MP1632C
Digital Data Analyzer
Remote Control
Operation Manual

Second Edition

To ensure that the equipment is used safety, read the "For Safety" in the MP1632C Digital Data Analyzer Operation Manual first.
Keep this manual with the equipment

Measurement Solutions
ANRITSU CORPORATION

Document No.:M-W1860AE-2.0

MP1632C Digital Data Analyzer
Remote Control
Operation Manual

1 April 2001 (First Edition)

15 July 2002 (Second Edition)

Printed in Japan

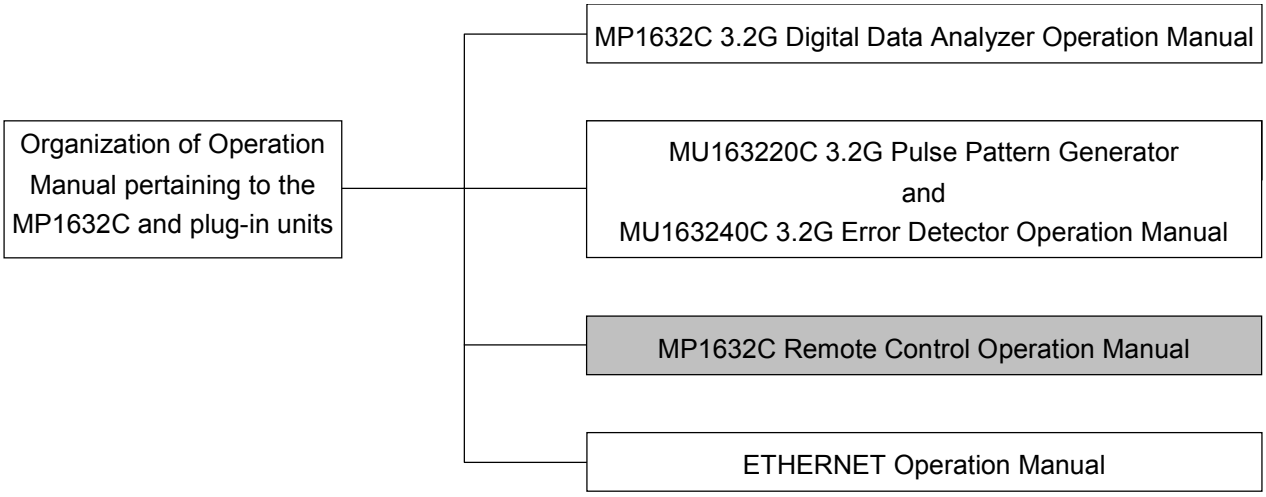
Trademark

Microsoft and Visual Basic are registered trademarks
of Microsoft Corporation in the U.S.A. and other countries.

INTRODUCTION

Organization of Operation Manuals

The MP1632C Digital Data Analyzer is a main unit of the measuring instrument in which a plug-in unit can be inserted. The following four operation manuals are prepared for the main unit and each connected unit. Select the appropriate one fit to your application.



- MP1632C Digital Data Analyzer Operation Manual
  This manual describes a brief overview, preparations before starting operation, the panel, specifications, performance and operation procedures of the MP1632C and Internal synthesizer option.
- MU163220C 3.2G Pulse Pattern Generator and MU163240C 3.2G Error Detector Operation Manual
  This manual describes a brief overview, specifications, performance, and operation procedures related to the units of the 3.2G Pulse Pattern Generator and the Error Detector.
- MP1632C Digital Data Analyzer Remote Control Operation Manual
  This manual describes the controls using the external interface (RS-232C, GPIB option, Ethernet option) and program examples.
- ETHERNET operation manual
  This manual describes how to connect the MP1632C connect to LAN, using FTP, sample program which controls MP1632C via ethernet.

CONTENTS

CHAPTER 1 GENERAL

1.1. Interface Functions

The MP1632C Digital Data Analyzer (hereinafter referred to as the instrument) installs an RS-232C interface as standard. It can also install a GPIB interface and/or Ethernet interface as option.

RS-232C, GPIB and Ethernet can be installed together at the same time.

The control commands of the measuring instrument conform to the SCPI (Standard Commands for Programmable Instruments) and the same command can be used regardless of the type of installed interface.

- RS-232C interface
  The instrument installs an RS-232C (COM1) as standard. Use this interface to control the Instrument in connection with the controller.
- GPIB interface (option 01)
  The instrument installs a GPIB interface that conforms to the IEEE standard 488.1.1987. Use this interface to control the Instrument in connection with the controller.
- Ethernet interface (option 02)
  The instrument installs an Ethernet interface that conforms to the IEEE802.3. Use this interface to control the Instrument in connection with the controller.

The instrument provides the following interface functions:

- Controls all the functions except those listed below:
  ・Power switch and Local key
  ・Interface settings (for example, RS-232C communication rate)
  ・Open/close of dialog box and screen/window operation (such as Pattern Editor cursor operation)
- Reading of all the setting parameters and data displayed in the screen
- Auto measuring system configuration in connection with a PC
- Interruption function and serial pole operation (when the GPIB interface is used)

Additionally, the instrument provides the Native command realizing compatibility with Anritsu's 3G BERTS (MP1652A/MP1653A). It can be selected which Native or SCIP.

1.2. Example of System Configuration

The following shows an example of system setup using the instrument interface.

- Host computer control
  The auto measuring system can be configured in connection with the host computer.

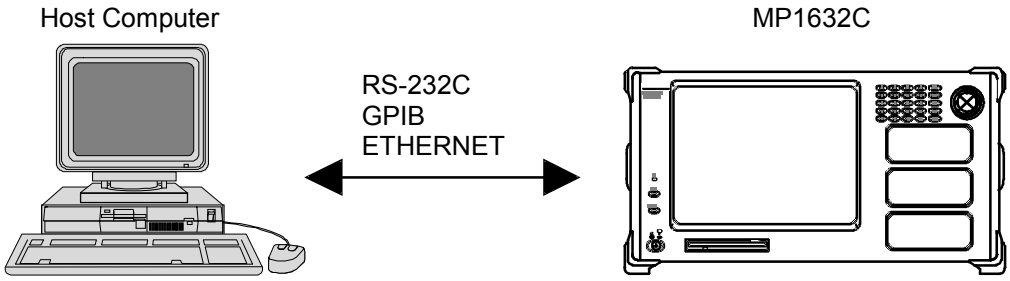


Fig.1-1 Host computer control

CHAPTER 2 CONNECTION AND SETTING OF INTERFACE

2.1. RS-232C Interface

2.1.1. Connecting RS-232C Cable

A connector to the RS-232C cable is mounted on the rear panel. Be sure to connect the RS-232C cable to the Instrument before turning ON the power switch.

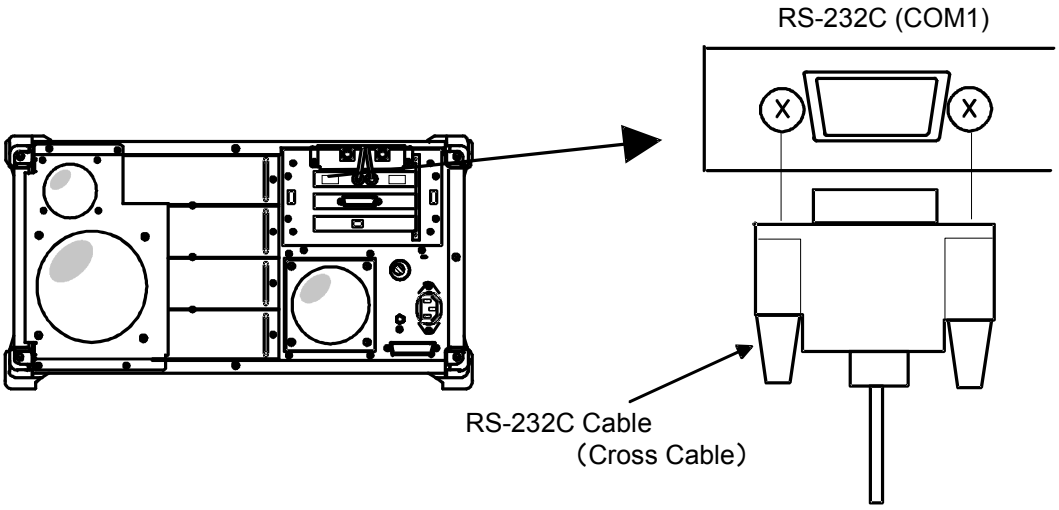


Fig. 2-1 Connecting the RS-232C Cable

2.1.2. Setting RS-232C

When using the RS-232C as the remote interface, set the instrument in the local mode and do the settings listed below on the Instrument screen. You can change the settings any time as long as the instrument is in the local mode.

Table 2-1 Setting of RS-232C

Setting description	Setting item	Setting range	Initial value
Interface for remote control	Remote	RS-232C(COM1) GPIB(When OPT01 is installed) Ethernet(When OPT02 is installed) None	RS-232C(COM1)
Baud rate	Speed	110, 300, 600, 1200, 2400,4800, 9600, 19200	9600
Parity	Parity	None, Even, Odd	None
Character length	Data Bit	8bits, 7bits	8bits
Stop bit length	Stop Bit	2bits, 1bit	1bits
Flow control	Flow Control	Xon/Xoff, Hardware, None	None

Do the above setting from the System:System panel. Follow the setting procedure described below:

1. Open the System:System panel.

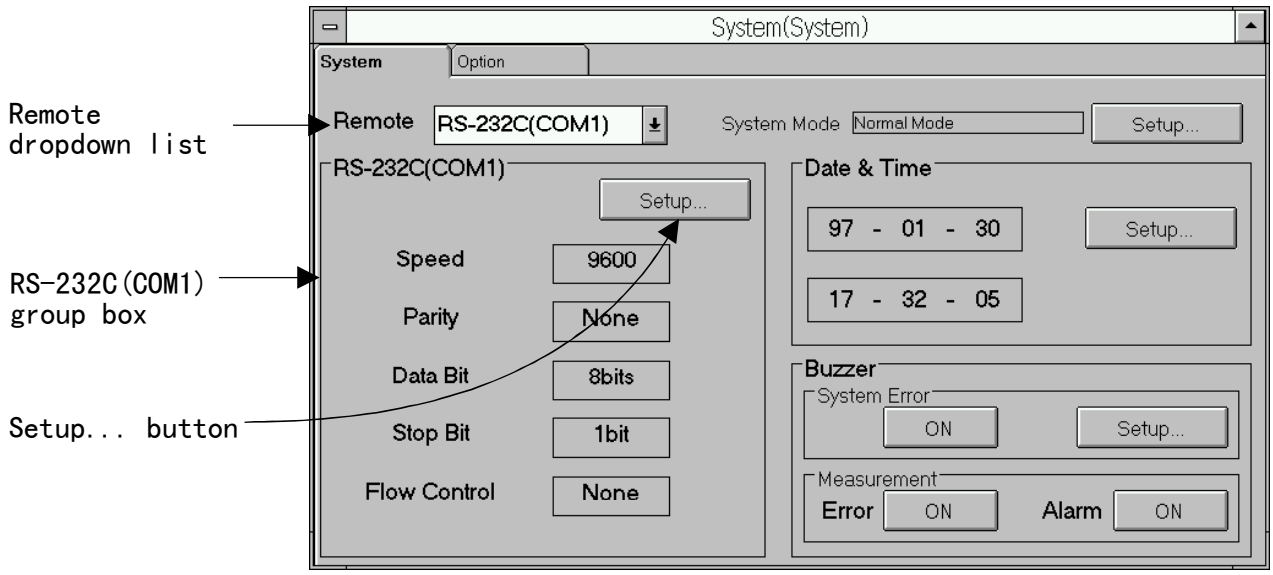


Fig. 2-2 System:System Panel

2. Select RS-232C (COM1) from the Remote drop down list.
   The RS-232C (COM1) group box appears in the System:System panel.

3. Click the Setup ... button within the RS-232C (COM1) group box. The RS-232C (COM1) dialog box opens.

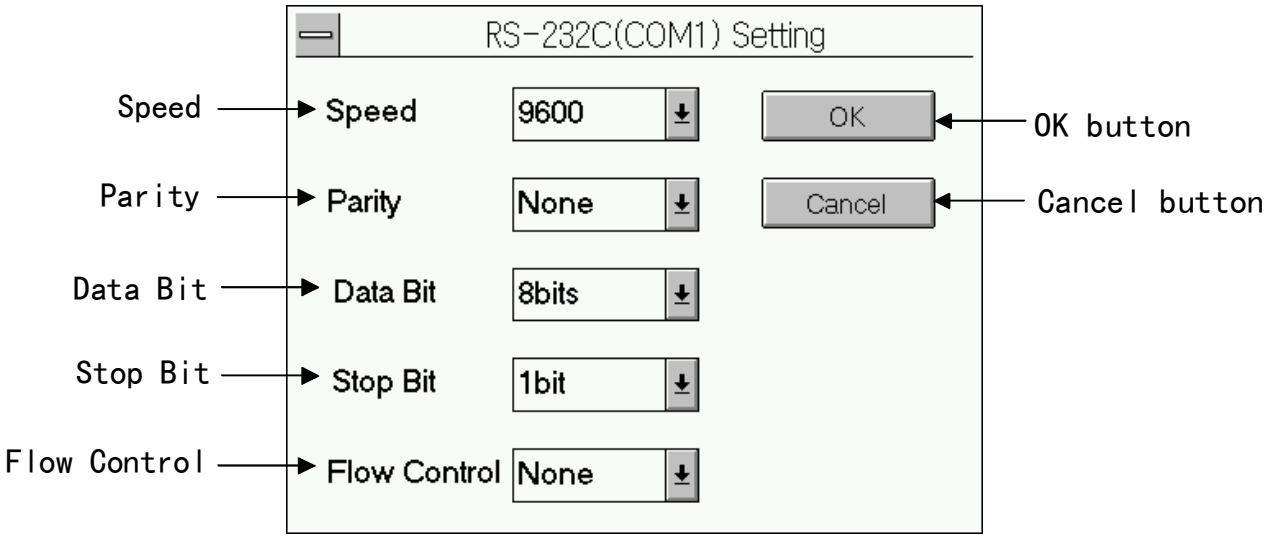


Fig. 2-3 RS-232C (COM1) Dialog Box

4. Set each parameter to fit your operating environments.
5. Now, the RS-232C settings are completed. To enable the settings, click the OK button. To disable the settings and return the parameters to the original values, click the Cancel button.

2.2. GPIB Interface

2.2.1. GPIB Interface Function

IEEE488.2 determines the minimum required subsets to be incorporated to each measuring instrument, from the GPIB interface functions regulated in IEEE4888.1, to ensure at least a system level for each measuring instrument.
The Instrument GPIB has the following subsets listed in the code column in the table below.

Table 2-2 GPIB Interface Functions

Code	Interface function	IEEE488.2 standard
SH1	Provides all the functions of source hand-shake. Takes timing in data transmission.	All the functions are provided as standard.
AH1	Provides all the functions of acceptor hand-shake. Takes timing in data reception.	All the functions are provided as standard.
T6	Provides the basic talker function. Provides the serial pole function. Does not provide the talk-only function. Provides the talker reset function by MLA.	The device should have any one of the subsets T5, T6, TE5 or TE6.
L4	Provides the basic listener function. Provides the listen-only function. Provides the listener reset function by MLA.	The device should have any one of subsets, L3, L4, LE3 or LE4.
SR1	Provides all the functions of service request and status byte.	All the functions are provided as standard.
RL1	Provides all the functions of remote/local. Provides the local lockout function.	RL0 (no function) or RL1 (all functions)
PP0	Does not provide the parallel pole function.	PP0 (no function) or PP1 (all functions)
DC1	Provides all the functions of device clear.	All the functions are provided as standard.
DT1	Provides all the functions of device trigger.	DT0 (no function) or DT1 (all functions).
C0	Does not provide the controller function.	Either C0 (no function) or C4 and C5, C7, C9 or C11

2.2.2. Bus Command

A bus command is an interface internal communication that is sent when the GPIB bus mode is set to the command mode (when the TN line is "L").

When using the GPIB as the remote control interface, you can use a bus command from the controller to make the following control available.

The bus command list is shown below.

Table 2-3 Bus Command

Bus command	Operation
DCL(Device clear)	Initializes message exchange of all the devices connected to the GPIB bus.
SDC(Selected Device Clear)	Initializes message exchange of the addressed devices. Functions in the same manner as DCL.
GET(Group Execute Trigger)	Functions in the same manner when the Start key is clicked.
IFC(Interface Clear)	Initializes the interface.

2.2.3. Connecting GPIB Cable

When the GPIB option (OPT01) is installed, a connector to the GPIB cable is mounted on the rear panel. (The GPIB option is factory-set.) Be sure to connect the GPIB cable before turning ON the power switch.

The number of devices connectable to a single system is up to 15 including the controller. The cable length has the following limitation:

- Total cable length ≦ 20 m
- Cable length between devices ≦ 4 m

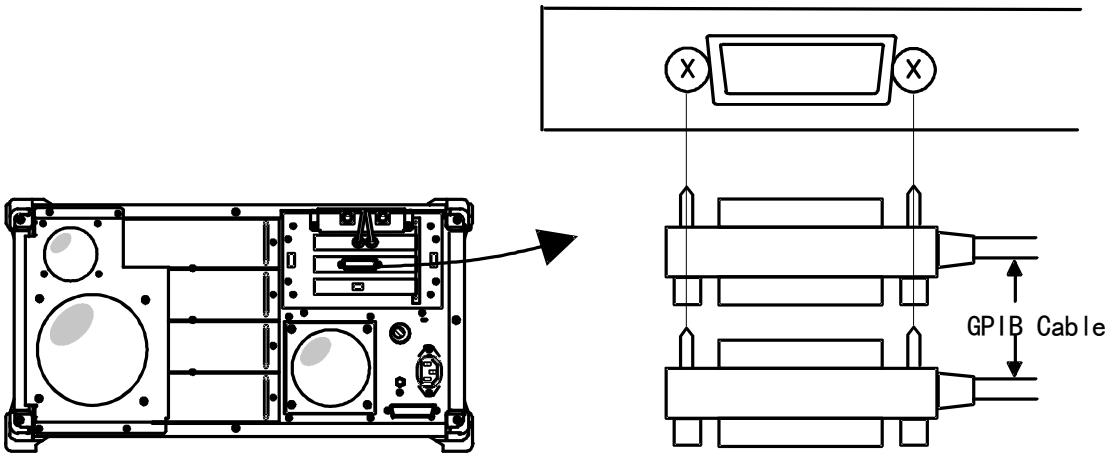


Fig. 2-4 Connecting the GPIB Cable

2.2.4. Setting GPIB

When using the GPIB as the remote interface, set the device in the local mode and do the settings listed below from the Instrument screen. You can change the settings at any time as long as the device is in the local mode.

Table 2-4 GPIB Setting

Setting description	Setting item	Setting range	Initial value
Interface for remote control	Remote	RS-232C(COM1) GPIB (when OPT01 is installed) Ethernet (when OPT02 is installed) None	RS-232C(COM1)
GPIB address	GPIB Address	1 to 30/Step:1	1

Do the above setting from the System:System panel. Follow the setting procedure described below:

1. Open the System:System panel.

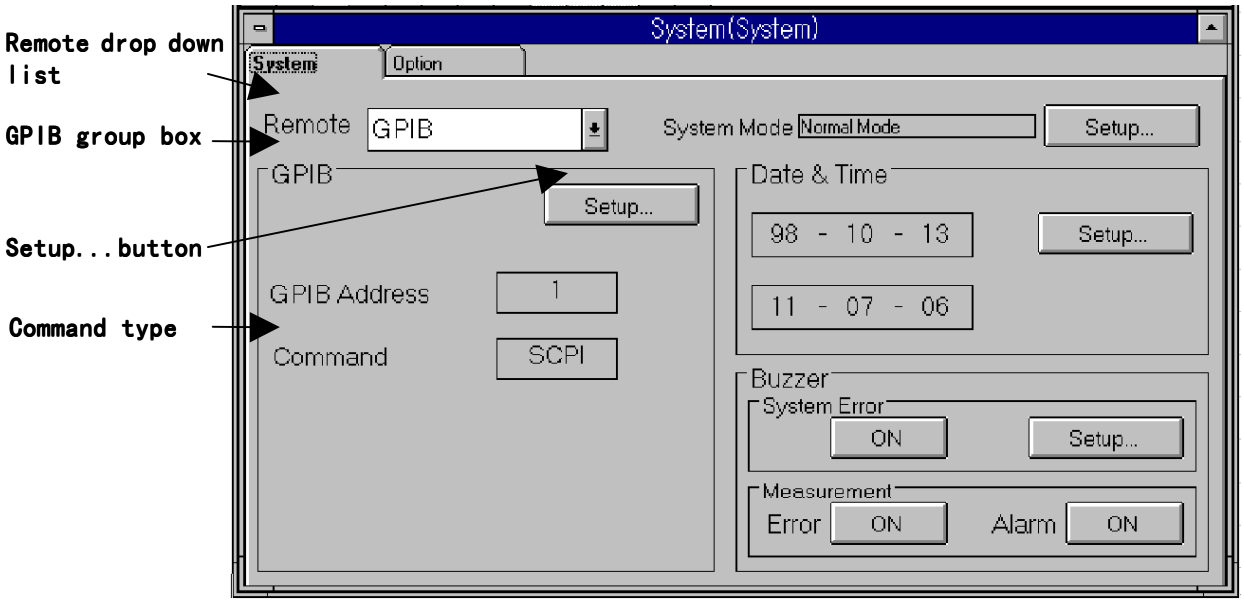


Fig. 2-5 System:System Panel

2. Select GPIB from the Remote drop down list.
   The GPIB group box appears in the System:System panel.

3. Click the Setup ... button within the GPIB group box. The GPIB dialog box opens.
When using SCPI commands, select SCIP from Command Type option switch. In this case, GPIB Address must be within 1 to 30.

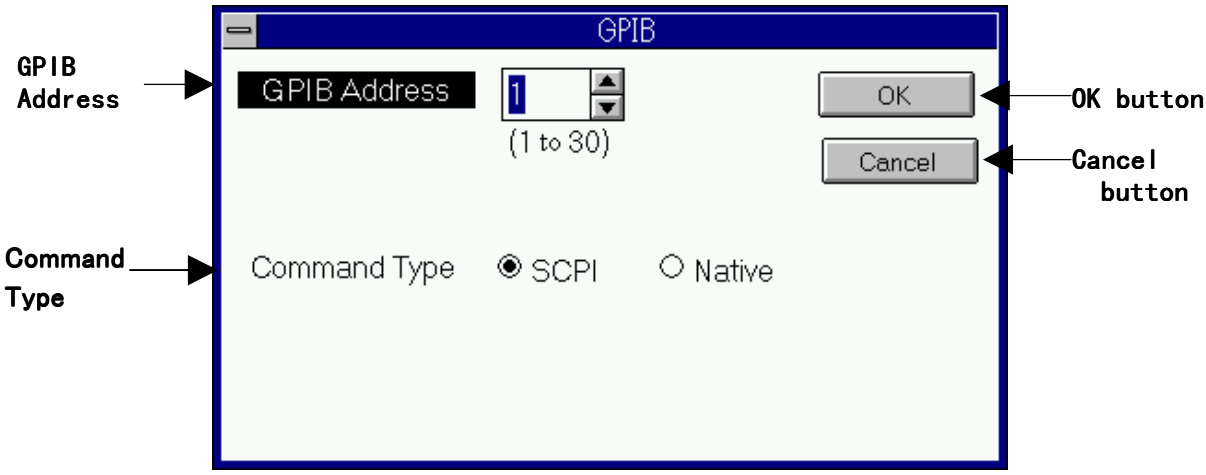


Fig. 2-6 GPIB Dialog Box (SCIP)

4. When using SCPI commands, select SCIP from Command Type option switch. In this case, GPIB Address (Address1) must be within 2 to 28.
The Address2 must be next value of Address1. It will be set to such value automatically.
And, select to which unit is assigned each addresses.

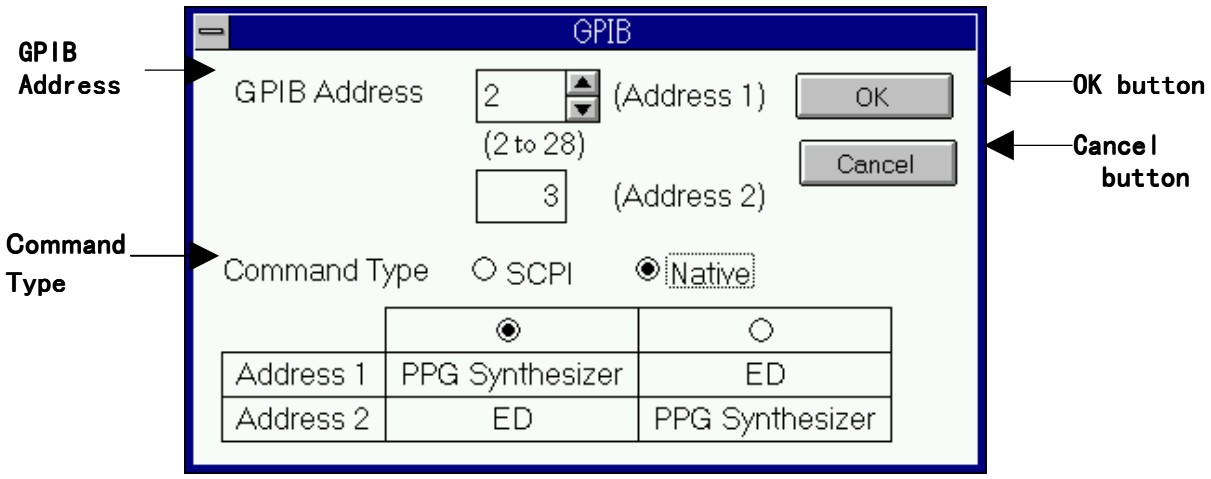


Fig. 2-7 GPIB Dialog Box (Native)

Now, the GPIB settings are completed. To enable the settings, click the OK button. To disable the settings and return the parameters to the original values, click the Cancel button.

Changing the command type will cause rebooting Windows. To enable setting above, press OK button on the attempt dialog box displayed after closing the GPIB dialog box.

Notes

The GPIB address in the Instrument is retained in the backup memory even when the power switch is turned OFF. After the power switch is turned ON, the Instrument application sets the own address.

During power-ON, you can change the GPIB address at any time unless the device is in the remote mode. When closing the address change dialog box by clicking the OK button, the address is changed.

2.3. Ethernet Interface

2.3.1. Ethernet Control Method

When using Ethernet as the remote interface, MP1632C operates as a server. Since it uses port No. 5001, designate this port No. for connection on the client.

Notes

When using Ethernet as the remote interface, you should link the Instrument to the TCP/IP network.
For further information on linking to the network, see the MP1632C Digital Data Analyzer Ethernet Option Operation Manual.

2.3.2. Connecting Ethernet Cable

When an Ethernet option (option 02) is installed, a connector to the Ethernet cable is mounted on the rear panel. (The Ethernet option is factory-set.)
Insert a RJ-45 jack in the correct direction until you hear a click sound.

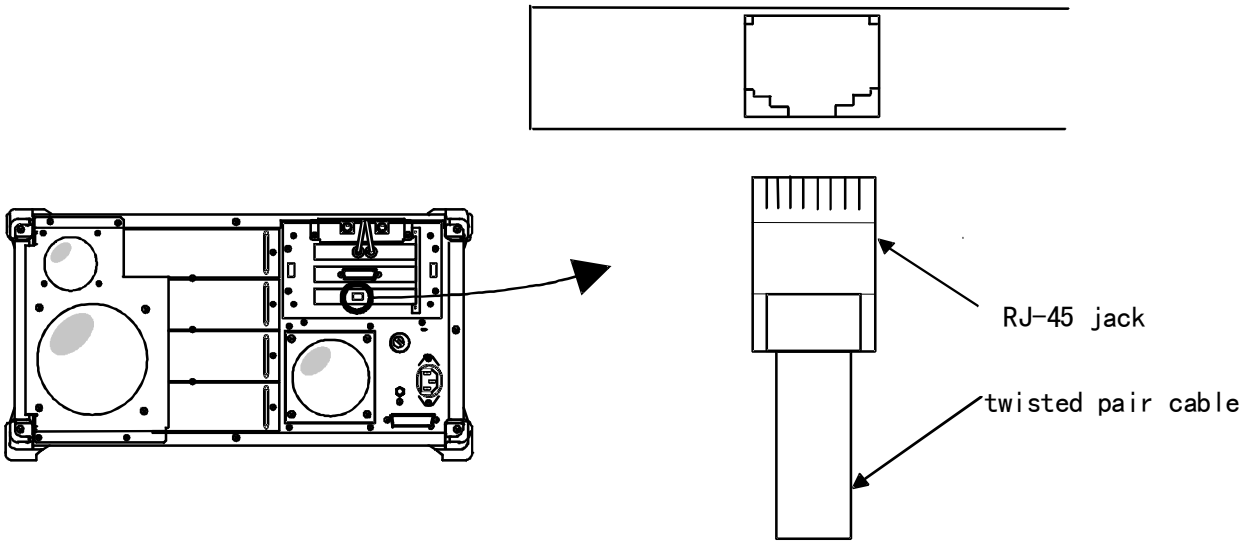


Fig. 2-8 Connecting the Ethernet Cable

2.3.3. Setting Ethernet

When using Ethernet as the remote interface, set the device in the local mode and do the settings listed below on the Instrument screen. You can change the settings any time as long as the device is in the local mode.

Table 2-5 Setting of Ethernet

Setting description	Setting item	Setting range	Initial value
Interface for remote control	Ethernet	RS-232C(COM1) GPIB (when OPT01 is mounted) Ethernet (when OPT02 is mounted) None	RS-232C(COM1)

Do the above setting from the System:System panel. Follow the setting procedure described below:

1. Open the System:System panel.

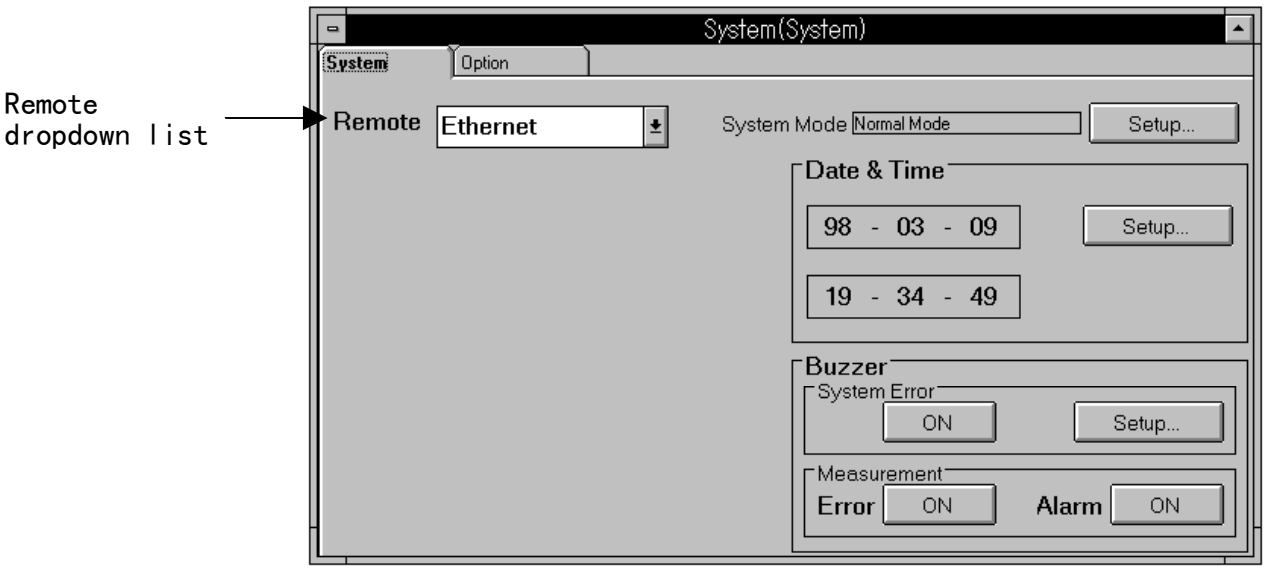


Fig. 2-9 System:System Panel

2. Select Ethernet from the Remote drop down list.

CHAPTER 3 INITIAL SETTING

3.1. Initializing Devices

The Instrument uses the following two commands to initialize a device. These commands initialize a device at the third level (initialization of device) stipulated in IEEE488.2.

- *RST common command
- :SYSTem:MEMory:INITialize command (SCPI)
- INI command(Native)

■ Initializing a device using the *RST common command

*RST is one of the IEEE488.2 common commands and is available regardless of type of the interface used.

By executing the *RST command, the device is set in the following state:

1. Returns the device-specific function to the preset state regardless of its usage history. On the Instrument, the device is returned to the factory-set state.
2. Sets the device in the OCIS (Operation Complete Command Idle State).
3. Sets the device in the OQIS (Operation Complete Query Idle State).

Notes

The *RST command does not influence the following elements:

1. State of IEEE488.2 interface
2. Device address
3. Output queue
4. Service request enable register
5. Standard event status enable register
6. Power-on-status-clear flag setting
7. Configuration data which may influence the device standard
8. Macro defined by the DMC command
9. Response message to PUD inquiry
10. Response message to RDT inquiry
    (The Instrument does not support 6 and 8 to10.)

■ Initializing a device using the :SYSTem:MEMory:INITialize command

The :SYSTem:MEMory:INITialize or INI command is one of the device-specific commands and is available regardless of the type of the interface used.

By executing this command, the device setting value is initialized to the factory-set value.

Notes

This command does not initialize the interface settings (such as RS-232C baud rate).

3.2. Initializing the GPIB System

When using GPIB as the interface for remote control, the GPIB system is initialized in the following levels in addition to at the third initialization level (level 3) described previously (initialization of device).

The following explains initialization of bus (level 1) and initialization of message (level 2).

Table 3-1 Initialization Levels of the GPIB System

Level	Initialization type	Description
1	Initialization of bus	Initializes all the interface functions connected to the bus using an IFC message from the controller.
2	Initialization of message	Disables a function to report the end of initialization/operation of message exchange of all the devices on GPIB using the GPIB bus command DCL, or disables a function to report the end of initialization/operation of message exchange of the designated device using the bus command SDC.
3	Initialization of device	Returns the device to the preset state specific to the device using *RST or :SYSTem:MEMory:INITialize, regardless of the usage history.

■ Initializing the bus using the IFC statement

You can use the IFC statement to initialize all the device functions connected to the GPIB bus line.

Initialization of interface functions is to reset the device interface function state set by the controller and return it to the initial state.

The following table lists the interface functions which can be initialized by the IFC message. The functions marked with "○" are fully initialized and those marked with "△" are partially initialized. This command does not influence the device operation state.

Table 3-2 Change of Interface Function State when Using the IFC Message

No	Function	Symbol	Initialization by IFC
1	Source handshake	SH	○
2	Acceptor handshake	AH	○
3	Talker or expanded talker	T or TE	○
4	Listener or expanded listener	L or LT	○
5	Service request	SR	△
6	Remote/local	RL	
7	Parallel pole	PP	
8	Device clear	DC	
9	Device trigger	DT	
10	Controller	C	○

The following table shows the change of device state when using the IFC message.

Table 3-3 Change of Device State When Using the IFC Message

Talker/listener	All the talkers and listeners are set to the idle state (TIDS, LIDS) within 100 μ .
Controller	The controller is set to the idle state CIDS (Control IDle State) within 100 μ if it is not active (SACS: System control ACtive State).
Service request device	The state in which the device is sending a SRQ message to the controller (the state in which the SRQ line is set to the LOW level by the device) cannot be reset. The state in which the controller sets all the devices connected to the system bus in the serial pole mode can be reset.
Device in remote mode	A device currently in the remote mode is not reset by the IFC message.

■ Initializing message exchange using the DCL or SDC bus command

You can use the DCL or SDC bus command to initialize message exchange of all the devices on GPIB or of the designated device.

The following shows the items to be initialized as message exchange.

Table 3-4 Items to Be Initialized as Message Exchange

Input buffer and output queue	Cleared.
Syntax analyzer, execution controller, response creator	Reset.
Device commands including *RST	All the commands which interfere execution of these commands are cleared.
Messages to parameter/program	To treat parameters, all the commands and queries in the areas in which execution is delayed are discarded.
Process of *OPC command	The device is set to OCIS (Operation Complete Command Idle State). As a result, the end-of-operation bit cannot be flagged in the standard event status register.
Process of OPC? query	The devices is set to OQIS (Operation Complete Query Idle State). As a result, the end-of-operation bit 1 cannot be flagged in the output queue. The MAV bit is cleared.
Device function	The portions involved in message exchange are set in the idle state. The device continues waiting for a message from the controller.

3.3. Device State at Power-ON

This section describes the Instrument state when the power switch is ON.

At power-ON, the Instrument acts as below:

1. The Instrument is set to the state it was in the last time it was turned OFF.
2. The input buffer and output queue are cleared.
3. The syntax analyzer, execution controller and response creator are reset.
4. The Instrument sets the device to OCIS.
5. The Instrument sets the device to OQIS.
6. When the PSC flag is "true", the event status enable register is fully cleared. Then, the event is reported.

(The Instrument does not support function 6.)

CHAPTER 4 DETAILS ON DEVICE MESSAGES

4.1. Device Message List

A device message is a data message that is transmitted/received between the controller and the device via the system interface. Program message and response message are available.

A program message is an ASCII data message that is transferred from the controller to the device. A response message is an ASCII data message that is transferred from the device to the controller.

Both messages have the following types:

Table 4-1 Device Message List

Program message	Response message
● Program command ・Device-specific command (see Chapter 6 and 7) ・IEEE488.2 common command (see 4.4) ● Program query (see Chapter 6 and 7)	● Status message (see Chapter 5) ● Response message (see Chapter 6 and 7)

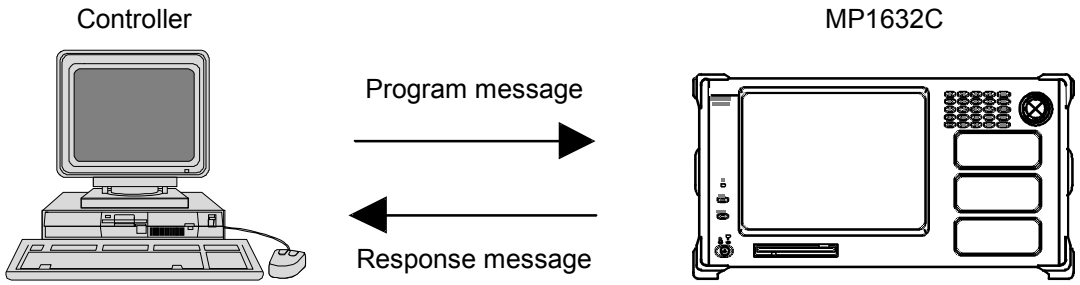


Fig. 4-1 Device Message List

The above-mentioned messages can be retried through the device I/O buffer. The I/O buffer is briefly described below:

Table 4-2 I/O Buffer

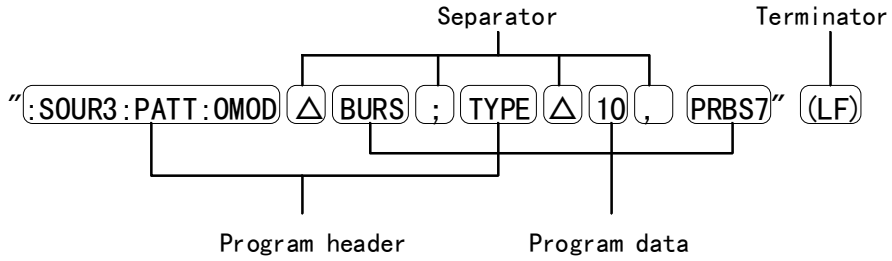
Input buffer	Output buffer
A FIFO (first-in first-out) type memory area in which DAB (program message and query message) can be stored temporarily before syntax analysis. The Instrument has a 16-kbyte input buffer.	A FIFO type queue memory area. All the DABs (response messages) to be sent from the device to the controller are stored in this memory until the controller finishes reading them. The Instrument has a 16-kbyte output buffer.

4.2. SCPI Listener Input Format

The Instrument commands conform both to SCPI and IEEE488.2. IEEE488.2 stipulates the program message as a sequence of functional elements.

A functional element is the minimum-level unit that can represent the function. These elements are roughly classified into four groups: separator, terminator, program header and program data.

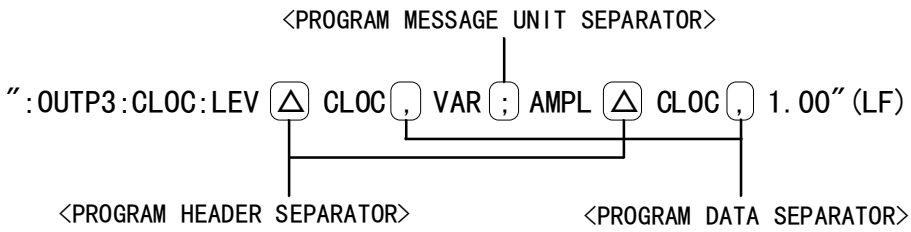
Example: When sending a program message ":OUTP3:CLOC:LEV CLOC,VAR;AMPL CLOC, 1.00" (LF) from the controller to the device
(Command set for clock output)



This section describes the format of the Instrument commands by functional element group. (△ in the text description stands for a space.)

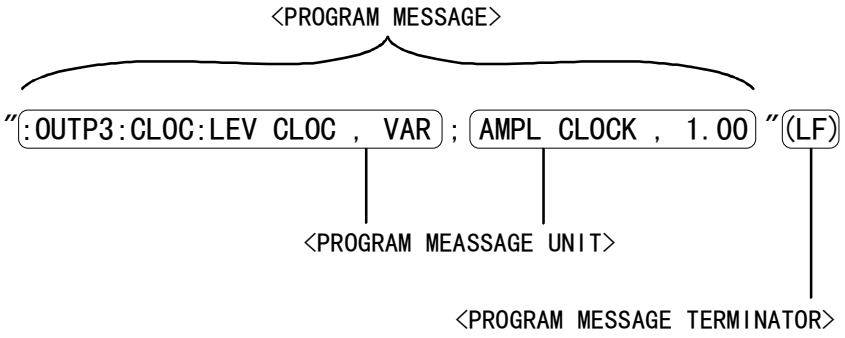
■ Separator

A separator contains the following three functional elements.



Functional element	Description
<PROGRAM MESSAGE UNIT SEPARATOR>	Separates multiple <PROGRAM MESSAGE UNIT> elements. Represented with zero or more spaces plus semicolon.
<PROGRAM DATA SEPARATOR>	Separates multiple <PROGRAM DATA> elements. Represented with zero or more spaces plus comma plus zero or more spaces.
<PROGRAM HEADER SEPARATOR>	Separates the program header from <PROGRAM DATA> elements. Represented with 1 or more spaces.

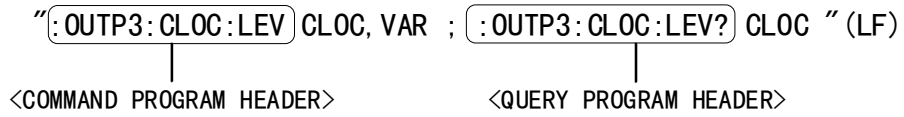
■ **Terminator**



Functional element	Description
<PROGRAM MESSAGE TERMINATOR>	Appended to the end of <PROGRAM MESSAGE> to terminate one or more sequences of <PROGRAM MESSAGE UNIT>. Represented with one or more spaces plus (CR+LF or LF) plus (EOI ON or EOI OFF).

■ Program header

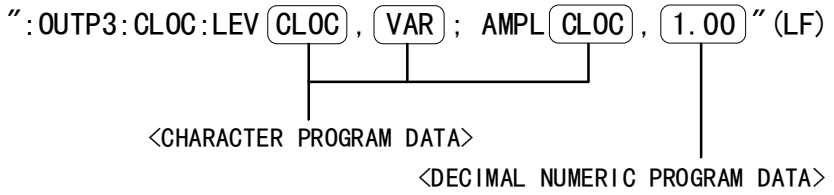
A program header contains the following two functional elements:



Functional element	Description
<COMMAND PROGRAM HEADER>	This is a command which allows the controller to control the device. As <COMMAND PROGRAM HEADER>, the Instrument uses <common command program header> and <compound program header> for the encoding syntax. <common command program header> The IEEE488.2 common commands use this syntax. Example: *RST <compound command program header> The device-specific command (SCPI) uses this syntax. Example: :OUT3:CLOC;LEV CLOC,VAR
<QUERY PROGRAM HEADER>	This is a query command to be sent to the device in advance because the controller receives a response message from the device. At the end of the header, query indicator ? is appended. As <QUERY PROGRAM HEADER>, the Instrument uses <common query program header> and <compound query program header> for the encoding syntax. <common query program header> The IEEE488.2 common query commands use this syntax. Example: *IDN? <compound query program header> The device-specific command (SCPI) uses this syntax. Example: OUT3:CLOC;LEV? CLOC

■ Program data

A program data of the Instrument contains six functional elements listed below.
In the command detail description, the program data type stipulated in IEEE488.2 corresponding to the device-specific parameter type is given.



Functional element	Description
<CHARACTER PROGRAM DATA>	A short string corresponding to the parameter function
<DECIMAL NUMERIC PROGRAM DATA>	Represents a decimal integer and a real number. The Instrument uses integers and fixed point numbers. Example: Integer 005, +5, 5, +5△ Fixed point number 12.345, .05, +0.05, 12.
<NON-DECIMAL NUMERIC PROGRAM DATA>	Represents a numeric value other than decimal integer. The Instrument uses hexadecimal values. They are displayed as #H+{A to F, a to F, 0 to 9}. Example: #HABC, #H123
<STRING PROGRAM DATA>	Represents a string enclosed with double-quotation or single quotation. Example: "ABCD", 1234'
<ARBITRARY BLOCK PROGRAM DATA>	This element is used to transfer a 8-bit binary data. #XYYYYY<binary> X : Digits of YYYY YYYYY : <binary> bytes <binary> : Binary data Example: #14<DAB><DAB><DAB><DAB> <DAB>: Data byte
<BOOLEAN PROGRAM DATA>	Program data stipulated in SCPI, representing a theoretical value. The theoretical value is represented with ON or 1 and OFF or 0.

Notes

SCPI uses a short form and a long form for the strings <CHARACTER PROGRAM DATA> and <STRING PROGRAM DATA>. The short form is an abbreviation of the long form. In the command statement, it is represented in uppercase within the string.

Either short form or long form can be compiled as the same function. Although uppercase is distinguished from lowercase in the description statement, there actually is no difference between uppercase and lowercase.

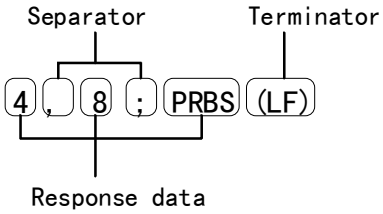
Example: OUTPut3:CLOCk:LEVel CLOCk,VAR (long form)
: OUTP3:CLOC:LEV CLOC,VAR (short form)

4.3. SCPI Talker Output Format

The instrument commands conform both to SCPI and IEEE488.2. IEEE488.2 stipulates the program message as a sequence of functional elements.

A functional element is the minimum-level unit that can represent the function. These elements are roughly classified into four groups: separator, terminator, program header and program data.

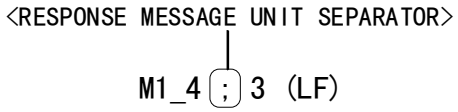
Example: When a response M1_4;3 is received after sending command ":SOUR3:PATT:PRBS:MRAT?;BSH?", which queries the PRBS pattern mark rate and bit shift



Because the Instrument response messages do not contain the response header, this section describes these messages of three groups. (△ in the text description stands for a space.)

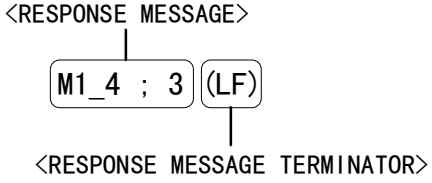
■ Separator

A separator contains the following two functional elements.



Functional element	Description
<RESPONSE MESSAGE UNIT SEPARATOR>	Separates multiple <RESPONSE MESSAGE UNIT> elements. Represented with semicolon.
<RESPONSE DATA SEPARATOR>	Separates multiple <RESPONSE DATA> elements. Represented with comma.

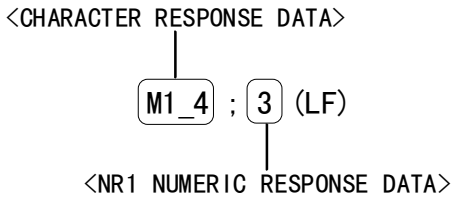
■ Terminator



Functional element	Description
<RESPONSE MESSAGE TERMINATOR>	Appended to the end of <RESPONSE MESSAGE> to complete message transfer. Represented with LF (LF + EOI for GPIB).

■ Response data

A response data of the Instrument contains 7 functional elements as below. In the command detail description, the response data type stipulated in IEEE488.2 corresponding to the device-specific response type is given.



Functional element	Description
<CHARACTER RESPONSE DATA>	A short string corresponding to the function
<NR1 NUMERIC RESPONSE DATA>	Represents a decimal integer.
<NR2 NUMERIC RESPONSE DATA>	Represents a decimal fixed point number.
<HEXADECIMAL NUMERIC RESPONSE DATA>	Represents a hexadecimal value. It is displayed as #H+{A to F, a to F, 0 7 Represents a string enclosed with double-quotation or single quotation.
<STRING RESPONSE DATA>	Represents a string enclosed with double-quotation.
<DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>	Represents a 8-bit binary data of known length. #XYYYYY<binary> X : Digits of YYYY YYYYY : <binary> bytes <binary> : Binary data Example: #14<DAB><DAB><DAB><DAB> <DAB>: Data byte
<ARBITRARY ASCII RESPONSE DATA>	Represents a 7-bit ASCII text data that is not separated.

Notes

SCPI uses a short form and a long form for the strings <CHARACTER PROGRAM DATA> and <STRING PROGRAM DATA>. The short form is an abbreviation of the long form. When a program command parameter is above type and the corresponding query command response is <CHARACTER RESPONSE DATA> or <STRING RESPONSE DATA>, a response to the query is a string in the short form.

4.4. Common Commands

The IEEE488.2 common commands supported by the Instrument are listed in the following table by functional group.

Table 4-3 IEEE488.2 Common Command List

Group	Function by group	Mnemonic
System data	Provides the information specific to devices connected to the GPIB system, for example, the device manufacturer, model and serial number.	*IDN?
Internal operation	Controls device inside: (1) Resets the device at level 3. (2) Performs the self-test for the device inside and detects errors.	*RST *TST?
Synchronization	Synchronize the device with the controller by means of (1) waiting for service request or (2) waiting for response from the device output queue. All the Instrument commands are sequential commands.	*OPC *OPC?
Status & Event	A status byte is composed of status summary messages. Individual summary bits in the message are supplied from the standard event register, output queue and expanded event register. To set, clear, enable or disable the data in the register or queue or to know the register setting condition through query, three commands and four queries are prepared.	*CLS *ESE (SES)† *ESE? (ESE?)† *ESR? (ESR?)† *SRE (SRQ)† *SRE? (SRQ?)† *STB? (STB?)†
Device trigger	Defines the command sequence to be executed when the device receives the GET bus command stipulated in IEEE488.2.	*TRG
Option check	Inquire the option function installed in the device.	*OPT?

†) Native GPIB command only

*IDN?

Response <ARBITRARY ASCII RESPONSE DATA>
<Product manufacturer>,<model>,<serial number>,<firmware revision number>
The Instrument is reported as:
ANRITSU,MP1632C,0,1.0

Function Reports the product manufacturer and model. (For the Instrument, the main frame model is returned as response.)

Application example *IDN?

*RST

Parameter None

Function Resets the setting to the factory-set value.

Application example *RST

***TST?**

Response <NR1 NUMERIC RESPONSE DATA>
0: No error
1: Error occurs.

Function Queries the result of self-diagnosis (RAM/LED/buzzer check) performed at power-ON.

Application example *TST?

***OPC**

Parameter None

Function Flags bit 0 (completion-of-operation bit) in the standard event status register when the previous command has been executed and sets SRQ ON.

Application example *OPC

***OPC?**

Response <NR1 NUMERIC RESPONSE DATA>
1

Function Returns 1 when the previous command has been executed.

Application example OPC?

***CLS**

Parameter None

Function Clears all the event registers and queues except the output queue and its MAV summary messages. To reset the enable register and the transition filter in the device-specific status register, use the SCPI command :STATus:PRESet.
When sending this command right after the element <PROGRAM MESSAGE TERMINATOR> before <Query MESSAGE UNIT>, both output queue and MAV bit are cleared.

Application example *CLS

***ESE (ESE is also available with Native command)**

Parameter <DECIMAL NUMERIC PROGRAM DATA>

Integer of 0 to 255

The total number of bit digits when selecting bits to be enabled from the standard event status enable register bits is used as the parameter. The number of digits of bits to be disabled becomes 0.

The Instrument has the following settings:

Bit0(2^0= 1)	End of operation
Bit2(2^2= 4)	Query error
Bit3(2^3= 8)	Command error, query error, errors other than execution error
Bit4(2^4= 16)	Execution error
Bit5(2^5= 32)	Command error
Bit6(2^6= 64)	Local control execution (SCPI only)
Bit7(2^7=128)	Power-ON (SCPI only)

Function Sets and clears the standard event status enable register.

Application example When setting bit 2 and bit 4 in the enable register:
*ESE 20

***ESE? (ESE? is also available with Native command)**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

Function Queries the current value of the standard event status enable register.

Application example When sending *ESE? after execution of *ESE 20, "20" is returned.

***ESR? (ESR? is also available with Native command)**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

The total number of bit digits in the standard event status register is used as response.

For the standard event status register bit setting, see the parameter of the *ESE command.

Function Queries the current value of the standard event status register.

Application example When a command error occurs, "32" is returned in response to query of *ESR?

***SRE (SRQ is also available with Native command)**

Parameter <DECIMAL NUMERIC PROGRAM DATA>

0 to 255

The total number of bit digits when selecting bits to be enabled from the service request enable register bits is used as parameter. The total number of digits of bits to be disabled becomes 0.

The Instrument has the following settings:

Bit1(2^2= 2)	Expanded event status register summary (Native only)
Bit2(2^2= 4)	Indicates that the error and event queues are not empty. (SCPI only)
Bit3(2^3= 8)	Questionable status register summary (SCPI only)
Bit4(2^4= 16)	Indicates that the output queue is not empty.
Bit5(2^5= 32)	Event status register summary (SCPI) Standard event status register summary (Native)
Bit7(2^7=128)	Operation status register summary (SCPI only)

Function Sets a bit in the service request enable register.

Application example When setting bit 4 in the enable register:

*SRE 16

***SRE? (SRQ? is also available with Native command)**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

Function Queries the current value of the service request enable register.

Application example When sending *SRE? after execution of *SRE16, "16" is returned.

***STB? (STB? is also available with Native command)**

Response <NR1 NUMERIC RESPONSE DATA>

0 to 255

The total number of bit digits in the status byte register is used as response.

The Instrument has the following settings:

Bit2(2^2=4)	Indicates that the error and event queues are not empty.
Bit3(2^3=8)	Questionable status register summary
Bit4(2^4=16)	Indicates that the output queue is not empty.
Bit5(2^5=32)	Event status register summary
Bit6(2^6=64)	MSS (Master Summary Status) summary message
Bit7(2^7=128)	Operation status register summary

Function Returns the current value of status byte including MSS (Master Summary Status) bit.

Application example When the event status register summary is "true", "32" is returned in

response to query of *STB?.

***TRG**

Parameter None

Function Acts in the same manner as GET (Group Execution Trigger bus command) of IEEE488.1.

Application example *TRG

***OPT?**

Response <ARBITRARY ASCII RESPONSE DATA>

Character corresponding to option

0	No option
OPT01	GPIB option
OPT02	Ethernet option
OPT03	3.2G Internal synthesizer option

Function Reports the option list installed in the MP1632C mainframe. All the options installed are separated each other with a comma in the report.

Application example When the GPIB interface option is installed and sending *OPT?, the following response is returned:

OPT01

4.5. Details On SCPI

The Instrument uses SCPI (Standard Commands for Programmable Instruments) as a remote control command.

This section describes details on SCPI and command structure.

General

SCPI is an hardware-independent device command language defined by the SCPI consortium. The purpose of SCPI is to shorten the auto measuring instruments (ATE) program development term. To do this, consistency is given to the programming environments such as device control and data handling.
The same models have compatibility and even different models having the same function have compatibility. To control different devices having the same function, you can use an equal control to the devices, which can be a unique characteristic of SCPI.

Command structure

```
:SOURce
    :PATTern
        :TYPE          <type>
        :TYPE?
```

The SCPI commands are built based on a hierarchical structure. They are grouped by the relevant function and each group constitutes hierarchy structure called "subsystem".

This manual represents these subsystems as the command tree as shown above. The same header may appear in the tree. Each header corresponds to different functions depending on the position in which the header exists. Thus, you should describe the command using the full path format containing the header to be used.

Example:
SOURce subsystem
SOURce is the highest level node.
PATTern is the second highest level node.
TYPE,TYPE? is the third highest level node.
<type> is a TYPE parameter type.

Command description

:SOURce:PATTern:TYPE <type> :SOURce:PATTern:TYPE?	<type> = PRBS7,PRBS9,PRBS11,...

The command tree introduced in the previous section is expressed as the above commands.

The command description rules are detailed below:

- Command format
  A command stars with ":". A command is composed by connecting header to header with ":".

- Abbreviation form of header
  A header has a short form and a long form. The short form is an abbreviation of the long form and corresponds to the uppercase in the header name.
  A command in the short form and the one in the long form can be compiled as the same command. (You may mix the two forms.)
  When describing the command, uppercase and lowercase are used to distinguish the short form from the long form. However, the uppercase and lowercase are not distinguished in actual applications.
  (SOURCE, Source and source are complied as the same header.)

  Example:

Long form	:SOURCE:PATTern:TYPE PRBS11
Short form	:SOUR:PATT:TYPE PRBS11
Long + short	:SOUR:PATTern:TYPE PRBS11

- Option node
  [ ] represents an option node. A header enclosed with [ ] can be omitted. A command in which the header description is omitted and one having a header description are compiled as the same command.

  Example:

When a header is not omitted	:DISPlay:SYSTem:NAME SYSTem
When a header is omitted	:DISPlay:SYSTem SYSTem

- Command separator
  One or more spaces are required between the command and the parameter. Two or more parameters must be separated with ",".

Combining commands

Commands can be combined with ";". The second command is referenced as the same level as the lowest layer of the first command.
This allows you to describe the second command in the full path format as shown in Example 1 or to omit the headers higher than TYPE as shown in Example 2.

Examples:

1) :SOURce:PATTern:OMODe REPeat;:SOURce:PATTern:OMODe?
2) :SOURce:PATTern:OMODe REPeat;OMODe?

4.6. Native Message Format

4.6.1. Command Format

The instrument Native command consists of a header field that indicates the command type, and a numeric data field that indicates spaces and values.
Use at least one space after the header field to separate it from the numeric data field.
In the data request message format, the header field, which shows the command type, is followed by [?] .

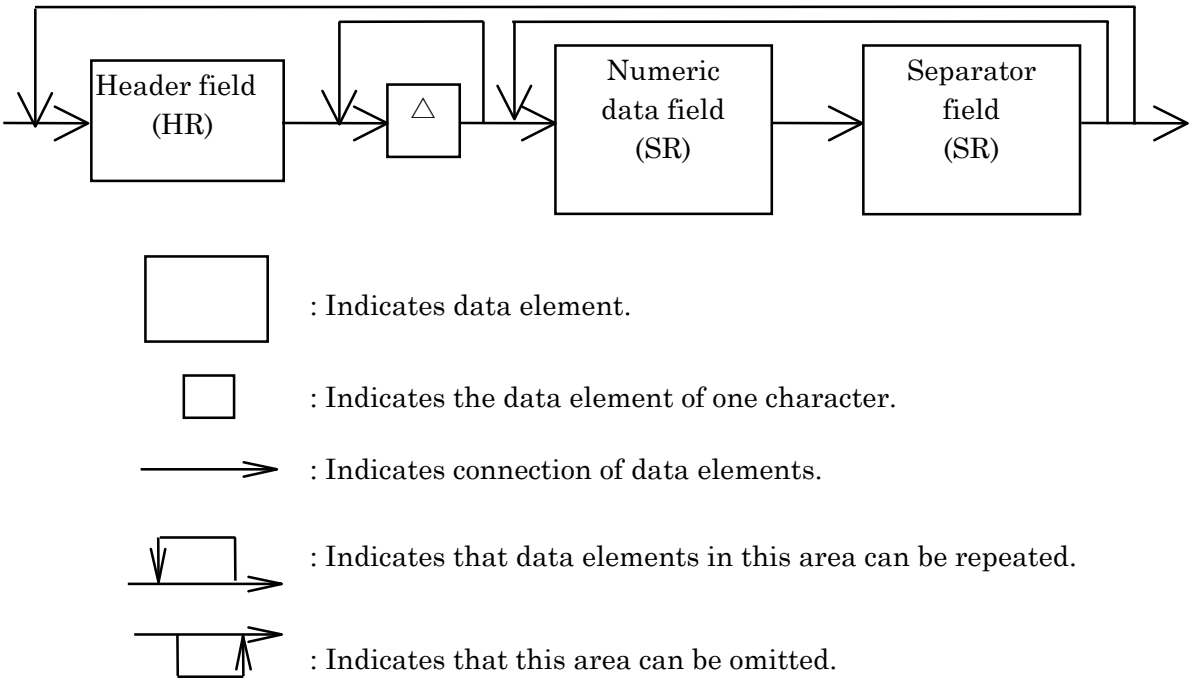
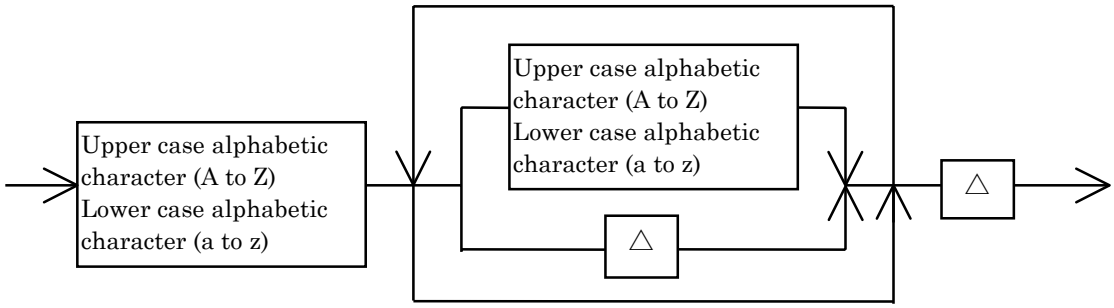


Fig. 4-2 Basic Command Format

4.6.2. Header Field (HR)

The header field has an HR2 format consisting of three alphabetic characters (A to Z, a to z).



4.6.3. Numeric Data Field (NR)

The numeric data field has five formats depending on the command: NR1 format (integer type), NR2 format (real number type), NR3 format (floating type), Hexadecimal format (HEX type), Binary format (BIN type).

