ADVANTEST

ADVANTEST CORPORATION

Q7761
Programming Guide

MANUAL NUMBER FFE-8440122B00

First printing March 15, 2004
Printed in Japan

Safety Summary

To ensure thorough understanding of all functions and to ensure efficient use of this instrument, please read the manual carefully before using. Note that Advantest bears absolutely no responsibility for the result of operations caused due to incorrect or inappropriate use of this instrument.

If the equipment is used in a manner not specified by Advantest, the protection provided by the equipment may be impaired.

• Warning Labels

Warning labels are applied to Advantest products in locations where specific dangers exist. Pay careful attention to these labels during handling. Do not remove or tear these labels. If you have any questions regarding warning labels, please ask your nearest Advantest dealer. Our address and phone number are listed at the end of this manual.

Symbols of those warning labels are shown below together with their meaning.

DANGER: Indicates an imminently hazardous situation which will result in death or serious personal injury.

WARNING: Indicates a potentially hazardous situation which will result in death or serious personal injury.

CAUTION: Indicates a potentially hazardous situation which will result in personal injury or a damage to property including the product.

• Basic Precautions

Please observe the following precautions to prevent fire, burn, electric shock, and personal injury.

- Use a power cable rated for the voltage in question. Be sure however to use a power cable conforming to safety standards of your nation when using a product overseas.
- When inserting the plug into the electrical outlet, first turn the power switch OFF and then insert the plug as far as it will go.
- When removing the plug from the electrical outlet, first turn the power switch OFF and then pull it out by gripping the plug. Do not pull on the power cable itself. Make sure your hands are dry at this time.
- Before turning on the power, be sure to check that the supply voltage matches the voltage requirements of the instrument.
- Connect the power cable to a power outlet that is connected to a protected ground terminal. Grounding will be defeated if you use an extension cord which does not include a protected ground terminal.
- Be sure to use fuses rated for the voltage in question.
- Do not use this instrument with the case open.
- Do not place anything on the product and do not apply excessive pressure to the product. Also, do not place flower pots or other containers containing liquid such as chemicals near this

product.

- When the product has ventilation outlets, do not stick or drop metal or easily flammable objects into the ventilation outlets.
- When using the product on a cart, fix it with belts to avoid its drop.
- When connecting the product to peripheral equipment, turn the power off.

• Caution Symbols Used Within this Manual

Symbols indicating items requiring caution which are used in this manual are shown below together with their meaning.

DANGER: Indicates an item where there is a danger of serious personal injury (death or serious injury).

WARNING: Indicates an item relating to personal safety or health.

CAUTION: Indicates an item relating to possible damage to the product or instrument or relating to a restriction on operation.

• Safety Marks on the Product

The following safety marks can be found on Advantest products.

⚠ : ATTENTION - Refer to manual.

⏚ : Protective ground (earth) terminal.

ϟ : DANGER - High voltage.

⚡ : CAUTION - Risk of electric shock.

• Replacing Parts with Limited Life

The following parts used in the instrument are main parts with limited life.
Replace the parts listed below before their expected lifespan has expired to maintain the performance and function of the instrument.
Note that the estimated lifespan for the parts listed below may be shortened by factors such as the environment where the instrument is stored or used, and how often the instrument is used.
The parts inside are not user-replaceable. For a part replacement, please contact the Advantest sales office for servicing.

Each product may use parts with limited life.
For more information, refer to the section in this document where the parts with limited life are described.

Main Parts with Limited Life

Part name	Life
Unit power supply	5 years
Fan motor	5 years
Electrolytic capacitor	5 years
LCD display	6 years
LCD backlight	2.5 years
Floppy disk drive	5 years
Memory backup battery	5 years

- **Hard Disk Mounted Products**

The operational warnings are listed below.

- Do not move, shock and vibrate the product while the power is turned on.
  Reading or writing data in the hard disk unit is performed with the memory disk turning at a high speed. It is a very delicate process.

- Store and operate the products under the following environmental conditions.
  An area with no sudden temperature changes.
  An area away from shock or vibrations.
  An area free from moisture, dirt, or dust.
  An area away from magnets or an instrument which generates a magnetic field.

- Make back-ups of important data.
  The data stored in the disk may become damaged if the product is mishandled. The hard disc has a limited life span which depends on the operational conditions. Note that there is no guarantee for any loss of data.

- **Precautions when Disposing of this Instrument**

When disposing of harmful substances, be sure dispose of them properly with abiding by the state-provided law.

Harmful substances: (1) PCB (polycarbon biphenyl)
(2) Mercury
(3) Ni-Cd (nickel cadmium)
(4) Other
Items possessing cyan, organic phosphorous and hexadic chromium and items which may leak cadmium or arsenic (excluding lead in solder).

Example: fluorescent tubes, batteries

Environmental Conditions

This instrument should be only be used in an area which satisfies the following conditions:

- An area free from corrosive gas
- An area away from direct sunlight
- A dust-free area
- An area free from vibrations
- Altitude of up to 2000 m

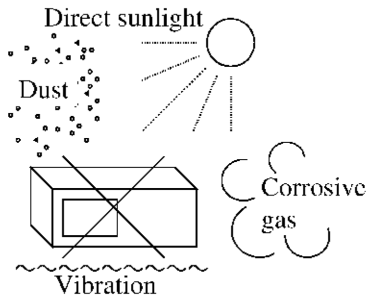
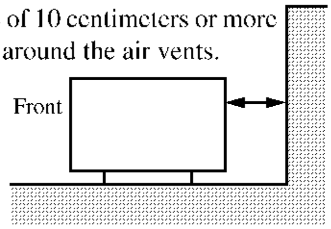


Figure-1 Environmental Conditions

- Operating position

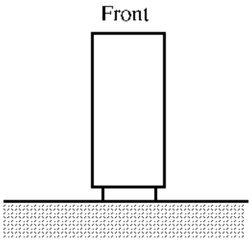


A clear space of 10 centimeters or more must be kept around the air vents.

The instrument must be used in a horizontal position.
A cooling fan, which prevents the internal temperature from rising, is equipped with the instrument.
The air vents on the case must be unblocked.

Figure-2 Operating Position

- Storage position



This instrument should be stored in a horizontal position.
When placed in a vertical (upright) position for storage or transportation, ensure the instrument is stable and secure.

-Ensure the instrument is stable.
-Pay special attention not to fall.

Figure-3 Storage Position

- The classification of the transient over-voltage, which exists typically in the main power supply, and the pollution degree is defined by IEC61010-1 and described below.

  Impulse withstand voltage (over-voltage) category II defined by IEC60364-4-443

  Pollution Degree 2

Types of Power Cable

Replace any references to the power cable type, according to the following table, with the appropriate power cable type for your country.

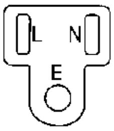
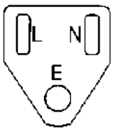
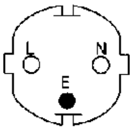
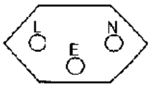
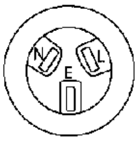
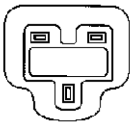
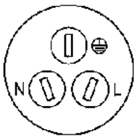
Plug configuration	Standards	Rating, color and length	Model number (Option number)
	PSE: Japan Electrical Appliance and Material Safety Law	125 V at 7 A Black 2 m (6 ft)	Straight: A01402 Angled: A01412
	UL: United States of America CSA: Canada	125 V at 7 A Black 2 m (6 ft)	Straight: A01403 (Option 95) Angled: A01413
	CEE: Europe DEMKO: Denmark NEMKO: Norway VDE: Germany KEMA: The Netherlands CEBEC: Belgium OVE: Austria FIMKO: Finland SEMKO: Sweden	250 V at 6 A Gray 2 m (6 ft)	Straight: A01404 (Option 96) Angled: A01414
	SEV: Switzerland	250 V at 6 A Gray 2 m (6 ft)	Straight: A01405 (Option 97) Angled: A01415
	SAA: Australia, New Zealand	250 V at 6 A Gray 2 m (6 ft)	Straight: A01406 (Option 98) Angled: ---------
	BS: United Kingdom	250 V at 6 A Black 2 m (6 ft)	Straight: A01407 (Option 99) Angled: A01417
	CCC:China	250 V at 10 A Black 2 m (6 ft)	Straight: A114009 (Option 94) Angled: A114109

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

This chapter describes the outline of this manual for efficient utilization.

1.1 Outline of This Manual

This manual is a program creation guide for the Q7761 Optical Network Analyzer. This manual is organized so that the user can learn the method of creating programs that perform remote control of the Q7761 Optical Network Analyzer by using SCPI commands.

This manual assumes that the user is familiar with the manual operation of the Q7761 Optical Network Analyzer. Therefore, this manual does not include a detailed explanation about the operation method and functions of the Q7761 Optical Network Analyzer. For more information on the operation method and functions, refer to the User's Guide.

The outline of each chapter is shown below:

Chapter 1, "INTRODUCTION"	This chapter describes the outline of this manual for efficient utilization.
Chapter 2, "OVERVIEW OF REMOTE CONTROL"	This chapter describes the overview of the remote control system and SCPI commands.
Chapter 3, "EXAMPLE REMOTE CONTROL PROGRAMS"	This chapter provides example programs, which allow this instrument to be controlled by remote.
Chapter 4, "COMMAND REFERENCE"	Command reference. The command reference explains the commands in order of function. In explanation, the following items are explained: • Command syntax (command syntax, query syntax, or both) • Response data format (when query exists)

1.2 Other Manuals for This Instrument

The following manuals are provided for this instrument:

- User's Guide (Product code: {EQ7761/U}, English)

  This manual describes the information required to use the Q7761. It includes setup, basic operation, application measurement, function explanation, specifications, and maintenance.

- Programming Guide (Product code: {EQ7761/P}, English, This manual}

  This manual describes information related to programming for automatic measurement of the Q7761. It includes an overview of remote control, SCPI command reference, and application program examples.

- Performance Test Guide (Product code: {EQ7761/T}, English)

  This manual describes the information required to verify the performance of the Q7761. It includes the performance test procedure and specifications.

2. OVERVIEW OF REMOTE CONTROL

This chapter describes the overview of the remote control system and SCPI commands.

2.1 Types of Remote Control Systems

The following two types of remote control systems can be configured, depending on the interface:

Interface	Overview
GPIB (Talker/Listener mode)	This system controls the Q7761 and other devices connected from the external controller through GPIB. For more information, refer to “2.2 GPIB Remote Control System”.
LAN	This system controls the Q7761 and other devices connected from the external controller through LAN. For more information, refer to “2.3 LAN Remote Control System”.

2.2 GPIB Remote Control System

The GPIB (General Purpose Interface Bus) that is compliant with IEEE standards 488.1-1987 and 488.2-1987 comes standard with this instrument so that remote control can be performed from the external controller.

The controlling method using the GPIB remote control function is explained below.

