**Function** Sets or queries the data logging mode (the maximum number of channels and logging count).

**Syntax** :CALCulate3:DLOGging:LPAParameter:  
LMODE <wsp> MODE1 | MODE2 | MODE3  
:CALCulate3:DLOGging:LPAParameter:  
LMODE ?  
MODE1: Up to 1024 channels will be logged for 5001 times.  
MODE2: Up to 256 channels will be logged for 20001 times.  
MODE3: Up to 64 channels will be logged for 100001 times.

**Example** :CALC3:DLOG:LPA:LMOD MODE3  
:CALC3:DLOG:LPA:LMOD? -> MODE3<END>

**Description** • This command is invalid when data logging is in progress.  
• This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:DLOGging:LPAParameter:****TDURATION**

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

**Syntax** :CALCulate3:DLOGging:LPAParameter:  
TDURATION <wsp> <integer> [S]  
:CALCulate3:DLOGging:LPAParameter:  
TDURATION  
<integer>: 1 to 8639999

**Example** :CALC3:DLOG:LPA:TDURATION 86400  
:CALC3:DLOG:LPA:TDUR? -> 86400<END>

**Description** This command is invalid when data logging is in progress.

**:CALCulate3:DLOGging:MEASure:STATE**

**Function** Starts, stops, or queries data logging.

**Syntax** :CALCulate3:DLOGging:MEASure:  
STATE <wsp> 0 | OFF | 1 | ON  
:CALCulate3:DLOGging:MEASure:STATE ?  
0 | OFF: Data logging is stopped.  
1 | ON: Data logging is started.

**Example** :CALC3:DLOG:MEAS:STAT ON  
:CALC3:DLOG:MEAS:STAT? -> 1<END>

**Description** • This command is invalid when data logging is in progress.  
• This is an overlappable command.

**:CALCulate3:DLOGging:STATE**

**Function** Sets or queries the on/off state of the logging data display on the analysis screen.

**Syntax** :CALCulate3:DLOGging:  
STATE <wsp> 0 | OFF | 1 | ON  
:CALCulate3:DLOGging:STATE ?  
0 | OFF: Logging data display is off.  
1 | ON: Logging data display is on.

**Example** :CALC3:DLOG:STAT ON  
:CALC3:DLOG:STAT? -> 1<END>

**Description** This command is invalid when data logging is in progress.



## 5.5 Device-specific Commands

### **:CALCulate3:DRIFt:DIFFerence[:STATe]**

Function	Sets or queries the on/off state of the MAX-MIN measurement of drift measurement.
Syntax	:CALCulate3:DRIFt:DIFFerence [:STATe]<wsp>0 OFF 1 ON :CALCulate3:DRIFt:DIFFerence[: STATe]? 0 OFF: The MAX-MIN value is not measured. 1 ON: The MAX-MIN value is measured.
Example	:CALC3:DRIF:DIFF ON :CALC3:DRIF:DIFF? -> 1<END>
Description	This cannot be set to ON if any of the following is already being measured: MAX, MIN, Ref, Power, or Wavelength.

### **:CALCulate3:DRIFt:ETIMe?**

Function	Queries the elapsed drift measurement time.
Syntax	:CALCulate3:DRIFt:ETIMe?
Example	:CALC3:DRIF:ETIM? -> 312<END>
Description	This command returns the elapsed time in seconds as an integer.

### **:CALCulate3:DRIFt:MAXimum[:STATe]**

Function	Sets or queries the on/off state of the MAX measurement of drift measurement.
Syntax	:CALCulate3:DRIFt:MAXimum[:STATe] <wsp>0 OFF 1 ON :CALCulate3:DRIFt:MAXimum[:STATe]? 0 OFF: The MAX value is not measured. 1 ON: The MAX value is measured.
Example	:CALC3:DRIF:MAX ON :CALC3:DRIF:MAX? -> 1<END>
Description	This cannot be set to ON if any of the following is already being measured: MAX-MIN, MIN, Ref, Power, or Wavelength.

### **:CALCulate3:DRIFt:MINimum[:STATe]**

Function	Sets or queries the on/off state of the MIN measurement of drift measurement.
Syntax	:CALCulate3:DRIFt:MINimum [:STATe]<wsp>0 OFF 1 ON :CALCulate3:DRIFt:MINimum[:STATe]? 0 OFF: The MIN value is not measured. 1 ON: The MIN value is measured.
Example	:CALC3:DRIF:MIN ON :CALC3:DRIF:MIN? -> 1<END>
Description	This cannot be set to ON if any of the following is already being measured: MAX-MIN, MAX, Ref, Power, or Wavelength.

### **:CALCulate3:DRIFt:POWer[:STATe]**

Function	Sets or queries the on/off state of the power measurement of drift measurement.
Syntax	:CALCulate3:DRIFt:POWer [:STATe]<wsp>0 OFF 1 ON :CALCulate3:DRIFt:POWer[:STATe]? 0 OFF: The POWER value is not measured. 1 ON: The POWER value is measured.
Example	:CALC3:DRIF:POW ON :CALC3:DRIF:POW? -> 1<END>
Description	This cannot be set to ON if any of the following is already being measured: MAX-MIN, MAX, MIN, Ref, or Wavelength.

### **:CALCulate3:DRIFt:WAVelength[:STATe]**

Function	Sets or queries the on/off state of the wavelength measurement of drift measurement.
Syntax	:CALCulate3:DRIFt:WAVelength[:STATe] <wsp>0 OFF 1 ON :CALCulate3:DRIFt:WAVelength [:STATe]? 0 OFF: The wavelength value is not measured. 1 ON: The wavelength value is measured.
Example	:CALC3:DRIF:WAV ON :CALC3:DRIF:WAV? -> 1<END>
Description	This cannot be set to ON if any of the following is already being measured: MAX-MIN, MAX, MIN, Ref, or Power.

### **:CALCulate3:DRIFt:PRESet**

Function	Sets the MAX-MIN, MAX, MIN, and Ref measurements of drift measurement to OFF.
Syntax	:CALCulate3:DRIFt:PRESet
Example	:CALC3:DRIF:PRES
Description	After this command is executed, the response to the :CALCulate3:DATA? command will contain the delta measurement result.

### **:CALCulate3:DRIFt[:STATe]**

Function	Sets or queries the on/off state of drift measurement.
Syntax	:CALCulate3:DRIFt[:STATe]<wsp> 0 OFF 1 ON :CALCulate3:DRIFt[:STATe]? 0 OFF: Drift measurement is disabled. 1 ON: Drift measurement is enabled.
Example	:CALC3:DRIF ON :CALC3:DRIF? -> 1<END>

**:CALCulate3:DRIFT:REfERENCE:RESet**

**Function** Re-executes drift measurement using the current detected peak as the reference.

**Syntax** :CALCulate3:DRIFT:REfERENCE:RESet

**Example** :CALC3:DRIF:REF:PRES

**Description** Drift measurement results are displayed from when this command is executed.

**:CALCulate3:DRIFT:REfERENCE[:STATe]**

**Function** Sets or queries whether Ref values will be returned in response to :CALCulate3:DATA? commands for drift measurement.

**Syntax** :CALCulate3:DRIFT:REfERENCE[:STATe] <wsp>0|OFF|1|ON  
:CALCulate3:DRIFT:REfERENCE[:STATe]?  
0|OFF: The Ref value is not returned.  
1|ON: The Ref value is returned.

**Example** :CALC3:DRIF:REF ON  
:CALC3:DRIF:REF? -> 1<END>

**:CALCulate3:FPERot[:STATe]**

**Function** Sets or queries the on/off state of FP-LD analysis.

**Syntax** :CALCulate3:FPERot[:STATe] <wsp>0|OFF|1|ON  
:CALCulate3:FPERot[:STATe]?  
0|OFF: FP-LD analysis is disabled.  
1|ON: FP-LD analysis is enabled.

**Example** :CALC3:FPER ON  
:CALC3:FPER? -> 1<END>

**Description** This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:FPERot:FWMH?**

**Function** Queries the FWHM value of FP-LD analysis.

**Syntax** :CALCulate3:FPERot:FWMH{[:WAVelength]|:FREQuency|:WNUMber}?  
WAVelength  
FREQuency  
WNUMber: Wavelength

**Example** :CALC3:FPER:FWMH?  
-> +3.12095579E-009<END>

**Description**

- Query results are returned in m for wavelength, Hz for frequency, and m<sup>-1</sup> for wavenumber.
- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:FPERot:MEAN?**

**Function** Queries the center value of FP-LD analysis.

**Syntax** :CALCulate3:FPERot:MEAN  
{[:WAVelength]|:FREQuency|:WNUMber}?  
WAVelength  
FREQuency  
WNUMber: Wavelength

**Example** :CALC3:FPER:MEAN?  
-> +1.54721566E-006<END>

**Description**

- Query results are returned in m for wavelength, Hz for frequency, and m<sup>-1</sup> for wavenumber.
- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:FPERot:MODE:SPACing?**

**Function** Queries the channel spacing of FP-LD analysis.

**Syntax** :CALCulate3:FPERot:MODE:SPACing  
{[:WAVelength]|:FREQuency|:WNUMber}?  
WAVelength  
FREQuency  
WNUMber: Wavelength

**Example** :CALC3:FPER:MODE:SPAC?  
-> +1.50681284E-009<END>

**Description**

- Query results are returned in m for wavelength, Hz for frequency, and m<sup>-1</sup> for wavenumber.
- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:FPERot:PEAK?**

**Function** Queries the peak value of FP-LD analysis.

**Syntax** :CALCulate3:FPERot:PEAK  
{[:WAVelength]|:FREQuency|:WNUMber|:POWer{[:DBM]|:WATTs}}?  
WAVelength  
FREQuency  
WNUMber: Wavelength  
POWer

**Example** :CALC3:FPER:PEAK?  
-> +1.54742260E-006<END>

**Description**

- Query results are returned in m for wavelength, Hz for frequency, and m<sup>-1</sup> for wavenumber.
- The power is returned in dBm or W.
- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

**:CALCulate3:FPERot:POWer?**

**Function** Queries the total power of FP-LD analysis.

**Syntax** :CALCulate3:FPERot:POWer  
{[:DBM]|:WATTs}?  
WAVelength  
FREQuency  
WNUMber: Wavelength

**Example** :CALC3:FPER:POW?  
-> -1.21722665E+000<END>

**Description**

- Query results are returned in dBm or W.
- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.



## 5.5 Device-specific Commands

### :CALCulate3:FPERot:SIGMa?

Function	Queries the $\sigma$ value of FP-LD analysis.
Syntax	:CALCulate3:FPERot:SIGMa {[:WAVelength] :FREQuency :WNUMber}? WAVelength FREQuency WNUMber: Wavenumber
Example	:CALC3:FPER:SIGM? -> +1.32524662E-009<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m for wavelength, Hz for frequency, and <math>m^{-1}</math> for wavenumber.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :CALCulate3:GRID:DISPlay:ALL

Function	Sets or queries whether all grids are to be displayed.
Syntax	:CALCulate3:GRID:DISPlay:ALL<wsp>0 OFF 1 ON :CALCulate3:GRID:DISPlay:ALL? 0 OFF: Only grids with peaks are displayed. 1 ON: All grids are displayed.
Example	:CALC3:GRID:DISP:ALL ON :CALC3:GRID:DISP:ALL? -> 1<END>
Description	This command is valid on the -MW (multi-wavelength) suffix code type.