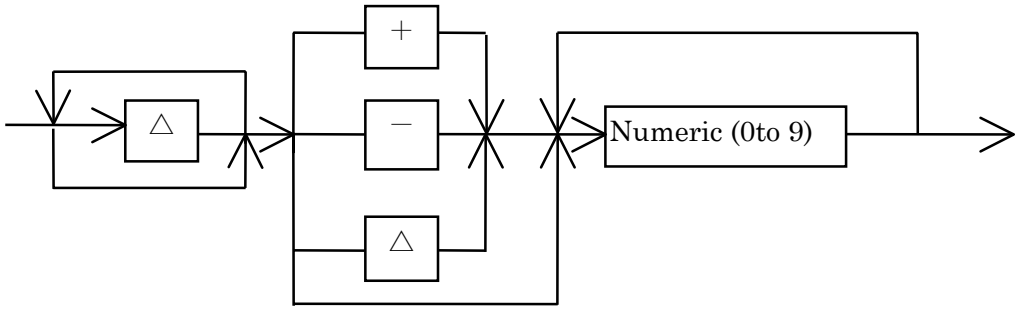
4.6.3.1. NR1 format (integer type)

NR1 format is an integer type numeric representation format. It consists of a one character sign (+, -) followed by a string of integers (0 to 9). The + sign can be omitted, or replaced by a space. Leading spaces can also be inserted to align the number of digits of data, etc.

Examples:

- General representation +1234 , - 567 , +0
- + sign omitted +1234 → 1234
- + sign replaced +1234 → Δ1234
- Spaces inserted +1234 → ΔΔ+12 3 4
  -567 → ΔΔΔ-567

(Δ : Space)



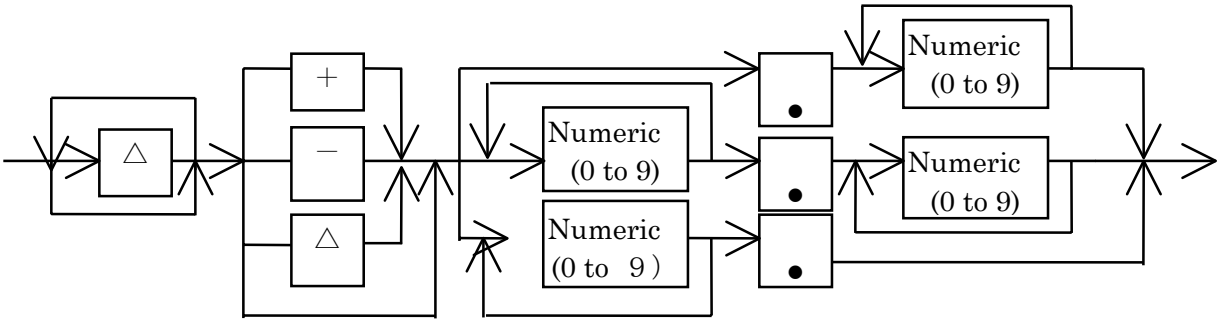
4.6.3.2. NR2 format (real number type)

NR2 format is real number type numeric representation. It consists of a character sign (+, -) followed by a string of integers (0 to 9), decimal point (.), and another string of integers (0 to 9). The + sign can be omitted or replaced and leading spaces can be inserted to align data, as for the NR1 format. When the decimal point is preceded, or followed by 0, the 0 can be omitted.

Examples:

- General representation +1.23, -45.6, -0.12, +34.0, +0.0
- + sign omitted +1.23 → 1.23
- + sign replaced +1.23 → Δ1.23
- Space insertion +1.23 → ΔΔ+1.2 3
  -45.6 → Δ-45.6
- High-order 0 omitted -0.12 → -.12
- Low-order 0 omitted +34.0 → +34
- +0.0 omitted +0.0 → +0. or +.0

(Δ : Space)



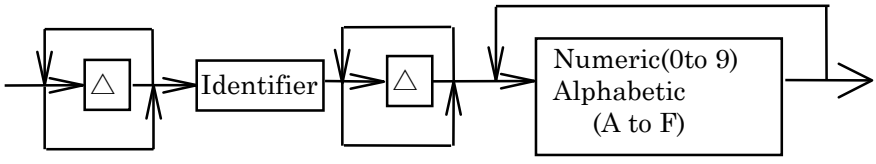
4.6.3.3. Hexadecimal format (HEX type)

Hexadecimal format consists of the identifier #H followed by a string of integers (0 to 9) or alphabetic characters (A to F). In hexadecimal format, a space can be inserted between the identifier and data. High-order 0 can be omitted.

Examples:

- General representation #H1234,#H00FF,#H0000
- Space insertion #H1234 → #HΔ1234
  #H00AF → #HΔΔΔ00AF
- 0 omission #H00FF → #HFF
  #H0000 → #H0

(Δ : Space)



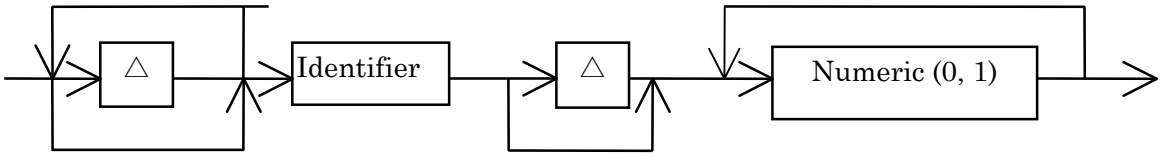
4.6.3.4. Binary format (BIN type)

Binary format consists of the identifier #B followed by the numerics (0 and 1). In binary format, a space can be inserted between the identifier and data.
When the most significant bit is 0, the zeroes following it can be omitted.

Examples:

- General representation #B11011011,#B00100100, #B00000000
- Space insertion #B11011011 → #B△11011011
  #B00100100 → #B△△△00100100
- 0 omission #B00100100 → #B100100
  #B00000000 → #B0

(△ : Space)

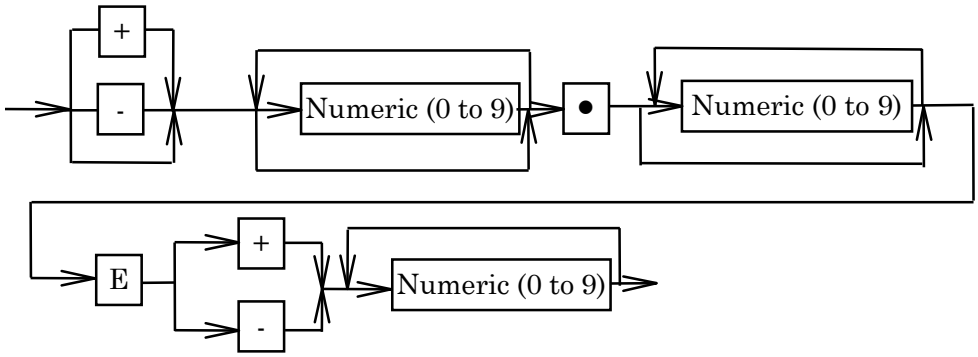


4.6.3.5. NR3 Format (Floating-point number type)

NR3 format is used to represent floating-point numbers.
It consists of a character sign (+,-) followed by a string of integers (0 to 9), a decimal point(.), a string of integers(0 to 9), a string representing exponent(E), a character sign (+,-), and another string of integers (0 to 9).
The first + sign can be omitted or replaced and leading spaces can be inserted to align data. But the + sign in exponent cannot. Any spaces cannot inserted after and before "E", and "E" cannot be replaced to its small letter "e".

Examples:

- General representation +1.23E+3, -2.455E-10
- + sign omitted +1.23E+3 → 1.23E+3



4.6.4. Command Separator

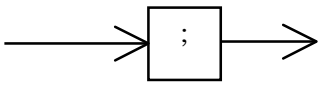
The command separator has the following four formats:
SR1 format (semicolon ;) SR1 format (comma ,) SR2 format (line feed LF), SR3 format (END message).

4.6.4.1. SR1 format (semicolon ;)

SR1 format (semicolon ;) is used to separate the commands from each other when commands are sent consecutively.

Example:

"ABC123;DEF45.6" is decoded as two commands "ABC123" and "DEF45.6".

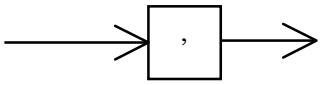


4.6.4.2. SR1 format (comma ,)

The SR1 format (comma ,) is used to separate data when data are sent after a header filed.

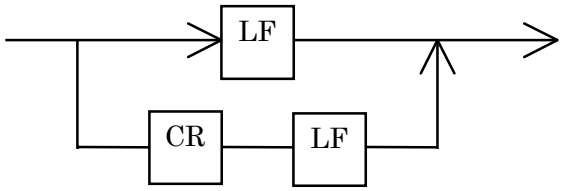
Example:

ABC123,45.6 is decoded as the two data "123" and "45.6" following the header "ABC".



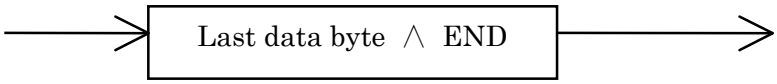
4.6.4.3. SR2 format (line feed LF)

The SR2 format is used as the symbol that indicates the end of the
command. The line feed (LF) code can be used as the SR2 format code. The carriage return (CR)code can also be inserted before the line feed code.
When the instrument is designed as a listener, the receiving operation is performed until this SR2 format code or the SR3 format code described next is received. When the SR2 format or SR3 format code is received, the receiving operation ends and the make up of the received command is decoded.



4.6.4.4. SR3 format (END message)

The SR3 format is used to indicate the end of a command sequence, as does the SR2 format.
An END message (EOI line used) can be used as the SR3 format code. (SR3 format and SR2 format can be used together.)



4.6.4.5. Definition of Space

The instrument treats the ASCII codes 00H to 09H, and 0BH to 20H as blank space.

CHAPTER 5 STATUS REPORT

5.1. MP1632C Status Register Configuration

The Instrument status register configuration conforms to the SCPI regulations. SCPI stipulates that the status register configuration should have the configuration regulated by IEEE488.2 and the SCPI-specific OPERation status register and QUEStionable status register.

The following figure is a sketch of the status register configuration installed in the Instrument.

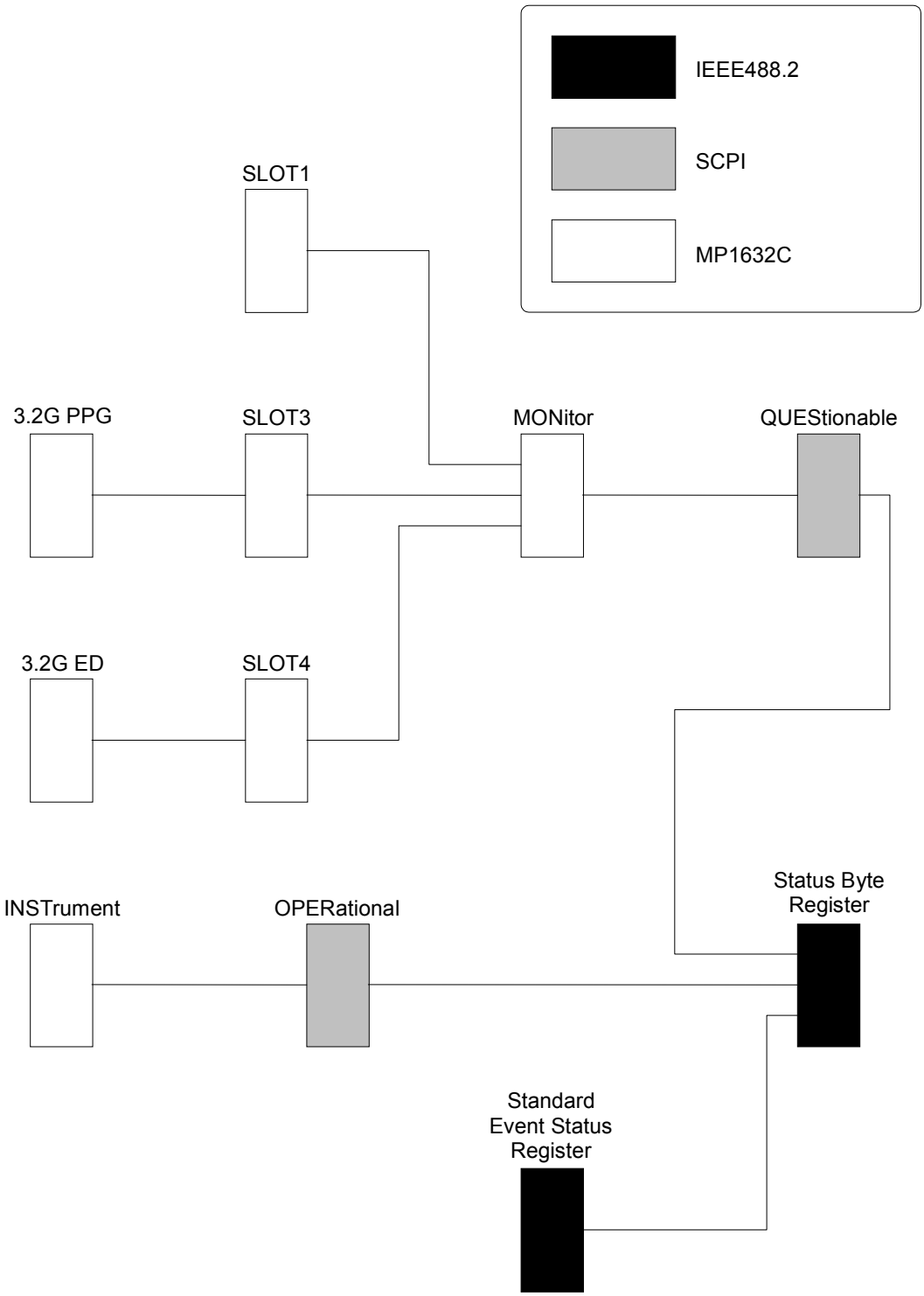


Fig. 5-1 MP1632C Status Register Configuration

The status register stipulated by SCPI, and the device-specific status register have the following configuration.

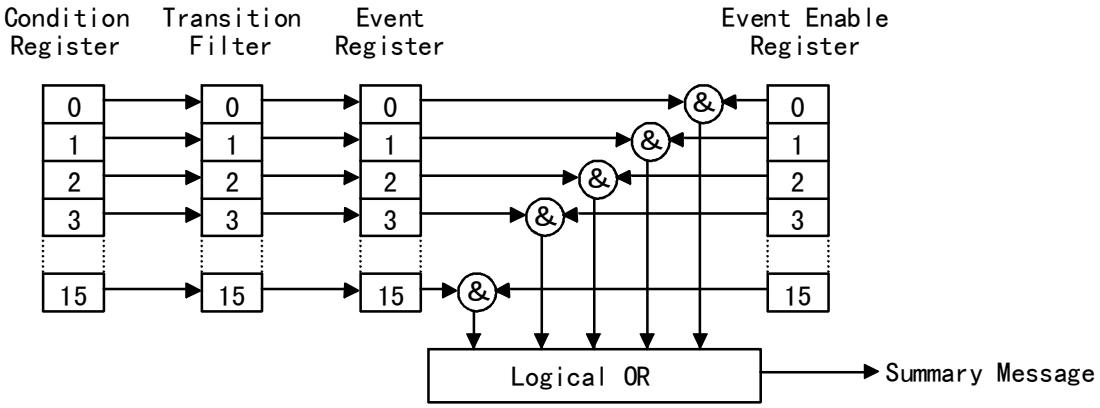


Fig. 5-2 Status Register Configuration

Table 5-1 Register Filter Functions

Register filter	Function
Condition Register	Monitors the device status and changes real time in response to the device status. Thus, this register does not memorize the status.
Transition Filter	Sets the Condition Register data in the Event Register. The following three types of transition filters are available depending on which change of the Condition Register is to be evaluated. Positive direction change: The event becomes true only when the corresponding condition changes from false to true. Negative direction change: The event becomes true only when the corresponding condition changes from true to false. Bi-directional change: The event becomes true when a change arise either in the positive or negative direction.
Event Register	Memorizes output from Transition Filter.
Event Enable Register	Selects which bit of the Event Register is set to set the summary message true.

5.2. IEEE488.2-regulated Status Register

IEEE488.2 regulates the following two status registers.

Status byte register	This register can set RQS and 7 summary message bits. It is used as a pair with the service request enable register. When OR of both registers is not 0, SRQ turns ON. RQS is programmed in bit 6. This bit is used to report that a service request is given in the external controller.
Standard event status register	Stores 8 events which the device encounters as the standard events. The logical OR output bit is summarized and displayed in bit 5 of the status byte register as an ESB (Event Status Bit) summary message.

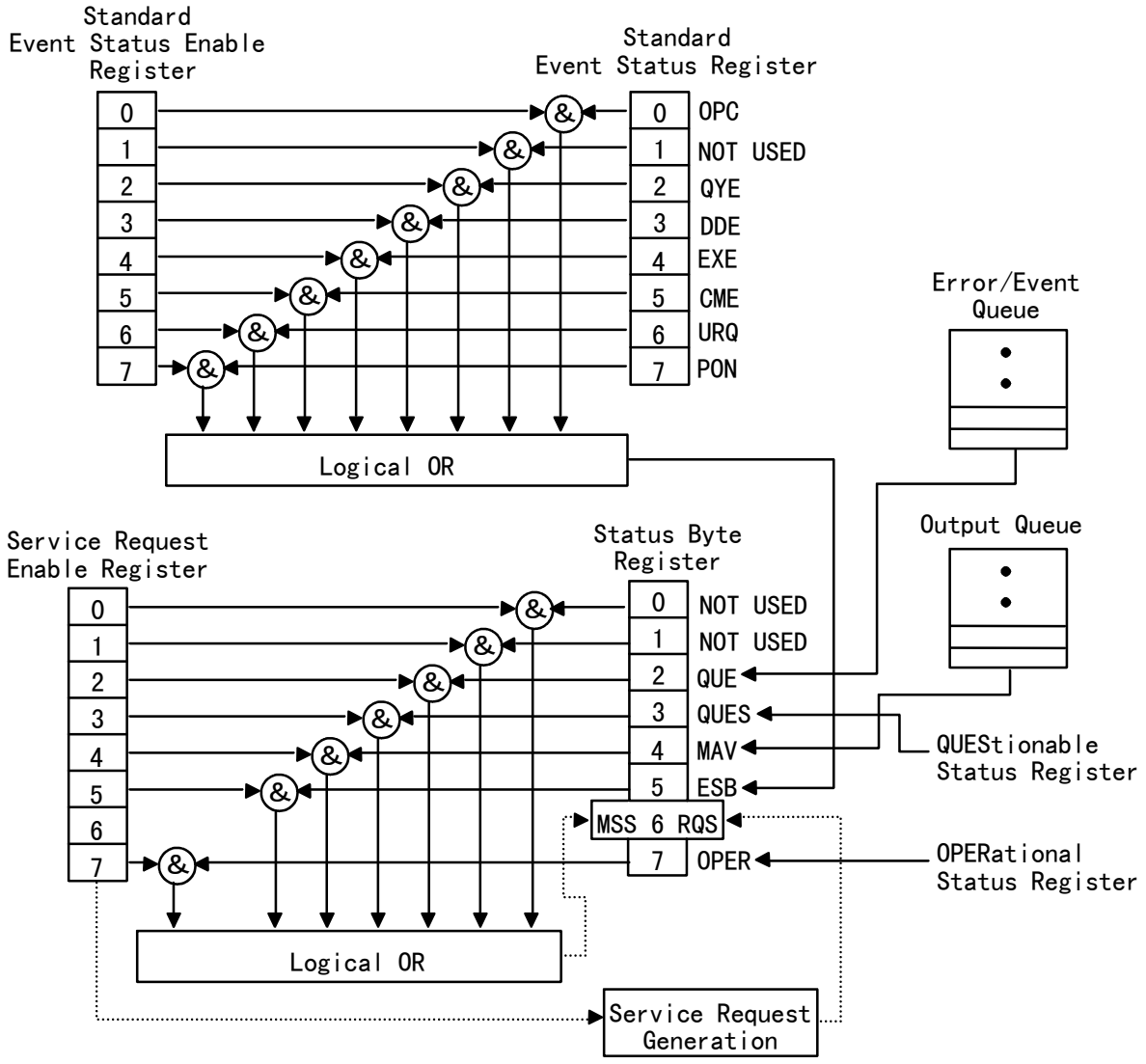


Fig. 5-3 IEEE488.2 Status Register

Table 5-2 Status Byte Register Bit Definition

Bit	Mnemonic	Description
2	QUE (Error/Event QUEue)	Indicates that the error and event queues are not empty.
3	QUES (QUEStionable status register summary)	QUEStionable status register summary
4	MAV (Message AVailable)	Indicates that the output queue is not empty. When the device is ready to receive the response message send-out request from the controller, the MAV summary message bit is set to 1 (TRUE). This message can be used to allow the controller to send the queue command to the device and wait until MAV turns TRUE.
5	ESB (Event Summary Bit)	Standard event status register summary
6	RQS (ReQuest Service) MSS (Master Summary Status)	RQS Returns a 7-bit status byte and this RQS to the controller in the serial pole mode. MSS Indicates that the device has reason to request at least one service. When inquiring the status byte by the *STB? command, this MSS summary message appears in bit 6 in place of the RQS message.
7	OPER (OPERation status register summary)	OPEration status register summary

Table 5-3 Standard Event Status Register Bit Definition

Bit	Mnemonic	Description
0	OPC(Operation Complete)	Indicates that all the specified operations are completed.
2	QYE(Query Error)	Indicates that a query error occurs.
3	DDE(Device-dependent Error)	Indicates that an error other than command error, query error or execution error occurs.
4	EXE(Execution Error)	Indicates that an execution error occurs.
5	CME(Command Error)	Indicates that a command error occurs.
6	URQ(User Request)	Indicates that a local control has been executed. Since this is not supported by the Instrument, bit 6 is always set to 0.
7	PON(Power on)	Indicates that the power supply turns from OFF to ON.

5.3. SCPI-regulated Status Register

SCPI regulates that the Instrument should contain the following registers in addition to those regulated in IEEE488.2.

Table 5-4 SCPI-regulated Status Registers

OPERation status register	Reports the device status such as measurement condition. You can use this register when the controller should wait until an operation such as Error/Alarm is completed, which takes long time.
QUEStionable status register	Reports the signal status such as measured result. You can use this register to send a service request to the controller when an error occurs.

■ OPERation status register

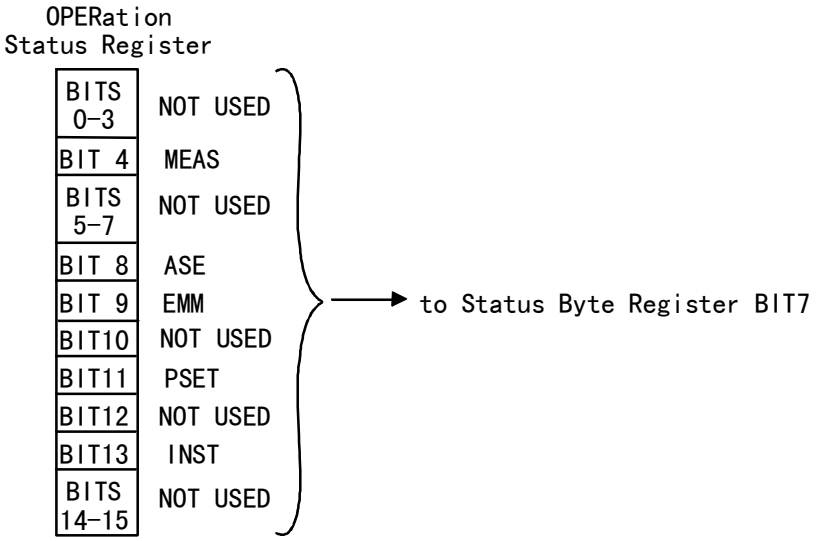


Fig. 5-4 OPERation Status Register

Table 5-5 OPERation Status Register Bit Definition

Bit	Mnemonic	Description
4	MEAS (MEASuring)	Indicates that the Instrument is measuring error/alarm.
8	ASE (Auto SEaching)	Indicates that the Instrument is in Auto Search.
9	EMM (Eye Margin Measuring)	Indicates that the Instrument is measuring eye margin.
11	PSET (Pattern SETting)	Indicates that the Instrument is setting pattern.
13	INST (INSTrument status register summary)	INSTrument status register summary

■ **QUEStionable status register**

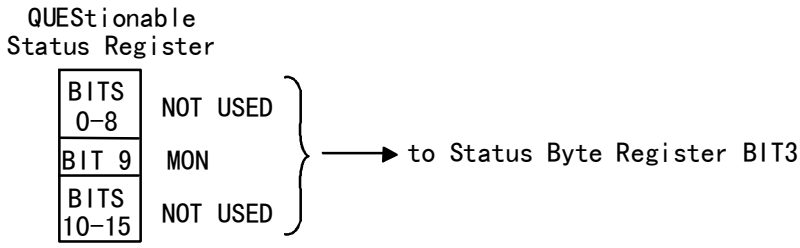


Fig. 5-5 QUEStionable Status Register

Table 5-6 QUEStionable Status Register Bit Definition

Bit	Mnemonic	Description
9	MON (MONitor status register summary)	MONitor status register summary

5.4. Device-specific Status Register

In addition to the IEEE488.2-regulated status registers and SCPI-regulated status registers mentioned above, the Instrument has the device-specific status registers listed below:

Table 5-7 Device-specific Status Registers

INSTrument status register	Reports the device status and measurement condition.
MONitor status register	Reports the signal status of the unit installed in the Instrument.
SLOT1 status register	Reports the signal status of the unit inserted in Slot1. Can read a Clock Generator alarm.
SLOT3 status register	Reports the signal status of the unit inserted in Slot3. Can read an alarm of PGG in Slot3.
3.2G PPG (Slot3) status register	Reports the signal status of the 3.2G PPG unit inserted in Slot3. Can read an alarm of PGG in Slot3.
SLOT4 status register	Reports the signal status of the unit inserted in Slot4. Can read an alarm/error of PGG in Slot4.
3.2G ED (Slot4) status register	Reports the signal status of the 3.2G ED unit inserted in Slot4. Can read an alarm/error of ED in Slot4.

■ INSTrument status register

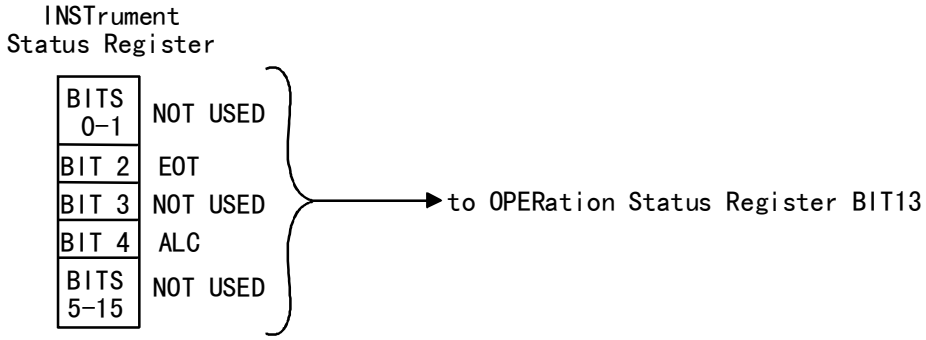


Fig. 5-6 INSTrument Status Register

Table 5-8 INSTrument Status Register Bit Definition

Bit	Mnemonic	Description
2	EOT (End of Test priod)	Indicates that the test (measurement) is completed. In the report measurement mode, this bit turns ON at every repeat cycle.
4	ALC (Alarm Change)	Indicates that an alarm changes.

■ **MONitor status register**

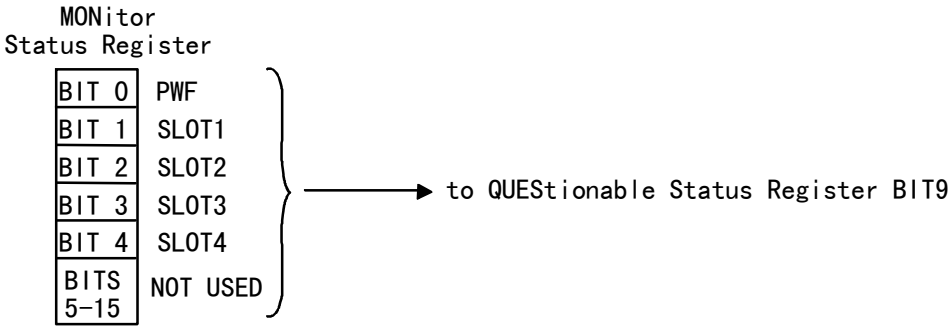


Fig. 5-7 MONitor Status Register

Table 5-9 MONitor Status Register Bit Definition

Bit	Mnemonic	Description
0	PWF (Power Fail)	Indicates that a power failure occurs.
1	SLOT1	SLOT1 status register summary Indicates that an event occurs in the unit (Clock Generator) inserted in Slot1.
2	SLOT2	SLOT2 status register summary
3	SLOT3	SLOT3 status register summary Indicates that an event occurs in the PPG unit inserted in Slot3.
4	SLOT4	SLOT4 status register summary Indicates that an event occurs in the ED unit inserted in Slot4.

■ **SLOT1 status register**

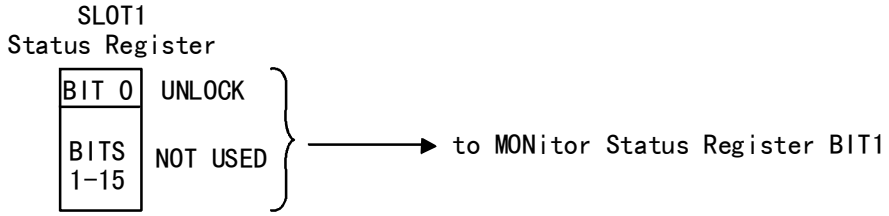


Fig. 5-8 SLOT1 Status Register

Table 5-10 SLOT1 Status Register Bit Definition

Bit	Mnemonic	Description
0	UNLOCK (PLL UNLOCK)	Indicates that PLL Unlock occurs.

■ SLOT3 status register/SLOT4 status register

The bit layout of the SLOT3 status register is the same as that of the SLOT4 status register.

As shown below, the Instrument assigns a different status register in which an event appears even to the same type of units, depending on the slot in which the unit is inserted.

- When the unit is inserted in Slot3
  An event occurrence appears in the appropriate bit of the Slot3 status register.
  Example: When monitoring alarm of 3.2G PPG inserted in Slot3, read the value of this status register.
- When the unit is inserted in Slot4
  An event occurrence appears in the appropriate bit of the Slot4 status register.
  Example: When monitoring alarm of 3.2G PPG inserted in Slot4, read the value of this status register.

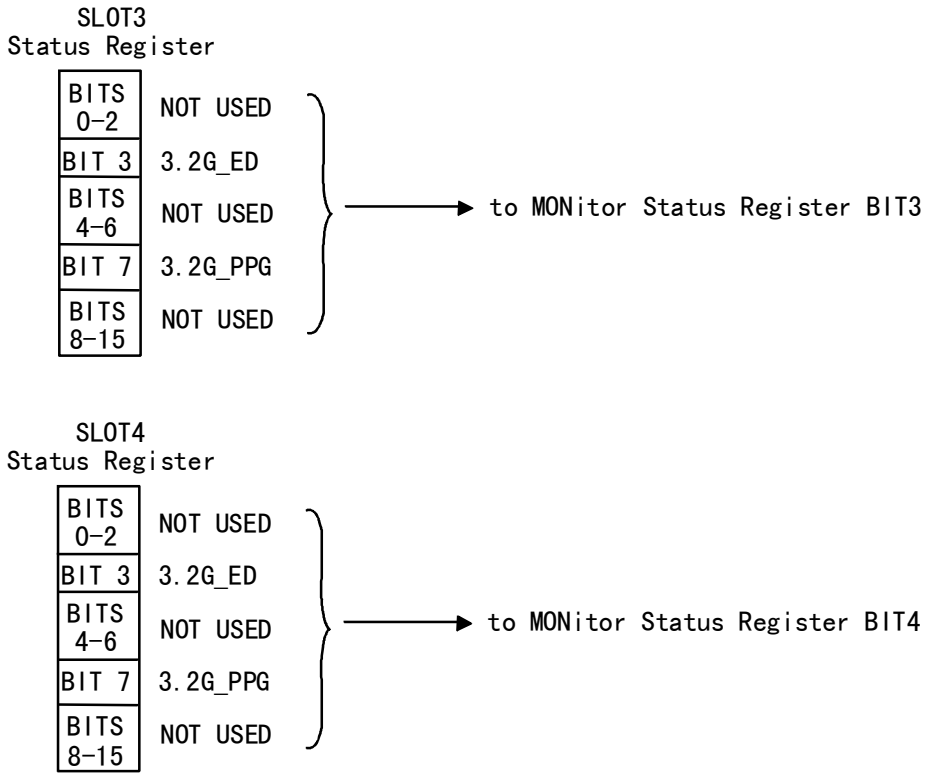


Fig. 5-9 SLOT3 Status Register/SLOT4 Status Register

Table 5-11 SLOT3 Status Register/SLOT4 Status Register Bit Definition

Bit	Mnemonic	Description
3	3.2G_ED	3.2G ED status register summary
7	3.2G_PPG	3.2G PPG status register summary

■ **3.2G ED status register/3.2G PPG status register**

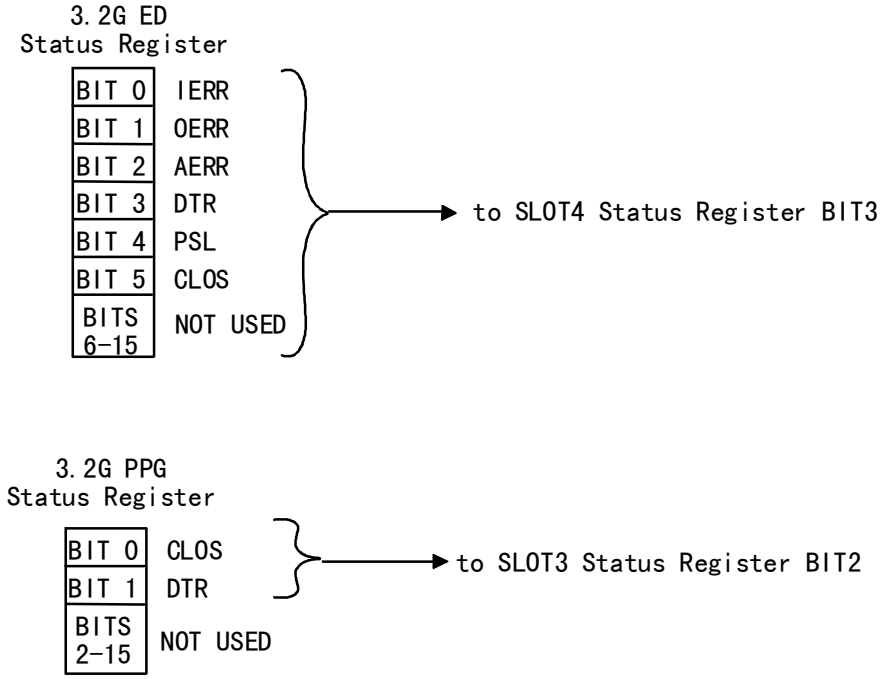


Fig. 5-10 3.2G ED Status Register/3.2G PPG Status Register

Table 5-12 3.2G ED Status Register Bit Definition

Bit	Mnemonic	Description
0	IERR	Insertion ERRor occurs.
1	OERR	Omission ERRor occurs.
2	AERR	All ERRor occurs.
3	DTR	Delay TRouble occurs.
4	PSL	Pattern Sync Loss occurs.
5	CLOS	Clock LOSs occurs.

Table 5-13 3.2G PPG Status Register Bit Definition

Bit	Mnemonic	Description
0	CLOS	Clock LOSs occurs.
1	DTR	Delay TRouble occurs.

5.5. Read/Write to/from Status Register

The following shows how to read and write the status register data.

Table 5-14 Reading and Writing the Status Registers

Register	Reading	Writing
Status byte register	Read the register bits using the serial pole. A 7-bit status byte and a RQS message bit are returned. The status byte value does not change. *STB? common query A numeric value composed of the status byte register value and the MSS summary message is returned.	Disabled
Service request enable register	*SRE? common query The register bits do not change.	*SRE common command
Standard status register	*ESR? common query The register bits are cleared after being read.	Disabled
Standard event status enable register	*ESE? common query The register bits do not change.	*ESE common command
SCPI event register	Device-specific command :STATus:...:EVENt? The register bits are cleared.	Disabled
SCPI condition register	Device-specific command :STATus:...:CONDition?	Disabled
SCPI enable register	Device-specific command :STATus:...:ENABle? The register bits do not change.	Device-specific command :STATus:...:ENABle
SCPI Transition filter	Device-specific command :STATus:...:PTRansition? :STATus:...:NTRansition? The register bits do not change.。	Device-specific command :STATus:..:PTRansition :STATus:..:NTRansition
Error/event queue	Device-specific command :SYSTem:ERRor?	Disabled

Notes:

- The SCPI...registers listed above refer to the SCPI-regulated status registers and device-specific status registers.
- See Chapter 4 for details on the common commands and queries. See Chapter 6 for details on the device-specific commands.

5.6. Clear/Reset Status Register

The following shows how to clear/reset the status registers and the range influenced by the clear/reset operation.

Table 5-15 Clearing and Resetting the Status Registers

Register	*RST	*CLS	Power ON	STATus: PRESet	Other method to clear register
Status byte register	No change	Clear	Clear	No change	
Service request enable register	No change	No change	Clear	No change	Executing *SER 0
Standard event status register	No change	Clear	Clear	No change	Cleared when an event is read by *ESR?
Standard event status enable register	No change	No change	Clear	No change	Executing *ESE 0
SCPI event register	No change	Clear	Clear	No change	Cleared when an event is read by :STATus:...:EVENt?
SCPI enable register	No change	No change	Reset	Reset	:Executing :STATus:...:ENABle 0
SCPI Transition filter	No change	No change	Reset	Reset	Executing :STATus:...:PTRansition 0 and :STATus:...:NTRansition 0
Error/event queue	No change	Clear	Clear	No change	Reading all events by :SYSTem:ERRor?

Notes :

- The SCPI...registers listed above refer to the SCPI-regulated status registers and device-specific status registers.

The following shows the reset values of the registers influenced by the :STATus:PRESet command.

Table 5-16 Values Reset by the :STATus:PRESet Command

Register	Enable/filter	Reset value
OPERational status register QUEStionable status register	Enable register PTRansition filter NTRansition filter	ALL 0 ALL 1 ALL 0
INSTrument status register	Enable register PTRansition filter NTRansition filter	ALL 1 ALL 1 ALL 0
Other status registers	Enable register PTRansition filter NTRansition filter	ALL 1 ALL 1 ALL 1

5.7. Native Related Status Register

5.7.1. ED Status Related

Describes the setting/query of the status register of the Slot 4 ED unit.

5.7.1.1. SRQ Service Request Enable Register (Service request enable register?)

Function Sets/queries the sending conditions of the service request.Switches ON and OFF the RQS bit by bit of the status byte. When a cause of a bit set at "1" in the service request enable register takes place, the RQS bit becomes "1" to notify the controller of interrupt.

Header	Program	Query	Response (No. of characters)
SRQ	SRQ△m0	SRQ?	SRQ△m1 (FIX2)

Parameter

m0 = <NR1>

Description of each bit(N)

BIT7 : Not used.
BIT6 : Not used.
BIT5 : COMMAND ERROR bit :weight32
BIT4 : BUSY bit :weight16
BIT3 : Not used.
BIT2 : Not used.
BIT1 : EXTENSION bit :weight2
BIT0 : Not used.

The value of m0

The value equaling 0 sets the sending conditions of the service request OFF.
The value equaling 1 signifies an unused bit.
The value equaling 2 switches the mode into a state of sending when the extension event status register is set.
The value equaling 4 signifies an unused bit.
The value equaling 8 signifies an unused bit.
The value equaling 16 switches the mode into a state of sending when data to be output to GPIB exists.
The value equaling 32 switches the mode into a state of sending when the standard event status register is set.
Setting is done by the sum of the weight.

m1 = <NR1>

Returns the sum of the weight of each bit.

5.7.1.1 SRQ Service Request Enable Register (Service request enable register?) (continued)

Application example

Program

・Sends the service request when the extension event status register is set.

```
>SRQ△2
```

Query

・Makes a query when the standard event status register is set as the sending condition.

```
> SRQ?
< SRQ△32
```

Remarks Receiving an SRQ order means that all the sending conditions previous set will be canceled.
Set the sum of the weight of each bit to set multiple conditions. All SRQ sending conditions are canceled immediately after the activation of the application and the reception of the device clear and the initialize orders.

5.7.1.2. STB Status Byte Register (Status byte register?)

Function Queries the standard status byte register. The same data as that read by the serial poll.

Header	Program	Query	Response (No. of characters)
STB	None	STB?	STB△m (FIX10)

Parameter m = <Binary numeral format>

```
#B**********
  ↑         ↑
BIT7       BIT0
```

Description of each bit(N)

BIT7 : Not used. (Always 0)
BIT6 : RQS bit
BIT5 : COMMAND ERROR bit
BIT4 : BUSY bit
BIT3 : Not used. (Always 0)
BIT2 : Not used. (Always 0)
BIT1 : EXTENSION bit
BIT0 : Not used. (Always 0)

Application example

Query

・Queries when the state of the status byte is 01010000.

```
> STB?
< STB△#B01010000
```

5.7.1.3. ESE Standard Event Status Enable Register (Event status enable register? (standard))

Function Sets/queries the sending conditions of the standard event status register. Only the bits enabled here constitute the target of sending to the high-order register.

Header	Program	Query	Response (No. of characters)
ESE	ESE△m0	ESE?	ESE△m1 (FIX2)

Parameter m0 = <NR1>

Description of each bit(N)

BIT7 : Not used.
BIT6 : Not used.
BIT5 : CMD bit : weight 32
BIT4 : EXE bit : weight 16
BIT3 : DDE bit : weight 8
BIT2 : QYE bit : weight 4
BIT1 : Not used.
BIT0 : Not used.

The value of m0

The value equaling 0 sets the sending conditions of the standard event status OFF.
The value equaling 1 signifies an unused bit.
The value equaling 2 signifies an unused bit.
The value equaling 4 switches the mode into a state of sending when the talker specification is made though the transmitter buffer is vacant, or when the transmitter buffer overflows.
The value equaling 8 switches the mode into a state of sending when the command received falls outside the conditions for use.
The value equaling 16 switches the mode into a state of sending when it is used for the command which disables the use of the number range error and the HEX numerical value.
The value equaling 32 sets the mode into a state of sending when a command header error is detected.
Sets the sum of the weight of each bit.

m1 = <NR1>

Returns the sum of the weight of each bit.

Application example

Program

・Sets the QYE bit and the EXE bit as the sending conditions.

```
>ESE△20
```

Query

・Makes a query when the DDE bit is set as the sending condition.

```
> ESE?
< ESE△△8
```

5.7.1.3. ESE Standard Event Status Enable Register (continued)

Remarks Receiving an ESE order means that all the sending conditions previous set will be canceled. Set the sum of the weight of each bit to set multiple conditions. All ESE sending conditions are canceled immediately after the activation of the application and the reception of the device clear and the initialize orders.

5.7.1.4. ESR Standard Event Status Register (Event status byte register? (standard))

Function Queries the standard event status byte register.

Header	Program	Query	Response (No. of characters)
ESR	None	ESR?	ESR△m (FIX10)

Parameter m = <Binary numeral format>

#B**********

↑ ↑

BIT7 BIT0

Description of each bit(N)

BIT7 : Not used. (Always 0)
BIT6 : Not used. (Always 0)
BIT5 : CME bit
BIT4 : EXE bit
BIT3 : DDE bit
BIT2 : QYE bit
BIT1 : Not used. (Always 0)
BIT0 : Not used. (Always 0)

Application example

Query

・Queries when the state of the standard event status byte is 00010000.

```
> ESR?
< ESR△#B00010000
```

Remarks All the bits of the standard event status byte are reset when read out by the ESR command. They are not cleared when the device clear, the interface clear or the initialize order is received. In addition, the standard event status byte is not related to the serial poll.

5.7.1.5. EES Extended Event Status Enable Register
(Event status enable register? (extension))

Function Sets/queries the sending conditions of the extension event status register. Only the bits enabled here constitute the target of sending to the high-order register.

Header	Program	Query	Response (No. of characters)
EES	EES△m0	EES?	EES△m1 (FIX2)

Parameter m0 = <NR1>

Description of each bit(N)

BIT7 : Not used.
BIT6 : MEASURE bit : weight64
BIT5 : FILE ACCESS bit : weight32
BIT4 : PATTERN bit : weight16
BIT3 : DELAY bit : weight8
BIT2 : SYNC LOSS bit : weight4
BIT1 : CLOCK LOSS bit : weight2
BIT0 : ERRORS bit : weight1

The value of m0

The value equaling 0 sets the sending conditions of the extension event status OFF.

The value equaling 1 switches the mode into a state of sending when an error is detected in the state free of errors.

The value equaling 2 switches the mode into a state of sending when a change in the state, clock interrupt or clock recovery, is detected.

The value equaling 4 switches the mode into a state of sending when a change in the state, stepping out or synchronization, is detected.

The value equaling 8 switches the mode into a state of sending after the Clock Delay of the ED unit is set.

The value equaling 16 switches the mode into a state of sending when the setting of the PRGM pattern of ED is completed.

The value equaling 32 switches the mode into a state of sending when the FD is accessed.

The value equaling 64 switches the mode into a state of sending when the measurement is finished.

Sets the sum of each bit.

m1 = <DECIMAL NUMERIC DATA>

Returns the sum of the weight of each bit.

5.7.1.5. EES Extended Event Status Enable Register (Continued)

Application example

Program

・Sets the Pattern bit and the Errors bit as the sending conditions.

```
>EES△17
```

Query

・Makes a query when the SYNC LOSS bit is set as the sending condition.

```
> EES?
< EES△△4
```

Remarks Receiving an EES order means that all the sending conditions previous set will be canceled. Set the sum of the weight of each bit to set multiple conditions. All EES sending conditions are canceled immediately after the activation of the application and the reception of the device clear and the initialize orders.

5.7.1.6. EER Extended Event status register (Event status byte register?(extension))

Function Queries the extension event status byte register.

Header	Program	Query	Response (No. of characters)
EER	None	EER?	EER△m (FIX10)

Parameter m = <Binary numeral format>

```
#B**********
  ↑         ↑
BIT7      BIT0
```

Description of each bit(N)

BIT7 : Not used. (Always 0)
BIT6 : MEASURE bit
BIT5 : FILE ACCESS bit
BIT4 : PATTERN bit
BIT3 : DELAY bit
BIT2 : SYNC LOSS bit
BIT1 : CLOCK LOSS bit
BIT0 : ERRORS bit

Application example

Query

・Queries when the state of the extension event status byte is 00010000.

```
> EER?
< EER△#B00010000
```

Remarks All the bits of the extension event status byte are reset when read out by the EER command. They are not cleared when the device clear, the interface clear or the initialize order is received. In addition, the extension event status byte is not related to the serial poll.

5.7.2. PPG/Synthesizer Status Related

Describes the setting/query of the status register of the synthesizer built in Slot 1 and the Slot 3 PPG unit.

5.7.2.1. SRQ Service Request Enable Register (Service request enable register?)

Function Sets/queries the sending conditions of the service request. Switches ON and OFF the RQS bit by bit of the status byte. When a cause of a bit set at "1" in the service request enable register takes place, the RQS bit becomes "1" to notify the controller of interrupt.

Header	Program	Query	Response (No. Of characters)
SRQ	SRQ△m0	SRQ?	SRQ△m1 (FIX2)

Parameter m0 = <NR1>

Description of each bit(N)

BIT7 : Not used.
BIT6 : Not used.
BIT5 : COMMAND ERROR bit : weight32
BIT4 : BUSY bit : weight16
BIT3 : Not used.
BIT2 : Not used.
BIT1 : EXTENSION bit : weight2
BIT0 : Not used.

The value of m0

The value equaling 0 sets the sending conditions of the service request OFF.
The value equaling 1 signifies an unused bit.
The value equaling 2 switches the mode into a state of sending when the extension event status register is set.
The value equaling 4 signifies an unused bit.
The value equaling 8 signifies an unused bit.
The value equaling 16 switches the mode into a state of sending when data to be output to GPIB exists.
The value equaling 32 switches the mode into a state of sending when the standard event status register is set.
Sets the sum of each bit.

m1 = <NR1>

Returns the sum of the weight of each bit.

Application example

Program

・Sends the service request when the extension event status register is set.

```
>SRQ△2
```

Query

・Makes a query when the standard event status register is set as the sending condition.

```
> SRQ?
< SRQ△32
```

Remarks Receiving an SRQ order means that all the sending conditions previous set will be canceled. Set the sum of the weight of each bit to set multiple conditions. All SRQ sending conditions are canceled immediately after the activation of the application and the reception of the device clear and the initialize orders.

5.7.2.2. STB Status Byte Register (Status byte register?)

Function Queries the standard status byte register. The same data as that read by the serial poll.

Header	Program	Query	Response (No. of characters)
STB	None	STB?	STB△m (FIX10)

Parameter m = <Binary numeral format>

#B**********

↑ ↑

BIT7 BIT0

Description of each bit(N)

BIT7 : Not used. (Always 0)
BIT6 : RQS bit
BIT5 : COMMAND ERROR bit
BIT4 : BUSY bit
BIT3 : Not used. (Always 0)
BIT2 : Not used. (Always 0)
BIT1 : EXTENSION bit
BIT0 : Not used. (Always 0)

Application example

Query

・Queries when the state of the status byte is 01010000.

```
> STB?
< STB△#B01010000
```

5.7.2.3. ESE Standard Event Status Enable Register (Event status enable register? (standard))

Function Sets/queries the sending conditions of the standard event status register.Only the bits enabled here constitute the target of sending to the high-order register.

Header	Program	Query	Response (No. of characters)
ESE	ESE△m0	ESE?	ESE△m1 (FIX2)

Parameter m0 = <NR1>

Description of each bit(N)

BIT7 : Not used.
BIT6 : Not used.
BIT5 : CMD bit : weight32
BIT4 : EXE bit : weight16
BIT3 : DDE bit : weight8
BIT2 : QYE bit : weight4
BIT1 : Not used.
BIT0 : Not used.

The value of m0

The value equaling 0 sets the sending conditions of the standard event status OFF.
The value equaling 1 signifies an unused bit.
The value equaling 2 signifies an unused bit.
The value equaling 4 switches the mode into a state of sending when the talker specification is made though the transmitter buffer is vacant, or when the transmitter buffer overflows.
The value equaling 8 switches the mode into a state of sending when the command received falls outside the conditions for use.
The value equaling 16 switches the mode into a state of sending when it is used for the command which disables the use of the number range error and the HEX numerical value.
The value equaling 32 sets the mode into a state of sending when a command header error is detected.
Sets the sum of the weight of each bit.

m1 = <NR1>

Returns the sum of the weight of each bit.

Application example

Program

・Sets the QYE bit and the EXE bit as the sending conditions.

```
>ESE△20
```

Query

・Makes a query when the DDE bit is set as the sending condition.

```
> ESE?
< ESE△△8
```

5.7.2.3. ESE Standard Event Status Enable Register (Continued)

Remarks Receiving an ESE order means that all the sending conditions previous set will be canceled. Set the sum of the weight of each bit to set multiple conditions. All ESE sending conditions are canceled immediately after the activation of the application and the reception of the device clear and the initialize orders.

5.7.2.4. ESR Standard Event Status Register (Event status byte register? (standard))

Function Queries the standard event status byte register.

Header	Program	Query	Response (No. of characters)
ESR	None	ESR?	ESR△m (FIX10)

Parameter m = <Binary numeral format>

#B**********

↑ ↑

BIT7 BIT0

Description of each bit(N)

BIT7 : Not used. (Always 0)
BIT6 : Not used. (Always 0)
BIT5 : CME bit
BIT4 : EXE bit
BIT3 : DDE bit
BIT2 : QYE bit
BIT1 : Not used. (Always 0)
BIT0 : Not used. (Always 0)

Application example

Query

・Makes a query when the state of the standard event status byte is 00010000.

```
> ESR?
< ESR△#B00010000
```

Remarks All the bits of the standard event status byte are reset when read out by the ESR command. They are not cleared when the device clear, the interface clear or the initialize order is received. In addition, the standard event status byte is not related to the serial poll.

5.7.2.5. EES Extended Event Status Enable Register (Event status enable register? (extension))

Function Sets/queries the sending conditions of the extension event status register.Only the bits enabled here constitute the target of sending to the high-order register.

Header	Program	Query	Response (No. of characters)
EES	EES△m0	EES?	EES△m1 (FIX2)

Parameter m0 = <NR1>

Description of each bit(N)

BIT7 : Not used.
BIT6 : MEASURE bit : weight64
BIT5 : FILE ACCESS bit : weight32
BIT4 : PATTERN bit : weight16
BIT3 : DELAY bit : weight8
BIT2 : PLL UNLOCK bit : weight4
BIT1 : BACKUP ERROR bit : weight2
BIT0 : Not used.

The value of m0

The value equaling 0 sets the sending conditions of the extension event status OFF.
The value equaling 1 signifies an unused bit.
The value equaling 2 switch the mode into a state of sending when an backup error is detected.
The value equaling 4 switches the mode into a state of sending when the PLL UNLOCK is detected.
The value equaling 8 switches the mode into a state of sending after Clock Delay of the PPG unit is set.
The value equaling 16 switches the mode into a state of sending when the setting of the PRGM pattern of the PPG is completed.
The value equaling 32 sets the mode into a state of sending when the FD is accessed.
The value equaling 64 sets the mode into a state of sending when masurement is completed.
Sets the sum of the weight of each bit.

m1 = <NR1>

Returns the sum of the weight of each bit.

5.7.2.5. EES Extended Event Status Enable Register (Event status enable register? (extension)) (continued)

Application example

Program

・Sets the Pattern bit and the PLL UNLOCK bit as the sending conditions.

```
>EES△20
```

Query

・Makes a query when the Pattern bit is set as the sending condition.

```
> EES?
< EES△16
```

Remarks Receiving an EES order means that all the sending conditions previous set will be canceled. Set the sum of the weight of each bit to set multiple conditions. All ESE sending conditions are canceled immediately after the activation of the application and the reception of the device clear and the initialize orders.

5.7.2.6. EER Extended Event Status Register (Event status byte register? (extension))

Function Queries the extension event status byte register.

Header	Program	Query	Response (No. of characters)
EER	None	EER?	EER△m (FIX10)

Parameter m = <Binary numeral format>

```
#B**********
  ↑         ↑
BIT7      BIT0
```

Description of each bit(N)

BIT7 : Not used.
BIT6 : Not used.
BIT5 : FILE ACCESS bit
BIT4 : PATTERN bit
BIT3 : DELAY bit
BIT2 : PLL UNLOCK bit
BIT1 : BACKUP ERROR bit
BIT0 : Not used.

Application example

Query

・Makes a query when the state of the extension event status byte is 00010000.

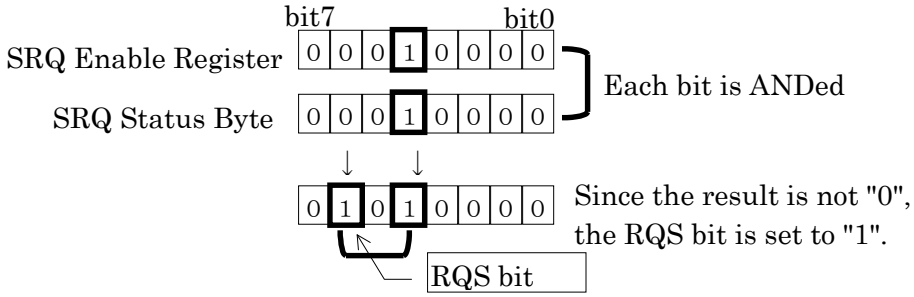
```
> EER?
< EER△#B00010000
```

Remarks All the bits of the extension event status byte are reset when read out by the EER command. They are not cleared when the device clear, the interface clear or the initialize order is received. In addition, the extension event status byte is not related to the serial poll.

5.7.3. Description of Each Register and Status Byte

5.7.3.1. SRQ Status Byte

Status bytes is the information for the data that is output on the bus when serial polling is received from the controller. When an event is generated, it is ANDed with the SRQ enable byte. If the result is not "0", the RQS bit (bit 6) is set to "1".

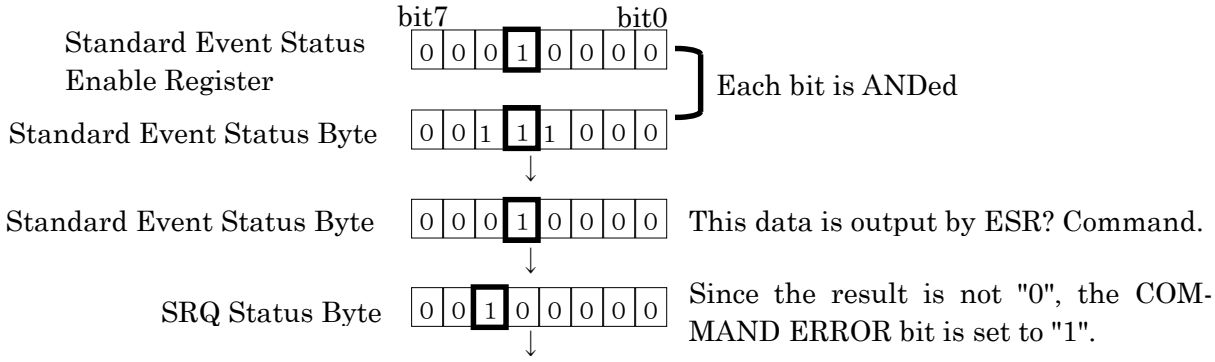


Name	bit	Function
	7	Not used.
RQS	6	This bit indicates that a service request is issued. It is set when the bit set by SRQ command was set from 0 to 1, and is reset when there is no events requesting service or after the controller has read the status. It is also reset when a device clear or initialize command is received.
COMMAND ERROR	5	This bit indicates the error state. It is set when the condition is satisfied by standard event status byte and standard event status enable register conditions (See paragraph 5.7.1.3 or 5.7.2.3. It is reset when the standard event status byte is read. It is also reset when a device clear or initialize command is received. When this bit is set after setting the COMMAND ERROR bit (weight 32) by SRQ command, SRQ is sent.
BUSY	4	This bit I set when there is data to be sent on the GPIB. It is reset when data transfer is completed or when there is no data to be sent. It is also reset when a device clear or initialize command is received. When this bit is set after setting the BUSY bit (weight 16) by SRQ command, SRQ is sent
	3	Not used.
	2	Not used.
EXTENSION	1	This bit indicates the MP1632C's unique status. It is set when the extended event status byte and extended event status enable register conditions are satisfied (See paragraph 5.7.1.5 or 5.7.2.5). It is reset when the extended event status byte is read. It is also reset when the a device clear or initialize command is received. When this bit is set after setting the EXTENSION bit (weight 2) by SRQ command, SRQ is sent.
	0	Not used.

5.7.3.2. Standard Event Status Byte

This byte is the information for the data which is to be output on the bus when a data request message is received. When an event is generated, it is ANDed with the standard event status enable register. If the result is not "0", the COMMAND ERROR bit of the SRQ status byte is set to "1".

Note:The standard event status enable register is reset immediately after the activation of the application and the reception of the device clear and the initialize orders, and all ESE sending conditions are canceled.
Each bit of the standard event status byte is reset when data is read by data request message, and, not reset even if a device clear, interface clear or initialize command is received.



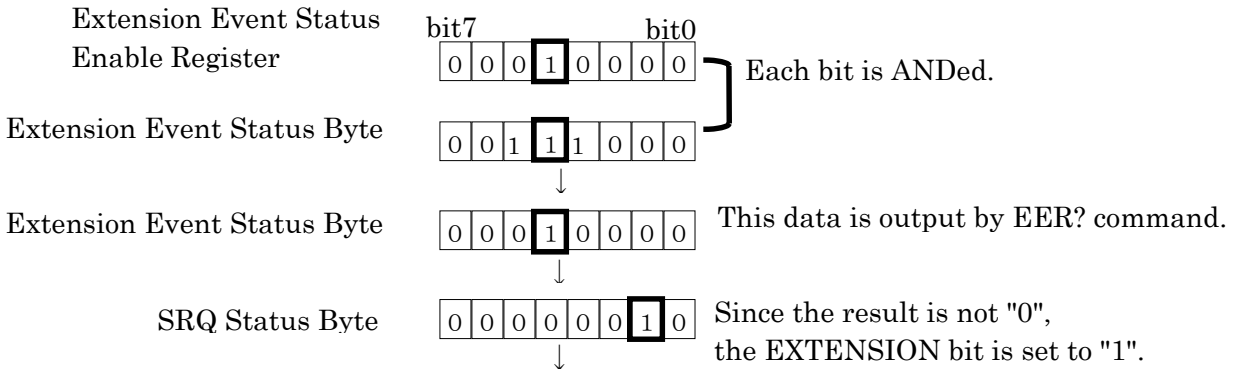
Name	bit	Function
	7	Not used.
	6	Not used.
CME	5	CME : Command error This bit is set when an undefined header is received. It is reset when the standard event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the CME bit (weight 32) by ESE command, the COMMAND ERROR bit of the status byte is set.
EXE	4	EXE = Execution error This bit is set when there is an error after the header (value range error or HEX value used in a command in which HEX cannot be used). It is reset when the standard event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the EXE bit (weight 16) by ESE command, the COMMAND ERROR bit of the status byte is set.

Name	bit	Function
DDE	3	DDE = Device dependent error This bit is set when the received command is outside the usage condition. It is reset when the standard event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the DDE bit (weight 8) by ESE command, the COMMAND ERROR bit of the status byte is set.
QYE	2	QYE = Query error This bit is set when the instrument is designated a talker even though the send buffer is empty or when the send buffer has overflowed. It is reset when the standard event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the QYE bit (weight 4) by ESE command, the COMMAND ERROR bit of the status byte is set.
	1	Not used.
	0	Not used.

5.7.3.3. Extended Event Status Byte

This byte is the information for the data that is sent to the bus when a data request message is received. When an event is generated, it is ANDed with the extended event status enable register. If the result is not "0", the EXTENSION bit of the SRQ status byte is set to "1".

Note:The extended event status enable register is reset immediately after the activation of the application and the reception of the device clear and the initialize orders, and all EES sending conditions are canceled.
Each bit of the standard event status byte is reset when data is read by data request message, and, not reset even if a device clear, interface clear or initialize command is received.



Extended Event Status Byte of PPG / Synthesizer

Name	bit	Function
	7	Not used.
	6	Not used.
FILE ACCESS	5	This bit is set at the end of a floppy disk access. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the FILE ACCESS bit (weight 32) by EES command, the status byte EXTENSION bit is set.
PATTERN	4	This bit is set at the end of programmable pattern setting. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the PATTERN bit (weight 16) by EES command, the status byte EXTENSION bit is set.