2.2.1 What is the GPIB?

The GPIB (General Purpose Interface Bus) is a high performance bus that integrates computers and measuring instruments.

Operation of the GPIB is defined by IEEE standard 488.1-1987. Since the GPIB has bus structure interfaces, a specific device can be specified by assigning a unique device address to each device. Up to 15 devices can be connected to one bus in parallel. A GPIB device is equipped with at least one of the following functions:

- Talker

  The device that is set to send data to the bus is called the "talker". On the GPIB bus, only one device acts as an active talker.

- Listener

  The device that is set to receive data on the bus is called the "listener". Two or more active listener devices can exist on a GPIB bus.

- Controller

  The device that specifies talkers and listeners is called the "controller". On a GPIB bus, only one device acts as an active controller. Of these controllers, the device that can control IFC and REN messages is expressly called the "system controller".

  Only one system controller is permitted on a GPIB bus. If there are two or more controllers on a bus, the system controller becomes the active controller at the time of system startup and the other devices with controller capability act as addressable devices.

  To set another controller to an active controller, use Take Control (TCT) interface messages. At this time, this controller becomes a non-active controller.

  The controller controls the entire system by sending interface or device messages to each measuring instrument. The roles of these messages are shown below.

  - Interface message: Controls the GPIB bus.
  - Device message: Controls the measuring instruments.

2.2.2 Setting up the GPIB

1. GPIB connection

   The standard GPIB connection is shown in Figure 2-1. Fix the GPIB connector firmly with two screws such that they do not loosen during use.

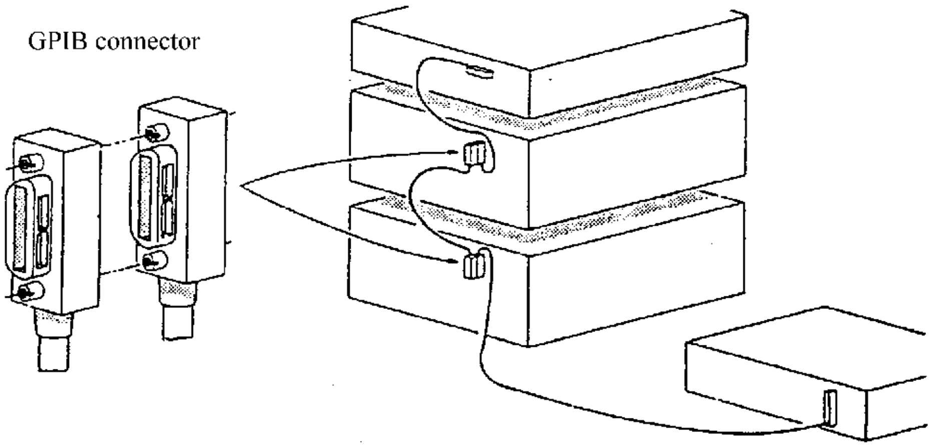


Figure 2-1 GPIB Connection

Note the following when using the GPIB interface:

- Connect the GPIB cable to the BUS 1 connector on the rear panel of this instrument.
- The total cable length of the GPIB cable used in one bus system is not longer than 2 m × {the number of connected devices (the GPIB controller is counted as one device)}.
  The total cable length should be 20 m or less.
- Up to 15 devices can be connected to one bus system.
- There is no restriction in the method of connecting cables. However, four or more GPIB connectors should not be stacked on one device. If four or more GPIB connectors are stacked, the joints of the connectors may be broken because excessive force is applied to them.

  For example, a system consisting of five devices can use cables of up to 10 m (2 m/device × 5 devices = 10 m) in total. Cable lengths can be allocated freely unless the total cable length exceeds the permitted maximum length. When 10 or more devices are to be connected, however, some devices should be connected with cables of 2 m or less so that the total cable length does not exceed 20 m.

2. GPIB address setting

   GPIB addresses should be set from the GPIB Address in the System menu.

2.2.3 GPIB Bus Functions

2.2.3.1 GPIB Interface Functions

Table 2-1 GPIB Interface Functions

Code	Description
SH1	Has the source handshake function
AH1	Has the accepter handshake function
T6	Basic talker function, serial polling function, listener-specified talker cancel function
TE0	No extended talker function
L4	Basic listener function, talker-specified listener cancel function
LE0	No extended listener function
SR1	Has service request function
RL1	Remote function, local function, local lockout function
PP0	No parallel polling function
DC1	Device clear function
DT1	Device trigger function
C1	System controller function
C2	IFC transmission, controller-in-charge function
C3	REN transmission function
C4	SRQ response function
C12	Interface message transmission function, pass control back function
E1	Using the open-collector bus driver

2.2.3.2 Responses to Interface Messages

The responses of this instrument to interface messages explained in this section are defined in IEEE standards 488.1-1987 and 488.2-1987.

For information on the method of sending interface messages to this instrument, refer to the operation manual of the controller used.

- Interface clear (IFC)

  This message is directly sent to this instrument through a signal line.

  With this message, this instrument stops the operation of the GPIB bus. Though all input/output is stopped, the I/O buffer is not cleared (it is cleared by DCL). When this instrument is defined as an active controller, the control right of the GPIB bus is canceled and the system controller gets the control right.

- Remote enable (REN)

  This message is directly sent to this instrument through a signal line.

  If this instrument is specified as a listener when this message is TRUE, it enters the remote state.

  This state continues until this instrument receives GTL, REN is changed to FALSE, or the LOCAL key is pressed.

  This instrument ignores all the received data when it is in the local state.

  When it is in the remote state, this instrument ignores all key entry except the LOCAL key.

  When it is in local lockout state (refer to "Local lockout (LLO)"), this instrument ignores all key entry.

- Serial port enable (SPE)

  When receiving this message from outside, this instrument enters the serial polling mode.

  When this instrument specified as a talker in this mode, it sends status bytes instead of ordinary messages. This mode continues until this instrument receives a serial polling disable (SPD) message or an IFC message.

  When this instrument is sending a service request (SRQ) message to the controller, bit6 (RQS bit) of response data is set to 1 (TRUE). After transmission is completed, RQS bit is set to 0 (FALSE). A service request (SRQ) message is directly sent through a signal line.

- Device clear (DCL)

  When receiving DCL, this instrument performs the following operations:

  - Clearing the input and output buffers
  - Resetting the syntax analysis, execution control, and response data generation units
  - Canceling all the commands that impede the remote command to be executed next
  - Canceling the command that is temporarily stopped to wait for other parameters
  - Canceling *OPC and *OPC?

  The following operations are not executed:

  - Changing data set or stored in this instrument
  - Interrupting the front panel operation
  - Affecting or interrupting the operation of this instrument during execution
  - Changing the status byte except MAV (MAV is set to 0 as the result of clearing the output buffer)

- Selected device clear (SDC)

  Performs the same operation as DCL. However, SDC is executed only when this instrument is a listener. In other cases, it is ignored.

- Go to local (GTL)

  This message sets this instrument to the local state. In the local state, all the front panel operations are enabled.

- Local lockout (LLO)

  This message sets this instrument to the local lockout state. When this instrument enters the remote state in this state, all front panel operation is prohibited (In the ordinary remote state, the front panel operation can be performed with the LOCAL key).

  In this case, this instrument can be set to the local state by any of the following three methods:

  - Sending a GTL message to this instrument
  - Setting the REN message to FALSE (At this time, the local lockout state is also canceled)
  - Turning on the power again

  If this instrument receives this message when it is set to a talker, it is set to an active controller by path control. When receiving an IFC message, this instrument enters the addressable mode again.

2.3 LAN Remote Control System

The LAN (Local Area Network) interface that is compliant with IEEE standard 802.3 comes standard with this instrument so that remote control by socket communication between the external controller and this instrument can be performed.

The controlling method using the LAN remote control function is explained below.

2.3.1 Setting up LAN

1. LAN connection

   The standard LAN connection is shown in Figure 2-2. To perform communication through LAN between an external controller and this instrument or other devices, connect them with the 10BASE-T LAN cable of the RJ45 connector. To directly connect this instrument and an external controller with a LAN cable, use a LAN cable (cross over cable) having connection as shown in Table 2-2. To connect this instrument and devices other than an external controller with a LAN, use an external device designed to connect devices having two or more LAN interfaces such as an Ethernet hub. The LAN cable used in this case is a LAN cable (straight cable) having connection as shown in Table 2-2.

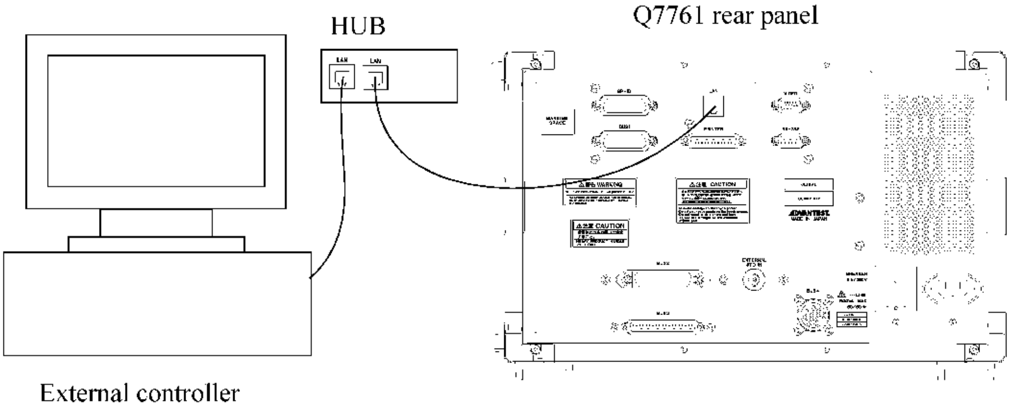


Figure 2-2 LAN Connection

Table 2-2 Connection of 10BASE-T Straight Cables

Signal name	RJ45 Pin number	Line color	Pair number
RX+	1	White/Orange	2
RX-	2	Orange	
TX+	3	White/Green	3
TX-	6	Green	
Not Used	4	Blue	1
	5	White/Blue	
	7	White/Brown	4
	8	Brown	

2.3.2 IP Address Setting

The IP address should be set from the Network Setup in the System menu.

2.3.3 Control from a Program

To control this instrument from a program of an external controller, a port number for socket communication is required. The port number "5025" is provided for socket communication for the remote control on the side of this instrument. To write a program for socket communication, a library for network connection with the TCP/IP protocol is required. The library differs depending on the environment, such as the OS of the external controller. In the Windows OS environment, for example, WinSock is provided.

Of the functions available in the GPIB remote control system, some functions specific to the GPIB bus, such as service request, cannot be used in the LAN remote control system.

2.4 Message Exchanging Protocol

This instrument receives program messages from the controller or other devices through the GPIB bus or LAN and generates response data. Program messages include commands, queries (which are commands that ask for response data) and data.

2.4.1 Buffers

This instrument has three buffers.

- Input buffer

  A buffer for storing data temporarily to analyze commands.

