MP1580A
Portable 2.5G/10G Analyzer
Operation Manual
Vol. 2
Remote Control

Third Edition

Read this manual before using the equipment.
Keep this manual with the equipment.

Measurement Solutions
ANRITSU CORPORATION

Document No.: M-W1890AE-3.0

Safety Symbols

To prevent the risk of personal injury or loss related to equipment malfunction, Anritsu Corporation uses the following safety symbols to indicate safety-related information. Insure that you clearly understand the meanings of the symbols BEFORE using the equipment.
Some or all of the following five symbols may not be used on all Anritsu equipment. In addition, there may be other labels attached to products which are not shown in the diagrams in this manual.

Symbols used in manual

DANGER This indicates a very dangerous procedure that could result in serious injury or death if not performed properly.

WARNING This indicates a hazardous procedure that could result in serious injury or death if not performed properly.

CAUTION This indicates a hazardous procedure or danger that could result in light-to-severe injury, or loss related to equipment malfunction, if proper precautions are not taken.

Safety Symbols Used on Equipment and in Manual

The following safety symbols are used inside or on the equipment near operation locations to provide information about safety items and operation precautions. Insure that you clearly understand the meanings of the symbols and take the necessary precautions BEFORE using the equipment.

This indicates a prohibited operation. The prohibited operation is indicated symbolically in or near the barred circle.

This indicates an obligatory safety precaution. The obligatory operation is indicated symbolically in or near the circle.

This indicates warning or caution. The contents are indicated symbolically in or near the triangle.

This indicates a note. The contents are described in the box.

These indicate that the marked part should be recycled.

MP1580A
Portable 2.5G/10G Analyzer
Operation Manual Vol. 2 Remote Control

1 March 2001 (First Edition)
29 August 2001 (Third Edition)

For Safety

WARNING ⚠

1. ALWAYS refer to the operation manual when working near locations at which the alert mark shown on the left is attached. If the operation, etc., is performed without heeding the advice in the operation manual, there is a risk of personal injury. In addition, the equipment performance may be reduced.
   Moreover, this alert mark is sometimes used with other marks and descriptions indicating other dangers.

or

2. When supplying power to this equipment, connect the accessory 3-pin power cord to a 3-pin grounded power outlet. If a grounded 3-pin outlet is not available, before supplying power to the equipment, use a conversion adapter and ground the green wire, or connect the frame ground on the rear panel of the equipment to ground. If power is supplied without grounding the equipment, there is a risk of receiving a severe or fatal electric shock.

Repair

3. This equipment cannot be repaired by the user. DO NOT attempt to open the cabinet or to disassemble internal parts. Only Anritsu-trained service personnel or staff from your sales representative with a knowledge of electrical fire and shock hazards should service this equipment. There are high-voltage parts in this equipment presenting a risk of severe injury or fatal electric shock to untrained personnel. In addition, there is a risk of damage to precision parts.

Falling Over

4. This equipment should be used in the correct position. If the cabinet is turned on its side, etc., it will be unstable and may be damaged if it falls over as a result of receiving a slight mechanical shock.

For Safety

WARNING ⚠

Battery Fluid

5. DO NOT short the battery terminals and never attempt to disassemble it or dispose of it in a fire. If the battery is damaged by any of these actions, the battery fluid may leak.
   This fluid is poisonous.
   DO NOT touch it, ingest it, or get in your eyes. If it is accidentally ingested, spit it out immediately, rinse your mouth with water and seek medical help. If it enters your eyes accidentally, do not rub your eyes, irrigate them with clean running water and seek medical help. If the liquid gets on your skin or clothes, wash it off carefully and thoroughly.

LCD

6. This instrument uses a Liquid Crystal Display (LCD); DO NOT subject the instrument to excessive force or drop it. If the LCD is subjected to strong mechanical shock, it may break and liquid may leak.
   This liquid is very caustic and poisonous.
   DO NOT touch it, ingest it, or get in your eyes. If it is ingested accidentally, spit it out immediately, rinse your mouth with water and seek medical help. If it enters your eyes accidentally, do not rub your eyes, irrigate them with clean running water and seek medical help. If the liquid gets on your skin or clothes, wash it off carefully and thoroughly.

For Safety

CAUTION ⚠

Changing Fuse

CAUTION ⚠

1. Before changing the fuses, ALWAYS remove the power cord from the poweroutlet and replace the blown fuses. ALWAYS use new fuses of the type and rating specified on the fuse marking on the rear panel of the cabinet.

   T___A indicates a time-lag fuse.
   ___A or F___ A indicate a normal fusing type fuse.

   There is risk of receiving a fatal electric shock if the fuses are replaced with the power cord connected.

Cleaning

2. Keep the power supply and cooling fan free of dust.
   - Clean the power inlet regularly. If dust accumulates around the power pins, there is a risk of fire.
   - Keep the cooling fan clean so that the ventilation holes are not obstructed. If the ventilation is obstructed, the cabinet may overheat and catch fire.

Floppy Disk

3. Floppy disk is easily damaged by dust. Please, pay attention to the dust.
   For maintaining the normal operation, clean the magnetic head, periodically.

For Safety

CAUTION ⚠

Changing memory back-up battery

This equipment uses a lithium battery to back-up the memory. This battery must be replaced by a service engineer when it has reached the end of its useful life; contact the Anritsu sales section or your nearest representative.

Note: The battery used in this equipment has a maximum useful life of 7 years. It should be changed before this period has elapsed.

Equipment Certificate

Anritsu Corporation certifies that this equipment was tested before shipment using calibrated measuring instruments with direct traceability to public testing organizations recognized by national research laboratories including the Electrotechnical Laboratory, the National Research Laboratory of Metrology and the Communications Research Laboratory, and was found to meet the published specifications.

Anritsu Warranty

Anritsu Corporation will repair this equipment free-of-charge if a malfunction occurs within 1 year after shipment due to a manufacturing fault, provided that this warranty is rendered void under any or all of the following conditions.

- The fault is outside the scope of the warranty conditions described in the operation manual.
- The fault is due to mishandling, misuse, or unauthorized modification or repair of the equipment by the customer.
- The fault is due to severe usage clearly exceeding normal usage.
- The fault is due to improper or insufficient maintenance by the customer.
- The fault is due to natural disaster including fire, flooding, earthquake, etc.
- The fault is due to use of non-specified peripheral equipment, peripheral parts, consumables, etc.
- The fault is due to use of a non-specified power supply or in a non-specified installation location.

In addition, this warranty is valid only for the original equipment purchaser. It is not transferable if the equipment is resold.

Anritsu Corporation will not accept liability for equipment faults due to unforeseen and unusual circumstances, nor for faults due to mishandling by the customer.

Anritsu Corporation Contact

If this equipment develops a fault, contact Anritsu Corporation or its representatives at the address in this manual.

Notes On Export Management

This product and its manuals may require an Export License/Approval by the Government of the product's country of origin for re-export from your country.
Before re-exporting the product or manuals, please contact us to confirm whether they are export-controlled items or not.

Disposing of Product

This equipment uses chemical compound semiconductors including arsenide, and a lithium battery. At the end of its life, the equipment should be recycled or disposed properly according to the local disposal regulations.

CE Marking

Anritsu affixes the CE Conformity Marking on the following product (s) in accordance with the Council Directive 93/68/EEC to indicate that they conform with the EMC directive of the European Union (EU).

CE Conformity Marking

CE

1. Product Name/Model Name

Product Name: Portable 2.5G/10G Analyzer, 2.5G/10G Jitter Unit
Model Name: MP1580A, MU150018A

2. Applied Directive

EMC: Council Directive 89/336/EEC
LVD: Council Directive 73/23/EEC

3. Applied Standards

EMC:

Emission: EN61326: 1997/A1: 1998 (Class A)
Immunity: EN61326: 1997/A1: 1998 (Annex A)

	Performance Criteria*
IEC61000-4-2 (ESD)	B
IEC61000-4-3 (EMF)	A
IEC61000-4-4 (Burst)	B
IEC61000-4-5 (Surge)	B
IEC61000-4-6 (CRF)	A
IEC61000-4-8 (RPFMF)	A
IEC61000-4-11 (V dip/short)	B

*: Performance Criteria

A: During testing normal performance within the specification limits.

B: During testing, temporary degradation, or loss of function or which is self-recovering.

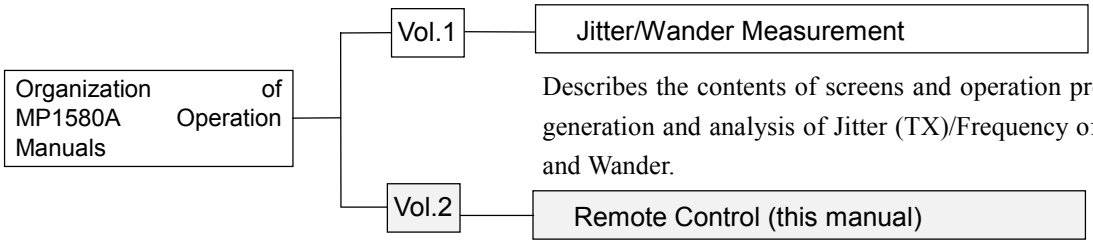
Harmonic current emissions:

EN61000-3-2: 1995/A2: 1998 (Class □ equipment)

LVD: EN61010-1: 1993/A2: 1995 (Installation Category II, Pollution Degree 2)

About MP1580A Operation Manuals

MP1580A Portable 2.5G/10G Analyzer Operation Manuals comprise of the following two documents. Use them properly according to the usage purpose.



Describes the contents of screens and operation procedures for the generation and analysis of Jitter (TX)/Frequency offset, Jitter (RX) and Wander.

Describes the remote control and program examples using the GPIB, RS-232C, and Ethernet interfaces (MP1580A is compatible with SPCI standard).

This Operation Manual describes the followings.

This manual (MP1580A Portable 2.5G/10G Analyzer Operation Manual Vol. 2 Remote Control) mainly describes commands and operation procedures for remote control of MP1580A.

Screen Names

MP1580A has 4 major screens, namely, "Setup", "Test Menu", "Result" and "Analyze", and each major screen has its own subscreens (for details, see "Section 3 Screens and Parameter Setting" of Vol. 1).