### :CALCulate3:GRID:REFeRence:FREQuency

Function	Sets or queries the reference frequency of grid display.
Syntax	:CALCulate3:GRID:REFeRence: FREQuency<wsp><freq> :CALCulate3:GRID:REFeRence: FREQuency? <freq>: Fundamental frequency (186 THz to 202 THz) DEFAult <NRf> DEFAult: 193.1THz
Example	:CALC3:GRID:REF:FREQ 195THZ :CALC3:GRID:REF:FREQ? -> +1.95000000E+014<END>

### :CALCulate3:GRID:STARt[:WAVelength]

Function	Sets or queries the grid's start wavelength.
Syntax	:CALCulate3:GRID:STARt [:WAVelength]<wsp><wavelength> :CALCulate3:GRID:STARt [:WAVelength]? <wavelength>: Start wavelength in unit of m <NRf>
Example	:CALC3:GRID:STAR 1500NM :CALC3:GRID:STAR? -> +1.50000000E-006<END>

### :CALCulate3:GRID:STARt:FREQuency

Function	Sets or queries the grid's start frequency.
Syntax	:CALCulate3:GRID:STARt:FREQuency<wsp><freq> :CALCulate3:GRID:STARt:FREQuency? <freq>: Start frequency in unit of Hz <NRf>
Example	:CALC3:GRID:STAR:FREQ 191THZ :CALC3:GRID:STAR:FREQ? -> +1.91000000E+014<END>

### :CALCulate3:GRID:STARt:WNUMber

Function	Sets or queries the grid's start wavenumber.
Syntax	:CALCulate3:GRID:STARt:WNUMber<wsp> <wnumber> :CALCulate3:GRID:STARt:WNUMber? <wnumber>: Start wavelength in unit of $m^{-1}$ <NRf>
Example	:CALC3:GRID:STAR:WNUM 640000 :CALC3:GRID:STAR:WNUM? -> +6.40000000E+005

### :CALCulate3:GRID[:STATe]

Function	Sets or queries the grid display on/off state.
Syntax	:CALCulate3:GRID[:STATe]<wsp>0 OFF 1 ON :CALCulate3:GRID[:STATe]? 0 OFF: Grid display is disabled. 1 ON: Grid display is enabled.
Example	:CALC3:GRID ON :CALC3:GRID? -> 1<END>

### :CALCulate3:GRID:STOP[:WAVelength]

Function	Sets or queries the grid's stop wavelength.
Syntax	:CALCulate3:GRID:STOP[:WAVelength] <wsp><wavelength> :CALCulate3:GRID:STOP[:WAVelength]? <wavelength>: Stop wavelength in unit of m <NRf>
Example	:CALC3:GRID:STOP 1500NM :CALC3:GRID:STOP? -> +1.50000000E-006<END>

### :CALCulate3:GRID:STOP:FREQuency

Function	Sets or queries the grid's stop frequency.
Syntax	:CALCulate3:GRID:STOP:FREQuency <wsp><freq> :CALCulate3:GRID:STOP:FREQuency? <freq>: Stop frequency in unit of Hz <NRf>
Example	:CALC3:GRID:STOP:FREQ 195THZ :CALC3:GRID:STOP:FREQ? -> +1.95000000E+014<END>

**:CALCulate3:GRID:STOP:WNUMber**

Function Sets or queries the grid's stop wavenumber.

Syntax :CALCulate3:GRID:STOP:WNUMber<wsp>  
<wnumber>  
:CALCulate3:GRID:STOP:WNUMber?  
<wnumber>: Stop wavelength in unit of  $\text{m}^{-1}$   
<NRf>

Example :CALC3:GRID:STOP:WNUM 640000  
:CALC3:GRID:STOP:WNUM?  
-> +6.400000000E+005<END>

**:CALCulate3:GRID:SPACing:FREQuency**

Function Sets or queries the grid spacing.

Syntax :CALCulate3:GRID:SPACing:FREQuency  
<wsp><freq>  
:CALCulate3:GRID:SPACing:FREQuency?  
<freq>: Start frequency in unit of Hz (5 G to 1000 GHz)  
<NRf>

Example :CALC3:GRID:SPAC:FREQ 100GHZ  
:CALC3:GRID:SPAC:FREQ?  
-> +1.000000000E+011<END>

**:CALCulate3:GRID:SARea:FREQuency**

Function Sets or queries the range to search for peaks.

Syntax :CALCulate3:GRID:SARea:FREQuency  
<wsp><freq>  
:CALCulate3:GRID:SARea:FREQuency?  
<freq>: Search range in unit of Hz (1 G to 100 GHz)  
<NRf>

Example :CALC3:GRID:SAR:FREQ 1GHZ  
:CALC3:GRID:SAR:FREQ?  
-> +1.000000000E+009<END>

Description A value greater than the grid spacing can not be set.

**:CALCulate3:POINTs?**

Function Queries the number of data points in the response to a :CALCulate3:DATA? command.

Syntax :CALCulate3:POINTs?

Example :CALC3:POIN? -> +4<END>

Description

- The maximum number of response data points is 1024.
- If the mode is not delta measurement or drift measurement, 0 is returned.
- This is an overlapping command.

**:CALCulate3:PRESet**

Function Sets delta measurement, drift measurement, FP-LD analysis, WDM analysis, and Grid analysis to OFF.

Syntax :CALCulate3:PRESet

Example :CALC3:PRESet

**:CALCulate3:SNR:AUTO**

Function Sets or queries the SNR noise detection method.

Syntax :CALCulate3:SNR:AUTO<wsp>0|OFF|1|ON  
:CALCulate3:SNR:AUTO?  
0|OFF: The noise level is calculated in MANUAL-FIX mode.  
1|ON: The noise level is calculated in AUTO-CENTER mode.

Example :CALC3:SNR:AUTO ON  
:CALC3:SNR:AUTO? -> 1<END>

**:CALCulate3:SNR:REference[:WAVelengt h]:RELative**

Function Sets or queries the noise measurement point when the noise detection method is set to MANUAL-FIX.

Syntax :CALCulate3:SNR:REference  
[:WAVelength]:RELative<wsp><ref>  
:CALCulate3:SNR:REference  
[:WAVelength]:RELative?  
<ref>: Wavelength in unit of <nrf>m

Example :CALC3:SNR:REF:REL 10nm  
:CALC3:SNR:REF:REL?  
> +1.000000000E-008<END>

Description

- Set the measurement point as a relative wavelength from the peak.

**:CALCulate3:SNR:REference:BWIDth**

Function Sets or queries the noise calculation bandwidth.

Syntax :CALCulate3:SNR:REference:BWIDth<wsp><band>  
:CALCulate3:SNR:REference:BWIDth?  
<ref>: Wavelength in unit of m  
<NRf>

Example :CALC3:SNR:REF:BWID 0.1nm  
:CALC3:SNR:REF:BWID?  
> +1.000000000E-010<END>

**:CALCulate3:SNR[:STATe]**

Function Sets or queries the on/off state of OSNR analysis.

Syntax :CALCulate3:SNR[:STATe]<wsp>0|OFF|1|ON  
:CALCulate3:SNR[:STATe]?  
0|OFF: OSNR analysis is disabled.  
1|ON: OSNR analysis is enabled.

Example :CALC3:SNR ON  
:CALC3:SNR? -> 1<END>

## 5.5 Device-specific Commands

### CONFigure Sub System Commands

#### Overview

- This subsystem is a function for setting or querying how the peak detection results of the instrument are displayed. The commands in this subsystem change the view mode of the instrument, just like the panel keys change the view. The displayed contents will change as a result of these commands.
- Executing a CONFigure[:SCALar] command changes the instrument display to single view.
- Executing a CONFigure:ARRay command changes the instrument display to multi view. This command is valid on the -MW (multi-wavelength) suffix code type.

#### :CONFigure?

Function	Queries the current display settings.
Syntax	:CONFigure?
Example	:CONF? -> "ARR:POW DEF,DEF"<END>
Description	<ul style="list-style-type: none"><li>• The display conditions specified by CONFigure commands are returned in the command syntax. (single view/multi view): (wavelength/frequency/wavenumber)&lt;value&gt;,(resolution) single view: POW multi view: ARR:POW Wavelength: WAV Frequency: FREQ Wavenumber: WNUM Value: A peak (floating-point number) closest to the maximum (MAX) minimum (MIN) current (DEF) specified value Resolution: current value (DEF)</li><li>• This is an overlapping command.</li><li>• Multi display is valid on the -MW (multi-wavelength) suffix code type.</li></ul>

#### :CONFigure[:SCALar]:POWER

Function	Sets the peak for the screen in which View Mode is set to single view by specifying the power.
Syntax	:CONFigure[:SCALar]:POWER<wsp> [<expected_value>] <expected_value> (power specifying the peak): MAXimum MINimum DEFault <NRf> MAXimum: Maximum power peak MINimum: Minimum power peak DEFault: Selected peak <NRf>: Peak closest to the specified power
Example	:CONF:POW -4dbm
Description	<ul style="list-style-type: none"><li>• If the parameter is omitted, DEF will be set.</li><li>• If the parameter is not set to &lt;NRf&gt; or DEF, the auto peak search function will be set to OFF.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

#### :CONFigure[:SCALar]:POWER:FREQUENCY

Function	Sets the peak for the screen in which View Mode is set to single view by specifying the frequency.
Syntax	:CONFigure[:SCALar]:POWER:FREQUENCY <wsp>[<expected_value>] <expected_value> (frequency specifying the peak): MAXimum MINimum DEFault <NRf> MAXimum: Maximum frequency peak MINimum: Minimum frequency peak DEFault: Selected peak <NRf>: Peak closest to the specified frequency
Example	:CONF:POW:FREQ 193.6THZ
Description	<ul style="list-style-type: none"><li>• If the parameter is omitted, DEF will be set.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

#### :CONFigure[:SCALar]:POWER:WAVELENGTH

Function	Sets the peak for the screen in which View Mode is set to single view by specifying the wavelength.
Syntax	:CONFigure[:SCALar]:POWER:WAVELENGTH <wsp>[<expected_value>] <expected_value> (wavelength specifying the peak): MAXimum MINimum DEFault <NRf> MAXimum: Maximum wavelength peak MINimum: Minimum wavelength peak DEFault: Selected peak <NRf>: Peak closest to the specified wavelength
Example	:CONF:POW:WAV 1547.4nm
Description	<ul style="list-style-type: none"><li>• If the parameter is omitted, DEF will be set.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