  (1024-byte length)

  The input buffer can be cleared by the following two methods:

  - Power on
  - Executing DCL or SDC
- Output buffer

  A buffer for storing data until data is read by the controller.

  (1024-byte length)

  The output buffer can be cleared by the following two methods:

  - Power on
  - Executing DCL or SDC
- Error queue

  The error queue exists only in IEEE488.2-1987 command mode.

  This is a queue that stores error messages of remote commands, and its depth is 10.

  Each time an error occurs during analysis or execution of remote commands, messages are stacked in the queue.

  Messages can be read with the SYST:ERR command. Each time a message is read, it is deleted from the queue.

  The error queue can be cleared by the following two methods:

  - Power on
  - Executing *CLS

2.4.2 IEEE488.2-1987 Command Mode

The IEEE488.2-1987 command mode sends and receives messages in accordance with the message exchange protocol that is compliant with IEEE standard 488.2-1987.

When other controllers or devices receive messages from this instrument in this mode, the following items are especially important:

- Generating response data by receiving a query (Refer to "Parser").
- Generating data in the order queries are executed (Refer to "Generating response data").

- Parser

  The parser receives command messages in the order they are received from the input buffer, executes syntax analysis, and determines what operations are to be executed by the commands received.

  It also traces the tree structure of commands when performing the syntax analysis of commands.

  For the next command analysis, it remembers the part of the tree structure from which analysis should be performed.

  This information is returned to the head of the tree structure when the parser is cleared.

  The parser can be cleared by the following four methods:

  - Power on
  - Receiving DCL or SDC
  - Receiving ":" next to ";"
  - Receiving the terminator or EOI

- Generating response data

  When the parser executes a query, this instrument generates data on the output buffer as its response (that is, a query must be sent immediately before outputting data).

  It means that data is not cleared until the controller reads data generated by a query.

  Besides the controller reading data, there are two conditions in which data is cleared. In these conditions, a Query Error is generated.

  - Unterminated condition:

    When the controller reads response data without terminating the query (with ASCII LF code or GPIB END message) or when the controller reads response data without sending a query.

  - Interrupted condition:

    When the controller receives the next program message before reading response data.

2.5 Command Syntax

This chapter describes the command syntax.

2.5.1 IEEE488.2-1987 Command Mode

The command syntax is defined in the following format:

Header → Space (space character) ⇨ Data

MEMO: ⇨ ***means repetition.***

1. Header

   The header has a layered structure consisting of two or more mnemonics delimited by colons (:). A mnemonic consisting of four or more characters has a "short form" of four (or three) characters (a non-abbreviated mnemonic is called a "long form"). Any combination of these forms is allowed.

   If a question mark (?) is attached immediately after a header, it becomes a query command.

2. Space (space character)

   A space of one or more characters is required. If any character except a space is used, an error will result.

3. Data

   When the command requires more than one data item, list these data items by delimiting them with commas (,).

   A space (space character) may be inserted before or after the comma (,).

   For more information on data type, refer to "2.5.2 Data Format."

4. Writing more than one command

   In IEEE488.2-1987 command mode, more than one command may be written on one line by delimiting them with semicolons (;).

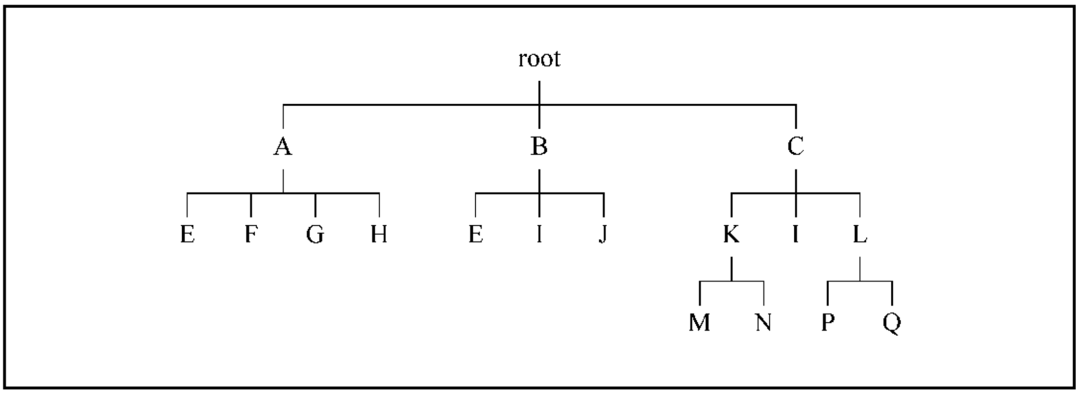
   When commands are written in this way, the system executes commands while moving the current path in the layered structure of the header.

5. Moving the current path

   The current path moves in accordance with the following rule:

   - When power is turned on: The current path is set to root.
   - Terminator: The current path is set to root.
   - Colon (:): If the colon (:) that moves the current path to one layer below in the command tree is the initial character, the colon (:) sets the current path to root.
   - Semicolon (;): The current path is not changed.
   - Common command: This command can be executed regardless of the current path. When an *RST command is executed, the current path is set to root (* See the following example).

Example: Assume the following header structure:



In the above structure, the current path moves in the following way:

1. :A:E;:B:E

   Since the colon (:) in the second command moves the current path to root, both A:E and B:E are correct commands.

2. :A:E<END>B:E

   Since <END> (terminator) moves the current path to root, both A:E and B:E are correct commands.

3. :A:E;F;G;H

   Since a semicolon (;) does not move the current path, :A:E;F;G;H results in the four commands A:E, A:F, A:G, and A:H.

4. :C:I;K:N;M

   Since a colon (:) moves the current path, K:N is seen from the layer of :C:.

   Therefore, K:N is equal to C:K:N. At the same time, the current path is changed to :C:K: because K:N contains a colon (:) and the last M is treated as C:K:M.

5. :A:E;*ESR 16

   Since the common command is independent of the current path, *ESR 16 is executed correctly.

6. :A:E;*ESR 16;F;G;H

   Since the common command does not change the current path, the third F is searched for from the current path :A: that is set in the first :A:E.

   Therefore, F, G, and H are equal to A:F, A:G, and A:H, respectively.

In the following examples, a syntax error will result.

1. :A:E;B:E

   A:E changes the current path to :A:.

   Therefore B:E is searched for from the layer of :A: but the mnemonic B cannot be found. As a result, an error will result.

2. :C:K:M;L:P

   :C:K:M changes the current path to :C:K:.

   Therefore L:P is searched for from :C:K: but the mnemonic L cannot be found. As a result, an error will result.

2.5.2 Data Format

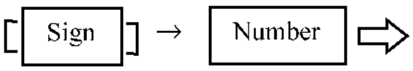
In IEEE488.2-1987 command mode, the data types shown in this section are used in data input/output.

1. Numeric data

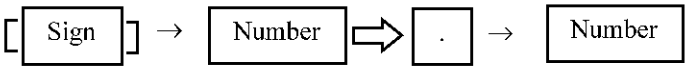
   There are three formats for numeric data as shown below. When entering numeric values for this instrument, any format may be used (the value is rounded depending on the data type entered).

   Depending on the command, a unit may be attached to the entered numeric value. For units, see Section (5), which is described later.

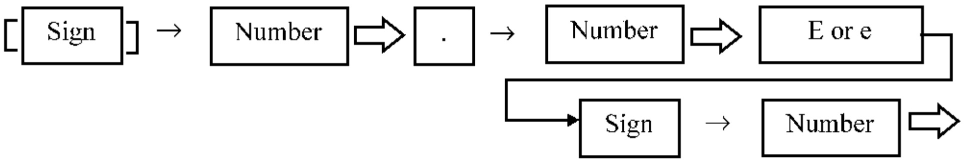
   - Integer type: NR1 format



   - Fixed point type: NR2 format



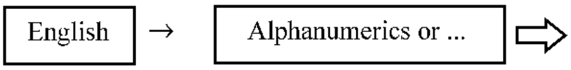
   - Floating point type: NR3 format



MEMO: ⇨ ***means repetition.***
The sign at the head may be omitted.

2. Character data

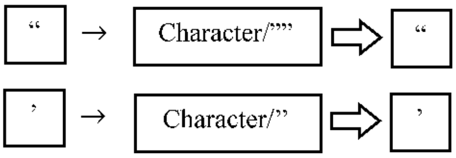
   The format of character data is shown below:



MEMO: ⇨ ***means repetition.***

3. String data

   There are two formats for string data.



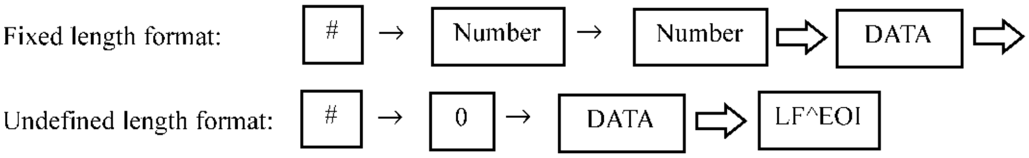
In string data, ASCII 7bit code characters can be used.

MEMO: ***In string data starting with ”, ” should be represented as ””.***
In string data starting with ’, ’ should be represented as ’’.
⇨ means repetition.

If response data is string data, string data starting with ” is always output.

4. Block data

   There are two formats for block data. When entering data into this instrument, either format may be used.



MEMO: ***⇨ means repetition.***

In fixed length format, one character number following # indicates the number of digits for the subsequent bytes. 0 cannot be used (because it becomes the undefined length format).

Example: Block data #3128<data byte>

The number 3 following # indicates the number of digits of the subsequent string (128), and the number 128 represents the number of bytes of the subsequent <data byte>.

5. Unit

The unit is a suffix following the numeric value. For units, a suffix can be used as a prefix. Available suffixes and units are listed below:

Table 2-3 Available Units

Unit	Description
m	Distance unit
Hz*	Frequency unit
dB	Level unit
s	Time unit
PCT	Percent

Table 2-4 Available Suffixes

Available Suffixes	
1E18	EX
1E15	PE
1E12	T
1E9	G
1E6	MA
1E3	K
1E-3	M *
1E-6	U
1E-9	N
1E-12	P
1E-15	F
1E-18	A

*: If the unit is HZ, the suffix is 1E6 (equivalent to MA).

2.6 Status Bytes

This instrument has a layered status register structure that is compliant with IEEE standard 488.2-1987, and can send various statuses of the devices to the controller. This section describes the behavioral model of status bytes and allocation of events.

1. Status registers

   This instrument adopts the model of the status registers defined in IEEE standard 488.2-1987. The status registers consist of the condition register, event register, and enable register.