For example, if "Setup" is selected as the main screen and "Interface" as the subscreen, they are stated as the "Setup: Interface" screen in this manual.

Front Panel Keys

In this manual, (*****) [*****] indicate front panel keys.

Contents

Section 1 General

This section explains the outline of the remote control function for MP1580A Portable 2.5G/10G Analyzer.

1.1 Connection

MP1580A supports the GPIB, RS-232C, and Ethernet interfaces for remote control. (These interfaces can be used by installing required option.)

Compliance

The GPIB interface of MP1580A conforms to the IEEE (Institute of Electrical and Electronic Engineers) std 488.1-1987. The software specifications conform to the IEEE488.2 and SCPI (Standard Commands for Programmable Instruments) (see Appendix).

Allocating an option

Select the option to be used for remote control from GPIB, RE-232C and Ethernet options in the Setup: System screen. (See Section 2.5 "GPIB Setting" Section 3.3 "RS-232C Setting" and Section 4.6 "Ethernet Settings.")

1.2 Device Messages

Device messages are the data messages communicated between the controller and device, and classified into program and response messages. Device messages can be used when MP1580A functions as a device.

Program message

The ASCII data message which is transmitted from the controller to the device. This message consists of the program command (Command) and program query (Query).

(→ *For details, see "Section 5 Listener Input Format"*)

The program command (Command) consists of the equipment-unique command and IEEE488.2 common command. IEEE488.2 common commands are applicable for other IEEE488.2-compliant measurement instruments.

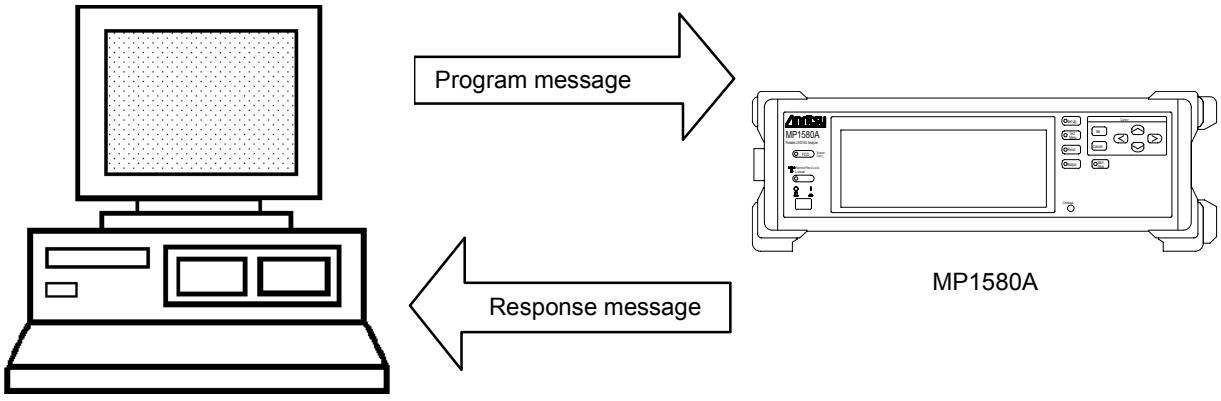
(→ *For details on equipment-unique commands, see Section 11*)
(→ *For details on IEEE488.2 common commands, see Section 7*)

The program query (Query) is a command to obtain a response message form a device. It is transferred from the controller to the device to receive a response message from the device at the controller.

Response message

The ASCII data message which is transmitted from the device to the controller.

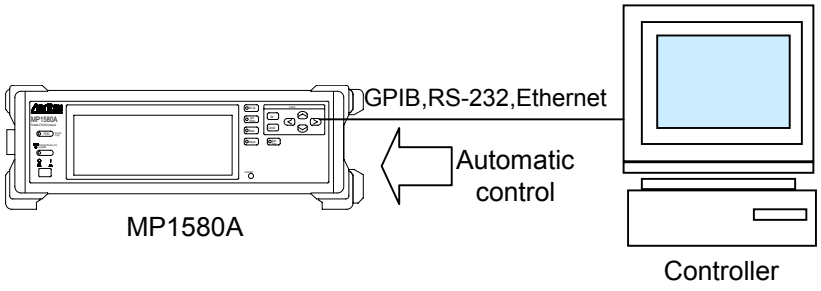
(→ *For details, see "Section 6 Talker Output Format"*)



1.3 Typical System Up

Controlling from the host computer

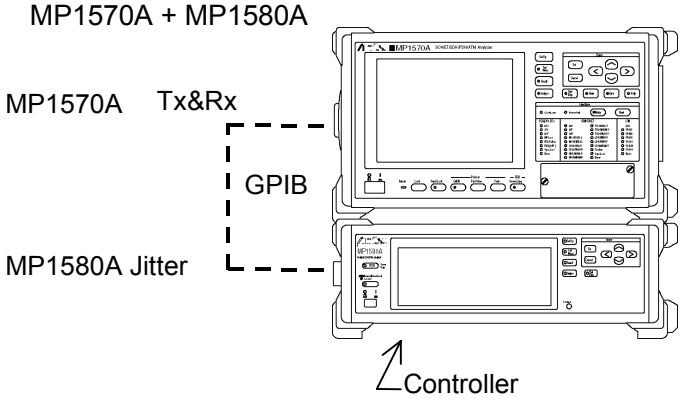
MP1580A Portable 2.5G/10G Analyzer can operate as a device when connected to an external controller and perform automatic measurement.



Automatic jitter/wander measurement using a controller

When the MP1580A functions as a controller of the GPIB function, the remote interface of the GPIB function cannot be used.

MP1570A + MP1580A



Section 2 GPIB Interface

This section explains the GPIB interface function and settings when using the GPIB interface option.

2.1 GPIB Interface Function

MP1580A has both the device and controller functions when connected using the GPIB interface. When the GPIB interface is used as a remote interface, MP1580A can be used as a device. MP1580A interface function corresponding to the IEEE488.2 standard is shown in the table below.

Code	Interface function	IEEE488.2 standard
SH1	With all source handshake functions	All functions are standard equipped.
AH1	With all acceptor handshake functions	All functions are standard equipped.
T5 TE5	With basic talker function, talk only mode function and talker clear function by MLA	Device must have T5, T6, TE5 or TE6 subset
L3 LE3	With basic listener function, listen only mode function and listener clear function by MTA	Device must have L3, L4, LE3 or LE4 subset
SR1	With all service request functions	All functions are standard equipped.
RL1	With all remote/local functions	RL0 (without function) or RL1 (all functions) all remote/local functions
PP0	Without parallel pole function	PP0 (without function) or PP1 (all functions)
DC1	With all device clear functions	All functions are standard equipped.
DT1	With all device trigger functions	DT0 (without function) or DT1 (all functions)
C1 C2 C3 C4 C7	With control functions other than parallel pole	C0 (without function), C4 and C5, or one of C7, C9 and C11.

(Reference)

C1: System controller

C2: Send IFC and Take Charge.

C3: Send REN.

C4: Respond to SRQ.

C7: Send I.F Messages, Receive/Pass Control, Pass control to Self.

2.2 Device Messages (GPIB)

When MP1580A is used as a device, the device messages are communicated between the controller and device via the system interface when the bus is set to the data mode (the ATN line is "H" level).

The data messages are communicated via the I/O buffer of the device. The I/O buffer is briefly explained below.

Input buffer	Output queue
This is FIFO (First in First out) type memory area which tentatively saves DAB (program and query messages) before conducting syntax analysis. Input buffer capacity of MP1580A is 512 bytes.	This is FIFO type queue memory area. All DABs (response messages) transmitted from the device to controller are stored in this memory until read by controller is complete.

2.3 Bus Commands

The bus commands are the internal communications within the interface transmitted when the bus mode is set to the command mode (the ATN line is "L" level).

The bus commands are shown in the table below.

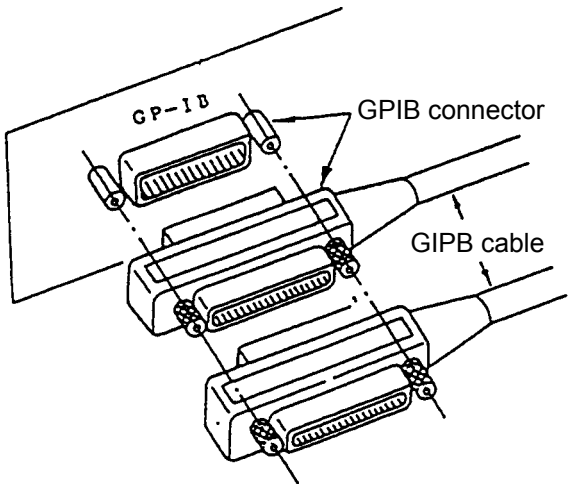
Bus command	Operation
DCL (Device clear)	Initializes all message communications for all units connected to the GPIB bus.
SDC (Selected Device Clear)	Initializes message communications for the addressed unit. Operation is the same as DCL.
GET (Group Execute Trigger)	Operation is the same as when Start /Stop key is pressed.
IFC (Interface Clear)	Initializes the interface.

2.4 GPIB Cable Connection

Connect the GPIB cable to the GPIB connector on the rear panel.

The limitations for the system using GPIB are shown below. Observe the conditions for device connection:

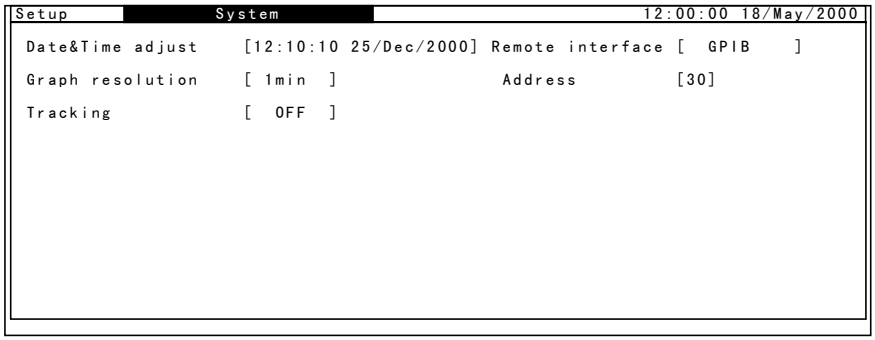
Number of devices to be connected ≤ 15
Total length of cables ≤ 2 m × number of devices
(maximum 20 m)



2.5 GPIB Setting

Set MP1580A to the local mode, and then perform settings in the "Setup : System" screen as shown below.