#### :CONFigure[:SCALar]:POWER:WNUMBER

Function	Sets the peak for the screen in which View Mode is set to single view by specifying the wavenumber.
Syntax	:CONFigure[:SCALar]:POWER:WNUMBER <wsp>[<expected_value>] <expected_value> (wavenumber specifying the peak): MAXimum MINimum DEFault <NRf> MAXimum: Maximum wavenumber peak MINimum: Minimum wavenumber peak DEFault: Selected peak <NRf>: Peak closest to the specified wavenumber
Example	:CONF:POW:WNUM 646710
Description	<ul style="list-style-type: none"><li>• If the parameter is omitted, DEF will be set.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

**:CONFigure:ARRay:POWer**

Function	Sets the peak for the screen in which View Mode is set to multi view by specifying the power.
Syntax	<pre>:CONFigure:ARRay:POWer&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (power specifying the peak):     MAXimum MINimum DEFAULT &lt;NRf&gt; MAXimum: Maximum power peak MINimum: Minimum power peak DEFAULT: Selected peak &lt;NRf&gt;: Peak closest to the specified power</pre>
Example	<code>:CONF:ARR:POW -4DBM</code>
Description	<ul style="list-style-type: none"> <li>• If the parameter is omitted, DEF will be set.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:CONFigure:ARRay:POWer:FREQuency**

Function	Sets the peak for the screen in which View Mode is set to multi view by specifying the frequency.
Syntax	<pre>:CONFigure:ARRay:POWer:FREQuency &lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (frequency specifying the peak):     MAXimum MINimum DEFAULT &lt;NRf&gt; MAXimum: Maximum frequency peak MINimum: Minimum frequency peak DEFAULT: Selected peak &lt;NRf&gt;: Peak closest to the specified frequency</pre>
Example	<code>:CONF:ARR:POW:FREQ 193.6THZ</code>
Description	<ul style="list-style-type: none"> <li>• If the parameter is omitted, DEF will be set.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:CONFigure:ARRay:POWer:WAVelength**

Function	Sets the peak for the screen in which View Mode is set to multi view by specifying the wavelength.
Syntax	<pre>:CONFigure:ARRay:POWer:WAVelength &lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (wavelength specifying the peak):     MAXimum MINimum DEFAULT &lt;NRf&gt; MAXimum: Maximum wavelength peak MINimum: Minimum wavelength peak DEFAULT: Selected peak &lt;NRf&gt;: Peak closest to the specified wavelength</pre>
Example	<code>:CONF:ARR:POW:WAV 1548.5NM</code>
Description	<ul style="list-style-type: none"> <li>• If the parameter is omitted, DEF will be set.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:CONFigure:ARRay:POWer:WNUMber**

Function	Sets the peak for the screen in which View Mode is set to multi view by specifying the wavenumber.
Syntax	<pre>:CONFigure:ARRay:POWer:WNUMber&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (wavenumber specifying the peak):     MAXimum MINimum DEFAULT &lt;NRf&gt; MAXimum: Maximum wavenumber peak MINimum: Minimum wavenumber peak DEFAULT: Selected peak &lt;NRf&gt;: Peak closest to the specified wavenumber</pre>
Example	<code>:CONF:ARR:POW:WNUM 645760</code>
Description	<ul style="list-style-type: none"> <li>• If the parameter is omitted, DEF will be set.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

## 5.5 Device-specific Commands

### DISPlay Sub System Commands

#### :DISPlay:COLOr

Function Sets or queries the display color.

Syntax :DISPlay:COLOr<wsp>0|1  
:DISPlay:COLOr?  
0: Black and white  
1: Color

Example :DISP:COL 1  
:DISP:COL? -> 1<END>

Description This is an overlapping command.

#### :DISPlay[:WINDow]

Function Sets the on/off state of the display.

Syntax :DISPlay[:WINDow]<wsp>OFF|0|ON|1  
:DISPlay[:WINDow]?  
0|OFF: Display off  
1|ON: Display on

Example :DISP OFF  
:DISP? -> 0<END>

Description This is an overlapping command.

#### :DISPlay:MARKer:MAXimum

Function Sets the current peak to the maximum power peak.

Syntax :DISPlay:MARKer:MAXimum

Example :DISP:MARK:MAX

Description

- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.


#### :DISPlay:MARKer:MAXimum:LEFT

Function Moves the current peak to the left adjacent peak.

Syntax :DISPlay:MARKer:MAXimum:LEFT

Example :DISP:MARK:MAX:LEFT

Description

- This is an overlapping command.
-  This command is valid on the -MW (multi-wavelength) suffix code type.

#### :DISPlay:MARKer:MAXimum:NEXT

Function Moves the current peak to the peak with the next lower power.

Syntax :DISPlay:MARKer:MAXimum:NEXT

Example :DISP:MARK:MAX:NEXT

Description

- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

#### :DISPlay:MARKer:MAXimum:PREVIOUS

Function Moves the current peak to the peak with the next higher power.

Syntax :DISPlay:MARKer:MAXimum:PREVIOUS

Example :DISP:MARK:MAX:PREV

Description

- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

#### :DISPlay:MARKer:MAXimum:RIGHT

Function Moves the current peak to the right adjacent peak.

Syntax :DISPlay:MARKer:MAXimum:RIGHT

Example :DISP:MARK:MAX:RIGH

Description

- This is an overlapping command.
- This command is valid on the -MW (multi-wavelength) suffix code type.

#### :DISPlay:RESolution[:WAVelength]

Function Sets or queries the number of decimal places for the wavelengths to be displayed on the screen.

Syntax :DISPlay:RESolution[:WAVelength]  
<wsp>R0.0001|R0.001|R0.01|R0.1|  
MAXimum|MINimum  
:DISPlay:RESolution[:WAVelength]?  
R0.0001: 4 decimal places  
R0.001: 3 decimal places  
R0.01: 2 decimal places  
R0.1: 1 decimal place

MAXimum: 4 decimal places of the maximum value

MINimum: 1 decimal place of the minimum value

Example :DISP:RES R0.01  
:DISP:RES? -> R0.01<END>

Description This is an overlap command.

#### :DISPlay:RESolution:FREQuency

Function Sets or queries the number of decimal places for the frequencies to be displayed on the screen.

Syntax :DISPlay:RESolution:FREQuency  
<wsp>R0.00001|R0.0001|R0.001|R0.01|  
MAXimum|MINimum  
:DISPlay:RESolution:FREQuency?

R0.00001: 5 decimal place

R0.0001: 4 decimal places

R0.001: 3 decimal places

R0.01: 2 decimal places

MAXimum: 5 decimal places of the maximum value

MINimum: 2 decimal place of the minimum value

Example :DISP:RES:FREQ R0.01  
:DISP:RES:FREQ? -> R0.01<END>

Description This is an overlap command.

**:DISPlay:RESolution:WNUMber**

Function	Sets or queries the number of decimal places for the wavenumbers to be displayed on the screen.
Syntax	:DISPlay:RESolution::WNUMber<wsp> R0.001 R0.01 R0.1 R1MAXimum MINimum :DISPlay:RESolution::WNUMber? R0.001: 3 decimal places R0.01: 2 decimal places R0.1: 1 decimal place R1: Integer MAXimum: 3 decimal places of the maximum value MINimum: Integer of the minimum value
Example	:DISP:RES:WNUM R0.01 :DISP:RES:WNUM? -> R0.01<END>
Description	This is an overlap command.

**:DISPlay:UNIT:WAVelength**

Function	Sets or queries the wavelength unit.
Syntax	:DISPlay:UNIT:WAVelength <wsp>NM THZ ICM :DISPlay:UNIT:WAVelength? NM: Wavelength (nm) THZ: Frequency (THz) ICM: Wavenumber (cm <sup>-1</sup> )
Example	:DISP:UNIT:WAV NM :DISP:UNIT:WAV? -> NM<END>
Description	This is an overlapping command.

**:DISPlay[:WINDow]:TEXT:DATA**

Function	Sets or queries the label text on the display.
Syntax	:DISPlay[:WINDow]:TEXT: DATA<wsp><"string"> :DISPlay[:WINDow]:TEXT:DATA? <"string">: Label string (up to 52 characters excluding the double quotation marks)
Example	:DISP:TEXT:DATA "AQ6150B Optical Wavelength Meter" :DISP:TEXT:DATA? -> AQ6150B Optical Wavelength Meter<END>
Description	This is an overlapping command.

**:DISPlay[:WINDow]:STATE**

Function	Sets or queries the on/off state of the multi wavelength display window.
Syntax	:DISPlay[:WINDow]:STATE<wsp>0 OFF 1 ON :DISPlay[:WINDow]:STATE? 0 OFF: Multi wavelength display window off 1 ON: Multi wavelength display window on
Example	:DISP:STAT ON :DISP:STAT? -> 1<END>
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:STATE**

Function	Sets or queries the on/off state of the spectrum window.
Syntax	:DISPlay:WINDow2: STATE<wsp>0 OFF 1 ON :DISPlay:WINDow2:STATE? 0 OFF: Spectrum window off 1 ON: Spectrum window on
Example	:DISP:WIND2:STAT ON :DISP:WIND2:STAT? -> 1<END>
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:TRACe[:SCALE]:**

Function	Executes a single measurement and then auto scaling.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE] :AUTOmeasure
Example	:DISP:WIND2:TRAC:AUTO
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:TRACe[:SCALE]:AScale**

Function	Optimizes (auto scales) the waveform in the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]: AScale
Example	:DISP:WIND2:TRAC:ASC
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

## 5.5 Device-specific Commands

### **:DISPlay:WINDow2:TRACe[:SCALE]:**

#### **INITialize**

Function	Initializes the horizontal scale (frequency, wavelength, and wavenumber) of the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]:INITialize
Example	:DISP:WIND2:TRAC:INIT
Description	<ul style="list-style-type: none"> <li>The left edge of the scale is set to the start wavelength; the right edge is set to the end wavelength.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### **:DISPlay:WINDow2:TRACe[:SCALE]:**

#### **LEFT[:WAVelength]**

Function	Sets or queries the start wavelength of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]:LEFT[:WAVelength]<wsp><wavelength> :DISPlay:WINDow2:TRACe[:SCALE]:LEFT[:WAVelength]? <wavelength> (start wavelength): <NRf> MINimum MAXimum MINimum: 1270 nm MINimum: 1270 nm (-10 suffix code type) 1200 nm (-20 suffix code type) 900 nm (-30 suffix code type) MAXimum: Stop wavelength - 1 nm
Example	:DISP:WIND2:TRAC:LEFT 1550NM :DISP:WIND2:TRAC:LEFT? -> +1.55000000E-006<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### **:DISPlay:WINDow2:TRACe[:SCALE]:LEFT:**