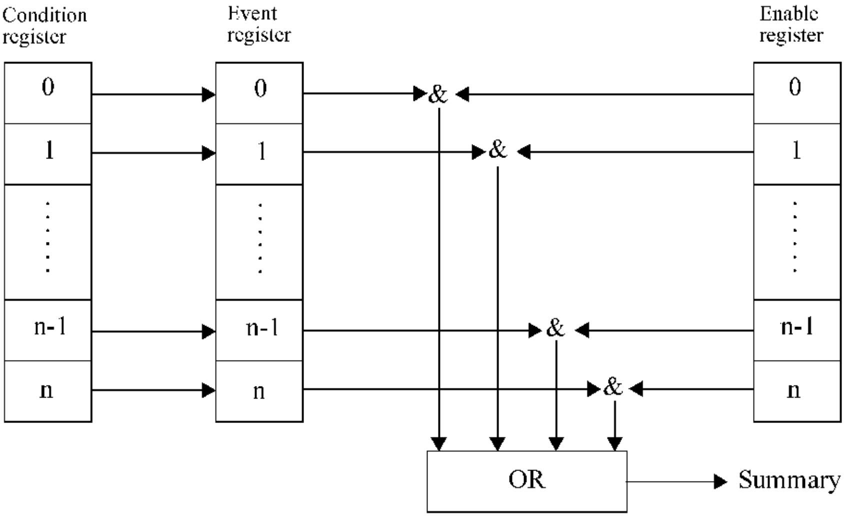


Figure 2-3 Status Register Configuration

a. Condition register

   The condition register is always monitoring the status of the device. That is, this register always retains the latest status of the device.

   However, the condition register cannot read and write data because it retains data as internal information.

b. Event register

   The event register latches and retains the status from the condition register (or retains change).

   Once this register is set, the setting value is kept until it is read by a query or cleared by *CLS.

   Data cannot be written to the event register.

c. Enable register

   The enable register specifies which bit in the event register is set as an effective status to generate a summary. The enable register is ANDed with the event register and the OR of the result is generated as a summary. The summary is written to the status byte register.

   Data can be written to the enable register.

This instrument has the following five types of status registers:

- Status byte register
- Standard event register
- Standard operation status register

The detail of the status registers is shown in Figure 2-4.

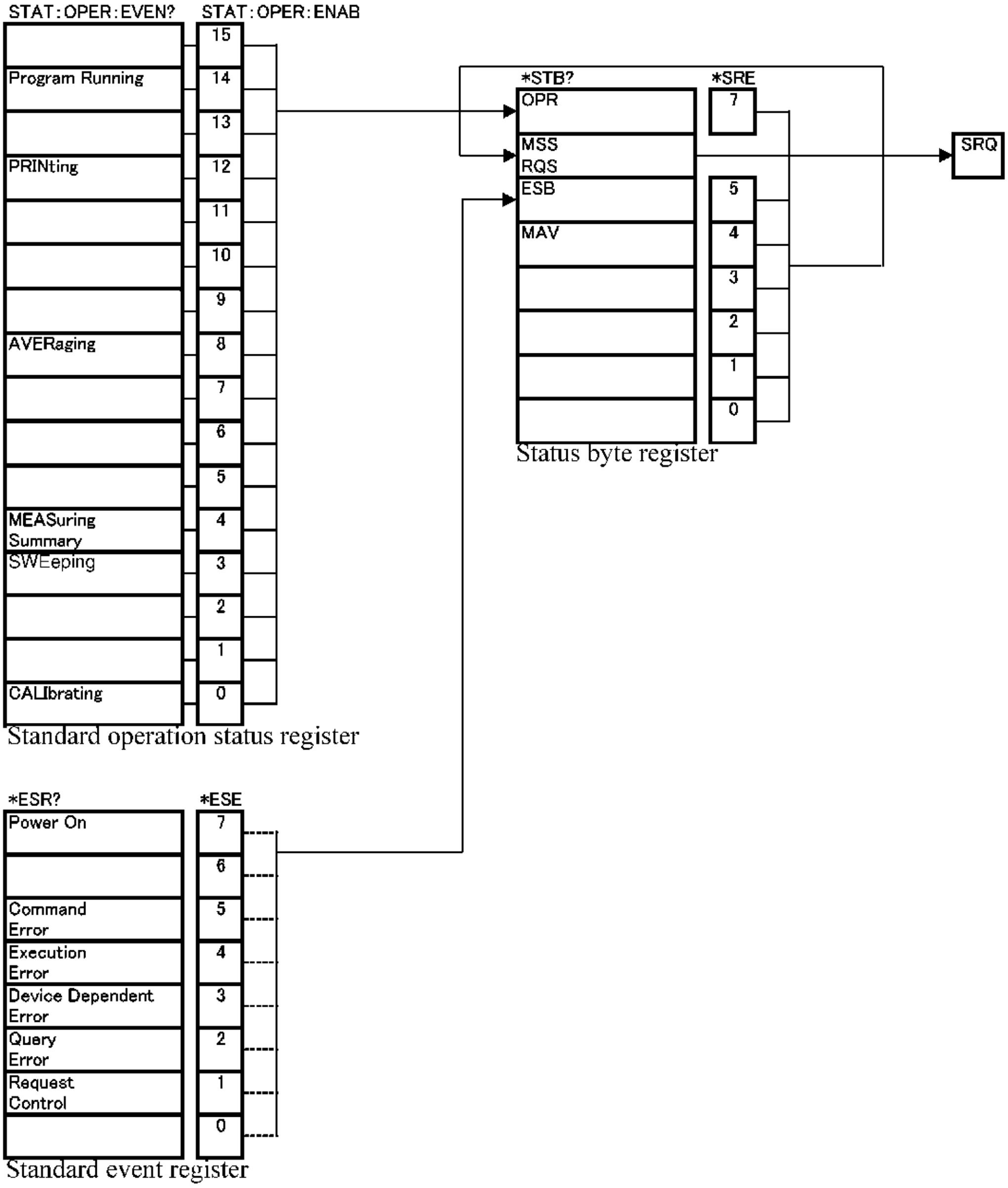


Figure 2-4 Status Register Detail

2. Event enable register
   Each event register has an enable register that determines which bit is to be enabled.
   - Service request enable register set: *SRE
   - Standard event status enable register set: *ESE
   - Operation status enable register set: OPR
3. Standard operation status register
   Allocation in the standard operation status register is listed in Table 2-5:

Table 2-5 Allocation in the Standard Operation Status Register

bit	Function definition	Description
15	-	Always 0
14	-	Reserved
13	-	Always 0
12	-	Always 0
11 to 9	-	Always 0
8	AVERaging	Set to 1 when averaging is completed.
7 to 5	-	Always 0
4	MEASuring Summary	Set to 1 depending on the status of the measuring status register.
3	SWEeping	Set to 1 when sweep is completed.
2	-	Always 0
1	-	Always 0
0	CALIbrating	Set to 1 when correction data acquisition is completed.

4. Status byte register

The status byte register summarizes the information from the status register.

The summary of this status byte register is sent to the controller as a service request. Therefore, the status byte register operates slightly differently than the status register structure.

This section describes the status byte register.

The structure of the status byte register is shown in Figure 2-5.

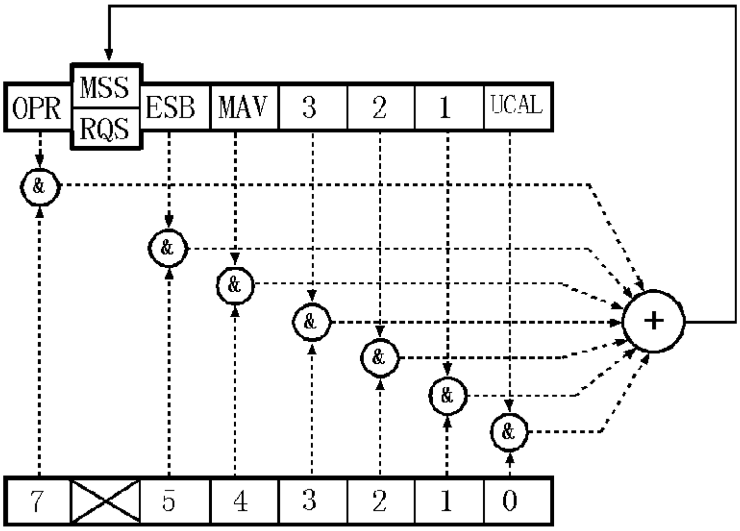


Figure 2-5 Structure of Status Byte Register

This status byte register follows the status register except for the following three points:

- The summary of the status byte register is written to bit6 of the status byte register.
- The bit6 of the enable register is always enabled and it cannot be changed.
- The bit6 (MSS) of the status byte register writes RQS of the service request.

This register responds to the serial polling from the controller. When responding to the serial polling, bit0 to bit5, bit7, and RQS of the status byte register are read, and then RQS is reset to 0. The other bits are not cleared until each factor is set to 0.

The status byte register, RQS, and MSS can be cleared by executing "*CLS". When they are cleared, the SRQ line is also set to FALSE.

The meaning of each bit in the status byte register is shown in Table 2-6:

Table 2-6 Meaning of Status Byte Register

bit	Function definition	Description
7	OPR	The OPR is the summary of the standard operation status register.
6	MSS	The RQS is set to TRUE when the MSS of the status byte register is set to 1, and the MSS is the summary bit of all of the status data structure. The MSS cannot be read in serial polling (but it is known that the MSS is 1 when RQS is 1). To read the MSS, use the common command *STB?. With *STB?, bit0 to bit5, bit7, and the MSS of the status byte register are read. In this case, the status byte register and the MSS are not cleared. The MSS is not set to 0 until all the unmasked factors in the status register structure are cleared.
5	ESB	The ESB is the summary of the standard event register.
4	MAV	The summary bit of the output buffer. It is set to 1 when there is output data in the output buffer and set to 0 after data is read.
3 to 0		Always 0

5. Standard event register

Allocation in the standard event register is listed in Table 2-7:

Table 2-7 Allocation in the Standard Event Register

bit	Function definition	Description
7	Power on	Set to 1 when power is turned on.
6	-	Always 0
5	Command Error	Set to 1 when the parser detects a syntax error.
4	Execution Error	Set to 1 when execution of an instruction received as a GPIB command fails for some reason (e.g., the parameter is out of range).
3	Device Dependent Error	Set to 1 when an error except Command Error, Execution Error, and Query Error occurs.
2	Query Error	Set to 1 when no data exists or data is lost when the controller tries to read data from this instrument.
1	Request Control	Set to 1 when this instrument is required to be an active controller.
0	Operation Complete	Set to 1 when there is no command that is being executed for this instrument after the *OPC command is received.

3. EXAMPLE REMOTE CONTROL PROGRAMS

This chapter provides example programs, which allow this instrument to be controlled by remote.

The example programs in this chapter use the Microsoft Corp. Visual Basic language. If programming in another language, an appropriate description that suits each language must be given.

The explanation of the programs here assumes that the GPIB board provided by the National Instruments Corp. (hereafter referred to as the NI Corp.) is used as a GPIB controller.

3.1 Basic Steps for the GPIB Control

This section describes the operations required to control the GPIB from Visual Basic.

For Visual Basic dependent operations such as initializing the variables and defining the routine functions, the notational rules for the Visual Basic program must be applied.