(When using MP1580A as a remote interface, check that it operates as a device.)



Setting procedure

(1) Display Setup:System screen.

(2) Set "Tracking" to "OFF."

(3) Set each parameter, depending on the table below.

Setting	Item	Setting value
On/OFF setting for the tracking function (can be used as a remote interface when this item is set to OFF.)	Tracking	OFF
Interface selection for remote control	Remote Interface	GPIB
Address setting	Address	0 to 30

(Reference)

- Setting the item "Tracking" to "ON (MP1570A)" enables Jitter tolerance, Jitter sweep, Freq. sweep and Wander sweep measurements of automatic jitter measurement.
- When MP1580A is used as a controller, remote control is disabled.
- The GPIB address of MP1580A is fixed to 0 when used as a controller.

2.6 Device Initialization

The IEEE488.2 specifies the system initialization in the following three levels.

Level	Initialization type	Description	Remarks
1	Bus initialization	Initializes all interface functions connected to the bus by the IFC message form the controller.	Can be performed also in combination with a command of other levels. Note that this should be performed prior to level 2 initialization.
2	Message initialization	Disables the function to report to the controller of completion of initialization or operation for initialization of message exchange, for all devices connected to GPIB by GPIB bus command of DCL, and for the specified device by bus command of SDC.	Can be performed also in combination with a command of other levels. Note that this should be performed prior to level 3 initialization.
3	Device initialization	Using *RST command, the device is returned to the unique, known condition regardless of the past operating conditions.	Can be performed also in combination with a command of other levels. Note that this should be performed after levels 1/2 initialization.

2.6.1 Bus initialization

IFC — Bus initialization by IFC statement

Function — Interface functions of all devices connected to the GPIB bus lines are initialized by setting the IFC line to active for approximately 100 μs. Only the system controller can transmit the IFC.

Format — IFCΔ @ select code

Example use — IFC @1

2.6.2 Message initialization

DCL, SDC — Initialization of message exchange by DCL or SDC bus command

Function — Initializes message exchange for all devices or specified devices on the GPIB.

Initializes message exchange when the area inside a device relating to the message exchange has become improper status to be controlled by the controller because of execution of another program, although the panel setting status need not be changed. The controller is able to transmit new commands after initializing message exchange.

DCL: Initializes the message exchange of all devices on the GPIB.

SDC: Initializes the message exchange of the specified devices only.

Format — DCLΔ @ select code[primary address][secondary address]

Example use — DCL @1 (Initializes message exchange of all devices connected to the bus. Transmits DCL.)

DCL @103 (Initializes message exchange of device of address 3. Transmits SDC.)

2.6.3 Device initialization

* RST — Device initialization by *RST command.

Function — This function initializes a function unique to the device to the known status, regardless of the past operating history.

With MP1580A, the status immediately after initialization is the same as those immediately after power on.

See the section for the common commands.

2.6.4 Device status immediately after power on

MP1580A is set to the following conditions, immediately after the power switch is turned on.

- The settings immediately before the preceding power-off are reproduced.
- The input/output buffer queues are cleared.
- The syntax analysis, execution control and response generation sections are reset.

Section 3 RS-232C Interface

This section explains the setting procedures when the optional RS-232C interface is used.

3.1 Device Messages

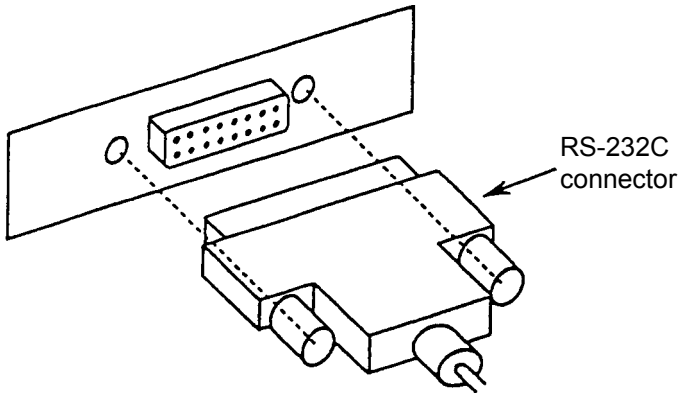
Device messages are the data messages communicated between the controller and device via the system interface. The data messages are communicated via the I/O buffers of the device.

The I/O buffers are briefly explained below.

Input buffer	Output queue
This is a FIFO (First in First out) type memory area which tentatively saves DAB (program and query messages) before conducting system analysis. Input buffer capacity of MP1580A is 512 bytes.	This is a FIFO type queue memory area. All DABs (response messages) transmitted from device to controller are stored in this memory until read by controller is complete.

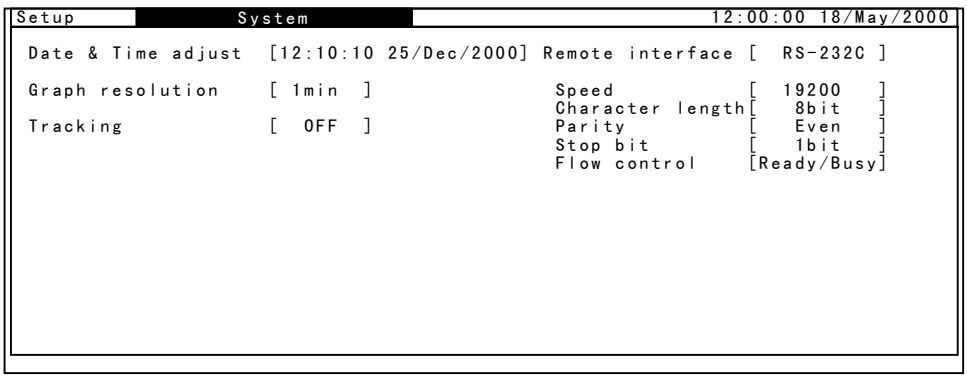
3.2 RS-232C Cable Connection

Connect the RS-232C cable to the RS-232C connector on the rear panel.



3.3 RS-232C Setting

When the RS-232C is used as the remote control interface, set MP1580A to the Local mode and enter settings listed below on the Setup : System screen using the front panel keys.



Setting procedure

(1) Display Setup:System screen.

(2) Set "Tracking" to "OFF."

(3) Set each parameter of "Remote interface", "Speed", "Character length", "Parity", "Stop bit", and "Flow control", depending on the table below.

Setting	Setting item	Setting range
Interface selection for remote control	Remote Interface	RS-232C
Bit rate setting	Speed	19200 bps or 9600 bps
Character length setting	Character length	8 bits or 7 bits
Parity setting	Parity	None: No parity Even: Even parity Odd: Odd parity
Stop-bit length setting	Stop Bit	2 bits,1 bit
Flow-control method setting	Flow control	Ready/Busy: RS-232C control line is used. XON/XOFF: XON/XOFF characters are used.

3.4 Device Initialization

The *RST command is supported as the command for initializing the device using the RS-232C interface.

3.4.1 Device initialization

＊RST Device initialization by *RST command.

Function This function initializes the functions unique to the device to the known status, regardless of the past operating history.
With MP1580A, the status immediately after initialization is the same as those immediately after power on.

3.4.2 Device status immediately after power on

MP1580A is set to the following conditions immediately after the power switch is turned on.

- The settings immediately before the preceding power-off are reproduced.
- The input and output buffer queues are cleared.
- The syntax analysis, execution control and response generation sections are reset.

Section 4 Ethernet Interface

This section explains settings when using an optional Ethernet interface.

4.1 Ethernet Interface

The network interface complies with IEEE802.3 and IEEE802.3u.

The transmission speed and method support the auto negotiation function.

Transmission speed: 100/10 Mbps

Transmission method: Full/Half duplex

4.2 Client and Server

Client: A personal computer or work station

Server: MP1580A

4.3 Ethernet Control

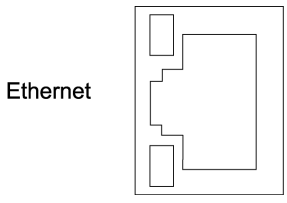
MP1580A acts as the server when using with an Ethernet option.

In this case, to make a connection specify 15800 (factory setting, fixed) for the port number at client-side. MP1580A should be connected to a TCP/IP network.

Ethernet control supports remote commands only. Other applications (HTTP, Telnet, etc.) are not supported.

4.4 Ethernet Cable Connection

Connect the Ethernet cable (10BASE-T/100BASE-TX) to the connector on the rear panel. Be sure to connect the Ethernet cable before turning on MP1580A.



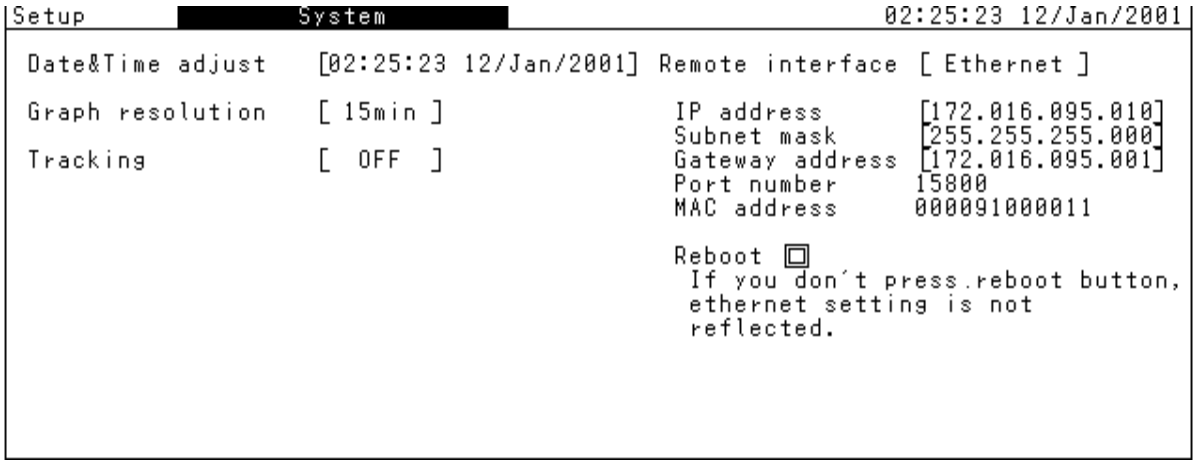
4.5 Device Messages

Device messages are data messages transmitted/received between the controller and device through the system interface. Data messages are transmitted/received via the device's I/O buffer. The I/O buffer is explained below:

Input buffer	Output queue
FIFO (First In First Out) memory area to temporarily store DAB (program messages or query messages) before syntax analysis. MP1580A has an input buffer of 511 bytes.	FIFO-type memory area for DAB queue. Stores all DAB (response messages) output from device to the controller until the controller reads all of them.