#### **FREQuency**

Function	Sets or queries the start frequency of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]:LEFT:FREQuency<wsp><frequency> :DISPlay:WINDow2:TRACe[:SCALE]:LEFT:FREQuency? <frequency> (start frequency): <NRf> MINimum MAXimum MINimum: 181.69 THz (-10 suffix code type) 176.35 THz (-20 suffix code type) 176.35 THz (-30 suffix code type) MAXimum: Stop frequency - 0.1 THz
Example	:DISP:WIND2:TRAC:LEFT:FREQ 190THZ :DISP:WIND2:TRAC:LEFT:FREQ? -> +1.90000000E+014<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in Hz.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### **:DISPlay:WINDow2:TRACe[:SCALE]:LEFT:**

#### **WNUMber**

Function	Sets or queries the start wavenumber of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]:LEFT:WNUMber<wsp><wnumber> :DISPlay:WINDow2:TRACe[:SCALE]:LEFT:WNUMber? <wnumber> (start wavenumber): <NRf> MINimum MAXimum MINimum: 6060.0 cm <sup>-1</sup> (-10 suffix code type) 5882.4 cm <sup>-1</sup> (-20 suffix code type) 5882.4 cm <sup>-1</sup> (-30 suffix code type) MAXimum: Stop wavenumber - 1 cm <sup>-1</sup>
Example	:DISP:WIND2:TRAC:LEFT:WNUM 609000 :DISP:WIND2:TRAC:LEFT:WNUM? -> +6.09000000E+004<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m<sup>-1</sup>.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>



**:DISPlay:WINDow2:TRACe[:SCALE]:  
RIGHT[:WAVelength]**

Function	Sets or queries the stop wavelength of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]: RIGHT[:WAVelength]<wsp><wavelength> :DISPlay:WINDow2:TRACe[:SCALE]: RIGHT[:WAVelength]? <wavelength> (stop wavelength): <NRf> MINimum MAXimum MINimum: Start wavelength + 1 nm MAXimum: 1650 nm (-10 suffix code type) 1700 nm (-20 suffix code type) 1700 nm (-30 suffix code type)
Example	:DISP:WIND2:TRAC:RIGH 1600NM :DISP:WIND2:TRAC:RIGH? -> +1.60000000E-006<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:TRACe[:SCALE]:  
RIGHT:FREQuency**

Function	Sets or queries the stop frequency of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]: RIGHT:FREQuency<wsp><frequency> :DISPlay:WINDow2:TRACe[:SCALE]: RIGHT:FREQuency? <frequency> (stop frequency): <NRf> MINimum MAXimum MINimum: Start frequency + 0.1 THz MAXimum: 230.06 THz (-10 suffix code type) 249.83 THz (-20 suffix code type) 333.11 THz (-30 suffix code type)
Example	:DISP:WIND2:TRAC:RIGH:FREQ 190THZ :DISP:WIND2:TRAC:RIGH:FREQ? -> +1.90000000E+014<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in Hz.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:TRACe[:SCALE]:  
RIGHT:WNUmber**

Function	Sets or queries the stop wavenumber of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]: RIGHT:WNUmber<wsp><wnumber> :DISPlay:WINDow2:TRACe[:SCALE]: RIGHT:WNUmber? <wnumber> (stop wavenumber): <NRf> MINimum MAXimum MINimum: Start wavenumber MAXimum: 7875.0 cm <sup>-1</sup> (-10 suffix code type) 8333.3 cm <sup>-1</sup> (-20 suffix code type) 11111.1 cm <sup>-1</sup> (-30 suffix code type)
Example	:DISP:WIND2:TRAC:RIGH:WNUM 609000 :DISP:WIND2:TRAC:RIGH:WNUM? -> +6.09000000E+005<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m<sup>-1</sup>.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:TRACe[:SCALE]:  
CENTER[:WAVelength]**

Function	Sets or queries the center wavelength of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]: CENTER[:WAVelength]<wsp><wavelength> :DISPlay:WINDow2:TRACe[:SCALE]: CENTER[:WAVelength]? <wavelength> (center wavelength): <NRf>
Example	:DISP:WIND2:TRAC:CEN 1550NM :DISP:WIND2:TRAC:CEN? -> +1.55000000E-006<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:DISPlay:WINDow2:TRACe[:SCALE]:  
CENTER:FREQuency**

Function	Sets or queries the center frequency of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALE]: CENTER:FREQuency<wsp><frequency> :DISPlay:WINDow2:TRACe[:SCALE]: CENTER:FREQuency? <frequency> (center frequency): <NRf>
Example	:DISP:WIND2:TRAC:CEN:FREQ 190THZ :DISP:WIND2:TRAC:CEN:FREQ? -> +1.90000000E+014<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in Hz.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>



## 5.5 Device-specific Commands

### :DISPlay:WINDow2:TRACe[:SCALe]:

#### CENTer:WNUMber

Function	Sets or queries the center wavenumber of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALe]: CENTer:WNUMber<wsp><wnumber> :DISPlay:WINDow2:TRACe[:SCALe]: CENTer:WNUMber? <wnumber> (center wavenumber): <NRf>
Example	:DISP:WIND2:TRAC:CEN:WNUM 609000 :DISP:WIND2:TRAC:CEN:WNUM? -> +6.09000000E+005<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in <math>\text{m}^{-1}</math>.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :DISPlay:WINDow2:TRACe[:SCALe]:

#### CENTer:PEAK

Function	Displays the current peak in the center of the horizontal scale.
Syntax	:DISPlay:WINDow2:TRACe[:SCALe]: CENTer:PEAK
Example	:DISP:WIND2:TRAC:CEN:PEAK
Description	<ul style="list-style-type: none"> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :DISPlay:WINDow2:TRACe[:SCALe]:

#### SPAN[:WAVelength]

Function	Sets or queries the display span wavelength of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALe]: SPAN[:WAVelength]<wsp><wavelength> :DISPlay:WINDow2:TRACe[:SCALe]: SPAN[:WAVelength]? <wavelength> (span wavelength): <NRf> MAXimum MAXimum: 380 nm (-10 suffix code type) 500 nm (-20 suffix code type) 800 nm (-30 suffix code type)
Example	:DISP:WIND2:TRAC:SPAN 50NM :DISP:WIND2:TRAC:SPAN? -> +5.00000000E-008<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in m.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :DISPlay:WINDow2:TRACe[:SCALe]:SPAN:

#### FREQuency

Function	Sets or queries the display span frequency of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALe]: SPAN:FREQuency<wsp><frequency> :DISPlay:WINDow2:TRACe[:SCALe]: SPAN:FREQuency? <frequency> (span frequency): <NRf> MAXimum MAXimum: 48.37 THz (-10 suffix code type) 73.48 THz (-20 suffix code type) 156.76 THz (-30 suffix code type)
Example	:DISP:WIND2:TRAC:SPAN:FREQ 20THZ :DISP:WIND2:TRAC:SPAN:FREQ? -> +2.00000000E+014<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in Hz.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :DISPlay:WINDow2:TRACe[:SCALe]:SPAN:

#### WNUMber

Function	Sets or queries the display span wavenumber of the horizontal scale on the spectrum window.
Syntax	:DISPlay:WINDow2:TRACe[:SCALe]: SPAN:WNUMber<wsp><wnumber> :DISPlay:WINDow2:TRACe[:SCALe]: SPAN:WNUMber? <wnumber> (span wavenumber): <NRf> MAXimum MAXimum: 1815 $\text{cm}^{-1}$ (-10 suffix code type) 2450.9 $\text{cm}^{-1}$ (-20 suffix code type) 5228.7 $\text{cm}^{-1}$ (-30 suffix code type)
Example	:DISP:WIND2:TRAC:SPAN:WNUM 10000 :DISP:WIND2:TRAC:SPAN:WNUM? -> +1.00000000E+003<END>
Description	<ul style="list-style-type: none"> <li>Query results are returned in <math>\text{m}^{-1}</math>.</li> <li>This is an overlapping command.</li> <li>This command is valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

## FETCh Sub System Commands

## Overview

- This subsystem is a function for querying the most recent measured values.  
If a query is made during a measurement, the result is returned after the measurement is complete. For details, see "Example of Operation Status Change" in section 4.4.
- The commands do not affect the instrument operation. (Related commands: MEAS Sub System, READ Sub System)

## :FETCh?

Function	Queries the most recent peak measurement result.
Syntax	:FETC?
Example	:FETC? -> 3,+6.46241320E+005, +6.45768650E+005,+6.46714090E+005 <END>
Description	<ul style="list-style-type: none"> <li>• If the previous query command was in single view mode (a command that contains ":SCALar" in its command string), one measurement result is returned.</li> <li>• If the previous query command was in multi view mode (a command that contains ":ARRay" in its command string), measurement results are returned for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. Power &lt;peak_num&gt;,&lt;power1&gt;,&lt;power2&gt;,... Wavelength &lt;peak_num&gt;,&lt;wav1&gt;,&lt;wav2&gt;,... Frequency &lt;peak_num&gt;,&lt;freq1&gt;,&lt;freq2&gt;,... Wavenumber &lt;peak_num&gt;,&lt;wnum1&gt;,&lt;wnum2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;power1&gt;,&lt;power2&gt;,...: Peak power &lt;wav1&gt;,&lt;wav2&gt;,...: Peak wavelengths &lt;freq1&gt;,&lt;freq2&gt;,...: Peak frequencies &lt;wnum1&gt;,&lt;wnum2&gt;,...: Peak numbers</li> <li>• After power-on, the instrument returns wavelength values (in unit of meters).</li> <li>• This is an overlapping command.</li> </ul>

## :FETCh:ARRay:POWer?

Function	Queries the most recent peak's power values in multi view mode.
Syntax	:FETCh:ARRay:POWer?<wsp> [<expected_value>] <expected_value> (power): MAXimum MINimum DEFault <NRf> MAXimum: Specifies the maximum power peak MINimum: Specifies the minimum power peak DEFault: Specifies the selected peak <NRf>: Specifies the peak closest to the specified power
Example	:FETC:ARR:POW? -> ,-3.99000000E+000, -7.28000000E+000,-1.08300000E+001 <END>
Description	<ul style="list-style-type: none"> <li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;power1&gt;,&lt;power2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;power1&gt;,&lt;power2&gt;,... Peak power values</li> <li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li> <li>• If the parameter is not set to &lt;NRf&gt; or DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in dBm or W depending on the parameter.</li> <li>• This is an overlapping command.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

## :FETCh:ARRay:POWer:FREQuency?

Function	Queries the most recent peak's frequency values in multi view mode.
Syntax	:FETCh:ARRay:POWer:FREQuency?<wsp> [<expected_value>] <expected_value> (frequency): MAXimum MINimum DEFault <NRf> MAXimum: Specifies the maximum frequency peak MINimum: Specifies the minimum frequency peak DEFault: Specifies the selected peak <NRf>: Specifies the peak closest to the specified frequency
Example	:FETC:ARR:POW:FREQ? -> 3, +1.93738272E+014,+1.93596570E+014, +1.93880006E+014<END>
Description	<ul style="list-style-type: none"> <li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;freq1&gt;,&lt;freq2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;freq1&gt;,&lt;freq2&gt;,... Peak frequencies</li> </ul>

## 5.5 Device-specific Commands

- If you specify a parameter, the current (selected) peak on the instrument screen will change.
- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in Hz.
- This is an overlapping command.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

### :FETCh:ARRay:POWer:WAVeLength?

Function	Queries the most recent peak's wavelength values in multi view mode.
Syntax	:FETCh:ARRay:POWer:WAVeLength?<wsp>[<expected_value>] <expected_value> (wavelength): MAXimum MINimum DEFault <NRf> MAXimum: Specifies the maximum wavelength peak MINimum: Specifies the minimum wavelength peak DEFault: Specifies the selected peak <NRf>: Specifies the peak closest to the specified wavelength
Example	:FETC:ARR:POW:WAV? -> 3, +1.54740958E-006,+1.54854220E-006, +1.54627836E-006<END>
Description	<ul style="list-style-type: none"> <li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;wav1&gt;,&lt;wav2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;wav1&gt;,&lt;wav2&gt;,... Peak wavelengths</li> <li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in m.</li> <li>• This is an overlapping command.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :FETCh:ARRay:POWer:WNUMber?