3.1.1 Loading the GPIB Control Library for Visual Basic

To control the GPIB board provided by the NI Corp. by using the program described in the Visual Basic language, the two files must be built into the Visual Basic Project. Those two files are the VBIB-32.BAS file, in which the GPIB communication interface for the Visual Basic language provided by the NI Corp. is described, and the NIGLOBAL.BAS file, in which errors and timeout values are defined.

3.1.1.1 Initializing the Controller

To communicate with this instrument through the GPIB, the GPIB controller must be initialized.

The following program shows how to initialize the GPIB controller:

```
Rem ----------------- Initialize GPIB Controller ----------------------------
Public Sub InitGPIB()

onaaddress% = 8

Call ibfind ("GPIB0", boardID%)         ' Open GPIB board
Call ibfind ("DEV1", analyzer%)         ' Open analyzer port
Call ibpad( analyzer%, onaaddress%)     ' Set the ONA's GPIB address

Call ibtmo( analyzer%, 12)              ' Set timeout value to 3 sec

End Sub
```

3.1.1.2 Initializing the Q7761 Optical Network Analyzer

The following program shows how to initialize this instrument prior to the GPIB control.

```
Rem ----------------- Initialize Optical Network Analyzer -----------------------------
Public Sub InitONA( )

Call ibwrt ( analyzer%, "*CLS" )          ' Reset status register
Call ibwrt ( analyzer%, "*RST" )          ' Reset this instrument

End Sub
```

3.1.1.3 Simple Setting Command

The following program shows how to set the center wavelength and span wavelength of this instrument.

```
Rem ----------------- Brief setting of Optical Network Analyzer ---------------------
Public SUB ONASetting( )

Call ibwrt( analyzer%, ":SOUR:CENT 1550NM" ) ' Set center wavelength to 1550nm
Call ibwrt( analyzer%, ":SOUR:SPAN 2NM" )    ' Set span wavelength to 2nm

End Sub
```

3.1.1.4 Reading the Setting Value

In this section, setting values of this instrument are read. The following program shows how to read the center wavelength and span wavelength.

```
Rem ----------------- Read the setting value of Optical Network Analyzer -----------
Public Sub ReadONASetting( )

CF$= Space$(20)                           ' Prepare the text variable for read
Call ibwrt( analyzer%, ":SOUR:CENT?" )    ' Read request of center wavelength
Call ibrd( analyzer%, CF$ )               ' Read setting value
SP$= Space$(20)                           ' Prepare the text variable for read
Call ibwrt( analyzer%, ":SOUR:SPAN?" )    ' Read request of span wavelength
Call ibrd( analyzer%, SP$ )               ' Read setting value

Rem ------------------- Display setting value -------------------------------------
Call MsgBox ( "Center wavelength.: "& CF$ )
Call MsgBox ( "Span wavelength.: "& SP$ )

End Sub
```

3.1.1.5 Executing the Measurement and Reading its Result

This section describes how to synchronize this instrument and GPIB controller by using programs, which perform measurements and read results, as examples.

Common commands used for synchronization

The *WAI, *OPC?, and *OPC commands are commonly used to synchronize this instrument and the controller when various commands are executed. Example programs, which use these common commands, are shown below and on the following pages. Those example programs read result data and display a cursor after the sweeping operation ends.

Example 1 *WAI command
If the *WAI command is used, any subsequent commands are paused until all commands which precede the *WAI command are complete. During the sweep, the subsequent commands are not executed until the sweeping operation ends.

```
Rem -----------------Get level data after sweeping---------------
Public Sub GetPoint1( )

Call ibwrt( analyzer%, ":ABOR")              ' Stop sweeping
Call ibwrt( analyzer%, ":INIT:IMM" )         ' Start sweeping
Call ibwrt( analyzer%, "*WAI")               ' Wait for end of sweep

CursLevel$= Space$(128)
Call ibwrt(analyzer%, ":CURS:X1 ON")         ' Turn on X1 cursor
Call ibwrt(analyzer%, ":CURS:X1:WAV 1550NM") ' Move X1 cursor to 1550nm
Call ibwrt(analyzer%, ":CURS:X1:DATA?")      ' Query X1 cursor level
Call ibrd( analyzer%, CursLevel$ )           ' Read it
Rem ------------------- Display setting value ------------------------------
Call MsgBox ( "Get Cursor level after sweeping := "& CursLevel$ )

End Sub
```

Example 2 *OPC? command

The OPC? command responds with '1' when all the commands which were sent before the OPC? command are complete. During the sweep, this query response is not performed until the sweeping operation ends. If the subsequent command is executed after the query data '1' is received, this instrument can be synchronized with the controller. This example shows that the synchronization is enabled by waiting for Operation Complete.

The *OPC? command responds after a period of time which depends on the sweep time. Therefore, to prevent a timeout occurring in the ibrd execution, set sufficient time for the timeout or loop the ibrd and check the timeout.

```
Rem ----------------- Get level data after sweeping---------------
Public Sub GetPoint2( )

Call ibwrt( analyzer%, ":ABOR")               ' Stop sweeping
Call ibwrt( analyzer%, ":INIT:IMM" )          ' Start sweeping

OPEND$ = Space$(3)
Call ibwrt( analyzer%, "*OPC?")               ' Request Operation complete
Call ibrd( analyzer%, OPEND$)                 ' Read status

CursLevel$= Space$(128)
Call ibwrt(analyzer%, ":CURS:X1 ON")          ' Turn on X1 cursor
Call ibwrt(analyzer%, ":CURS:X1:WAV 1550NM")  ' Move X1 cursor to 1550nm
Call ibwrt(analyzer%, ":CURS:X1:DATA?")       ' Query X1 cursor level
Call ibrd( analyzer%, CursLevel$ )            ' Read it
Rem -------------------- Display setting value ------------------------------
Call MsgBox ( "Get Cursor level after sweeping := "& CursLevel$)

End Sub
```

Example 3 Synchronizing by using *OPC? command when SRQ is generated.

If the *OPC command is executed, the "Operation Complete" bit of the standard event status register is set to 1 when all the commands, which are sent before the *OPC command, are completed. During the sweep, "Operation Complete" is set to 1 after the sweeping operation ends. The instrument and the controller can be synchronized when the command, which follows the SRQ generation, is executed in the controller.

```
Rem ----------------- Get level data after sweeping---------------
Public Sub GetPoint3( )

Call ibwrt( analyzer%, "*SRE 32" )          ' Set SRQ for ESR to enable
Call ibwrt( analyzer%, "*ESE 1" )           ' Set enable bit for OPC

Call ibwrt( analyzer%, ":ABOR")             ' Stop sweeping
Call ibwrt( analyzer%, "*CLS" )             ' Clear status
Call ibwrt( analyzer%, "*OPC" )             ' Send OPC for synchronization
Call ibwrt( analyzer%, ":INIT:IMM" )        ' Start sweeping

Call WaitSRQ( boardID%, res% )              ' Wait for SRQ using driver's func.
Call ibrsp( analyzer%, stb% )               ' Execute serial poll

CursLevel$= Space$(128)
Call ibwrt(analyzer%, ":CURS:X1 ON")        ' Turn on X1 cursor
Call ibwrt(analyzer%, ":CURS:X1:WAV 1550NM") ' Move X1 cursor to 1550nm
Call ibwrt(analyzer%, ":CURS:X1:DATA?")     ' Query X1 cursor level
Call ibrd( analyzer%, CursLevel$ )          ' Read it
Rem -------------------- Display setting value ------------------------------
Call MsgBox ( "Get Cursor level after sweeping := "& CursLevel$ )

End Sub
```

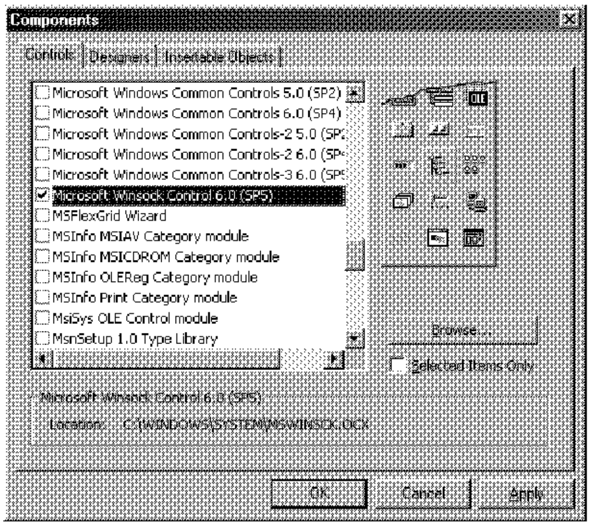
3.2 Basic Steps for the LAN Control

This section describes the operations required to control the LAN interface from Visual Basic.
For Visual Basic dependent operations such as initializing variables and defining routine functions, the notational rules for the Visual Basic program must be applied.

3.2.1 Loading the LAN Control Library for Visual Basic

To control the LAN interface board by using the program described in the Visual Basic language, the Winsock control provided by Microsoft Corp. must be built into the Visual Basic Project.

To use the Winsock control, WINSCK.OCX must be set available in the Visual Basic component settings. The dialog box below shows an example of adding WINSCK.OCX to the Visual Basic 6.0 (Procedure: Select the component (O) submenu from the project (P) menu, check the Microsoft Winsock Control 6.0 in the list of the displayed component dialog box, and click the OK button.)



□ Microsoft Windows Common Controls 5.0 (SP2)
□ Microsoft Windows Common Controls 6.0 (SP4)
□ Microsoft Windows Common Controls-2 5.0 (SP
□ Microsoft Windows Common Controls-2 6.0 (SP
□ Microsoft Windows Common Controls-3 6.0 (SP
☑ Microsoft Winsock Control 6.0 (SP5)
□ MSFlexGrid Wizard
□ MSInfo MSIAV Category module
□ MSInfo MSICDROM Category module
□ MSInfo OLEReg Category module
□ MSInfo Print Category module
□ MsiSys OLE Control module
□ MsnSetup 1.0 Type Library

□ Selected Items Only

Microsoft Winsock Control 6.0 (SP5)
Location: C:\WINDOWS\SYSTEM\MSWINSCK.OCX

By performing the procedure above, the Winsock control objects are displayed in the toolbox. If the Winsock controls are placed in a form, LAN control is enabled by using the placed objects. If the Winsock controls are placed and an object is created, a specific object name is specified. In this manual, the object name tcpClient is used for the following explanations.