4.6 Ethernet Settings

When using Ethernet as the remote interface, set MP1580A to local status and set parameters in the Setup: System screen.



The setting procedures are shown below:

(1) Display the Setup: System screen.

(2) Set Tracking to OFF.

(3) Set parameters for Remote Interface, IP address, Subnet mask and Gateway address as shown in the table below.

(4) Press [Reboot] button and restart, or turn the power off and then on, to enable the parameter settings.

tting contents	Setting item	Setting range
Selecting the remote control interface	Remote Interface	Ethernet
IP address (Address for the unit (server-side))	IP address	000.000.000.000 to 255.255.255.255
Subnet mask	Subnet mask	000.000.000.000 to 255.255.255.255
Gateway address	Gateway address	000.000.000.000 to 255.255.255.255
Port number	Port number	Fixed
MAC address	MAC address	Fixed
Reboot button (initialize button)	Reboot	–

Setting procedure

(1) Press [Set Up].

"Setup" is indicated in the upper left part of the Setup main screen.

(2) Press [Set]. (A window opens.)

(3) Select "System."

(4) Press [Set]. (System screen appears.)

(5) Check that the item "Tracking" is set to "OFF."

(6) Set the item "Remote Interface" to "Ethernet."

(7) Select the item "IP address."

(8) Select the item "Subnet mask."

(9) Select the item "Gateway address."

(10) When the above settings are completed, press the Reboot button to restart the system. (System can be restarted by turning the power switch off and then on.)

Note:

- IP address

In a TCP/IP network, the connected units are identified by their IP addresses. The IP address therefore must be unique for each unit. An IP address is a 32-bit number and is normally expressed by delimiting 8 bits into four using dots (dot expression).

An IP address contains network information as well as unit (host) information. Each network class has different specifications on data length for the network portion and host portion of the IP address. For Class C, up to 254 hosts can be connected with a 24-bit network portion and 8-bit host portion. Normally, Classes A to C are used, though Classes A to E are available.

Class	Network portion length	Host portion length	Number of allocable hosts
A	8 bits	24 bits	16,777,214 units
B	16 bits	16 bits	65,534 units
C	24 bits	8 bits	254 units

- Subnet mask

Subnet mask indicates the network portion of an IP address when the network is divided into subnets. The network portion for the above IP address is set with '1' (including the expanded subnet portion), while the host portion is defined with '0.' If the subnet mask is not set correctly, IP packets cannot be transmitted/received normally when the connecting network uses subnet.

- Gateway

A unit called a "gateway" is used to connect one network to another. Gateways also include routers that are like dedicated units. Within a TCP/IP network, IP packets can be directly exchanged within the same network. For differing networks (an IP address terminal with a different network portion), one network must communicate with a machine inside of an other network through a gateway connecting the two.

4.7 Device Initialization

MP1580A supports the *RST command for device initialization when using the Ethernet interface.

4.7.1 Initializing a device

*RST: Initializes a device by using the *RST command.

Function: Returns a device to a known status regardless of usage history of device-unique functions. This command returns MP1580A to the status immediately after power is turned on.

4.7.2 Device status when power turned on

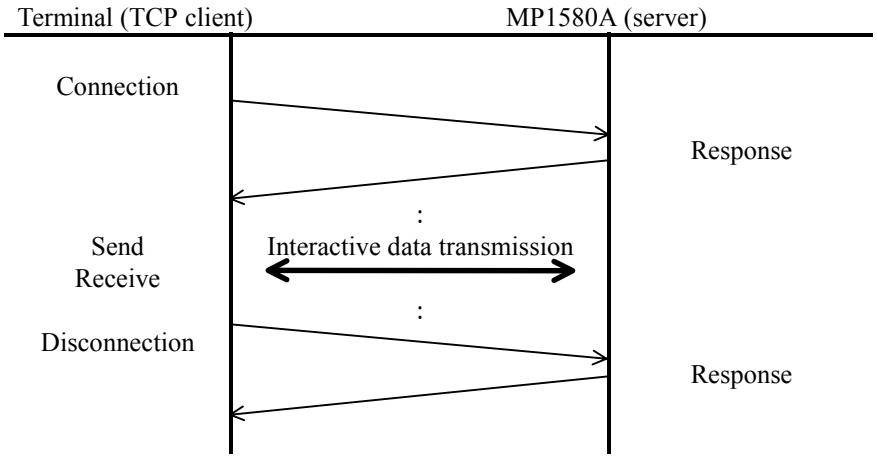
When MP1580A is turned on, the following statuses are restored:

- The status from when power was turned off is set.
- The input buffer and output queue are cleared.
- The syntax analysis, execution control and response creation portions are reset.

4.8 Network Connection and Data Flow

Data communication between MP1580A and clients uses TCP connection. The communication program (socket client) must be created at the terminal-side. For details on the IP packet used in data communication, refer to any commercially available reference materials.

For more information on the socket interface used for communication, refer to the manuals provided with the terminal, installed network board and the driver software as well as commercially available reference materials.

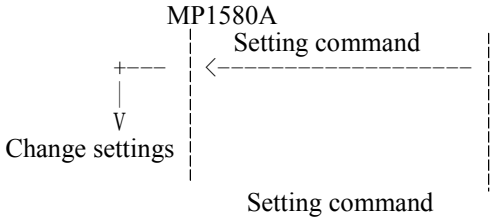


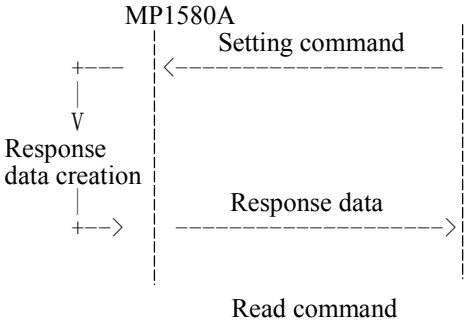
4.9 Client Connection

4.9.1 Control command

The control command for MP1580A is a character string command complying with IEEE488.2 and SCPI standards. (The same as for the RS-232C and GPIB options.)

Control commands include setting and read commands. A setting command changes MP1580A settings while a read command reads MP1580A setting status and measured results. When receiving a read command, MP1580A prepares response data corresponding to the command to automatically send it back to the sender (a PC, etc.).

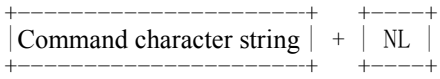




4.9.2 Terminator

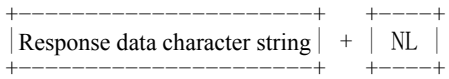
A terminator is specified to indicate the end of a command or response data character string.

Terminator for a command
The terminator is LF (0x0A). (When the character string contains CD(0x0D), CR is excluded.)



NL (New Line) … Terminator

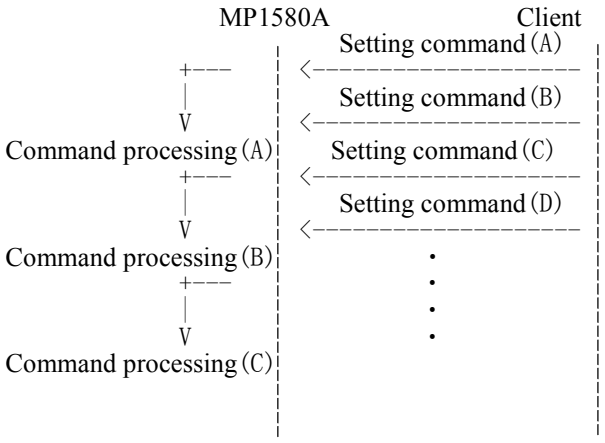
Terminator for a response data
The terminator is LF(0x0A) or CR+LF(0x0D+0x0A). (Can be switched between the two. LF at power On.)



NL (New Line) … Terminator

4.9.3 Executing action

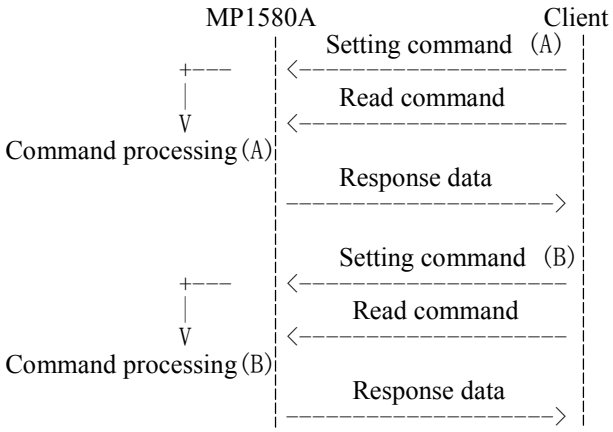
MP1580A sequentially executes commands. That is, the received commands are executed one by one in the order received. Ack on TCP is immediately returned at command reception so that the clients can transmit commands continuously regardless of the execution status of MP1580A. Though MP1580A has an internal buffer, some part of command character strings may be lost when the number of received commands exceeds its processing speed. It is also possible that a significant time difference in processing status occurs between MP1580A and the client application. The synchronization procedures shown below are recommended for the client-side:



No synchronization procedure

synchronization procedure

Synchronize the processing status by transmitting a setting command and then a read command to confirm the response data at the client-side.