Name	bit	Function
DELAY	3	This bit is set at the end of clock setting. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the DELAY bit (weight 8) by EES command, the status byte EXTENSION bit is set.
PLL UNLOCK	2	This bit is set when the internal clock circuit PLL is in th UNLOCK state. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the PLL UNLOCK bit (weight 4) by EES command, the status byte EXTENSION bit is set.
BACKUP ERROR	1	This bit is set when the contents of the backup RAM are checked at power ON and then an error is detected. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the BACKUP bit (weight 2) by EES command, the status byte EXTENSION bit is set.
	0	Not used.

Extended Event Status Byte of ED

Name	bit	Function
	7	Not used.
MEASURE	6	This bit is set when measurement is complete. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the MEASURE bit (weight 64) by EES command, the status byte EXTENSION bit is set.
FILE ACCESS	5	This bit is set at the end of a floppy disk access. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the FILE ACCESS bit (weight 32) by EES command, the status byte EXTENSION bit is set.
PATTERN	4	This bit is set at the end of programmable pattern setting. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the PATTERN bit (weight 16) by EES command, the status byte EXTENSION bit is set.
DELAY	3	This bit is set at the end of clock setting. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the DELAY bit (weight 8) by EES command, the status byte EXTENSION bit is set.
SYNC LOSS	2	This bit is set when synchronization is lost or recovered. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the SYNC LOSS bit (weight 4) by EES command, the status byte EXTENSION bit is set.
CLOCK LOSS	1	This bit is set when clock is lost or recovered. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the CLOCK LOSS bit (weight 2) by EES command, the status byte EXTENSION bit is set.
ERRORS	0	This bit is set the first time an error is detected. It is reset when the extended event status byte is read. It is not reset even if a device clear, interface clear, or initialize command is received. When this bit is set after setting the ERROR bit (weight 1) by EES command, the status byte EXTENSION bit is set.

5.7.4. Configuration of Status Registers

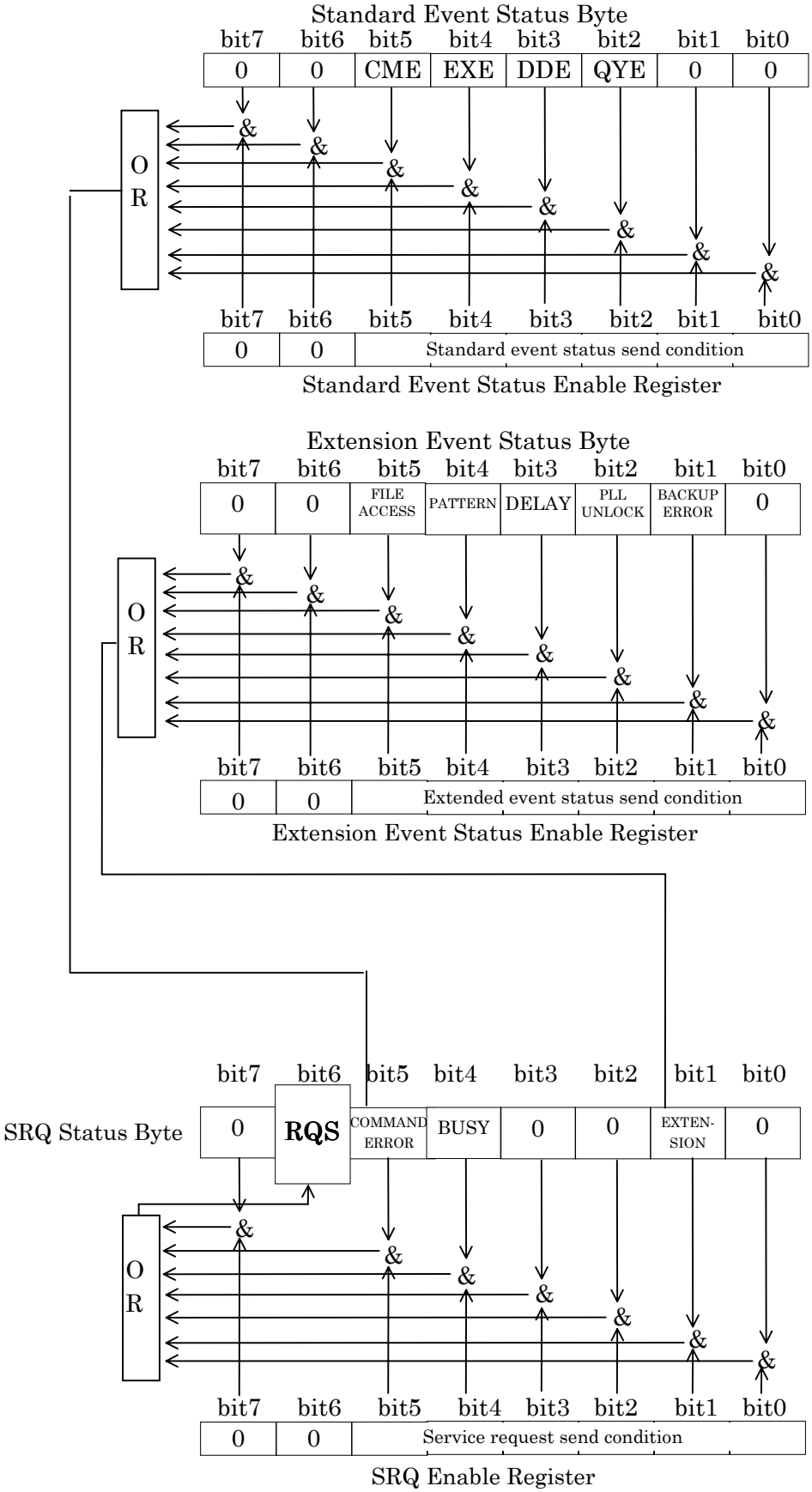


Fig. 5-11 Configuration of each status byte and enable register(PPG/Synthesizer)

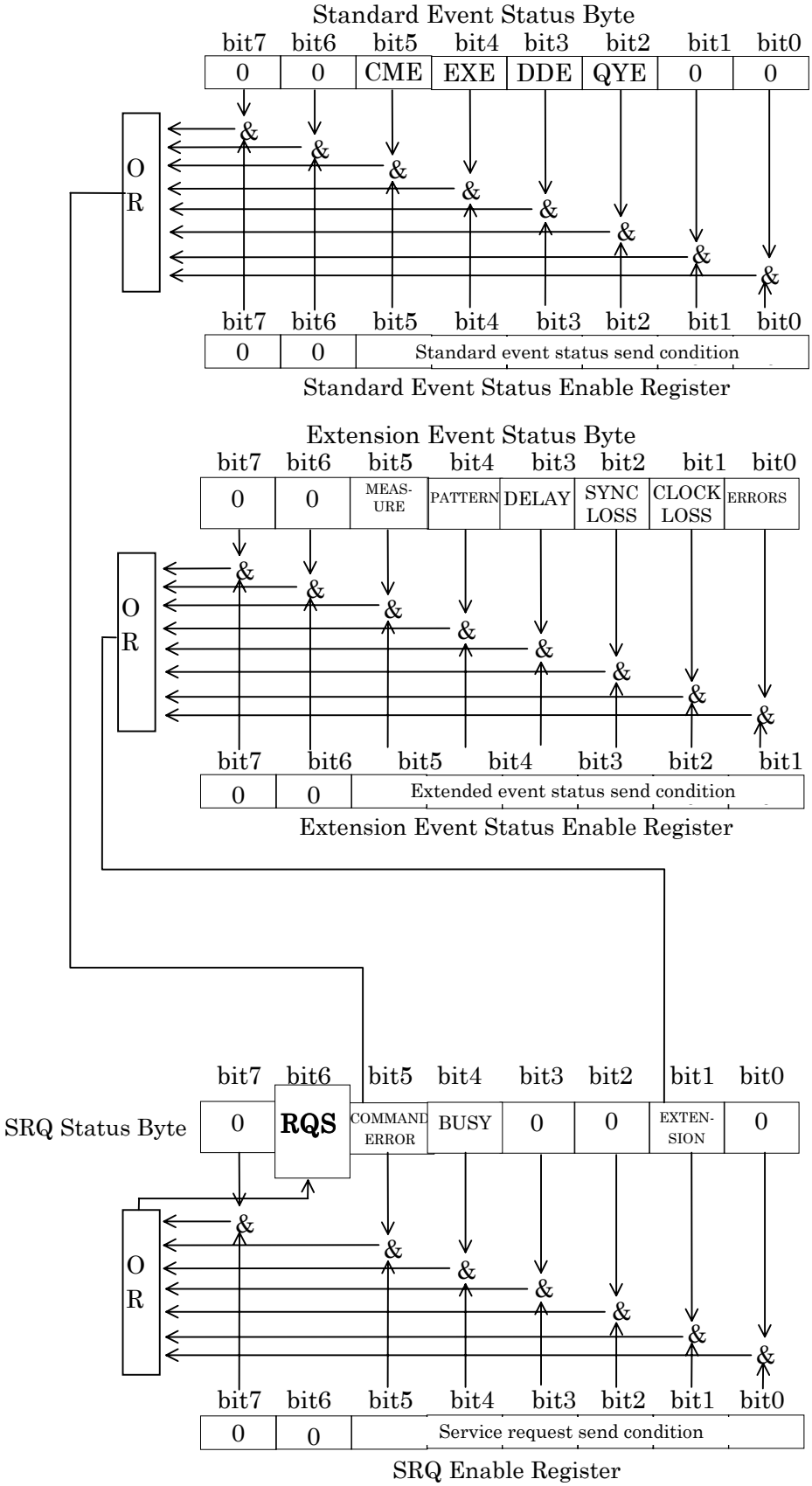


Fig. 5-12 Configuration of each status byte and enable register(ED)

CHAPTER 6 DETAILS ON DEVICE MESSAGES (SCPI)

6.1. Command List

6.1.1.Subsystem configuration

The Instrument's device-specific commands conform to SCPI.

SCPI configures commands by roughly classifying the device functions into several subsystems. Under each subsystem, the commands for detail functions are created to form a command tree structure.

The Instrument has subsystem that performs the unit-specific settings and queries and those which perform the settings and queries unrelated to the unit.

Table 6-1 Subsystem Configuration

	3.2G PPG	3.2G ED	3.2G Internal synthesizer
Theoretical signal setting (ex.: pattern setting)	SOURce	SENSe	
Electrical signal setting (ex.: Clock I/F setting)	OUTPut	INPut OUTPut	OUTPut
Measurement result read	CALCulate	CALCulate	CALCulate
Common setting	INSTrument	INSTrument	
Screen operation	DISPlay		
Other settings (ex.: print, date)	SYSTem		
Status register setting	STATus		

6.1.2. Command Tree

From the next page, the Instrument's command tree is described in the following order:

1. 3.2G PPG (MU163220C) unit setting/query commands
2. 3.2G ED (MU163240C) unit setting/query commands
3. 3.2G Internal synthesizer option (OPT03) setting/query commands
4. Other commands

3.2G PPG (MU163220C) unit setting/query command tree

■ **SOURce subsystem**

KEYWORD	PARAMETER FORM	Page
:SOURce		
:PATTern		
:EADDition		6-30
:SET	<boolean>	
:SET?		
:SINGle		
:RATE	<mode>	
:RATE?		
:ROUTe	<route>	
:ROUTe?		
:LOGic		6-31
:PRBS	<mark>	
:PRBS?		
:PRGM	<mark>	
:PRGM?		

■ OUTPut subsystem

■ **CALCulation subsystem**

■ **INSTrument subsystem**

3.2G ED (MU163240C) unit setting/query command tree

■ **SENSe subsystem**

KEYWORD	PARAMETER FORM	Page
:SENSe		6-41
:PATTern		6-41
:IMODe	<mode>	
:IMODe?		
:TYPE	<type>	
:TYPE?		
:PRBS		6-42
:MRATio	<mratio>	
:MRATio?		
:BSHift	<numeric>	
:BSHift?		
:ZSUBstitute		6-42
:LENGth	<numeric>	
:LENGth?		
:ZLENgth	<numeric>	
:ZLENgth?		
:LOGic	<logic>	
:LOGic?		
:PROGram		6-43
:LENGth	<numeric>	
:LENGth?		
:LOGic	<logic>	
:LOGic?		
:SYNC		6-44
:ASYNc	<boolean>	
:ASYNc?		
:THReshold		6-44
:LOSS	<thre>	
:LOSS?		
:GAIN	<thre>	
:GAIN?		
:INTernal	<boolean>	
:INTernal?		
:PSMode	<mode>	
:PSMode?		
:FLENgth	<numeric>	
:FLENgth?		

KEYWORD	PARAMETER FORM	Page
:SENSe		
:MEASure		6-49
:TEST	<item>	
:TEST?		
:STARt		
:STOP		
:MREStart	<boolean>	
:MREStart?		
:EALarm		6-50
:MODE	<mode>	
:MODE?		
:PERiod	<d>,<h>,<m>,<s>	
:PERiod?		
:BTIMe		6-51
:SET	<boolean>	
:SET?		
:STARt	<y>,<m>,<d>,<h>,<m>,<s>	
:STARt?		
:ERRor		6-51
:TYPE	<type>	
:TYPE?		
:MASK		6-52
:ROUTe	<route>,<boolean>	
:ROUTe?		
:STARt?		
:STOP?		
:STATe?		
:ELAPsed?		
:TIMed?		
:ITIMe?		
:AOCCur?	<alarm>	
:ARECver?	<alarm>	

KEYWORD	PARAMETER FORM	Page
:SENSe		
:MEASure		
:EMARgin		6-54
:MODE	<mode>	
:MODE?		
:TYPE	<type>	
:TYPE?		
:MARGin		6-55
:THReshold	<thre>	
:THReshold?		
:RESolution	<type>	
:RESolution?		
:DIAGram		6-56
:THReshold	<thre>,<boolean>	
:THReshold?		
:POINt?	<thre>	
:STATe?		
:STARt?		
:STOP?		
:ELAPsed?		
:ASEarch		6-58
:MODE	<mode>	
:MODE?		
:STARt		
:STOP		
:PATTern	<boolean>	
:PATTern?		
:STATe?		

■ INPut subsystem

■ CALCulation subsystem

■ OUTPut subsystem

■ INSTrument subsystem

3.2G Internal synthesizer option (OPT03) setting/query command tree

■ **OUTPut subsystem**

KEYWORD	PARAMETER FORM	Page
:OUTPut		6-73
:CLOCk		6-73
:FREQuency	<numeric>	
:FREQuency?		
:RCLock		6-73
:SELect	<clock>	
:SELect?		

■ **CALCulation subsystem**

KEYWORD	PARAMETER FORM	Page
:CALCulate		6-74
:DATA		6-74
:MONitor?	<string>	

Other setting/query command tree

■ **DISPlay subsystem**

■ **DISPlay subsystem**

KEYWORD	PARAMETER FORM	Page
:DISPlay		
:RESult		
:EMARgin		6-81
:ERATe	<erate>,<boolean>	
:ERATe?		
:SCALe		6-81
:ASCale		
:VOLTage	<min>,<step>	
:VOLTage?		
:PHASe	<min>,<step>	
:PHASe?		
:MARKer		6-81
:SET	<marker>,<boolean>	
:SET?	<marker>	
:MODE	<mode>	
:MODE?		
:ERATe	<erate>	
:ERATe?		
:MOVE	<marker>,<mode>	
:POSition?	<marker>	
:TEMPlate		6-84
:SELect	<select>,<boolean>	
:SELect?		
:MOVE	<template>,<mode>	
:PNUMber	<template>,<point>	
:PNUMber?	<template>	
:OFFSet		6-85
:VOLTage	<template>,<voltage>	
:VOLTage?	<template>	
:PHASe	<template>,<phase>	
:PHASe?	<template>	
:POINt		6-86
:VOLTage	<template>,<point>,<voltage>	
:VOLTage?	<template>,<point>	
:PHASe	<template>,<point>,<phase>	
:PHASe?	<template>,<point>	
:CONNect	<template>,<boolean>	
:SELect?	<template>	

■ **SYSTem subsystem**

■ STATus subsystem

KEYWORD	PARAMETER FORM	Page
:STATus		6-100
:PRESet		
:OPERation		6-100
[:EVENt]?		
:CONDition?		
:ENABle	<numeric>	
:ENABle?		
:PTRansition	<numeric>	
:PTRansition?		
:NTRansition	<numeric>	
:NTRansition?		
:INSTrument		6-101
[:EVENt]?		
:CONDition?		
:ENABle	<numeric>	
:ENABle?		
:PTRansition	<numeric>	
:PTRansition?		
:NTRansition	<numeric>	
:NTRansition?		

KEYWORD	PARAMETER FORM	Page
:STATus		
:QUEStionable		6-101
[:EVENt]?		
:CONDition?		
:ENABle	<numeric>	
:ENABle?		
:PTRansition	<numeric>	
:PTRansition?		
:NTRansition	<numeric>	
:NTRansition?		
:MONitor		6-102
[:EVENt]?		
:CONDition?		
:ENABle	<numeric>	
:ENABle?		
:PTRansition	<numeric>	
:PTRansition?		
:NTRansition	<numeric>	
:NTRansition?		
:SLOT1|:SLOT3|:SLOT4		6-102
[:EVENt]?		
:CONDition?		
:ENABle	<numeric>	
:ENABle?		
:PTRansition	<numeric>	
:PTRansition?		
:NTRansition	<numeric>	
:NTRansition?		
:SLOT3|:SLOT4		6-104
:G32P|:G32E		
[:EVENt]?		
:CONDition?		
:ENABle	<numeric>	
:ENABle?		
:PTRansition	<numeric>	
:PTRansition?		
:NTRansition	<numeric>	
:NTRansition?		

6.2. Before Reading Command Details

6.2.1.Important points when sending a command

To send an Instrument command, you should keep several points listed below in mind.

- Setting by slot

When sending a command on the Instrument, add the slot No. in which the unit to be set is inserted to the subsystem header. (This rule is not applied to the DISPlay, SYSTem and STATus subsystems.)

Example:

Slot No.	Inserted unit	Command to be sent
Slot1	3.2G Internal Synthesizer option	:OUTPut1:CLOCk:FREQuency 3200000 (Sets the frequency of the 3.2G internal synthesizer in Slot1.)
Slot3	3.2G PPG	:SOURce3:PATTern:TYPE PRBS7 (Sets the transmission pattern of 3.2G PPG in Slot3.)
Slot4	3.2G ED	:SENSe4:PATTern:TYPE PRBS7 (Sets the reception pattern of 3.2G ED in Slot4.)

- Operation when the PPG and ED units have the common setting

Suppose that the PPG unit is inserted in one slot and the ED unit is in other slot and the common setting of two units turns ON. When the setting of one unit turns ON, the same setting of the other unit also turns ON.
However, the other unit does not have the same setting, it is not influenced.

Example:
When 3.2G PPG is in Slot3 and 3.2G ED is in Slot4 and the common setting of the pattern of both units turns ON

Command	Setting
:SOURce3:PATTern:TYPE PRBS7 (Pattern mode setting: Setting common to PPG and ED)	The pattern of both PPG and ED units is set in PRBS7.
:SENSe4:PATTern:SYNC:ASYNc ON (Auto Sync setting: Setting of ED only)	This setting is only given to the ED unit and does not influence PPG.

- When different commands set the same parameter

The Instrument has a command for each unit to set a function, such as starting measurement, which is not supposed to set by unit or slot. The Instrument can set the same parameter using a command for any unit.

Example: When 3.2G PPG is in Slot3 and 3.2G ED in Slot4, turn ON the setting common to both units

Command for PPG in Slot3: :INSTrument3:COUPle:PATTern:SET ON
Command for ED in Slot4: :INSTrument4:COUPle:PATTern:SET ON
Either command will turn ON the common setting.

6.2.2. Buffer Size

The Instrument provides a 16-k byte output buffer, which enables setting/query of a large amount of data between the controller and the device. Thus, the controller should prepare a buffer large enough to receive the response correctly.

The buffer size required for each command is listed below. For details on the response size for each command, see the command detail explanation.

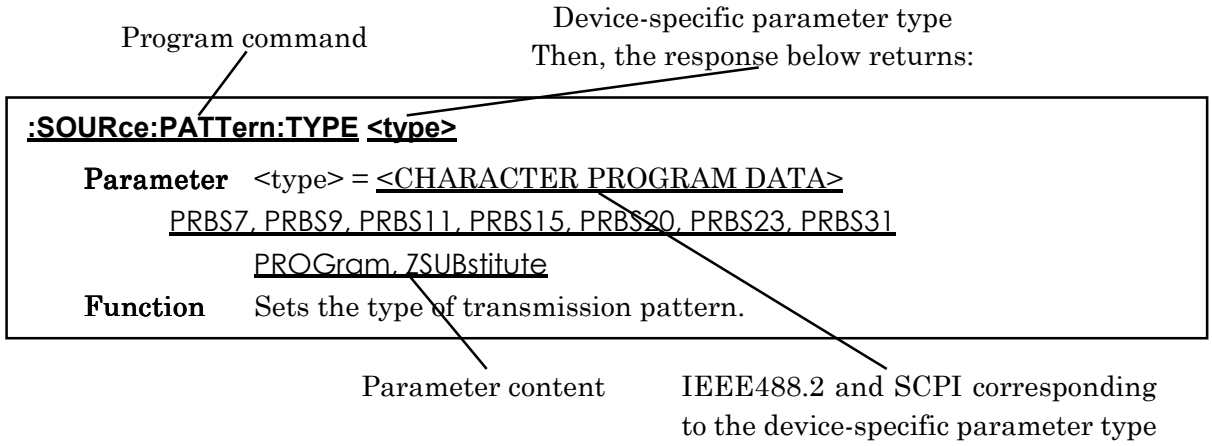
- When reading program data as string
  512 bytes max.
  (Command) :SOURce:PATTern:DATA:WHOLe
- When reading program data as binary data
  16,384 bytes max.
  (Command) :SOURce:PATTern:BDATa:WHOLe
- Other query commands
  256 bytes max.
  Although the required buffer size varies depending on the command, you can keep 256 bytes to read data correctly.

6.2.3. How to Read Details on Commands

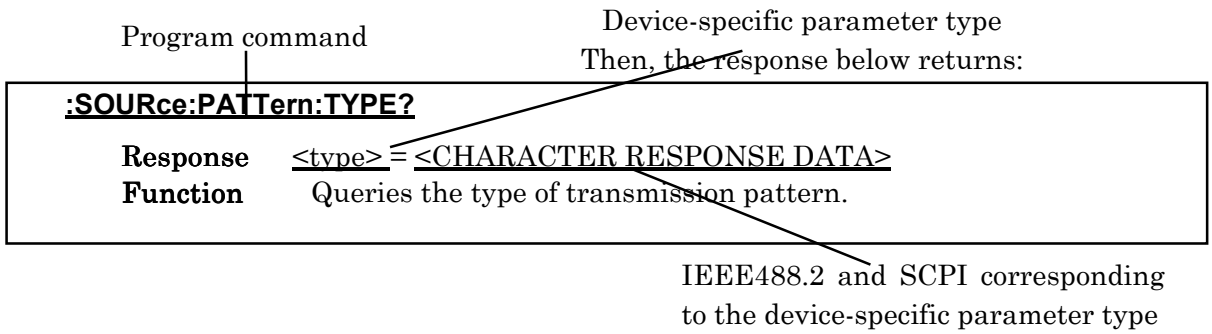
The following sections describe command details. This section highlights how to read the commands and considerations to be taken.

- Order of command description
  The command details are described in the order of the command tree given in the previous sections. For the correspondence between the command and the screen/function, see 8.2 "Command List".

- Program command example
  To send a command listed below, specify the command as:
  :SOURce3:PATTern:TYPE PRBS7



- Query command example
  When the corresponding program command exists, the response data is same as the short form of its parameter. To send the following command, specify the command as:
  :SOURce3:PATTern:TYPE?
  Response: PRBS7



Notes

1. All the Instrument commands are sequential commands.
2. If a command affects other settings, the command may have restriction. For setting parameters subject to be affected and command conditions to be restricted, see the MP1632C Operation Manual and operation manual of each unit.

6.3. Setting 3.2G PPG (MU163220C)

6.3.1. SOURce Subsystem

The SOURce Subsystem mainly performs setting/query of the following functions:

- Setting of transmission pattern
- Setting of error insertion

Notes

When 3.2G PPG is in Slot3, send the command in the form of SOURce3:...
Example: :SOURce3:PATTern:TYPE PRBS7

:SOURce:PATTern:OMODe <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
REPeat, BURSt

Function Sets the pattern output mode.

:SOURce:PATTern:OMODe?

Response <mode> = <CHARACTER RESPONSE DATA>
REP, BURS

Function Queries the pattern output mode.

:SOURce:PATTern:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31
PROGram, ZSUBstitute

Function Sets the type of transmission pattern.

:SOURce:PATTern:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>
PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31
PROG, ZSUB

Function Queries the pattern of transmission pattern.

:SOURce:PATTern:PRBS:MRATio <mratio>

Parameter <mratio> = <CHARACTER PROGRAM DATA>

M1_2	1/2
M1_4	1/4
M1_8	1/8
M0_8	0/8
I1_2	1/2 INVT
M3_4	3/4
M7_8	7/8
M8_8	8/8

Function Sets the mark rate when the transmission pattern is PRBS.

:SOURce:PATTern:PRBS:MRATio?

Response <mratio> = <CHARACTER RESPONSE DATA>
:SOURce:PATTern:PRBS:MRATio Same as the MRAtio command

Function Queries the mark rate when the transmission pattern is PRBS.

:SOURce:PATTern:PRBS:BSHift <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1	1bit Shift
3	3bit Shift

Function Sets the bit shift when the transmission pattern is PRBS.

:SOURce:PATTern:PRBS:BSHift?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift when the transmission pattern is PRBS.

:SOURce:PATTern:ZSUBstitute:LENGth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

7	2^7
9	2^9
11	2^11
15	2^15

Function Sets the pattern length when the transmission pattern is Zero-Subst.

:SOURce:PATTern:ZSUBstitute:LENgth?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the transmission pattern is Zero-Subst.

:SOURce:PATTern:ZSUBstitute:ZLENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

7 to 127	7 to 127 bits/Step:1	(Pattern Length:2^7)
9 to 511	9 to 511 bits/Step:1	(Pattern Length:2^9)
11 to 2047	11 to 2047 bits/Step:1	(Pattern Length2^11)
15 to 32767	15 to 32767 bits/Step:1	(Pattern Length:2^15)

Function Sets the zero bit insertion length when the transmission pattern is Zero-Subst.

:SOURce:PATTern:ZSUBstitute:ZLENgth?

Response <numeric> = <NR1 NUMERIC RESONSE DATA>

Function Queries the zero bit insertion length when the transmission pattern is Zero-Subst.

:SOURce:PATTern:ZSUBstitute:LOGic <logic>

Parameter <logic> = <CHARACTER PROGRAM DATA>
POSitive, NEGative

Function Sets the pattern logic when the transmission pattern is Zero-Subst.

:SOURce:PATTern:ZSUBstitute:LOGic?

Response <logic> = <CHARACTER RESPONSE DATA>
POS, NEG

Function Queries the pattern logic when the transmission pattern is Zero-Subst.

:SOURce:PATTern:PROGram:LENGth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2 to 8,388,608	2 to 131,072 bits/Step:1
	131,072 to 262,144 bits/Step:2
	262,144 to 524,288 bits/Step:4
	524,288 to 1,048,576 bits/Step:8
	1,048,576 to 2,097,152 bits/Step:16
	2,097,152 to 4,194,304 bits/Step:32
	4,194,304 to 8,388,608 bits/Step:64

Function Sets the pattern length when the transmission pattern is PRGM.

:SOURce:PATTern:PROGram:LENGth?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the transmission pattern is PRGM.

:SOURce:PATTern:PROGram:LOGic <logic>

Parameter <logic> = <CHARACTER PROGRAM DATA>
POSitive,NEGative

Function Sets the pattern logic when the transmission pattern is PRGM.

:SOURce:PATTern:PROGram:LOGic?

Response <logic> = <CHARACTER RESPONSE DATA>
POS,NEG

Function Queries the pattern logic when the transmission pattern is PRGM.

:SOURce:PATTern:BURSt:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
INTernal, EXTernal

Function Sets the Burst signal source.

:SOURce:PATTern:BURSt:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>
INT, EXT

Function Queries the Burst signal source.

:SOURce:PATTern:BURSt:CYCLe <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 50000 2 to 500000 us/Step:1

Function Sets Burst Cycle of the Burst signal.

:SOURce:PATTern:BURSt:CYCLe?

Response <numeric> = <NR1 NUMERIC PROGRAM DATA>

Function Queries Burst Cycle of the Burst signal.

:SOURce:PATTern:BURSt:ELENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 49999 1 to 49999 us/Step:1

Function Sets Enable Length of the Burst signal.

:SOURce:PATTern:BURSt:ELENgth?

Response <numeric> = <NR1 NUMERIC PROGRAM DATA>

Function Queries Enable Length of the Burst signal.

:SOURce:PATTern:DATA:WHOLe <start>,<end>,<data>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<data> = <STRING PROGRAM DATA>
"H***..." Specifies the pattern data in hexadecimal.
***...:1 to 400 characters (400 x 4 bits of data pattern)
Specifies the string composed of 0 to 9 and A to F.
"B***..." Specifies the pattern data in binary.
***... : 1 to 400 characters (400 bits of pattern data)
Specifies the string composed of 0 and 1.

Function Sets the program data from <start> to <end> when the transmission pattern is PRGM. Overwrites the data to the specified range.

When the bits of data specified in <data> is smaller than the range specified by <start> and <end>	Sets <data> repeatedly within the specified range. <Example> -- <start>=#H0, <end>=#H1F, <data>="HABC" Setting data: ABCABCAB -- <start>=#H0, <end>=#H7, <data>="B011" Setting data: 01101101
When the bits of data specified in <data> is larger than the range specified by <start> and <end>	Truncates <data> which exceeds from the specified range. <Example> -- <start>=#H0, <end>=#HF, <data>="HABCDEF" Setting data: ABCD -- <start>=#H0, <end>=#H3, <data>="B01100110"" Setting data: 0110

Application example

When setting 1F program pattern to 0 starting from Address (Hex)0

```
> :SOURce3:PATTern:DATA:WHOLe△#H0,#H1F,"H0"
```

:SOURce:PATTern:DATA:WHOLe? <start>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)

Response <data> = <STRING RESPONSE DATA>
"H***..." Returns the pattern data in hexadecimal.
***... :400 characters max. (400 x 4 bits of pattern data)

Function Queries the program data from <start> address when the transmission pattern is PRGM. 400 × 4 bits of data is returned from <start> address. If 400 × 4 bits of data is not available from <start> address, the data from <start> address to the end of address is returned.

:SOURce:PATTern:BDATa:WHOLe <start>,<end>,<bdata>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<bdata> = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<binary> X : Digits of YYY
YYY : Bytes of <binary>, 1 to 16,000 (bytes)
<binary> : Binary data up to 16,000 bytes

Function Sets the program data from <start> to <end> addresses when the transmission pattern is PRGM. Overwrites the data to the specified range.

Application example
When setting 1F program pattern to 41 from Address (Hex) 0

```
>:SOURce3:PATTern:BDATa:WHOLe△#H0,#H1F,#11A
```

:SOURce:PATTern:BDATa:WHOLe? <start>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)

Response <bdata> = <DEFINITE LENGTH ARBITARY BLOCK RESPONSE DATA>
#XYYY<binary> X : Digits of YYY
YYY : Bytes of <binary>, 1 to 16,000 (bytes)
<binary> : Binary data up to 16,000 bytes

Function Queries the program data from <start> address when the transmission pattern is PRGM. 16,000 bytes of data from <start> address is returned. If 16,000 bytes of data is not available from <start> address, the data from <start> address to the end of address is returned.

:SOURce:PATTern:DREVerse:ADDRess <start>,<end>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)

Function Reverses the program data of the transmission pattern. Specifies the range to be reversed using start/end addresses.

:SOURce:PATTern:DREVerse:DELTa <start>,<delta>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<delta> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8388608 1 to 8388608 bit/Step: 1

Function Reverses the program data of the transmission pattern. Specifies the range to be reversed using the relative bits from the start address.

:SOURce:PATTern:EADDition:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets ON/OFF of error addition.

:SOURce:PATTern:EADDition:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Error addition OFF
1 Error addition ON

Function Queries ON/OFF of error addition.

:SOURce:PATTern:EADDition:SINGle

Function Adds a single error.

:SOURce:PATTern:EADDition:RATE <rate>

Parameter <rate> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8
E_9	1E-9
SINGle	Single
EXT	External

Function Selects the error insertion method.

:SOURce:PATTern:EADDition:RATE?

Response <rate> = <CHARACTER RESPONSE DATA>
E_3, E_4, E_5, E_6, E_7, E_8, E_9, SING, EXT

Function Queries the error insertion method.

:SOURce:PATTern:EADDition:ROUTe <route>

Parameter <route> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8 route 1 to route 8/Step:1

Function Sets the route in which an error is inserted.

:SOURce:PATTern:EADDition:ROUTe? <route>

Parameter <route> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the route in which an error is inserted.

:SOURce:PATTern:LOGic:PRBS <mark>

Parameter <mark> = <CHARACTER PROGRAM DATA>

MHIGh	Sets the PRBS High level to 1.
MLOW	Sets the PRBS Low level to 1.

Function Sets whether the PRBS logical value 1 is High level or Low level.

:SOURce:PATTern:LOGic:PRBS?

Response <mark> = <CHARACTER RESPONSE DATA>

MHIG, MLOW

Function Queries whether the PRBS logical value 1 is High level or Low level.

:SOURce:PATTern:LOGic:PRGM <mark>

Parameter <mark> = <CHARACTER PROGRAM DATA>

MHIGh	Sets the PRGM High level to 1.
MLOW	Sets the PRGM Low level to 1.

Function Sets whether the PRBS logical value 1 is High level or Low level.

:SOURce:PATTern:LOGic:PRGM?

Response <mark> = <CHARACTER RESPONSE DATA>

MHIG, MLOW

Function Queries whether the PRGM logical value 1 is High level or Low level.

6.3.2.OUTPut Subsystem

The OUTPut subsystem performs setting/query of the following functions:

- Setting of the Clock output interface
- Setting of the Data output interface
- Setting of SYNC Output

Notes:

When 3.2G PPG is in Slot3, send a command in the form of :OUTPut3:...
Example: OUTPut3:CLOCk:OUTPut CLOCk,ON

:OUTPut:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1
Function Sets ON/OFF of the clock output and data output.

:OUTPut:SET?

Response <boolean> = <NR1 NUMERIC RESPONSE DATA>
0 Output OFF
1 Output ON
Function Queries ON/OFF of the clock output and data output.

:OUTPut:CLOCk:OUTPut <port>,<boolean>

Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk,XCLock
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1
Function Sets ON/OFF of the clock output at the specified port.

:OUTPut:CLOCk:OUTPut? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <boolean> = <NR1 NUMERIC RESPONSE DATA>
0 Output OFF
1 Output ON
Function Queries ON/OFF of the clock output at the specified port.

:OUTPut:CLOCk:DELay <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-1000 to 1000 1000 ps from -1000/Step:2ps
Function Sets the clock output delay.

:OUTPut:CLOCk:DELay?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Function Queries the clock output delay.

:OUTPut:CLOCk:LEVel <port>,<level>

Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk,XCLock
<level> = <CHARACTER PROGRAM DATA>
VAR,ECL
Function Sets the clock output level at the specified port.

:OUTPut:CLOCk:LEVel? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
VAR,ECL
Response <level> = <CHARACTER RESPONSE DATA>
Function Queries the clock output level at the specified port.

:OUTPut:CLOCk:AMPLitude <port>,<numeric>

Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk,XCLock
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.50 to 2.00 0.50 to 2.00 Vpp/Step:0.01
Function Sets the clock output amplitude at the specified port.

:OUTPut:CLOCk:AMPLitude? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries the clock output amplitude at the specified port.

:OUTPut:CLOCk:OFFSet <port>,<numeric>

Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk,XCLock
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-4.000 to 2.000 -4.000 to 2.000 V/Step:0.005
Function Sets the clock output offset value at the specified port.

:OUTPut:CLOCk:OFFSet? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries the clock output offset value at the specified port.

:OUTPut:CLOCk:OREFerence <port>,<offset>

Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk,XCLock
<offset> = <CHARACTER PROGRAM DATA>

VOH	High level
VTH	Center value between High level and Low level
VOL	Low level

Function Sets the clock output offset reference value at the specified port.

:OUTPut:CLOCk:OREFerence? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <offset> = <CHARACTER RESPONSE DATA>
VOH,VTH,VOL
Function Queries the clock output offset reference value at the specified port.

:OUTPut:CLOCk:DUTY <port>,<numeric>

Parameter <port> = <CHARACTER PROGRAM DATA>
CLOCk,XCLock
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>

-25 to 25	-25 to 25	-25 to 25/Step:1

Function Sets Duty of the clock output.

:OUTPut:CLOCk:DUTY? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Function Queries Duty of the clock output.

:OUTPut:CLOCk:IMPedance <impedance>

Parameter <impedance> = <CHARACTER PROGRAM DATA>

OHM0	0Ω
OHM5	50Ω

Function Sets DC impedance of the clock output.

:OUTPut:CLOCk:IMPedance?

Response <impedance> = <CHARACTER RESPONSE DATA>
OHM0, OHM5
Function Queries DC impedance of the clock output.

:OUTPut:DATA:OUTPut <port>,<boolean>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets ON/OFF of the data output at the specified port.

:OUTPut:DATA:OUTPut? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>

Response <boolean> = <BOOLEAN RESPONSE DATA>
0 Output OFF
1 Output ON

Function Queries ON/OFF of the data output at the specified port.

:OUTPut:DATA:IMPedance <impedance>

Parameter <impedance> = <CHARACTER PROGRAM DATA>
OHM0 0Ω
OHM5 50Ω

Function Sets DC impedance of the data output.

:OUTPut:DATA:IMPedance?

Response <impedance> = <CHARACTER RESPONCE DATA>
OHM0, OHM5

Function Queries DC impedance of the data output.

:OUTPut:DATA:CPOint <port>,<numeric>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
25 to 75 25 to 75%/Step:1

Function Sets cross point of the clock output.

:OUTPut:DATA:CPOint? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>

Response <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries cross point of the clock output.

:OUTPut:DATA:LEVel <port>,<level>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
<level> = <CHARACTER PROGRAM DATA>
VAR,ECL
Function Sets the data output level at the specified port.

:OUTPut:DATA:LEVel? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <level> = <CHARACTER RESPONSE DATA>
VAR,ECL
Function Queries the data output level at the specified port.

:OUTPut:DATA:AMPLitude <port>,<numeric>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.50 to 2.00 0.50 to 2.00 Vpp/Step:0.01
Function Sets the data output amplitude at the specified port.

:OUTPut:DATA:AMPLitude? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries the data output amplitude at the specified port.

:OUTPut:DATA:OFFSet <port>,<numeric>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-4.000 to 2.000 -4.000 to 2.000 V/Step:0.005
Function Sets the data output offset value at the specified port.

:OUTPut:DATA:OFFSet? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries the data output offset value at the specified port.

:OUTPut:DATA:OREFerence <port>,<offset>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
<offset> = <CHARACTER PROGRAM DATA>

VOH	High level
VTH	Center value between High level and Low level
VOL	Low level

Function Sets the data output offset reference value at the specified port.

:OUTPut:DATA:OREFerence? <port>

Parameter <port> = <CHARACTER PROGRAM DATA>
DATA,XDATa
Response <offset> = <CHARACTER RESPONSE DATA>
VOH, VTH, VOL
Function Queries the data output offset reference value at the specified port.

:OUTPut:SYNC:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>

CLOCk8	1/8 Clock Sync
PATTern	Pattern Sync

Function Selects the output signal of Trigger Output.

:OUTPut:SYNC:SOURce?

Response <source> = <CHARACTER RESPONSE DATA>
CLOCk8, PATTern
Function Queries the output signal of Trigger Output.

:OUTPut:SYNC:POSition <address>

Parameter <address> = <DECIMAL PROGRAM DATA>

1 to 121	1 to 121/Step:8	(Zero-Subst 2^7)
1 to 505	1 to 505/Step:8	(Zero-Subst 2^9)
1 to 2041	1 to 2041/Step:8	(Zero-Subst 2^11)
1 to 32761	1 to 32761/Step:8	(Zero-Subst 2^15)

1 to Pattern Length 1 to Pattern Length/Step:8 (PRGM)

The maximum setting data at Pattern = PRGM becomes the maximum value below

the Pattern Length value calculated as below:

Pattern Length * The maximum setting data value = 1 + 8 x n (n: integer)

Example: At Pattern Length = 127, the maximum value becomes 121.

1 to 511	1 to 511/Step:1	(PRBS9)
1 to 2047	1 to 2047/Step:1	(PRBS11)
1 to 32767	1 to 32767/Step:1	(PRBS15)
1 to 1048575	1 to 1048575/Step:1	(PRBS20)
1 to 8388607	1 to 8388607/Step:1	(PRBS23)
1 to 2147483647	1 to 2147483647/Step:1	(PRBS31)

Function Sets the synchronous output position when outputting Pattern Sync.

:OUTPut:SYNC:POSition?

Response <address> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the synchronous position of Pattern Sync output.

6.3.3. **CALCulate Subsystem**

The CALCulate subsystem queries the alarm occurrence condition (such as PPG Clock Loss)

Notes

When 3.2G PPG is in Slot3, send a command in the form of :CALCulate3:...
Example: :CALCulate3:DATA:MONitor?

:CALCulate:DATA:MONitor? <item>

Parameter <item> = <STRING PROGRAM DATA>
"PCL" PPG Clock Loss
"DTR3" Delay Trouble(Slot3)

Response <string> = <STRING RESPONSE DATA>

Type	Format	Description
Form1	"Occur"	When an alarm occurs
	(NL^END)	When no alarm occurs
	"-----"	When there is no data corresponding to query

Function Queries the occurrence condition of the monitoring item corresponding to the parameter.

6.3.4.**INSTrument Subsystem**

The INSTrument subsystem performs setting/query of tracking.

Notes

When 3.2G PPG is in Slot3, send a command in the form of "INSTrument3:...
Example: :INSTrument3:COUPle:CLOCk:TRACking ON

:INSTrument:COUPle:CLOCk:TRACking <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1
Function Selects ON/OFF of clock interface tracking.

:INSTrument:COUPle:CLOCk:TRACking?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Tracking OFF
1 Tracking ON
Function Queries ON/OFF of clock interface tracking.

:INSTrument:COUPle:DATA:TRACking <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1
Function Selects ON/OFF of data interface tracking.

:INSTrument:COUPle:DATA:TRACking?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Tracking OFF
1 Tracking ON
Function Queries ON/OFF of data interafce tracking.

:INSTrument:COUPle:PATTern:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1
Function Sets ON/OFF to common settings of Slot3 and Slot4.

:INSTrument:COUPle:PATTern:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Common setting OFF
1 Common setting ON
Function Queries ON/OFF of common settings of Slot3 and Slot4.

6.4. Setting 3.2G ED (MU163240C)

6.4.1. SENSe Subsystem

The SENSe subsystem performs setting/query of the following functions:

- Setting of reception pattern
- Setting of Error/Alarm measurement

Notes

When 3.2G ED is in Slot4, send a command in the form of :SENSe4:...
Example: :SENSe4:PATTern:TYPE PRBS7

:SENSe:PATTern:IMODe <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
REPeat,BURSt
Function Sets the pattern input mode.

:SENSe:PATTern:IMODe?

Response <mode> = <CHARACTER RESPONSE DATA>
REP, BURS
Function Queries the pattern input mode.

:SENSe:PATTern:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31
PROGram, ZSUBstitute
Function Sets the reception pattern type.

:SENSe:PATTern:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>
PRBS7, PRBS9, PRBS11, PRBS15, PRBS20, PRBS23, PRBS31
PROG, ZSUB
Function Queries the reception pattern type.

:SENSe:PATTern:PRBS:MRATio <mratio>

Parameter <mratio> = <CHARACTER PROGRAM DATA>

M1_2	1/2
M1_4	1/4
M1_8	1/8
M0_8	0/8
I1_2	1/2 INVT
M3_4	3/4
M7_8	7/8
M8_8	8/8

Function Sets the mark rate when the reception pattern is PRBS.

:SENSe:PATTern:PRBS:MRATio?

Response <mratio> = <CHARACTER RESPONSE DATA>
M1_2, M1_4, M1_8, M0_8, I1_2, M3_4, M7_8, M8_8

Function Queries the mark rate when the reception pattern is PRBS.

:SENSe:PATTern:PRBS:BSHift <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1	1bit Shift
3	3bit Shift

Function Sets the bit shift when the reception pattern is PRBS.

:SENSe:PATTern:PRBS:BSHift?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit shift when the reception pattern is PRBS.

:SENSe:PATTern:ZSUBstitute:LENGth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

7	2^7
9	2^9
11	2^11
15	2^15

Function Sets the pattern length when the reception pattern is Zero-Subst.

:SENSe:PATTern:ZSUBstitute:LENGth?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the reception pattern is Zero-Subst.

:SENSe:PATTern:ZSUBstitute:ZLENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

7 to 127	7 to 127 bits/Step:1	(Pattern Length: 2^9)
9 to 511	9 to 511 bits/Step:1	(Pattern Length: 2^7)
11 to 2047	11 to 2047 bits/Step:1	(Pattern Length: 2^11)
15 to 32767	15 to 32767 bits/Step:1	(Pattern Length: 2^15)

Function Sets the zero bit insertion length when the reception pattern is Zero-Subst.

:SENSe:PATTern:ZSUBstitute:ZLENgth?

Response <numeric> = <NR1 NUMERIC RESONSE DATA>

Function Queries the zero bit insertion length when the reception pattern is Zero-Subst.

:SENSe:PATTern:ZSUBstitute:LOGic <logic>

Parameter <logic> = <CHARACTER PROGRAM DATA>
POSitive, NEGative

Function Sets the pattern logic when the reception pattern is Zero-Subst.

:SENSe:PATTern:ZSUBstitute:LOGic?

Response <logic> = <CHARACTER RESPONSE DATA>
POS, NEG

Function Queries the pattern logic when the reception pattern is Zero-Subst.

:SENSe:PATTern:PROGram:LENGth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

2 to 8388608	2 to 131072 bits/Step:1
	131072 to 262144 bits/Step:2
	262144 to 524288 bits/Step:4
	524288 to 1048576 bits/Step:8
	1048576 to 2097153 bits/Step:16
	2097152 to 4194304 bits/Step:32
	4194304 to 8388608 bits/Step:64

Function Sets the pattern length when the reception pattern is PRGM.

:SENSe:PATTern:PROGram:LENGth?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the pattern length when the reception pattern is PRGM.

:SENSe:PATTern:PROGram:LOGic <logic>

Parameter <logic> = <CHARACTER PROGRAM DATA>
POSitive, NEGative

Function Sets the pattern logic when the reception pattern is PRGM.

:SENSe:PATTern:PROGram:LOGic?

Response <logic> = <CHARACTER RESPONSE DATA>
POS, NEG

Function Queries the pattern logic when the reception pattern is PRGM.

:SENSe:PATTern:SYNC:ASYNc <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets ON/OFF of Auto Sync.

:SENSe:PATTern:SYNC:ASYNc?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Auto Sync OFF
1 Auto Sync ON

Function Queries ON/OFF of Auto Sync.

:SENSe:PATTern:SYNC:THReshold:LOSS <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>

E_2	1E-2
E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7

Function Sets the detection threshold of Pattern Sync Loss.

:SENSe:PATTern:SYNC:THReshold:LOSS?

Response <thre> = <CHARACTER RESPONSE DATA>
E_2, E_3, E_4, E_5, E_6, E_7

Function Queries the detection threshold of Pattern Sync Loss.

:SENSe:PATTern:SYNC:THReshold:GAIN <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8

Function Sets the detection threshold of Pattern Sync Gain.

:SENSe:PATTern:SYNC:THReshold:GAIN?

Parameter <thre> = <CHARACTER RESPONSE DATA>

E_3, E_4, E_5, E_6, E_7, E_8

Function Queries the detection threshold of Pattern Sync Gain.

:SENSe:PATTern:SYNC:THReshold:INTernal <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

0 or OFF,1 or ON

Function Sets the Sync threshold to the device value.

:SENSe:PATTern:SYNC:THReshold:INTernal?

Parameter <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Device value
1	Loss/Gain detection threshold

Function Queries whether the Sync threshold is set to device value.

:SENSe:PATTern:SYNC:PSMode <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
NORMal,FRAMe

Function Sets the pattern Sync method of PRGM.

:SENSe:PATTern:SYNC:PSMode?

Response <mode> = <CHARACTER RESPONSE DATA>
NORM, FRAM

Function Queries the pattern Sync method of PRGM.

:SENSe:PATTern:SYNC:FLENgth <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

8	8 bits
16	16 bits
24	24 bits
32	32 bits

Function Sets the frame length when Frame is selected for the pattern Sync method.

:SENSe:PATTern:SYNC:FLENgth?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the frame length when Frame is selected for the pattern Sync method.

:SENSe:PATTern:DATA:WHOLe <start>,<end>,<data>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<data> = <STRING PROGRAM DATA>
"H***..." Specifies the pattern data in hexadecimal.
***...: 1 to 400 characters (400 x 4 bits of pattern data)
Specifies the string composed of 0 to 9 and A to F.
"B***..." Specifies the pattern data in binary.
***... : 1 to 400 characters (400 bits of pattern data)
Specifies the string composed of 0 and 1.

Function Sets the program data from <start> to <end> addresses when the reception pattern is PRGM. Overwrites the data to the specified range.

When the bits of data specified in <data> is smaller than the range specified by <start> and <end>	Sets <data> repeatedly within the specified range. <Example> -- <start> = #H0, <end> = #H1F, <data> = "HABC" Setting data: ABCABCAB -- <start> = #H0, <end> = #H7, <data> = "B011" Setting data: 01101101
When the bits of data specified in <data> is larger than the range specified by <start> and <end>	Truncates <data> which exceeds from the specified range. <Example> -- <start> = #H0, <end> - #HF, <data> = "HABCDEF" Setting data: ABCD -- <start> = #H0, <end> = #H3, <data> = "B01100110"" Setting data: 0110

Application example

When setting the program pattern from address (hex) 0 to 1F to 0

```
>SENSe3:PATTern:DATA:CHANnel△#H0,#H1F,"H0"
```

:SENSe:PATTern:DATA:WHOLe? <start>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bits (in hexadecimal)
Response <data> = <STRING RESPONSE DATA>
"H***..." Returns the pattern data in hexadecimal.
***... : 400 characters max. (400 x 4 bits of pattern data)

Function Queries the program data from <start> address when the reception pattern is PRGM. 400 x 4 bits of data is returned from <start> address. If 400 x 4 bits of data is not available from <start> address, the data from <start> address to the end of address is returned.

:SENSe:PATTern:BDATa:WHOLe <start>,<end>,<data>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<bdata> = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<binary> X : Digits of YYY
YYY : Bytes of <binary>, 1 to 16,000 (bytes)
<binary> : Binary data up to 16,000 bytes

Function Sets the program data from <start> to <end> addresses when the reception pattern is PRGM. Overwrites the data to the specified range.

Application example
When setting 1F program pattern to 41 from Address (Hex) 0
`>SENSe4:PATTern:BDATa:WHOLe△#H0,#H1F,#11A`

:SENSe:PATTern:BDATa:WHOLe? <start>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
Response <bdata> = <DEFINITE LENGTH ARBITARY BLOCK RESPONSE DATA>
Function Queries the program data from <start> address when the reception pattern is PRGM. 16,000 bytes of data from <start> address is returned. If 16,000 bytes of data is not available from <start> address, the data from <start> address to the end of address is returned.

:SENSe:PATTern:DREVerse:ADDRess <start>,<end>

Parameter <start> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<end> = <NON-DECIMAL PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
Function Reverses the program data of the reception pattern. Specifies the range to be reversed using start/end addresses.

:SENSe:PATTern:DREVerse:DELTa <start>,<delta>

Parameter <start> = <NON-DECIMAL NUMERIC PROGRAM DATA>
#H0 to #H7FFFFF 0 to 7FFFFF bit/Step:1 (specifies in hexadecimal)
<delta> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 8388608 0 to 8388608 bit/Step:1
Function Reverses the program data of the transmission pattern. Specifies the range to be reversed using the relative bits from the start address.

:SENSe:PATTern:LOGic:PRBS <mark>

Parameter <mark> = <CHARACTER PROGRAM DATA>

MHIGh	Sets the PRBS High level to 1.
MLOW	Sets the PRBS Low level to 1.

Function Sets whether the PRBS logical value 1 is High level or Low level.

:SENSe:PATTern:LOGic:PRBS?

Response <mark> = <CHARACTER RESPONSE DATA>

MHIG, MLOW

Function Queries whether the PRBS logical value 1 is High level or Low level.

:SENSe:PATTern:LOGic:PRGM <mark>

Parameter <mark> = <CHARACTER PROGRAM DATA>

MHIGh	Sets the PRGM High level to 1.
MLOW	Sets the PRGM Low level to 1.

Function Sets whether the PRBS logical value 1 is High level or Low level.

:SENSe:PATTern:LOGic:PRGM?

Response <mark> = <CHARACTER RESPONSE DATA>

MHIG, MLOW

Function Queries whether the PRBS logical value 1 is High level or Low level.

:SENSe:MEASure:TEST <item>

Parameter <item> = <CHARACTER PROGRAM DATA>

EALarm	Error/Alarm
EMARgin	Eye Margin

Function Sets the measurement parameter.

:SENSe:MEASure:TEST?

Response <item> = <CHARACTER RESPONSE DATA>

EAL, EMAR

Function Queries the measurement parameter.

:SENSe:MEASure:STARt

Function Starts measurement.

:SENSe:MEASure:STOP

Function Stops measurement.

:SENSe:MEASure:MREStart <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Measurement restart OFF
ON or 1	Measurement restart ON

Function Sets ON/OFF of measurement restart when changing threshold or delay.

:SENSe:MEASure:MREStart?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Measurement restart OFF
1	Measurement restart ON

Function Queries ON/OFF of measurement restart when changing threshold or delay.

:SENSe:MEASure:EALarm:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

REPeat,SINGle,UNTimed

Function Sets the measurement mode for Error/Alarm measurement.

:SENSe:MEASure:EALarm:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

REP, SING, UNT

Function Queries the measurement mode for Error/Alarm measurement.

:SENSe:MEASure:EALarm:PERiod <day>,<hour>,<min>,<sec>

Parameter <day>,<hour>,<min>,<sec> = <DECIMAL NUMERIC PROGRAM DATA>

<day>	0 to 99	0 to 99 days/Step:1
<hour>	0 to 23	0 to 23 hours/Step:1
<min>	0 to 59	0 to 59 minutes/Step:1
<sec>	0 to 59	0 to 59 seconds/Step:1

Function Sets the measurement time.

:SENSe:MEASure:EALarm:PERiod?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement time.

:SENSe:MEASure:EALarm:BTIMe:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Selects ON/OFF of the function to set the measurement start time.

:SENSe:MEASure:EALarm:BTIMe:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Measurement start time setting OFF
1 Measurement start time setting ON

Function Queries ON/OFF of the function to set the measurement start time.

:SENSe:MEASure:EALarm:BTIMe:STARt <year>,<month>,<day>,<hour>,<min>,<sec>

Parameter <year>,<month>,<day>,<hour>,<min>,<sec> =
<DECIMAL NUMERIC PROGRAM DATA>

Function Sets the measurement start date and time .

:SENSe:MEASure:EALarm:BTIMe:STARt?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement start date and time.

:SENSe:MEASure:EALarm:ERRor:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
TOTal Total
IOMission INS/OMI

Function Sets the type of error to be measured.

:SENSe:MEASure:EALarm:ERRor:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>
TOT, IOM

Function Queries the type of error to be measured.

:SENSe:MEASure:EALarm:MASK:ROUTe <route>,<boolean>

Parameter <route> = <CHARACTER PROGRAM DATA>

R1	route1
R2	route2
R3	route3
R4	route4
R5	route5
R6	route6
R7	route7
R8	route8
ALL	All routes

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not mask measurement.
ON or 1	Masks measurement.

Function Sets ON/OFF of 8 internal routes mask for measurement.

:SENSe:MEASure:EALarm:MASK:ROUTe?

Response <route> = <CHARACTER RESPONSE DATA>

R#,R#,..	Separates each R numbers to be error-masked with "," and returns them. #: 1 to 8
NONE	When all routes are not masked

Function Queries the route to be masked for measurement.

:SENSe:MEASure:EALarm:STARt?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement start time for Error/Alarm measurement. When no measurement start time data is found, 0,0,0,0,0,0 is returned.

:SENSe:MEASure:EALarm:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement stop time for Error/Alarm measurement. When no measurement end time is found, 0,0,0,0,0,0 is returned.

:SENSe:MEASure:EALarm:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Measurement stops.
1	Measurement goes on.

Function Queries progress status of Error/Alarm measurement.

:SENSe:MEASure:EALarm:ELAPsed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement passing time for Error/Alarm measurement.

:SENSe:MEASure:EALarm:TIMed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the residual measurement time for Error/Alarm measurement.

:SENSe:MEASure:EALarm:ITIMe?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries time when the intermediate data for Error/Alarm measurement was created. If no intermediate data was found, 0,0,0,0,0,0 is returned.

:SENSe:MEASure:EALarm:AOCCur? <alarm>

Parameter <alarm> = <CHARACTER PROGRAM DATA>

PFA	Power Fail
CLOS	Clock Loss
PSL	Pattern Sync Loss

Response <year>,<month>,<day>,<day>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

0,0,0,0,0,0	When no alarm occurs

Function Queries the time when the specified alarm occurred.

:SENSe:MEASure:EALarm:ARECover? <alarm>

Parameter <alarm> = <CHARACTER PROGRAM DATA>

PFA	Power Fail
CLOS	Clock Loss
PSL	Pattern Sync Loss

Response <year>,<mon>,<day>,<day>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

0,0,0,0,0,0	When Alarm was not recovered (generated).

Function Queries the time when the specified alarm was recovered.

:SENSe:MEASure:EMARgin:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

MARGin	Set measurement mode to Eye Margin.
DIAGram	Set measurement mode to Eye Diagram.

Function Sets the Eye Margin measurement mode.

:SENSe:MEASure:EMARgin:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>
MARG, DIAG

Function Queries the Eye Margin measurement mode.

:SENSe:MEASure:EMARgin:TYPE <type>

Parameter <type> = <CHARACTER PROGRAM DATA>

PTHReshold	Phase & Threshold
PHASe	Phase
THReshold	Threshold
POINt8	8points
POINt16	16points
POINt32	32points
POINt64	64points

Function Sets the Eye Margin measurement parameter for Eye Margin measurement or the measurement point at the same error rate for Eye Diagram measurement.

:SENSe:MEASure:EMARgin:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>
PTHR, PHAS, THR, POIN8, POIN16, POIN32, POIN64

Function Queries the Eye Margin measurement parameter or the measurement point at the same error rate.

:SENSe:MEASure:EMARgin:MARGin:THReshold <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8
E_9	1E-9
E_10	1E-10
E_11	1E-11
E_12	1E-12

Function Sets the threshold when the measurement mode is Margin.

:SENSe:MEASure:EMARgin:MARGin:THReshold?

Response <thre> = <CHARACTER REPONSE DATA>
E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12

Function Queries the threshold when the measurement mode is Margin.

:SENSe:MEASure:EMARgin:MARGin:RESolution <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
COARse,FINE

Function Sets the error rate resolution when the measurement mode is Margin.

:SENSe:MEASure:EMARgin:MARGin:RESolution?

Response <type> = <CHARACTER RESPONSE DATA>
COAR, FINE

Function Queries the error rate resolution when the measurement mode is Margin.

:SENSe:MEASure:EMARgin:DIAGram:THReshold <thre>,<boolean>

Parameter <thre> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8
E_9	1E-9
E_10	1E-10
E_11	1E-11
E_12	1E-12

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0,ON or 1

Function Sets the threshold when the measurement mode is Diagram.

:SENSe:MEASure:EMARgin:DIAGram:THReshold?

Response <thre> = <CHARACTER RESPONSE DATA>

E_#,E_#,.. Separates the selected error rates with "," and returns them.
#: 3 to 12

Function Queries the threshold when the measurement mode is Diagram.

:SENSe:MEASure:EMARgin:DIAGram:POINt? <thre>

Parameter <thre> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8
E_9	1E-9
E_10	1E-10
E_11	1E-11
E_12	1E-12

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

X,Y,Z,...	Returns the valid point. From X, Y, Z, ...: 1 to 64
0	When <thre> is invalid or when no valid point is found

Function Queries the measurement valid point at each threshold when the measurement mode is Diagram.

:SENSe:MEASure:EMARgin:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Measurement stops.
1	Measurement goes on.
-1	Measurement failed.

Function Queries the progress status of Eye Margin measurement.

:SENSe:MEASure:EMARgin:STARt?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement start time for Eye Margin measurement. When no measurement start time data is found, 0,0,0,0,0,0 is returned.

:SENSe:MEASure:EMARgin:STOP?

Response <year>,<month>,<day>,<hour>,<min>,<sec> =
<NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement stop time for Eye Margin. When no measurement end time is found, 0,0,0,0,0,0 is returned.

:SENSe:MEASure:EMARgin:ELAPsed?

Response <day>,<hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the measurement passing time for Eye Margin measurement.

:SENSe:MEASure:ASEarch:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

PTHReshold	Phase & Threshold
THReshold	Threshold
PHASe	Phase
OFF	OFF

Function Sets the operation mode of Auto Search.

:SENSe:MEASure:ASEarch:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>
PTH, THR, PHAS, OFF

Function Queries the operation mode of Auto Search.

:SENSe:MEASure:ASEarch:STARt

Function Starts Auto Search.

:SENSe:MEASure:ASEarch:STOP

Function Force to stop Auto Search.

:SENSe:MEASure:ASEarch:PATTern <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets ON/OFF of PRBS Pattern Search.

:SENSe:MEASure:ASEarch:PATTern?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	PRBS Pattern Search OFF
1	PRBS Pattern Search ON

Function Queries ON/OFF of PRBS Pattern Search.

:SENSe:MEASure:ASEarch:STATe?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Auto Search stops.
1	Auto Search goes on.
-1	Auto Search failed.

Function Queries the progress status of Auto Search.

6.4.2. INPut Subsystem

The INPut subsystem performs setting/query of the following functions:

- Setting of the Clock output interface
- Setting of the Data output interface
- Setting of the reference clock

Notes

When 3.2G ED is in Slot4, send a command in the form of :INPut4:...
Example: INPut4:CLOCk:POLarity POSitive

:INPut:CLOCk:POLarity <pol>

Parameter <pol> = <CHARACTER PROGRAM DATA>
POSive,NEGative
Function Sets the clock input polarity.

:INPut:CLOCk:POLarity?

Response <pol> = <CHARACTER RESPONSE DATA>
POS, NEG
Function Queries the clock input polarity.

:INPut:CLOCk:DELay <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-1000 to 1000 -1000 to 1000 ps/Step:2
Function Sets the clock input delay.

:INPut:CLOCk:DELay?

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Function Queries the clock input delay.

:INPut:CLOCk:TERMination <term>

Parameter <term> = <CHARACTER PROGRAM DATA>

GND	GND
V_2	-2V
V3	+3V

Function Sets the clock input termination condition.

:INPut:CLOCk:TERMination?