3.2.1.1 Opening the Socket Interface (initialization)

To communicate with this instrument through the LAN, the LAN interface must be connected to the ports of this instrument. Before connecting the LAN interface, specify the instrument IP Address (or the host name) in the RemoteHost property and specify the port number in the RemotePort property. For the protocol (Protocol property), specify TCP (sckTCPProtocol). After specifying the above specifications, connect this instrument to the ports by using the Connect method of the Winsock control. The LAN uses port number 5025 to communicate with this instrument.

```
Rem ----------------- Connection LAN Interface ----------------------------
Public Sub ConnectTCP( )

tcpClient.RemoteHost = "192.0.0.1"        ' Set IP Address of ONA
tcpClient.Protocol = sckTCPProtocol       ' Set protocol to TCP
tcpClient.RemotePort = 5025               ' Set port no. 5025 of ONA


tcpClient.Connect                         ' Connect to ONA's port

End Sub
```

MEMO:

1. ***The above connection procedure must be executed before the LAN is controlled. If the above procedure is complete, connection continues until the closing procedure (in the case above, the tcpClient.Close method is executed.) is performed.***
2. ***The closing procedure is automatically performed when the program is closed.***
3. ***If the connection procedure and closing procedure are performed repeatedly, the Winsock control may be damaged. Therefore, only insert one connection procedure into the program.***

3.2.1.2 Initializing the Q7761 Optical Network Analyzer

The following program shows how to initialize this instrument before the LAN control.

```
Rem ----------------- Initialize Optical Network Analyzer ----------------------------
Public Sub InitONA( )

tcpClient.SendData "*CLS" & vbCrLf        ' Reset status register
tcpClient.SendData "*RST" & vbCrLf        ' Reset this instrument

End Sub
```

3.2.1.3 Simple Setting Command

The following program shows simple settings of this instrument.

```
Rem ----------------- Brief setting of Optical Network Analyzer ----------------
Rem ----------------- Set Center wavelength to 1550nm, Span to 2nm -------------
Public Sub ONASetting( )

tcpClient.SendData ":SOUR:CENT 1550NM" & vbCrLf
tcpClient.SendData ":SOUR:SPAN 2NM" & vbCrLf

End Sub
```

3.2.1.4 Reading the Setting Value

The following program shows how to read the setting value of this instrument.

```
Rem ----------------- Read the setting value of Optical Network Analyzer ---------------
Public Sub ReadONASetting( )

CF$= Space$(20)                           ' Prepare the text variable for read
tcpClient.SendData ":SOUR:CENT?" & vbCrLf ' Read request of center wavelength.

Do While (tcpClient.BytesReceived = 0)    ' Wait for receiving a character
    DoEvents
Loop

tcpClient.GetData CF$                     ' Read setting value
SP$= Space$(20)                           ' Prepare the text variable for read
tcpClient.SendData ":SOUR:SPAN?" & vbCrLf ' Read request of span wavelength.

Do While (tcpClient.BytesReceived = 0)    ' Wait for receiving a character
    DoEvents
Loop

tcpClient.GetData SP$                     ' Read setting value

Rem -------------------- Display setting value ------------------------------------
Call MsgBox( "Center wavelength.: " & CF$ & _
             "Span wavelength.: " & SP$)

End Sub
```

3.2.1.5 Performing a Measurement and Reading its Result

The following program shows how to perform a measurement and read its result.

```
Rem -----------------Get level data after sweeping ---------------------------------
Public Sub GetPoint( )

tcpClient.SendData ":ABOR" & vbCrLf                ' Stop sweeping
tcpClient.SendData ":INIT:IMM" & vbCrLf            ' Start sweeping

OPEND$ = Space$(3)
tcpClient.SendData "*OPC?" & vbCrLf
Do While (tcpClient.BytesReceived = 0)             ' Wait for receiving
    DoEvents
Loop
tcpClient.GetData OPEND$

CursLevel$= Space$(128)
tcpClient.SendData ":CURS ON" & vbCrLf
tcpClient.SendData ":CURS:X1 ON" & vbCrLf          ' Turn on X1 cursor
tcpClient.SendData ":CURS:X1:MOVE 1550NM" & vbCrLf ' Move X1 cursor to 1550nm
tcpClient.SendData ":CURS:X1:DATA?" & vbCrLf       ' Query X1 cursor level
Do While (tcpClient.BytesReceived = 0)             ' Wait for receiving
    DoEvents
Loop
tcpClient.GetData CursLevel$

Call MsgBox("Get Cursor level after sweeping : " & CursLevel$)

End Sub
```

4. COMMAND REFERENCE

This chapter describes the command reference for this instrument.

4.1 Command Reference Format

This section describes the format of explanations of each command described in this chapter.

Explanations of each command include the following items:

- Function description
- Command syntax
- Parameter
- Query reply

- [Function description]

  The usage of commands and operation of this instrument when they are executed.

- [Command syntax]

  The command syntax shows the syntax of a command sent from the external controller to this instrument. The syntax consists of a command part and a parameter part. The command part and parameter part are delimited by a space.

  When there are multiple parameters, they are delimited by commas (,).The three points (...) displayed between commas represent the parameter(s) omitted in the position.

  For example, the description <real 1>,..., <real 4> shows that four parameters,<real 1>, <real 2>, <real 3>, and <real 4>, are required.

  When the parameter is a character string type such as" <str>,<str 1>, the parameter must be enclosed in double quotation marks ("). If the parameter is <block>, the block format data is required.

  The part written in lowercase alphabetical characters in the syntax shows that it can be omitted.

  For example, ":CALibration:CABLe" can be abbreviated to ":CAL:CABL".

  The marks used in the syntax are defined as follows:

  If the parameter in the header part is omitted, the parameter is treated as 1.

< >	Shows a parameter required for sending a command
[ ]	Shows that the command is optional It can be omitted
{}	Shows that only one item is required to be selected from multiple items
|	Written in curly brackets {..} and used as a delimiter for multiple items

For example, when a syntax below is specified, :CALC:CORR:EDEL:TIME 0.1 and :CALCULATE1:SELECTED:CORR:EDEL:TIME 25E-3 are valid.

Syntax: CALCulate{[1]|2|3|4}[:SELected]:CORRection:EDELay:TIME <numeric value>

- [Parameter]

  Describes a parameter required for sending a command.

  When the parameter is a numeric type or a character (string) type, it is enclosed in angle brackets (< >).

  When the parameter is an optional type, it is enclosed in curly brackets { }. Square brackets indicate parameters that can be omitted when a command is sent.

  In this manual, parameter types are described in the following formats:

  < int >: A numeric value that can be input in the format NR1, NR2, or NR3 and rounded to an integer in this instrument

  < real >: A numeric value that can be input in the format NR1, NR2, or NR3 and rounded to a valid-digit real number in this instrument

  <realX>: X axis real value (Numeric value + Unit (can be specified))
  Wavelength: Numeric value + [*m] (* = U, N, P)
  Frequency: Numeric value + [*Hz] (* = T, G, M, K)

  <realY>: Y axis real value (Numeric value + (Unit (can be specified))
  Trace mode LogMag/PDL: Numeric value + [dB]
  Trace mode LinMag/GD/CD/CDSL/PMD/2ndPMD: Numeric value + [P|N|U]

  < bool >: Character string of OFF or ON, or value of 0 or 1

  < str >: A character string or alphanumeric symbols enclosed in quotation (‘ ’) or double quotation (“ ”) marks

  < block >: Block data type
  The content of data is an 8-bit binary data array

  ---: No specified parameter

  ×: Nonusable

  ←: Same as the command parameter

- [Query reply]

  When there is a query reply to the command, the data format used for reading the query is described.

  The same format as [parameter] is used to describe the query reply.

  Square brackets indicate parameters that can be omitted in query replies, depending on the measurement result.

4.2 Common Commands

Function	Command syntax	Parameter	Query reply	Remarks
Clears the status byte and related data	*CLS	---	×	
Sets the standard event status enable register	*ESE	<int>	←	
Reads the standard event status register	*ESR?	×	<int>	
Device inquiry	*IDN?	×	<str>	*1
Notice of completion of running operations	*OPC	---	1	
Recalls device settings	*RCL	<int>	×	
Resets the device	*RST	---	×	
Saves the device settings	*SAV	<int>	×	
Sets the service request enable register	*SRE	<int>	←	
Reads the status byte register	*STB?	×	<int>	
Self-test execution and inquiry for the results	*TST?	×	<int>	
Waits for the completion of all running operations	*WAI	---	×	

*1: <str> is output in the following order: maker name, model name, serial number, and version number.

4.3 ABORt Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Measurement abort	:ABORt	---	×	

4.4 CALCulate Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Waveform display				
Display mode	:CALCulate:TRACe<1|2|3|4>:FORMat	{MAGNitude|GDELay|CD|CDSL|PMD|SNDPMD|PDL}	{MAGN|GDEL|CD|CDSL|PMD|SNDPMD|PDL}	
MAG linear display	:CALCulate:MAGLinear[:STATe]	<bool>	{ON|OFF}	
Port				
Port	:CALCulate:TRACe<1|2|3|4>:PORT	{1|2}	←	
Normalize				
ON/OFF	:CALCulate:NORMalize:PORT<1|2>[:STATe]	<bool>	{ON|OFF}	
Data save	:CALCulate:NORMalize:PORT<1|2>:SAVE	---	×	
Fitting ON/OFF	:CALCulate:NORMalize:PORT<1|2>:FITTing[:STATe]	<bool>	{ON|OFF}	
Fitting function	:CALCulate:NORMalize:PORT<1|2>:FITTing:FUNCtion	{LINear|QUAD|SELlMeier3|SELlMeier5}	{LIN|QUAD|SELM3|SELM5}	
Smoothing				
ON/OFF	:CALCulate:TRACe<1|2|3|4>:SMOothing[:STATe]	<bool>	{ON|OFF}	
Window	:CALCulate:SMOothing:WINDow	<realX>	←	
Ripple				
ON/OFF	:CALCulate:TRACe<1|2|3|4>:RIPPle[:STATe]	<bool>	{ON|OFF}	
Fitting function	:CALCulate:RIPPle:FITTing:FUNCtion	{LINear|QUAD|SELlMeier3|SELlMeier5}	{LIN|QUAD|SELM3|SELM5}	
Fitting				
ON/OFF	:CALCulate:TRACe<1|2|3|4>:FITTing[:STATe]	<bool>	{ON|OFF}	
Fitting function	:CALCulate:TRACe<1|2|3|4>:FITTing:FUNCtion	{LINear|QUAD|SELlMeier3|SELlMeier5}	{LIN|QUAD|SELM3|SELM5}	
Measurement value dot display	:CALCulate:TRACe<1|2|3|4>:FITTing:DOT[:STATe]	<bool>	{ON|OFF}	
Partial fitting	:CALCulate:TRACe<1|2|3|4>:FITTing:PARTial[:STATe]	<bool>	{ON|OFF}	
Coefficient	:CALCulate:TRACe<1|2|3|4>:FITTing:COEFficient <1|2|3|4|5>?	×	<real>	
Fitting error	:CALCulate:TRACe<1|2|3|4>:FITTing:ERRor?	×	<real>	
Zero-dispersion wavelength	:CALCulate:TRACe<1|2|3|4>:FITTing:ZERolambda?	×	<real>	
/km, /√km				
/km and /√km selection	:CALCulate:TRACe<1|2|3|4>:PerKM	{0|1|2}	←	*1