Synchronization procedure

4.9.4 Connection/disconnection

Establish connection by socket communication. Perform disconnection procedures at the client-side. When no command is received from the client-side for a specified time (approximately 10 minutes), the server disconnects the connection.

4.10 Notes on Usage

4.10.1 Ethernet option

The Ethernet option does not work when Option 03 Ethernet is not mounted on MP1580A. If it is not mounted, the Remote Local LED does not turn on when a remote command is transmitted.

4.10.2 Transmission speed and method

Generally, the transmission speed between units is connected at maximum speed as the auto negotiation function is used. However, when the Hub or unit setting is fixed, the desired transmission method may not meet expectations.

4.10.3 TCP/IP network

MP1580A supports TCP only. UDP is not supported.

Use TCP when connecting to the network.

4.10.4 Disconnecting the connection

Normally make the disconnection at the client-side. MP11580A disconnects the connection when no command is received from the client-side for a specified time (approximately 10 minutes).

4.10.5 When the IP address is changed

When the IP address, Subnet mask or Gateway address is changed, the Reboot button appears on the same screen (lower right part of the Setup: System screen). Press the Reboot button or turn MP1580A off and then on to enable the new setting. It is not enabled by simply changing the parameters.

4.10.6 Supporting remote commands only

MP1580A Ethernet only supports remote commands. Any other commands are nvalid.

4.10.7 Terminator for remote command

When transmitting a remote command to MP1580A, LF(0x0A) or CR+LF(0x0D+0x0A) must be added as the terminator.

Response data from MP1580A always have a terminator. The terminator is LF(0x0A) or CR+LF(0x0D+0x0A). LF(0x0A) is the initial value. To change the terminator to LF(0x0A) or CR+LF(0x0D+0x0A), use the following command:

To set the terminator to CR+LF:

>:SYSTem:TERMination 1

To set the terminator to LF, set the parameter to 0.

For details on this command → Refer to Section 11.5.1
For details on the terminator → Refer to <PROGRAM MESSAGE TERMINATOR> in Section 5.2.2 or <RESPONSE MESSAGE TERMINATOR> in Section 6.2.2.

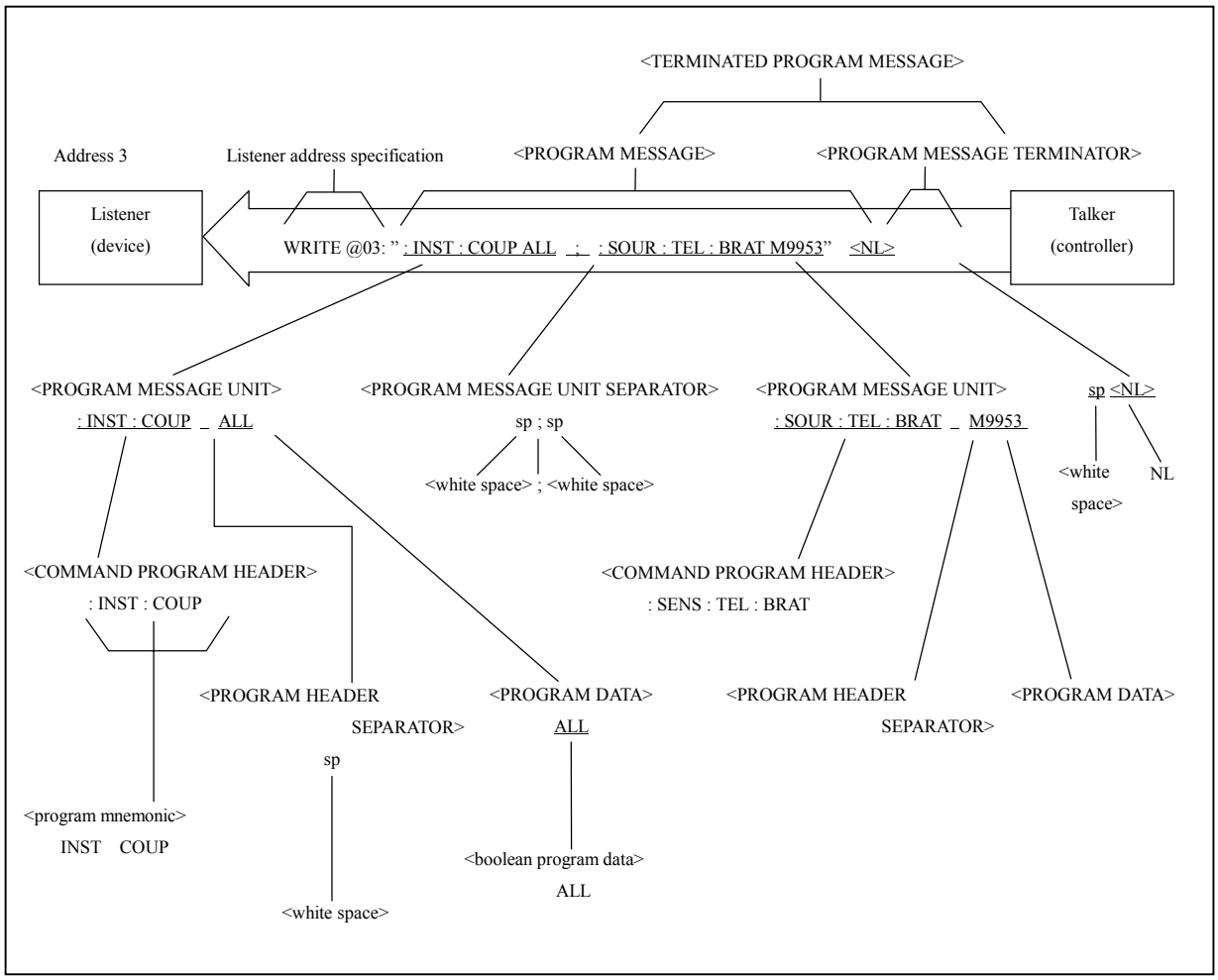
Section 5 Listener Input Format

This section describes the format of program messages received by the listener (MP1580A) from the talker (controller).

Since the equipment unique commands conform to the SCPI standards, examples are given using the SCPI commands.

5.1 Listener Input Program Message Format

As an example, the program message is shown below when transmission and reception modes for the jitter unit is set to the same mode, and the bit rate is set to 9953M.



The program message format is composed of the functional element sequence, being broken down to the minimum level unit capable of expressing the function. Functional element examples are shown by upper case characters in parentheses < > in the figure above.

Listener address setting

→ *next page*

Program messages

→ *5.2.1 <TERMINATED PROGRAM MESSAGE>*

Separator (separation/connection)

→ *5.2.3 <white space>*

→ *5.2.5 <PROGRAM MESSAGE UNIT SEPARATOR>*

→ *5.2.10 <PROTRAM HEADER SEPARATOR>*

→ *5.2.11 <PROGRAM DATA SEPARATOR>*

Terminator indicating end of the command

→ *5.2.2 <PROGRAM MESSAGE TERMINATOR>*

Formats for WRITE and READ commands are as shown below.

WRITE @

Outputs data to a device.

* Format

WRITE @ Device No.: Data

Data→	Numerical expression Character string expression	

<Example> Same as that on the previous page.

WRITE @03: ” : INST : COUP ALL ; : SENS : TEL : BRAT M9953”

Listener address (when GPIB address of MP1580A is set to 3)

READ @

Substitutes the data inputted from device into variable.

* Format

READ @ Device No.: Variable

<Example> Substitute the data (input connector setting) inputted from device into variable A$.

WRITE @03: ” : INST : COUP?” ←— Queries the send/receive setting.

READ @03: A$

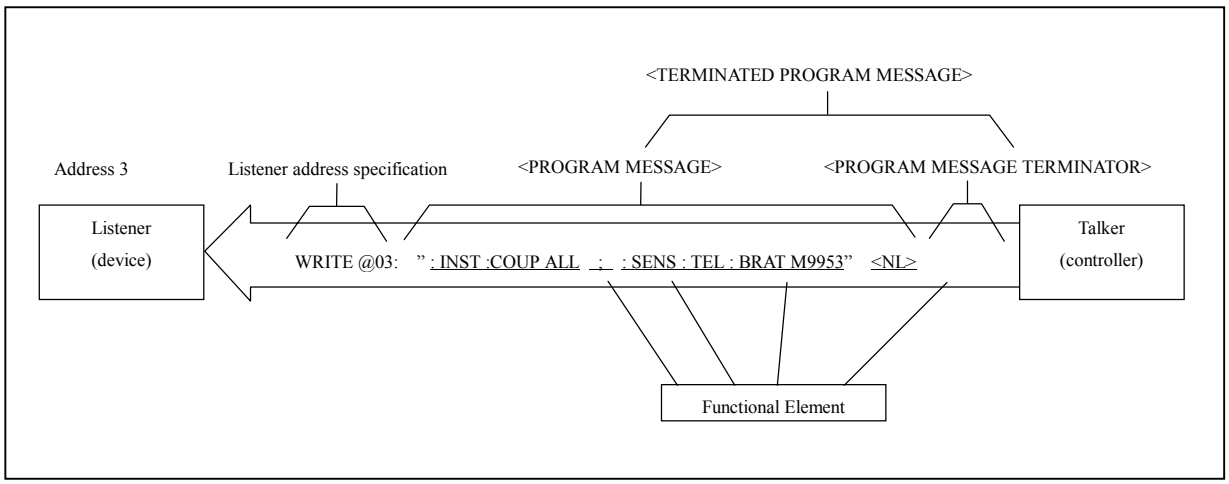
Listener address (when GPIB address of MP1580A is set to 3)

5.2 Functional Element of Program Message

MP1580A receives a program message when it detects a terminator located at the end of a program message. Functional elements of program messages are explained below.