Function	Queries the most recent peak's wavenumber values in multi view mode.
Syntax	:FETCh:ARRay:POWer:WNUMber?<wsp>[<expected_value>] <expected_value> (wavenumber): MAXimum MINimum DEFault <NRf> MAXimum: Specifies the maximum wavenumber peak MINimum: Specifies the minimum wavenumber peak DEFault: Specifies the selected peak <NRf>: Specifies the peak closest to the specified wavenumber
Example	:FETC:ARR:POW:WNUM? -> 3, +6.46241320E+005,+6.45768650E+005, +6.46714090E+005<END>
Description	<ul style="list-style-type: none"> <li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;wnum1&gt;,&lt;wnum2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;wnum1&gt;,&lt;wnum2&gt;,... Peak wavenumbers</li> <li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in m<sup>-1</sup>.</li> <li>• This is an overlapping command.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

### :FETCh[:SCALar]:POWer?

Function	Queries the most recent peak's power value in single view mode.
Syntax	:FETCh[:SCALar]:POWer?<wsp>[<expected_value>] <expected_value> (power): MAXimum MINimum DEFault <NRf> MAXimum: Specifies the maximum power peak MINimum: Specifies the minimum power peak DEFault: Specifies the selected peak <NRf>: Specifies the peak closest to the specified power
Example	:FETC:POW? -> -7.28000000E+000<END>
Description	<ul style="list-style-type: none"> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in dBm or W depending on the parameter.</li> <li>• This is an overlapping command.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:FETCh[:SCALar]:POWer:FREQuency?**

**Function** Queries the most recent peak's frequency value in single view mode.

**Syntax** :FETCh[:SCALar]:POWer:FREQuency?<wsp>  
[<expected\_value>]  
<expected\_value> (frequency):  
MAXimum|MINimum|DEFAult|<NRf>  
MAXimum: Specifies the maximum frequency peak  
MINimum: Specifies the minimum frequency peak  
DEFAult: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified frequency

**Example** :FETC:POW:FREQ?  
-> +1.93596570E+014<END>

**Description**

- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in Hz.
- This is an overlapping command.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**:FETCh[:SCALar]:POWer:WAVelength?**

**Function** Queries the most recent peak's wavelength value in single view mode.

**Syntax** :FETCh[:SCALar]:POWer:WAVelength?  
<wsp>[<expected\_value>]  
<expected\_value> (wavelength):  
MAXimum|MINimum|DEFAult|<NRf>  
MAXimum: Specifies the maximum wavelength peak  
MINimum: Specifies the minimum wavelength peak  
DEFAult: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified wavelength

**Example** :FETC:POW:WAV?  
-> +1.54854220E-006<END>

**Description**

- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in m.
- This is an overlapping command.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**:FETCh[:SCALar]:POWer:WNUMber?**

**Function** Queries the most recent peak's wavenumber in single view mode.

**Syntax** :FETCh[:SCALar]:POWer:WNUMber?<wsp>  
[<expected\_value>]  
<expected\_value> (wavenumber):  
MAXimum|MINimum|DEFAult|<NRf>  
MAXimum: Specifies the maximum wavenumber peak  
MINimum: Specifies the minimum wavenumber peak  
DEFAult: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified wavenumber

**Example** :FETC:POW:WNUM?  
-> +6.45768650E+005<END>

**Description**

- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in m<sup>-1</sup>.
- This is an overlapping command.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**FORMat Sub System Commands****:FORMat:NDATa[:WAVelength]**

**Function** Sets or queries the response value when a peak is not detected.

**Syntax** :FORMat:NDATa[:WAVelength]<wsp>  
<wavelength>  
:FORMat:NDATa[:WAVelength]?  
<wavelength>: 0 to 300 nm  
<NRf>

**Example** :FORM:NDAT 100NM  
:FORM:NDAT? -> +1.00000000E-007<END>

**Description**

- This is valid for the following commands.  
:FETC:SCAL:{FREQ|WAV|WNUM}?  
:MEAS:SCAL:{FREQ|WAV|WNUM}?  
:READ:SCAL:{FREQ|WAV|WNUM}?

## 5.5 Device-specific Commands

### MEASure Sub System Commands

#### Overview

- This subsystem is a function for starting a measurement on the instrument and querying the measured results when the measurement is complete.  
The commands in this subsystem change the view mode of the instrument, just like the panel keys change the view. The displayed contents will change as a result of these commands.
- If the measurement on the instrument is stopped, the instrument will execute a single measurement and return the measured results.
- If the instrument is measuring (repeat measurement), it will return a execution error.
- If averaged measurement is in progress, the instrument returns averaged measured results.
- Executing a MEASure[:SCALar] command changes the instrument display to single view and returns one measurement result.
- Executing a MEASure:ARRay command changes the instrument display to multi view and returns measurement results for the number of data points (1024 maximum). On the -SW (single-wavelength) suffix code type, one measured result is returned.
- To query measurement results without changing the instrument settings, use the READ Sub System commands. (Related commands: FETCh Sub System, READ Sub System)

#### :MEASure:ARRay:POWer?

Function	Queries the peak power in multi view mode for single measurement (View Mode is set to MULTI).
Syntax	:MEASure:ARRay:POWer?<wsp> [<expected_value>] <expected_value> (power): MAXimum MINimum DEFault <NRf> MAX: Specifies the maximum power peak MIN: Specifies the minimum power peak DEF: Specifies the selected peak <NRf>: Specifies the peak closest to the specified power You can specify the power in unit of dBm or W. If you omit the unit, the instrument will assume W.
Example	:MEAS:ARR:POW? -> 3, -3.97000000E+000,-7.31000000E+000, -1.08700000E+001<END>
Description	<ul style="list-style-type: none"><li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;power1&gt;,&lt;power2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;power1&gt;,&lt;power2&gt;,...: Peak power values</li><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to &lt;NRf&gt; or DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in dBm or W depending on the parameter.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

#### :MEASure:ARRay:POWer:FREQuency?

Function	Queries the peak frequencies in multi view mode for single measurement (View Mode is set to MULTI).
Syntax	:MEASure:ARRay:POWer:FREQuency? <wsp>[<expected_value>] <expected_value> (frequency): MAXimum MINimum DEFault <NRf> MAX: Specifies the maximum frequency peak MIN: Specifies the minimum frequency peak DEF: Specifies the selected peak <NRf>: Specifies the peak closest to the specified frequency
Example	:MEAS:ARR:POW:FREQ? -> 3, +1.93738414E+014,+1.93596724E+014, +1.94163516E+014<END>
Description	<ul style="list-style-type: none"><li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;freq1&gt;,&lt;freq2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;freq1&gt;,&lt;freq2&gt;,...: Peak frequencies</li><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in Hz.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

#### :MEASure:ARRay:POWer:WAVelength?

Function	Queries the peak wavelengths in multi view mode for single measurement (View Mode is set to MULTI).
Syntax	:MEASure:ARRay:POWer:WAVelength?<wsp> [<expected_value>] <expected_value> (wavelength): MAXimum MINimum DEFault <NRf> MAX: Specifies the maximum wavelength peak MIN: Specifies the minimum wavelength peak DEF: Specifies the selected peak <NRf>: Specifies the peak closest to the specified wavelength
Example	:MEAS:ARR:POW:WAV? -> 3, +1.54740844E-006,+1.54854097E-006, +1.54402055E-006<END>
Description	<ul style="list-style-type: none"><li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. &lt;peak_num&gt;,&lt;wav1&gt;,&lt;wav2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;wav1&gt;,&lt;wav2&gt;,...: Peak wavelengths</li></ul>

- If you specify a parameter, the current (selected) peak on the instrument screen will change.
- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in m.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**:MEASure:ARRAY:POWER:WNUMber?**

Function	Queries the peak wavenumbers in multi view mode for single measurement (View Mode is set to MULTI).
Syntax	<pre>:MEASure:ARRay:POWer:WNUMber?&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (wavenumber):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum wavenumber peak MIN: Specifies the minimum wavenumber peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the specified wavenumber</pre>
Example	<pre>:MEAS:ARR:POW:WNUM? -&gt; 3, +6.46241790E+005,+6.45769160E+005, +6.47659780E+005&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"> <li>• This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned.  <pre>&lt;peak_num&gt;,&lt;wnum1&gt;,&lt;wnum2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;wnum1&gt;,&lt;wnum2&gt;,... : Peak wavenumbers</pre> </li> <li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in <math>m^{-1}</math>.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:MEASure[:SCALar]:POWER?**

Function	Queries the peak power in single view mode for single measurement (View Mode is set to SINGLE).
Syntax	<pre>:MEASure[:SCALar]:POWer?&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (power):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum power peak MIN: Specifies the minimum power peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the specified power You can specify the power in unit of dBm or W. If you omit the unit, the instrument will assume W.</pre>
Example	<pre>:MEAS:POW? -&gt; -7.84000000E+000&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"> <li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li> <li>• If the parameter is not set to &lt;NRf&gt; or DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in dBm or W depending on the parameter.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>

**:MEASure[:SCALar]:POWER:FREQuency?**

Function	Queries the peak frequency in single view mode for single measurement (View Mode is set to SINGLE).
Syntax	<pre>:MEASure[:SCALar]:POWer:FREQuency? &lt;wsp&gt;[&lt;expected_value&gt;] &lt;expected_value&gt; (frequency):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum frequency peak MIN: Specifies the minimum frequency peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the specified frequency</pre>
Example	<pre>:MEAS:POW:FREQ? -&gt; +1.93596757E+014&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"> <li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li> <li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li> <li>• Query results are returned in Hz.</li> <li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li> </ul>



## 5.5 Device-specific Commands

### :MEASure[:SCALar]:POWER:WAVelength?

Function	Queries the peak wavelength in single view mode for single measurement (View Mode is set to SINGLE).
Syntax	<pre>:MEASure[:SCALar]:POWER:WAVelength? &lt;wsp&gt;[&lt;expected_value&gt;] &lt;expected_value&gt; (wavelength):     MAXimum MINimum DEFAULT &lt;NRf&gt; MAX: Specifies the maximum wavelength peak MIN: Specifies the minimum wavelength peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the     specified wavelength</pre>
Example	<pre>:MEAS:POW:WAV? -&gt; +1.54854010E-006&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in m.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

### :MEASure[:SCALar]:POWER:WNUMber?

Function	Queries the peak wavenumber in single view mode for single measurement (View Mode is set to SINGLE).
Syntax	<pre>:MEASure[:SCALar]:POWER:WNUMber?&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (wavenumber):     MAXimum MINimum DEFAULT &lt;NRf&gt; MAX: Specifies the maximum wavenumber peak MIN: Specifies the minimum wavenumber peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the     specified wavenumber</pre>
Example	<pre>:MEAS:POW:WNUM? -&gt; +6.45769370E+005&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in <math>\text{m}^{-1}</math>.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

## MMEMory Sub System Commands

### Overview

- To include a directory in <"file name">, specify it in the following manner.
  - Absolute path  
To specify the absolute path, specify a backslash as the first character of <"file name">.
  - Relative path  
To specify the relative path from the current directory, do not specify a backslash as the first character of <"file name">.  
To specify the current directory, use the :MMEMory:CDIRectory command.
- If you omit INTernal|EXTernal, the current drive will be accessed.  
To specify the current drive, use the :MMEMory:CDRive command.
- If you omit the extension when you save a file, the appropriate extension will be added according to the data type.