Response <term> = <CHARACTER RESPONSE DATA>
GND, V_2, V3
Function Queries the clock input termination condition.

:INPut:DATA:LEVel <level>

Parameter <level> = <CHARACTER PROGRAM DATA>
VAR, ECL, PECL, TTL
Function Sets the data input offset level.

:INPut:DATA:LEVel?

Response <level> = <CHARACTER RESPONSE DATA>
VAR, ECL, PECL, TTL
Function Queries the data input offset level.

:INPut:DATA:THReshold <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-4.000 to 4.000 -4.000 to 4.000 V/Step:0.001
Function Sets the data input threshold level.

:INPut:DATA:THReshold?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Function Queries the data input threshold level.

:INPut:DATA:TERMination <term>

Parameter <term> = <CHARACTER PROGRAM DATA>

GND	GND
V_2	-2V
V3	+3V

Function Sets the data input termination condition.

:INPut:DATA:TERMination?

Response <term> = <CHARACTER RESPONSE DATA>
GND, V_2, V3
Function Queries the data input termination condition.

6.4.3. CALCulate Subsystem

The CALCulate subsystem performs setting/query of the following functions:

- Setting of Error/Alarm measurement condition
- Reading of measurement results

Notes

When 3.2G ED is in Slot3, send a command in the form of :CALCulate3:... When 3.2G ED is in Slot4, send a command in the form of :CALCUlate4:...
Example: :CALCulate3:EALaram:CLEValuation ON

:CALCulate:EALarm:CLEValuation <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not include in measurement to be evaluated.
ON or 1	Includes in measurement to be evaluated.

Function Sets whether the Clock Loss time is included in measurement to be evaluated (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:CLEValuation?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Does not include in measurement to be evaluated.
1	Includes in measurement to be evaluated.

Function Queries whether the Clock Loss time is included in measurement to be evaluated (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:SLEValuation <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not include in measurement to be evaluated.
ON or 1	Includes in measurement to be evaluated.

Function Sets whether the Sync Loss time is included in measurement to be evaluated (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:SLEValuation?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Does not include in measurement to be evaluated.
1	Includes in measurement to be evaluated.

Function Queries whether the Sync Loss time is included in measurement to be evaluated (Performance,EI/%EFI,Threshold EI/%EFI).

:CALCulate:EALarm:ERRor:INTerval <numeric>,<suffix>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 100
<suffix> = <CHARACTER PROGRAM DATA>
ms,s

Function Sets interval of EI and %EFI measurement. Interval can be set only using <100,ms> and <1,s>.

:CALCulate:EALarm:ERRor:INTerval?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
<suffix> = <CHARACTER RESPONSE DATA>
1,s or 100,ms

Function Queries interval of EI and %EFI measurement.

:CALCulate:EALarm:PERFomance:MEASurement <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0, ON or 1

Function Sets ON/OFF of the error performance measurement.

:CALCulate:EALarm:PERFomance:MEASurement?

Parameter <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Error performance measurement OFF
1	Error performance measurement ON

Function Queries ON/OFF of the error performance measurement.

:CALCulate:EALarm:PERFomance:THReshold <thre>

Parameter <thre> = <STRING PROGRAM DATA>

"SES_3:DM_6"	SES:1E-3/DM:1E-6
"SES_4:DM_8"	SES:1E-4/DM:1E-8

Function Sets the performance measurement threshold for SES and DM calculation.

:CALCulate:EALarm:PERFomance:THReshold?

Response <thre> = <STRING RESPONSE DATA>
"SES_3:DM_6", "SES_4:DM_8"

Function Queries the performance measurement threshold for SES and DM calculation.

:CALCulate:EALarm:TEINterval:MEASurement <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets ON/OFF of the Threshold EI/%EFI measurement.

:CALCulate:EALarm:TEINterval:MEASurement?

Response <numeric> = <NR1 NUMERIC RESPNSE DATA>
0 Threshold EI/%EFI measurement OFF
1 Threshold EI/%EFI measurement ON

Function Queries ON/OFF of the Threshold EI/%EFI measurement.

:CALCulate:EALarm:PFAil <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets ON/OFF of the Power Fail measurement.

:CALCulate:EALarm:PFAil?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Power Fail measurement OFF
1 Power Fail measurement ON

Function Queries ON/OFF of the Power Fail measurement.

:CALCulate:DATA:EALarm? <string>

Parameter <string> = <STRING PROGRAM DATA>

"[CURRent:]<result>"	Current data
"LAST:<result>"	Measurement termination data of the previous measurement
"INTermediate:<result>"	Measurement intermediate data

For description of <result>, see Table 6-1.

Response <string> = <STRING RESPONSE DATA>

Type	Format	Description
Form1 Integer type	" XXXXXXX"	0 to 9999999
	"X.XXXXEXX"	1.0000E07 to 9.9999E16
	"---------"	No data is found for query.
Form2 decimal fraction type	"X.XXXXE-XX"	0.0000E-16 to 1.0000E00
	"----------"	No data is found for query.
Form3 % type	"XXX.XXXX"	0.0000 to 100.0000
	"--------"	No data is found for query.
Form4 Frequency type	"XXXXXXXXXX"	0 to MAX (Hz)
	"----------"	No data is found for query.

Function Queries the measurement data corresponding to the parameter. The measurement termination data and measurement intermediate data can be stored in the buffer using the :CALCulate:DATA:STORe command. After the data is stored in the buffer, the buffered data is used for response to query for the measurement termination data and measurement intermediate data.

Application examples

1) When inquiring the current data of Error Rate (INS)

```
>:CALCulate4:DATA:EALarm?△"CURRent:ER:INS"
```

or `> :FALCulate4:DATA:EALarm?△"ER:INSertion"`

(CURRent: can be omitted.)

Response: "1.0000E-02"

2) When inquiring the measurement completion data of Clock Loss

```
> :CALCulate4:DATA:EALarm?△"LAST:AINTerval:CLOSs"
```

Response: " 123"

3) When inquiring the measurement intermediate data at power failure

```
>:CALCulate4:DATA:EALaram?△"INTermediate:AINTerval:POWer"
```

Response: " 22"

Table 6-2 Error/Alarm Measurement Parameters <result> (1/3)

Parameter		<result>	Response format
Error Rate	INS OMI Total	"ER:INSertion" "ER:OMISsion" "ER:TOTal"	Form2
Error Count	INS OMI Total	"EC:INSertion" "EC:OMISsion" "EC:TOTal"	Form1
Clock Count	Total	"CC:TOTal"	Form1
EI	Total	"EI:TOTal"	Form1
%EFI	Total	"EFI:TOTal"	Form3
Average Error Rate per second	INS OMI Total	"AVERage:ER:INSertion" "AVERage:ER:OMISsion" "AVERage:ER:TOTal"	Form2
Error Count per second	INS OMI Total	"AVERage:EC:INSertion" "AVERage:EC:OMISsion" "AVERage:EC:TOTal"	Form1
Clock Count per second	Total	"AVERage:CC:TOTal"	Form1
Frequency		"FREQuency"	Form4
Power Fail		"AINTerval:POWer"	Form1
Clock Loss		"AINTerval:CLOSs"	
Pattern Sync Loss		"AINTerval:PSLoss"	

Table 6-2 Error/Alarm Measurement Parameters <result> (2/3)

Parameter			<result>	Response format
G.821	%ES	INS	"G821:ES:INSertion"	Form3
		OMI	"G821:ES:OMISsion"	
		Total	"G821:ES:TOTal"	
	%EFS	INS	"G821:EFS:INSertion"	
		OMI	"G821:EFS:OMISsion"	
		Total	"G821:EFS:TOTal"	
	%SES	INS	"G821:SES:INSertion"	
		OMI	"G821:SES:OMISsion"	
		Total	"G821:SES:TOTal"	
	%US	INS	"G821:US:INSertion"	
		OMI	"G821:US:OMISsion"	
		Total	"G821:US:TOTal"	
	%DM	INS	"G821:DM:INSertion"	
		OMI	"G821:DM:OMISsion"	
		Total	"G821:DM:TOTal"	
	ES	INS	"G821:ES2:INSertion"	Form1
		OMI	"G821:ES2:OMISsion"	
		Total	"G821:ES2:TOTal"	
	EFS	INS	"G821:EFS2:INSertion"	
		OMI	"G821:EFS2:OMISsion"	
		Total	"G821:EFS2:TOTal"	
	SES	INS	"G821:SES2:INSertion"	
		OMI	"G821:SES2:OMISsion"	
		Total	"G821:SES2:TOTal"	
	US	INS	"G821:US2:INSertion"	
		OMI	"G821:US2:OMISsion"	
		Total	"G821:US2:TOTal"	
	DM	INS	"G821:DM2:INSertion"	
		OMI	"G821:DM2:OMISsion"	
		Total	"G821:DM2:TOTal"	
	EC	INS	"G821:EC2:INSertion"	
		OMI	"G821:EC2:OMISsion"	
		Total	"G821:EC2:TOTal"	

Table 6-2 Error/Alarm Measurement Parameters <result> (3/3)

Parameter			<result>	Response format
Threshold EI	INS	>1E-3,..,	"THReshold:EI:INSertion:E_3",..,	Form1
		=<1E-8	"THReshold:EI:INSertion:UE_8"	
	OMI	>1E-3,..,	"THReshold:EI:OMISsion:E_3",..,	
		=<1E-8	"THReshold:EI:OMISsion:UE_8"	
	Total	>1E-3,..,	"THReshold:EI:TOTal:E_3",..,	
		=<1E-8	"THReshold:EI:TOTal:UE_8"	
Threshold %EFI	INS	>1E-3,..,	"THReshold:EFI:INSertion:E_3",..,	Form3
		=<1E-8	"THReshold:EFI:INSertion:UE_8"	
	OMI	>1E-3,..,	"THReshold:EFI:OMISsion:E_3",..,	
		=<1E-8	"THReshold:EFI:OMISsion:UE_8"	
	Total	>1E-3,..,	"THReshold:EFI:TOTal:E_3",..,	
		=<1E-8	"THReshold:EFI:TOTal:UE_8"	

:CALCulate:DATA:EMARgin? <string>

Parameter <string> = <STRING PROGRAM DATA>
"<result>" Measurement data
For description of <result>, see Table 6-2.

Response <string> = <STRING RESPONSE DATA>

Type	Format	Description
Form1 Phase type	" XXXX"	0 to MAX (ps)
	"Failed"	Measurement is failed.
	"------"	No data is found for query.
Form2 Threshold type	" XXXX"	0 to MAX (mVpp)
	"Failed"	Measurement is failed.
	"------"	No data is found for query.
Form3 Eye Map type	"$XXXX:$Y.YYY"	$: Sign XXXX : Phase(ps) Y.YYY : Threshold Voltage(V)

Function Queries the Eye Margin measurement data corresponding to the parameter.

Application examples

1) When inquiring the phase margin measurement data

```
> :CALCulate4:DATA:EMARgin?△"PHASe"
```

Response:" 452"

2) When inquiring 1E-4 Eye Map value at point 1

```
> :CALCulate4:DATA:EMARgin?△"MAP:E_4:POINt1"
```

Response:" -323:-0.200"

Table 6-3 Eye Margin Measurement Parameters <result>

Parameter		<result>	Response format
Phase margin		"PHASe"	Form1
Threshold Margin		"THReshold"	Form2
Eye Map Current meas-urement result	1E-3 1E-4 : 1E-11 1E-12	"MAP:E_3:POINt1",..,"MAP:E_3:POINt64" "MAP:E_4:POINt1",..,"MAP:E_4:POINt64" : "MAP:E_11:POINt1",..,"MAP:E_11:POINt64" "MAP:E_12:POINt1",..,"MAP:E_12:POINt64" Returns the value at each point at each error rate.	Form3

:CALCulate:DATA:STORe <string>

Parameter <string> = <STRING PROGRAM DATA>

"EALarm:LAST"	Measurement termination data
"EALarm:INTermediate"	Measurement intermediate data

Function Stores the specified measurement data in the buffer for measurement result. When inquiring the above measurement result during execution of repeat measurement, the data may change halfway because the next measurement starts before the Instrument finishes reading all the data. This command can be used to store the data to be read in the buffer and read the measurement result from the buffered data. The data to be buffered is listed below:

Parameter	Data to be buffered
"EALarm:LAST"	Measurement termination data displayed on the screen when the command is received
"EALarm:INTermediate"	Measurement intermediate data created before the command is received

Application example

When storing the Error/Alarm measurement termination data in the buffer and then inquiring the termination data of ER(INS)

```
>:CALCulate4:DATA:STORe "EALarm:LAST"
>:CALCulate4:DATA:EALarm? "LAST:ER:INS"
```

Notes

1. The buffered data is retained until it is cleared or the power switch is turned OFF.
2. Once the data is buffered, the buffered data is used for the response to query. If the data is not buffered, the data displayed on the screen is returned. To clear the buffer, use the :CALCulate:DATA:CLEar command.
3. You cannot buffer the Error/Alarm current data. For query for the current data, the current data being displayed on the screen is returned.

:CALCulate:DATA:CLEar <string>

Parameter <string> = <STRING PROGRAM DATA>

"EALarm:LAST"	Error/AlarmError/Alarm measurement termination data
"EALarm:INTermediate"	Error/AlarmError/Alarm measurement intermediate data

Function Clears the specified measurement data stored in the buffer for measurement result.

:CALCulate:DATA:MONitor? <item>

Parameter <item> = <STRING PROGRAM DATA>

"BIT:TOTal"	Bit Error(Total Error)
"BIT:IOMission:INSertion"	Bit Error(INS/OMI Error(INS))
"BIT:IOMission:OMISsion"	Bit Error(INS/OMI Error(OMI))
"CLOSs "	Clock Loss
"PSLoss"	Pattern Sync Loss
"POWer"	Power Fail
"DTR4"	Delay Trouble(Slot4)

Response <string> = <STRING RESPONSE DATA>

Type	Format	Description
Form1	"Occur"	Alarm occurs.
	(NL^END)	Alarm does not occur.
	"-----"	No data is found for query.

Function Queries the occurrence condition of the monitor item corresponding to the parameter.

6.4.4. OUTPut Subsystem

The OUTPut subsystem performs setting/query of SYNC Output.

Notes:

When 3.2G ED is in Slot4, send a command in the form of :OUTPut4:...
Example: :OUTPut4:SYNC:SOURce CLOCk8

:OUTPut:SYNC:SOURce <source>

Parameter <source> = <CHARACTER PROGRAM DATA>

CLOCk8	1/8 Clock Sync
PATTern	Pattern Sync
SGLoss	SYNC Gain/Loss

Function Sets the output signal of Trigger Output.

:OUTPut:SYNC:SOURce?

Response <source> = <CHARACTER RESPONSE DATA>
CLOC8, PATT, SGL

Function Queries the output signal of Trigger Output.

6.4.5. INSTrument Subsystem

The INSTrument subsystem performs common setting/query.

Notes:

When 3.2G ED is in Slot4, send a command in the form of :INSTrument4:...
Example: :INSTrument4:COUPle:PATTern:SET ON

:INSTrument:COUPle:PATTern:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Slot3-Slot4 common setting OFF
ON or 1	Slot3-Slot4 common setting ON

Function Sets ON/OFF of common pattern setting for two units.

:INSTrument:COUPle:PATTern:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Common setting OFF
1	Common setting ON

Function Queries ON/OFF of common pattern setting for two units.

6.5. Setting 3.2G Internal Synthesizer option(OPT03)

6.5.1. OUTPut Subsystem

The OUTPut Subsystem performs setting/query of the following functions:

Note

So that 3.2G Internal Synthesizer option is in Slot1, send a command in the form of :OUTPut1:...
Example: :OUTPut1:CLOCk:FREQuency 3200000

:OUTPut:CLOCk:FREQuency <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
50000 to 3200000 50000 to 3200000 kHz/Step:1

Function Set the clock frequency.

:OUTPut:CLOCk:FREQuency?

Response <numeric> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the clock frequency.

:OUTPut:RCLock:SELect <clock>

Parameter <clock> = <CHARACTER PROGRAM DATA>

INTernal	Internal signal
EXT10	EXT 10MHz

Function Set reference signal(EXT 10MHz or Internal signal) sychronization with output clock signal.

:OUTPut:RCLock:SELect?

Response <clock> = <CHARACTER RESPONSE DATA>
INT, EXT10

Function Inquire reference signal(EXT 10MHz or Internal signal) sychronization with output clock signal.

6.5.2. CALCulate Subsystem

The CALCulate Subsystem performs setting/query of the following functions:

Note

So that 3.2G Internal Synthesizer option is in Slot1, send a command in the form of :OUTPut1:...
Examples: :CALCulate1:DATA:MONitor?

:CALCulate:DATA:MONitor? <result>

Parameter <result> = <STRING PROGRAM DATA>
"PLL" PLL Unlock

Response <string> = <STRING RESPONSE DATA>

Type	Format	Description
Form1	"Occur"	Alarm occurs.
	(NL^END)	Alarm does not occur.
	"-----"	No data is found for query.

Function Queries the occurrence condition of the monitor item corresponding to the parameter.

Example > :CALCulate1:DATA:MONitor?△"PLL"
Response:"Occur"

6.6. Other settings

6.6.1. DISPlay Subsystem

The DISPlay subsystem performas setting/query of window operations.

:DISPlay:WINDow:OPEN <disp>

Parameter <disp> = <CHARACTER PROGRAM DATA>

SYSTem	System subwindow
SETup	Setup subwindow
TMENu	Test Menu subwindow
RESult	Result subwindow
CUSTomize	Customize subwindow

Function Open specified subwindow.

:DISPlay:WINDow:OPEN?

Response <disp> = <CHARACTER RESPONSE DATA>

XXX,XXX,,,	Returns open subwindow names, separated by a comma(,).
NONE	No subwindow is opened.

Function Queries the opened subwindow names.

:DISPlay:WINDow:CLOSe <disp>

Parameter <disp> = <CHARACTER PROGRAM DATA>

SYSTem	System subwindow
SETup	Setup subwindow
TMENu	Test Menu subwindow
RESult	Result subwindow
CUSTomize	Customize subwindow

Function Close a specified subwindow.

:DISPlay:SYSTem[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>
SYSTem,OPTion

Function Switches the panel in the System subwindow.

:DISPlay:SYSTem[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>
SYST, OPT

Function Queries the opened panel in System subwindow.

:DISPlay:SETup[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>
SETup,FREQuency,CLOCk,DATA,PATTern,TRIGger,UTILity
Function Switches the panel in the Setup subwindow.

:DISPlay:SETup[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>
SET, FREQ, CLOC, DATA, PATT, TRIG, UTIL
Function Queries the panel in the Setup subwindow.

:DISPlay:TEST[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>

MEASurement	Measurement panel
EADDition	Error Addition panel

Function Switches panel in the Test Menu subwindow.

:DISPlay:TEST[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>
MEAS, EADD
Function Queries the panel in the Test Menu subwindow.

:DISPlay:RESult:TIME <time>

Parameter <time> = <CHARACTER PROGRAM DATA>

ELAPsed	Elapsed display
TIMed	Timed display
DTIMe	Date & Time display
STARt	Start display

Function Set measurement time display mode of Result subwindow.

:DISPlay:RESult:TIME?

Response <time> = <CHARACTER RESPONSE DATA>
Function Queries measurement time display mode of Result subwindow.
ELAP, TIM, DTIM, STAR

:DISPlay:RESult:EALarm[:NAME] <name>

Parameter <name> = <CHARACTER PROGRAM DATA>
ALL,ZOOM,MONitor
Function Switches the Result subwindow (Error/Alarm) panel.

:DISPlay:RESult:EALarm[:NAME]?

Response <name> = <CHARACTER RESPONSE DATA>
ALL,ZOOM,MON
Function Queries the Result subwindow (Error/Alarm) panel.

:DISPlay:RESult:EALarm:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>
CURRent, LAST

Function Sets the Result subwindow (Error/Alarm) display mode.

:DISPlay:RESult:EALarm:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>
CURR, LAST

Function Queries the Result subwindow (Error/Alarm) display mode.

:DISPlay:RESult:EALarm:ALL:PTYPe <type>

Parameter <type> = <CHARACTER PROGRAM DATA>
COUNt Count Item (ES,EFS,SES,US,DM,EC)
PERCent Percent Item (%ES,%EFS,%SES,%US,%DM,EC)

Function Set performance measurement display items of Result:ALL(Error/Alarm) panel.

:DISPlay:RESult:EALarm:ALL:PTYPE?

Response <type> = <CHARACTER RESPONSE DATA>
COUN, PERC

Function Queries performance measurement display items of Result:ALL(Error/Alarm) panel.

:DISPlay:RESult:EALarm:ZOOM1:SET <boolean>
:DISPlay:RESult:EALarm:ZOOM2:SET <boolean>
:DISPlay:RESult:EALarm:ZOOM3:SET <boolean>
:DISPlay:RESult:EALarm:ZOOM4:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets the display ON/OFF status of the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:SET?
:DISPlay:RESult:EALarm:ZOOM2:SET?
:DISPlay:RESult:EALarm:ZOOM3:SET?
:DISPlay:RESult:EALarm:ZOOM4:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Display OFF
1	Display ON

Function Queries the display ON/OFF statusof the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:ITEM <item>

:DISPlay:RESult:EALarm:ZOOM2:ITEM <item>

:DISPlay:RESult:EALarm:ZOOM3:ITEM <item>

:DISPlay:RESult:EALarm:ZOOM4:ITEM <item>

Parameter <item> = <CHARACTER PROGRAM DATA>

ER	Error Rate
EC	Error Count
EI	EI
EFI	%EFI
PSLoss	Pattern Sync Loss
CLOSs	Clock Loss
PFAil	Power Fail
TEI	Threshold EI
TEFI	Threshold %EFI
PERFormance	Performance
FREQuency	Frequency

Function Sets the item to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:ITEM?

:DISPlay:RESult:EALarm:ZOOM2:ITEM?

:DISPlay:RESult:EALarm:ZOOM3:ITEM?

:DISPlay:RESult:EALarm:ZOOM4:ITEM?

Response <item> = <CHARACTER PROGRAM DATA>

ER, EC, EI, EFI, PSL, CLOS, PFA, TEI, TEFI, PERF, FREQ

Function Queries the item to be displayed on the Result:Zoom panel.

:DISPlay:RESult:EALarm:ZOOM1:PITem <pitem>

:DISPlay:RESult:EALarm:ZOOM2:PITem <pitem>

:DISPlay:RESult:EALarm:ZOOM3:PITem <pitem>

:DISPlay:RESult:EALarm:ZOOM4:PITem <pitem>

Parameter <pitem> = <CHARACTER PROGRAM DATA>

ES2	ES
EFS2	EFS
SES2	SES
DM2	DM
US2	US
EC	EC
ES	%ES
EFS	%EFS
SES	%SES
DM	%DM
US	%US

Function Sets the performance measurement item to be displayed when Performance has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:ZOOM1:PITem?

:DISPlay:RESult:EALarm:ZOOM2:PITem?

:DISPlay:RESult:EALarm:ZOOM3:PITem?

:DISPlay:RESult:EALarm:ZOOM4:PITem?

Response <pitem> = <CHARACTER RESPONSE DATA>

ES2, EFS2, SES2, DM2, US2, EC, ES, EFS, SES, DM, US

Function Queriesthe performance measurement item to be displayed when Performance has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:ZOOM1:TITem <titem>

:DISPlay:RESult:EALarm:ZOOM2:TITem <titem>

:DISPlay:RESult:EALarm:ZOOM3:TITem <titem>

:DISPlay:RESult:EALarm:ZOOM4:TITem <titem>

Parameter <titem> = <CHARACTER PROGRAM DATA>

E_3	>1.0E-3	|	E_7	>1.0E-7
E_4	>1.0E-4	|	E_8	>1.0E-8
E_5	>1.0E-5	| UE_8	=<1.0E-8	
E_6	>1.0E-6			

Function Sets the performance measurement item to be displayed when Threshold EI or Threshold %EFI has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:ZOOM1:TITem?

:DISPlay:RESult:EALarm:ZOOM2:TITem?

:DISPlay:RESult:EALarm:ZOOM3:TITem?

:DISPlay:RESult:EALarm:ZOOM4:TITem?

Response <titem> = <CHARACTER RESPONSE DATA>

E_3, E_4, E_5, E_6, E_7, E_8, UE_8

Function Queries the performance measurement item to be displayed when Threshold EI or Threshold %EFI has been selected as the Result:Zoom panel display item.

:DISPlay:RESult:EALarm:HRESet

Function Resets the history lamp on Result:Monitor panel.

:DISPlay:RESult:EMARgin:ERATe <erate>,<boolean>

Parameter <erate> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8
E_9	1E-9
E_10	1E-10
E_11	1E-11
E_12	1E-12

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Sets the display ON/OFF status of each error rate for Eye Diagram result

:DISPlay:RESult:EMARgin:ERATe?

Response <erate> = <CHARACTER RESPONSE DATA>

E_X,E_X,..	The opened subwindows name(s), which is separated with a comma.
NONE	No diagram is displayed.

Function Queries the display ON/OFF status of each error rate for Eye Diagram result

:DISPlay:RESult:EMARgin:SCALe:ASCale

Function Adjusts the scale automatically so that the entire measurement result is included in the Result subwindow.

:DISPlay:RESult:EMARgin:SCALe:VOLTage <min>,<step>

Parameter <min> = <DECIMAL NUMERIC PROGRAM DATA>
-4.000 to 3.990 -4.000 to 3.990 V/Step:0.001
<step> = <DECIMAL NUMERIC PROGRAM DATA>
0.001 to 1.000 0.001 to 1.000 V/Step:0.001

Function Set the minimum value of threshold scale on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:SCALe:VOLTage?

Response <min>,<step> = <NR2 NUMERIC RESPONSE DATA>

Function Queries the minimum value of threshold scale on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:SCALe:PHASe <min>,<step>

Parameter <min> = <DECIMAL NUMERIC PROGRAM DATA>
-1000 to 980 -1000 to 980 ns/Step:2
<step> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 200 2 to200 ns/Step:2

Function Sets the minimum value of phase scale on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:SCALe:PHASe?

Response <min>,<step> = <NR2 NUMERIC RESPONSE DATA>

Function Queries Sets the minimum value of phase scale on the Result subwindow (Ewhen measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:MARKer:SET <marker>,<boolean>

Parameter <marker> = <CHARACTER PROGRAM DATA>
BASE Base marker
REFerence Reference marker
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets display ON/OFF status of the specified marker on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:MARKer:SET? <marker>

Parameter <marker> = <CHARACTER PROGRAM DATA>

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 or 1

Function Queries the display ON/OFF status of the specified marker on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:MARKer:MODE <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

FREE	Moves the marker within the display area irrespective of the measurement result.
FIX	Moves the marker along plotted measurement result.

Function Sets a marker movement mode on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:MARKer:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

FREE, FIX

Function Queries a marker movement mode on the Result subwindow (when measurement parameter is Eye Diagram).

:DISPlay:RESult:EMARgin:MARKer:ERATe <erate>

Parameter <erate> = <CHARACTER PROGRAM DATA>

E_3	1E-3
E_4	1E-4
E_5	1E-5
E_6	1E-6
E_7	1E-7
E_8	1E-8
E_9	1E-9
E_10	1E-10
E_11	1E-11
E_12	1E-12

Function Sets the error rate of diagram on which the marker(s) can move.

:DISPlay:RESult:EMARgin:MARKer:ERATe?

Response <erate> = <CHARACTER RESPONSE DATA>

E_3, E_4, E_5, E_6, E_7, E_8, E_9, E_10, E_11, E_12

Function Queries the error rate of diagram on which the marker(s) can move.

:DISPlay:RESult:EMARgin:MARKer:MOVE <marker>,<mode>

Parameter <marker> = <CHARACTER PROGRAM DATA>

BASE	Base marker
REFerence	Reference marker

<mode> = <CHARACTER PROGRAM DATA>

LEFT	Move the marker left (FREE mode) or counterclockwise along the measurement points (FIX mode)
RIGHt	Move the marker right (FREE mode) or clockwise along the measurement points (FIX mode)
UP	Move the marker up (FREE mode) or clockwise along the measurement points (FIX mode).
DOWN	Move the marker down (FREE mode) or counterclockwise along the measurement points (FIX mode)

Function Move the specified marker.

:DISPlay:RESult:EMARgin:MARKer:POSition? <marker>

Parameter <marker> = <CHARACTER PROGRAM DATA>

BASE	Base marker
REFerence	Reference marker

Response <phase>,<voltage>

<phase> = <NR2 NUMERIC RESPONSE DATA>

-X.XXX to X.XXX	Phase value

<voltage> = <NR2 NUMERIC RESPONSE DATA>

-X.XXX to X.XXX	Threshold voltage value

Function Queries marker position. In case base marker is select absolute position is returned, in another case relactive position of the reference marker to the base maeker is returned.

:DISPlay:RESult:EMARgin:TEMPlate:SELect <select>,<boolean>

Parameter <select> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 4	Template1 to Template4/Step: 1

<boolean>=<BOOLEAN PROGRAM DATA>

OFF or 0, ON or 1

Function Selects the template to be displayed in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:SELect ?

Response <select> = <NR1 NUMERIC RESPONSE DATA>

#,#,..	Returns the selected templates separated by a comma ",".
0	All the templates are set to OFF.

Function Queries the template to be displayed in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:MOVE <template>,<mode>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<mode> = <CHARACTER PROGRAM DATA>
LEFT Moves to the left
RIGHt Moves to the right
UP Moves upward
DOWN Moves downward

Function Moves the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:PNUMber <template>,<point>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<point> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 32 2 to 32/Step: 1

Function Sets the number of points in the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:PNUMber? <template>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1

Response <point> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the number of points for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:VOLTage <template>,<voltage>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<voltage> = <DECIMAL NUMERIC PROGRAM DATA>
–4.000 to 4.000 –4.000 to 4.000 V/Step: 0.001

Function Sets the offset voltage value for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:VOLTage? <template>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1

Response <voltage> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the offset voltage value for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:PHASe <template>,<phase>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<phase> = <DECIMAL NUMERIC PROGRAM DATA>
–1000 to 1000 –1000 to 1000 ns/Step: 2

Function Sets the offset phase value for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:PHASe? <template>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1

Response <phase> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the offset phase value for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage <template>,<point>,<voltage>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 32 Point1 to Point32/Step: 1
<voltage> = <DECIMAL NUMERIC PROGRAM DATA>
–8.000 to 8.000 –8.000 to 8.000 V/Step: 0.001

Function Sets the voltage value at the specified point for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage? <template>,<point>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 32 Point1 to Point32/Step: 1

Response <voltage> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the voltage value at the specified point for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe <template>,<point>,<phase>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 32 Point1 to Point32/Step: 1
<phase> = <DECIMAL NUMERIC PROGRAM DATA>
–2000 to 2000 –2000 to 2000 ns/Step: 2

Function Sets the phase value at the specified point for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe? <template>,<point>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 32 Point1 to Point32/Step: 1

Response <phase> = <DECIMAL NUMERIC PROGRAM DATA>

Function Queries the phase value at the specified point for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:CONNect <template>,<boolean>

Function Selects the line connection method for the specified template in the Result sub-window.

:DISPlay:RESult:EMARgin:TEMPlate:SELect? <template>

Parameter <template> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 4 Template1 to Template4/Step: 1
<boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0 Does not connect the end and start points.
ON or 1 Connects the end and start points.

Function Queries the line connection method for the specified template in the Result sub-window.

:DISPlay:CUSTomize:SETup1:UNIT <unit>
:DISPlay:CUSTomize:SETup2:UNIT <unit>
:DISPlay:CUSTomize:SETup3:UNIT <unit>
:DISPlay:CUSTomize:SETup4:UNIT <unit>
:DISPlay:CUSTomize:SETup5:UNIT <unit>
:DISPlay:CUSTomize:SETup6:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>
0 Let no unit is not assigned.
1 Unit inserted in slot1
3 Unit inserted in slot3
4 Unit inserted in slot4

Function Sets the units to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:UNIT?

:DISPlay:CUSTomize:SETup2:UNIT?

:DISPlay:CUSTomize:SETup3:UNIT?

:DISPlay:CUSTomize:SETup4:UNIT?

:DISPlay:CUSTomize:SETup5:UNIT?

:DISPlay:CUSTomize:SETup6:UNIT?

Response <unit> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the units to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:ITEM <string>

:DISPlay:CUSTomize:SETup2:ITEM <string>

:DISPlay:CUSTomize:SETup3:ITEM <string>

:DISPlay:CUSTomize:SETup4:ITEM <string>

:DISPlay:CUSTomize:SETup5:ITEM <string>

:DISPlay:CUSTomize:SETup6:ITEM <string>

Parameter <string> = <STRING PROGRAM DATA>

"FREQuency:FREQuency"	Frequency/Frequency
"CLOCk:AMPLitude"	Clock/Amplitude
"CLOCk:OFFSet"	Clock/Offset
"CLOCk:DELay"	Clock/Delay
"CLOCk:POLarity"	Clock/Polarity
"CLOCk:DUTY"	Clock/Duty
"DATA:AMPLitude"	Data/Amplitude
"DATA:OFFSet"	Data/Offset
"DATA:CPOint"	Data/Cross Point
"DATA:THReshold"	Data/Threshold
"PATTern:LOGic"	Pattern/Logic
"PATTern:MRATio"	Pattern/Mark Ratio

Function Sets the items to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:ITEM?

:DISPlay:CUSTomize:SETup2:ITEM?

:DISPlay:CUSTomize:SETup3:ITEM?

:DISPlay:CUSTomize:SETup4:ITEM?

:DISPlay:CUSTomize:SETup5:ITEM?

:DISPlay:CUSTomize:SETup6:ITEM?

Response <string> = <STRING RESPONSE DATA>
"FREQ:FREQ", "CLOC:AMPL", "CLOC:OFFS", "CLOC:DEL", "CLOC:POL"
"CLOC:DUTY", "DATA:AMPL", "DATA:OFFS", "DATA:CPO", "DATA:THR"
"PATT:LOG", "PATT:MRAT"

Function Queries the items to be displayed in the Customize subwindow Setup 1 to 6 group boxes.

:DISPlay:CUSTomize:SETup1:PORT <port>

:DISPlay:CUSTomize:SETup2:PORT <port>

:DISPlay:CUSTomize:SETup3:PORT <port>

:DISPlay:CUSTomize:SETup4:PORT <port>

:DISPlay:CUSTomize:SETup5:PORT <port>

:DISPlay:CUSTomize:SETup6:PORT <port>

Parameter <port> = <STRING PROGRAM DATA>
CLOck, XCLOck, DATa, XDATa

Function Sets the port to be displayed in the Customize subwindow Setup 1 to 6 group boxes when PPG data/clock output item has been selected as the correspond panel.

:DISPlay:CUSTomize:SETup1:PORT?

:DISPlay:CUSTomize:SETup2:PORT?

:DISPlay:CUSTomize:SETup3:PORT?

:DISPlay:CUSTomize:SETup4:PORT?

:DISPlay:CUSTomize:SETup5:PORT?

:DISPlay:CUSTomize:SETup6:PORT?

Response <port> = <STRING RESPONSE DATA>
CLO, XCLO, DAT, XDAT

Function Queries which port is assigned to be displayed on Setup1 to Setup6 group boxes on the Customize subwindow.

:DISPlay:CUSTomize:PATTern:UNIT <unit>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

0	Let no unit is not assigned.
3	Unit inserted in Slot3
4	Unit inserted in Slot4

Function Sets the unit to be displayed in the Customize subwindow Pattern group box.

:DISPlay:CUSTomize:PATTern:UNIT?

Response <unit> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the unit to be displayed in the Customize subwindow Pattern group box.

:DISPlay:CUSTomize:PATTern:OFFSet <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0 to MAX	0 to MAX bits/Step:8 MAX is depend on pattern length of unit which is assigned on Pattern group box.

Function Specifies the bit position of the pattern to be edited as a byte boundary.

:DISPlay:CUSTomize:PATTern:OFFSet?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the bit position of the pattern to be edited on the Customize subwindow.

:DISPlay:CUSTomize:RESult:ITEM <item>

Parameter <item> = <CHARACTER PROGRAM DATA>

ER	Error Rate
EC	Error Count

Function Sets the item to be displayed in the Customize subwindow Result group box.

:DISPlay:CUSTomize:RESult:ITEM?

Response <item> = <CHARACTER RESPONSE DATA>

EC, ER

Function Queries the item to be displayed in the Customize subwindow Result group box.

6.6.2. SYSTem Subsystem

The SYSTem subsystem sets and queries the beeper, date, and other settings.

:SYSTem:BEEPer:ERRor:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets the ON/OFF status of the beeper on occurrence of errors.

:SYSTem:BEEPer:ERRor:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	OFF
1	ON

Function Queries the ON/OFF status of the beeper on occurrence of errors.

:SYSTem:BEEPer:ALARm:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets the ON/OFF status of the alarm beeper on occurrence of alarms.

:SYSTem:BEEPer:ALARm:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	OFF
1	ON

Function Queries the ON/OFF status of the alarm beeper on occurrence of alarms.

:SYSTem:BEEPer:SYSTem:SET <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0,ON or 1

Function Sets the ON/OFF status of the system beeper on occurrence of alarms.

:SYSTem:BEEPer:SYSTem:SET?

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

0	OFF
1	ON

Function Queries the ON/OFF status of the system beeper on occurrence of alarms.

:SYSTem:BEEPer:SYSTem:TYPE <type>,<boolean>

Parameter <type> = <CHARACTER PROGRAM DATA>

PUNLock	PLL Unlock(Slot1)
PCLoss	PPG Clock Loss
DTRouble3	Delay Trouble(Slot3)
DTRouble4	Delay Trouble(Slot4)

<boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0,ON or 1

Function Sets the ON/OFF status of items that are signaled by the system error beeper.

:SYSTem:BEEPer:SYSTem:TYPE?

Response <type> = <CHARACTER RESPONSE DATA>

XXX,XXX,..	Returns the errors having the system error beeper set ON, separated by a comma (,).
NONE	No errors have the system error beeper set ON.

Function Queries the ON/OFF status of items that are signaled by the system error beeper.

:SYSTem:MODE?

Response <mode> = <CHARACTER RESPONSE DATA>

PFA	Power Fail Measurement Mode
NORM	Normal Mode

Function Queries the mode of system operation.

:SYSTem:DATE <year>,<month>,<day>

Parameter <year>,<month>,<day> = <DECIMAL NUMERIC PROGRAM DATA>

<year>	1996 to 2037/Step:1
<month>	1 to 12/Step:1
<day>	1 to 31/Step:1

Function Sets the current date.

:SYSTem:DATE?

Response <year>,<month>,<day> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the current date.

:SYSTem:TIME <hour>,<min>,<sec>

Parameter <hour>,<min>,<sec> = <DECIMAL NUMERIC PROGRAM DATA>

<hour>	0 to 23/Step:1
<min>	0 to 59/Step:1
<sec>	0 to 59/Step:1

Function Sets the current time.

:SYSTem:TIME?

Response <hour>,<min>,<sec> = <NR1 NUMERIC RESPONSE DATA>

Function Queries the current time.

:SYSTem:BSIZe? <mode>

Parameter <mode> = <CHARACTER PROGRAM DATA>

INPut	Input buffer size
OUTPut	Output buffer size

Response <size> = <NR1 NUMERIC RESPONSE DATA>
Returns the I/O buffer size in bytes.

Function Queries the size of the instrument's I/O buffers.

:SYSTem:ERRor?

Response <error/event_number>,<error/event_description>
<error/event_number> = <NR1 NUMERIC RESPONSE DATA>
-32768 to 32767
<error/event_description> = <STRING RESPONSE DATA>

Function Queries the error status. See Appendix A for a detailed definition of errors.

:SYSTem:VERSion?

Response <version> = <NR2 NUMERIC RESPONSE DATA>

YYYY.V	YYYY	:year
	V	:revision number

Function Queries the SCPI version to which this instrument conforms.

:SYSTem:ORGanization:HARDware?

Response <slot1>,<slot2>,<slot3>,<slot4>
<slotX> = <unit_name>,<version>,<option>
<unit_name> = <STRING RESPONSE DATA> Unit name

"3.2G Clock"	3.2G Clock Generator
"3.2G PPG"	3.2G PPG Unit
"3.2G ED"	3.2G ED Unit
"---"	No unit is inserted in the slot

<version> = <NR1 NUMERIC RESPONSE DATA>

1 to XX	Unit version
0	No unit is inserted in the slot

<option> = <STRING RESPONSE DATA>

"OPTXX:OPTXX.."	Returns unit options, separated by a comma (,). OPTXX = OPT01, OPT02,.. (XX: Two digits)
"NONE"	No option is installed.
"---"	No unit is inserted in the slot (This response is returned unconditionally for Slot1)

Function Queries the instrument's system configuration. Use comman commands to query the mainframe version and options.

:SYSTem:ORGanization:SOFTware?

Response <NR2 NUMERIC RESPONSE DATA>
X.X Version X.X

Function Queries the instrument's software system version.

:SYSTem:MMEMory:RECall <type>,<file_name>,<unit>

Parameter <type> = <STRING PROGRAM DATA>
"MAIN" Main Frame Setup
"SLOT1" Slot1 Setup
"PPG" PPG Setup
"ED" ED Setup
"PATTern" Pattern Data
"SLOT2" Slot2 Setup
<file_name> = <STRING PROGRAM DATA>
"<drv>:/[<dir>]<file>"
<drv> = A or C
<dir> = <dir1>/<dir2>/... (Omitted for a root directory)
<file> = a filename
<unit> = <DECIMAL NUMERIC PROGRAM DATA>
If the type of file to read is PPG/ED Setup or Pattern Data, specify the unit to read to.
3 Unit inserted in slot3
4 Unit inserted in slot4
0 Any other type of file

Function Reads a specified file to a specified unit.

:SYSTem:MMEMory:RECall "EYE:TEMPlate1",<file_name>,0

Template1: "EYE:TEMPlate1"
Template2: "EYE:TEMPlate2"
Template3: "EYE:TEMPlate3"
Template4: "EYE:TEMPlate4"

<file_name> = <STRING PROGRAM DATA>
"<drv>:/[<dir>]<file>"
<drv> = A, C, ... (existing drives)
<dir> = <dir1>/<dir2>/... (omitted for the root directory)
<file> = File name (extensions are not required)

Function Calls the specified template in the Result sub-window.

:SYSTem:MMEMory:STORe <type>,<file_name>,<unit>

Parameter <type> = <STRING PROGRAM DATA>

"MAIN"	Main Frame Setup
"SLOT1"	Slot1 Setup
"PPG"	PPG Setup
"ED"	ED Setup
"PATTern"	Pattern Data
"SLOT2"	Slot2 Setup

<file_name> = <STRING PROGRAM DATA>

"<drv>:/[<dir>]<file>"

<drv> = A or C

<dir> = <dir1>/<dir2>/...(Omitted for a root directory)

<file> = a filename

<unit> = <DECIMAL NUMERIC PROGRAM DATA>

Specify the unit (slot) to save.

3	Unit inserted in slot3
4	Unit inserted in slot4
0	Source unit setting is not required.

Function Saves specified data from a specified unit to a file.

:SYSTem:MMEMory:STORe "EYE:TEMPlate1",<file_name>,0

Template1: "EYE:TEMPlate1"

Template2: "EYE:TEMPlate2"

Template3: "EYE:TEMPlate3"

Template4: "EYE:TEMPlate4"

<file_name> = <STRING PROGRAM DATA>

"<drv>:/[<dir>]<file>"

<drv> = A, C, ... (existing drives)

<dir> = <dir1>/<dir2>/... (omitted for the root directory)

<file> = File name (extensions are not required)

Function Saves the specified template in the Result sub-window.

:SYSTem:MMEMory:QRECall <file_name>

Parameter <file_name> = <STRING PROGRAM DATA>
"<drv>:/[<dir>]<file>"
<drv>= A,C
<dir> = <dir1>/<dir2>/...(Omitted for a root directory)
<file> = a filename (use the extension ".ENV")

Function Quick-opens

:SYSTem:MMEMory:QSTore <file_name>,<comment>

Parameter <file_name> = <STRING PROGRAM DATA>
"<drv>:/[<dir>]<file>"
<drv> = A, C
<dir> = <dir1>/<dir2>/...(Omitted for a root directory)
<file> = a filename
<comment> = <STRING PROGRAM DATA>
"XXXX..." Up to 60 characters as a file comment.

Function Quick-saves.

:SYSTem:MMEMory:CATalog? <drv_dir>,<type>

Parameter <drv_dir> = <STRING PROGRAM DATA>
"<drv>:/[<dir>]"
<drv> = A, C
<dir> = <dir1>/<dir2>/...(Omitted for a root directory)
<type> = <STRING PROGRAM DATA>

"MAIN"	Main Frame Setup
"SLOT1"	Slot1 Setup
"PPG"	PPG Setup
"ED"	ED Setup
"PATTern"	Pattern Data
"SLOT2"	Slot2 Setup
"ALL"	All data

Response <use_byte>,<free_byte>,<current_dir>,<current_file>,<file_entry>
<use_byte> = <NT1 NUMERIC RESPONSE DATA>
Space used (bytes)
<free_byte> = <NR1 NUMERIC RESPONSE DATA>
Available space (bytes)
<current_dir> = <STRING RESPONSE DATA>
Current directory
<current_file> = <NR1 NUMERIC RESPONSE DATA>
Number of files that match the <type> specification existing in the current directory, including subdirectories if <"ALL"> has been specified as <type>.
<file_entry> = <STRING RESPONSE DATA>
Returns, in the following format, files or directories that match <type>, or "" if no files match.
"<file_name>,<file_size>,<date_time>,..."

<file_name>	Filename or subdirectory name XXXX.XXX:Filename <XXXX>:Directory name
<file_size>	File size. XXX:File size(bytes)
<date_time>	Date of file creation. <year>.<month>.<day>△<hour>:<min>:<sec>

Function Queries HDD or FDD file information. 0, 0, "...", 0, "" is returned when disk information cannot be read.

:SYSTem:MEMory:INITialize

Function Resets the instrument settings to their factory defaults.

:SYSTem:PRINt:COPY

Function Copies the contents currentry displayed on the screen.

:SYSTem:PRINt:TEXT <string>

Parameter <string> = <STRING PROGRAM DATA>

"character string" Up to 40 characters

Function Outputs the character string specified by the parameter to the printer.

Note: Japanese language and double-quotation marks (") cannot be printed.

:SYSTem:PRINt:EALarm <item>

Parameter <item> = <STRING PROGRAM DATA>

"XXX:YYY:..." Sets output items, separated by a colon (:).

Output items

ERRor	Error
THReshold	Threhsold EI/%EFI
PCOunt	Performance Count
PPERcent	Performance %
ALARm	Alarm Interval
FREQuency	Clock Frequency Data

Function Prints the specified items of immediate error/alarm measurement data.

:SYSTem:PRINt:PATTern:TABLe:ADDRess <unit>,<form>,<start>,<end>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3 Unit inserted in slot3

4 Unit inserted in slot4

<form> = <CHARACTER PROGRAM DATA>

HEX Prints in hex

BIN Prints in binary

<start> = <NON-DECIMAL NUMERIC PROGRAM DATA>

#H0 to <end> Address(Hex) 0 to <end>/Step:1
(Print starting address: Specified in hex)

<end> = <NON-DECIMAL NUMERIC PROGRAM DATA>

<start> to Max Address(Hex) <start> to Max/Step:1
(Print ending address: Specified in hex)

Function Prints addressed pattern data (type Table) for a specified unit.

:SYSTem:PRINt:PATTern:TABLe:ALL <unit>,<form>

Parameter <unit> = <DECIMAL NUMERIC PROGRAM DATA>

3	Unit inserted in slot3.
4	Unit inserted in slot4.

<form> = <CHARACTER PROGRAM DATA>

HEX	Prints in hex.
BIN	Prints in binary.

Function Prints all pattern data (type Table) for a specified unit.

:SYSTem:PRINt:CANCel

Function Cancel to print.

:SYSTem:PRINt:EMARgin <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

[OFF] or [0]	Does not print the template data (when parameters omitted)
ON or 1	Prints the template data

Function Adds the template data to the Eye Margin result and prints.

:SYSTem:PRINt:REMargin <boolean>

Parameter <boolean> = <BOOLEAN PROGRAM DATA>

[OFF] or [0]	Does not print the template data (when parameters omitted)
ON or 1	Prints the template data

Function Adds the template data to the Recall Eye Margin and prints.

:SYSTem:TERMination <numeric>

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0	LF only
1	CR + LF

Function Specifies the termination characters to be added to the response character string outputted from MP1632C.

:SYSTem:TERMination?

Response <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0	LF only
1	CR + LF

Function Queries the termination characters to be added to the response character string outputted from MP1632C.

Note:

1. The termination character setting is maintained after initialization as shown below:
   1) Manual initialization
   2) Initialization using INI, :SYSTem:MEMory:INITialize or *RST commands
2. The termination character setting is not backed up. It is always set to 0 (LF only) at the activation of the main application software.

6.6.3. STATus Subsystem

The STATus subsystem sets and queries SCPI-prescribed status registers and device-specific status registers.

:STATus:PRESet

Function Initializes status registers. The following registers are influenced by the execution of this command; see Chapter 5 for more details:
- SCPI-prescribed status register Transition Filter, Enable Register
- Device-specific status register Transition Filter, Enable Register

■ **Setting and querying operation status registers**

<Event> **:STATus:OPERation[EVENt]?**

<Condition> **:STATus:OPERation:CONDition?**

<Enable> **:STATus:OPERation:ENABle <numeric>**

:STATus:OPERation:ENABle?

<PTRansition> **:STATus:OPERation:PTRansition <numeric>**

:STATus:OPERation:PTRansition?

<NTRansition> **:STATus:OPERation:NTRansition <numeric>**

:STATus:OPERation:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false
16	(BIT 4)	Error/alarm measurement in progress
256	(BIT 8)	An auto search in progress
512	(BIT 9)	Eye margin measurement in progress
2048	(BIT11)	Pattern load in progress
8192	(BIT13)	INSTrument status register summary

Set the sum total of the bit position values of the bits to be set.

Response <numeric> = <NR1 NUMERIC RESPNSE DATA>

Function Sets or queries the individual OPERation status register settings.

■ **Setting and querying operation status registers**

<Event> **:STATus:OPERation:INSTrument[:EVENt]?**

<Condition> **:STATus:OPERation:INSTrument:CONDition?**

<Enable> **:STATus:OPERation:INSTrument:ENABle <numeric>**

:STATus:OPERation:INSTrument:ENABle?

<PTRansition> **:STATus:OPERation:INSTrument:PTRansition <numeric>**

:STATus:OPERation:INSTrument:PTRansition?

<NTRansition> **:STATus:OPERation:INSTrument:NTRansition <numeric>**

:STATus:OPERation:INSTrument:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false.
4	(BIT 2)	Measurement ends.
16	(BIT 4)	Alarm changes.

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual INSTrument status register settings.

■ **Setting and querying questionable status registers**

<Event> **:STATus:QUEStionable[:EVENt]?**

<Condition> **:STATus:QUEStionable:CONDition?**

<Enable> **:STATus:QUEStionable:ENABle <numeric>**

:STATus:QUEStionable:ENABle?

<PTRansition> **:STATus:QUEStionable:PTRansition <numeric>**

:STATus:QUEStionable:PTRansition?

<NTRansition> **:STATus:QUEStionable:NTRansition <numeric>**

:STATus:QUEStionable:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false
512	(BIT 9)	MONitor status register summary

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual QUEStionable status register settings.

■ **Setting and querying MONitor status registers**

<Event> **:STATus:QUEStionable:MONitor[:EVENt]?**

<Condition> **:STATus:QUEStionable:MONitor:CONDition?**

<Enable> **:STATus:QUEStionable:MONitor:ENABle <numeric>**

:STATus:QUEStionable:MONitor:ENABle?

<PTRansition> **:STATus:QUEStionable:MONitor:PTRansition <numeric>**

:STATus:QUEStionable:MONitor:PTRansition?

<NTRansition> **:STATus:QUEStionable:MONitor:NTRansition <numeric>**

:STATus:QUEStionable:MONitor:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false.
1	(BIT 0)	Power fail occurred.
2	(BIT 1)	SLOT1 status register summary
8	(BIT 3)	SLOT3 status register summary
16	(BIT 4)	SLOT4 status register summary

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual MONitor status register settings.

■ **Setting and querying SLOT1 status registers**

<Event> **:STATus:QUEStionable:MONitor:SLOT1[:EVENt]?**

<Condition> **:STATus:QUEStionable:MONitor:SLOT1:CONDition?**

<Enable> **:STATus:QUEStionable:MONitor:SLOT1:ENABle <numeric>**

:STATus:QUEStionable:MONitor:SLOT1:ENABle?

<PTRansition> **:STATus:QUEStionable:MONitor:SLOT1:PTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT1:PTRansition?

<NTRansition> **:STATus:QUEStionable:MONitor:SLOT1:NTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT1:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false
1	(BIT 0)	PLL UNLOCK occurred.

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual SLOT1 status register settings.

■ **Setting and querying SLOT3 status registers**

<Event> **:STATus:QUEStionable:MONitor:SLOT3[:EVENt]?**

<Condition> **:STATus:QUEStionable:MONitor:SLOT3:CONDition?**

<Enable> **:STATus:QUEStionable:MONitor:SLOT3:ENABle <numeric>**

:STATus:QUEStionable:MONitor:SLOT3:ENABle?

<PTRansition> **:STATus:QUEStionable:MONitor:SLOT3:PTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT3:PTRansition?

<NTRansition> **:STATus:QUEStionable:MONitor:SLOT3:NTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT3:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false.
128	(BIT 7)	3.2G PPG status register summary

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual SLOT3 status register settings.

■ **Setting and querying SLOT4 status registers**

<Event> **:STATus:QUEStionable:MONitor:SLOT4[:EVENt]?**

<Condition> **:STATus:QUEStionable:MONitor:SLOT4:CONDition?**

<Enable> **:STATus:QUEStionable:MONitor:SLOT4:ENABle <numeric>**

:STATus:QUEStionable:MONitor:SLOT4:ENABle?

<PTRansition> **:STATus:QUEStionable:MONitor:SLOT4:PTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT4:PTRansition?

<NTRansition> **:STATus:QUEStionable:MONitor:SLOT4:NTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT4:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false
8	(BIT 3)	3.2G ED status register summary

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries the individual SLOT4 status register settings.

■ **Setting and querying 3.2G ED(Slot4) status registers**

<Event> **:STATus:QUEStionable:MONitor:SLOT4:G32E[:EVENt]?**

<Condition> **:STATus:QUEStionable:MONitor:SLOT4:G32E:CONDition?**

<Enable> **:STATus:QUEStionable:MONitor:SLOT4:G32E:ENABle <numeric>**

:STATus:QUEStionable:MONitor:SLOT4:G32E:ENABle?

<PTRansition> **:STATus:QUEStionable:MONitor:SLOT4:G32E:PTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT4:G32E:PTRansition?

<NTRansition> **:STATus:QUEStionable:MONitor:SLOT4:G32E:NTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT4:G32E:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false
1	(BIT 0)	Ins error occurred.
2	(BIT 1)	Omi error occurred.
4	(BIT 2)	Total Error occurred.
8	(BIT 3)	Delay Trouble occurred.
16	(BIT 4)	Pattern Sync Loss occurred.
32	(BIT 5)	Clock Loss occurred.

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries 3.2G ED(Slot4) status registers setting.

■ **Setting or querying 3.2G PPG(Slot3) status registers**

<Event> **:STATus:QUEStionable:MONitor:SLOT3:G32P[:EVENt]?**

<Condition> **:STATus:QUEStionable:MONitor:SLOT3:G32P:CONDition?**

<Enable> **:STATus:QUEStionable:MONitor:SLOT3:G32P:ENABle <numeric>**

:STATus:QUEStionable:MONitor:SLOT3:G32P:ENABle?

<PTRansition> **:STATus:QUEStionable:MONitor:SLOT3:G32P:PTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT3:G32P:PTRansition?

<NTRansition> **:STATus:QUEStionable:MONitor:SLOT3:G32P:NTRansition <numeric>**

:STATus:QUEStionable:MONitor:SLOT3:G32P:NTRansition?

Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0		All bits are false.
1	(BIT 0)	PPG Clock Loss occurred.
2	(BIT 1)	Delay Trouble occurred.

Response <numeric> = <NR1 NUMERIC RESPONSE DATA>

Function Sets or queries 3.2G ED(Slot4) 3.2G PPG(Slot3) status registers setting.