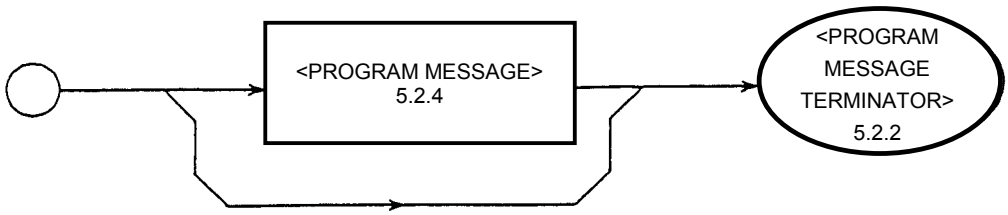
5.2.1 <TERMINATED PROGRAM MESSAGE>

A <TERMINATED PROGRAM MESSAGE> containing two commands is shown below.



Definition

<TERMINATED PROGRAM MESSAGE> is defined as follows.



<TERMINATED PROGRAM MESSAGE> is a data message containing all functional elements needed for transmission from the controller to the device (MP1580A.)

For completing a <PROGRAM MESSAGE> transmission, a <PROGRAM MESSAGE TERMINATOR> is added at the and of a <PROGRAM MESSAGE>.

<PROGRAM MESSAGE>

→ 5.2.4 <PROGRAM MESSAGE>

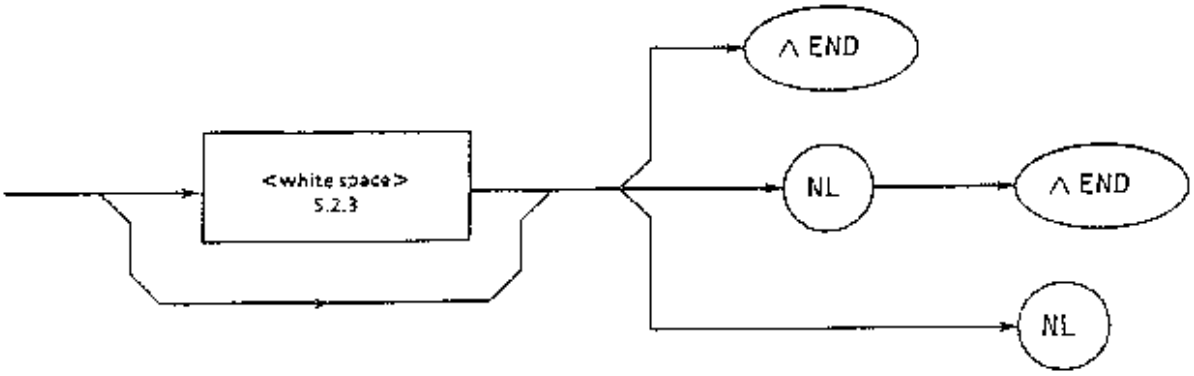
<PROGRAM MESSAGE TERMINATOR> → next page

5.2.2 <PROGRAM MESSAGE TERMINATOR>

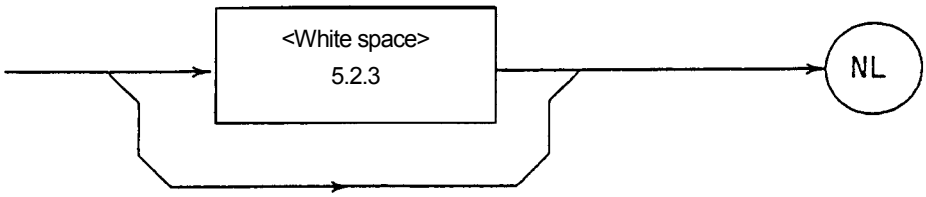
<PROGRAM MESSAGE TERMINATOR> terminates the sequence with a length of one or more <PROGRAM MESSAGE UNIT> elements.

Definition of <PROGRAM MESSAGE TERMINATOR> varies depending on the interface.

(1) When the GPIB interface is used



(2) When the RS-232C or Ethernet interface is used



NL..... Defined as a single ASCII code byte 0A (10 in decimal notation.) That is, it represents LF (line feed) of ASCII control character which carries out the carriage return, returning the printing position at the same character position of the next line. It is called NL (new line) as printing of a new line is started.

END..... An EOI signal can be generated by setting the EOI (end-or-identify) line, one of GPIB control buses to TRUE (LOW level). The EOI ON/OFF statement is one of the control statements for EOI lines.

Note:

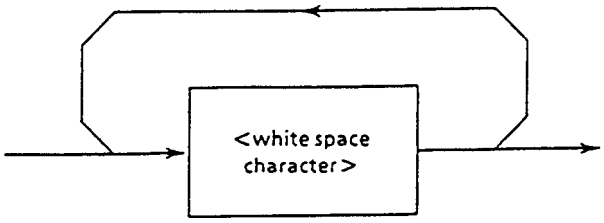
In addition to LF code, CR+LF can be used for NL (new line).

When the controller system is MS-DOS or Windows, CR+LF is used for to begin a new line. For UNIX, LF is used.

<white space> in front of NL, → next page

5.2.3 <white space>

<white space> is defined as follows.

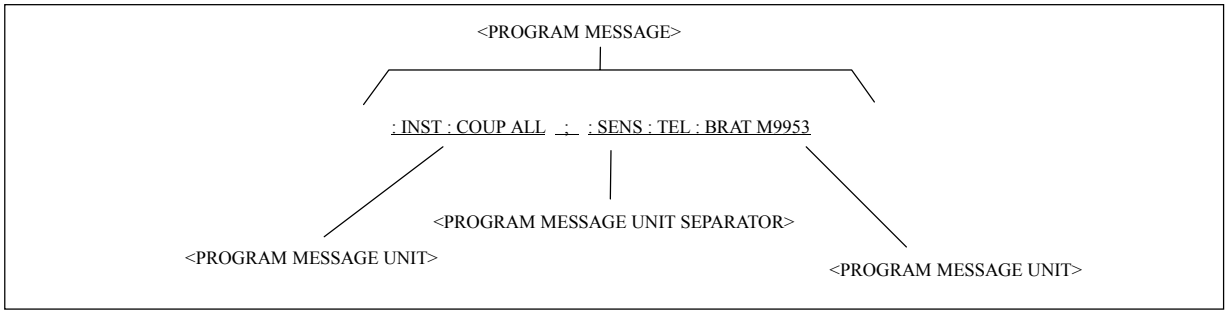


A <white space character> is defined as a single ASCII code type within a range of ASCII code bytes 00 to 09 or 0B to 20 (0 to 9 or 11 to 32 in decimal notation.)

Although the range includes the ASCII control code and space signal, except the new line; MP1580A handles it merely as a space or ignores without interpreting it as an ASCII control code.

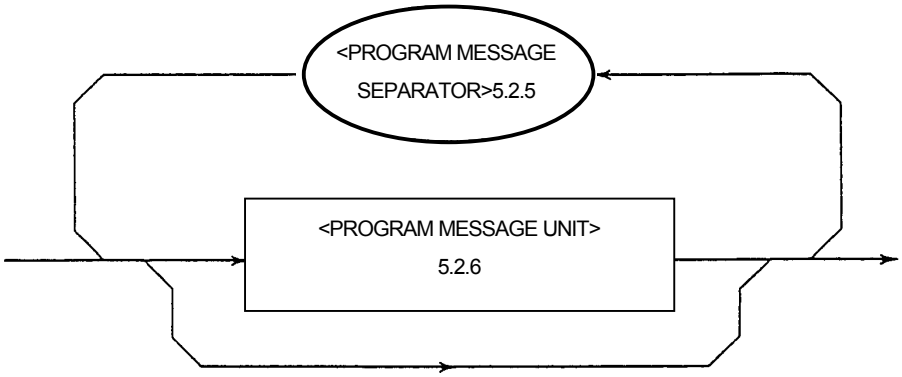
5.2.4 <PROGRAM MESSAGE>

As an example, transmission and reception modes are set to the same mode and the bit rate is set to 9953M.



Definition

<PROGRAM MESSAGE> is defined as follows.



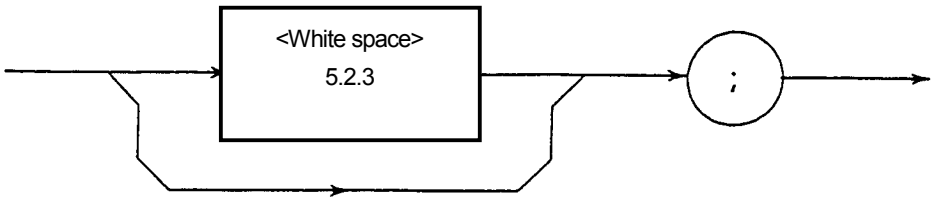
<PROGRAM MESSAGE> is a sequence of 0, 1 or more <PROGRAM MESSAGE UNIT> elements.
<PROGRAM MESSAGE UNIT> element is a programming command or data being transmitted from the controller to the device.
The <PROGRAM MESSAGE UNIT SEPARATOR> is used to separate multiple <PROGRAM MESSAGE>.

<PROGRAM MESSAGE UNIT SEPARATOR>
→ 5.2.6 <PROGRAM MESSAGE UNIT>

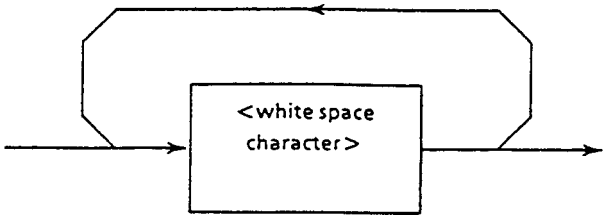
<PROGRAM MESSAGE UNIT>
→ 5.2.5 <PROGRAM MESSAGE UNIT SEPARATOR>

5.2.5 <PROGRAM MESSAGE UNIT SEPARATOR>

<PROGRAM MESSAGE UNIT SEPARATOR> is defined as follows.



<white space> is defined as follows.