### :MMEMory:CATalog?

Function	Queries all the files in the current directory.
Syntax	<pre>:MMEMory:CATalog?&lt;wsp&gt; [{"&lt;"directory"&gt; ROOT}[ , INTernal  EXTernal]] &lt;"directory"&gt;: A directory of your choice. Use     backslashes to specify the directory hierarchy. ROOT: Root directory INTernal: Internal memory EXTernal: USB memory</pre>
Example	<pre>:MMEM:CAT? "\Data\test" -&gt; 3 \Data\test  &lt;DIR&gt; result 24.5KB data.csv 12.3KB image.bmp &lt;END&gt;</pre>
Description	<p>In the above example, the \Data\test directory contains one directory and two files.</p> <p>For files, file sizes are indicated in front of file names.</p> <p>Response format: (indicated with line feeds)</p> <ul style="list-style-type: none"><li>• The number of files and directories</li><li>• Current directory</li><li>• An empty line</li><li>• File or directory name (each file or directory is displayed in its own line)</li><li>• This is an overlapping command.</li></ul>

**:MMEMory:CDIRectory**

Function Changes the current directory.

Syntax :MMEMory:CDIRectory<wsp><"directory">  
[,INTernal|EXTernal]  
<"directory">: A directory of your choice. Use  
backslashes to specify the directory hierarchy.  
ROOT: Root directory  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:CDIR "MYDIRETORY"

Description This is an overlapping command.

**:MMEMory:CDRive**

Function Sets or queries the current drive.

Syntax :MMEMory:CDRive<wsp>[INTernal|  
EXTernal]  
:MMEMory:CDRive?  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:CDR EXT  
:MMEM:CDR? -> EXT<END>

Description This is an overlapping command.

**:MMEMory:COPY**

Function Copies the specified file.

Syntax :MMEMory:COPY<wsp><"source\_file\_name">  
[INTernal|EXTernal],<"dest\_file\_name">  
[,INTernal|EXTernal]  
<"source\_file\_name">: Copy source file name  
<"dest\_file\_name">: Copy destination file name  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:COPY "test1.csv",,"test2.csv"

Description This is an overlapping command.

**:MMEMory:DATA?**

Function Reads the data of the specified file.

Syntax :MMEMory:DATA?<wsp><"file name">  
[,INTernal|EXTernal]  
<"file name">: Name of the file to read data from  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:DATA? "data2.csv"  
-> #238YOKOGAWA AQ6151B  
Data1, 2,3  
Data2, 3,4  
<END>

Description

- Response data is returned as block data.
- If the data does not fit in the buffer, a Too much data error (223) is returned.
- This is an overlapping command.

**:MMEMory:DELeTe**

Function Deletes the specified file.

Syntax :MMEMory:DELeTe<wsp><"file name">  
[,INTernal|EXTernal]  
<"file name">: Name of the file to delete  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:DEL "data1.txt"

Description This is an overlapping command.

**:MMEMory:INFormation?**

Function Queries the information about the specified file.

Syntax :MMEMory:INFormation?<wsp>  
<"file name">[,INTernal|EXTernal]  
<"file name">: Name of the specified file  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:INF? "data1.txt"  
-> 1024,2014/09/01 11:55:23<END>

Description

- <"file name"> Returns the file size and timestamp.  
<file\_size>,<time\_stamp>  
<file\_size>: Target file size in bytes in decimal notation  
<time\_stamp>: Update time in yyyy/mm/dd  
hh:mm:ss notation
- If the specified file does not exist, a USB Storage not inserted (30), USB Storage not initialized (31), or File not found(33) error is returned.
- This is an overlap command.

**:MMEMory:LOAD**

Function Loads the specified setup file into the instrument.

Syntax :MMEMory:LOAD<wsp><"file name">  
[,INTernal|EXTernal]  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:LOAD "SETTING1"

Description

- The file name extension can be omitted.
- This is an overlapping command.

**:MMEMory:MDIRectory**

Function Makes a directory.

Syntax :MMEMory:MDIRectory<wsp>  
<"directory\_name">[,INTernal|  
EXTernal]  
<"directory\_name">: Name of the directory to  
make  
INTernal: Internal memory  
EXTernal: USB memory

Example :MMEM:MDIR "MYDIR"

Description This is an overlapping command.



## 5.5 Device-specific Commands

### :MMEMory:PWDirectory?

Function	Queries the current directory.
Syntax	:MMEMory:PWDirectory?
Example	:MMEM:PWD? -> \MYDIR<END>
Description	This is an overlapping command.

### :MMEMory:REMove

Function	Makes the USB memory medium ready to be removed. Or, queries whether the USB storage medium can be removed.
Syntax	:MMEMory:REMove :MMEMory:REMove?
Response	0: Can be removed 1: Cannot be removed
Example	:MMEM:REM :MMEM:REM? -> 1<END>
Description	This is an overlapping command.

### :MMEMory:REName

Function	Renames the specified file.
Syntax	:MMEMory:REName<wsp><"new_file_name">,<"old_file_name">[,INTernal EXTernal] <"new_file_name">: New file name <"old_file_name">: Old file name INTernal: Internal memory EXTernal: USB memory medium
Example	:MMEM:REN "test1.csv", "test2.csv"
Description	This is an overlapping command.

### :MMEMory:STORE

Function	Saves wavelength data, setup data, screen capture, or logging data to a file.
Syntax	:MMEMory:STORE<wsp><source>,<"file name">[,INTernal EXTernal] <source> (data type): TABLE SETup SIMage1 SIMage2 SIMage3 DLOGging1 DLOGging2 TABLE: Wavelength data SETup: Setup data SIMage1: Screen capture (black and white) SIMage2: Screen capture (color) SIMage3: Screen capture (color, no background color) DLOGging1: Logging data (binary) DLOGging2: Logging data (CSV) <"file name">: File name INTernal: Internal memory EXTernal: USB memory medium
Example	:MMEM:STOR SET, "SETTINGS"
Description	<ul style="list-style-type: none"><li>The file name extension will be added automatically.</li><li>This is an overlapping command.</li></ul>

## READ Sub System Commands

### Overview

- This subsystem is a function for starting a measurement on the instrument and querying the measured results when the measurement is complete.  
These commands can be used to make queries without changing the instrument settings (executing these commands will not change the View Mode of the instrument).
- If the measurement on the instrument is stopped, the instrument will execute a single measurement and return the measured results.
- If the instrument is measuring (repeat measurement), it will return a execution error.
- If averaged measurement is in progress, the instrument returns averaged measured results.
- Executing a READ[:SCALar] command returns one measurement result.
- Executing a READ:ARRay command returns measurement results for the number of data points (1024 maximum).  
On the -SW (single-wavelength) suffix code type, one measured result is returned.
- Executing these commands will not change the display (single view or multi view). (Related commands: FETCH Sub System, MEASure Sub System)

### :READ?

Function	Queries the peak measurement result for single measurement.
Syntax	:READ?
Example	:READ? -> 3,+6.46241450E+005, +6.45768920E+005,+6.47659390E+005 <END>
Description	<ul style="list-style-type: none"><li>If the previous query command was in single view mode (a command that contains ":SCALar" in its command string), one measurement result is returned.</li><li>If the previous query command was in multi view mode (a command that contains ":ARRay" in its command string), measurement results are returned for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned. Power &lt;peak_num&gt;,&lt;power1&gt;,&lt;power2&gt;,... Wavelength &lt;peak_num&gt;,&lt;wav1&gt;,&lt;wav2&gt;,... Frequency &lt;peak_num&gt;,&lt;freq1&gt;,&lt;freq2&gt;,... Wavenumber &lt;peak_num&gt;,&lt;wnum1&gt;,&lt;wnum2&gt;,... &lt;peak_num&gt;: Number of peaks (0 to 1024) &lt;power1&gt;,&lt;power2&gt;,...: Peak power values &lt;wav1&gt;,&lt;wav2&gt;,...: Peak wavelengths &lt;freq1&gt;,&lt;freq2&gt;,...: Peak frequencies &lt;wnum1&gt;,&lt;wnum2&gt;,...: Peak wavenumbers</li><li>After power-on, the instrument returns wavelength values (in unit of meters).</li></ul>

**:READ:ARRAY:POWER?**

Function Queries the most recent peak's power values in multi view mode for single measurement.

Syntax :READ:ARRAY:POWER:<wsp>  
[<expected\_value>]  
<expected\_value> (power):  
MAXimum|MINimum|DEFAULT|<NRf>  
MAX: Specifies the maximum power peak  
MIN: Specifies the minimum power peak  
DEF: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified power

Example :READ:ARR:POW? -> 3,  
-3.77000000E+000,-7.72000000E+000,  
-1.04900000E+001<END>

Description

- This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned.  
<peak\_num>,<power1>,<power2>,...  
<peak\_num>: Number of peaks (0 to 1024)  
<power1>,<power2>,...: Peak power values
- If you specify a parameter, the current (selected) peak on the instrument screen will change.
- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in dBm or W depending on the parameter.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**:READ:ARRAY:POWER:FREQUENCY?**

Function Queries the most recent peak's frequencies in multi view mode for single measurement.