CHAPTER 7 DETAILS ON DEVICE MESSAGES(NATIVE)

7.1. Command List

Table 7-1 PPG Commadn List (1/2)

Function	Control Message		DataRequest Message	Details
	Header Part	Numeric data Part	Header Part	
Pattern Output Mode	PTO	NR1 Format	PTO?	7.3.1.1
Transmission Pattern	PTN	NR1 Format	PTN?	7.3.1.2
Transmission Pattern Mark Ratio	MRK	NR1 Format	MRK?	7.3.1.3
Transmission Pattern Logic	LGC	NR1 Format	LGC?	7.3.1.4
Bit Shift	SFT	NR1 Format	SFT?	7.3.1.5
Zero-Subst Pattern Length	ZPL	NR1 Format	ZPL?	7.3.1.6
Length of Consecutive Zeros of Zero-Subst Pattern	ZLN	NR1 Format	ZLN?	7.3.1.7
PRGM Pattern Length	DLN	NR1 Format	DLN?	7.3.1.8
Burst Mode	BRM	NR1 Format	BRM?	7.3.1.9
Burst Cycle	BRC	NR1 Format	BRC?	7.3.1.10
Burst Enable Length	BRE	NR1 Format	BRE?	7.3.1.11
Number of Pattern Pages	PAG ADR	NR1 Format	PAG? ADR?	7.3.1.12
Pattern Bit	BIT	NR1 Format or HEX Format	BIT?	7.3.1.13
Pattern Data	PDT	HEX Format or Binary Format	PDT?	7.3.1.14
Pattern Binary Data	BDT	HEX Format	BDT?	7.3.1.15
Reversion of Transmission Pattern Specification Address Data (address sepecification)	DRA	HEX Format	-	7.3.1.16
Reversion of Transmission Pattern Specification Address Data (delta specification)	DRD	HEX Format	-	7.3.1.17
Pattern Data Input Byte Number	WRT	NR1 Format	-	7.3.1.18
Pattern Data Output Byte Number	-	-	RED	7.3.1.19
ALL Preset(All Pages)	ALL	NR1 Format	-	7.3.1.20
ALL Preset(One Page)	PST	NR1 Format	-	7.3.1.21
Error Insertion	EAD	NR1 Format	EAD?	7.3.1.22
Error Insertion Route	ECH	NR1 Format	ECH?	7.3.1.23
Level of PRBS 1	PML	NR1 Format	PML?	7.3.1.24
Level of PRGM 1	GML	NR1 Format	GML?	7.3.1.25
On/Off of Output	OON	NR1 Format	OON?	7.3.2.1
ON/OFF of Clock Output	CON	NR1 Format	CON?	7.3.2.2
ON/OFF of XClock Output	XCO	NR1 Format	XCO?	7.3.2.3
Setting of Clock Output Delay	CDL	NR1 Format	CDL?	7.3.2.4
Clock Output Level	CLL	NR1 Format	CLL?	7.3.2.5
XClock Output Level	XCL	NR1 Format	XCL?	7.3.2.6
Clock Output Amplitude	CAP	NR2 Format	CAP?	7.3.2.7

Table 7-1 PPG Command List (2/2)

Function	Control Message		DataRequest Message	Details
	Header Part	Numeric data Part	Header Part	
XClock Output Amplitude	XCA	NR2 Format	XCA?	7.3.2.8
Clock Output Offset	COS	NR2 Format	COS?	7.3.2.9
XClock Output Offset	XCF	NR2 Format	XCF?	7.3.2.10
Offset Reference Port Selection	OPS	NR1 Format	OPS?	7.3.2.11
Offset Reference Value	OFS	NR1 Format	OFS?	7.3.2.12
Setting of Clock Duty	CDT	NR1 Format	CDT?	7.3.2.13
Setting of XClock Duty	XDT	NR1 Format	XDT?	7.3.2.14
setting of Clock DC impedance	CIM	NR1 Format	CIM?	7.3.2.15
ON/OFF of Data Output	DON	NR1 Format	DON?	7.3.2.16
ON/OFF of XData Output	XDO	NR1 Format	XDO?	7.3.2.17
Setting of Data DC impedance	DIM	NR1 Format	DIM?	7.3.2.18
Data Cross Point	DCR	NR1 Format	DCR?	7.3.2.19
XData Cross Point	XDC	NR1 Format	XDC?	7.3.2.20
Data Output level	DAL	NR1 Format	DAL?	7.3.2.21
XData Output level	XDL	NR1 Format	XDL?	7.3.2.22
Data Output Amplitude	DAP	NR2 Format	DAP?	7.3.2.23
XData Output Amplitude	NAP	NR2 Format	NAP?	7.3.2.24
Data Output Offset	DOS	NR2 Format	DOS?	7.3.2.25
XData Output Offset	NOS	NR2 Format	NOS?	7.3.2.26
Selection of Sync Output Signal	SOP	NR1 Format	SOP?	7.3.2.27
Selection of Pattern Sync Output Position	PSP	NR1 Format	PSP?	7.3.2.28
PPG Clock Loss	-	-	PCL?	7.3.3.1
Delay Trouble3	-	-	DTR?	7.3.3.2
Delay Setting State	-	-	DLY?	7.3.3.3
Setting of Clock/XClock Grouping	CGR	NR1 Format	CGR?	7.3.4.1
Setting of Data/XData Grouping	TRK	NR1 Format	TRK?	7.3.4.2
PPG/ED Pattern Common Setting	PCO	NR1 Format	PCO?	7.3.4.3

Table 7-2 Internal Synthesyzer Command List

Function	Control Message		Data Request Message	Details
	Header Part	Numeric data Part	Header Part	
Setting of Clock Frequency	FRQ	NR1 Format	FRQ?	7.4.1.1
Clock Reference Signal	RFC	NR1 Format	RFC?	7.4.1.2
PLL Lock State	-	NR1 Format	PLL?	7.4.1.3

Table 7-3 ED Command List (1/3)

Function	Control Message		DataRequestMessage	Details
	Header Part	Numeric data Part	Header Part	
Pattern Reception Mode (Pattern input mode)	PTI	NR1 Format	PTI?	7.5.1.1
Reception Pattern	PTN	NR1 Format	PTN?	7.5.1.2
Reception Pattern Mark Ratio	MRK	NR1 Format	MRK?	7.5.1.3
Reception Pattern Logic	LGC	NR1 Format	LGC?	7.5.1.4
Bit Shift	SFT	NR1 Format	SFT?	7.5.1.5
Zero-Subst Pattern Length	ZPL	NR1 Format	ZPL?	7.5.1.6
Length of Consecutive Zeros of Zero-Subst Pattern	ZLN	NR1 Format	ZLN?	7.5.1.7
PRGM Pattern Length	DLN	NR1 Format	DLN?	7.5.1.8
Automatic Synchronization Function	SYN	NR1 Format	SYN?	7.5.1.9
Synchronization Loss Threshold	LTH	NR1 Format	LTH?	7.5.1.10
Synchronization Gain Threshol	GTH	NR1 Format	GTH?	7.5.1.11
Synchronization Gain Threshol	ITH	NR1 Format	ITH?	7.5.1.12
Internal Synchronization Threshold	FSY	NR1 Format	FSY?	7.5.1.13
Frame Length	FLN	NR1 Format	FLN?	7.5.1.14
Number of Pattern Pages	PAG ADR	NR1 Format	PAG? ADR?	7.5.1.15
BIT Pattern Bit	BIT	NR1 Format	BIT?	7.5.1.16
Pattern Data	PDT	HEX Format or BINARY Format	PDT?	7.5.1.17
Pattern Binary Data	BDT	HEX Format or BINARY Format	BDT?	7.5.1.18
Reversion of Transmission Pattern Specification Address Data (address specification)	DRA	HEX Format	-	7.5.1.19
Reversion of Transmission Pattern Specification Address Data (delta specification)	DRD	HEX Format	-	7.5.1.20
Pattern Data Input Byte Number	WRT	NR1 Format	-	7.5.1.21
Pattern Data Output Byte Number	-	-	RED	7.5.1.22
Preset (All pages)	ALL	NR1 Format	-	7.5.1.23
Preset (One page)	PST	NR1 Format	-	7.5.1.24
Level of PRBS 1	PML	NR1 Format	PML?	7.5.1.25
Level of PRGM 1	GML	NR1 Format	GML?	7.5.1.26
Setting of Measurement Item	TIT	NR1 Format	TIT?	7.5.1.27
Measurement Start/Restart	STA	-	-	7.5.1.28
Measurement Stop	STO	-	-	7.5.1.29
measurement restart setting	MRS	NR1 Format	MRS?	7.5.1.30
Setting of Measurement Mode	MOD	NR1 Format	MOD?	7.5.1.31
Measurement Period Setting	PRD	NR1 Format	PRD?	7.5.1.32
Timed Start Setting	TSS	NR1 Format	TSS?	7.5.1.33
Timed Start Time Setting	STI	NR1 Format	STI?	7.5.1.34
Error Detection Mode Selection	ETY	NR1 Format	ETY?	7.5.1.35
Measurement Mask Route Selection	SCH	BINARY Format	SCH?	7.5.1.36

Table 7-3 ED Command List (2/3)

Function	Control Message		DataRequestMessage	Details
	Header Part	Numeric data Part	Header Part	
Measurement Start Time	-	-	MSA?	7.5.1.37
Measurement Stop Time	-	-	MSO?	7.5.1.38
Measurement Status	-	-	MSR?	7.5.1.39
Measurement Elapsed Time	-	-	MLP?	7.5.1.40
Measurement Residual Time	-	-	ETI?	7.5.1.41
Intermediate Data Creation Time	-	-	INT?	7.5.1.42
Alarm Occurrence Time	-	-	AOT?	7.5.1.43
Alarm Recovery Time	-	-	ART?	7.5.1.44
Setting of Eye Margin Measurement Mode	EMD	NR1 Format	EMD?	7.5.1.45
Eye Margin Measurement Item Setting	EYT	NR1 Format	EYT?	7.5.1.46
Eye Margin Measurement Threshold Setting	EMT	NR1 Format	EMT?	7.5.1.47
Eye Margin Measurement Resolution Setting	EMR	NR1 Format	EMR?	7.5.1.48
Eye Diagram Measurement Threshold Setting	EDT	BINARY Format	EDT?	7.5.1.49
Eye Diagram Measurement Effective Measurement Points	-	-	EDP?	7.5.1.50
Setting of Auto Search Mode	ASM	NR1 Format	ASM?	7.5.1.51
Auto Search Function	SRH	NR1 Format	SRH?	7.5.1.52
PRBS Pattern Search Function	PSH	NR1 Format	PSH?	7.5.1.53
Clock Input Polarity	CPL	NR1 Format	CPL?	7.5.2.1
Clock Input Phase Setting	CPA	NR1 Format	CPA?	7.5.2.2
Clock Input Terminal Voltage	CTM	NR1 Format	CTM?	7.5.2.3
Data Input Level	DAL	NR1 Format	DAL?	7.5.2.4
Data Input Threshold Setting	DTH	NR1 Format	DTH?	7.5.2.5
Sets/queries Data Input Terminal Voltage.	DTM	NR1 Format	DTM?	7.5.2.6
Clock Loss Process Option	CLS	NR1 Format	CLS?	7.5.3.1
Sync Loss Process Option	SLS	NR1 Format	SLS?	7.5.3.2
Measurement Interval Time Selection	ITV	NR1 Format	ITV?	7.5.3.3
Error Performance Data Measurement ON/OFF	PRF	NR1 Format	PRF?	7.5.3.4
Error Performance Threshold Setting	ETH	NR1 Format	ETH?	7.5.3.5
Threshold EI / %EFI Measurement ON/OFF	TEI	NR1 Format	TEI?	7.5.3.6
Power Fail Measurement ON/OFF	PFM	NR1 Format	PFM?	7.5.3.7
Error Ratio Measurement Result	-	-	ER?	7.5.3.8
Error Count Measurement Result	-	-	EC?	7.5.3.9
Clock Count Measurement Result	-	-	CC?	7.5.3.10
Error Interval Count Measurement Result	-		EI?	7.5.3.11

Table 7-3 ED Command List (3/3)

Function	Control Message		DataRequest Message Header Part	Details
	Header Part	Numeric data Part		
Error Free Interval Rate Measurement	-	-	EFI?	7.5.3.12
Clock Frequency Measurement Result	-	-	FRQ?	7.5.3.13
Performance % Measurement Result	-	-	PFP?	7.5.3.14
Performance Count Measurement Result	-	-	PFC?	7.5.3.15
Threshold EI Measurement Result	-	-	THE?	7.5.3.16
Threshold %EFI Measurement Result	-	-	THF?	7.5.3.17
Alarm Interval Measurement Result	-	-	AIN?	7.5.3.18
One Second Average Error Ratio Measurement Result	-	-	OER?	7.5.3.19
One Second Average Error Count Measurement Result	-	-	OEC?	7.5.3.20
Data Output Format	FMT	NR1 Format	FMT?	7.5.3.21
Intermediate Measurement Result Output Function	-	-	IMD?	7.5.3.22
Eye Margin Measurement Result	-	-	EMM?	7.5.3.23
Eye Diagram Measurement Result	-	-	EDM?	7.5.3.24
Measurement Result Store Function	BST	NR1 Format	-	7.5.3.25
Measurement Result Store Buffer Clear Function	BCL	-	-	7.5.3.26
Clock Loss Status	-	-	CLI?	7.5.3.27
Sync Loss Status	-	-	SLI?	7.5.3.28
Error Detection Status	-	-	ERS?	7.5.3.29
Error Detection Status	-	-	BES?	7.5.3.30
Power Fail Status	-	-	POF?	7.5.3.31
Delay Trouble	-	-	DTR?	7.5.3.32
Delay Setting Status	-	-	DLY?	7.5.3.33
Synchronous Signal Output Selection	SOP	NR1 Format	SOP?	7.5.4.1
PPG/ED Pattern Common Setting	PCO	NR1 Format	PCO?	7.5.5.1

Table 7-4 Main Frame Command List (1/3)

Function	Control Message		DataRequestMessage	Details
	Header Part	Numeric data Part	Header Part	
Child Window Open/Close	WOC	BINARY Format	WOC?	7.6.1.1
System Child Window panel selection	SYW	NR1 Format	SYW?	7.6.1.2
Setup Child Window Panel Selection	SEW	NR1 Format	SEW?	7.6.1.3
Test Menu Child Window Panel Selection	TEW	NR1 Format	TEW?	7.6.1.4
Real Time/Measurement Time Display Selection	TIM	NR1 Format	TIM?	7.6.1.5
Result Child Window Sub Window Selection	REW	NR1 Format	REW?	7.6.1.6
Intermediate Result Display Function	CUR	NR1 Format	CUR?	7.6.1.7
Performance Result Display Selection	PFD	NR1 Format	PFD?	7.6.1.8
Zoom window display ON/OFF setting	ZOM	NR1 Format	ZOM?	7.6.1.9
Zoom Window Display Item Setting	ZOI	NR1 Format	ZOI?	7.6.1.10
Zoom Window Performance Item Setting	ZOP	NR1 Format	ZOP?	7.6.1.11
Zoom Window Threshold Ei/%Efi Setting	ZOT	NR1 Format	ZOT?	7.6.1.12
History Lamp Reset	HRE	-	-	7.6.1.13
Eye Margin Measurement Display Error Ratio	DER	BINARY Format	DER?	7.6.1.14
Eye Diagram Display Window Selection	EDD	NR1 Format	EDD?	7.6.1.15
Eye Diagram Display Auto Scale	ASC	-	-	7.6.1.16
Eye Diagram Voltage Scale Setting	VSC	NR2 Format	VSC?	7.6.1.17
Eye Diagram Phase Scale Setting	PSC	NR1 Format	PSC?	7.6.1.18
Eye Diagram Marker Display ON/OFF	EMA	NR1 Format	EMA?	7.6.1.19
Marker Operation Mode	MKM	NR1 Format	MKM?	7.6.1.20
Marker Specification Threshold	MKT	NR1 Format	MKT?	7.6.1.21
Marker Move Specification	MMV	NR1 Format	-	7.6.1.22
Marker Position	MKP	NR1 Format	MKP?	7.6.1.23
Customize Customize Window Display Slot Setting	CUS	NR1 Format	CUS?	7.6.1.24
Customize Window Display Item Setting	CUI	NR1 Format	CUI?	7.6.1.25
Customize Window Display Item Port Setting	CUP	NR1 Format	CUP?	7.6.1.26
Customize Window Pattern Setting Unit	CPT	NR1 Format	CPT?	7.6.1.27
Customize Edit Pattern Position Specification	CPF	NR1 Format	CPF?	7.6.1.28
Customize Window Measurement Display Item	CRI	NR1 Format	CRI?	7.6.1.29
System Error Monitor Buzzer ON/OFF Setting	SYS	NR1 Format	SYS?	7.6.2.1
Error Monitor Buzzer ON/OFF Setting	MON	NR1 Format	MON?	7.6.2.2
Alarm Monitor Buzzer ON/OFF Setting	ALM	NR1 Format	ALM?	7.6.2.3

Table 7-4 Main Frame Command List (2/3)

Function	Control Message		DataRequest Message	Details
	Header Part	Numeric data Part	Header Part	
System Error Type	SYT	BINARY Format	SYT?	7.6.2.4
Internal Timer Setting	RTM	NR1 Format	RTM?	7.6.2.5
Input / Output Buffer Size	BUF	NR1 Format	BUF?	7.6.2.6
Hardware Organization	HDR	NR1 Format	HDR?	7.6.2.7
Software Organization	SFR	NR2 Format	SFR?	7.6.2.8
Data Recall	RCL	NR1 Format	-	7.6.2.9
Data Save	SAV	NR1 Format	-	7.6.2.10
Quick Recall	QRC	STRING DATA	-	7.6.2.11
Quick Save	QSA	STRING DATA	-	7.6.2.12
File No./Directory Mode Selection	FIL	NR1 Format	FIL?	7.6.2.13
File Catalog	-	-	CAT?	7.6.2.14
Floppy Access Status	-	NR1 Format	MAC?	7.6.2.15
Initialize	INI	-	-	7.6.2.16
Hard Copy	HCP	-	-	7.6.2.17
Prints Text	PTX	STRING DATA	-	7.6.2.18
Threshold EI/%EFI Print Selection	THR	NR2 Format	THR?	7.6.2.19
Performance Data Print Selection	EPF	NR1 Format	EPF?	7.6.2.20
Error Measurement Print	ERP	NR1 Format	ERP?	7.6.2.21
Alarm Interval Print Select	ALP	NR1 Format	ALP?	7.6.2.22
Frequency Data Print Select	FRP	NR1 Format	FRP?	7.6.2.23
Manual Print Start	PSA	-	-	7.6.2.24
Addressed Pattern Print	PAD	NR1 Format	-	7.6.2.25
All Pattern Print	PAL	NR1 Format	-	7.6.2.26
Eye Margin Measurement Result Print	PEM	-	-	7.6.2.27
Recall Eye Diagram Print	PER	-	-	7.6.2.28
LSB,MSB Swap Command	SWP	NR1 Format	SWP?	7.6.2.29
Termination Select	TRM	-	TRM?	7.6.2.30

Table 7-4 Main Frame Command List (3/3)

Function	Control Message		DataRequest Message	Details
	Header Part	Numeric data Part	Header Part	
Service Request Enable Register(ED)	SRQ	NR1 Format	SRQ?	5.7.1.1
Status Byte Register(ED)	STB	BINARY Format	STB?	5.7.1.2
Standard Event Status Enable Register(ED)	ESE	NR1 Format	ESE?	5.7.1.3
Event Status Register(ED)	ESR	BINARY Format	ESR?	5.7.1.4
Extended Event Status Enable Register(ED)	EES	NR1 Format	EES?	5.7.1.5
Event Status Register(ED)	EER	BINARY Format	EER?	5.7.1.6
Service Request Enable Register (PPG/Synthe)	SRQ	NR1 Format	SRQ?	5.7.2.1
Status Byte Register (PPG/Synthe)	STB	BINARY Format	STB?	5.7.2.2
Standard Event Status Enable Register(PPG/Synthe)	ESE	NR1 Format	ESE?	5.7.2.3
Event Status Register (PPG/Synthe)	ESR	BINARY Format	ESR?	5.7.2.4
Extended Event Status Enable Register (PPG/Synthe)	EES	NR1 Format	EES?	5.7.2.5
Event Status Register(PPG/Synthe)	EER	BINARY Format	EER?	5.7.2.6

7.2. Difference with Conventional BERTS (MP1650A/MP1651A)

To run programs which have conventionally been used by MP1650A or MP1651A on this instrument, these programs need to be re-configured to a certain extent in regard to the following items.

7.2.1. Restriction on Address

MP1650A and MP1651A allowed the user to have freedom to set any value in the range between 0 and 30 as the GPIB address. On this instrument, however, for reasons relating to the hardware restrictions, the address of MP163220C PPG + OPT-03 Internal Synthesizer and the address used by MP163240C ED must be two consecutive addresses. Addresses can be specified in the range between 2 and 28. In this case, it is up to the user to decide which one of these addresses will be allocated to an odd number address or an even number address.

7.2.2. Difference between PPG and Internal Synthesizer Commands

The table below shows commands enabled by MP1650A but disabled by this instrument.

Table 7-5 Commands Disabled by MP1632C

Control Message	Request Message	Function
CLK	CLK?	Switching of the operating clock
RES	RES?	Internal clock frequency setting esolution
WNB	WNB?	No. of words
WLN	WLN?	Word length
SPD	SPD?	Switching of output between the front and the rear
SCH	SCH?	No. of channels for external pattern input
MEM	MEM?	Switching of the memory function
RSV	—	Re-saving of data
—	PWI?	Power interruption and recovery state

The table below shows the commands not fully compatible with the commands of MP1650A.

Table 7-6 Commands Not Entirely Compatible

Control Message	Request Message	Function	Difference
FRQ	FRQ?	Internal clock frequency	The resolution of setting the frequency is fixed at kHz. The return value is a fixed, seven-digit value.
PTN	PTN?	Output pattern	Selecting PROG.WORD is disabled. Zero-Subst was added.
DLN	DLN?	Data length	With the setting range having been expanded, the return value comes in a fixed, seven-digit value.
PAG ADR	PAG? ADR?	Number of pages	With the setting range having been expanded, the return value comes in a fixed, six-digit value.
BIT	BIT?	Pattern bit	The method of making a query changed. See page 7-14 "How to Use Bit?" for more information.
WRT	—	Number of bytes for pattern data input	The maximum number of bytes to be transferred at a time was changed from 65536 to 400 bytes. In addition, as the new transfer method does not use DMA, a maximum of approximately three minutes need to be set for a single transfer.
RED	—	Number of bytes for pattern data output	The maximum number of bytes to be transferred at a time was changed from 65536 to 400 bytes. Moreover, the new transfer method does not use DMA.
OFS	OFS?	Reference value for offsetting output	The commands used in the past specified reference values for all ports. However, the new commands only require the changing of reference values for pre-specified ports.
FIL	FIL?	Switching of the file No. and the directory mode	The format of the return value of directory information was changed. See examples of command application for more information.
RCL	—	Recalling of data	The parameter was changed.
SAV	—	Saving of data	The parameter was changed.
ECH	ECH?	Error insertion channel	The number of error insertion channels decreased from 32 channels to 8 channels.

7.2.3. Difference in ED Command

The table below shows commands enabled by MP1651A but disabled by this instrument.

Table 7-7 Commands Disabled by MP1632C

Control Message	Request Message	Function
WNB	WNB?	Number of words
WLN	WLN?	Word length
MEM	MEM?	Switching of the memory function
RSV	—	Re-saving of data
FRM	FRM?	Frame / word length
DMS	DMS?	Display mode
PRN	PRN?	Printer function
MSK	MSK?	Selection of the masked route length
DGT	DGT?	Selection of the number of digits to be displayed
CAL	CAL?	Function to calculate data during the measurement
ITM	ITM?	Selection of intermediate data printing
OSC	OSC?	Selection of a single second data printing
DOT	DOT?	Selection of a single second data printing threshold
PSV	PSV?	Paper saving function
—	MTY?	Function to select GPIB output of measurement results

Table 7-8 Commands Not Entirely Compatible

Control Message	Request Message	Function	Difference
PTN	PTN?	Output pattern	Selecting PROG.WORD is disabled. Zero-Subst was added.
FLN	FLN?	Frame length	The frame length can be set in eight steps between the minimum value of 8 and the maximum value of 32.
DLN	DLN?	Data length	With the setting range having been expanded, the return value comes in a fixed, seven-digit value.
PAG ADR	PAG? ADR?	Number of pages	With the setting range having been expanded, the return value comes in a fixed, six-digit value.
BIT	BIT?	Pattern bit	The method of making a query changed. See page 7-14 "How to Use Bit?" for more information.
WRT	—	Number of bytes for pattern data input	The maximum number of bytes to be transferred at a time was changed from 65536 to 400 bytes. In addition, as the new transfer method does not use DMA, a maximum of approximately three minutes need to be set for a single transfer.
RED	—	Number of bytes for pattern data output	The maximum number of bytes to be transferred at a time was changed from 65536 to 400 bytes. Moreover, the new transfer method does not use DMA.
TIM	TIM?	Switching of real-time measurement time	The parameter was changed.
FIL	FIL?	Switching of the file No. and the directory mode	The format of the return value of directory information was changed. See examples of command application for more information.
RCL	—	Recalling of data	The parameter was changed.
SAV	—	Saving of data	The parameter was changed.
SCH	SCH?	Number of masked route channels	The number of channels is always fixed at no more than eight.
ITV	ITV?	Selection of the measurement interval time	Selecting 200 mS and 500 mS is disabled, but selecting 1S is enabled.

How to Use BIT?

If the output order to GPIB, the input order from GPIB, the GPIB address and the variable to hold the value read out are assumed to be OUTPUT, ENTER, Gadr and B$ respectively, 29 pages equivalent of PRGM patterns were read out by MP1650A or MP1651A in the manner shown below. In this case, 16 bits constitute a page.

```
OUTPUT  Gadr ; "PAG 1"
OUTPUT  Gadr ; "BIT?"
FOR I = 1 TO 4
   ENTER Gadr ; B$
   PRINT B$
NEXT I
```

However, the method was changed as shown below on MP1632C. More specifically, the commands are sent before the value is read out.

```
OUTPUT  Gadr ; "PAG 1"
FOR I = 1 TO 4
  OUTPUT  Gadr ; "BIT?"
   ENTER Gadr ; B$
   PRINT B$
NEXT I
```

The printed results of both the application examples mentioned above are as follows.

```
PAG  1 ; BIT #H0000, #H0000, #H0000, #H0000, #H0000, #H0000, #H0000, #H0000
PAG  9 ; BIT #H0000, #H0000, #H0000, #H0000, #H0000, #H0000, #H0000, #H0000
PAG 17 ; BIT #H0000, #H0000, #H0000, #H0000, #H0000, #H0000, #H0000, #H0000
PAG 25 ; BIT #H0000, #H0000, #H0000, #H0000, #H0000
```

7.3. Setting of 3.2G PPG (MU163220C)

This section describes the details of the setting/query commands concerning 3.2G PPG.

Note 1) For unstipulated commands and parameters, corresponding errors take place.

Note 2) A command error takes place when an intrinsic command of this device is sent in a state where the 3.2G PPG unit is not in the measuring instrument.

Note 3) In regard to parameter setting, if a parameter is set in a step smaller than the pre-set steps, the smallest pre-set value which comes closest to the value specified will be set. In other words, any number which fails to reach a full unit will be truncated.

Note 4) The space is expressed by a △.

7.3.1. Sending Pattern Related

Makes settings and queries concerning the transmission patterns.

7.3.1.1. PTO Pattern Output Mode (Pattern Output mode)

Function Sets/queries the method of outputting the transmission patterns.

Header	Program	Query	Response (No. of characters)
PTO	PTO△m	PTO?	PTO△m (FIX1)

Parameter m = <NR1>
0 : Repeat mode
1 : Burst mode

Application example

Program

・Sets the transmission pattern at the repeat mode.

```
> PTO△0
```

Query

・Queries the method of outputting the transmission pattern when it is in the burst mode.

```
>PTO?
<PTO△1
```

7.3.1.2. PTN Transmission Pattern (Pattern mode)

Function Sets/queries the type of transmission pattern.

Header	Program	Query	Response (No. of characters)
PTN	PTN△m	PTN?	PTN△m (FIX2)

Parameter m = <NR1>

1 : PRGM
2 : PRBS7
3 : PRBS9
5 : PRBS11
6 : PRBS15
7 : PRBS20
8 : PRBS23
9 : PRBS31
10 : Zero-Subst

Application example

Program

・Sets the transmission pattern at PRGM.

```
> PTN△1
```

Query

・Queries the transmission pattern when the parameter is set at PRBS23.

```
>PTN?
< PTN△△8
```

7.3.1.3. MRK Transmission Pattern Mark Ratio (Mark ratio mode)

Function Sets/queries the mark ratio when the transmission pattern is set at PRBS.

Header	Program	Query	Response (No. of characters)
MRK	MRK△m	MRK?	MRK△m (FIX1)

Parameter m = <NR1>

0 : 0/8 or 8/8
1 : 1/8 or 7/8
2 : 1/4 or 3/4
3 : 1/2 or 1/2INVT

Application example

Program

・Sets the mark ratio at 1/8 or 7/8.

```
> MRK△1
```

Query

・Queries the output pattern when the mark ratio is set at either 3/4 or 1/4

```
> MRK?
< MRK△2
```

・Making a query when the PRGM pattern is set.

```
> MRK?
< ERR
```

Remarks On MP1650A and MP1651A, 0/8 and 8/8, 1/8 and 7/8, 1/4 and 3/4, and 1/2 and 1/2INVT had the same set value depending on the POS/NEG state of Logic. However, MP1632C has no Logic POS/NES setting of the PRBS pattern. Accordingly, to establish compatibility with the commands of MP1650A and MP1651A, a pseudo Logic setting was added to PRBS. The table below shows the relations between Logic and the Mark ratio.

Logic	Mark ratio			
POS	0/8	1/8	1/4	1/2
NEG	8/8	7/8	3/4	1/2INVT

7.3.1.4. LGC Transmission Pattern Logic (Logic mode)

Function Sets/queries the transmission pattern logic.

Header	Program	Query	Response (No. of characters)
LGC	LGC△m	LGC?	LGC△m (FIX1)

Parameter m = <NR1>

0 : POS(PRGM、at Zero-Subst) 0/8,1/8,1/4,1/2(at PRBS)

1 : NEG(PRGM、at Zero-Subst) 8/8,7/8,3/4,1/2INVT(at PRBS)

Application example

Program

・Sets logic at POS when the transmission pattern logic is set at PRBS.

```
> LGC△0
```

・Sets the mark ratio at 3/4 when the transmission pattern logic is set at PRBS and the mark ratio is 1/4.

```
> LGC△1
```

Query

・Queries logic when the transmission pattern logic is set at PRBS and the mark ratio is 1/8.

```
> LGC?
< LGC△0
```

・Queries logic when the transmission pattern logic is set NEG at Zero-Subst.

```
> LGC?
< LGC△1
```

7.3.1.5. SFT Bit Shift (Mark ratio bit shift)

Function Sets/queries the number of bit shifts of the PRBS pattern mark ratio.

Header	Program	Query	Response (No. of characters)
SFT	SFT△m	SFT?	SFT△m (FIX1)

Parameter m = <NR1>

0 : 1 bit shift

1 : 3 bit shift

Application example

Program

・Sets the bit shift at 1 bit.

```
> SFT△0
```

Query

・Makes a query when the parameter is set at 3 bit shift.

```
> SFT?
< SFT△1
```

・Queries logic when the bit shift is NEG at Zero-Subst.

```
> SFT?
< ERR
```

7.3.1.6. ZPL Zero-Subst Pattern Length (Zero-Subst Pattern Length)

Function Sets/queries the length of the Zero-Subst pattern.

Header	Program	Query	Response (No. of characters)
ZPL	ZPL△m	ZPL?	ZPL△m (FIX1)

Parameter m = <NR1>

2 : 2^7
3 : 2^9
5 : 2^11
6 : 2^15

Application example

Program

・Sets the pattern length at 2 ^ 15.

```
> ZPL△6
```

Query

・Makes a query when the pattern length is 2 ^ 9.

```
> ZPL?
< ZPL△3
```

・Makes a query when the pattern length is set at PRBS.

```
> ZPL?
< ERR
```

7.3.1.7. ZLN Length of Consecutive Zeros of Zero-Subst Pattern (Zero Length)

Function Sets/queries the length of consecutive zeros of the Zero-Subst pattern.

Header	Program	Query	Response (No. of characters)
ZLN	ZLN△m	ZLN?	ZLN△m (FIX5)

Parameter m = <NR1>

Sets the length of consecutive zeros within the range shown below.

7 to 127 / step 1 : 2^7
9 to 511 / step 1 : 2^9
11 to 2047 / step 1 : 2^11
15 to 32767 / step 1 : 2^15

Application example

Program

・Sets the length of consecutive zeros at 98 bits.

```
> ZLN△98
```

Query

・Makes a query when the length of consecutive zeros is 423 bits.

```
> ZLN?
< ZLN△△△423
```

・Makes a query when the length of consecutive zeros is set at PRBS.

```
> ZLN?
< ERR
```

7.3.1.8. DLN PRGM Pattern Length (PRGM data length)

Function Sets/queries the PRGM pattern length.

Header	Program	Query	Response (No. of characters)
DLN	DLN△m	DLN?	DLN△m (FIX7)

Parameter m = <NR1>

Sets the pattern length within the range shown below.

2 to 131072 : step 1
131072 to 262144 : step 2
262144 to 524288 : step 4
524288 to 1048576 : step 8
1048576 to 2097152 : step 16
2097152 to 4194304 : step 32
4194304 to 8388608 : step 64

Application example

Program

・Sets the pattern length at 1048576.

```
> DLN△1048576
```

Query

・Makes a query when the pattern length is 514 bits.

```
> DLN?
< DLN△△△△△514
```

・Makes a query when the pattern length is set at PRBS.

```
> DLN?
< ERR
```

7.3.1.9. BRM Burst Mode (Pattern burst mode)

Function Sets/queries the burst mode of the transmission pattern.

Header	Program	Query	Response (No. of characters)
BRM	BRM△m	BRM?	BRM△m (FIX1)

Parameter m = <NR1>

0 : INT (Internal burst)
1 : EXT (External burst)

Application example

Program

・Sets the Mode at INT.

```
> BRM△0
```

Query

・Makes a query when the Mode is EXT.

```
> BRM?
< BRM△1
```

・Makes a query when the Mode is set at Repeat.

```
> BRM?
< ERR
```

7.3.1.10. BRC Burst Cycle (Pattern burst cycle)

Function Sets/queries the burst cycle of the transmission pattern.

Header	Program	Query	Response (No. of characters)
BRC	BRC△m	BRC?	BRC△m (FIX5)

Parameter m = <NR1>
2 to 50000 2 to 50000 us/Step:1

Application example

Program

・Sets the Burst Cycle at 10.us

```
> BRC△10
```

Query

・Makes a query when the Burst Cycle is 10000 us.

```
> BRC?
< BRC△10000
```

・Makes a query when the Mode is set at Repeat.

```
> BRC?
< ERR
```

7.3.1.11. BRE Burst Enable Length (Burst enable length)

Function Sets/queries the enable length when the transmission pattern is set at burst.

Header	Program	Query	Response (No. of characters)
BRE	BRE△m	BRE?	BRE△m (FIX5)

Parameter m = <NR1>
1 to 49999 1 to 49999 us/Step:1

Application example

Program

・Sets the Burst Enable Length at 10 us.

```
> BRE△10
```

Query

・Makes a query when the Burst Cycle is 10000 us.

```
>BRE?
< BRE△10000
```

・Makes a query when the Mode is set at Repeat.

```
> BRE?
< ERR
```

7.3.1.12. PAG Number of Pattern Pages (Page)

Function Specifies/queries the head of the PRGM pattern to be edited by 16 bits. This unit of 16 bits is called a page.

Header	Program	Query	Response (No. of characters)
PAG	PAG△m	PAG?	PAG△m (FIX6)
ADR	ADR△m	ADR?	ADR△m (FIX6)

Parameter m = <NR1>
1 to 524288 / Step 1

Application example

Program

・Sets the head of the editing target at the 1000th page.
(The bit number from the head of the pattern is the 16000th bit.)

```
> PAG△1000
```

Sets the head of the editing target at the 1000th page.
(The bit number from the head of the pattern is the 16000th bit.)

```
> ADR△1000
```

Query

・Makes a query when the head of the editing target is the 100th page.

```
> PAG?
< PAG△△△△100
```

・Makes a query when the head of the editing target is the 100th page.

```
> ADR?
< ADR△△△△100
```

・Makes a query when the head of the editing target is set at Zero-Subst.

```
> PAG?
< ERR
```

・Makes a query when the head of the editing target is set at Zero-Subst.

```
> ADR?
< ERR
```

Remarks This command, available in two types, PAG and ADR, has the same function in both formats. The same result is obtained by executing either format. The value enabled to be set varies by the PRGM pattern length. The maximum value enabled to be set is the quotient of the pattern length divided by 16. In the presence of any remainder, the value to be set is the quotient + 1. Moreover, when a value which exceeds the maximum value enabled to be set is set, the maximum value effective in that state will be set. Incidentally, the initial value will be set at 1 when no value is set by this command.

7.3.1.13. BIT BIT Pattern Bit (Pattern bit)

Function Specifies/queries a maximum of eight pages equivalent of PRGM patterns by 16 bits from the position specified by PAG/ADR.

Header	Program	Query	Response (No. of characters)
BIT	BIT△m	BIT?	BIT△m (as shown below)
	BIT△#Hm		

Parameter m = <NR1>
0 to 65535 / Step 1
#Hm = <Hexadecimal format>
0000 to FFFF / Step 1

m or #Hm is enabled to specify data of a maximum of eight pages equivalent by using a comma (,) as a delimiter. Incidentally, the bit types to be set here are bit1 for LSB and bit16 for MSB. In other words, when BIT△32768 or BIT△#H8000 is set, 1 is only set for the bit16 of MSB. On the other hand, when BIT△1 or BIT△#H0001 is set, 1 is only set for the bit 1 of LSB.

Response Returns, in the format shown below, the number of pages set and the details of the bits from the page to a maximum of eight pages equivalent, leading to the maximum pattern setting bit.

PAG△******;BIT△#H****,#H****,#H****,#H****,#H****,#H****,#H****,#H****

Application example

Program

・Sets three pages equivalent of pattern bits from the page currently set.

```
> BIT△10,20,30
> BIT△#H1000,2000,3000
```

・Sets the number of pages as well as four pages equivalent of pattern bits from that page.

```
> PAG△10;BIT△10,20,30,40
> PAG△10;BIT△#H1000,#H2000,#H3000,#H4FFF
> ADR△10;BIT△10,20,30,40
> ADR△10;BIT△#H1000,#H2000,#H3000,#H4FFF
```

Query

・Makes a query when the header of the editing target is 1 and the maximum number of pages enabled to be obtained is 29.

```
>BIT?
<PAG△△△△△△1;BIT△#H0000,#H0000,#H0000,#H0000,#H0000,
  #H0000,#H0000,#H0000
```

Moreover, as the example below shows, patterns can be read to the end.

```
>BIT?
<PAG△△△△△△9;BIT△#H0000,#H0000,#H0000,#H0000,#H0000,
  #H0000,#H0000,#H0000
>BIT?
<PAG△△△△△17;BIT△#H0000,#H0000,#H0000,#H0000,#H0000,
  #H0000,#H0000,#H0000
>BIT?
<PAG△△△△△25;BIT△#H0000,#H0000,#H0000,#H0000,#H0000
```

The addresses of the patterns to be output increases in the equipment interior. When a different command is executed before all the patterns are output, the output of the patterns is terminated.

・Makes a query when the pattern bit is set at Zero-Subst.

```
> BIT?
< ERR
```

7.3.1.14. PDT Pattern Data (Pattern data)

Function Sets/queries the transmission pattern data in the program division from the <start> to the <end> address.

Header	Program	Query	Response (No. of characters)
PDT	PDT△ m0,m1,m2	PDT?△m0	PDT△m (as shown below)

Parameter m0 = <Hexadecimal format>
Address from which to start setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m1 = <Hexadecimal format>
Address at which to end setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m2 = <Hexadecimal or binary format>
#H***... Hexadecimal data
***... :data of 1 to 400 characters (equivalent to 400 × 4 bits) (0 to 9, A to F)
#B***... Binary data
***... :data of 1 to 400 characters (equivalent to 400 bits) (0, 1)

Bit number smaller than the range specified by <m1> and <m0> but larger than <m2>	Makes setting by repeating <m2> by the number of bits specified by <m1> and <m0>. ＜*Example*＞ ・<m0>=#H0, <m1>=#H1F, <m2>=#HABC Seting data:D5533DD5 (at LSB First) ABCABCAB (at MSB First) ・<m0>=#H0, <m1>=#H7, <m2>=#B011 Setting data : 10110110 (at LSB First) 01101101 (at MSB First)
Bit number larger than the range specified by <m1> and <m0> but smaller than <m2>	Truncates <m2> by the number of bits specified by <m1> and <m0>. ＜*Example*＞ ・<m0>=#H0, <m1>=#HF, <m2>=#HABCDEF Setting data : D5B3 (at LSB First) ABCD (at MSB First) ・<m0>=#H0, <m1>=#H3, <m2>=#B01100110 Setting data : 00000110 (at LSB First) 01100000 (at MSB First)

7.3.1.14 PDT Pattern Data (Pattern data) (continued)

Response

Bits of data from <m0> equal to or greater than 400 × 4 bits	***...: Bits of data from <m0>/4 characters
Bits of data from <m0> smaller than 400 × 4 bits	***...: Bits of data from <m0>/4 characters *<Example>* Data following the <m0> address - 010100 (Bin) Response:"#H28" (at LSB First hour(s)) "#H54" (at MSB First hour(s)) (When the last character comes less than 4 bits, the number of bits filled plus zeros inserted for the number of empty bits constitutes a response. The character in the section filled by no data fails to constitute a response.)

Application example

Program

・Sets the program pattern of Address (Hex) 0 to 1F at 0.

```
> PDT△#H0,#H1F,#H0
```

Query

・Queries the program pattern from Address (Hex) 0.

```
> PDT?△#H0
< #H000000.....
```

Note

When maximum pattern range (from #H0 to #H7FFFFF) is specified, it takes about 25 seconds until pattern loading.starts.

7.3.1.15. BDT Pattern Binary Data (Pattern binary data)

Function Sets/queries the transmission pattern data in the program division from the <start> to the <end> address.

Header	Program	Query	Response (No. of characters)
BDT	BDT△ m0,m1,m2	BDT?△m0	BDT△m (as shown below)

Parameter m0 = <Hexadecimal format>
Address from which to start setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m1 = <Hexadecimal format>
Address at which to end setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m2 = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<Z> X : Number of digits of YYY
YYY : Byte number of <Z> 1 to 16000 (bytes)
<Z> : Binary data up to a maximum of 16000 bytes

Bit number smaller than the range specified by <m1> and <m0> but larger than <Z>	Makes setting by repeating <Z> by the number of bits specified by <m1> and <m0>. ＜*Example*＞ <m0>=#H0, <m1>=#H1F, <m2>=#12AB Setting data:82428242 (at LSB First) 41424142 (at MSB Firs)
Bit number larger than the range specified by <m1> and <m0> but smaller than <Z>	Truncates <Z> by the number of bits specified by <m1> and <m0>. ＜*Expamle*＞ <m0>=#H0, <m1>=#HF, <m2>=#13ABC Setting data:8242 (at LSR First) 4142 (at MSB First)

Response

Bits of data from <m0> equal to or greater than 16000×4 bits	***...:16000 characters (XXX x 4 bits)
Bits of data from <m0> smaller than 16000×4 bits	***...:Bits of data from <m0>/4 characters ＜*Exapmle*＞ Data following the <m0> address - 010101 (Bin) Response:"#11(" (at LSB First) "#11T" (at MSB First) (When the last character comes less than 4 bits, the number of bits filled plus zeros inserted for the number of empty bits constitutes a response. The character in the section filled by no data fails to constitute a response.)

7.3.1.15. BDT Pattern Binary Data (Pattern binary data) (continued)

Application example

Program

・Sets the program pattern of Address (Hex) 0 to 1F at #H41414141.

```
> BDT△#H0,#H1F,#11A
```

Query

・Queries the program pattern from Address (Hex) 0.

```
> BDT?△#H0
< #H000000.....
```

Note

When maximum pattern range (from #H0 to #H7FFFFF) is specified, it takes about 25 seconds until pattern loading.starts.

7.3.1.16. DRA Reversion of Transmission Pattern Specification Address Data (Data reverse address)

Function Reverses the data of the transmission pattern specification address. (Address specification)

Header	Program	Query	Response (No. of characters)
DRA	DRA△m1,m2	None	None

Parameter m1 = <Hexadecimal format>
#H0 to #H7FFFFF Start Address(Hex): 0 to 7FFFFFbit/Step:1
m2 = <Hexadecimal format>
#H0 to #H7FFFFF End Address(Hex): 0 to 7FFFFFbit/Step:1

Application example

Reverses the program patterns from Address (Hex) 0 to FF.

```
> DRA△#H0,#HFF
```

7.3.1.17. DRD Reversion of Transmission Pattern Specification Address Data (Data reverse address delta)

Function Reverses the transmission pattern data in the program division. (Delta specification)

Header	Program	Query	Response (No. of characters)
DRD	DRD△m1,m2	None	None

Parameter m1 = <Hexadecimal format>
#H0 to #H7FFFFF Start Address(Hex): 0 to 7FFFFFbit/Step:1
m2 = <NR1>
1 to 8388608 delta :1 to 8388608 bit/Step:1

Application example

Reverses 256 bits equivalent of the program patterns from Address (Hex) 0.

```
> DRD△#H0,256
```

7.3.1.18. WRT Pattern Data Input Byte Number (Pattern data write)

Function Sets the pattern data by the specified number of blocks from the specified address.

Header	Program	Query	Response (No. of characters)
WRT	WRT△m1,m2	None	None

Parameter m1 = <NR1>

1 to 400 / Step1 Number of bytes of pattern transfer

m2 = <NR1>

0 to 524287 / Step1 Head address of pattern input

Application example

Specifies 18 bytes to set data from the 1st to the 9th page since a page consists of 16 bits.

```
> WRT△18,0
> ABCDEFGHIJKLMNOPQR
```

In this case, patterns are stored into memory as shown below.

```
Address  : Pattern
   0     : 0100 0010 1000 0010 0010 0010 1100 0010
   4     : 0110 0010 1010 0010 0001 0010 1110 0010
   8     : 0101 0010 1001 0010 0011 0010 1101 0010
   C     : 0111 0010 1011 0010 0000 1010 1111 0010
   10    : 0100 1010 1000 1010
```

7.3.1.19. RED Pattern Data Output Byte Number (Pattern data read?)

Function Reads out the pattern data by the specified number of blocks from the specified address.

Header	Program	Query	Response (No. of characters)
RED	None	RED△m1,m2	Data pattern sequence (as shown below)

Parameter m1 = <NR1>

1 to 400 / Step1 Number of bytes of pattern transfer

m2 = <NR1>

0 to 524287 / Step1 Head address of pattern output

Application example

When reading out data from the 1st to the 10th page.

```
> RED△20,0
< ABCDEFGHIJKLMNOPQRST
```

7.3.1.20. ALL ALL Preset (All bits of all pages) (Preset all 0 or 1)

Function Sets all the transmission patterns either 1 or 0.

Header	Program	Query	Response (No. of characters)
ALL	ALL△m	None	None

Parameter m = <NR1>

0 : Clears all bits on all pages.

1 : Sets all bits on all pages.

Application example

Sets all bits to 1.

```
> ALL△1
```

7.3.1.21. PST Preset (All bits on a page) (Preset page 0 or 1)

Function Sets all the transmission patterns on pages set by PAG/ADR either 1 or 0.

Header	Program	Query	Response (No. of characters)
PST	PST△m	None	None

Parameter m = <NR1>

0 : Clears all bits on a page.

1 : Sets all bits on a page.

Application example

Sets all bits on a page to 1.

```
> PST△1
```

7.3.1.22. EAD Error Insertion (Error addition mode)

Function Switches/queries the error insertion state.

Header	Program	Query	Response (No. of characters)
EAD	EAD△m	EAD?	EAD△m (FIX1)

Parameter m = <NR1>

0 : Error insertion OFF
1 : 1E-4
2 : 1E-5
3 : 1E-6
4 : 1E-7
5 : 1E-8
6 : 1E-9
7 : Single
8 : 1E-3
9 : External

Application example

Program

・Inserts an error at the rate of 1E-3

```
> EAD△8
```

Query

・Make a query when error insertion is OFF.

```
> EAD?
< EAD△0
```

7.3.1.23. ECH Error Insertion Route (Error addition channel)

Function Sets/queries the route in which an error is inserted.

Header	Program	Query	Response (No. of characters)
ECH	ECH△m	ECH?	ECH△m (FIX1)

Parameter m = <NR1>

1 : Route1
2 : Route2
3 : Route3
4 : Route4
5 : Route5
6 : Route6
7 : Route7
8 : Route8
9 : Route1
10 : Route2
11 : Route3
12 : Route4
13 : Route5
14 : Route6
15 : Route7
16 : Route8

Application example

Program

・Sets the parameter at Route1.

```
> ECH△1
```

Query

・Makes a query when the parameter is set at Route2.

```
> ECH?
< ECH△2
```

7.3.1.24. PML Level of PRBS 1 (PRBS Mark Logic)

Function Sets/queries whether the PRBS 1 is High level or Low level.

Header	Program	Query	Response (No. of characters)
PML	PML△m	PML?	PML△m (FIX1)

Parameter m = <NR1>

0 Sets PRBS Logic to Mark High.

1 Sets PRBS Logic to Mark Low.

Application example

Program

・Sets the PRBS Logic at Mark Low.

```
> PML△1
```

Query

・Makes a query when the PRBS Logic is set at Mark High.

```
> PML?
< PML△0
```

7.3.1.25. GML Level of PRGM 1 (PRGM Mark Logic)

Function Sets/queries whether the PRGM 1 is High level or Low level.

Header	Program	Query	Response (No. of characters)
GML	GML△m	GML?	GML△m (FIX1)

Parameter m = <NR1>

```
0：Sets PRGM Logic to Mark High.
1：Sets PRGM Logic to Mark Low.
```

Application example

Program

・Sets PRGM Logic to Mark Low.

```
> GML△1
```

Query

・Makes a query when the PRGM Logic is set at Mark High.

```
> GML?
< GML△0
```

7.3.2. Clock/Data Output Related

Sets/queries the transmit clock and the data.

Note 1) In regard to parameter setting, if a parameter is set in a step other than the pre-set steps, the smallest pre-set value which comes closest to the value specified will be set. In other words, any number which fails to reach a full unit will be truncated.

Note 2) An execution error takes place when a PG unit is not in the slot constituting the setting target.

7.3.2.1. OON On/Off of Output (Output On / Off)

Function Sets/queries ON/OFF of clock output and data output.

When this command sets output OFF, the output of clock and data signals will be OFF. However, when the command sets output ON, signals will be output in accordance with the output ON/OFF state at each clock and data port. In other words, signals are output only from the port where the output ON/OFF setting of both the command and the port is ON.

Header	Program	Query	Response (No. of characters)
OON	OON△m	OON?	OON△m (FIX1)

Parameter m = <NR1>
0 : Output OFF
1 : Output ON

Application example

Program

・Sets the parameter at output ON.

```
> OON△1
```

Query

・Makes a query when the parameter is set at output OFF.

```
> OON?
< OON△0
```

・Makes a query when no PPG unit is located in the slot constituting the setting target.

```
> OON?
< ERR
```

7.3.2.2. CON ON/OFF of Clock Output (Clock Output On/Off)

Function Sets/queries ON/OFF of clock output.

Header	Program	Query	Response (No. of characters)
CON	CON△m	CON?	CON△m (FIX1)

Parameter m = <NR1>
0 : Output OFF
1 : Output ON

Application example

Program

・Sets the parameter at output ON.

```
> CON△1
```

Query

・Makes a query when the parameter is set at output OFF.

```
> CON?
< CON△0
```

・Makes a query when no PPG unit is located in the slot constituting the setting target.

```
> CON?
< ERR
```

7.3.2.3. XCO ON/OFF of XClock Output (XClock Output On/Off)

Function Sets/queries ON/OFF of XClock output.

Header	Program	Query	Response (No. of characters)
XCO	XCO△m	XCO?	XCO△m (FIX1)

Parameter m = <NR1>
0 : Output OFF
1 : Output ON

Application example

Program

・Sets the parameter at output ON.

```
> XCO△1
```

Query

・Makes a query when the parameter is set at output OFF.

```
> XCO?
< XCO△0
```

・Makes a query when no PPG unit is located in the slot constituting the setting target.

```
> XCO?
< ERR
```

7.3.2.4. CDL Setting of Clock Output Delay (Clock delay)

Function Sets/queries the clock output delay.

Header	Program	Query	Response (No. of characters)
CDL	CDL△m	CDL?	CDL△m (FIX5)

Parameter m = <NR1>
-1000 to 1000 -1000 to 1000 ps/Step:2ps

Application example

Program

・Sets the clock output delay at 500 ps.
> CDL△500

Query

・Makes a query when the clock output delay is set at 500 ps.
> CDL?
< CDL△△-500

・Makes a query when no PPG unit is located in the slot constituting the setting target.
> CDL?
< ERR

7.3.2.5. CLL Clock Output Level (Clock Level)

Function Sets/queries the clock output level.

Header	Program	Query	Response (No. of characters)
CLL	CLL△m	CLL?	CLL△m (FIX1)

Parameter m = <NR1>
0 : VAR level
1 : ECL level

Application example

Program

・Sets the clock output level at VAR.
> CLL△0

Query

・Makes a query when the clock output level is set at ECL.
> CLL?
< CLL△1

7.3.2.6. XCL XClock Output Level (XClock Level)

Function Sets/queries the XClock output level.

Header	Program	Query	Response (No. of characters)
XCL	XCL△m	XCL?	XCL△m (FIX1)

Parameter m = <NR1>
0 : VAR level
1 : ECL level

Application example

Program

・Sets the XClock output level at VAR.
> XCL△0

Query

・Makes a query when the XClock output level is set at ECL.
> XCL?
< XCL△1

7.3.2.7. CAP Clock Output Amplitude (Clock Amplitude)

Function Sets/queries the Clock output amplitude.

Header	Program	Query	Response (No. of characters)
CAP	CAP△m	CAP?	CAP△m (FIX5)

Parameter m = <NR2>
0.50 to 2.00 0.50 to 2.00Vpp/Step:0.01

Application example

Program

・Sets the Clock output amplitude at 1.00Vpp.
> CAP△1.0

Query

・Makes a query when the Clock output amplitude is set at 2.00Vpp.
> CAP?
< CAP△2.000

・Makes a query when the Level is ECL.
> CAP?
< ERR

7.3.2.8. XCA XClock Output Amplitude (XClock Amplitude)

Function Sets/queries the XClock output amplitude.

Header	Program	Query	Response (No. of characters)
XCA	XCA△m	XCA?	XCA△m (FIX5)

Parameter m = <NR2>

0.50 to 2.00 0.50 to 2.00Vpp/Step:0.01

Application example

Program

・Sets the XClock output amplitude at 1.00Vpp.

```
> XCA△1.0
```

Query

・Makes a query when the XClock output amplitude is set at 0.5Vpp.

```
> XCA?
< XCA△0.500
```

・Makes a query when the Level is ECL.

```
> XCA?
< ERR
```

7.3.2.9. COS Clock Output Offset (Clock offset)

Function Sets/queries the Clock output offset.

Header	Program	Query	Response (No. of characters)
COS	COS△m	COS?	COS△m（FIX6)

Parameter m = <NR2>

-4.000 to 2.000 -4.000 to 2.000V/Step:0.005

Application example

Program

・Sets Clock offset at 1.0V.

```
> COS△1.00
```

Query

・Makes a query when the Clock offset is 2.0V.

```
> COS?
< COS△△2.000
```

・Makes a query when the Level is ECL.

```
> COS?
< ERR
```

7.3.2.10. XCF XClock Output Offset (XClock offset)

Function Sets/queries the XClock output offset.

Header	Program	Query	Response (No. of characters)
XCF	XCF△m	XCF?	XCF△m (FIX6)

Parameter m = <NR2>
-4.000 to 2.000 -4.000 to 2.000V/Step:0.005

Application example

Program

・Sets XClock offset at 1.0V.

```
> XCF△1.00
```

Query

・Makes a query when the Clock offset is -2.0V.

```
> XCF?
< XCF△-2.000
```

・Makes a query when the Level is ECL.

```
> XCF?
< ERR
```

7.3.2.11. OPS Selection of Port for Offset Reference Value Setting (Offset mode port select)

Function Selects/queries the port constituting the target of setting the offset reference value.
The initial value effective before this command is executed is the "Data port."

Header	Program	Query	Response (No. of characters)
OPS	OPS△m	OPS?	OPS△m (FIX1)

Parameter m = <NR1>
0 : Data port
1 : XData port
2 : Clock port
3 : XClock port

Application example

Program

・Sets the port where to set the offset reference value for Clock at Clock.

```
> OPS△2
```

Query

・Makes a query when XData is the port where to set the offset reference value.

```
> OPS?
< OPS△1
```

7.3.2.12. OFS Offset Reference Value (Offset mode)

Function Sets/queries the offset reference value for the port specified by OPS. The initial value effective before OPS is executed is the "Data port."

Header	Program	Query	Response (No. of characters)
OFS	OFS△m	OFS?	OFS△m (FIX1)

Parameter m = <NR1>
0 : Voh
1 : Vth
2 : Vol

Application example

Program

・Sets the offset reference value at Voh.

```
> OFS△0
```

Query

・Makes a query when the offset reference value is Vth.

```
> OFS?
< OFS△1
```

・Makes a query when the Level is ECL.

```
> OFS?
< ERR
```

7.3.2.13. CDT Setting of Clock Duty (Clock duty)

Function Sets/queries the duty of the Clock signal.

Header	Program	Query	Response (No. of characters)
CDT	CDT△m	CDT?	CDT△m (FIX3)

Parameter m = <NR1>
-25 to 25 -25 to 25 / Step:1

Application example

Program

・Sets the duty at 20.

```
> CDT△20
```

Query

・Makes a query when the duty is -5.

```
> CDT?
< CDT△△-5
```

7.3.2.14. XDT Setting of XClock Duty (XClock duty)

Function Sets/queries the duty of the XClock signal.

Header	Program	Query	Response (No. of characters)
XDT	XDT△m	XDT?	XDT△m (FIX3)

Parameter m = <DECIMAL NUMERIC PROGRAM DATA>
-25 to 25 25 to 25 / Step:1

Application example

Program

・Sets the duty at 0.

```
> XDT△0
```

Query

・Makes a query when the duty is 5.

```
> XDT?
< XDT△△△5
```

7.3.2.15. CIM setting of Clock DC impedance (Clock DC impedance)

Function Sets/queries the DC impedance of the Clock/XClock signal.

Header	Program	Query	Response (No. of characters)
CIM	CIM△m	CIM?	CIM△m (FIX1)

Parameter m = <NR1>
0 : Sets DC impedance at 0ohm
1 : Sets DC impedance at 50ohm

Application example

Program

・Sets DC Impedance at 0ohm.

```
> CIM△0
```

Query

・Makes a query when the DC impedance is set at 50ohm.

```
> CIM?
< CIM△1
```

7.3.2.16. DON ON/OFF of Data Output (Data Output On/Off)

Function Sets/queries ON/OFF of Data Output.

Header	Program	Query	Response (No. of characters)
DON	DON△m	DON?	DON△m (FIX1)

Parameter m = <NR1>
0 : Output OFF
1 : Output ON

Application example

Program

・Sets the parameter at output ON.

```
> DON△1
```

Query

・Makes a query when the parameter is set at output OFF.

```
> DON?
< DON△0
```

・Makes a query when no PPG unit is located in the slot constituting the setting target.

```
> DON?
< ERR
```

7.3.2.17. XDO ON/OFF of XData Output (XData Output On/Off)

Function Sets/queries ON/OFF of XData Output.

Header	Program	Query	Response (No. of characters)
XDO	XDO△m	XDO?	XDO△m (FIX1)

Parameter m = <NR1>
0 : Output OFF
1 : Output ON

Application example

Program

・Sets the parameter at output ON.

```
> XDO△1
```

Query

・Makes a query when the parameter is set at output OFF.

```
> XDO?
< XDO△0
```

・Makes a query when no PPG unit is located in the slot constituting the setting target.

```
> XDO?
< ERR
```

7.3.2.18. DIM Setting of Data DC impedance (Data DC impedance)

Function Sets/queries the DC impedance of the Data/XData signal.

Header	Program	Query	Response (No. of characters)
DIM	DIM△m	DIM?	DIM△m (FIX1)

Parameter m = <NR1>

0 : Sets DC impedance at 0ohm

1 : Sets DC impedance at 50ohm

Application example

Program

・Sets DC Impedance at 0ohm.

```
> DIM△0
```

Query

・Makes a query when the DC Impedance is set at 50ohm

```
> DIM?
< DIM△1
```

7.3.2.19. DCR Setting of Data cross point (Data cross point)

Function Sets/queries the cross point of the Data signal.

Header	Program	Query	Response (No. of characters)
DCR	DCR△m	DCR?	DCR△m (FIX2)

Parameter m = <NR1>

25 to 75 25% to 75% / Step:1%

Application example

Program

・Sets the Cross Point at 55%.

```
> DCR△55
```

Query

・Makes a query when the Cross Point is 45%.

```
> DCR?
< DCR△45
```

7.3.2.20. XDC Setting of XData cross point (XData cross point)

Function Sets/queries the cross point of the XData signal.

Header	Program	Query	Response (No. of characters)
XDC	XDC△m	XDC?	XDC△m (FIX2)

Parameter m = <NR1>
25 to 75 25% to 75%/Step:1%

Application example

Program

・Sets the Cross Point at 55%.

```
> XDC△55
```

Query

・Makes a query when the Cross Point is 45%.

```
> XDC?
< XDC△45
```

7.3.2.21. DAL Data Output level (Data Level)

Function Sets/queries the Data output level.

Header	Program	Query	Response (No. of characters)
DAL	DAL△m	DAL?	DAL△m (FIX1)

Parameter m = <DECIMAL NUMERIC PROGRAM DATA>
0 : VAR level
1 : ECL level

Application example

Program

・Sets the Data output level at VAR.

```
> DAL△0
```

Query

・Makes a query when the Data output level is set at ECL.

```
> DAL?
< DAL△1
```

7.3.2.22. XDL XData Output level (XData Level)

Function Sets/queries the XData output level.

Header	Program	Query	Response (No. of characters)
XDL	XDL△m	XDL?	XDL△m (FIX1)

Parameter m = <NR1>
0 : VAR level
1 : ECL level

Application example

Program

・Sets the XData output level at VAR.

```
> XDL△0
```

Query

・Makes a query when the XData output level is set at ECL.

```
> XDL?
< XDL△1
```

7.3.2.23. DAP Data Output Amplitude (Data amplitude)

Function Sets/queries the Data output amplitude.

Header	Program	Query	Response (No. of characters)
DAP	DAP△m	DAP?	DAP△m (FIX5)

Parameter m = <NR2>
to 2.00 0.50 to 2.00Vpp/Step:0.01

Application example

Program

・Sets the Data output amplitude at 0.5Vpp.

```
> DAP△0.5
```

Query

・Makes a query when the Data output amplitude is set at 0.5Vpp.

```
> DAP?
< DAP△0.500
```

・Makes a query when the Level is ECL.

```
> DAP?
< ERR
```

7.3.2.24. NAP XData Output Amplitude (Invert data Amplitude)

Function Sets/queries the XData output amplitude.

Header	Program	Query	Response (No. of characters)
NAP	NAP△m	NAP?	NAP△m (FIX5)

Parameter m = <NR2>

0.50 to 2.00 0.50 to 2.00Vpp/Step:0.01

Application example

Program

・Sets the XData output amplitude at 1.00Vpp.

```
> NAP△1.0
```

Query

・Makes a query when the XData output amplitude is set at 1.5Vpp.

```
> NAP?
< NAP△1.500
```

・Makes a query when the Level is ECL.

```
> NAP?
< ERR
```

7.3.2.25. DOS Data Output Offset (Data offset)

Function Sets/queries the Data output offset.

Header	Program	Query	Response (No. of characters)
DOS	DOS△m	DOS?	DOS△m (FIX6)

Parameter m = <NR2>

-4.000 to 2.000 -4.000 to 2.000V/Step:0.005

Application example

P*rogram*

・Sets Data offset at 1.0V.

```
> DOS△1.0
```

Query

・Makes a query when the Data offset is 2.0V.

```
> DOS?
< DOS△△2.000
```

・Makes a query when the Level is ECL.

```
> DOS?
< ERR
```

7.3.2.26. NOS XData Output Offset (Invert data offset)

Function Sets/queries the XData output offset.

Header	Program	Query	Response (No. of characters)
NOS	NOS△m	NOS?	NOS△m (FIX6)

Parameter m = <NR2>
-4.000 to 2.000 -4.000 to 2.000V/Step:0.005

Application example

Program

・Sets XData offset at -1.0V.

```
> NOS△-1.00
```

Query

・Makes a query when the Data offset is -0.5V.

```
> NOS?
< NOS△-0.500
```

・Makes a query when the Level is ECL.

```
> NOS?
< ERR
```

7.3.2.27. SOP Selection of Sync Output Signal (Sync Output)

Function Sets/queries the type of Sync Output signal.

Header	Program	Query	Response (No. of characters)
SOP	SOP△m	SOP?	SOP△m (FIX1)

Parameter m = <NR1>
0 : 1/8 Clock Sync
1 : Pattern Sync

Application example

Program

・Sets Sync Output at 1/8 Clock Sync.

```
> SOP△0
```

Query

・Makes a query when Sync Output is Pattern Sync.

```
> SOP?
< SOP△1
```

7.3.2.28. PSP Selection of Pattern Sync Output Position (Pattern sync position)

Function Sets/queries the output position when Sync Output is Pattern.

Header	Program	Query	Response (No. of characters)
PSP	PSP△m	PSP?	PSP△m (FIX10)

Parameter m = <NR1>

1 to 121	1 to 121 / 8 (Zero-Subst 2^7)
1 to 505	1 to 505 / 8 (Zero-Subst 2^9)
1 to 2,041	1 to 2,041 / 8 (Zero-Subst 2^11)
1 to 32,761	1 to 32,761 / 8 (Zero-Subst 2^15)
1 to 127	1 to 127 / 1 (PRBS7)
1 to 511	1 to 511 / 1 (PRBS9)
1 to 2,047	1 to 2,047 / 1 (PRBS11)
1 to 32,767	1 to 32,767 / 1 (PRBS15)
1 to 1,048,575	1 to 1,048,575 / 1 (PRBS20)
1 to 8,388,767	1 to 8,388,607 / 1 (PRBS23)
1 to 2,147,483,647	1 to 2,147,483,647 / 1 (PRBS31)
1 to Pattern Length	1 to Pattern Length / 8 (PRGM)

The maximum setting data at Pattern = PRGM becomes the maximum value below the Pattern Length value calculated as below.

Pattern Length ≧ The maximum setting data value ＝ 1+8×n

(n : integer)

Example : At Pattern Length = 127, the maximum value becomes 121.

Application example

Program

・Sets the Position of Pattern Sync at 129.

```
> PSP△129
```

Query

・Makes a query when the Position of Pattern Sync is 500.

```
> PSP?
< PSP△△△△△△△△500
```

7.3.3. Alarm Queries

Queries the alarm output state (such as PPG Clock Loss).

7.3.3.1. PCL? PPG Clock Loss (PPG clock loss)

Function Queries the presence of PPG Clock Loss.

Header	Program	Query	Response (No. of characters)
None	None	PCL?	PCL△m (FIX1)

Parameter m = <NR1>
0 : PPG Clock Loss not detected.
1 : PPG Clock Loss detected

Application example

Query

・Makes a query when the clock signal is supplied to PPG.

```
> PCL?
< PCL△0
```

7.3.3.2. DTR? Delay Trouble3 (Delay trouble 3)

Function Queries the presence of the Delay trouble.

Header	Program	Query	Response (No. of characters)
None	None	DTR?	DTR△m (FIX1)

Parameter m = <NR1>
0 : PPG delay trouble not detected.
1 : PPG delay trouble detected.

Application example

Query

・Makes a query when a delay trouble is encountered at PPG.

```
> DTR?
< DTR△1
```

7.3.3.3. DLY Delay Setting State (Delay unlock?)

Function Queries the delay setting state of PPG.

Header	Program	Query	Response (No. of characters)
DLY	None	DLY?	DLY△m (FIX1)

Parameter m = <NR1>
0 : Ready
1 : Delay Busy

Application example

Query

・Queries the delay setting state.

```
> DLY?
< DLY△0
```

7.3.4. Grouping Related

This section describes setting/query on Grouping.

7.3.4.1. CGR Setting of Clock/XClock Grouping (Clock/XClock grouping)

Function Sets/queries ON/OFF of Clock/XClock grouping.

Header	Program	Query	Response (No. of characters)
CGR	CGR△m	CGR?	CGR△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets grouping ON.

```
> CGR△1
```

Query

・Makes a query when grouping is OFF.

```
> CGR?
< CGR△0
```

7.3.4.2. TRK Setting of Data/XData Grouping (Data/XData tracking)

Function Sets/queries ON/OFF of Data/XData grouping.

Header	Program	Query	Response (No. of characters)
TRK	TRK△m	TRK?	TRK△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets grouping ON.

```
> TRK△1
```

Query

・Makes a query when grouping is OFF.

```
> TRK?
< TRK△0
```

7.3.4.3. PCO PPG/ED Pattern Common Setting (PPG/ED pattern common setting)

Function Sets/queries ON/OFF of PPG/ED pattern common setting.