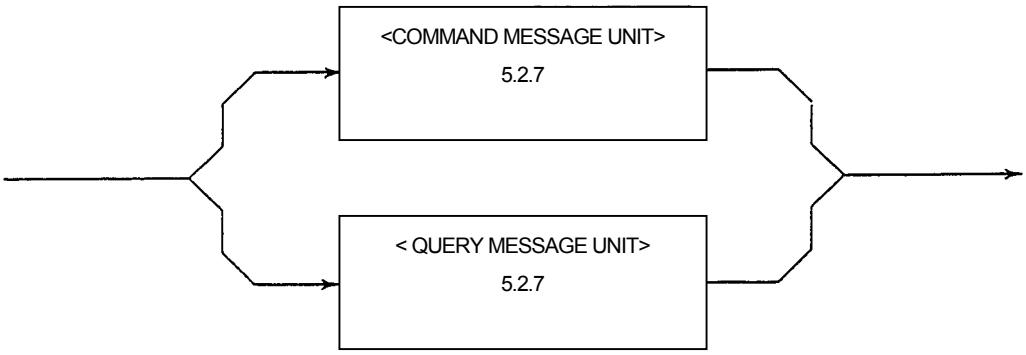


<PROGRAM MESSAGE UNIT SEPARATOR> divides multiple <PROGRAM MESSAGE UNIT> element sequences within the <PROGRAM MESSAGE> range.

MP1580A interprets a semicolon ";" as the <PROGRAM MESSAGE UNIT> separator. Therefore, <white space character> before and after a semicolon ";" are ignored. The <white space character> is effective to make a program easy to read.

5.2.6 <PROGRAM MESSAGE UNIT>

<PROGRAM MESSAGE UNIT> is defined as follows.



<PROGRAM MESSAGE UNIT> consists of a single command message, <COMMAND MESSAGE UNIT>, or a single query message, <QUERY MESSAGE UNIT>, received by MP1580A.

Setting and query can be made by each <PROGRAM MESSAGE UNIT>.

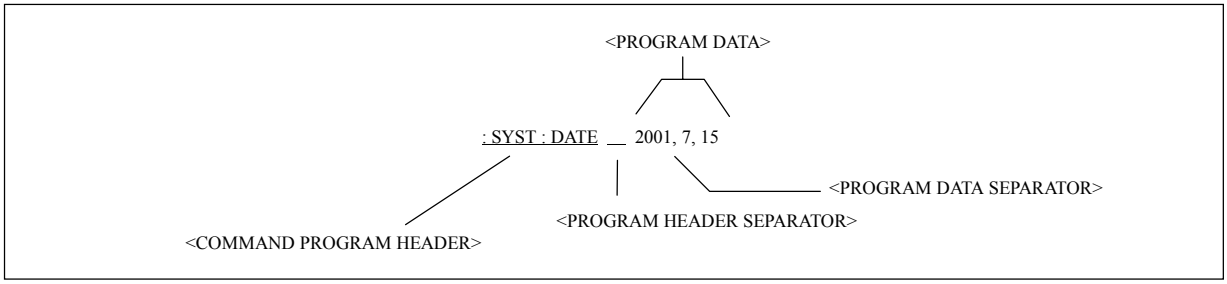
<COMMAND MESSAGE UNIT> / <QUERY MESSAGE UNIT>

→ *5.2.7 <COMMAND MESSAGE UNIT> / <QUERY MESSAGE UNIT> on the next page*

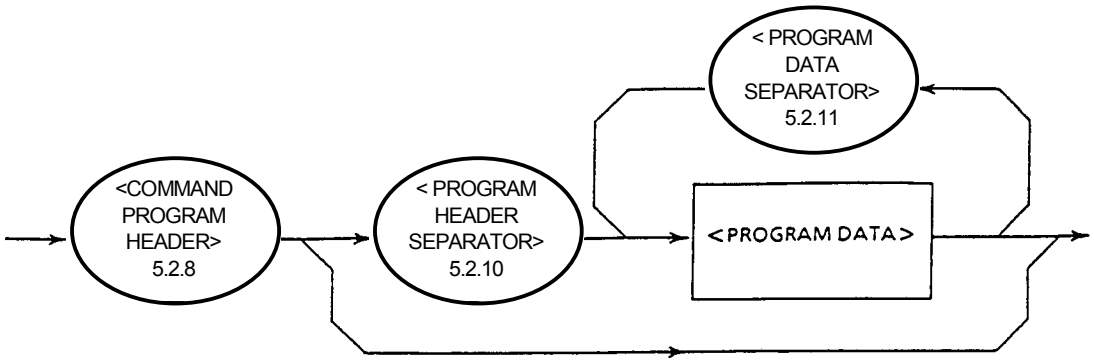
5.2.7 <COMMAND MESSAGE UNIT> / <QUERY MESSAGE UNIT>

(1) <COMMAND MESSAGE UNIT>

Example: Setting the date.

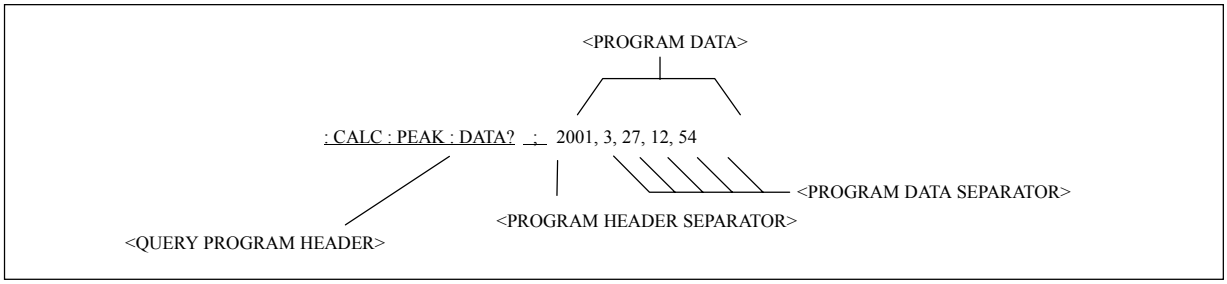


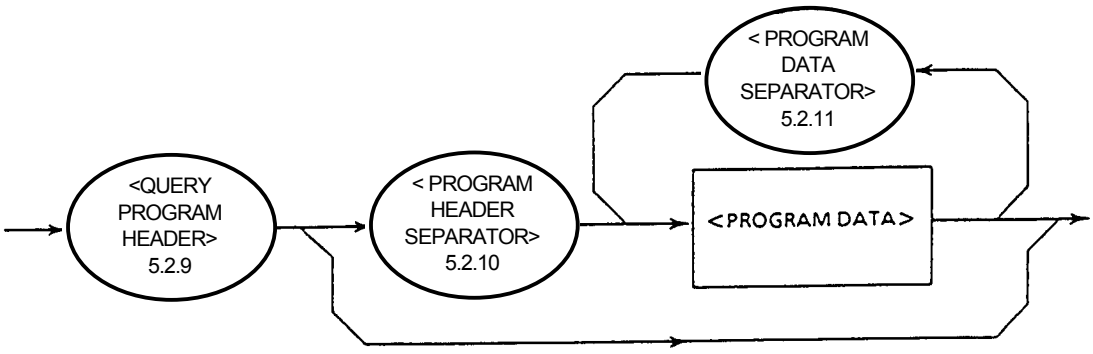
<COMMAND MESSAGE UNIT> is defined as follows.



(2) <QUERY MESSAGE UNIT>

Example: Querying Error/Alarm analyze data.





<QUERY MESSAGE UNIT> is defined as follows.

For the <COMMAND MESSAGE UNIT> or <QUERY MESSAGE UNIT>, a space is always inserted as a separator after the program header when program data follows after a program header. The program header identifies the application, function or operation of the program. When no program data are attached, a header identifies the application, function or operation to be executed inside MP1580A.

The <COMMAND PROGRAM HEADER> among program headers is used as the command for controller to control MP1580A. The <QUERY PROGRAM HEADER> is a query command; the controller sends it to MP1580A beforehand in order to receive a response message from MP1580A. The feature of this header is that the query indicator "?" is always attached at the end of the header.

<COMMAND PROGRAM HEADER>

→ *5.2.8 <COMMAND PROGRAM HEADER> on the next page*

<QUERY PROGRAM HEADER>

→ *5.2.9 <QUERY PROGRAM HEADER>*

<PROGRAM HEADER SEPARATOR>

→ *5.2.10 <PROGRAM HEADER SEPARATOR>*

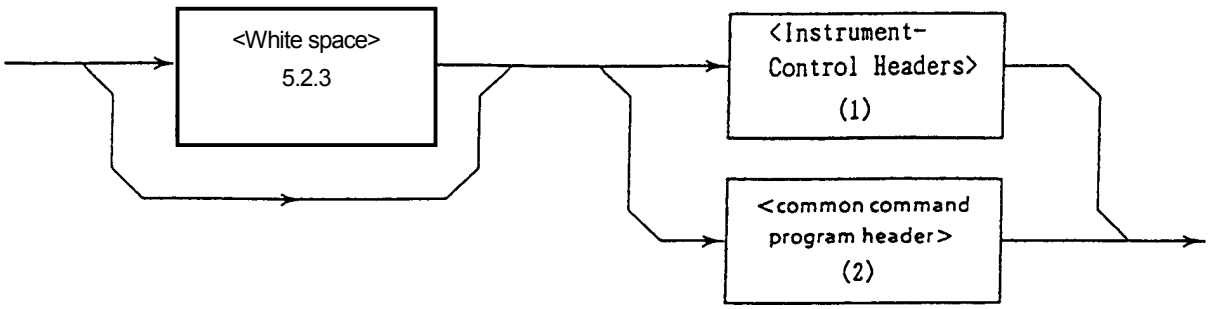
<PROGRAM DATA SEPARATOR>

→ *5.2.11 <PROGRAM DATA SEPARATOR>*

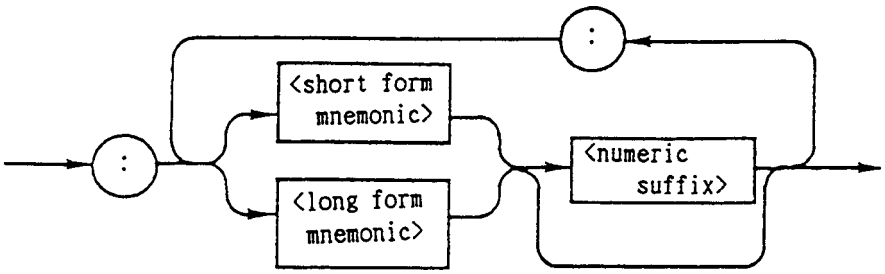
5.2.8 <COMMAND PROGRAM HEADER>

<COMMAND PROGRAM HEADER> is defined as follows.