Syntax :READ:ARRAY:POWER:FREQUENCY:<wsp>  
[<expected\_value>]  
<expected\_value> (frequency):  
MAXimum|MINimum|DEFAULT|<NRf>  
MAX: Specifies the maximum frequency peak  
MIN: Specifies the minimum frequency peak  
DEF: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified frequency

Example :READ:ARR:POW:FREQ? -> 3,  
+1.93738284E+014,+1.93596611E+014,  
+1.94163376E+014<END>

Description

- This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned.  
<peak\_num>,<freq1>,<freq2>,...  
<peak\_num>: Number of peaks (0 to 1024)  
<freq1>,<freq2>,...: Peak frequencies
- If you specify a parameter, the current (selected) peak on the instrument screen will change.
- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in Hz.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**:READ:ARRAY:POWER:WAVELENGTH?**

Function Queries the most recent peak's wavelengths in multi view mode for single measurement.

Syntax :READ:ARRAY:POWER:WAVELENGTH:<wsp>  
[<expected\_value>]  
<expected\_value> (wavelength):  
MAXimum|MINimum|DEFAULT|<NRf>  
MAX: Specifies the maximum wavelength peak  
MIN: Specifies the minimum wavelength peak  
DEF: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified wavelength

Example :READ:ARR:POW:WAV? -> 3,  
+1.54740962E-006,+1.54854218E-006,  
+1.54402171E-006<END>

Description

- This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned.  
<peak\_num>,<wav1>,<wav2>,...  
<peak\_num>: Number of peaks (0 to 1024)  
<wav1>,<wav2>,...: Peak wavelengths
- If you specify a parameter, the current (selected) peak on the instrument screen will change.
- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in m.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

**:READ:ARRAY:POWER:WNUMBER?**

Function Queries the most recent peak's wavenumbers in multi view mode for single measurement.

Syntax :READ:ARRAY:POWER:WNUMBER:<wsp>  
[<expected\_value>]  
<expected\_value> (wavenumber):  
MAXimum|MINimum|DEFAULT|<NRf>  
MAX: Specifies the maximum wavenumber peak  
MIN: Specifies the minimum wavenumber peak  
DEF: Specifies the selected peak  
<NRf>: Specifies the peak closest to the specified wavenumber

Example :READ:ARR:POW:WNUM? -> 3,  
+6.46241320E+005,+6.45768650E+005,  
+6.46714090E+005<END>

Description

- This command returns measured results for the number of data points. On the -SW (single-wavelength) suffix code type, one measured result is returned.  
<peak\_num>,<wnum1>,<wnum2>,...  
<peak\_num>: Number of peaks (0 to 1024)  
<wnum1>,<wnum2>,...: Peak wavenumbers
- If you specify a parameter, the current (selected) peak on the instrument screen will change.
- If the parameter is not set to DEF, the auto peak search function will be set to OFF.
- Query results are returned in m<sup>-1</sup>.
- Parameters are valid on the -MW (multi-wavelength) suffix code type.

## 5.5 Device-specific Commands

### **:READ[:SCALar]:POWer?**

Function	Queries the most recent peak's power values for single view for single measurement.
Syntax	<pre>:READ[:SCALar]:POWer?&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (power):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum power peak MIN: Specifies the minimum power peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the     specified power</pre>
Example	<pre>:READ:POW? -&gt; -7.43000000E+000&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in dBm or W depending on the parameter.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

### **:READ[:SCALar]:POWer:FREQuency?**

Function	Queries the most recent peak's frequencies in single view mode for single measurement.
Syntax	<pre>:READ[:SCALar]:POWer:FREQuency? &lt;wsp&gt;[&lt;expected_value&gt;] &lt;expected_value&gt; (frequency):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum frequency peak MIN: Specifies the minimum frequency peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the     specified frequency</pre>
Example	<pre>:READ:POW:FREQ? -&gt; +1.93596574E+014&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in Hz.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

### **:READ[:SCALar]:POWer:WAVelength?**

Function	Queries the most recent peak's wavelengths in single view mode for single measurement.
Syntax	<pre>:READ[:SCALar]:POWer:WAVelength? &lt;wsp&gt;[&lt;expected_value&gt;] &lt;expected_value&gt; (wavelength):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum wavelength peak MIN: Specifies the minimum wavelength peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the     specified wavelength</pre>
Example	<pre>:READ:POW:WAV? -&gt; +1.54854253E-006&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in m.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

### **:READ[:SCALar]:POWer:WNUMber?**

Function	Queries the most recent peak's wavenumbers in single view mode for single measurement.
Syntax	<pre>:READ[:SCALar]:POWer:WNUMber?&lt;wsp&gt; [&lt;expected_value&gt;] &lt;expected_value&gt; (wavenumber):     MAXimum MINimum DEFault &lt;NRf&gt; MAX: Specifies the maximum wavenumber     peak MIN: Specifies the minimum wavenumber peak DEF: Specifies the selected peak &lt;NRf&gt;: Specifies the peak closest to the     specified wavenumber</pre>
Example	<pre>:READ:POW:WNUM? -&gt; +6.45768740E+005&lt;END&gt;</pre>
Description	<ul style="list-style-type: none"><li>• If you specify a parameter, the current (selected) peak on the instrument screen will change.</li><li>• If the parameter is not set to DEF, the auto peak search function will be set to OFF.</li><li>• Query results are returned in m<sup>-1</sup>.</li><li>• Parameters are valid on the -MW (multi-wavelength) suffix code type.</li></ul>

**SENSe Sub System Commands****[ :SENSe] :CORRection:DEVIce**

**Function** Sets or queries the type of light under measurement (Device Mode).

**Syntax** [ :SENSe] :CORRection:  
DEVIce<wsp>NARRow|BROad  
[ :SENSe] :CORRection:DEVIce?  
NARRow: CW light  
BROad: Modulated light

**Example** :CORR:DEV NARR  
:CORR:DEV? -> NARR<END>

**[ :SENSe] :CORRection:MEDIum**

**Function** Sets or queries the medium that the light travels through (MEAS WL).

**Syntax** [ :SENSe] :CORRection:  
MEDIum<wsp>AIR|VACuum  
[ :SENSe] :CORRection:MEDIum?  
AIR: Standard air  
VACuum: Vacuum

**Example** :SENS:CORR:MED AIR  
:SENS:CORR:MED? -> AIR<END>

**[ :SENSe] :CORRection:OFFSet[ :MAGNItude]**

**Function** Sets or queries the power offset.

**Syntax** [ :SENSe] :CORRection:  
OFFSet[ :MAGNItude]<wsp><offset>  
[ :SENSe] :CORRection:OFFSet[ :  
MAGNItude]?  
<offset> (power offset):  
<NRf>|MINimum|MAXimum  
MINimum: -10 db  
MAXimum: +10 db

**Example** :CORR:OFFS 1.2  
:CORR:OFFS? -> +1.20000000E+000<END>

**Description** Query results are returned in dB.

**[ :SENSe] :URATe**

**Function** Sets or queries the update rate (wavelength measurement speed).

**Syntax** [ :SENSe] :URATe<wsp>NORMal|FAST  
[ :SENSe] :URATe?  
NORMal: Data is updated at the normal measurement speed.  
FAST: Data is updated at the high-speed measurement speed.

**Example** :SENS:URAT FAST  
:SENS:URAT? -> FAST<END>

**STATus Sub System Commands****Overview**

The commands in this group are used to make settings and queries related to the status report.

There are no front panel keys that correspond to the commands in this group.

For information about status reports, see chapter 4.

**:STATus:OPERation:CONDition?**

**Function** Queries the contents of the operation status condition register.

**Syntax** :STATus:OPERation:CONDition?

**Example** :STAT:OPER:COND? -> +2048<END>

**Description** This is an overlapping command.

**:STATus:OPERation:ENABle**

**Function** Sets or queries the contents of the operation status enable register.

**Syntax** :STATus:OPERation:  
ENABle<wsp><integer>  
:STATus:OPERation:ENABle?  
<integer>: 0 to 65535

**Example** :STAT:OPER:ENAB 4095  
:STAT:OPER:ENAB? -> +4095<END>

**Description** This is an overlapping command.

**:STATus:OPERation[ :EVENT] ?**

**Function** Queries the contents of the operation status event register.

**Syntax** :STATus:OPERation[ :EVENT] ?

**Example** :STAT:OPER? -> +4096<END>

**Description** This is an overlapping command.

**:STATus:OPERation:NTRansition**

**Function** Sets or queries the contents of the operation status N Transition register.

**Syntax** :STATus:OPERation:NTRansition<wsp>  
<integer>  
:STATus:OPERation:NTRansition?  
<integer>: 0 to 65535

**Example** :STAT:OPER:NTR 4096  
:STAT:OPER:NTR? -> +4096<END>

**Description** This is an overlapping command.

**:STATus:OPERation:PTRansition**

**Function** Sets or queries the contents of the operation status P Transition register.

**Syntax** :STATus:OPERation:  
PTRansition<wsp><integer>  
:STATus:OPERation:PTRansition?  
<integer>: 0 to 65535

**Example** :STAT:OPER:PTR 4096  
:STAT:OPER:PTR? -> +4096<END>

**Description** This is an overlapping command.

## 5.5 Device-specific Commands

### **:STATus:PRESet**

Function Clears the event register and set all bits in the enable register.

Syntax :STATus:PRESet

Example :STAT:PRES

Description

- This command clears the contents of the ENABLE, NTRansition, and PTRansition registers.
- This is an overlapping command.

### **:STATus:QUESTionable:CONDition?**

Function Queries the contents of the questionable status condition register.

Syntax :STATus:QUESTionable:CONDition?

Example :STAT:QUES:COND? -> +24<END>

Description This is an overlapping command.

### **:STATus:QUESTionable:ENABLE**

Function Sets or queries the contents of the questionable status enable register.

Syntax :STATus:QUESTionable:  
ENABLE<wsp><integer>  
:STATus:QUESTionable:ENABLE?  
<integer>: 0 to 65535

Example :STAT:QUES:ENAB 4095  
:STAT:QUES:ENAB? -> +4095<END>

Description This is an overlapping command.

### **:STATus:QUESTionable[:EVENT]?**

Function Queries the contents of the questionable status event register.

Syntax :STATus:QUESTionable[:EVENT]?

Example :STAT:QUES? -> +8<END>

Description This is an overlapping command.

### **:STATus:QUESTionable:NTRansition**

Function Sets or queries the contents of the questionable status N Transition register.

Syntax :STATus:QUESTionable:  
NTRansition<wsp><integer>  
:STATus:QUESTionable:NTRansition?  
<integer>: 0 to 65535

Example :STAT:QUES:NTR 24  
:STAT:QUES:NTR? -> +24<END>

Description This is an overlapping command.

### **:STATus:QUESTionable:PTRansition**

Function Sets or queries the contents of the questionable status P Transition register.

Syntax :STATus:QUESTionable:  
PTRansition<wsp><integer>  
:STATus:QUESTionable:PTRansition?  
<integer>: 0 to 65535

Example :STAT:QUES:PTR 24  
:STAT:QUES:PTR? -> +24<END>

Description This is an overlapping command.