Header	Program	Query	Response (No. of characters)
PCO	PCO△m	PCO?	PCO△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets the common setting ON.

```
> PCO△1
```

Query

・Makes a query when the common setting is OFF.

```
> PCO?
```

7.4. 3.2G Setting of 3.2G Internal Synthesizer Option (OPT03)

This section describes the details of the setting/query commands concerning the 3.2G internal synthesizer.

Note 1) For unstipulated commands and parameters, corresponding errors take place.

Note 2) A command error takes place when an intrinsic command is sent in a state where the 3.2G internal synthesizer is not in the measuring instrument.

7.4.1. Sending Signal Related

Makes settings and queries concerning the sending signals of the 3.2G internal synthesizer.

7.4.1.1. FRQ Setting of Clock Frequency (Frequency)

Function Sets/queries the clock input phase.

Header	Program	Query	Response (No. of characters)
FRQ	FRQ△m	FRQ?	FRQ△m (FIX8)

Parameter m = <NR1>
50000 to 3200000 50000 to 3200000kHz/Step:1

Application example

Program

・Sets the clock frequency at 3200000 kHz.

```
> FRQ△3200000
```

Query

・Makes a query when the clock frequency is 50000.

```
> FRQ?
< FRQ△△△50000
```

7.4.1.2. RFC Clock Reference Signal (Reference clock)

Function Sets/queries the reference signal.

Header	Program	Query	Response (No. of characters)
RFC	RFC△m	RFC?	RFC△m (FIX1)

Parameter m = <NR1>
0 : INT
1 : EXT 10MHz

Application example

Program

・Sets the reference signal at INT.

```
> RFC△0
```

Query

・Makes a query when the reference signal is EXT 10 MHz.

```
> RFC?
< RFC△1
```

7.4.1.3. PLL PLL Lock State (PLL unlock)

Function Queries the PLL lock state.

Header	Program	Query	Response (No. of characters)
PLL	None	PLL?	PLL△m (FIX1)

Parameter m = <NR1>

0 : Lock state

1 : Unlock state

Application example

Query

・Makes a query when the parameter is PLL Unlock.

```
> PLL?
< PLL△1
```

7.5. Setting of 3.2G ED (MU163240C)

This section describes the details of the setting/query commands concerning 3.2G ED.

Note 1) For unstipulated commands and parameters, corresponding errors take place.

Note 2) A command error takes place when an intrinsic command of this instrument is sent in a state where the 3.2G ED unit is not in the measuring instrument.

Note 3) The space is expressed by a △.

7.5.1. Reception Pattern Related

Makes settings and queries concerning the reception patterns.

7.5.1.1. PTI Pattern Reception Mode (Pattern input mode)

Function Sets/queries the method of outputting the reception patterns.

Header	Program	Query	Response (No. of characters)
PTI	PTI△m	PTI?	PTI△m (FIX1)

Parameter m = <NR1>
0 : Repeat mode
1 : Burst mode

Application example

Program

・Sets the reception pattern at the repeat mode.

```
> PTI△0
```

Query

・Queries the method of outputting the reception pattern when the reception pattern is in the burst mode.

```
> PTI?
< PTI△1
```

7.5.1.2. PTN Reception Pattern (Pattern mode)

Function Sets/queries the type of reception pattern.

Header	Program	Query	Response (No. of characters)
PTN	PTN△m	PTN?	PTN△m (FIX2)

Parameter m = <NR1>

1 : PRGM
2 : PRBS7
3 : PRBS9
5 : PRBS11
6 : PRBS15
7 : PRBS20
8 : PRBS23
9 : PRBS31
10 : Zero-Subst

Application example

Program

・Sets the reception pattern at PRGM.

```
> PTN△1
```

Query

・Queries the reception pattern when the parameter is set at PRBS23.

```
>PTN?
< PTN△△8
```

7.5.1.3. MRK Reception Pattern Mark Ratio (Mark ratio mode)

Function Sets/queries the mark ratio when the reception pattern is set at PRBS.

Header	Program	Query	Response (No. of characters)
MRK	MRK△m	MRK?	MRK△m (FIX1)

Parameter m = <NR1>
0 : 0/8 or 8/8
1 : 1/8 or 7/8
2 : 1/4 or 3/4
3 : 1/2 or 1/2INVT

Application example

Program

・Sets the mark ratio at 1/8 or 7/8.

```
> MRK△1
```

Query

・Makes a query when the mark ratio is set at either 3/4 or 1/4

```
> MRK?
< MRK△2
```

・Making a query when the PRGM pattern is set.

```
> MRK?
< ERR
```

Remarks On MP1650A and MP1651A, 0/8 and 8/8, 1/8 and 7/8, 1/4 and 3/4, and 1/2 and 1/2INVT had the same set value depending on the POS/NEG state of Logic. However, MP1632C has no Logic POS/NES setting of the PRBS pattern. Accordingly, to establish compatibility with the commands of MP1650A and MP1651A, a pseudo Logic setting was added to PRBS. The table below shows the relations between Logic and the Mark ratio.

Logic	Mark ratio			
POS	0/8	1/8	1/4	1/2
NEG	8/8	7/8	3/4	1/2INVT

7.5.1.4. LGC Reception Pattern Logic (Logic mode)

Function Sets/queries the reception pattern logic.

Header	Program	Query	Response (No. of characters)
LGC	LGC△m	LGC?	LGC△m (FIX1)

Parameter m = <NR1>

0 : POS(PRGM、at Zero-Subst) 0/8,1/8,1/4,1/2(at PRBS)

1 : NEG(PRGM、at Zero-Subst) 8/8,7/8,3/4,1/2INVT(at PRBS)

Application example

Program

・Sets logic at POS when the reception pattern logic is set at PRBS.

```
> LGC△0
```

・Sets the mark ratio at 3/4 when the reception pattern logic is set at PRBS and the mark ratio is 1/4.

```
> LGC△1
```

Query

・Queries logic when the reception pattern logic is set at PRBS and the mark ratio is 1/8.

```
> LGC?
< LGC△0
```

・Queries logic when the reception pattern logic is set NEG at Zero-Subst.

```
> LGC?
< LGC△1
```

7.5.1.5. SFT Bit Shift (Mark ratio bit shift)

Function Sets/queries the number of bit shifts of the PRBS pattern mark ratio.

Header	Program	Query	Response (No. of characters)
SFT	SFT△m	SFT?	SFT△m (FIX1)

Parameter m = <NR1>

0 : 1 bit shift

1 : 3 bit shift

Application example

Program

・Sets the bit shift at 1 bit.

```
> SFT△0
```

Query

・Makes a query when the parameter is set at 3 bit shift.

```
> SFT?
< SFT△1
```

・Queries logic when the bit shift is NEG at Zero-Subst.

```
> SFT?
< ERR
```

7.5.1.6. ZPL Zero-Subst Pattern Length (Zero-Subst Pattern Length)

Function Sets/queries the length of the Zero-Subst pattern.

Header	Program	Query	Response (No. of characters)
ZPL	ZPL△m	ZPL?	ZPL△m (FIX1)

Parameter m = <NR1>

2 : 2^7
3 : 2^9
5 : 2^11
6 : 2^15

Application example

Program

・Sets the pattern length at 2 ^ 15.

```
> ZPL△6
```

Query

・Makes a query when the pattern length is 2 ^ 9.

```
> ZPL?
< ZPL△3
```

・Makes a query when the pattern length is set at PRBS.

```
> ZPL?
< ERR
```

7.5.1.7. ZLN Length of Consecutive Zeros of Zero-Subst Pattern (Zero Length)

Function Sets/queries the length of consecutive zeros of the Zero-Subst pattern.

Header	Program	Query	Response (No. of characters)
ZLN	ZLN△m	ZLN?	ZLN△m (FIX5)

Parameter m = <NR1>

Sets the length of consecutive zeros within the range shown below.

7 to 127 / step 1 : 2^7
9 to 511 / step 1 : 2^9
11 to 2047 / step 1 : 2^11
15 to 32767 / step 1 : 2^15

Application example

Program

・Sets the length of consecutive zeros at 98 bits.

```
> ZLN△98
```

Query

・Makes a query when the length of consecutive zeros is 423 bits.

```
> ZLN?
< ZLN△△△423
```

・Makes a query when the length of consecutive zeros is set at PRBS.

```
> ZLN?
< ERR
```

7.5.1.8. DLN PRGM Pattern Length (PRGM data length)

Function Sets/queries the PRGM pattern length.

Header	Program	Query	Response (No. of characters)
DLN	DLN△m	DLN?	DLN△m (FIX7)

Parameter m = <NR1>

Sets the pattern length within the range shown below.

2 to 131072 : step 1
131072 to 262144 : step 2
262144 to 524288 : step 4
524288 to 1048576 : step 8
1048576 to 2097152 : step 16
2097152 to 4194304 : step 32
4194304 to 8388608 : step 64

Application example

Program

・Sets the pattern length at 1048576.

```
> DLN△1048576
```

Query

・Makes a query when the pattern length is 514 bits.

```
> DLN?
< DLN△△△△△514
```

・Makes a query when the pattern length is set at PRBS.

```
> DLN?
< ERR
```

7.5.1.9. SYN Automatic Synchronization Function (Auto sync)

Function Sets/queries ON/OFF of the automatic synchronization function.

Header	Program	Query	Response (No. of characters)
SYN	SYN△m	SYN?	SYN△m (FIX1)

Parameter m = <NR1>

0 : OFF
1 : ON

Application example

Program

・Sets the automatic synchronization function ON.

```
> SYN△1
```

Query

・Makes a query when the automatic synchronization function is OFF.

```
> SYN?
< SYN△0
```

7.5.1.10. LTH Synchronization Loss Threshold (Sync loss threshold)

Function Sets/queries the threshold of synchronization loss.

Header	Program	Query	Response (No. of characters)
LTH	LTH△m	LTH?	LTH△m (FIX1)

Parameter m = <NR1>

0 : E-2
1 : E-3
2 : E-4
3 : E-5
4 : E-6
5 : E-7

Application example

Program

・Sets the threshold of synchronization loss at E-3.

```
> LTH△1
```

Query

・Makes a query when the threshold of synchronization loss is E-6.

```
> LTH?
< LTH△4
```

・Makes a query when the Internal Threshold setting is ON.

```
> LTH?
< ERR
```

7.5.1.11. GTH Synchronization Gain Threshold (Sync gain threshold)

Function Sets/queries the threshold of synchronization gain.

Header	Program	Query	Response (No. of characters)
GTH	GTH△m	GTH?	GTH△m (FIX1)

Parameter m = <NR1>
0 : E-3
1 : E-4
2 : E-5
3 : E-6
4 : E-7
5 : E-8

Application example

Program

・Sets the threshold of synchronization gain at E-3.

```
> GTH△1
```

Query

・Makes a query when the threshold of synchronization gain is E-6.

```
> GTH?
< GTH△4
```

・Makes a query when the Internal Threshold setting is ON.

```
> GTH?
< ERR
```

7.5.1.12. ITH Internal Synchronization Threshold (Internal sync threshold)

Function Sets/queries the state of internally stipulated synchronization threshold.

Header	Program	Query	Response (No. of characters)
ITH	ITH△m	ITH?	ITH△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets the Internal Threshold ON.

```
> ITH△1
```

Query

・Makes a query when the Internal Threshold is OFF.

```
> ITH?
< ITH△0
```

7.5.1.13. FSY Frame Synchronization Function (Frame sync)

Function Sets/queries ON/OFF of the frame synchronization function.

Header	Program	Query	Response (No. of characters)
FSY	FSY△m	FSY?	FSY△m (FIX1)

Parameter m = <NR1>
0 : OFF (Normal synchronization)
1 : ON (Frame synchronization)

Application example

Program

・Sets frame synchronization ON.

```
> FSY△1
```

Query

・Makes a query when the parameter is set at normal synchronization.

```
> FSY?
< FSY△0
```

・Makes a query when the reception pattern is PRBS.

```
> FSY?
< ERR
```

7.5.1.14. FLN Frame Length (Frame length)

Function Sets/queries the frame length.

Header	Program	Query	Response (No. of characters)
FLN	FLN△m	FLN?	FLN△m (FIX2)

Parameter m = <NR1>
8 : 8bit
16 : 16bit
24 : 24bit
32 : 32bit

Application example

Program

・Sets the frame length at 16 bits.

```
> FLN△16
```

Query

・Makes a query when the frame length is 8 bits.

```
> FLN?
< FLN△△8
```

・Makes a query when the synchronization method is normal.

```
> FLN?
< ERR
```

7.5.1.15. PAG Number of Pattern Pages (Page)

Function Specifies/queries the head of the PRGM pattern to be edited by 16 bits. This unit of 16 bits is called a page.

Header	Program	Query	Response (No. of characters)
PAG	PAG△m	PAG?	PAG△m (FIX6)
ADR	ADR△m	ADR?	ADR△m (FIX6)

Parameter m = <NR1>
1 to 524288 / Step 1

Application example

Program

・Sets the head of the editing target at the 1000th page.
(The bit number from the head of the pattern is the 16000th bit.)

```
> PAG△1000
```

Sets the head of the editing target at the 1000th page.
(The bit number from the head of the pattern is the 16000th bit.)

```
> ADR△1000
```

Query

・Makes a query when the head of the editing target is the 100th page.

```
> PAG?
< PAG△△△△100
```

・Makes a query when the head of the editing target is the 100th page.

```
> ADR?
< ADR△△△△100
```

・Makes a query when the head of the editing target is set at Zero-Subst.

```
> PAG?
< ERR
```

・Makes a query when the head of the editing target is set at Zero-Subst.

```
> ADR?
< ERR
```

Remarks This command, available in two types, PAG and ADR, has the same function in both formats. The same result is obtained by executing either format. The value enabled to be set varies by the PRGM pattern length. The maximum value enabled to be set is the quotient of the pattern length divided by 16. In the presence of any remainder, the value to be set is the quotient + 1. Moreover, when a value which exceeds the maximum value enabled to be set is set, the maximum value effective in that state will be set. Incidentally, the initial value will be set at 1 when no value is set by this command.

7.5.1.16. BIT BIT Pattern Bit (Pattern bit)

Function Specifies/queries a maximum of eight pages equivalent of PRGM patterns by 16 bits from the position specified by PAG/ADR.

Header	Program	Query	Response (No. of characters)
BIT	BIT△m	BIT?	BIT△m (as shown below)
	BIT△#Hm		

Parameter m = <NR1>
0 to 65535 / Step 1
#Hm = <Hexadecimal format>
0000 to FFFF / Step 1
m or #Hm is enabled to specify data of a maximum of eight pages equivalent by using a comma (,) as a delimiter. Incidentally, the bit types to be set here are bit1 for LSB and bit16 for MSB. In other words, when BIT△32768 or BIT△#H8000 is set, 1 is only set for the bit16 of MSB. On the other hand, when BIT△1 or BIT△#H0001 is set, 1 is only set for the bit 1 of LSB.

Response Returns, in the format shown below, the number of pages set and the details of the bits from the page to a maximum of eight pages equivalent, leading to the maximum pattern setting bit.

PAG△******;BIT△#H****,#H****,#H****,#H****,#H****,#H****,#H****,#H****

Application example

Program

・Sets three pages equivalent of pattern bits from the page currently set.

```
> BIT△10,20,30
> BIT△#H1000,2000,3000
```

・Sets the number of pages as well as four pages equivalent of pattern bits from that page.

```
> PAG△10;BIT△10,20,30,40
> PAG△10;BIT△#H1000,#H2000,#H3000,#H4FFF
> ADR△10;BIT△10,20,30,40
> ADR△10;BIT△#H1000,#H2000,#H3000,#H4FFF
```

Query

・Makes a query when the header of the editing target is 1 and the maximum number of pages enabled to be obtained is 29.

```
>BIT?
<PAG△△△△△△1;BIT△#H0000,#H0000,#H0000,#H0000,#H0000,
  #H0000,#H0000,#H0000
```

Moreover, as the example below shows, patterns can be read to the end.

```
>BIT?
<PAG△△△△△△9;BIT△#H0000,#H0000,#H0000,#H0000,#H0000,
  #H0000,#H0000,#H0000
>BIT?
<PAG△△△△△17;BIT△#H0000,#H0000,#H0000,#H0000,#H0000,
  #H0000,#H0000,#H0000
>BIT?
<PAG△△△△△25;BIT△#H0000,#H0000,#H0000,#H0000,#H0000
```

The addresses of the patterns to be output increases in the equipment interior. When a different command is executed before all the patterns are output, the output of the patterns is terminated.

・Makes a query when the pattern bit is set at Zero-Subst.

```
> BIT?
< ERR
```

7.5.1.17. PDT Pattern Data (Pattern data)

Function Sets/queries the transmission pattern data in the program division from the <start> to the <end> address.

Header	Program	Query	Response (No. of characters)
PDT	PDT△ m0,m1,m2	PDT?△m0	PDT△m (as shown below)

Parameter m0 = <Hexadecimal format>
Address from which to start setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m1 = <Hexadecimal format>
Address at which to end setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m2 = <Hexadecimal or binary format>
#H***... Hexadecimal data
***... :data of 1 to 400 characters (equivalent to 400 × 4 bits) (0 to 9, A to F)
#B***... Binary data
***... :data of 1 to 400 characters (equivalent to 400 bits) (0, 1)

Bit number smaller than the range specified by <m1> and <m0> but larger than <m2>	Makes setting by repeating <m2> by the number of bits specified by <m1> and <m0>. ＜*Example*＞ ・<m0>=#H0, <m1>=#H1F, <m2>=#HABC Seting data:D5533DD5 (at LSB First) ABCABCAB (at MSB First) ・<m0>=#H0, <m1>=#H7, <m2>=#B011 Setting data:10110110 (at LSB First) 01101101 (at MSB First)
Bit number larger than the range specified by <m1> and <m0> but smaller than <m2>	Truncates <m2> by the number of bits specified by <m1> and <m0>. ＜*Example*＞ ・<m0>=#H0, <m1>=#HF, <m2>=#HABCDEF Setting data:D5B3 (at LSB First) ABCD (at MSB First) ・<m0>=#H0, <m1>=#H3, <m2>=#B01100110 Setting data:00000110 (at LSB First) 01100000 (at MSB First)

7.5.1.17. PDT Pattern Data (Pattern data) (continued)

Response

Bits of data from <m0> equal to or greater than 400 × 4 bits	***...: Bits of data from <m0>/4 characters
Bits of data from <m0> smaller than 400 × 4 bits	***...: Bits of data from <m0>/4 characters *<Example>* Data following the <m0> address - 010100 (Bin) Response : "#H28" (at LSB First hour(s)) "#H54" (at MSB First hour(s)) (When the last character comes less than 4 bits, the number of bits filled plus zeros inserted for the number of empty bits constitutes a response. The character in the section filled by no data fails to constitute a response.)

Application example

Program

・Sets the program pattern of Address (Hex) 0 to 1F at 0.

```
> PDT△#H0,#H1F,#H0
```

Query

・Queries the program pattern from Address (Hex) 0.

```
> PDT?△#H0
< #H000000.....
```

Note

When maximum pattern range (from #H0 to #H7FFFFF) is specified, it takes about 25 seconds until pattern loading.starts.

7.5.1.18. BDT Pattern Binary Data (Pattern binary data)

Function Sets/queries the transmission pattern data in the program division from the <start> to the <end> address.

Header	Program	Query	Response (No. of characters)
BDT	BDT△ m0,m1,m2	BDT?△m0	BDT△m (as shown below)

Parameter m0 = <Hexadecimal format>
Address from which to start setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m1 = <Hexadecimal format>
Address at which to end setting
#H0 to #H7FFFFF Address(Hex) 0 to 7FFFFFbit/Step:1
m2 = <ARBITRARY BLOCK PROGRAM DATA>
#XYYY<Z> X : Number of digits of YYY
YYY : Byte number of <Z> 1 to 16000 (bytes)
<Z> : Binary data up to a maximum of 16000 bytes

Bit number smaller than the range specified by <m1> and <m0> but larger than <Z>	Makes setting by repeating <Z> by the number of bits specified by <m1> and <m0>. *<Example>* <m0>=#H0, <m1>=#H1F, <m2>=#12AB Setting data:82428242 (at LSB First) 41424142 (at MSB Firs)
Bit number larger than the range specified by <m1> and <m0> but smaller than <Z>	Truncates <Z> by the number of bits specified by <m1> and <m0>. *<Expamle>* <m0>=#H0, <m1>=#HF, <m2>=#13ABC Setting data:8242 (at LSR First) 4142 (at MSB First)

Response

Bits of data from <m0> equal to or greater than 16000 × 4 bits	***...:16000 characters (XXX x 4 bits)
Bits of data from <m0> smaller than 16000 × 4 bits	***...:Bits of data from <m0>/4 characters *<Exapmle>* Data following the <m0> address - 010101 (Bin) Response:"#11(" (at LSB First) "#11T" (at MSB First) (When the last character comes less than 4 bits, the number of bits filled plus zeros inserted for the number of empty bits constitutes a response. The character in the section filled by no data fails to constitute a response.)

7.3.1.15. BDT Pattern Binary Data (Pattern binary data) (continued)

Application example

Program

・Sets the program pattern of Address (Hex) 0 to 1F at #H41414141.

```
> BDT△#H0,#H1F,#11A
```

Query

・Queries the program pattern from Address (Hex) 0.

```
> BDT?△#H0
< #H000000.....
```

Note

When maximum pattern range (from #H0 to #H7FFFFF) is specified, it takes about 25 seconds until pattern loading.starts.

7.5.1.19. DRA Reversion of Transmission Pattern Specification Address Data (Data reverse address)

Function Reverses the data of the transmission pattern specification address. (Address specification)

Header	Program	Query	Response (No. of characters)
DRA	DRA△m1,m2	None	None

Parameter m1 = <Hexadecimal format>

#H0 to #H7FFFFF Start Address(Hex): 0 to 7FFFFFbit/Step:1

m2 = <Hexadecimal format>

#H0 to #H7FFFFF End Address(Hex): 0 to 7FFFFFbit/Step:1

Application example

Reverses the program patterns from Address (Hex) 0 to FF.

```
> DRA△#H0,#HFF
```

7.5.1.20. DRD Reversion of Transmission Pattern Specification Address Data (Data reverse address delta)

Function Reverses the transmission pattern data in the program division. (Delta specification)

Header	Program	Query	Response (No. of characters)
DRD	DRD△m1,m2	None	None

Parameter m1 = <Hexadecimal format>

#H0 to #H7FFFFF Start Address(Hex): 0 to 7FFFFFbit/Step:1

m2 = <NR1>

1 to 8388608 delta :1 to 8388608 bit/Step:1

Application example

Reverses 256 bits equivalent of the program patterns from Address (Hex) 0.

```
> DRD△#H0,256
```

7.5.1.21. WRT Pattern Data Input Byte Number (Pattern data write)

Function Sets the pattern data by the specified number of blocks from the specified address.

Header	Program	Query	Response (No. of characters)
WRT	WRT△m1,m2	None	None

Parameter m1 = <NR1>

1 to 400 / Step1 Number of bytes of pattern transfer

m2 = <NR1>

0 to 524287 / Step1 Head address of pattern input

Application example

Specifies 18 bytes to set data from the 1st to the 9th page since a page consists of 16 bits.

```
> WRT△18,0
> ABCDEFGHIJKLMNOPQR
```

In this case, patterns are stored into memory as shown below.

```
Address  : Pattern
   0     : 0100 0010 1000 0010 0010 0010 1100 0010
   4     : 0110 0010 1010 0010 0001 0010 1110 0010
   8     : 0101 0010 1001 0010 0011 0010 1101 0010
   C     : 0111 0010 1011 0010 0000 1010 1111 0010
   10    : 0100 1010 1000 1010
```

7.5.1.22. RED Pattern Data Output Byte Number (Pattern data read?)

Function Reads out the pattern data by the specified number of blocks from the specified address.

Header	Program	Query	Response (No. of characters)
RED	None	RED△m1,m2	Data pattern sequence (as shown below)

Parameter m1 = <NR1>

1 to 400 / Step1 Number of bytes of pattern transfer

m2 = <NR1>

0 to 524287 / Step1 Head address of pattern output

Application example

When reading out data from the 1st to the 10th page.

```
> RED△20,0
< ABCDEFGHIJKLMNOPQRST
```

7.5.1.23. ALL ALL Preset (All bits of all pages) (Preset all 0 or 1)

Function Sets all the transmission patterns either 1 or 0.

Header	Program	Query	Response (No. of characters)
ALL	ALL△m	None	None

Parameter m = <NR1>

0 : Clears all bits on all pages.

1 : Sets all bits on all pages.

Application example

Sets all bits to 1.

```
> ALL△1
```

7.5.1.24. PST Preset (All bits on a page) (Preset page 0 or 1)

Function Sets all the transmission patterns on pages set by PAG/ADR either 1 or 0.

Header	Program	Query	Response (No. of characters)
PST	PST△m	None	None

Parameter m = <NR1>

0 : Clears all bits on a page.

1 : Sets all bits on a page.

Application example

Sets all bits on a page to 1.

```
> PST△1
```

7.5.1.25. PML Level of PRBS 1 (PRBS Mark Logic)

Function Sets/queries whether the PRBS 1 is High level or Low level.

Header	Program	Query	Response (No. of characters)
PML	PML△m	PML?	PML△m (FIX1)

Parameter m = <NR1>

0 Sets PRBS Logic to Mark High.

1 Sets PRBS Logic to Mark Low.

Application example

Program

・Sets the PRBS Logic at Mark Low.

```
> PML△1
```

Query

・Makes a query when the PRBS Logic is set at Mark High.

```
> PML?
< PML△0
```

7.5.1.26. GML Level of PRGM 1 (PRGM Mark Logic)

Function Sets/queries whether the PRGM 1 is High level or Low level.

Header	Program	Query	Response (No. of characters)
GML	GML△m	GML?	GML△m (FIX1)

Parameter m = <NR1>
0 : Sets PRGM Logic to Mark High.
1 : Sets PRGM Logic to Mark Low.

Application example

Program

・Sets PRGM Logic to Mark Low.

```
> GML△1
```

Query

・Makes a query when the PRGM Logic is set at Mark High.

```
> GML?
< GML△0
```

7.5.1.27. TIT Setting of Measurement Item (Test Item)

Function Sets/queries the items of the measurement target.

Header	Program	Query	Response (No. of characters)
TIT	TIT△m	TIT?	TIT△m (FIX1)

Parameter m = <NR1>
0 : Error / Alarm measurement
1 : Eye Margin measurement

Application example

Program

・Sets the parameter at Error/Alarm measurement.

```
> TIT△0
```

Query

・Makes a query when the parameter is set at Eye Margin measurement.

```
> TIT?
< TIT△1
```

7.5.1.28. STA Measurement Start/Restart (Start / Restart)

Function Starts measurement. Restarts measurement when executed during the measurement.

Header	Program	Query	Response (No. of characters)
STA	STA	None	None

Parameter None

Application example

Program

・Starts measurement.

```
> STA
```

7.5.1.29. STO Measurement Stop (Stop)

Function Stops measurement.

Header	Program	Query	Response (No. of characters)
STO	STO	None	None

Parameter None

Application example

Program

・Stops measurement.

```
> STO
```

7.5.1.30. MRS measurement restart setting (Measurement restart)

Function Sets/queries measurement restart ON/OFF when Threshold or Delay is changed.

Header	Program	Query	Response (No. of characters)
MRS	MRS△m	MRS?	MRS△m (FIX1)

Parameter m = <NR1>

0 : Measurement restart ON

1 : Measurement restart OFF

Application example

Program

・Sets measurement restart ON.

```
> MRS△0
```

Query

・Queries when measurement restart OFF is set.

```
> MRS?
< MRS△1
```

7.5.1.31. MOD Setting of Measurement Mode (Measurement mode)

Function Sets/queries the measurement mode for Error/Alarm measurement.

Header	Program	Query	Response (No. of characters)
MOD	MOD△m	MOD?	MOD△m (FIX1)

Parameter m = <NR1>

0 : Repeat measurement
1 : Single measurement
2 : Untimed measurement

Application example

Program

・Sets Single measurement.

```
> MOD△1
```

Query

・Queries when Repeat measurement is set.

```
> MOD?
< MOD△0
```

・Queries when Eye Margin measurement is set.

```
> MOD?
< ERR
```

7.5.1.32. PRD Measurement Period Setting (Measurement period)

Function Sets/queries the measurement period for Error/Alarm measurement.

Header	Program	Query	Response (No. of characters)
PRD	PRD△ m1,m2,m3,m4	PRD?	PRD△m1,m2,m3,m4 (FIX2)

Parameter m = <NR1>

All the setting steps are 1
m1 : 0 to 99 day(s)
m2 : 0 to 23 hour(s)
m3 : 0 to 59 minute(s)
m4 : 0 to 59 second(s)

Application example

Program

・Sets the measurement period to 1 day 10 hours 0 minute 0 second.

```
> PRD△1,10,0,0
```

Query

・Queries when the measurement period is 1 day 4 hours 15 minutes 5 seconds.

```
> PRD?
< PRD△00,04,15,05
```

・Queries when Eye Margin measurement is set.

```
> PRD?
< ERR
```

7.5.1.33. TSS Timed Start Setting (Timed start set)

Function Sets/queries ON/OFF for timed start.

Header	Program	Query	Response (No. of characters)
TSS	TSS△m	TSS?	TSS△m (FIX1)

Parameter m = <NR1>

0 : Timed start ON
1 : Timed start OFF

Application example

Program

・Sets timed start ON.

```
> TSS△0
```

Query

・Queries when timed start OFF is set.

```
> TSS?
< TSS△1
```

・Queries when the measurement

```
> TSS?
< ERR
```

7.5.1.34. STI Timed Start Time Setting (Timed start time)

Function Sets/queries measurement start time of timed start.

Header	Program	Query	Response (No. of characters)
STI	STI△ m1,m2,m3,m4	STI?	STI△m1,m2,m3,m4,m5,m6 (each FIX2)

Parameter m = <NR1>

All the setting steps are 1.
m1 : 0 to 99 year(s)
m2 : 1 to 12 month(s)
m3 : 1 to 31 day(s)
m4 : 0 to 23 hour(s)
m5 : 0 to 59 minute(s)
m6 : 0 to 59 second(s)

Application example

Program

・Sets measurement start date/time to 99-2-6 12:12:2.

```
> STI△99,2,6,12,12,02
```

Query

・Queries when measurement start time is set to 99-5-28 12:12:00.

```
> STI?
< STI△99,05,28,12,12,00
```

・Queries when Eye Margin measurement is set.

```
> STI?
< ERR
```

7.5.1.35. ETY Error Detection Mode Selection (Error type)

Function Sets/queries the error type to be measured.

Header	Program	Query	Response (No. of characters)
ETY	ETY△m	ETY?	ETY△m (FIX1)

Parameter m = <NR1>
0 : Total
1 : Insertion (INS/OMI)
2 : Omission (INS/OMI)

Application example

Program

・Sets measurement error type to Total.

```
> ETY△0
```

Query

・Queries when INS/OMI is set.

```
> ETY?
< ETY△1
```

・Queries when the measurement item is other than Error/Alarm.

```
> ETY?
< ERR
```

Remarks When specifying Insertion or Omission in this command, query to the measurement result such as ER? or EC? will be Insertion or Omission only.

7.5.1.36. SCH Measurement Mask Route Selection (Measure ch mask select)

Function Sets/queries the internal route which masks measurement.

Header	Program	Query	Response (No. of characters)
SCH	SCH△m	SCH?	SCH△m (FIX10)

Parameter m = <binary format>
Flags 1 in a bit which masks measurement. LSB is 1ch and MSB is 8ch.
#B00000000 : Mask none of routes.
#B00000001 : Mask 1ch (route 1) only.
#B00000011 : Mask 1ch (route 1) and 2ch (route 2).
#B10000000 : Mask 8ch (route 8) only.
#B11111110 : Mask 2ch to 8ch(routes to route 8).

Application example

Program

・Masks measurement of 1ch and 3ch.

```
> SCH△#B00000101
```

Query

・Queries when 2ch to 4ch are set to be masked for measurement.

```
> SCH?
< MSK△0;SCH△#B00001110
```

Note) The string "MSK△0;" is output constantly.

・Queries when the measurement item is other than Error/Alarm.

```
> SCH?
< ERR
```

7.5.1.37. MSA? Measurement Start Time (Measurement start time)

Function Queries measurement start time for Error/Alarm measurement or Eye Margin measurement.

Header	Program	Query	Response (No. of characters)
None	None	MSA?	MSA△m1,m2,m3,m4,m5,m6 (each FIX2)

Parameter m = <NR1>
m1 : 0 to 99 yesr(s)
m2 : 1 to 12 month(s)
m3 : 1 to 31 day(s)
m4 : 0 to 23 hour(s)
m5 : 0 to 59 minute(s)
m6 : 0 to 59 second(s)

Application example

Query

・Queries when measurement start date/time is 99-5-28 12:12:00.

```
> MSA?
< MSA△99,05,28,12,12,00
```

・Queries when measurement start time data does not exist.

```
> MSA?
< MSA△00,00,00,00,00,00
```

7.5.1.38. MSO? Measurement Stop Time (Measurement stop time)

Function Queries measurement stop time for Error/Alarm measurement.

Header	Program	Query	Response (No. of characters)
None	None	MSO?	MSO△m1,m2,m3,m4,m5,m6 (each FIX2)

Parameter m = <NR1>
m1 : 0 to 99 year(s)
m2 : 1 to 12 month(s)
m3 : 1 to 31 day(s)
m4 : 0 to 23 hour(s)
m5 : 0 to 59 minute(s)
m6 : 0 to 59 second(s)

Application example

Query

・Queries when measurement stop date/time is 99-5-28 12:12:00.

```
> MSO?
< MSO△99,05,28,12,12,00
```

・Queries when measurement stop time data does not exist.

```
> MSO?
< MSO△00,00,00,00,00,00
```

7.5.1.39. MSR? Measurement Status (Measurement in progress or stop?)

Function Queries the measurement status for Error/Alarm measurement or Eye Margin measurement.

Header	Program	Query	Response (No. of characters)
None	None	MSR?	MSR△m (FIX1)

Parameter m = <NR1>

0 : Measurement stop
1 : Measurement in progress
2 : Eye Margin measurement failed (only for Margin / Diagram measurement)

Application example

Query

・Queries when measurement is being in progress.

```
> MSR?
< MSR△1
```

7.5.1.40. MLP? Measurement Elapsed Time (Measurement elapsed time)

Function Queries the measurement elapsed time for Error/Alarm measurement or Eye Margin measurement.

Header	Program	Query	Response (No. of characters)
None	None	MLP?	MLP△m1,m2,m3,m4 (m1:FIX5, other FIX2)

Parameter m = <NR1>

m1 : 0 to 36135	day(s)
m2 : 0 to 23	hour(s)
m3 : 0 to 59	minute(s)
m4 : 0 to 59	second(s)

Application example

Query

・Queries when measurement date and time is 99 days 12 hours 12 minutes 00 second.

```
> MLP?
< MLP△△△△99,12,12,00
```

7.5.1.41. ETI? Measurement Residual Time (Error/Alarm measurement timed time)

Function Queries the measurement residual time for Error/Alarm measurement.

Header	Program	Query	Response (No. of characters)
None	None	ETI?	ETI△m1,m2,m3,m4 (m1:FIX5,other FIX2)

Parameter m = <NR1>

m1 : 0 to 36135 day(s)
m2 : 0 to 23 hour(s)
m3 : 0 to 59 minute(s)
m4 : 0 to 59 second(s)

Application example

Query

・Makes a query when the measurement residual time is 99 days 12 hours 12 minutes 00 second.

```
> ETI?
< ETI△△△△99,12,12,00
```

・Makes a query when Eye Margin measurement is set.

```
> ETI?
< ERR
```

7.5.1.42. INT? Intermediate Data Creation Time (Measurement intermediate time)

Function Queries the intermediate data creation time for Error/Alarm measurement.

Header	Program	Query	Response (No. of characters)
None	None	INT?	INT△m1,m2,m3,m4,m5,m6 (each FIX2)

Parameter m = <NR1>

m1 : 0 to 99 year(s)
m2 : 1 to 12 month(s)
m3 : 1 to 31 day(s)
m4 : 0 to 23 hour(s)
m5 : 0 to 59 minute(s)
m6 : 0 to 59 second(s)

Application example

Query

・Makes a query when the intermediate data creation time is 99-5-28 12:12:00.

```
> INT?
< INT△99,05,28,12,12,00
```

・Makes a query when the intermediate data creation time does not exist.

```
> INT?
< INT△00,00,00,00,00,00
```

7.5.1.43. AOT? Alarm Occurrence Time (Alarm measurement occur time)

Function Queries the Alarm occurrence time.

Header	Program	Query	Response (No. of characters)
None	None	AOT?△m	AOT△m1,m2,m3,m4,m5,m6 (FIX2)

Parameter m = <NR1>
0 : Power Fail
1 : Clock Loss
2 : Pattern Sync Loss

Response m? = <NR1>
m1 : 0 to 99 year(s)
m2 : 1 to 12 month(s)
m3 : 1 to 31 day(s)
m4 : 0 to 23 hour(s)
m5 : 0 to 59 minute(s)
m6 : 0 to 59 second(s)

Application example

Query

・Queries when the clock loss time is 99-5-28 12:12:00.

```
> AOT?△1
< AOT△99,05,28,12,12,00
```

・Queries when sync loss time data does not exist.

```
> AOT?△2
< AOT△00,00,00,00,00,00
```

・Queries when Eye Margin measurement is set.

```
> AOT?△0
< ERR
```

7.5.1.44. ART? Alarm Recovery Time (Alarm measurement recover time)

Function Queries the Alarm recovery time.

Header	Program	Query	Response (No of charactors)
None	None	ART?△m	ART△m1,m2,m3,m4,m5,m6 (FIX2)

Parameter m = <NR1>

0 : Power Fail
1 : Clock Loss
2 : Pattern Sync Loss

Response m? = <NR1>

m1 : 0 to 99 : year(s)
m2 : 1 to 12 : month(s)
m3 : 1 to 31 : day(s)
m4 : 0 to 23 : hour(s)
m5 : 0 to 59 : minute(s)
m6 : 0 to 59 : second(s)

Application example

Query

・Queries when the clock loss recovery time is 99-5-28 12:12:00.

```
> ART?△1
< ART△99,05,28,12,12,00
```

・Queries when sync loss time recovery data does not exist.

```
> ART?△2
< ART△00,00,00,00,00,00
```

・Queries when Eye Margin measurement is set.

```
> ART?△0
< ERR
```

7.5.1.45. EMD Setting of Eye Margin Measurement Mode (Eye Margin Measurement mode)

Function Sets/queries the measurement mode for Eye Margin measurement.

Header	Program	Query	Response (No. of characters)
EMD	EMD△m	EMD?	EMD△m (FIX1)

Parameter m = <NR1>
0 : Margin measurement
1 : Diagram measurement

Application example

Program

・Sets Margin measurement.

```
> EMD△0
```

Query

・Queries when Diagram measurement is set.

```
> EMD?
< EMD△1
```

・Queries when Error/Alarm measurement is set.

```
> EMD?
< ERR
```

7.5.1.46. EYT Eye Margin Measurement Item Setting (Eye margin measurement type)

Function Sets/queries the measurement item and number of points for Eye Margin measurement.

Header	Program	Query	Response (No. of characters)
EYT	EYT△m	EYT?	EYT△m (FIX1)

Parameter m = <NR1>
0 : Phase & Threshold
1 : Phase
2 : Threshold
3 : 8 points
4 : 16 points
5 : 32 points
6 : 64 points

Application example

Program

・Sets Phase in Margin measurement.

```
> EYT△1
```

Query

・Queries when 16 points is set in Diagram measurement.

```
> EYT?
< EYT△4
```

・Queries when Error/Alarm measurement is set.

```
> EYT?
< ERR
```

7.5.1.47. EMT Eye Margin Measurement Threshold Setting (Eye margin measurement threshold)

Function Sets/queries threshold when Eye Margin measurement mode is set.

Header	Program	Query	Response (No. of characters)
EMT	EMT△m	EMT?	EMT△m (FIX1)

Parameter m = <NR1>

0 : E_3
1 : E_4
2 : E_5
3 : E_6
4 : E_7
5 : E_8
6 : E_9
7 : E_10
8 : E_11
9 : E_12

Application example

Program

・Set E-4.

```
> EMT△1
```

Query

・Makes a query when E-9 is set.

```
> EMT?
< EMT△6
```

・Makes a query when Error/Alarm measurement is set.

```
> EMT?
< ERR
```

7.5.1.48. EMR Eye Margin Measurement Resolution Setting (Eye margin measurement resolution)

Function Sets/queries measurement resolution when Eye Margin measurement mode is set.

Header	Program	Query	Response (No. of characters)
EMR	EMR△m	EMR?	EMR△m (FIX1)

Parameter m = <NR1>
0 : Course
1 : Fine

Application example

Program

・Sets to Fine

> EMR△1

Query

・Queries when Course is set.

> EMR?

< EMR△0

・Queries when Error/Alarm measurement is set.

> EMR?

< ERR

7.5.1.49. EDT Eye Diagram Measurement Threshold Setting (Eye diagram measurement threshold)

Function Sets/queries threshold when Eye Diagram measurement mode is set.

Header	Program	Query	Response (No. of characters)
EDT	EDT△m	EDT?	EDT△m (FIX12)

Parameter m = <binary format>

・Flags 1 in a bit to be masked for measurement. Where MSB is 1E_3 and LSB is 1E-12:
#B1000000000 : Measures 1E-3 only.
#B1010000001 : Measures 1E-3,1E-5,and 1E-12.
#B1111111111 : Measures all thresholds.

Application example

Program

・Sets to E-4.

> EDT△#B0100000000

Query

・Queries when E-9 and E-4 are set.

> EDT?

< EDT△#B0100001000

・Queries when Error/Alarm measurement is set.

> EDT?

< ERR

7.5.1.50. EDP Eye Diagram Measurement Effective Measurement Points (Eye diagram measured point)

Function Queries the number of effective measurement points at each threshold when Eye Diagram measurement mode is set.

Header	Program	Query	Response (No. of characters)
EDP	None	EDP?△m	EDP△m0,m1 (FIX2,FIX64)

Parameter m = <NR1>

Specifies threshold to be measured.

3 : 1E-3
4 : 1E_4
5 : 1E_5
6 : 1E_6
7 : 1E_7
8 : 1E_8
9 : 1E_9
10 : 1E_10
11 : 1E_11
12 : 1E_12

m0 = <NR1>

Number of measurement points specified in the EYT command

0 : 8points
1 : 16points
2 : 32points
3 : 64points

m1 = <binary format>

Effective points are set to 1. Ineffective points or those not to be measured are set to 0.

#B11110111000000000000000000000000000000000000000000000000000000000000
: When points 1, 2, 3, 4, 6, 7 and 8 are effective

#B11111110001111110000000000000000000000000000000000000000000000000001
: When points 1 to 7, 11 to 16 and 64 are effective

Application example

Query

・Makes a query the effective measurement points when measuring E-9 at 16 points.

```
> EDP?△9
< EDP△1,#B11111101111111110000000000000000000000000000000000000000
  00000000000000
```

・Makes a query when Error/Alarm measurement is set.

```
> EDP?△1
< ERR
```

7.5.1.51. ASM Setting of Auto Search Mode (Auto search mode)

Function Sets/queries the operation mode for Auto Search.

Header	Program	Query	Response (No. of characters)
ASM	ASM△m	ASM?	ASM△m (FIX1)

Parameter m = <NR1>
0 : Phase & Threshold
1 : Threshold
2 : Phase
3 : OFF

Application example

Program

・Set the operation mode to Threshold search.

```
> ASM△1
```

Query

・Query when the operation mode is set to Phase & Threshold search.

```
> ASM?
> ASM△0
```

7.5.1.52. SRH Auto Search Function (Auto search)

Function Sets/queries the operation for Auto Search.

Header	Program	Query	Response (No. of characters)
SRH	SRH△m	SRH?	SRH△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON
2 : Auto Search faied (for query only)

Application example

Program

・Starts Auto Search

```
> SRH△1
```

Query

・Makes a query during Auto Search measurement

```
> SRH?
> SRH△1
```

・Make a query when Auto Search ends in fail.

```
> SRH?
> SRH△2
```

7.5.1.53. PSH PRBS Pattern Search Function (PRBS pattern search)

Function Sets/queries ON/OFF for PRBS Pattern Search.

Header	Program	Query	Response (No. of characters)
PSH	PSH△m	PSH?	PSH△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets PRBS Pattern Search to ON.

```
> PSH△1
```

Query

・Queries when PRBS Pattern Search is OFF.

```
> PSH?
> PSH△0
```

・Queries when Auto Sync is OFF.

```
> PSH?
> ERR
```

7.5.2. Clock/data input Related

The commands described in this section set/query the reception clock and data.

Note 1) If a step which is not preset is to be used when setting NR1, a command selects the smallest value nearest to the allowable value, that is, truncates a fraction.

Note 2) If no ED unit exist in the slot to be set, a command issues an execution error and returns ERR for query.

7.5.2.1. CPL Clock Input Polarity (Clock polarity)

Function Sets/queries the clock input polarity.

Header	Program	Query	Response (No. of characters)
CPL	CPL△m	CPL?	CPL△m (FIX1)

Parameter m = <NR1>
0 : CLK (POS)
1 : XCLK (NEG)

Application example

Program

・Set clock polarity to CLK (POS).

```
> CPL△0
```

Query

・Query when clock polarity is XCLK (NEG).

```
> CPL?
< CPL△1
```

7.5.2.2. CPA Clock Input Phase Setting (Clock phase adjust)

Function Sets/queries the clock input phase.

Header	Program	Query	Response (No. of characters)
CPA	CPA△m	CPA?	CPA△m (FIX5)

Parameter m = <NR1>
-1000 to 1000 1000 to 1000 ps/Step : 2ps

Application example

Program

・Sets the phase 500ps

```
> CPA△500
```

Query

・Makes a query when the phase is set at -500ps

```
> CPA?
< CPA△△-500
```

7.5.2.3. CTM Clock Input Terminal Voltage (Clock termination)

Function Sets/queries clock input terminal voltage.

Header	Program	Query	Response (No. of characters)
CTM	CTM△m	CTM?	CTM△m (FIX1)

Parameter m = <NR1>
0 : GND
1 : -2V
2 : +3V

Application example

Program

・Sets clock input terminal voltage to GND.

```
> CTM△0
```

Query

・Queries when clock input terminal voltage is -2V.

```
> CTM?
< CTM△1
```

7.5.2.4. DAL Data Input Level (Data Level)

Function Sets/queries the data input level.

Header	Program	Query	Response (No. of characters)
DAL	DAL△m	DAL?	DAL△m (FIX1)

Parameter m = <NR1>
0 : VAR level
1 : ECL level

Application example

Program

・Sets Data input level to VAR

```
> DAL△0
```

Query

・Queries when Data input level is ECL

```
> DAL?
< DAL△1
```

7.5.2.5. DTH Data Input Threshold Setting (Data threshold)

Function Sets/queries the data input threshold.

Header	Program	Query	Response (No. of characters)
DTH	DTH△m	DTH?	DTH△m (FIX6)

Parameter m = <NR1>
-4.000 to 4.000 -4.000 to +4.000 V/Step : 0.001

Application example

Program

・Sets Data input threshold to -0.1V.

```
> DTH△-0.1
```

Query

・Queries when Data input threshold is set at 0.2V.

```
> DTH?
< DTH△△0.200
```

・Queries when Data input threshold is ECL

```
> DTH?
< ERR
```

7.5.2.6. DTM Sets/queries Data Input Terminal Voltage. (Data termination)

Function Sets/queries Data input terminal voltage.

Header	Program	Query	Response (No. of characters)
DTM	DTM△m	DTM?	DTM△m (FIX1)

Parameter m = <NR1>
0 : GND
1 : -2V
2 : +3V

Application example

Program

・Sets data input terminal voltage to GND.

```
> DTM△0
```

Query

・Queries when data input terminal voltage is -2V.

```
> DTM?
< DTM△1
```

7.5.3. Measurement results

The commands described in this section set/query the measurement result/measurement condition.

7.5.3.1. CLS Clock Loss Process Option (Clock loss process option)

Function Sets/queries whether Clock Loss period is to be evaluated for measurement.

Header	Program	Query	Response (No. of characters)
CLS	CLS△m	CLS?	CLS△m (FIX1)

Parameter m = <NR1>
0 : Exclude from calculation.
1 : Include from calculation.

Application example

Program

・-- Sets that Clock Loss is included in calculation.

```
> CLS△1
```

Query

・Queries when Clock Loss is not included in calculation.

```
> CLS?
< CLS△0
```

・Queries when Eye Margin measurement is set.

```
> CLS?
< ERR
```

7.5.3.2. SLS Sync Loss Process Option (Sync loss process option)

Function Sets/query whether Sync Loss period is to be evaluated for measurement (Performance, EI%EFI, Threshold EI%EFI).

Header	Program	Query	Response (No. of characters)
SLS	SLS△m	SLS?	SLS△m (FIX1)

Parameter m = <NR1>
0 : Exclude from calculation
1 : Include from calculation

Application example

Program

・Sets so that Sync Loss is included in calculation.

```
> SLS△1
```

Query

・Queries when Sync Loss is not included in calculation.

```
> SLS?
< SLS△0
```

・Queries when Eye Margin measurement is set.

```
> SLS?
< ERR
```

7.5.3.3. ITV Measurement Interval Time Selection (Interval time)

Function Sets/queries IE and %EFI measurement interval.

Header	Program	Query	Response (No. of characters)
ITV	ITV△m	ITV?	ITV△m (FIX1)

Parameter m = <NR1>
0 : 100mS
3 : 1S

Application example

Program

・-- Sets interval to 100 ms.

```
> ITV△0
```

Query

・Queries when interval is 1 s.

```
> ITV?
< ITV△3
```

・Queries when Eye Margin measurement is set.

```
> ITV?
< ERR
```

7.5.3.4. PRF Error Performance Data Measurement ON/OFF (Error performance measurement)

Function Sets/queries ON/OFF of error performance measurement.

Header	Program	Query	Response (No. of characters)
PRF	PRF△m	PRF?	PRF△m (FIX1)

Parameter m = <NR1>
0 : ON
1 : OFF

Application example

Program

・Sets error performance measurement to ON.

```
> PRF△0
```

Query

・Queries when error performance measurement is OFF.

```
> PRF?
< PRF△1
```

・Queries when Eye Margin measurement is set.

```
> PRF?
< ERR
```

7.5.3.5. ETH Error Performance Threshold Setting (Error performance threshold)

Function Sets/queries the threshold when calculating SES and DM for performance measurement.

Header	Program	Query	Response (No. of characters)
ETH	ETH△m	ETH?	ETH△m (FIX1)

Parameter m = <NR1>
0 : SES:1E-3/DM:1E-6
1 : SES:1E-4/DM:1E-8

Application example

Program

・Sets the threshold to SES:1E-3/DM:1E-6.

```
> ETH△0
```

Query

・Queries when the threshold is SES:1E-4/DM:1E-8.

```
> ETH?
< ETH△1
```

・Queries when Eye Margin measurement is set.

```
> ETH?
< ERR
```

7.5.3.6. TEI Threshold EI / %EFI Measurement ON/OFF (Threshold EI / %EFI measurement)

Function Sets/queries ON/OFF for threshold EI/%EFI measurement.

Header	Program	Query	Response (No. of characters)
TEI	TEI△m	TEI?	TEI△m (FIX1)

Parameter m = <NR1>
0 : ON
1 : OFF

Application example

Program

・Sets Threshold EI/%EFI measurement to ON.

```
> TEI△0
```

Query

・Queries when Threshold EI/%EFI measurement is OFF.

```
> TEI?
< TEI△1
```

・Queries when Eye Margin measurement is set.

```
> TEI?
< ERR
```

7.5.3.7. PFM Power Fail Measurement ON/OFF(Power fail measurement)

Function Sets/queries ON/OFF for Power Fail measurement.

Header	Program	Query	Response (No. of characters)
PFM	PFM△m	PFM?	PFM△m (FIX1)

Parameter m = <NR1>
0 : ON
1 : OFF

Application example

Program

・Sets Power Fail measurement to ON.

```
> PFM△0
```

Query

・Queries when Power Fail measurement is OFF.

```
> PFM?
< PFM△1
```

・Queries when Eye Margin measurement is set.

```
> PFM?
< ERR
```

7.5.3.8. ER Error Ratio Measurement Result (Error ratio?)

Function Outputs an error ratio measurement result.
The output data is a value (Last/Current) specified in Display on the Result sub-screen. Error type is a value specified in ETY. If no value is specified, the parameter returns Total when Error Type = Total and returns Insertion when Error Type = INS/OMI.

Header	Program	Query	Response (No. of characters)
ER	None	ER?	ER△△m (FIX10)

Response m = <NR3>

```
*.****E-**
```

When no measurement data exists because of change of measurement requirements (displayed as `****`) or when no measurement data is generated (displayed as `----`), the query returns `0.0000E-00`.

Application example

Query

・-- Queries when error ratio is 1.05E-06.

```
> ER?
< ER△△1.0500E-06
```

・Queries when error ratio measurement result is [****] (no data exists).

```
> ER?
< ER△△0.0000E-00
```

・Queries when Eye Margin measurement is set.

```
> ER?
< ERR
```

7.5.3.9. EC Error Count Measurement Result (Error count?)

Function Outputs an error count measurement result.
The output data is an item (Last/Current) specified in Display on the Result sub-screen. Error type is a value specified in ETY. If no value is specified, the parameter returns Total when Error Type = Total and returns Insertion when Error Type = INS/OMI.

Header	Program	Query	Response (No. of characters)
EC	None	EC?	EC△△m (FIX10)

Response m = <NR1 or NR3>
Returns a 10-character value in the following format when the error count is below 1E+9.
`**********  (0 to 999999999)`
Returns a value in the following format when the error count is 1E+9 or more.
`*.****E+**  (1.0000E+09 to 9.9999E+16)`
When no measurement data exists because of change of measurement requirements (displayed as `****`) or when no measurement data is generated (displayed as `----`), the query returns `1.0000E-99`.

Application example
Query
・Queries when error count is 1.05E+09.
```
> EC?
< EC△△1.0500E+09
```
・Queries when error count is 156.
```
> EC?
< EC△△△△△△△△△156
```
・Queries when error ratio measurement result is [****] (no data exits).
```
> EC?
< EC△△1.0000E-99
```
・Queries when Eye Margin measurement is set.
```
> EC?
< ERR
```

7.5.3.10. CC Clock count measurement result

Function Outputs the clock count measurement result.
The item (Last/Current) specified as the Display item in the Result sub-window is output.

Header	Program	Query	Response (No. of characters)
CC	None	CC?	CC△△m (FIX10)

Parameter m = < NR1 or NR3>

When the clock count is less than 1E+9, the response will be in the following format with 10 characters (fixed).

```
**********  (0 to 999999999)
```

When the clock count is 1E+9 or more, the response will be in the following format.

```
*.****E+**  (1.0000E+09 to 9.9999E+16)
```

1.0000E-99 will be returned when no measurement data exists (display: ****) or no measurement data has been generated (display: ----).

Application example

Query

・Querying when the clock count is 1.05E+9

```
> CC?
< CC△△1.0500E+09
```

・Querying when the clock count is 156

```
> CC?
< CC△△△△△△△△△156
```

・Querying when no data exists

```
> CC?
< CC△△1.0000E-99
```

・Querying when Eye Margin measurement

```
> CC?
< ERR
```

7.5.3.11. EI Error Interval Count Measurement Result (Error interval?)

Function Outputs an error interval count measurement result.
The output data is a value (Last/Current) specified in Display on the Result sub-screen.

Header	Program	Query	Response (No. of characters)
EI	None	EI?	EI△△m (FIX10)

Response m = <NR1 or NR3>
****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), the query returns 1.0000E-99.

Application example

Query

- Queries when error count is 156

```
> EI?
< EI△△△△△△△△△156
```

- Queries when error ratio measurement result is [****] (no data exits).

```
> EI?
< EI△△1.0000E-99
```

- Queries when Eye Margin measurement is set.

```
> EI?
< ERR
```

7.5.3.12. EFI Error Free Interval Rate Measurement Result (Error free interval?)

Function Outputs an error free interval rate measurement result.
The output data is a value (Last/Current) specified in Display on the Result sub-screen.

Header	Program	Query	Response (No. of characters)
EFI	None	EFI?	EFI△m (FIX8)

Response m = <NR2 or NR3>
△△***.****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), the query returns 1.0000E-99.

Application example

Query

- Queries when error free interval rate is 99.01%.

```
> EFI?
< EFI△△△△99.0100
```

- Queries when error ratio measurement result is [****] (no data exits).

```
> EFI?
< EFI△1.0000E-99
```

- Queries when Eye Margin measurement is set.

```
> EFI?
< ERR
```

7.5.3.13. FRQ Clock Frequency Measurement Result (Clock frequency?)

Function Outputs a clock frequency measurement result.

Header	Program	Query	Response (No. of characters)
FRQ	None	FRQ?	FRQ△m (FIX10)

Response m = <NR2 or NR3>

******.***

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), the query returns 0.0000.

Application example

Query

・Queries when clock frequency is 3200 MHz.

```
> FRQ?
< FRQ△△△3200.000
```

・Queries when clock frequency measurement result is [****] (no data exits).

```
> FRQ?
< FRQ△△△△△△0.000
```

・Queries when Eye Margin measurement is set.

```
> FRQ?
< ERR
```

7.5.3.14. PFP Performance % Measurement Result (Performance percent?)

Function Outputs % data among from the performance measurement results.

Header	Program	Query	Response (No. of characters)
PFP	None	PFP?	PFP△m0,m1,m2,m3,m4 (FIX10)

Response m0 = <NR2 or NR3>
%ES data
△△***.****
m1 = <NR2 or NR3>
%EFS data
△△***.****
m2 = <NR2 or NR3>
%SFS data
△△***.****
m3 = <NR2 or NR3>
%US data
△△***.****
m4 = <NR2 or NR3>
%DM data
△△***.****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), responses m0 to m4 return 1.0000E-99.

Application example

Query

・Queries when data other than %DM exists.

```
> PFP?
<  PFP △ △ △ △ 99.1234, △ △ △ 99.1234, △ △ △ 99.1234, △ △ △
  99.1234,1.0000E-99
```

・Queries when Eye Margin measurement is set.

```
> PFP?
< ERR
```

7.5.3.15. PFC Performance Count Measurement Result (Performance count?)

Function Outputs Count data among from the performance measurement results.

Header	Program	Query	Response (No. of characters)
PFC	None	PFC?	PFC△m0,m1,m2,m3,m4,m5 (FIX10)

Response m0 = <NR1 or NR3>
ES data
****
m1 = <NR1 or NR3>
EFS data
****
m2 = <NR1 or NR3>
SFS data
****
m3 = <NR1 or NR3>
US data
****
m4 = <NR1 or NR3>
DM data
****
m5 = <NR1 or NR3>
EC data
****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), responses m0 to m5 return 1.0000E-99.

Application example

Query

・Queries when data other than DM exists.

```
> PFP?
< PFP△△123991234,△△15991234,△△39941234,△145671234,△△
  45671234,1.0000E-99
```

・Queries when Eye Margin measurement is set.

```
> PFP?
< ERR
```

7.5.3.16. THE Threshold EI Measurement Result (Threshold EI measurement?)

Function Outputs a threshold EI measurement result.

Header	Program	Query	Response (No. of characters)
THE	None	THE?	THE△m0,m1,m2,m3,m4,m5,m6 (FIX10)

Response

m0 = <NR1 or NR3>
>1E-03 data
****

m1 = <NR1 or NR3>
>1E-04 data
****

m2 = <NR1 or NR3>
>1E-05 data
****

m3 = <NR1 or NR3>
>1E-06 data
****

m4 = <NR1 or NR3>
>1E-07 data
****

m5 = <NR1 or NR3>
>1E-08 data
****

m6 = <NR1 or NR3>
=<1E-08 data
****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), responses m0 to m6 return 1.0000E-99.

Application example

Query

・Queries when =<1E-08,>1E-08,>1E-06 data exists.

```
> THE?
< THE△1.0000E-99,1.0000E-99,1.0000E-99,△△34567890,
  1.0000E-99,△△39941234,△139941234
```

・Queries when Eye Margin measurement is set.

```
> THE?
< ERR
```

7.5.3.17. THF Threshold %EFI Measurement Result (Threshold %EFI measurement?)

Function Outputs a Threshold %EFI measurement

Header	Program	Query	Response (No. of characters)
THF	None	THF?	THF△m0,m1,m2,m3,m4,m5,m6 (FIX10)

Response m0 = <NR2 or NR3>
>1E-03 data
△△***.****
m1 = <NR2 or NR3>
>1E-04 data
△△***.****
m2 = <NR2 or NR3>
>1E-05 data
△△***.****
m3 = <NR2 or NR3>
>1E-06 data
△△***.****
m4 = <NR2 or NR3>
>1E-07 data
△△***.****
m5 = <NR2 or NR3>
>1E-08 data
△△***.****
m6 = <NR2 or NR3>
=<1E-08 data
△△***.****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), responses m0 to m6 return 1.0000E-99.

Application example

Query

・Queries when =<1E-08,>1E-08,>1E-06 data exists.

```
> THE?
< THE△1.0000E-99,1.0000E-99,1.0000E-99,△△△34.5678,
  1.0000E-99,△△△39.9412,△△△13.9941
```

・Queries when Eye Margin measurement is set.

```
> PFP?
< ERR
```

7.5.3.18. AIN Alarm Interval Measurement Result (Alarm interval?)

Function Outputs an alarm interval measurement result.

Header	Program	Query	Response (No. of characters)
AIN	None	AIN?	AIN△m0,m1,m2 (FIX10)

response m0 = <NR1 or NR3>
Pattern Sync Loss data
****

m1 = <NR2 or NR3>
Clock Loss data
****

m2 = <NR2 or NR3>
Power Fail data
****

When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), responses m0 to m2 return 1.0000E-99.

Application example

Query

・Queries when data other than Power Fail exists.

```
> AIN?
< AIN△△△△△△△△234, △△△△△△△△34,1.0000E-99
```

・Queries when Eye Margin measurement is set.

```
> AIN?
< ERR
```

7.5.3.19. OER One Second Average Error Ratio Measurement Result (One second error ratio?)

Function Outputs a one-second error ratio measurement result.
Error type is a value specified in ETY. If no value is specified, the command returns Total when Error type = Total and returns Insertion when Error Type = INS/OMI.

Header	Program	Query	Response (No. of characters)
OER	None	OER?	OER△m (FIX10)

Response m = <NR3>
*.****E-**
When no measurement data exists because of change of measurement requirements (displayed as ****) or when no measurement data is generated (displayed as ----), the query returns 0.0000E-00.

Application example

Query

・Queries when a one-second average error ratio is 1.05E-06.

```
> OER?
< OER△1.0500E-06
```

・Queries when a one-second average error ratio measurement result is [****] (no data exists).

```
> OER?
< OER△△0.0000E-00
```

・Queries when Eye Margin measurement is set.

```
> OER?
< ERR
```

7.5.3.20. OEC One Second Average Error Count Measurement Result (One second error count?)

Function Outputs a one-second average error count measurement result. Error type is a value specified in ETY. If no value is specified, the command returns Total when Error type = Total and returns Insertion when Error Type = INS/OMI.