*1: Setting values; 0: OFF, 1: /km, 2:/√km

Function	Command syntax	Parameter	Query reply	Remarks
Unwrap				
Level setting	:CALCulate:UNWRap:LEVel	<real>[dB]	←	
MAG level offset				
Offset value setting	:CALCulate:MAGNitude:LOFFset	<real>[dB]	←	
Limit line				
Judgement execution	:CALCulate:LIMit:JUDGe	---	{PASS|FAIL}	
Line 1 ON/OFF	:CALCulate:LIMit:UPPer[:STATe]	<bool>	{ON|OFF}	
Line 2 ON/OFF	:CALCulate:LIMit:LOWer[:STATe]	<bool>	{ON|OFF}	
Line 1 Pass range	:CALCulate:LIMit:UPPer:PASS	{ABOVe|BELow}	{ABOV|BEL}	
Line 2 Pass range	:CALCulate:LIMit:LOWer:PASS	{ABOVe|BELow}	{ABOV|BEL}	
Limit line edit				
Trace mode selection	:CALCulate:LIMit:CONTrol:Y:TRACe	{MAGNitude|GDELay|CD|CDSL|PMD|SNDPMD|PDL}	{MAGN|GDEL|CD|CDSL|PMD|SNDPMD|PDL}	
Line 1 data input	:CALCulate:LIMit:UPPer:DATA	<realX>, <realY>	×	
Line 2 data input	:CALCulate:LIMit:LOWer:DATA	<realX>, <realY>	×	
Line copy 1 → 2	:CALCulate:LIMit:UPPer:COPY	---	×	
Line copy 2 → 1	:CALCulate:LIMit:LOWer:COPY	---	×	
Line 1 data clear	:CALCulate:LIMit:UPPer:DELete	---	×	
Line 2 data clear	:CALCulate:LIMit:LOWer:DELete	---	×	
Half bandwidth				
Bandwidth execution	:CALCulate:BANDwidth	---	<real1>,..., <real5>	*2
Bandwidth XdB setting	:CALCulate:BANDwidth:NDB	<real>[dB]	<real>	
Band/Notch width clear	:CALCulate:BANDwidth:CLEar	---	×	
Statistics analysis				
Execution	:CALCulate:STATistics	---	<real1>,..., <real7>	*3

*2: The query reply is output in the following order: Band width, center, left line, right line, and X line.

*3: The query reply is output in the following order: Max Data, Min Data, Average, RMS, STD. deviation, Maxwell α, and M.P. value.

Function	Command syntax	Parameter		Query reply	Remarks
Data output					
Data point count output	:CALCulate:POINts	×	<int>	<int>	*4
Data output	:CALCulate:DATA	×	<int>	<real1>,..., <real n>	*4
Data output for each polarization					
Data point count output	:CALCulate:PMD:POINts	×	<int>	<int>	*5
Data output	:CALCulate:PMD:DATA	×	<int>	<real1>,..., <real n>	*5

*4: The command is invalid. The <int> parameter is used for a query command.
(Example) :CALC:POIN? 1
<int> indicates: 1 - 4 main trace Y axis data
5 - 8 main trace X axis data
9 - 12 reference trace Y axis data
13 - 16 reference trace X axis data
The query reply of the trace data is output for all data by delimiting with commas (,) in ASCII <real> format.

*5: For <int>, refer to the following table:

<int>	Port	Output polarization angle	Input polarization angle	Data
1	1	0°	0°	MAG
2	1	0°	0°	X axis of MAG
3	1	0°	0°	GD
4	1	0°	0°	X axis of GD
5	1	0°	90°	MAG
6	1	0°	90°	X axis of MAG
7	1	0°	90°	GD
8	1	0°	90°	X axis of GD
9	1	90°	0°	MAG
10	1	90°	0°	X axis of MAG
11	1	90°	0°	GD
12	1	90°	0°	X axis of GD
13	1	90°	90°	MAG
14	1	90°	90°	X axis of MAG
15	1	90°	90°	GD
16	1	90°	90°	X axis of GD
17	2	0°	0°	MAG
18	2	0°	0°	X axis of MAG
19	2	0°	0°	GD

<int>	Port	Output polarization angle	Input polarization angle	Data
20	2	0°	0°	X axis of GD
21	2	0°	90°	MAG
22	2	0°	90°	X axis of MAG
23	2	0°	90°	GD
24	2	0°	90°	X axis of GD
25	2	90°	0°	MAG
26	2	90°	0°	X axis of MAG
27	2	90°	0°	GD
28	2	90°	0°	X axis of GD
29	2	90°	90°	MAG
30	2	90°	90°	X axis of MAG
31	2	90°	90°	GD
32	2	90°	90°	X axis of GD
33	1	45°	45°	MAG
34	1	45°	45°	X axis of MAG
35	1	45°	45°	GD
36	1	45°	45°	X axis of GD
37	1	45°	135°	MAG
38	1	45°	135°	X axis of MAG
39	1	45°	135°	GD
40	1	45°	135°	X axis of GD
41	1	135°	45°	MAG
42	1	135°	45°	X axis of MAG
43	1	135°	45°	GD
44	1	135°	45°	X axis of GD
45	1	135°	135°	MAG
46	1	135°	135°	X axis of MAG
47	1	135°	135°	GD
48	1	135°	135°	X axis of GD
49	2	45°	45°	MAG
50	2	45°	45°	X axis of MAG
51	2	45°	45°	GD
52	2	45°	45°	X axis of GD
53	2	45°	135°	MAG

<int>	Port	Output polarization angle	Input polarization angle	Data
54	2	45°	135°	X axis of MAG
55	2	45°	135°	GD
56	2	45°	135°	X axis of GD
57	2	135°	45°	MAG
58	2	135°	45°	X axis of MAG
59	2	135°	45°	GD
60	2	135°	45°	X axis of GD
61	2	135°	135°	MAG
62	2	135°	135°	X axis of MAG
63	2	135°	135°	GD
64	2	135°	135°	X axis of GD

4.5 CALibration Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Port1/2 PMD normalize	:CALibration:PORT<1|2>:PMD:NORMalize:ACQuire	---	×	
Port1/2 PMD normalize ON/OFF	:CALibration:PORT<1|2>:PMD:NORMalize[:STATe]	<bool>	{ON|OFF}	
Light Source Cal	:CALibration:LSOurce	---	×	
Light Source Cal Auto ON/OFF	:CALibration:LSOurce:AUTO[:STATe]	<bool>	{ON|OFF}	
Light Source Cal state acquisition	:CALibration:LSOurce:STATe	---	{WARMUP| UNCAL| NORMAL}	
Polarization Search TE-TM	:CALibration:POLarization:TETM	---	×	
Polarization Search Max-Min	:CALibration:POLarization:MAXMin	---	×	
Optical source UNIT Calibration	:CALibration:LSOurce	---	×	

4.6 CURSor Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Cursor function				
All cursor ON/OFF	:CURSor[:STATe]	<bool>	{ON|OFF}	
Cursor on trace specification	:CURSor:TRACe	{1|2|3|4}	←	
Cursor display mode	:CURSor:MODE	{1|2}	←	
Cursor display page	:CURSor:PAGE	{1|2|3|4|5|6|7|8}	←	
Cursor to center	:CURSor:X1:SET:CENTer	---	×	
Cursor display				
X1 cursor ON/OFF	:CURSor:X1[:STATe]	<bool>	{ON|OFF}	
X2 cursor ON/OFF	:CURSor:X2[:STATe]	<bool>	{ON|OFF}	
Y1 cursor ON/OFF	:CURSor:Y1[:STATe]	<bool>	{ON|OFF}	
Y2 cursor ON/OFF	:CURSor:Y2[:STATe]	<bool>	{ON|OFF}	
Cursor display				
X1 cursor ON/OFF	:CURSor:X1:MOVE	<realX>	←	
X2 cursor ON/OFF	:CURSor:X2:MOVE	<realX>	←	
Y1 cursor ON/OFF	:CURSor:Y1:MOVE	<realY>	←	
Y2 cursor ON/OFF	:CURSor:Y2:MOVE	<realY>	←	
Cursor value reading				
X1 cursor data reading	:CURSor:X1:DATA?	×	<real>,<real>, <real>,<real>	*1
X2 cursor data reading	:CURSor:X2:DATA?	×	<real>,<real>, <real>,<real>	*1

*1: The query reply is output in the following order: Trace1, Trace2, Trace3, and Trace4.

4.7 DISPlay Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Y axis				
Scale	:DISPlay:TRACe<1|2|3|4>:Y[:SCALe]:PDIVision	<realY>	<real>	
Reference value	:DISPlay:TRACe<1|2|3|4>:Y[:SCALe]:RLEVel	<realY>	<real>	
Reference position	:DISPlay:TRACe<1|2|3|4>:Y[:SCALe]:RLEVel:OFFSet	<int>[PCT]	←	0 to 100
Auto scale (TRACE)	:DISPlay:TRACe<1|2|3|4>:Y[:SCALe]:AUTO	---	×	
Auto scale (ALL)	:DISPlay:Y[:SCALe]:AUTO	---	×	
Trace control				
Active trace specification	:DISPlay:TRACe:ACTive	{1|2|3|4}	←	
Main trace display ON/OFF	:DISPlay:TRACe<1|2|3|4>[:MAIN][:STATe]	<bool>	{ON|OFF}	
Reference trace display ON/OFF	:DISPlay:TRACe<1|2|3|4>:REFerence[:STATe]	<bool>	{ON|OFF}	
Reference trace save execution	:DISPlay:SAVE:REFerence	---	×	
Window				
Window count	:DISPlay:WINDow:NUMBer	{SINGle|DUAL|QUAD}	{SING|DUAL|QUAD}	
Grid ON/OFF	:DISPlay:WINDow:GRID[:STATe]	<bool>	{ON|OFF}	
Title				
Title setting	:DISPlay:TITLe	<str>	←	

4.8 FILE Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Data save	:FILE:STORe	<str>	×	*1
Data load	:FILE:LOAD	<str>	×	*1
Setting data save	:FILE:STORe:SETUp	<str>	×	*1
Setting data load	:FILE:LOAD:SETUp	<str>	×	*1
Measured data output in text format	:FILE:STORe:TRACe	<str>	×	*1
Image file save	:FILE:STORe:IMAGe	<str>	×	*1
Data file delete	:FILE:DELete	<str>	×	*1

*1: <str> indicates a file name

4.9 HCOPy Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Print execution	:HCOPy[:IMMediate]	---	×	

4.10 INITiate Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Start measurement	:INITiate[:IMMediate]	---	×	
Repeat measurement	:INITiate:REPeat	---	×	