## SYSTem Sub System Commands

### **:SYSTem:BUZZer[:CLICK]**

Function Sets or queries whether the click sounds (buzzer) that are produced when a key is pressed is on or off.

Syntax :SYSTem:BUZZer[:CLICK]<wsp>0|OFF|1|  
ON  
:SYSTem:BUZZer[:CLICK]?  
0|OFF: Click sound off  
1|ON: Click sound on

Example :SYST:BUZZ ON  
:SYST:BUZZ? -> 1<END>

Description This is an overlapping command.

### **:SYSTem:BUZZer:WARNing**

Function Sets or queries whether the buzzer notifications that are produced when alarms occur is on or off.

Syntax :SYSTem:BUZZer:WARNing<wsp>0|OFF|1|  
ON  
:SYSTem:BUZZer:WARNing?  
0|OFF: Buzzer notification off  
1|ON: Buzzer notification on

Example :SYST:BUZZ:WARN ON  
:SYST:BUZZ:WARN? -> 1<END>

Description This is an overlapping command.

### **:SYSTem:CAPability:WAVelength?**

Function Queries measurable wavelength range.

Syntax :SYSTem:CAPability:WAVelength?

Example :SYST:CAP:WAV? -> +1.27000000E-006,  
+1.65000000E-006<END>

Description The response is returned with the measurable minimum wavelength (m) first and then the measurable maximum wavelength (m).

### **:SYSTem:DATE**

Function Sets or queries the date.

Syntax :SYSTem:DATE<wsp><year>,<month>,<day>  
:SYSTem:DATE?  
<year>: Year  
<month>: Month  
<day>: Day

Example :SYST:DATE 2012,04,09  
:SYST:DATE? -> 2012,04,09<END>

Description

- The response is returned in this order: year, month, and day.
- This is an overlapping command.

**:SYSTem:ENVironment?**

Function	Queries the instrument's status.
Syntax	:SYSTem:ENVironment?
Example	:SYST:ENV? -> 2.30000000E+001, 2.05000000E+001,9.90000000E+002, -1<END>
Description	<ul style="list-style-type: none"> <li>The response is returned in the following order: internal instrument temperature (°C), internal interferometer temperature (°C), pressure (hPa), -1 (this instrument does not have a cooling fan).</li> <li>This is an overlap command.</li> </ul>

**:SYSTem:ERRor?**

Function	Queries the error information of the instrument.
Syntax	:SYSTem:ERRor?
Example	:SYST:ERR? -> +0,"No error"<END>
Description	<ul style="list-style-type: none"> <li>The response is returned in this order: error number, error message.</li> <li>For details on messages, see section 4.1 in the User's Manual, IM AQ6150B-01EN.</li> <li>This is an overlapping command.</li> </ul>

**:SYSTem:INFormation?**

Function	Queries device-specific information (model and special codes).
Syntax	:SYSTem:INFormation?<wsp>0 1
	0: Model code
	1: Special code
Responses	
	<string>: Model code or special code
Example	:SYST:INF? 0 -> AQ6150B-10-MW<END>
Description	If there is no special code information, "NONE" is returned.

**:SYSTem:LANGuage**

Function	Sets or queries the language.
Syntax	:SYSTem:LANGuage<wsp>ENGLish  CHINese JAPanese
	:SYSTem:LANGuage?
	ENGLish: English
	CHINese: Chinese
	JAPanese: Japanese
Example	:SYSTem:LANG ENGL :SYSTem:LANG? -> ENGL<END>
Description	This is an overlapping command.

**:SYSTem:PRESet**

Function	Initializes the measurement setup conditions of the instrument.
Syntax	:SYSTem:PRESet
Example	:SYST:PRES
Description	<p>Buzzer, network, and some other settings are not initialized.</p> <p>For details on what is initialized, see section 7.5 in the User's Manual, IM AQ6150B-01EN.</p>

**:SYSTem:REFLaser:CONDition?**

Function	Queries the internal reference light source condition.
Syntax	:SYSTem:REFLaser:CONDition?
Response	0: Laser output off 1: Laser starting 2: Laser stabilizing 3: Normal 4: Replacement period 5: Laser malfunction
Example	:SYST:REFL:COND? -> 3<END>
Description	<ul style="list-style-type: none"> <li>The instrument can make measurements in conditions 2 to 4 above.</li> <li>For the replacement period, see section 2.8 in the Getting Started Guide, IM AQ6150B-02EN.</li> <li>This is an overlapping command.</li> </ul>

**:SYSTem:REFLaser:COUNter?**

Function	Queries the number of times the internal reference light source has been turned on.
Syntax	:SYSTem:REFLaser:COUNter?
Example	:SYST:REFL:COUN? -> 40<END>

**:SYSTem:REFLaser:OTIME?**

Function	Queries the total running time (in hours) of the internal reference light source.
Syntax	:SYSTem:REFLaser:OTIME?
Example	:SYST:REFL:OTIM? -> 100<END>
Description	<p>This command indicates the total duration that the internal reference light source has been on. You can use this duration as a guideline for deciding whether to replace the internal reference light source.</p> <p>For information about the time, see section 2.8 in the Getting Started Guide, IM AQ6150B-02EN.</p>

## 5.5 Device-specific Commands

### :SYSTem:TIME

Function	Sets or queries the time.
Syntax	:SYSTem:TIME<wsp><hour>,<minute>,<second> :SYSTem:TIME? <hour>: Hour <minute>: Minute <second>: Second
Example	:SYST:TIME 17,20,00 :SYST:TIME? -> 17,20,00<END>
Description	<ul style="list-style-type: none"><li>• The response is returned in this order: hour, minute, and second.</li><li>• This is an overlapping command.</li></ul>

### :SYSTem:VERSion?

Function	Queries the SCPI (Standard Commands for Programmable Interfaces) version.
Syntax	:SYSTem:VERSion?
Example	:SYST:VERS? -> 1999.0<END>
Description	This is an overlapping command.

## TRIGger Sub System Commands

### [ :TRIGger ] :ABORt

Function	Stops measurement.
Syntax	[ :TRIGger ] :ABORt
Example	:ABOR
Description	This is an overlapping command.

### [ :TRIGger ] :INITiate:CONTinuous

Function	Executes or queries repeat measurement.
Syntax	[ :TRIGger ] :INITiate:CONTinuous<wsp> 0 OFF 1 ON [ :TRIGger ] :INITiate:CONTinuous? 0 OFF: Repeat measurement stopped 1 ON: Repeat measurement running
Example	:INIT:CONT ON :INIT:CONT? -> 1<END>
Description	This command can be overlapping command while parameter is on. This command can be overlappable command while parameter is off.

### [ :TRIGger ] :INITiate [ :IMMediate ]

Function	Executes a single measurement.
Syntax	[ :TRIGger ] :INITiate [ :IMMediate ]
Example	:INIT
Description	<ul style="list-style-type: none"><li>• This command is discarded while repeat measurement is in progress.</li><li>• This is an overlappable command.</li></ul>

**UNIT Sub System Commands****:UNIT[:POWER]**

Function	Sets or queries the power unit.
Syntax	:UNIT[:POWER]<wsp>W DBM :UNIT[:POWER]? W: Watt DBM: dBm
Example	:UNIT DBM :UNIT? -> DBM<END>
Description	<ul style="list-style-type: none"><li>• If you specify W, the instrument displays mW.</li><li>• This is an overlapping command.</li></ul>

**:UNIT:WL**

Function	Sets or queries the wavelength unit.
Syntax	:UNIT:WL<wsp>THZ NM ICM :UNIT:WL? THZ: Hz (frequency) NM: m (wavelength) ICM: cm <sup>-1</sup> (wavenumber)
Example	:UNIT:WL THZ :UNIT:WL? -> THZ<END>
Description	This is an overlapping command.



# Appendix 1 About the IEEE 488.2-1992 Standard

The instrument's GP-IB interface conforms to the IEEE 488.2-1992 standard. This standard specifies that the following 23 items be stated in the document. This section describes these items.

- (1) **Of the IEEE 488.1 interface functions, the subsets that are supported**  
See section 2.3, "GP-IB Interface Specifications."
- (2) **The operation of the device when it is assigned an address outside the 0 to 30 range.**  
The address of this instrument cannot be set to an address outside the 0 to 30 range.
- (3) **Reaction of the device when the user changes the address**  
The address change is detected when the user presses SYSTEM and then the GP-IB ADDRESS soft key, and changes the address.  
The new address is valid until the next time it is changed.
- (4) **Device settings at power-up. The commands that can be used at power-up.**  
As a basic rule, the previous settings (the settings that were in use when the instrument was turned off) are used.  
There are no limitations on the commands that can be used at power-up.
- (5) **Message exchange options**
  - (a) **Input buffer size**  
2 MB
  - (b) **Queries that return multiple response messages**  
See the example of the commands given in chapter 5.
  - (c) **Queries that create response data when the command syntax is being analyzed**  
All queries create response data when the command syntax is analyzed.
  - (d) **Queries that create response data during reception**  
There are no queries of which the response data are created upon receiving a send request from the controller.
  - (e) **Commands that have parameters that restrict one another**  
There are no commands that have parameters that restrict one another.
- (6) **Items that are included in the functional or composite header elements constituting a command**  
See chapter 5.
- (7) **Buffer sizes that affect block data transmission**  
2 MB including the header length
- (8) **A list of program data elements that can be used in equations and their nesting limitations**  
Equations cannot be used.
- (9) **Syntax of the responses to queries**  
See the example of the commands given in chapter 5.
- (10) **Communication between devices that do not follow the response syntax**  
Not supported.
- (11) **Size of the response data block**  
2 MB including the header length
- (12) **A list of supported common commands**  
See section 5.4, "Common Commands."
- (13) **Device condition after a successful calibration**  
\*CAL? is not supported.
- (14) **The maximum length of block data that can be used for the \*DDT trigger macro definition**  
Not supported.
- (15) **The maximum length of the macro label for defining macros, the maximum length of block data that can be used for the macro definition, and the process when recursion is used in macro definitions**  
Macro functions are not supported.
- (16) **Reply to the \*IDN? query**  
See section 5.4, "Common Commands."
- (17) **Size of storage area for protected user data for PUD and \*PUD?**  
\*PUD and \*PUD? are not supported.
- (18) **The length of the \*RDT and \*RDT? resource names**  
\*RDT and \*RDT? are not supported.
- (19) **The change in the status due to \*RST, \*LRN?, \*RCL, and \*SAV**  
\*RST, \*RCL, \*SAV  
See section 5.4, "Common Commands."  
\*LRN?  
This common command is not supported.

**(20) The extent of the self-test using the \*TST? command**

See section 5.4, "Common Commands."

**(21) The structure of the extended return status**

See chapter 4.

**(22) Whether each command is processed in an overlapped manner or sequentially**

See chapter 5.

**(23) The description of the execution of each command**

See the explanations of each command's function in chapter 5; the user's manual, IM AQ6150B-01EN; and the getting started guide, IM AQ6150B-02EN.