Header	Program	Query	Response (No. of characters)
OEC	None	OEC?	OEC△m (FIX10)

Response m = <NR1 or NR3>

- Returns a 10-character value in the following format when the one-second average error count is below 1E+9.
  `********** (0 to 999999999)`
- Returns a value in the following format when the error count is 1E+9 or more.
  `*.****E+** (1.0000E+09 to 9.9999E+16)`
  When no measurement data exists because of change of measurement requirements (displayed as `****`) or when no measurement data is generated (displayed as `----`), the query returns 1.0000E-99.

Application example

Query

- Queries when the one-second average error count is 1.05E+08.
  `> OEC?`
  `< OEC△1.0500E+09`
- Queries when the one-second average error count is 156.
  `> OEC?`
  `< OEC△△△△△△△△156`
- Queries when a one-second average error ratio measurement result is [****] (no data exits).
  `> OEC?`
  `< OEC△△1.0000E-99`
- Queries when Eye Margin measurement is set.
  `> OEC?`
  `< ERR`

7.5.3.21. FMT Data Output Format (Output data format)

Function Selects/queries data output on GPIB and output items of the printout data when outputting Error/Alarm measurement results.

Header	Program	Query	Response (No. of characters)
FMT	FMT△m	FMT?	FMT△m (FIX1)

Parameter m = <NR1>

0 : Standard format

1 : Short format

2 : Other format (for query only)

For the standard and short format output on GPIB, see the IMD command. When specifying the output format using this command, the items to be printed by the Print command are changed as below.

Standard format: Error and Alarm Interval are selected.

Short format: Error is selected and Alarm Interval is not selected.

Other format: Error is not selected.

Application example

Program

・-- Sets the standard format.

```
> FMT△0
```

Query

・Queries when the short format is set.

```
> FMT?
< FMT△1
```

・Queries when Eye Margin measurement is set.

```
> FMT?
< ERR
```

7.5.3.22. IMD Intermediate Measurement Result Output Function (Intermediate measure data output?)

Function Outputs an intermediate measurement result.
Error type is a value specified in ETY. If no value is specified, the command returns Total when Error type = Total and returns Insertion when Error Type = INS/OMI.

Header	Program	Query	Response (No. of characters)
IMD	None	IMD?	IMD△m

Response m = <NR1,NR2,NR3>

When selecting the standard format in the FMT command:

```
STA△**,**,**,**,**,**(CR/LF)
```
(Measurement start time in the query reception period; in YMDHMS order)
```
INT△**,**,**,**,**,**(CR/LF)
```
(Final measurement time ; in YMDHMS order)*1
```
END△**,**,**,**,**,**,(CR/LF)
```
(Final measurement time; in YMDHMS order)*2

```
ELP△*****,**,**,**(CR/LF)   (Elapsed time)
ER△△*.****E-**(CR/LF)       Error rate
EC△△**********(CR/LF)       Error count (When the count is below 1E+9)
EC△△*.****E+**(CR/LF)       Error count (When the ratio is 1E+9 or more)
CC△△**********(CR/LF)       Clock count (When the count is below 1E+9)
CC△△*.****E+**(CR/LF)       Clock count (When the ratio is 1E+9 or more)
EI△△**********(CR/LF)       Error interval count
EFI△△△***.****(CR/LF)       Error free interval rate
PFI△**********(CR/LF)       Power fail interval count
CLI△**********(CR/LF)       Clock loss interval count
SLI△**********(CR/LF+EOI)   Sync loss interval count
```

When selecting the short format in the FMT command

```
STA△**,**,**,**,**,**(CR/LF)   (Measurement start time)
INT△**,**,**,**,**,**(CR/LF)   (Final measurement time)
ER△△*.****E-**(CR/LF)          Error ratio
EC△△**********(CR/LF)          Error count (When the count is below 1E+9)
EC△△*.****E+**(CR/LF)          Error count (When the ratio is 1E+9 or more)
CC△△**********(CR/LF+EOI)      Clock count (When the count is below 1E+9)
CC△△*.****E+**(CR/LF+EOI)      Clock count (When the ratio is 1E+9 or more)
```

Either *1 or *2 is output.
*1 is output when querying within 1 second after start of measurement until end of measurement. *2 is output when querying within 1 second upon completion of measurement. ERR is issued when querying within 1 second after start of measurement without data left in the previous measurement zone because the measurement start command or setting may change.

Since EOI cannot be detected in RS-232C mode, it becomes necessary to read the data up to the line of SLI or CC.

7.5.3.22. IMD Intermediate Measurement Result Output Function (Intermediate measure data output?) (continued)

When the threshold EI/%EFI data print function is ON and Threshold EI/%EFI measurement = ON in "Test Menu: Measurement Condition", the lines listed below are output after the above format:

TH3△**********,***.****(CR/LF)(Threshold EI,Threshold %EFI : > E-3)
TH4△**********,***.****(CR/LF)(Threshold EI,Threshold %EFI : > E-4)
TH5△**********,***.****(CR/LF)(Threshold EI,Threshold %EFI : > E-5)
TH6△**********,***.****(CR/LF)(Threshold EI,Threshold %EFI : > E-6)
TH7△**********,***.****(CR/LF)(Threshold EI,Threshold %EFI : > E-7)
TH8△**********,***.****(CR/LF)(Threshold EI,Threshold %EFI : > E-8)
TH9△**********,***.****(CR/LF+EOI)(Threshold EI,Threshold %EFI : =<E-8)

When the error performance data print function is ON and Performance measurement = ON in "Test Menu: Measurement Condition", the lines listed below are output after the above format:

ES△△△△***.****(CR/LF) (%ES)
EFS△△△***.****(CR/LF) (%EFS)
SES△△△***.****(CR/LF) (%SES)
DM△△△△***.****(CR/LF) (%DM)
US△△△△***.****(CR/LF+EOI) (%US)

When the frequency measurement print function is ON, the lines listed below are output after the above format:

FRQ△******.***(CR/LF+EOI)

If no data is specified in the FMT command, the data is output in the standard format.

In the above format, EOI is appended to the item to be finally output only.

Application example

Query

・-- Queries when the short format is set.

```
> IMD?
< STA△98,05,28,08,00,00(CR/LF)
< INT△98,05,28,08,00,06(CR/LF)
< ER△△1.2345E-06(CR/LF)
< EC△△1.2355E+05(CR/LF)
< CC△△1.0000E+11(CR/LF+EOI)
```

・Queries when the standard format is set and the Threshold EI/%EFI print function is ON.

```
> IMD?
< STA△98,05,28,16,05,05(CR/LF)
< INT△98,05,28,16,05,06(CR/LF)
< ELP△△△△△0,00,00,01(CR/LF)
< ER△△0.0000E-06(CR/LF)
< EC△△△△△△△△△△0(CR/LF)
< CC△△1.0000E+06(CR/LF)
< EI△△△△△△△△△△0(CR/LF)
< EFI△△△100.0000(CR/LF)
< PFI△△△△△△△△△△0(CR/LF)
< CLI△△△△△△△△△△0(CR/LF)
< SLI△△△△△△△△△△0(CR/LF)
< TH3△△△△△△△△△△0,100.0000(CR/LF)
< TH4△△△△△△△△△△0,100.0000(CR/LF)
< TH5△△△△△△△△△△0,100.0000(CR/LF)
< TH6△△△△△△△△△△0,100.0000(CR/LF)
< TH7△△△△△△△△△△0,100.0000(CR/LF)
< TH8△△△△△△△△△△0,100.0000(CR/LF)
< TH9△△△△△△△△△△0,100.0000(CR/LF+EOI)
```

・Queries when Eye Margin measurement is set.

```
> IMD?
< ERR
```

7.5.3.23. EMM Eye Margin Measurement Result (Eye margin measurement result)

Function Queries an Eye Margin measurement result.

Header	Program	Query	Response (No. of characters)
EMM	None	EMM?△m	EMM△m1 (FIX4)

Parameter m = <NR1>

0 : Queries a Phase Margin measurement result.

1 : Queries a Threshold Margin measurement result.

Response m1 = <NR1>

When Threshold Margin is selected.

`EMM△****  : 0 to 8000 mVp-p`

When Phase Margin is selected

`EMM△****  : 0 to 2000 psp-p`

When no measurement data exists:

`EMM△----` is returned.

When a measurement is failed, `EMM△Fail` is returned.

Application example

Query

・Queries Threshold Margin.

```
> EMM?△1
< EMM△1200
```

・Queries when Error/Alarm measurement is set.

```
> EMM?
< ERR
```

7.5.3.24. EDM Eye Diagram Measurement Result (Eye diagram measurement result)

Function Query an Eye Diagram measurement result.

Header	Program	Query	Response (No. of characters)
EDM	None	EDM?△m,m0,m1	EDM△m2,m3 (FIX5)

Parameter m = <NR1>

0 : Queries an Eye Diagram measurement result.

1 : Queries an Recall Eye Diagram display result.

m0 = <NR1>

3 to 12 Specifies error ratio (1E-3 to 1E-12).

m1 = <NR1>

1 to 64 Specifies the measurement point (1 to 64)

Response m2 = <NR1>

Phase

***** : -1000 to +1000 ps

When no measurement data exists, ---- is returned.

m3 = <NR1>

Threshold

***** : -4000 to +4000 mV

When no measurement data exists, ---- is returned.

Application example

Query

・Queries an Eye Margin measurement result.

```
> EDM?△0,3,1
< EDM△△+400,△-230
```

・Queries when Error/Alarm measurement is set.

```
> EDM?△0,3,1
< ERR
```

7.5.3.25. BST Measurement Result Store Function (Measurement buffer store)

Function Stores the specified measurement data in the read buffer.

The details are described below:

- The stored data is retained until it is cleared or the power supply is turned OFF.
- The data is stored in the buffer in the same timing as when the BST command is received.
- The data to be stored is shown as below:

Header	Program	Query	Response (No. of characters)
BST	BST△m	None	None

Parameter m = <NR1>

0 : Last

1 : Intermediate

Last is the last data displayed on the screen when the command is received. Intermediate is the intermediate data created immediately before the command is received.

Application example

Program

- Stores the measurement result buffer.

```
> BST△
```

7.5.3.26. BCL Measurement Result Store Buffer Clear Function (Measurement buffer clear)

Function Clears the read buffer for the specified measurement data.

Header	Program	Query	Response (No. of characters)
BCL	BCL	None	None

Parameter None

Application example

Program

- Clear the buffer in which measurement results are stored.

```
> BCL
```

7.5.3.27. CLI Clock Loss Status (Clock loss interval)

Function Queries whether Clock Loss occurs or not.

Header	Program	Query	Response (No. of characters)
CLI	None	CLI?	CLI△m (FIX1)

Parameter m = <NR1>

0 : Clock Loss does not occur.

1 : Clock Loss occurs.

Application example

Query

- Queries when Clock Loss occurs.

```
> CLI?
< CLI△1
```

7.5.3.28. SLI Sync Loss Status (Sync loss interval)

Function Queries whether Sync Loss occurs or not.

Header	Program	Query	Response (No. of characters)
SLI	None	SLI?	SLI△m (FIX1)

Parameter m = <NR1>
0 : Sync Loss does not occur.
1 : Sync Loss occurs.

Application example

Query

・Queries when Sync Loss occurs.

```
> SLI?
< SLI△1
```

・Queries when Eye Margin measurement is set.

```
> SLI?
< ERR
```

7.5.3.29. ERS Error Detection Status (Errors?)

Function Queries whether an error is detected or not.

Header	Program	Query	Response (No. of characters)
ERS	None	ERS?	ERS△m (FIX1)

Parameter m = <NR1>
0 : An error is not detected.
1 : An error is detected.
When Error Type of Test Menu is INS/OMI, the parameter returns OR of INS error and OMI error.

Application example

Query

・Queries when an error is detected.

```
> ERS?
< ERS△1
```

・Queries when Eye Margin measurement is detected.

```
> ERS?
< ERR
```

7.5.3.30. BES Error Detection Status(Bit Errors?)

Function Error Detection Status

Header	Program	Query	Response (No. of characters)
BES	None	BES?△m0	BES△m1 (FIX1)

Parameter m0 = <NR1>
Queries whether an error is detected or not.
0 : Total Error
1 : Insertion
2 : Omission

Response m1 = <DECIMAL NUMERIC DATA>
0 : An error is not detected.
1 : An error is detected.

Application example

Query

・Queries when a INS error is detected.

```
> BES?△1
< BES△1
```

・Queries when Eye Margin measurement is set.

```
> BES?
< ERR
```

7.5.3.31. POF Power Fail Status (Power fail?)

Function Queries whether power fail occurred in the previous measurement interval.

Header	Program	Query	Response (No. of characters)
POF	None	POF?	POF△m (FIX1)

Parameter m = <NR1>
0 : Power fail does not occur.
1 : Power fail occurs.
Updates the information on a measurement interval basis.

Application example

Query

・Queries the measurement interval immediately after power fail has been recovered.

```
> POF?
< POF△1
```

・Queries when System mode is set to Normal mode.

```
> POF?
< ERR
```

7.5.3.32. DTR Delay Trouble (Delay trouble4?)

Function Queries whether ED trouble occurs or not.

Header	Program	Query	Response (No. of characters)
DTR	None	DTR?	DTR△m (FIX1)

Parameter m = <NR1>
0 : Normal
1 : Delay trouble

Application example

Query
・Query when delay trouble occurs.
> DTR?
< DTR△1

7.5.3.33. DLY Delay Setting Status (Delay unlock?)

Function Queries ED delay setting status.

Header	Program	Query	Response (No. of characters)
DLY	None	DLY?	DLY△m (FIX1)

Parameter m = <NR1>
0 : Ready
1 : Returns when Delay Trouble4 occurs.

Application example

Query
Query the delay setting status.
> DLY?
< DLY△0

7.5.4. Output signal

This section describes setting/query on trigger output.

7.5.4.1. SOP Synchronous Signal Output Selection (Sync output)

Function Selects/queries the type of synchronous signal.

Header	Program	Query	Response (No. of characters)
SOP	SOP△m	SOP?	SOP△m (FIX1)

Parameter m = <NR1>
0 : 1/8 clock
1 : Pattern Sync
2 : Sync Gain/Loss

Application example

Program

・Sets synchronous signal type to 1/8 Clock.

```
> SOP△0
```

Query

・Queries when Pattern Sync is set.

```
> SOP?
< SOP△1
```

7.5.5. Grouping Related

This section describes setting/query on Grouping.

7.5.5.1. PCO PPG/ED Pattern Common Setting (PPG/ED pattern common setting)

Function Sets/queries ON/OFF of PPG/ED pattern common setting.

Header	Program	Query	Response (No. of characters)
PCO	PCO△m	PCO?	PCO△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets the common setting ON.

```
> PCO△1
```

Query

・Makes a query when the common setting is OFF.

```
> PCO?
```

7.6. Settings on other main frames

The commands described in this section perform settings/queries which are not related to window operation or file operation.

7.6.1. Window operation commands

These commands set/query the window operation.

7.6.1.1. WOC Child Window Open/Close (Window open / close)

Function Sets/queries open/close of the child window.

Header	Program	Query	Response (No. of characters)
WOC	WOC△m	WOC?	WOC△m (FIX7)

Parameter m = <binary format>

Flags 1 to a bit corresponding to the child window to be opened. To close the child window, flags 0 to the bit.

#B00000 : All windows are closed.
#B10000 : Only the System child window is opened.
#B01000 : Only the Setup child window is opened.
#B00100 : Only the Test Menu child window is opened.
#B00010 : Only the Result child window is opened.
#B00001 : Only the Customize child window is opened.
#B11111 : All windows are opened.

Application example

Program

・ Opens the Result child window and the Setup child window.

```
> WOC△#B01010
```

Query

・ Queries when the System child window and the Customize child window are open.

```
> WOC?
< WOC△#B10001
```

7.6.1.2. SYW System Child Window panel selection (System sub window)

Function Selects/queries a panel within the System child window.

Header	Program	Query	Response (No. of characters)
SYW	SYW△m	SYW?	SYW△m (FIX1)

Parameter m = <NR1>
0 : System panel
1 : Option panel

Application example

Program

・Sets the setup System panel
> SYW△0

Query

・ Sets the System sub window.
> SYW?
< SYW△1

・ Queries when the Option sub window is set.
> SYW
< ERR

7.6.1.3. SEW Setup Child Window Panel Selection (Setup sub window)

Function Selects/queries a panel within the Setup child window.

Header	Program	Query	Response (No. of characters)
SEW	SEW△m	SEW?	SEW△m (FIX1)

Parameter m = <NR1>
0 : Setup panel
1 : Frequency panel
2 : Clock I/F panel
3 : Data I/F panel
4 : Pattern panel
5 : Trigger I/F panel
6 : Utility panel

Application example

Program

・ Sets the Setup panel.
> SEW△0

Query

・ Queries when the Pattern panel is set.
> SEW?
< SEW△4

・ Query when the Setup child window is not open.
> SEW
< ERR

7.6.1.4. TEW Test Menu Child Window Panel Selection (Test Menu sub window)

Function Selects/queries a panel within the Test Menu child window.

Header	Program	Query	Response (No. of characters)
TEW	TEW△m	TEW?	TEW△m (FIX1)

Parameter m = <NR1>
0 : Measurement panel
1 : Error Addition panel

Application example

Program

・ Sets the Measurement sub window.

```
> TEW△0
```

Query

・ Queries when the Error Addition sub window is set.

```
> TEW?
< TEW△1
```

・ Queries when the Test Menu child window is not open.

```
> TEW
< ERR
```

7.6.1.5. TIM Real Time/Measurement Time Display Selection (Real time or measurement time)

Function Sets/queries the measurement time display mode for the Result child window.

Header	Program	Query	Response (No. of characters)
TIM	TIM△m	TIM?	TIM△m (FIX1)

Parameter m = <NR1>
0 : Date & Time display
3 : Timed display
4 : Elapsed display
5 : Start display

Application example

Program

・Sets the time display mode to Elapsed.

```
> TIM△4
```

Query

・Queries when Timed is set.

```
> TIM?
< TIM△3
```

・Queries when Test Item is set to Eye Margin measurement and Display is set to Recall Diagram.

```
> TIM
< ERR
```

7.6.1.6. REW Result Child Window Sub Window Selection (Result sub window)

Function Selects/queries a panel within the Test Menu child window.

Header	Program	Query	Response (No. of characters)
REW	REW△m	REW?	REW△m (FIX1)

Parameter m = <NR1>
0 : ALL panel
1 : Zoom panel
2 : Monitor panel

Application example

Program

・ Sets All panel ON.

```
> REW△0
```

Query

・ Queries when Monitor sub window is set.

```
> REW?
< REW△2
```

・ Queries when the Result child window is not open.

```
> REW
< ERR
```

7.6.1.7. CUR Intermediate Result Display Function (Current data)

Function Sets/queries the display mode for the Result: Error/Alarm panel.

Header	Program	Query	Response (No. of characters)
CUR	CUR△m	CUR?	CUR△m (FIX1)

Parameter m = <NR1>
0 : Last data display
1 : Current data display

Application example

Program

・ Changes to Current display.

```
> CUR△1
```

Query

・ Queries when Last display is set.

```
> CUR?
< CUR△1
```

・ Queries when Eye Margin measurement is set.

```
> CUR
< ERR
```

7.6.1.8. PFD Performance Result Display Selection (Performance data display select)

Function Sets/queries the classification of Performance item to be displayed on the Result:Error/Alarm panel (ALL secondly panel).

Header	Program	Query	Response (No. of characters)
PFD	PFD△m	PFD?	PFD△m (FIX1)

Parameter m = <NR1>
0 : Count item
1 : Percent item

Application example

Program

・ Changes to Count display.

```
> PFD△0
```

Query

・ Queries when Percent display is set.

```
> PFD?
< PFD△1
```

・ Queries when Eye Margin measurement is set.

```
> PFD
< ERR
```

7.6.1.9. ZOM Zoom window display ON/OFF setting (Zoom window display)

Function Sets/queries ON/OFF of Display 1 to 4 for the Zoom secondary panel.

Header	Program	Query	Response (No. of characters)
ZOM	ZOM△m0,m1	ZOM?△m0	ZOM△m0,m1 (FIX1)

Parameter m0 = <NR1>
0 : display1
1 : display2
2 : display3
3 : display4
m1 = <NR1>
0 : ON
1 : OFF

Application example

Program

・ Sets Display 1 to ON.

```
> ZOM△0,0
```

Query

・ Queries when Display 3 is OFF.

```
> ZOM?△2
< ZOM△2,1
```

・ Queries when Eye Margin measurement is set.

```
> ZOM?△2
< ERR
```

7.6.1.10. ZOI Zoom Window Display Item Setting (Zoom window display Item)

Function Sets/queries the display item for the Result:Error/Alarm panel (Zoom panel).

Header	Program	Query	Response (No. of characters)
ZOI	ZOI△m0,m1	ZOI?△m0	ZOI△m0,m1 (FIX1,FIX2)

Parameter m0 = <NR1>
0 : display1
1 : display2
2 : display3
3 : display4
m1 = <NR1>
0 : Error Ratio
1 : Error Count
2 : Error Interval
3 : %Error Free Interval
4 : Pattern Sync Loss
5 : Clock Loss
6 : Power Fail
7 : Threshold EI
8 : Threshold %EFI
9 : Performance
10 : Frequency

Application example

Program

- Displays ER in Display 1.

```
> ZOI△0,0
```

Query

- Queries when Performance is displayed in Display 3.

```
> ZOI?△2
< ZOI△2,△9
```

- Queries when Eye Margin measurement is set.

```
> ZOI?△1
< ERR
```

7.6.1.11. ZOP Zoom Window Performance Item Setting (Zoom window display performance Item)

Function Sets/queries the display item when the Result: Error/Alarm panel (Zoom panel) is set in Performance.

Header	Program	Query	Response (No. of characters)
ZOP	ZOP△m0,m1	ZOP?△m0	ZOP△m0,m1 (FIX1,FIX2)

Parameter m0 = <NR1>
0 : display1
1 : display2
2 : display3
3 : display4
m1 = <NR1>
0 : ES
1 : EFS
2 : SES
3 : DM
4 : US
5 : %ES
6 : %EFS
7 : %SES
8 : %DM
9 : %US
10 : EC

Application example

P*rogram*

・Displays ES in Display 1.

```
> ZOP△0,0
```

Query

・ Queries when Performance is displayed in Display 3.

```
> ZOP?△2
< ZOP△2,△9
```

・ Queries when Performance measurement = OFF.

```
> ZOP?△1
< ERR
```

7.6.1.12. ZOT Zoom Window Threshold Ei/%Efi Setting (Zoom window display Threshold EI/%EFI)

Function Sets/queries the display item when the display item of the Result: Error/Alarm panel (Zoom panel) is set to Threshold EI or Threshold %EFI.

Header	Program	Query	Response (No. of characters)
ZOT	ZOT△m0,m1	ZOT?△m0	ZOT△m0,m1 (FIX1)

Parameter m0 = <NR1>
0 : display1
1 : display2
2 : display3
3 : display4
m1 = <NR1>
0 : >1.0E-3
1 : >1.0E-4
2 : >1.0E-5
3 : >1.0E-6
4 : >1.0E-7
5 : >1.0E-8
6 : =<1.0E-8

Application example

Program

・ Displays >1.0E-3 in Display 1.

```
> ZOT△0,0
```

Query

・ Queries when Threshold EI is displayed in Display 3.

```
> ZOT?△2
< ZOT△2,△9
```

・ Queries when Threshold EI/%EFI measurement = OFF.

```
> ZOT?△1
< ERR
```

7.6.1.13. HRE History Lamp Reset (History Lamp Reset)

Function Resets History Lamp on Result:Monitor panel.

Header	Program	Query	Response (No. of characters)
HRE	HRE	None	None

Parameter None

Application example

Program

・Resets the Histary Lamp.

```
> HRE
```

7.6.1.14. DER Eye Margin Measurement Display Error Ratio (Eye margin display error ratio)

Function Sets/queries the display error ratio for the Result: Eye Margin panel (Diagram). The result selected in Display Mode is to be set/queried.

Header	Program	Query	Response (No. of characters)
DER	DER△m	DER?	DER△m (FIX12)

Parameter m = <binary format>

Flags 1 to a bit corresponding to the error ratio to be displayed. To hide the display, flags 0 to the bit. (MSB corresponds to 1E-3. LSB corresponds to 1E-12.)

#B1000000000 : Displays 1E-3 only.
#B0100000000 : Displays 1E-4 only.
#B1010011111 : Displays 1E-3, 1E-5 and 1E-8 to 1E-12.
#B1111111111 : Display all error ratio data.

Application example

Program

・ Displays Eye Diagram result at the rate of 1E-4 and 1E-6.

```
> DER△#B0101000000
```

Query

・ Queries when 1E-3 and 1E-6 are displayed.

```
> DER?
< DER△#B1001000000
```

・ Queries when Error/Alarm measurement is set.

```
> DER?
< ERR
```

7.6.1.15. EDD Eye Diagram Display Window Selection (Eye diagram display select)

Function Selects/queries the display for the Result:Eye Margin panel (Diagram).

Header	Program	Query	Response (No. of characters)
EDD	EDD△m	EDD?	EDD△m (FIX1)

Parameter m = <NR1>

0 : Result
1 : Recall Diagram

Application example

Program

・ Sets the Result window for the Result:Eye Margin display.

```
> EDD△0
```

Query

・ Queries when the Recall Diagram window is set.

```
> EDD?
< EDD△1
```

・ Queries when Error/Alarm measurement is set.

```
> EDD
< ERR
```

7.6.1.16. ASC Eye Diagram Display Auto Scale (Eye diagram display auto scale)

Function Serves as auto scale execution trigger for the display area in the Result: Eye Margin panel. The result selected in Display Mode is to be triggered.

Header	Program	Query	Response (No. of characters)
ASC	ASC	None	None

Parameter None

Application example

Program

・ Executes auto scale.

```
> ASC
```

7.6.1.17. VSC Eye Diagram Voltage Scale Setting (Eye diagram voltage scale)

Function Sets/queries the vertical scale (Voltage) in the graph in the Result: Eye Margin sub window. The result selected in Display Mode is to be set/queried.

Header	Program	Query	Response (No. of characters)
VSC	VSC△m0,m1	VSC?	VSC△m0,m1 (FIX6,FIX5)

Parameter m0 = <NR2>

Threshold voltage minimum value

-4.000 to 3.990 : -4.000 to 3.990 V/Step:0.001

m1 = <NR2>

Axial step

0.001 to 0.800 : 0.001 to 0.800 V/Step:0.001

Application example

Program

・ Sets the minimum value to -4.0 V and step to 0.8 V.

```
> VSC△-4,0.8
```

Query

・ Queries when the minimum value is 1.0 V and step is 0.1 V.

```
> VSC?
< VSC△△1.000,0.100
```

・ Queries when Test Menu: MEAS. Mode is margin.

```
> VSC?
< ERR
```

7.6.1.18. PSC Eye Diagram Phase Scale Setting (Eye diagram phase scale)

Function Sets/queries the horizontal scale (Phase) in the graph in the Result: Eye Margin panel. The result selected in Display Mode is to be set/queried.

Header	Program	Query	Response (No. of characters)
PSC	PSC△m0,m1	PSC?	PSC△m0,m1 (FIX5,FIX3)

Parameter m0 = <NR1>
Threshold voltage minimum value
-1000 to 980 : -1000 to 980 ps/Step:2
m1 = <NR1>
Axial step
2 to 200 : 2 to 200 ps/Step:2

Application example

Program

・ Sets the minimum value to 100 ps and step to 50 ps.

```
> PSC△100,50
```

Query

・ Queries when the minimum value is 14 ps and step is 22 ps.

```
> PSC?
< PSC△△△△14,△22
```

・ Queries when Test Menu: MEAS. Mode is margin.

```
> PSC?
< ERR
```

7.6.1.19. EMA Eye Diagram Marker Display ON/OFF (Eye diagram marker on/off)

Function Sets/queries the marker ON/OFF in a graph in the Result: Eye Margin sub window diagram.

Header	Program	Query	Response (No. of characters)
EMA	EMA△m	EMA?	EMA△m (FIX1)

Parameter m = <NR1>
0 : Hides both basic and reference markers.
1 : Displays basic marker only.
2 : Displays both basic and reference markers.

Application example

Program

・ Hides both basic and reference markers.

```
> EMA△0
```

Query

・ Queries when basic maker is displayed.

```
> EMA?
< EMA△1
```

・ Queries when Error/Alarm measurement is set.

```
> EMA?
< ERR
```

7.6.1.20. MKM Marker Operation Mode (Marker mode)

Function Sets/queries the marker move mode for the Result: Eye Margin panel.

Header	Program	Query	Response (No. of characters)
MKM	MKM△m	MKM?	MKM△m (FIX1)

Parameter m = <NR1>
0 : Free
1 : Fix

Application example

Program

・ Sets the move mode to Free.

```
> MKM△0
```

Query

・ Queries when the Fix mode is set.

```
> MKM?
< MKM△1
```

・ Queries when Error/Alarm measurement is set.

```
> MKM?
< ERR
```

7.6.1.21. MKT Marker Specification Threshold (Marker threshold)

Function Sets/queries on which display error ratio line the marker in the Result: Eye Margin panel is to move. The result selected in Display Mode is to be set/queried.

Header	Program	Query	Response (No. of characters)
MKT	MKT△m	MKT?	MKT△m (FIX1)

Parameter m = <NR1>
3 : 1E-3
4 : 1E-4
5 : 1E-5
6 : 1E-6
7 : 1E-7
8 : 1E-8
9 : 1E-9
10 : 1E-10
11 : 1E-11
12 : 1E-12

7.6.1.21. MKT Marker Specification Threshold (Marker threshold)

Application example

Program

・Sets marker to move on 1E-3 line.

```
> MKT△3
```

Query

・ Queries when 1E-5 is specified.

```
> MKT?
< MKT△5
```

・ Query when the marker is hidden.

```
> MKT?
< ERR
```

7.6.1.22. MMV Marker Move Specification (Marker move)

Function Moves the marker in the Result: Eye Margin sub window. The result selected in Display Mode is to be moved.

Header	Program	Query	Response (No. of characters)
MMV	MMV△m0,m1	None	None

Parameter m0 = <NR1>

Marker to be moved

0 : Basic marker

1 : Reference marker

m1 = <NR1>

Moving direction

0 : Moves to left/rotates counterclockwise.

1 : Moves to right/rotates clockwise.

2 : Moves upward/rotates clockwise.

3 : Moves downward/rotates counterclockwise.

Application example

Program

・ Move basic marker upward.

```
> MMV△2
```

7.6.1.23. MKP Marker Position (Marker position)

Function Queries the marker position in the Result: Eye Margin sub window. For the basic marker, returns the absolute position. For the reference marker, returns the relative position to basic marker. The result selected in Display Mode is to be queried.

Header	Program	Query	Response (No. of characters)
MKP	None	MKP?△m0	MKP△m1,m2 (FIX6,FIX5)

Parameter m0 = <NR1>
0 : Basic marker
1 : Reference marker
m1 = <NR2>
Threshold voltage
-4.000 to 4.000 : -4.000 to 4.000V
m2 = <NR1>
Phase
-1000 to 1000 : -1000 to 1000ps

Application example

Query

- Queries the basic marker position.

```
> MKP?△0
< MKP△△4.000,△△100
```

- Queries when Error/Alarm measurement is set.

```
> MKP?△1
< ERR
```

7.6.1.24. CUS Customize Customize Window Display Slot Setting (Customize setup slot)

Function Sets/queries the unit to be displayed in Setup 1 to 6 frames in the customize child window.

Header	Program	Query	Response (No. of characters)
CUS	CUS△m0,m1	CUS?△m0	CUS△m0,m1 (FIX1)

Parameter m0 = <NR1>

Setup frame to be set

0 : setup1
1 : setup2
2 : setup3
3 : setup4
4 : setup5
5 : setup6

m1 = <NR1>

Slot to be set

0 : OFF
1 : Slot1
3 : Slot3
4 : Slot4

Application example

Program

- Sets Setup1 frame to OFF.

```
> CUS△0,0
```

Query

- Queries when Setup2 slot is set to Slot3.

```
> CUS?△1
< CUS△2,3
```

- Queries when the Customize child window is closed.

```
> CUS?△2
< ERR
```

7.6.1.25. CUI Customize Window Display Item Setting (Customize setup item)

Function Sets/queries the item to be displayed in Setup1 to 6 frames in the Customize child window.

Header	Program	Query	Response (No. of characters)
CUI	CUI△m0,m1	CUI?△m0	CUI△m0,m1 (FIX1,FIX2)

Parameter m0 = <NR1>

Setup frame to be set

0 : setup1
1 : setup2
2 : setup3
3 : setup4
4 : setup5
5 : setup6

m1 = <NR1>

Item to be set

0 : Frequency/Frequency
1 : Clock/Amplitude
2 : Clock/Offset
3 : Clock/Delay
4 : Clock/Polarity
5 : Clock/Duty
6 : Data/Amplitude
7 : Data/Offset
8 : Data/Cross Point
9 : Data/Threshold
10 : Pattern/Logic
11 : Pattern/Mark Ratio

Application example

Program

- Displays Frequency in Setup1 frame.

```
> CUI△0,0
```

Query

- Queries when the Setup2 item is Data/Threshold.

```
> CUI?△1
< CUI△1,△9
```

- Queries when Setup3 is OFF.

```
> CUI?△2
< ERR
```

7.6.1.26. CUP Customize Window Display Item Port Setting (Customize setup port)

Function Sets/queries the port to be displayed in Setup1 to 6 frames in the Customize child window.

Header	Program	Query	Response(No of characters)
CUP	CUP△m0,m1	CUP?△m0	CUP△m0,m1 (FIX1)

Parameter m0 = <NR1>
Setup frame to be set
0 : setup1
1 : setup2
2 : setup3
3 : setup4
4 : setup5
5 : setup6
m1 = <NR1>
Port to be set
0 : Clock
1 : XClock
2 : Data
3 : XData

Application example

Program

- Sets the port of Setup1 item to Data.

```
> CUP△0,2
```

Query

- Queries when the port of Setup2 item is XClock.

```
> CUP?△1
< CUP△1,1
```

- Queries when Setup3 does not have a port parameter.

```
> CUP?△2
< ERR
```

7.6.1.27. CPT Customize Window Pattern Setting Unit (Customize pattern setup unit)

Function Sets/queries the unit to be displayed in the Pattern frame in the customize child window.

Header	Program	Query	Response (No. of characters)
CPT	CPT△m	CPT?	CPT△m (FIX1)

Parameter m = <NR1>
Unit to be set
0 : OFF
1 : 3.2G PPG
2 : 3.2G ED

Application example

Program

- Sets 3.2 G PPG pattern for the target.

```
> CPT△1
```

Query

- Queries when 3.2G ED pattern is set for the target.

```
> CPT?
< CPT△2
```

- Queries when no unit exists in Slot3 and Slot4.

```
> CPT?
< ERR
```

7.6.1.28. CPF Customize Edit Pattern Position Specification (Customize pattern offset)

Function Sets/queries the offset position of the target pattern.

Header	Program	Query	Response (No. of characters)
CPF	CPF△m	CPF?	CPF△m (FIX7)

Parameter m = <NR1>
0 to 8388600 / step 1

Application example

Program

- Set offset to 8388600.

```
> CPF△8388600
```

Query

- Query when offset is 50.

```
> CPF?
< CPF△△△△△△50
```

- Query when the target pattern is PRBS.

```
> CPF?
< ERR
```

7.6.1.29. CRI Customize Window Measurement Display Item (Customize result display item)

Function Sets/queries the item to be displayed in the Result frame in the Customize child window.

Header	Program	Query	Response (No. of characters)
CRI	CRI△m	CRI?	CRI△m (FIX1)

Parameter m = <NR1>

Unit to be set

0 : Error Rate

1 : Error Count

Application example

Program

- Sets Error Rate.

```
> CRI△0
```

Query

- Queries when Error Count is set to be displayed.

```
> CRI?
< CRI△1
```

- Queries when no unit exists in Slot4.

```
> CRI?
< ERR
```

7.6.2. System setting commands

The commands described in this section set/query the buzzer and date.

7.6.2.1. SYS System Error Monitor Buzzer ON/OFF Setting (System error buzzer ON/OFF)

Function Sets/queries ON/OFF for the system error buzzer.

Header	Program	Query	Response (No. of characters)
SYS	SYS△m	SYS?	SYS△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・ Set the system error buzzer to ON.

```
> SYS△1
```

Query

・ Query when the system error buzzer is OFF.

```
> SYS?
< SYS△0
```

7.6.2.2. MON Error Monitor Buzzer ON/OFF Setting (Error monitor buzzer ON/OFF)

Function Sets/queries ON/OFF for the error buzzer.

Header	Program	Query	Response (No. of characters)
MON	MON△m	MON?	MON△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・ Sets the error buzzer to ON.

```
> MON△1
```

Query

・ Queries when the error buzzer is OFF.

```
> MON?
< MON△0
```

7.6.2.3. ALM Alarm Monitor Buzzer ON/OFF Setting (Alarm monitor buzzer ON/OFF)

Function Sets/queries ON/OFF for the alarm buzzer.

Header	Program	Query	Response (No. of characters)
ALM	ALM△m	ALM?	ALM△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・ Sets the alarm buzzer to ON.

```
> ALM△1
```

Query

・ Queries when the error buzzer is OFF.

```
> ALM?
< ALM△0
```

7.6.2.4. SYT System Error Type (System error type)

Function Sets/queries the type of system error which generates buzzer sound.

Header	Program	Query	Response (No. of characters)
SYT	SYT△m	SYT?	SYT△m (FIX6)

Parameter m = <binary format>
Flags 1 to a bit corresponding to the item to be generated. To cancel generation of the item, flags 0 to the bit.
#B1111 : Does not sound buzzer for all the system error items.
#B0000 : Sounds buzzer for PLL Unlock only.
#B1000 : Sounds buzzer for PPG Clock Loss only.
#B0100 : Sounds buzzer for PPG Clock Loss only.
#B0010 : Sounds buzzer for Delay Trouble (Slot3) only.
#B0001 : Sounds buzzer for Delay Trouble (Slot4) only.
#B1111 : Sounds buzzer for all the system error items.

Application example

Program

・ Set only PLL Unlock to ON.

```
> SYT△#B1000
```

Query

・ Query when PPG Clock Loss and Delay Trouble (Slot3) are ON.

```
> SYT?
< SYT△#B0110
```

7.6.2.5. RTM Internal Timer Setting (Real time setting)

Function Sets/queries the System operation mode.

Header	Program	Query	Response (No. of characters)
RTM	RTM△m	RTM?	RTM△ m0,m1,m2,m3,m4,m5 (each FIX2)

Parameter m = <NR1>

All setting steps are 1.

m1 : 0 to 99 year(s)
m2 : 1 to 12 month(s)
m3 : 1 to 31 day(s)
m4 : 0 to 23 hour(s)
m5 : 0 to 59 minute(s)
m6 : 0 to 59 second(s)

Application example

Program

・ Sets internal timer 8:20:15" in May 28, 1998.

```
> RTM△98,5,28,8,20,15
```

Query

・ Queries when the timer is set to 8:02:15" in April 23, 1999.

```
> RTM?
< RTM△99,04,23,08,02,15
```

7.6.2.6. BUF Input / Output Buffer Size (Buffer size)

Function Queries the instrument's I/O buffer size in byte.
For both input and output buffers, 16384 bytes should be returned.

Header	Program	Query	Response (No. of characters)
BUF	None	BUF?△m	BUF△m1 (FIX5)

Parameter m = <NR1>

0 : Input Buffer
1 : Output Buffer

Application example

Query

・ Query the input buffer size.

```
> BUF?
< BUF△16384
```

7.6.2.7. HDR Hardware Organization (Hardware organization)

Function Query the instrument's hardware organization.

Header	Program	Query	Response (No. of characters)
HDR	None	HDR?△m0	HDR△m1,m2 (FIX2,FIX18)

Parameter m0 = <NR1>
0 : 3.2G PPG
1 : 3.2G ED
m1 = <NR1>
1 to 31 : Hardware version
0 : No unit exists.
m2 = <binary format>
Flags 1 to a bit corresponding to the existing option.
#B0000000000000000 : Does not mount options
#B0000000000000001 : Mounts option 01.
#B0000000000000010 : Mounts option 02.
#B1000000000000000 : Mounts option 16.

Application example

Query

- Queries the 3.2G PPG organization (version 1 without option).

```
> HDR?△0
< HDR△△1,#B0000000000000000
```

- Queries the 3.2G ED organization (without unit).

```
> HDR?△1
< HDR△△0,#B0000000000000000
```

7.6.2.8. SFR Software Organization (Software organization)

Function Queries the instrument's software organization.

Header	Program	Query	Response (No. of characters)
SFR	None	SFR?	SFR△m (FIX5)

Response m = <NR2>
XX.YY: XX is a major version. YY is a minor version.

Application example

Query

- Query the software version (Ver. 1.3).

```
> SFR?
< SFR△△1.03
```

7.6.2.9. RCL Data Recall (Data recall)

Function Reads the specified file in the specified unit.

Header	Program	Query	Response (No. of characters)
RCL	RCL△m0,m1,m2	None	None

Parameter m0 = <NR1>
0 : Main Frame Setup
1 : Slot1 Setup
2 : PPG Setup
3 : ED Setup
4 : Pattern Data
5 : Slot2 Setup
6 : Eye Diagram Data
m1 = <STRING DATA>
<drv>:\[<dir>]<file>
<drv> = A or C
<dir> = <dir1>\<dir2>\...(Omit for route directory.)
<file> = File name
m2 = <NR1>
Unit in which the file is to be saved.
When m0 is 4, specify 1 or 2. When m0 is not 4, specify 0.
0 : Others
1 : PPG unit
2 : ED unit

Application example

Program

・Read C:\USERS\PTTN\TEST1.PTN (pattern data file) in the PPG unit.

```
> RCL△4,C:\USERS\PTTN\TEST1.PTN,1
```

・ Reads C:\USERS\EYE\TEST1.EYE (Eye diagram result file).

```
> RCL△6,C:\USERS\EYE\TEST1.EYE,0
```

7.6.2.10. SAV Data Save (Data save)

Function Saves the specified data in the specified unit in the file.

Header	Program	Query	Response (No. of characters)
SAV	SAV△m0,m1,m2	None	None

Parameter m0 = <NR1>
0 : Main Frame Setup
1 : Slot1 Setup
2 : PPG Setup
3 : ED Setup
4 : Pattern Data
5 : Slot2 Setup
6 : Eye Diagram Data
m1 = <STRING DATA>
File name
Same as m1 of RCL
When the extension is not specified, it depends on m0.
m2 = <NR1>
Unit in which the data to be saved is contained
When m0 is 4, specify 1 or 2. When m0 is not 4, specify 0.
0 : Others
1 : PPG unit
2 : ED unit

Application example

Program

・Saves the PPG pattern in C:\USERS\PTTN\TEST1.PTN (pattern data file).

```
> SAV△4,C:\USERS\PTTN\TEST1.PTN,1
```

・Saves the result in C:\USERS\EYE\TEST1.EYE (Eye diagram result file).

```
> SAV△6,C:\USERS\EYE\TEST1.EYE,0
```

7.6.2.11. QRC Quick Recall (Quick Data recall)

Function Reads a set of setup files at one time.

Header	Program	Query	Response (No. of characters)
QRC	QRC△m	None	None

Parameter m = <STRING DATA>
<drv> : \[<dir>]<file>
<drv> = A or C
<dir> = <dir1>\<dir2>\...(Omit for route directory.)
<file> = File name (Extension is ENV.)

Application example

Program

・ Read C:\MEAS\TEST.ENV".

```
> QRC△C:\MEAS:TEST.ENV
```

7.6.2.12. QSA Quick Save (Quick Data save)

Function Saves a set of setup files at one time.

Header	Program	Query	Response (No. of characters)
QSA	QSA△m0,m1	None	None

Parameter m0 = <STRING DATA>
File name
Same as m of QRC
m1 = <STRING DATA>
"Comment" (string of 1 to 60 characters)

Application example

Program

・Saves C:＼MEAS＼TESt.ENV under comment "Test for DEVICE1".

```
> QSA△C:＼MEAS:TEST.ENV,”Test for DEVICE1”
```

7.6.2.13. FIL File No./Directory Mode Selection (File No. / directory mode)

Function "File No." saves/reads a file. "DIR" confirms the contents of the file in drive A. However, only DIR mode is effective in actual applications because the analyzer allows files to be saved or read in other mode than File No. mode.

Header	Program	Query	Response (No. of characters)
FIL	FIL△m	FIL?△m	FIL△m1 (see below)

Parameter m = <NR1>

0 : File No.

1 : DIR

m1

********,***,********,**-**-**,**:**(CR/LF)

Returns the value in the order of file name (8 characters), extension (3 characters), file size (8 characters), year/month/day (two characters each), and time:minute (2 characters each).

After returning all the information on the existing file, the parameter returns FIL1 (CR/LF).

Application example

Program

・Sets File No.

```
> FIL△0
```

・Sets DIR (which reads the file organization in drive A in the analyzer.)

```
> FIL△1
```

Query

・ Queries when File No. is set.

```
> FIL?
< FIL△0
```

・ Queries when DIR is set

```
> FIL?
< R001△△△△,PTN,△△△65640,98-05-28,04:34
< R002△△△△,PTN,△△△△6640,98-04-23,04:11
< R004△△△△,EYE,△△△△△640,98-03-03,04:33
< FIL△1
```

7.6.2.14. CAT File Catalog (File catalog)

Function Queries the content in the specified drive and directory.

Header	Program	Query	Response (No. of characters)
CAT	None	CAT?△m1,m2	CAT△m3 (as below)

Parameter m1

Reference drive and directory

・When referencing the content under the directory ＼MP1632C＼OBJ in drive C:
Example:C:＼MP1632C＼OBJ

m2

Specifies extension.

・When specifying file PTN:
Example:PTN

・ When specifying all the files:
Example:ALL

Response m3

**********,********,**-**-**,**:**(CR/LF)

Returns the value in the order of file name (variable length), file size (variable length), year/month/day (2 characters each), and time:minute (2 characters each). However, when a sub directory exists, the parameter returns the sub directory name for the file name and ----- for the file size.

Application example

Query

・ Queries the contents in C:＼MP1632C.

```
> CAT?△C:＼MP1632C,ALL
< OBJ,-----, 98-05-28,04:34
< R001.PTN,65640,98-05-28,04:34
< R002.PTN,6640,98-04-23,04:11
< R004.EYE,640,98-03-03,04:33
```

7.6.2.15. MAC Floppy Access Status (Memory access condition?)

Function Queries the floppy disk access status.

The command always returns the value which indicates "access enabled" because the instrument is designed to return the response only upon completion of access.

Header	Program	Query	Response (No. of characters)
MAC	None	MAC?	MAC△m (FIX1)

Parameter m = <NR1>

0 : Access enabled

Application example

Query

・Queries the floppy access status.

```
> MAC?
< MAC△0
```

7.6.2.16. INI Initialize (Initialize)

Function Initializes the settings to the factory-set values.

Header	Program	Query	Response (No. of characters)
INI	INI	None	None

Parameter None

Application example

Program

・ Initialize the settings to the factory-set values.

```
> INI
```

7.6.2.17. HCP Hard Copy (Hard copy)

Function Creates a hard copy of the window.

Header	Program	Query	Response (No. of characters)
HCP	HCP	None	None

Parameter None

Application example

Program

・ Creates a hard copy.

```
> HCP
```

7.6.2.18. PTX Prints Text (Print text)

Function Prints text strings.

Header	Program	Query	Response (No. of characters)
PTX	PTX△m	None	None

Parameter m = <STRING DATA>

"*string*" : String of 1 to 40 characters

Application example

Program

・ Outputs "SDH Measurement" to a printer.

```
> PTX△"SDH Measurement"
```

7.6.2.19. THR Threshold EI/%EFI Print Selection (Threshold EI / %EFI print ON/OFF)

Function Selects/queries whether Threshold EI/%EFI is printed.
When setting ON in this command, the same setting is output to the measurement result.

Header	Program	Query	Response (No. of characters)
THR	THR	THR?	THR△m (FIX1)

Parameter m = <NR2>
0 : OFF
1 : ON

Application example

Program

・ Sets Threshold EI/%EFI print to ON.

```
> THR△1
```

Query

・ Queries when Threshold EI/%EFI print is OFF.

```
> THR?
< THR△0
```

・ Queries when no unit exists in Slot4.

```
> THR?
< ERR
```

7.6.2.20. EPF Performance Data Print Selection (Error performance print ON/OFF)

Function Selects/queries whether Threshold EI/%EFI is printed.
When setting ON in this command, the same setting is output to the measurement result.

Header	Program	Query	Response (No. of characters)
EPF	EPF	EPF?	EPF△m (FIX1)

Parameter m = <NR1>
0 : OFF(for both Count and %)
1 : ON (for both Count and %)
2 : ON for Count only
3 : ON for % only

Application example

Program

・ Sets performance data (Count and %) print to ON.

```
> EPF△1
```

Query

・ Queries when performance data (Count and %) is OFF.

```
> EPF?
< EPF△0
```

・ Queries when no unit exists in Slot4.

```
> EPF?
< ERR
```

7.6.2.21. ERP Error Measurement Print (Error measurement print ON/OFF)

Function Selects/queries whether the error measurement result is printed.

Header	Program	Query	Response (No. of characters)
ERP	ERP	ERP?	ERP△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・ Sets error measurement result print to ON.

```
> ERP△1
```

Query

・ Queries when error measurement result print is OFF.

```
> ERP?
< ERP△0
```

・ Queries when no unit exists in Slot4.

```
> ERP?
< ERR
```

7.6.2.22. ALP Alarm Interval Print Select (Alarm interval print ON/OFF)

Function Selects/queries whether the alarm interval result is printed.

Header	Program	Query	Response (No. of characters)
ALP	ALP	ALP?	ALP△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・Sets error measurement result print to ON.

```
> ALP△1
```

Query

・ Queries when error measurement result print is OFF.

```
> ALP?
< ERP△0
```

・ Queries when no unit exists in Slot4.

```
> ALP?
< ERR
```

7.6.2.23. FRP Frequency Data Print Select (Frequency data print ON/OFF)

Function Selects/queries whether the frequency measurement result is printed.

Header	Program	Query	Response (No. of characters)
FRP	FRP	FRP?	FRP△m (FIX1)

Parameter m = <NR1>
0 : OFF
1 : ON

Application example

Program

・ Sets frequency measurement result print to ON.

```
> FRP△1
```

Query

・Queries when frequency measurement result print is OFF.

```
> FRP?
< FRP△0
```

・Queries when no unit exists in Slot4.

```
> FRP?
< ERR
```

7.6.2.24. PSA Manual Print Start (Print start)

Function Prints the error alarm measurement result.

Header	Program	Query	Response (No. of characters)
PSA	PSA	None	None

Parameter None

Application example

Program

・ Start error alarm print.

```
> PSA
```

7.6.2.25. PAD Addressed Pattern Print (Addressed pattern print)

Function Prints the addressed pattern.

Header	Program	Query	Response (No. of characters)
PAD	PAD△m0,m1,m2,m3	None	None

Parameter m0 = <NR1>
Unit in which the patter to be printed is contained.
0 : PPG
1 : ED
m1 = <NR1>
Printing format
0 : Hexadecimal data
1 : Biary data
m2 = < hexadecimal format >
Pint start position addressed in hexadecimal
#H0 to #H start address / Step 1
m3 = < hexadecimal format >
Print end position addressed in hexadecimal
#H0 to #H end address / Step 1

Application example

Program

・Prints addresses 0 to ABC in PPG unit in hexadecimal.

```
> PAD△0,0,#H0,#HABC
```

7.6.2.26. PAL All Pattern Print (All pattern print)

Function Prints all patterns.

Header	Program	Query	Response (No. of characters)
PAL	PAL△m0,m1	None	None

Parameter m0 = <NR1>
Unit in which the pattern to be printed is contained
0 : PPG
1 : ED
m1 = <NR1>
Printing format
0 : Hexadecimal
1 : Binary

Application example

Program

・Prints the pattern in PPG unit in hexadecimal.

```
> PAL△0,0
```

7.6.2.27. PEM Eye Margin Measurement Result Print (Print eye margin measurement)

Function Prints the Eye Margin measurement result.

Header	Program	Query	Response (No. of characters)
PEM	PEM	None	None

Parameter None

Application example

Program

・Starts printing the Eye Margin measurement result.

```
> PEM
```

7.6.2.28. PER Recall Eye Diagram Print (Print eye diagram recall)

Function Prints Recall Eye Diagram.

Header	Program	Query	Response (No. of characters)
PER	PER	None	None

Parameter None

Application example

Program

・Print Recall Eye Diagram.

```
>PER
```

7.6.2.29. SWP LSB,MSB Swap Command (Swap/LSB and MSB)

Function Swaps LSB and MSB when specifying the bit pattern of PRGM.

Header	Program	Query	Response (No. of characters)
SWP	SWP△m	SWP?	SWP△m (FIX1)

Parameter m = <NR1>
0 : LSB First
1 : MSB First
Swaps in 16 bits is enabled for BIT, WRT and RED. Swap in 8 bits is enabled for PDT and BDT.

Application example

Program

・LSB First pattern status (example in the BIT command)

```
> SWP 0
> PAG 1
> BIT #HA040,#H1
```

The memory status in the above case:

```
Address      BIT15                BIT0
   0  :      0000   0010   0000   0101
   2  :      1000   0000   0000   0000
```

・ MSB First pattern status (example in the BIT command)

```
> SWP 1
> PAG 1
> BIT #HA040,#H1
```

The memory status in the above case:

```
Address      BIT15                BIT0
   0  :      1010   0000   0100   0000
   2  :      0000   0000   0000   0001
```

・ LSB First pattern status (example in the PDT command)

```
> SWP 0
> PDT #H0,#HF,#HABCD
```

The memory status in the above case:

```
Address      BIT15                BIT0
   0  :      1101   0101   1011   0011
```

・ MSB First pattern status (example in the PDT command)

```
> SWP 1
> PDT #H0,#HF,#HABCD
```

The memory status in the above case:

```
Address      BIT15                BIT0
   0  :      1010   1011   1100   1101
```

Remarks The bit pattern stored in the memory is output starting from BIT15 in Address 0.

7.6.2.30. TRM Selecting termination characters (Termination select)

Function Specifies the termination characters to be added to the response character string outputted from MP1632C.

Header	Program	Query	Response (No. of characters)
TRM	TRM	TRM?	TRM△m (FIX5)

Parameter m = <NR1>
0 : LF only
1 : CR + LF

Application example

Program

・To set CR+LF for the termination characters:

```
> TRM△1
```

Query

・To query only when the termination characters are LF:

```
> TRM?
< TRM△0
```

Note:

1. The termination character setting is maintained after initialization as shown below:
   1) Manual initialization
   2) Initialization using INI, :SYSTem:INITialize or *RST commands
2. The termination character setting is not backed up. It is always set to 0 (LF only) at the activation of the main application software.

CHAPTER8 SAMPLE PROGRAM

8.1. Environment and Overview

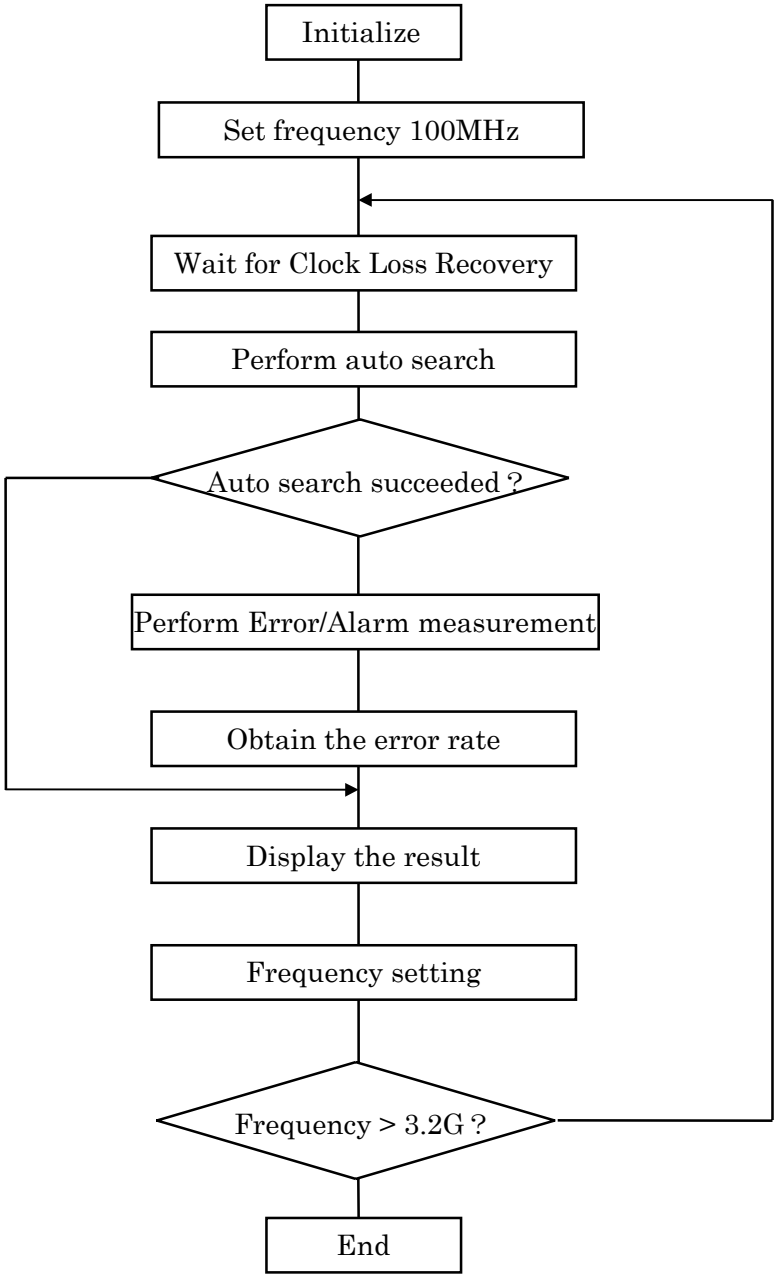
This chapter shows a sample of how to create MP1632C remote control program.

The program which appears in this chapter was written in IBM-PC/AT compatible computer with GPIB interface card of National Instruments (N.I).

This sample program runs on Windows95® operating system, and the development tool is Microsoft® Visual Basic® Version 4.0 Professional Edition.

This sample program measures and reports bit error rate at each frequency from 100 MHz to 3.2 GHz, with controlling MP1632C. It uses RS-232C or GPIB interface.

The flow chart of this sample program is shown below.



8.2. Sample program set up

The sample program and its source files are in a attached “Remote Sample” FD. To refer these files copy them to your PC simply. The FD includes followed files.

COMMON.BAS	NIGLOBAL.BAS	SAMPLE.EXE	MSCOMM32.DLL
FRMFREQ.FRM	GPIB.BAS	RS232C.BAS	SAMPLE.VBP
VBIB-32.BAS	FRMMAIN.FRM	FRMMAIN.FRX	

SAMPLE.EXE is the executive file of this sample program.
SAMPLE.VBP is a project file that Visual Basic® Version 4.0 uses to manage source files. To refer source codes, open the SAMPLE.VBP from Visual Basic® Version 4.0 after starting Visual Basic® Version 4.0.

8.3. Sample program execution

This section describes operation procedure of the sample program.
On starting the sample program, a below dialog box will appear at first.

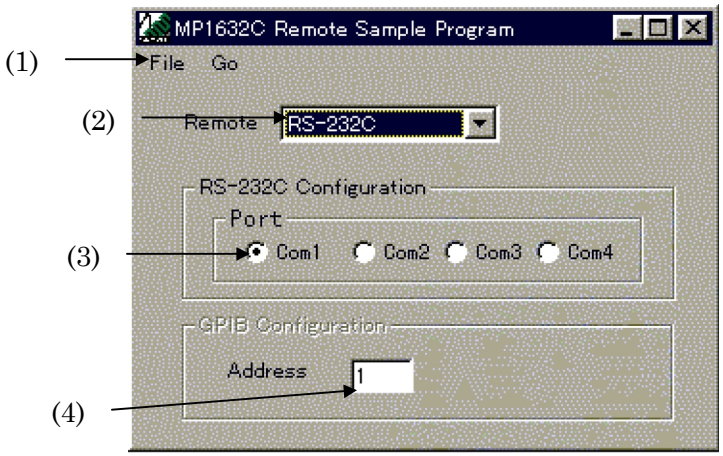


Fig. 8-1 Starting screen

(1) **Menu**
There is a only Exit in File menu. To quit the sample program, select the Exit.
If you select Go menu, Measurement dialog box will appear.(Detail of Measurement dialog box is described bellow)

(2) **Interface selection**
You can select remote control interface from among RS-232C and GPIB.
When you select RS-232C, you must make the interfaces of MP1632C follow condition.

- Speed 9600
- Parity None
- Data Bit 8bits
- Stop bit 1bit
- Flow Control None

(3) RS-232C port selection
You can select which COMM port of your controller is used. This setting is valid when you use RS-232C as remote interface.

(4) GPIB address setting
Input the GPIB address of the MP1632C. The valid range is from 1 to 30. This setting is valid when you use GPIB as remote interface.

If you select Go menu, Measurement dialog box which is shown on next page will appear.

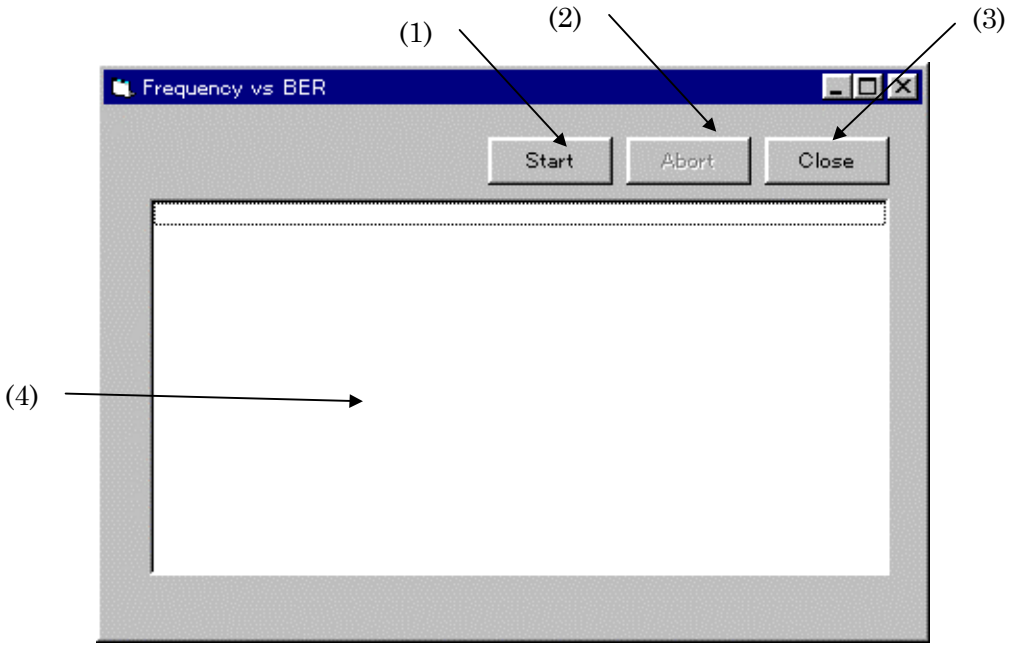


Fig. 8-2 Measurement Dialog box

(1) **Start button**
Clicking this button will start measurement. This button is invalid during measurement.