4.11 SENSe Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Measurement condition				
Sensitivity	:SENSe:SENSe	{HIGH| MIDDle| NORMal| FAST}	{HIGH|MIDD| NORM|FAST}	
Wavelength meter				
ON/OFF	:SENSe:WAVemeter[:STATe]	<bool>	{ON|OFF}	
Maker selection	:SENSe:WAVemeter:MODel	{WM_Q8331| WM_TQ8325| WM_Q8326| WM_86120C| WM_WA1650}	←	
Average				
ON/OFF	:SENSe:AVERage[:STATe]	<bool>	{ON|OFF}	
Count	:SENSe:AVERage:COUNt	<int>	←	1 to 256
Fiber length				
Measurement execution	:SENSe:FIBerlength	---	×	
INDEX setting	:SENSe:FIBerlength:INDex	<real>	←	0.1 to 3.0
Measured wavelength auto setting ON/OFF	:SENSe:FIBerlength:LAMBda:AUTO[:STATe]	<bool>	{ON|OFF}	
Measured wavelength setting	:SENSe:FIBerlength:LAMBda	<real>[m]	←	
Manual input	:SENSe:FIBerlength:MANual	<real>[m]	←	0.2 m to 10,000 km

4.12 SOURce Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
X axis				
Wavelength and frequency switching	:SOURce:STIMulus:MODE	{WAVelength|FREQuency}	{WAV|FREQ}	
Wavelength (frequency) setting				
Center	:SOURce:CENTer	<realX>	←	
Span	:SOURce:SPAN	<realX>	←	
Start	:SOURce:STARt	<realX>	←	
Stop	:SOURce:STOP	<realX>	←	
Measurement condition				
Sweep mode	:SOURce:SWEep:MODE	{CONTinuous|STEP}	{CONT|STEP}	
Measurement mode	:SOURce:MEASure:MODE	{CD|PMD|CDPower|PMDPower|POWer}	{CD|PMD|CDP|PMDP|POW}	
Number of measurement points	:SOURce:SWEep:POINts	<int>	←	
Measurement resolution	:SOURce:SWEep:RESolution	<realX>	←	Only wavelength
Modulation frequency	:SOURce:SWEep:MODuration	<realX>	←	Only frequency
Differential ON/OFF	:SOURce:DIFFerential[:STATe]	<bool>	{ON|OFF}	
Differential optical source selection	:SOURce:DIFFerential:SOURce	{INTernal|EXTernal}	{INT|EXT}	
STEP PMD polarization controller selection	:SOURce:POLarization:STEP:SOURce	{INTernal|EXTernal}	{INT|EXT}	
External blank wavelength	:SOURce:DIFFerential:BLANk	<realX>	←	Only wavelength
Polarization output status setting	:SOURce:POLarization:SOP	{DEFault|SOP1|SOP2|RANdom}	{DEF|SOP1|SOP2|RAN}	
Polarization output QWP setting	:SOURce:POLarization:SOP<1|2>:QWP	<real>	←	-270.0 to 270.0
Polarization output HWP setting	:SOURce:POLarization:SOP<1|2>:HWP	<real>	←	-270.0 to 270.0
Polarization output home return	:SOURce:POLarization:HOMe			
Return to the previous measurement setting	:SOURce:PREVious	---	×	

4.13 STATus Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Operation Status Enable register set and read	:STATus:OPERation[:EVENt]	---	<int>	0 to 65535
Operation Status Event register read	:STATus:OPERation:ENABle	<int>	←	0 to 65535
Status byte clear	:STATus:PRESet	---	×	

4.14 SYSTem Subsystem

Function	Command syntax	Parameter	Query reply	Remarks
Error read	:SYSTem:PRESet	---	×	
Equipment reset	:SYSTem:ERRor	---	<int>,<str>	*1

*1: <int> indicates error numbers, and <str> indicates error descriptions.

ALPHABETICAL INDEX

IMPORTANT INFORMATION FOR ADVANTEST SOFTWARE

PLEASE READ CAREFULLY: This is an important notice for the software defined herein. Computer programs including any additions, modifications and updates thereof, operation manuals, and related materials provided by Advantest (hereafter referred to as "SOFTWARE"), included in or used with hardware produced by Advantest (hereafter referred to as "PRODUCTS").

SOFTWARE License

All rights in and to the SOFTWARE (including, but not limited to, copyright) shall be and remain vested in Advantest. Advantest hereby grants you a license to use the SOFTWARE only on or with Advantest PRODUCTS.

Restrictions

(1) You may not use the SOFTWARE for any purpose other than for the use of the PRODUCTS.

(2) You may not copy, modify, or change, all or any part of, the SOFTWARE without permission from Advantest.

(3) You may not reverse engineer, de-compile, or disassemble, all or any part of, the SOFTWARE.

Liability

Advantest shall have no liability (1) for any PRODUCT failures, which may arise out of any misuse (misuse is deemed to be use of the SOFTWARE for purposes other than it's intended use) of the SOFTWARE. (2) For any dispute between you and any third party for any reason whatsoever including, but not limited to, infringement of intellectual property rights.

LIMITED WARRANTY

1. Unless otherwise specifically agreed by Seller and Purchaser in writing, Advantest will warrant to the Purchaser that during the Warranty Period this Product (other than consumables included in the Product) will be free from defects in material and workmanship and shall conform to the specifications set forth in this Operation Manual.

2. The warranty period for the Product (the "Warranty Period") will be a period of one year commencing on the delivery date of the Product.

3. If the Product is found to be defective during the Warranty Period, Advantest will, at its option and in its sole and absolute discretion, either (a) repair the defective Product or part or component thereof or (b) replace the defective Product or part or component thereof, in either case at Advantest's sole cost and expense.

4. This limited warranty will not apply to defects or damage to the Product or any part or component thereof resulting from any of the following:

   (a) any modifications, maintenance or repairs other than modifications, maintenance or repairs (i) performed by Advantest or (ii) specifically recommended or authorized by Advantest and performed in accordance with Advantest 's instructions;

   (b) any improper or inadequate handling, carriage or storage of the Product by the Purchaser or any third party (other than Advantest or its agents);

   (c) use of the Product under operating conditions or environments different than those specified in the Operation Manual or recommended by Advantest, including, without limitation, (i) instances where the Product has been subjected to physical stress or electrical voltage exceeding the permissible range and (ii) instances where the corrosion of electrical circuits or other deterioration was accelerated by exposure to corrosive gases or dusty environments;

   (d) use of the Product in connection with software, interfaces, products or parts other than software, interfaces, products or parts supplied or recommended by Advantest;

   (e) incorporation in the Product of any parts or components (i) provided by Purchaser or (ii) provided by a third party at the request or direction of Purchaser or due to specifications or designs supplied by Purchaser (including, without limitation, any degradation in performance of such parts or components);

   (f) Advantest's incorporation or use of any specifications or designs supplied by Purchaser;

   (g) the occurrence of an event of force majeure, including, without limitation, fire, explosion, geological change, storm, flood, earthquake, tidal wave, lightning or act of war; or

   (h) any negligent act or omission of the Purchaser or any third party other than Advantest.

5. **EXCEPT TO THE EXTENT EXPRESSLY PROVIDED HEREIN, ADVANTEST HEREBY EXPRESSLY DISCLAIMS, AND THE PURCHASER HEREBY WAIVES, ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, (A) ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND (B) ANY WARRANTY OR REPRESENTATION AS TO THE VALIDITY, SCOPE, EFFECTIVENESS OR USEFULNESS OF ANY TECHNOLOGY OR ANY INVENTION.**

6. **THE REMEDY SET FORTH HEREIN SHALL BE THE SOLE AND EXCLUSIVE REMEDY OF THE PURCHASER FOR BREACH OF WARRANTY WITH RESPECT TO THE PRODUCT.**

7. **ADVANTEST WILL NOT HAVE ANY LIABILITY TO THE PURCHASER FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS OF ANTICIPATED PROFITS OR REVENUES, IN ANY AND ALL CIRCUMSTANCES, EVEN IF ADVANTEST HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND WHETHER ARISING OUT OF BREACH OF CONTRACT, WARRANTY, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE. TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE.**

8. **OTHER THAN THE REMEDY FOR THE BREACH OF WARRANTY SET FORTH HEREIN, ADVANTEST SHALL NOT BE LIABLE FOR, AND HEREBY DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ANY LIABILITY FOR, DAMAGES FOR PRODUCT FAILURE OR DEFECT, WHETHER ARISING OUT OF BREACH OF CONTRACT, TORT (INCLUDING, WITHOUT LIMITATION, NEGLEGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE.**

CUSTOMER SERVICE DESCRIPTION

In order to maintain safe and trouble-free operation of the Product and to prevent the incurrence of unnecessary costs and expenses, Advantest recommends a regular preventive maintenance program under its maintenance agreement.

Advantest's maintenance agreement provides the Purchaser on-site and off-site maintenance, parts, maintenance machinery, regular inspections, and telephone support and will last a maximum of ten years from the date the delivery of the Product. For specific details of the services provided under the maintenance agreement, please contact the nearest Advantest office listed at the end of this Operation Manual or Advantest 's sales representatives.

Some of the components and parts of this Product have a limited operating life (such as, electrical and mechanical parts, fan motors, unit power supply, etc.). Accordingly, these components and parts will have to be replaced on a periodic basis. If the operating life of a component or part has expired and such component or part has not been replaced, there is a possibility that the Product will not perform properly. Additionally, if the operating life of a component or part has expired and continued use of such component or part damages the Product, the Product may not be repairable. Please contact the nearest Advantest office listed at the end of this Operation Manual or Advantest's sales representatives to determine the operating life of a specific component or part, as the operating life may vary depending on various factors such as operating condition and usage environment.

SALES & SUPPORT OFFICES

Advantest Korea Co., Ltd.
22BF, Kyobo KangNam Tower,
1303-22, Seocho-Dong, Seocho-Ku, Seoul #137-070, Korea
Phone: +82-2-532-7071
Fax: +82-2-532-7132

Advantest (Suzhou) Co., Ltd.
Shanghai Branch Office:
Bldg. 6D, NO.1188 Gumei Road, Shanghai, China 201102 P.R.C.
Phone: +86-21-6485-2725
Fax: +86-21-6485-2726

Shanghai Branch Office:
406/F, Ying Building, Quantum Plaza, No. 23 Zhi Chun Road,
Hai Dian District, Beijing,
China 100083
Phone: +86-10-8235-3377
Fax: +86-10-8235-6717

Advantest (Singapore) Pte. Ltd.
438A Alexandra Road, #08-03/06
Alexandra Technopark Singapore 119967
Phone: +65-6274-3100
Fax: +65-6274-4055

Advantest America, Inc.
3201 Scott Boulevard, Suite, Santa Clara, CA 95054, U.S.A
Phone: +1-408-988-7700
Fax: +1-408-987-0691

ROHDE & SCHWARZ Europe GmbH
Mühldorfstraße 15 D-81671 München, Germany
(P.O.B. 80 14 60 D-81614 München, Germany)
Phone: +49-89-4129-13711
Fax: +49-89-4129-13723

ADVANTEST®

http://www.advantest.co.jp

ADVANTEST CORPORATION
Shin-Marunouchi Center Building, 1-6-2 Marunouchi, Chiyoda-ku, Tokyo 100-0005, Japan
Phone: +81-3-3214-7500