(2) **Abort button**
Clicking this button during measurement will abort the measurement.

(3) **Close button**
Clicking this button will close the Measurement dialog box and back to the Starting screen. This button is invalid during measurement.

(4) **Result display area**
Measurement result will be reported on here. It reports frequency, error rat, and auto search status message.

8.4. Configuration of program

The sample program consists of two form modules (*.FRM file) and five code modules (*.BAS file). The details of these modules is described bellow.

- **frmMain.frm**
  This is the form to select and set the remote interface. This form also quits the sample program and opens the 'frmFreq' form.
  Following table explains the controls on this form.

Table 8-1 Controls on the 'frmMain'

	Type	Name	Explanation
(1)	menu	mnuExit	Selectiong this menu item quits the sample program
(2)	menu	mnuGo	Selectiong this menu item opens the frmFreq from
(3)	combo box	cmbRemote	This is a control to select the remote interface (RS-232C or GPIB)
(4)	communication	Comm1	This is a control to manage the RS-232C interface
(5)	option button	CommPort(1) to CommPort(4)	This is a control array to select the port of the RS-232C
(6)	text box	txtAddress	This is a control to input the GPIB address

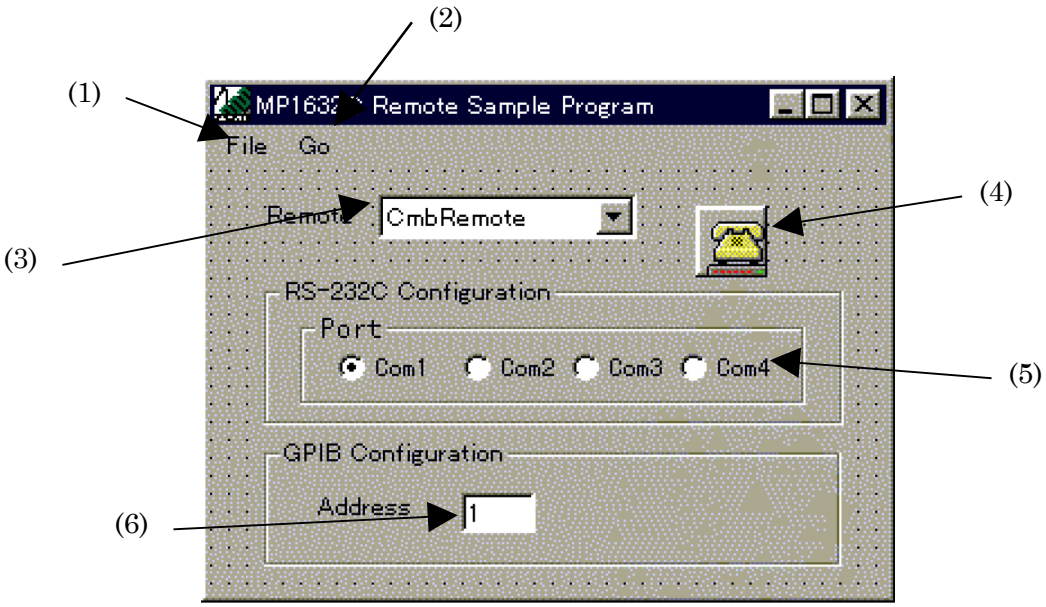


Fig. 8-3 frmMain form

Following table describes the event-handlers which process the events occurring on the 'frmMain' form.

Table 8-2 Event-handlers defined on the 'frmMain'

Name	Explanation
frmMain_Load	This handler is invoked only once immediately after the sample program starts. This sets interface selection RS-232C.
cmbRemote_Click	This handler is invoked when 'cmbRemote' selection is changed. This set interface setting controld valid or invalid according to interface selection.
mnuExit_Click	This handler is invoked when 'Exit' menu item is selected. This quits the sample program.
mnuGo_Click	This handler is invoked when 'Go' menu is selected. This opens the 'frmFreq' form
txtAddress_Chenge	This handler is invoked when the GPIB address is inputted in the 'txtAddress' text box. This checks whether the inputted address is valid.

- **frmFreq.frm**

  This is the form to control the MP1632C and to report the measurement result. This form also quits the sample program and opens the 'frmFreq' form.
  Following table explains the controls on this form.

Table 8-3 Controls on the 'frmFreq'

	Type	Name	Alternated property	Explanation
(1)	command button	mnuExit	None	Clicking this control starts measurement.
(2)	command button	cmdAbort	None	
(3)	command button	cmdClose	None	Clicking this control closes the 'frmFreq' form.
(4)	Timer	Timer1	Enable = False	
(5)	list box	lstFreq	None	This is a control to report the result.

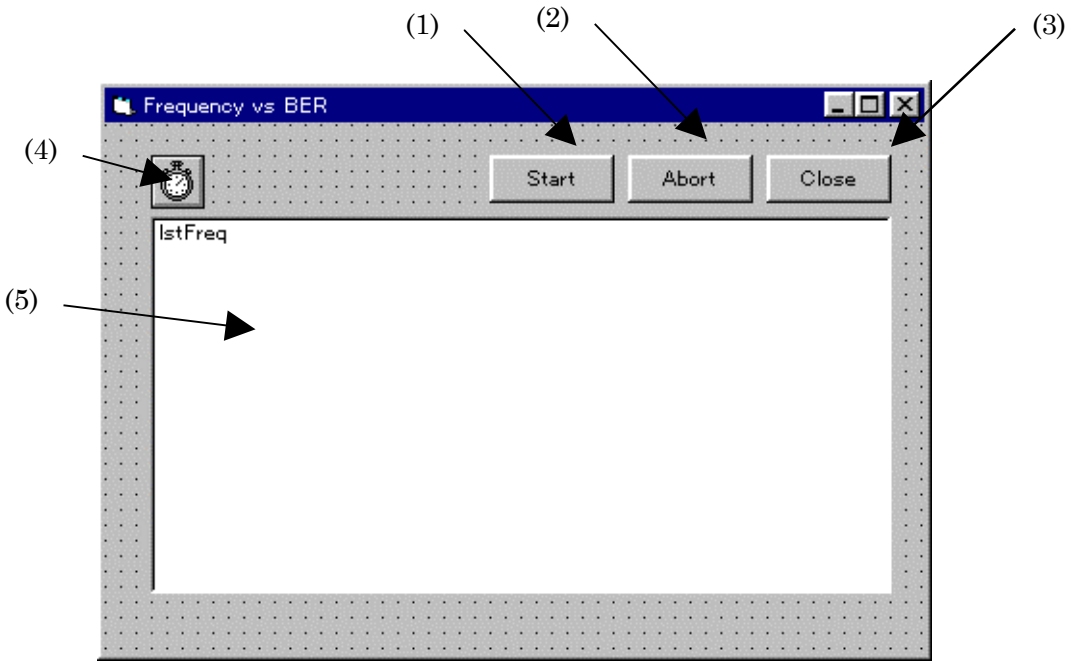


Fig. 8-4 frmFreq form

Following table describes the event-handlers which process the events occurring on the 'frmFreq' form.

Table 8-4 Event-handlers defined on the 'frmFreq'

Name	Explanation
frmFreq_Load	This handler is invoked immediately after 'frmFreq' opens. This opens the remote interface and is invoked only once while 'frmFreq' appears.
FrmFreq_Unload	This handler is invoked immediately before 'frmFreq' is closes. This closes the remote interface and is invoked only once while 'frmFreq' appears.
Timer1.Timer	This handler is invoked every 100ms cycle, but it doesn't when 'Enabled' property is 'False'.
CmdClose_Click	This handler is invoked when Close button is clinked. This closes the 'frmFreq'.
CmdAbort_Click	This handler is invoked when Abort button is clinked.
CmdStart_Click	This handler is invoked when Start button is clicked. This is main procedure of the sample program controlling MP1632C.

- **Common.bas**
  In this module following function and procedure to send / receive the remote commands are defined in this module.

Name	Explanation
ReceiveResponse	This function receives a response from the receive buffer of the specified remote interface, and returns the response.
SendCommand	This procedure sends a command to the send buffer of specified remote interface.
Wait	Wait for demanded time (ms).

- **GPIB.bas**
  In this module following function and procedures to manage the GPIB interface are defined.

Name	Explanation
Open_GPIB	This procedure opens GPIB interface.
Close_GPIB	This procedure closes GPIB interface.
IN_GPIB	This procedure puts string into the send buffer of the GPIB interface.
Out_GPIB	This function gains string from the receive buffer of the GPIB interface, and returns the string.

- **RS232C.bas**
  In this module following function and procedure to manage the RS232C interface are defined.

Name	Explanation
Open_232C	This procedure opens RS-232C interface.
Close_232C	This procedure closes RS-232C interface.
IN_232C	This procedure puts string into the send buffer of the RS-232C interface.
Out_232C	This function gains string from the receive buffer of the RS-232C interface, and returns the string.

- **Niglobal.bas**
  In this module constants to use the driver-level GPIB interface functions are defined. For details refer the User Manual affixed to GPIB interface card of National Instruments.

- **Niglobal.bas**
  This module declares the driver-level GPIB interface functions to call these functions from Visual Basic®. For details refer the User Manual affixed to GPIB interface card of National Instruments.

8.5. Source code

8.5.1. Source code of the frmMain.frm

```
Private Sub CmbRemote_Click()
    'Set the controls deponds on Reomote Type(232C or GPIB)
    If CmbRemote.ListIndex = 0 Then
        fra232C.Enabled = True
        FraGPIB.Enabled = False
    Else
        fra232C.Enabled = False
        FraGPIB.Enabled = True
    End If
End Sub

Private Sub Form_Load()
    CmbRemote.ListIndex = 0
End Sub

Private Sub mnuExit_Click()
    Unload Me
End Sub

Private Sub mnuGo_Click()
    ' Open the Form
    frmFreq.Show 1
End Sub


Private Sub txtAddress_Change()
    Dim addr%
    addr = Val(txtAddress.Text)
    If addr > 31 Then txtAddress.Text = "30"
    If addr < 1 Then txtAddress.Text = "1"
End Sub
```

8.5.2. Source code of the frmFreq.frm

```
Dim AbortFlag As Boolean

Private Sub cmdAbort_Click()
    AbortFlag = True
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdStart_Click()
    Dim Freq%, Report$, resp$, dummy%, cnt%
```

```
' Set Abort Flag false
AbortFlag = False
'Disenable the buttons
cmdStart.Enabled = False
cmdClose.Enabled = False
cmdAbort.Enabled = True
'Initialize MP1632C
SendCommand (":SYST:MEM:INIT")
SendCommand ("*cls")
SendCommand (":SENS4:MEAS:EAL:MODE SING")
SendCommand (":SENS4:MEAS:ASE:MODE PHAS")
'Main Loop (Frequency 50M - 3200M / 10M step)
For Freq = 100 To 3200 Step 100
    ' Setting Frequency
    SendCommand (":OUTP1:CLOC:FREQ " + Str$(Freq) + "000")
    Report$ = Str$(Freq) + "MHz "
    'Wait until Clock Loss recovers
    resp$ = """Occur"""
    cnt = 0
    While resp$ <> ""
        Wait 500
        SendCommand (":CALC4:DATA:MON? ""CLOS""")
        resp$ = ReceiveResponse()
        cnt = cnt + 1
        If cnt > 10 Then
            ' Check PLL Unlock Status
            SendCommand (":CALC1:DATA:MON? ""PLL""")
            resp$ = ReceiveResponse()
            If resp$ <> "" Then
                MsgBox "PLL Unlock occurred.", 16
                Exit For   ' Giveup whole measuring
            Else
                ' Clock input may be lossed
                MsgBox "Clock Loss is detected continuously", 16
                Exit For   ' Giveup whole measuring
            End If
         End If
    Wend  ' end of Clock Loss polling

    'Auto Search
    SendCommand (":SENS4:MEAS:ASE:STAR")
    resp$ = "1"
    While resp$ = "1"
        ' Wait for ending Auto Search
        SendCommand (":SENS4:MEAS:ASE:STAT?")
        resp$ = ReceiveResponse()
        Wait 500
    Wend
```

```
        If resp$ = "0" Then    'Auto Search Succeeded
            'Error/Alarm Measurement Start
            SendCommand (":SENS4:MEAS:STAR")
            resp$ = "1"
            'Wait for ending Error/Alarm Measurement
            While resp$ = "1"
                SendCommand (":SENS4:MEAS:EAL:STAT?")
                resp$ = ReceiveResponse()
                dummy = DoEvents()
            Wend
            'Get Error Rate
            SendCommand (":CALC4:DATA:EAL? ""ER:TOT""")
            Report$ = Report$ + ReceiveResponse()

        Else                    'Auto Search Failed
            Report$ = Report$ + "Auto Search N.G."
        End If
        dummy = DoEvents()
        lstFreq.AddItem Report$
        'checking the Abort button is pressed or not
        If AbortFlag Then Exit For

    Next Freq 'End of Main Loop

    'Enable the buttons
    cmdStart.Enabled = True
    cmdClose.Enabled = True
    cmdAbort.Enabled = False

End Sub
```

```
Private Sub Form_Load()
    Dim i%
    If frmMain.CmbRemote.ListIndex = 0 Then
        'Open RS-232C Port
        For i = 1 To 4
            If frmMain.ComPort(i).value = True Then
                frmMain.Comm1.CommPort = i
            End If
        Next i
        Open_232C
    Else
        'Open GPIB
        Open_GPIB
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If frmMain.CmbRemote.ListIndex = 0 Then
        'Close RS-232C Port
        Close_232C
    Else
        'Close GPIB
        Close_GPIB
    End If
End Sub

Private Sub Timer1_Timer()
    Timer1.Enabled = False
End Sub
```

8.5.3. Source code of the Common.bas

```
Function ReceiveResponse$()

    If frmMain.CmbRemote.ListIndex = 0 Then
        'RS-232C
        ReceiveResponse$ = IN_232C()
    Else
        'GPIB
        ReceiveResponse$ = IN_GPIB()
    End If

End Function

Sub SendCommand(c$)

    If frmMain.CmbRemote.ListIndex = 0 Then
        'RS-232C
        Out_232C (c$)
    Else
        'GPIB
        Out_GPIB (c$)
    End If

End Sub

Sub Wait(t%)

    Dim dummy%

    frmFreq.Timer1.Interval = t
    frmFreq.Timer1.Enabled = True

    While frmFreq.Timer1.Enabled
        dummy = DoEvents()
    Wend

End Sub
```

8.5.4. Source code of the GPIB.bas

```
Dim ud As Integer 'descriptor for a GPIB board

Sub Close_GPIB()
    Dim addrlist(30) As Integer
    addrlist(0) = NOADDR
    Call ibonl(ud, 0)
    Call EnableLocal(0, addrlist())
End Sub

Function IN_GPIB() As String
    Dim resp$
    resp$ = Space$(255)
    Call Receive(0, Val(frmMain.txtAddress.Text), resp$, 2)
    IN_GPIB$ = Left$(resp$, InStr(resp$, Chr$(10)) - 1)
End Function

Sub Open_GPIB()
    Dim addrlist(30) As Integer
    addrlist(0) = NOADDR
    Call ibfind("gpib0", ud)
    Call ibrsc(ud, 1)
    Call ibtmo(ud, T10s)
    Call ibeos(ud, 0)
    Call ibeos(ud, &HC0A)
    Call ibeot(ud, 1)
    Call ibpad(ud, 0)
    Call ibsad(ud, 0)
    Call SendIFC(0)
    Call EnableRemote(0, addrlist())
End Sub

Sub Out_GPIB(outstr$)
    Call Send(0, Val(frmMain.txtAddress.Text), outstr$, DABend)
End Sub
```

8.5.5. Source code of the RS232C.bas

```
Sub Close_232C()
    If frmMain.Comm1.PortOpen = False Then Exit Sub
    frmMain.Comm1.PortOpen = False
End Sub

Function IN_232C() As String
    Dim receive_str$
    Dim dummy%
    Dim timeout&
    Dim i%
    'Initilize Values
    receive_str$ = ""
    timeout& = 0
    i% = 0
    'Wait for Response from RS-232C
    While frmMain.Comm1.InBufferCount = 0
        dummy = DoEvents()
        If Err Then Error Err
        If timeout& = 100000 Then
            IN_232C = receive_str$
            Exit Function
        End If
        timeout& = timeout& + 1
    Wend
    'Input Response Input Buffer
    While i% = 0
        While frmMain.Comm1.InBufferCount <> 0
            receive_str$ = receive_str$ + frmMain.Comm1.Input
            dummy = DoEvents()
            If Err Then Error Err
        Wend
        i% = InStr(receive_str$, Chr$(10))
    Wend
    IN_232C = Left$(receive_str$, i - 1)
End Function

Sub Open_232C()
    If frmMain.Comm1.PortOpen = True Then Exit Sub
    frmMain.Comm1.PortOpen = True
End Sub

Sub Out_232C(outstr$)
    'Clear the input buffer
    frmMain.Comm1.InBufferCount = 0
    'Put on the output buffer
    frmMain.Comm1.Output = outstr$ & Chr$(10)
End Sub
```

8-18.

APPENDIX

Appendix A SCPI error message

SCPI defines error-specific codes and messages as responses to the SCPI command :SYSTem:ERRor?.
A detailed definition of the error messages supported by this instrument follows.

■ Command errors

The error codes -199 and -100 indicate that an IEEE488.2 syntax error occurred. Bit 5 of the device's event status register is set at the same time.
An error occurs when any one of the following events arises:

- The device received a message in violation of the specifications of IEEE488.2.
- A device-specific command or a header in violation of the specifications of common commands was received.
- GET (Group Execute Trigger) was transmitted within a program message.

Table A-1 Error Codes and Error Messages in Command Errors

Code	Message	Error detection condition
-101	Invalid character	An invalid character is found in a header or parameter. Examples: # is included in a header. :SOURce3:PATTern:M#DE NORMal $ is included in a parameter. :SENSe3:PATTern:MODE N$RMAL
-104	Data type error	The parameter type does not match its specification. Example: Character data was transmitted to a command having a parameter of the type numeric.
-105	GET not allowed	GET was transmitted within a program message.
-108	Parameter not allowed	Too many parameters were specified. Example: Four parameters were transmitted when only two were required.
-112	Program mnemonic too long	A program mnemonic of 12 characters or longer was entered. Example: A header colon is missing. :SOURce3:PATTern:ZSUBstituteLENGth 1,7
-113	Undefined header	The header was valid in its syntax but was not defined by the device. Example: A header was written incorrectly. :SOURce3:PATTern:MDE NORMal
-120	Numeric data error	There is an error in numeric data.
-121	Invalid character in number	An invalid character is found in numeric data. Example: A symbol is included in the numeric data. :SOURce3:PATTern:TYPE #,PRBS7
-130	Suffix error	There is an error in a suffix. Example: Part of the parameter of the type numeric was romanized by mistake. :OUTPut3:CLOCk:FREQuency 100I
-144	Character data too long	Character data of 12 characters or longer was entered.
-150	String data error	Undefined string data. Example: A single quotation mark and a double quotation mark were intermixed. :CALCulate3:DATA:EALarm “ER:INS’

The error codes -299 and -200 indicate that an error occurred in the executive control of the device. Bit 4 of the device's event status register is set at the same time.

An error occurs when any one of the following events arises:

- The <PROGRAM DATA> that follows a header is undefined by the device.
- A program message is not executable due to particular device status.

Table A-2 Error Codes and Error Messages in Execution Errors

Code	Message	Error detection condition
-220	Parameter error	There is an error in the parameter.
-221	Setting conflict	The parameter is valid but cannot be set due to particular device status. Example: An attempt was made to set a Burst pattern when the pattern mode was Repeat.
-222	Data out of range	Numeric data is out of range for the device
-223	Too much data	String data exceeds the defined length.
-224	Illegal parameter value	The received parameter is illegal. Examples: Character data is illegal. String data is illegal.
-232	Invalid format	Specified file is invalid format. This error occurs when you open PPG file as Pattern file.
-233	Invalid version	Specified file is saved by application software whose version is newer.
-241	Hardware missing	The command cannot be executed because the option is not installed.
-252	Missing media	The floppy disk is not inserted.
-253	Corrupt media	The floppy disk is not formatted.
-254	Media full	The hard disk or floppy disk does not have enough space available.
-255	Directory full	The hard disk or floppy disk directory does not have enough space available.
-256	File name not found	The file is not found on the hard disk or floppy disk.
-257	File name error	The filename is invalid.
-258	Media protected	Data cannot be written to the floppy disk because it is write-protected.

■ Device-specific errors

The error codes -399 and -300 indicate that the device has detected an error other than command, query, and execution errors. Such errors might convey hardware or firmware failures or self-test errors.

When device-specific errors occur, bit 3 of the device's event status register is set.

Table A-3 Error Codes and Error Messages in Device-Specific Errors

Code	Message	Error detection condition
-310	System error	A system error occurred.
-315	Configuration memory lost	Resume memory was lost.

■ Query errors

The error codes -499 and -400 indicate that a message switching control protocol error occurred under the device's output queue control. When query errors occur, bit 2 of the device's event status register is set.

An error occurs when any one of the following events arises:

- An attempt was made to read from the output queue when there was no output.
- Data on the output queue was lost.

Table A-4 Error Codes and Error Messages in Query Errors

Code	Message	Error detection condition
-410	Query INTERRUPTED	An interrupt due to a new command occurred before the device completed the transmission of a response massage. Example: *TRG was transmitted before the complete transmission of a response.
-420	Query UNTERMINATED	No query had been transmitted corresponding to the response message to be read, or the query is not terminated completely.
-430	Query DEADLOCKED	Buffering of more data than was permitted by the storage area space available was attempted. Example: The device cannot proceed with measurement any farther because both the input and output buffers are full.

Appendix B Command list (SCPI)

This appendix lists the SCPI commands supported by this instrument, grouped by window. Query commands are not stated (they are stated, however, when a corresponding program command does not exist).
See Chapter 6 for more command details, including parameter types and contents.

■ Parent window

Table B-1 Parent window command list

Setup item		Command	Page
File menu	Quick Open	:SYSTem:MMEMory:QRECall	6-96
	Quick Save	:SYSTem:MMEMory:QSTore	6-96
	Open	:SYSTem:MMEMory:RECall	6-94
		:SYSTem:MMEMory:RECall"EYE:TEMPlate1"	6-94
	Save	:SYSTem:MMEMory:STORe	6-95
		:SYSTem:MMEMory:STORe"EYE:TEMPlate1"	6-95
	Hard Copy	:SYSTem:PRINt:COPY	6-98
	Print...	:SYSTem:PRINt:EALarm	6-98
		:SYSTem:PRINt:PATTern:TABLe:ADDRess	6-98
		:SYSTem:PRINt:PATTern:TABLe:ALL	6-99
	Initialize	:SYSTem:MEMory:INITialize	6-98
Tool bar		:OUTPut:SET	6-32
Subwindow manipulations	Open	:DISPlay:WINDow:OPEN	6-75
	Close	:DISPlay:WINDow:CLOSe	
Information Message		:CALCulate:DATA:MONitor?	6-74

■ System sub window

Table B-2 System subwindow command list

Setup item	Command	Page
Panel switching	:DISPlay:SYSTem[:NAME]	6-75

● System:System panel

Table B-3 System:System panel command list

Setup item		Command	Page
System Mode		:SYSTem:MODE?	6-92
Buzzer (System Error)	ON/OFF status setting	:SYSTem:BEEPer:SYSTem:SET	6-91
	Item setting	:SYSTem:BEEPer:SYSTem:TYPE	6-92
Buzzer (Measruement)	Error	:SYSTem:BEEPer:ERRor:SET	6-91
	Alarm	:SYSTem:BEEPer:ALARm:SET	
Date & Time	Date	:SYSTem:DATE	6-92
	Time	:SYSTem:TIME	

- **System:Option Panel**

Table B-4 System:Option panel command list

Setup item	Command	Page
Hardware	:SYSTem:ORGanization:HARDware?	6-93
Software	:SYSTem:ORGanization:SOFTware?	6-94

- **Setup subwindow**

Table B-5 Setup subwindow command list

Setup item	Command	Page
Panel switching	:DISPlay:SETup[:NAME]	6-76

- **Setup:Setup panel**

Table B-6 Setup:Setup panel command list (3.2G PPG)

Setup item		Command	Page
Common Setting	Pattern	:INSTrument:COUPle:PATTern:SET	6-40

Table B-7 Setup:Setup panel command list (3.2G ED)

Setup item		Command	Page
Common Setting	Pattern	:INSTrument:COUPle:PATTern:SET	6-72

- **Setup:Frequency panel**

Table B-8 Setup:Frequency panel command list

Setup item		Command	Page
Clock	Reference Frequency	:OUTPut:RCLock:SELect :OUTPut:CLOCk:FREQuency	6-73
Alarm monitor		:CALCulate:DATA:MONitor?	6-74

- **Setup:Clock I/F panel**

Table B-9 Setup:Clock I/F panel command list (3.2G PPG)

Setup item		Command	Page
Tracking		:INSTrument:COUPle:CLOCk:TRACking	6-40
Output		:OUTPut:CLOCk:OUTPut	6-32
Duty		:OUTPut:CLOCk:DUTY	6-34
Delay		:OUTPut:CLOCk:DELay	6-32
Level		:OUTPut:CLOCk:LEVel	6-33
Amplitude		:OUTPut:CLOCk:AMPLitude	6-33
Offset	offset value	:OUTPut:CLOCk:OFFSet	6-33
	offset reference value	:OUTPut:CLOCk:OREFerence	6-34

Table B-10 Setup:Clock I/F panel command list (3.2G ED)

Setup item	Command	Page
Polarity	:INPut:CLOCk:POLarity	6-59
Delay	:INPut:CLOCk:DELay	
Termination	:INPut:CLOCk:TERMination	

- **Setup:Data I/F panel**

Table B-11 Setup:Data I/F panel command list (3.2G PPG)

Setup item		Command	Page
Tracking		:INSTrument:COUPle:DATA:TRACking	6-40
Output		:OUTPut:DATA:OUTPut	6-35
Cross Point		:OUTPut:DATA:CPOint	6-35
Level		:OUTPut:DATA:LEVel	6-36
Amplitude		:OUTPut:DATA:AMPLitude	6-36
Offset	offset value	:OUTPut:DATA:OFFSet	6-36
	offset reference value	:OUTPut:DATA:OREFerence	6-37

Table B-12 Setup:Data I/F panel command list (3.2G ED)

Setup item	Command	Page
Level	:INPut:DATA:LEVel	6-60
Threshold	:INPut:DATA:THReshold	
Termination	:INPut:DATA:TEMination	

- **Setup:Pattern panel**

Table B-13 Setup:Pattern panel command list (3.2G PPG)

Setup item		Command	Page
Output Pattern		:SOURce:PATTern:OMODe	6-24
Pattern Setting	Pattern	:SOURce:PATTern:TYPE	6-24
Pattern Setting (PRBS)	Mark Ratio Bit Shift	:SOURce:PATTern:PRBS:MRATio :SOURce:PATTern:PRBS:BSHift	6-25
Pattern Setting (Zero-Subst)	Pattern Length Zero Length Logic	:SOURce:PATTern:ZSUBstitute:LENGth :SOURce:PATTern:ZSUBstitute:ZLENgth :SOURce:PATTern:ZSUBstitute:LOGic	6-25 6-26 6-26
Pattern Setting (PRGM)	Pattern Length Logic	:SOURce:PATTern:PROGram:LENGth :SOURce:PATTern:PROGram:LOGic	6-26 6-27
Burst Setting	Mode Burst Cycle Enable Length	:SOURce:PATTern:BURSt:MODE :SOURce:PATTern:BURSt:CYCLe :SOURce:PATTern:BURSt:ELENgth	6-27
PRGM Pattern	Character string setting Binary data setting	:SOURce:PATTern:DATA:WHOLe :SOURce:PATTern:BDATa:WHOLe	6-28 6-29
PRGM Pattern Data reverse	Address specification Delta specification	:SOURce:PATTern:DREVerse:ADDRess :SOURce:PATTern:DREVerse:DELTa	6-29

Table B-14 Setup:Pattern panel command list (3.2G ED)

Setup item		Command	Page
Pattern Input		:SENSe:PATTern:IMODe	6-41
Pattern Setting	Pattern	:SENSe:PATTern:TYPE	6-41
Pattern Setting (PRBS)	Mark Ratio Bit Shift	:SENSe:PATTern:PRBS:MRATio :SENSe:PATTern:PRBS:BSHift	6-42
Pattern Setting (Zero-Subst)	Pattern Length Zero Length Logic	:SENSe:PATTern:ZSUBstitute:LENGth :SENSe:PATTern:ZSUBstitute:ZLENgth :SENSe:PATTern:ZSUBstitute:LOGic	6-42 6-43 6-43
Pattern Setting (PRGM)	Pattern Length Logic	:SENSe:PATTern:PROGram:LENGth :SENSe:PATTern:PROGram:LOGic	6-43 6-44
Sync Setting	Auto Sync Sync Threshold Gain Sync Threshold Loss Internal Threshold Pattern Sync Mode Frame Length	:SENSe:PATTern:SYNC:ASYNc :SENSe:PATTern:SYNC:THReshold:GAIN :SENSe:PATTern:SYNC:THReshold:LOSS :SENSe:PATTern:SYNC:THReshold:INTernal :SENSe:PATTern:SYNC:PSMode :SENSe:PATTern:SYNC:FLENgth	6-44 6-45 6-44 6-45 6-46 6-46
PRGM Pattern	Character string setting Binary data setting	:SENSe:PATTern:DATA:WHOLe :SENSe:PATTern:BDATa:WHOLe	6-47 6-48
PRGM Pattern Data reverse	Address specification Delta specification	:SENSe:PATTern:DREVerse:ADDRess :SENSe:PATTern:DREVerse:DELTa	6-48

- **Setup:Trigger I/F panel**

Table B-15 Setup:Trigger I/F panel command list (3.2G PPG)

Setup item	Command	Page
Source	:OUTPut:SYNC:SOURce	6-37
Position	:OUTPut:SYNC:POSition	6-38

Table B-16 Setup:Trigger I/F panel command list (3.2G ED)

Setup item	Command	Page
Source	:OUTPut:SYNC:SOURce	6-71

- **Setup:Utility panel**

Table B-17 Setup:Utility panel command list (3.2G PPG)

Setup item	Command	Page
PRBS Logic	:SOURce:PATTern:LOGic:PRBS	6-31
PRGM Logic	:SOURce:PATTern:LOGic:PRGM	6-31
DC Impedance	:OUTPut:CLOCk:IMPedance	6-34
	:OUTPut:DATA:IMPedance	6-35

Table B-18 Setup:Utility panel command list (3.2G ED)

Setup item	Command	Page
PRBS Logic	:SENSe:PATTern:LOGic:PRBS	6-49
PRGM Logic	:SENSe:PATTern:LOGic:PRGM	6-49
MEAS. Restart	:SENSe:MEASure:MREStart	6-50

- **Test Menu sub window**

Table B-19 Test Menu sub window command list

Setting item	Command	Page
Panel switching	:DISPlay:TEST[:NAME]	6-76

- **Test Menu:Measurement panel**

Table B-20 Test Menu:Measurement panel command list

Setting item	Command	Page
Test Item	:SENSe:MEASure:TEST	6-49

Table B-21 Test Menu:Measurement panel command list (Test Item:Error/Alarm)

Setting item		Command	Page
MEAS.Mode		:SENSe:MEASure:EALarm:MODE	6-50
MEAS.Time		:SENSe:MEASure:EALarm:PERiod	6-50
Timed Start	Timed Start Start Time	:SENSe:MEASure:EALarm:BTIMe:SET :SENSe:MEASure:EALarm:BTIMe:STARt	6-51
Error Type		:SENSe:MEASure:EALarm:ERRor:TYPE	6-51
Route Mask		:SENSe:MEASure:EALarm:MASK:ROUTe	6-52
EI/%EFI	Interval	:CALCulate:EALarm:ERRor:INTerval	6-61
Performance	Measurement Threshold	:CALCulate:EALarm:PERFomance:MEASurement :CALCulate:EALarm:PERFomance:THReshold	6-62
Threshold EI/%EFI	Measurement	:CALCulate:EALarm:TEINterval:MEASurement	6-63
Power Fail	Measurement	:CALCulate:EALarm:PFAil	6-63
Clock Loss Evaluation		:CALCulate:EALarm:CLEValuation	6-61
Sync Loss Evaluation		:CALCulate:EALarm:SLEValuation	6-61

Table B-22 Test Menu:Measurement panel command list (Test Item:Eye Margin)

Setting item	Command	Page
MEAS.Mode	:SENSe:MEASure:EMARgin:MODE :SENSe:MEASure:EMARgin:TYPE	6-54
Error Threshold	:SENSe:MEASure:EMARgin:MARGin:THReshold :SENSe:MEASure:EMARgin:MARGin:RESolution	6-55
Threshold Select	:SENSe:MEASure:EMARgin:DIAGram:THReshold	6-56

- **Test Menu:Error Addition panel**

Table B-23 Test Menu:Error Addition panel command list

Setting item	Command	Page
Error Addition	:SOURce:PATTern:EADDition:SET	6-30
Single	:SOURce:PATTern:EADDition:SINGle	6-30
Rate	:SOURce:PATTern:EADDition:RATE	6-30
Addition Route	:SOURce:PATTern:EADDition:ROUTe	6-30

■ **Result subwindow (Common to selection from Test Item drop-down list box)**

Table B-24 Result subwindow command list (3.2G ED)

Setting item		Command	Page
Auto Search	Mode	:SENSe:MEASure:ASEarch:MODE	6-58
	PRBS Pattern Search	:SENSe:MEASure:ASEarch:PATTern	
	Start	:SENSe:MEASure:ASEarch:STARt	
	Stop	:SENSe:MEASure:ASEarch:STOP	
	Status report	:SENSe:MEASure:ASEarch:STATe?	
Alarm monitor		:CALCulate:DATA:MONitor?	6-70
Start		:SENSe:MEASure:STARt	6-49
Stop		:SENSe:MEASure:STOP	6-49
Time		:DISPlay:RESult:TIME	6-76
Store measurement data to the buffer		:CALCulate:DATA:STORe	6-69
Clear the measurement data buffer		:CALCulate:DATA:CLEar	6-69

■ **Result subwindow (Test Item:Error/Alarm)**

Table B-25 Result subwindow (Error/Alarm) command list

Setting item	Command	Page
Measurement start time query	:SENSe:MEASure:EALarm:STARt?	6-52
Measurement end time query	:SENSe:MEASure:EALarm:STOP?	6-52
Measurement status query	:SENSe:MEASure:EALarm:STATe?	6-52
Alarm occurrence time query	:SENSe:MEASure:EALarm:AOCCur?	6-53
Alarm recovery time query	:SENSe:MEASure:EALarm:ARECver?	6-53
Elapsed time query	:SENSe:MEASure:EALarm:ELAPsed?	6-53
Left time query	:SENSe:MEASure:EALarm:TIMed?	6-53
Query when the intermediate data was created	:SENSe:MEASure:EALarm:ITIMe?	6-53
Measurement result query	:CALCulate:DATA:EALarm?	6-64
Panel Switching	:DISPlay:RESult:EALarm[:NAME]	6-76
Display	:DISPlay:RESult:EALarm:MODE	6-77

● **Result:ALL panel**

Table B-26 Result:ALL panel command list

Setting item	Command	Page
Performance	:DISPlay:RESult:EALarm:ALL:PTYPe	6-77

- **Reusult:Zoom panel**

Table B-27 Result:Zoom panel command list

Setting item		Command	Page
Display1	Display	:DISPlay:RESult:EALarm:ZOOM1:SET	
	Item	:DISPlay:RESult:EALarm:ZOOM1:ITEM	
	Performance Item	:DISPlay:RESult:EALarm:ZOOM1:PITem	
	Threshold EI/%EI Item	:DISPlay:RESult:EALarm:ZOOM1:TITem	
Display2	Display	:DISPlay:RESult:EALarm:ZOOM2:SET	6-77
	Item	:DISPlay:RESult:EALarm:ZOOM2:ITEM	
	Performance Item	:DISPlay:RESult:EALarm:ZOOM2:PITem	|
	Threshold EI/%EI Item	:DISPlay:RESult:EALarm:ZOOM2:TITem	
Display3	Display	:DISPlay:RESult:EALarm:ZOOM3:SET	6-80
	Item	:DISPlay:RESult:EALarm:ZOOM3:ITEM	
	Performance Item	:DISPlay:RESult:EALarm:ZOOM3:PITem	
	Threshold EI/%EI Item	:DISPlay:RESult:EALarm:ZOOM3:TITem	
Display4	Display	:DISPlay:RESult:EALarm:ZOOM4:SET	
	Item	:DISPlay:RESult:EALarm:ZOOM4:ITEM	
	Performance Item	:DISPlay:RESult:EALarm:ZOOM4:PITem	
	Threshold EI/%EI Item	:DISPlay:RESult:EALarm:ZOOM4:TITem	

- **Result:Monitor panel**

Table B-28 Result:Monitor panel command list

Setting item	Command	Page
History Reset	:DISPlay:RESult:EALarm:HRESet	6-80

■ **Result subwindow (Test Item:Eye Margin)**

Table B-29 Result subwindow (Eye Margin) command list

Setting item		Command	Page
Measurement start time query		:SENSe:MEASure:EMARgin:STARt?	6-57
Measurement end time query		:SENSe:MEASure:EMARgin:STOP?	6-57
Measurement status query		:SENSe:MEASure:EMARgin:STATe?	6-57
Alarm occurrence time query		:SENSe:MEASure:EMARgin:ELAPsed?	6-57
Query how many points is valid (Diagram		:SENSe:MEASure:EMARgin:DIAGram:POINt?	6-56
Measurement result query		:CALCulate:DATA:EMARgin?	6-64
Display		:DISPlay:RESult:EMARgin:ERATe	6-81
Scale	Auto Scale	:DISPlay:RESult:EMARgin:SCALe:ASCale	6-81
	Threshold Voltage	:DISPlay:RESult:EMARgin:SCALe:VOLTage	6-82
	Phase	:DISPlay:RESult:EMARgin:SCALe:PHASe	6-82
Marker,△Marker	Display ON/OFF	:DISPlay:RESult:EMARgin:MARKer:SET	6-82
	Movement mode	:DISPlay:RESult:EMARgin:MARKer:MODE	6-83
	Marker switching	:DISPlay:RESult:EMARgin:MARKer:ERATe	6-83
	Move	:DISPlay:RESult:EMARgin:MARKer:MOVE	6-84
	Position query	:DISPlay:RESult:EMARgin:MARKer:POSition?	6-84

Table B-30 Result subwindow (Eye Diagram template) command list

Setting item	Command	Page
Template selection	:DISPlay:RESult:EMARgin:TEMPlate:SELect	6-84
Template query	:DISPlay:RESult:EMARgin:TEMPlate:SELect?	6-84
Template movement	:DISPlay:RESult:EMARgin:TEMPlate:MOVE	6-84
Setting the number of points for the template	:DISPlay:RESult:EMARgin:TEMPlate:PNUMber	6-85
Querying the number of points for the template	:DISPlay:RESult:EMARgin:TEMPlate:PNUMber?	6-85
Setting the offset voltage value for the template	:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:VOLTage	6-85
Querying the offset voltage value for the template	:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:VOLTage?	6-85
Setting the offset phase value for the template	:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:PHASe	6-86
Querying the offset phase value for the template	:DISPlay:RESult:EMARgin:TEMPlate:OFFSet:PHASe?	6-86
Setting the offset voltage value at the specified point for the template	:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage	6-86
Querying the voltage value at the specified point for the template	:DISPlay:RESult:EMARgin:TEMPlate:POINt:VOLTage?	6-86
Setting the phase value at the specified point for the template	:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe	6-86
Querying the phase value at the specified point for the template	:DISPlay:RESult:EMARgin:TEMPlate:POINt:PHASe?	6-87
Selecting the line connection method for the template	:DISPlay:RESult:EMARgin:TEMPlate:CONNect	6-87
Querying the line connection method for the template	:DISPlay:RESult:EMARgin:TEMPlate:CONNect?	6-87

■ **Customizesub window**

Table B-31 Customizesub window command list

Setting item		Command	Page
Setup1	Unit	:DISPlay:CUSTOmize:SETup1:UNIT	
	Port	:DISPlay:CUSTOmize:SETup1:PORT	
	Item	:DISPlay:CUSTOmize:SETup1:ITEM	
Setup2	Unit	:DISPlay:CUSTOmize:SETup2:UNIT	
	Port	:DISPlay:CUSTOmize:SETup2:PORT	
	Item	:DISPlay:CUSTOmize:SETup2:ITEM	
Setup3	Unit	:DISPlay:CUSTOmize:SETup3:UNIT	6-87
	Port	:DISPlay:CUSTOmize:SETup3:PORT	
	Item	:DISPlay:CUSTOmize:SETup3:ITEM	|
Setup4	Unit	:DISPlay:CUSTOmize:SETup4:UNIT	
	Port	:DISPlay:CUSTOmize:SETup4:PORT	
	Item	:DISPlay:CUSTOmize:SETup4:ITEM	6-89
Setup5	Unit	:DISPlay:CUSTOmize:SETup5:UNIT	
	Port	:DISPlay:CUSTOmize:SETup5:PORT	
	Item	:DISPlay:CUSTOmize:SETup5:ITEM	
Setup6	Unit	:DISPlay:CUSTOmize:SETup6:UNIT	
	Port	:DISPlay:CUSTOmize:SETup6:PORT	
	Item	:DISPlay:CUSTOmize:SETup6:ITEM	
Pattern	Unit	:DISPlay:CUSTOmize:PATTern:UNIT	6-90
	Offset	:DISPlay:CUSTOmize:PATTern:OFFSet	
Result	Item	:DISPlay:CUSTOmize:RESult:ITEM	6-90

■ **Other commands**

Table B-32 Other command list

Setting item	Command	Page
I/O buffer size of devices query	:SYSTem:BSIZe?	6-93
Error status query	:SYSTem:ERRor?	6-93
SCPI Version query	:SYSTem:VERSion?	6-93
Status register setting/Query	:STATus subsystem	6-100
Disk information query	:SYSTem:MMEMory:CATalog?	6-97
String print out	:SYSTem:PRINt:TEXT	6-98
Printing the template data	:SYSTem:PRINt:EMARgin	6-99
Printing the template data (on recall)	:SYSTem:PRINt:REMargin	6-99
Termination setting	:SYSTem:TERMination	6-99

Appendix C Command list (Native)

This appendix lists the Native commands supported by this instrument, grouped by window. Query commands are not stated (they are stated, however, when a corresponding program command does not exist).
See Chapter 7 for more command details, including parameter types and contents.

■ Parent window

Table C-1 Parent window command list

Setting Item		Command	Page
File menu	Quick Open	QRC	7-140
	Quick Save	QSA	7-141
	Open	RCL	7-139
	Save	SAV	7-140
	Hard Copy	HCP	7-144
	Print...	FMT	7-105
		THR,EPF,ERP,ALPFRP,PSA,PAD, PAL,PEM	7-145 to 7-149
		PER	7-149
	Initialize	INI	7-144
Tool bar		OON	7-33
Subwindow manipulations	Open Close	WOC	7-116
Information Message		DTR,	7-48 , 7-113
		PLL,	7-52
		PCL	7-48
file operation commands		FIL,	7-142
		CAT	7-143

■ System sub window

Table C-2 System sub window Command list

Setting Item	Command	Page
Panel swutching	SYW	7-117

● System:System panel

Table C-3 System:System Panel Command list

Setting Item		Command	Page
Buzzer	ON/OFF	SYS	7-135
(System Error)	Item selection	SYT	7-136
Buzzer	Error	MON	7-135
(Measruement)	Alarm	ALM	7-136
Date & Time		RTM	7-137

- **System:Option panel**

Table C-4　System:Option Panel Command list

Setting Item	Command	Page
Hardware	HDR	7-138
Software	SFR	7-138

- **Setup sub window**

Table C-5　Setup sub window Command list

Setting Item	Command	Page
Panel 切り替え	SEW	7-117

- **Setup:Setup panel**

Table C-6　Setup:Setup Panel Command list (3.2G PPG)

Setting Item		Command	Page
Common Setting	Pattern	PCO	7-50

Table C-7　Setup:Setup Panel Command list (3.2G ED)

Setting Item		Command	Page
Common Setting	Pattern	PCO	7-115

- **Setup:Frequency panel**

Table C-8　Setup:Frequency Panel Command list

Setting Item		Command	Page
Clock	Reference Frequency	RFC FRQ	7-51
Alarm monitor		PLL	7-52

- **Setup:Clock I/F panel**

Table C-9 Setup:Clock I/F panel Command list (3.2G PPG)

Setting Item		Command	Page
Grouping		CGR,	7-49
Output		CON,XCO	7-34
Duty		CDT,XDT	7-39,40
Delay		CDL	7-35
Level		CLL,XCL	7-35,36
Amplitude		CAP,XCA	7-36,37
Offset	Offset value	COS,XCF	7-37,38
	Offset base	OFS,OPS	7-38,39

Table C-10 Setup:Clock I/F panel Command list (3.2G ED)

Setting Item	Command	Page
Polarity	CPL	7-87
Delay	CPA	7-87
Termination	CTM	7-88

- **Setup:Data I/F panel**

Table C-11 Setup:Data I/F panel Command list (3.2G PPG)

Setting Item		Command	Page
Grouping		TRK	7-49
Output		DON,XDO	7-41
Cross Point		DCR,XDC	7-32,43
Level		DAL,XDL	7-43,44
Amplitude		DAP,NAP	7-44,45
Offset	Offset value	DOS,NOS	7-45,46
	Offset base	OFS,OPS	7-38,39

Table C-12 Setup:Data I/F panel Command list (3.2G ED)

Setting Item	Command	Page
Level	DAL	7-88
Threshold	DTH	7-89
Termination	DTM	7-89

- **Setup:Pattern panel**

Table C-13 Setup:Pattern panel Command list (3.2G PPG)

Setting Item		Command	Page
Output Pattern		PTO	7-15
Pattern Setting	Pattern	PTN	7-16
Pattern Setting	Mark Ratio	MRK	7-17
(PRBS)	Bit Shift	SFT	7-18
	Logic	LGC	7-18
Pattern Setting	Pattern Length	ZPL	7-19
(Zero-Subst)	Zero Length	ZLN	7-19
	Logic	LGC	7-18
Pattern Setting	Pattern Length	DLN	7-20
(PRGM)	Logic	LGC	7-18
Burst Setting	Mode	BRM	7-20
	Burst Cycle	BRC	7-21
	Enable Length	BRE	7-21
PRGM Pattern	String data	PDT	7-25
Data	Binary data	BDT	7-27
PRGM Pattern	Address specification	DRA	7-28
Data Reversion	Delta specification	DRD	
PRGM editting commands		PAG,BIT,	7-22 to 7-24
		WRT,RED,ALL,PST	7-29 to 7-30

Table C-14 Setup:Pattern panel Command list (3.2G ED)

Setting Item		Command	Page
Pattern Input		PTI	7-53
Pattern Setting	Pattern	PTN	7-54
Pattern Setting	Mark Ratio	MRK	7-55
(PRBS)	Bit Shift	SFT	7-56
	Logic	LGC	7-56
Pattern Setting	Pattern Length	ZPL	7-57
(Zero-Subst)	Zero Length	ZLN	7-57
	Logic	LGC	7-56
Pattern Setting	Pattern Length	DLN	7-58
(PRGM)	Logic	LGC	7-56
Sync Setting	Auto Sync	SYN	7-58
	Sync Threshold Gain	GTH	7-60
	Sync Threshold Loss	LTH	7-59
	Internal Threshold	ITH	7-60
	Pattern Sync source	FSY	7-61
	Frame Length	FLN	7-61
PRGM Pattern	String data	PDT	7-65
Data Setting	Binary data	BDT	7-67
PRGM Pattern	Address specification	DRA	7-68
Data Reversion	Delta specification	DRD	
PRGM editting commands		WRT,RED,ALL,PST,	7-69 to 7-70
		DRA,PAG,BIT	7-62 to 7-68

- **Setup:Trigger I/F panel**

Table C-15 Setup:Trigger I/F panel Command list (3.2G PPG)

Setting Item	Command	Page
Source	SOP	7-46
Position	PSP	7-47

Table C-16 Setup:Trigger I/F panel Command list (3.2G ED)

Setting Item	Command	Page
Source	SOP	7-115

- **Setup:Utility panel**

Table C-17 Setup:Utility panel Command list (3.2G PPG)

Setting Item	Command	Page
PRBS Logic	PML	7-32
PRGM Logic	GML	7-32
DC Impedance	CIM	7-40
	DIM	7-42

Table C-18 Setup:Utility panel Command list (3.2G ED)

Setting Item	Command	Page
PRBS Logic	PML	7-70
PRGM Logic	GML	7-71
MEAS. Restart	MRS	7-72

- **Test Menusub window**

Table C-19 Test Menu sub window Command list

Setting Item	Command	Page
Panel switching	TEW	7-118

- **Test Menu:Measurement panel**

Table C-20 Test Menu:Measurement panel Command list

Setting Item	Command	Page
Test Item	TIT	7-71

Table C-21 Test Menu:Measurement panel Command list (Test Item:Error/Alarm)

Setting Item		Command	Page
MEAS.Mode		MOD	7-73
MEAS.Time		PRD	7-73
Timed Start	Timed Start	TSS	7-74
	Start Time	STI	
Error Type		ETY	7-75
Route Mask		SCH	7-75
EI/%EFI	Interval	ITV	7-91
Performance	Measurement	PRF	7-91
	Threshold	ETH	7-92
Threshold EI/%EFI Measurement		TEI	7-92
Power Fail Measurement		PFM	7-93
Clock Loss Evaluation		CLS	7-90
Sync Loss Evaluation		SLS	7-90

Table C-22 Test Menu:Measurement panel Command list (Test Item:Eye Margin)

Setting Item	Command	Page
MEAS.Mode	EMD	7-81
	EYT	
Error Threshold	EMT	7-82
	EMR	7-83
Threshold Select	EDT	7-83

- **Test Menu:Error Addition panel**

Table C-23 Test Menu:Error Addition panel Command list

Setting Item	Command	Page
Error Addition	EAD	7-31
Single		
Rate		
Addition Route	ECH	7-31

■ **Result sub window (Test Item common)**

Table C-24 Result sub window Command list (3.2G ED)

Setting Item		Command	Page
Auto Search	Mode	ASM	7-85
	PRBS Pattern Search	PSH	7-86
	Start Stop Status	SRH	7-85
Alarm Monitor		CLI,SLI ERS,BES POF	7-111,112 7-112,113 7-113
Start		STA	7-71
Stop		STO	7-72
Time Display Selection		TIM	7-118
Stores result in th buffer		BST	7-111
Clears the buffer		BCL	7-111

■ **Result sub window (Test Item:Error/Alarm)**

Table C-25 Result sub window (Error/Alarm) Command list

Setting Item	Command	Page
Start time	MSA	7-76
End time	MSO	7-76
Measurement status	MSR	7-77
Alarm occurrence time	AOT	7-79
Alarm recovery time	ART	7-80
Elapsed time	MLP	7-77
Left time	ETI	7-78
Intermediate time	INT	7-78
Measurement result data	ER,EC,EI,EFI,FRQ,PFP,PFC,THE, THF,AIN,OER,OEC,FMT,IMD CLI,SLI,ERS,BES,POF, THR,EPF,ERP,ALP,FRP	7-93 to 7-106 7-111 to 7-113 7-144 to 7-146
Panel switching	REW	7-119
Display	CUR	7-119

● **Result:ALL panel**

Table C-26 Result:ALL panel Command list

Setting Item	Command	Page
Performance	PFD	7-120

- **Result:ZoomPanel**

Table C-27 Result:Zoom panel Command list

Setting Item		Command	Page
Display1	Display	ZOM	7-120 to 7-123
	Item	ZOI	
	Performance Item	ZOP	
	Threshold EI/%EI Item	ZOT	
Display2	Display	ZOM	
	Item	ZOI	
	Performance Item	ZOP	
	Threshold EI/%EI Item	ZOT	
Display3	Display	ZOM	
	Item	ZOI	
	Performance Item	ZOP	
	Threshold EI/%EI Item	ZOT	
Display4	Display	ZOM	
	Item	ZOI	
	Performance Item	ZOP	
	Threshold EI/%EI Item	ZOT	

- **Result:MonitorPanel**

Table C-28 Result:Monitor panel Command list

Setting Item	Command	Page
History Reset	HRE	7-123

■ **Result sub window (Test Item:Eye Margin)**

Table C-29 Result sub window (Eye Margin) Command list

Setting Item		Command	Page
Start time		MSA	7-76
End time		MSO	7-76
Measurement status		MSR	7-77
Elapsed time		MLP	7-77
Number of valid eye diagram points		EDP	7-84
Measurement result		EMM,EDM	7-110,7-109
Display		EDD	7-124
Scale	Auto Scale	ASC	7-125
	Threshold Voltage	VSC	7-125
	Phase	PSC	7-126
Marker,△Marker	Display ON/OFF	EMA	7-126
	Movement mode	MKM	7-127
	Subject	MKT	7-127
	Movement	MMV	7-128
	Position	MKP	7-129

■ **Customize sub window**

Table C-30 Customize sub window Command list

Setting Item		Command	Page
Setup1	Unit	CUS	7-130 to 7-132
	Port	CUP	
	Item	CUI	
Setup2	Unit	CUS	
	Port	CUP	
	Item	CUI	
Setup3	Unit	CUS	
	Port	CUP	
	Item	CUI	
Setup4	Unit	CUS	
	Port	CUP	
	Item	CUI	
Setup5	Unit	CUS	
	Port	CUP	
	Item	CUI	
Setup6	Unit	CUS	
	Port	CUP	
	Item	CUI	
Pattern	Unit	CPT	7-133
	Offset	CPF	
Result	Item	CRI	7-134

■ **Other Commands**

Table C-31 other Command list

Setting Item	Command	Page
I/O buffer size	BUF	7-137
Status register	SRQ,STB,ESE,ESR,EES,EER	5-21 to 5-26
Print out any string	PTX	7-144
Delay setting status	DLY	7-48,7-114
FD access status	MAC	7-143
LSB,MSB swapping	SWP	7-150
Termination setting	TRM	7-151

■ SCPI version

The MP1632C conforms to SCPI version 1995.0.

■ SCPI commands

Among the command used by the MP1632C, the following commands are defined by SCPI 1995.0.

:SYSTem
- :DATE
- :DATE?
- :TIME
- :TIME?
- :ERRor?
- :VERSion?

:STATus
- :PRESet
- :OPERation
  - [:EVENt]?
  - :CONDtion?
  - :ENABle
  - :ENABle?
  - :PTRansition
  - :PTRansition?
  - :NTRansition
  - :NTRansition?
  - :INSTrument
    - [:EVENt]?
    - :CONDtion?
    - :ENABle
    - :ENABle?
    - :PTRansition
    - :PTRansition?
    - :NTRansition
    - :NTRansition?
  - :QUEStionable
    - [:EVENt]?
    - :CONDtion?
    - :ENABle
    - :ENABle?
    - :PTRansition
    - :PTRansition?
    - :NTRansition
    - :NTRansition?

*IDN?
*RST
*TST?
*OPC
*OPC?
*WAI
*CLS
*ESE
*ESE?
*ESR?
*SRE
*SRE?
*STB?
*TRG
*OPT?

■ Non-SCPI commands

Among the command used by the MP1632C, the following commands are not defined by SCPI 1995.0.

:SOURce
- :PATTern
  - :OMODE
  - :OMODE?
  - :TYPE
  - :TYPE?
  - :PRBS
    - :MRATio
    - :MRATio?
    - :BSHift
    - :BSHift?
  - :ZSUBstitute
    - :LENGth
    - :LENGth?
    - :ZLENgth
    - :ZLENgth?
    - :LOGic
    - :LOGic?
    - :MODE
    - :MODE?
  - :PROGram
    - :LENGth
    - :LENGth?
    - :LOGic
    - :LOGic?
    - :MODE
    - :MODE?
  - :DATA
    - :WHOLe
    - :WHOLe?
  - :BDATa
    - :WHOLe
    - :WHOLe?
  - :DREVerse
    - :ADDRess
    - :DELTa
  - :EADDition
    - :SET
    - :SET?
    - :SINGle
    - :RATE
    - :RATE?
    - :ROUTe
    - :ROUTe?
  - :LOGic
    - :PRBS
    - :PRBS?
    - :PRGM
    - :PRGM?

:OUTPut
- :CLOCk
  - :OUTPut
  - :OUTPut?
  - :DELay
  - :DELay?
  - :LEVel
  - :LEVel?
  - :AMPLitude
  - :AMPLitude?
  - :OFFSet
  - :OFFSet?
  - :OREFerence
  - :OREFerence?
  - :Duty
  - :Duty?
  - :IMPedance
  - :IMPedance?
- :DATA
  - :OUTPut
  - :OUTPut?
  - :IMPedance
  - :IMPedance?
  - :CPOint
  - :CPOint?
  - :LEVel
  - :LEVel?
  - :AMPLitude
  - :AMPLitude?
  - :OFFSet
  - :OFFSet?
  - :OREFerence
  - :OREFerence?
- :SYNC
  - :SOURce
  - :SOURce?
  - :POSition
  - :POSition?
- :CLOCk
  - :FREQuency
  - :FREQuency?

```
:CALCulate
    :DATA
        :MONitor?

:INSTrument
    :COUPle
        :CLOCk
            :TRACking
            :TRACking?
        :DATA
            :TRACking
            :TRACking?
        :PATTern
            :SET
            :SET?

:SENSe
    :PATTern
        :IMODe
        :IMODe?
        :TYPE
        :TYPE?
        :PRBS
            :MRATio
            :MRATio?
            :BSHift
            :BSHift?
        :ZSUBstitute
            :LENGth
            :LENGth?
            :ZLENgth
            :ZLENgth?
            :LOGic
            :LOGic?
            :MODE
            :MODE?
        :PROGram
            :LENGth
            :LENGth?
            :LOGic
            :LOGic?
            :MODE
            :MODE?
        :SYNC
            :ASYNc
            :ASYNc?
            :THReshold
                :LOSS
                :LOSS?
                :GAIN
                :GAIN?
                :INTernal
                :INTernal?
            :PSMode
            :PSMode?
            :FLENgth
            :FLENgth?
```

```
:SENSe:PATTern (continued)
        :BURSt
            :ELENgth
            :ELENgth?
        :DATA
            :WHOLe
            :WHOLe?
        :BDATA
            :WHOLe
            :WHOLe?
        :DREVerse
            :ADDRess
            :DELTa
        :LOGic
            :PRBS
            :PRBS?
            :PRGM
            :PRGM?
    :MEASure
        :TEST
        :TEST?
        :STARt
        :STOP
        :MREStart
        :MREStart?
        :EALarm
            :MODE
            :MODE?
            :PERiod
            :PERiod?
            :BTIMe
                :SET
                :SET?
                :STARt
                :STARt?
            :ERRor
                :TYPE
                :TYPE?
            :MASK
                :ROUTe
                :ROUTe?
            :STARt?
            :STOP?
            :STATe?
            :ELAPsed?
            :TIMed?
            :ITIMe?
            :AOCCur?
            :ARECver?
```

:SENSe:MEASure (continued)
:EMARgin
:MODE
:MODE?
:TYPE
:TYPE?
MARGin
:THReshold
:THReshold?
:RESolution
:RESolution?
:DIAGram
:THReshold
:THReshold?
:POINt?
:STATe?
:STARt?
:STOP?
:ELAPsed?
:ITIMe?
:ASEarch
:MODE
:MODE?
:STARt
:STOP
:PATTern
:PATTern?
:STATe?

:INPut
:CLOCk
:POLarity
:POLarity?
:DELay
:DELay?
:TERMination
:TERMination?
:DATA
:LEVel
:LEVel?
:THReshold
:THReshold?
:TERMination
:TERMination?

:CALCulate
:EALarm
:CLEValuation
:CLEValuation?
:SLEValuation
:SLEValuation?
:ERRor
:INTerval
:INTerval?
:PERFormance
:MEASurement
:MEASurement?
:THReshold
:THReshold?
:TEINterval
:MEASurement
:MEASurement?
:PFAil
:PFAil?
:DATA
:EALarm?
:EMARgin?
:STORe
:CLEar
:MONitor?

:OUTPut
:SYNC
:SOURce
:SOURce?
:CLOCk
:FREQuency
:FREQuency?
:RCLock
:SELect
:SELect?

```
:DISPlay
    :WINDow
        :OPEN
        :OPEN?
        :CLOSe
    :SYSTem
        [:NAME]
        [:NAME]?
    :SETup
        [:NAME]
        [:NAME]?
    :TEST
        [:NAME]
        [:NAME]?
    :RESult
        :TIME
        :TIME?
        :EALarm
            [:NAME]
            [:NAME]?
            :MODE
            :MODE?
            :ALL
                :PTYPE
                :PTYPE?
            :ZOOM1|..|:ZOOM4
                :SET
                :SET?
                :ITEM
                :ITEM?
                :PITem
                :PITem?
                :TITem
                :TITem?
            :HRESet
        :EMARgin
            :ERATe
            :ERATe?
            :SCALe
                :ASCale
                :VOLTage
                :VOLTage?
                :PHASe
                :PHASe?
            :MARKer
                :SET
                :SET?
                :MODE
                :MODE?
                :ERATe
                :ERATe?
                :MOVE
                :POSition?
```

```
:DISPlay（continued）
    :RESult
        :EMARgin
            :TEMPlate
                :SELect
                :SELect?
                :MOVE
                :PNUMber
                :PNUMber?
                :OFFSet
                    :VOLTage
                    :VOLTage?
                    :PHASe
                    :PHASe?
                :POINt
                    :VOLTage
                    :VOLTage?
                    :PHASe
                    :PHASe?
                :CONNect
                :SELect?
    :CUSTomize
        :SETup1|..|:SETup6
            :UNIT
            :UNIT?
            :ITEM
            :ITEM?
            :PORT
            :PORT?
        :PATTern
            :UNIT
            :UNIT?
            :OFFSet
            :OFFSet?
        :RESult
            :ITEM
            :ITEM?
```

:SYSTem
:BEEPer
:ERRor
:SET
:SET?
:ALARm
:SET
:SET?
:SYSTem
:SET
:SET?
:TYPE
:TYPE?
:MODE?
:BSIZe?
:ORGanization
:HARDware?
:SOFTware?
:MMEMory
:RECall
:STORe
:QRECall
:QSTore
:CATalog?
:MEMory
:INITialize
:PRINt
:COPY
:TEXT
:EALarm
:PATTern
:TABLe
:ADDRess
:ALL
:EMARgin
:REMARgin
:TERMination
:TERMination?

:STATus
:QUEStionable
:MONitor
[:EVENt]?
:CONDtion?
:ENABle
:ENABle?
:PTRansition
:PTRansition?
:NTRansition
:NTRansition?
:SLOT1|:SLOT2|:SLOT4
[:EVENt]?
:CONDtion?
:ENABle
:ENABle?
:PTRansition
:PTRansition?
:NTRansition
:NTRansition?
:SLOT3|:SLOT4
:G32P|:G32E
[:EVENt]?
:CONDtion?
:ENABle
:ENABle?
:PTRansition
:PTRansition?
:NTRansition
:NTRansition?