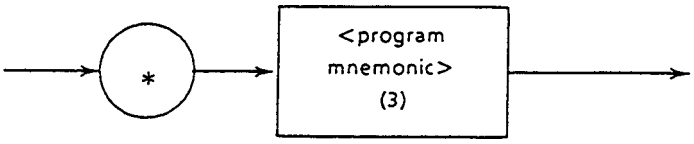
A <white space> can be located before each header.



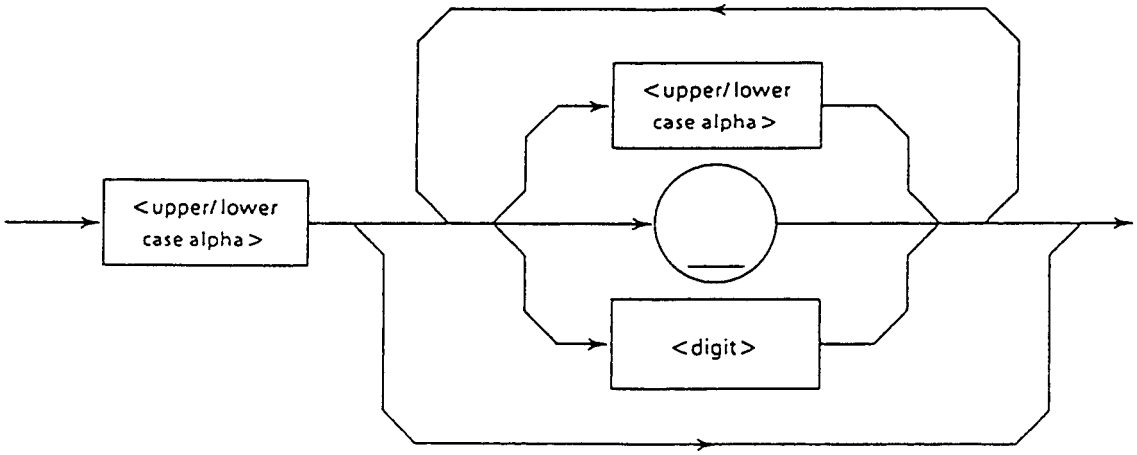
(1) <Instrument-Control Headers> is defined as follows.



(2) <common command program header> is defined as follows.



(3) <program mnemonic> is defined as follows.



* <COMMAND PROGRAM HEADER>
  This header identifies the application, function or operation of program data to be executed by MP1580A. When no program data are attached, the header alone identifies an application, function or operation to be executed inside MP1580A.
  The <program mnemonic> indicates contents of headers in the ASCII code characters which is generally called mnemonic.

* <program mnemonic>
  The mnemonic is always begun by a upper/lower alphabet followed by optional combinations of upper case "A" to "Z", lower case "a" to "z", under bar "_" and numerals "0" to "9." The maximum length of mnemonic is 12 characters and no spaces are inserted between characters.
  - <upper/lower case alpha>
    Defined as a single ASCII code byte within the range of ASCII code bytes 41 to 5A or 61 to 7A (65 to 90 or 97 to 122 in decimal notation = upper case alphabetical characters A to Z) or lower case alphabetical characters a to z.)
  - <digit>
    Defined as a single ASCII code byte within the range of ASCII code bytes 30 to 39 (48 to 57 in decimal notation = numerals 0 to 9.)
  - (_)
    Defined as a single ASCII code byte, representing ASCII code byte 5F (95 in decimal notation = underline.)
    Example: SYSTem (succeeding part is omitted)

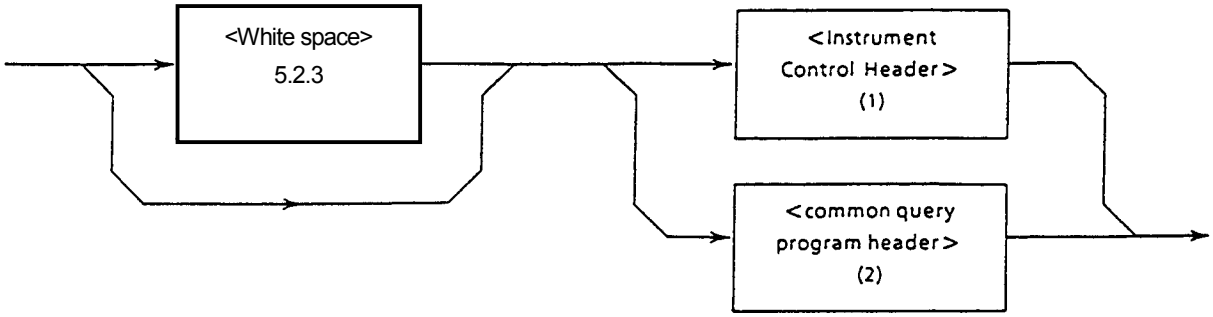
* <Instrument-Control Headers>
  The <Instrument-Control Headers> is specified in the SCPI standard. Since the equipment unique commands of MP1580A conform to the SCPI standard, the command formats also confirm to the standard. For details, see Section 9 "SCPI Overview."
  - <short form mnemonic>/<long form mnemonic>
    These correspond to the short form and long form of SCPI commands. As for contents of the mnemonic, those explained in <program mnemonic> apply as they are.
  - <numeric suffix>
    Defined as a single ASCII code byte within the range of ASCII code bytes 30 to 39 (48 to 57 in decimal notation = numerals 0 to 9.)

* <common command program header>
  As for the <common command program header>, an asterisk "*" is always attached before a <program mnemonic>. (For details, see Section 7 "IEEE488.2 Common Command.")

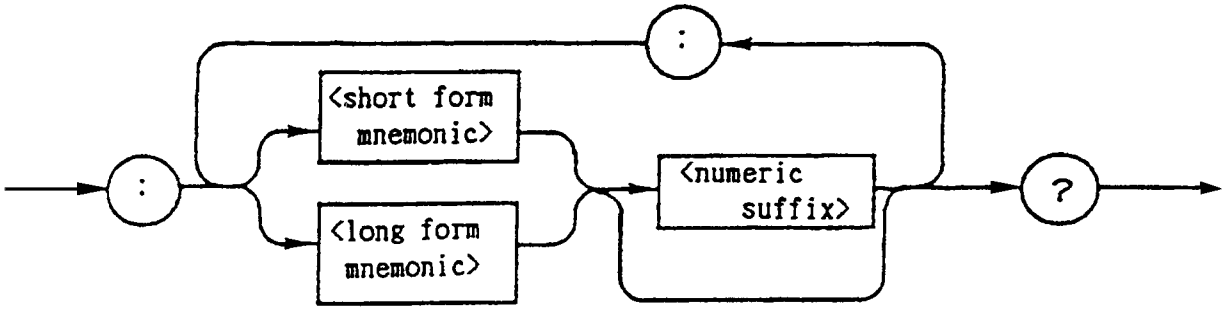
5.2.9 <QUERY PROGRAM HEADER>

<QUERY PROGRAM HEADER> is defined as follows.

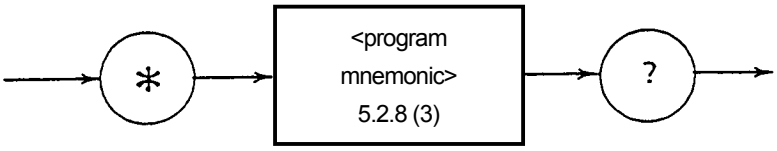
A <white space> can be located before each header.



(1) <Instrument-Control Headers> is defined as follows.



(2) <common query program header> is defined as follows.



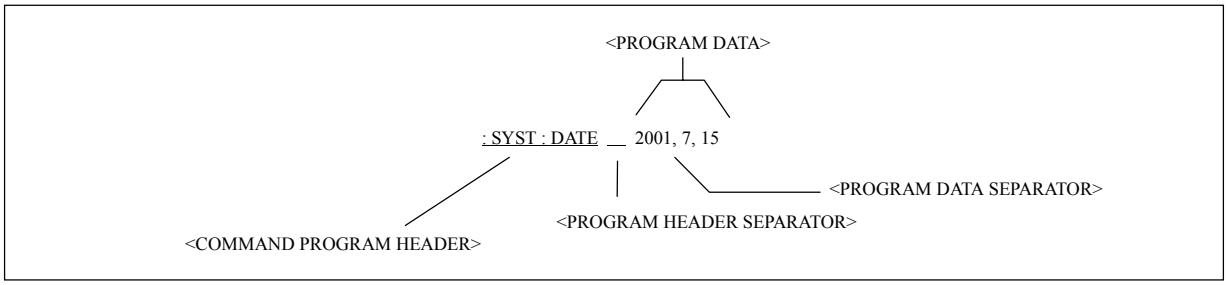
* <QUERY PROGRAM HEADER>

The <QUERY PROGRAM HEADER> is a query command; the controller sends it to MP1580A beforehand in order to receive a response message from MP1580A. The feature of this header is that the query indicator "?" is always attached at the end of the header.

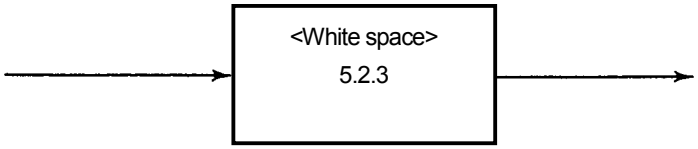
[ex.] :SYSTem:DATE?

The <QUERY PROGRAM HEADER> format explained above is identical to that of <COMMAND PROGRAM HEADER>, except a query indicator "?" is attached at the end of header. See 5.2.8 "<COMMAND PROGRAM HEADER>" for details.

5.2.10 <PROGRAM HEADER SEPARATOR>



<PROGRAM HEADER SEPARATOR> is defined as follows.



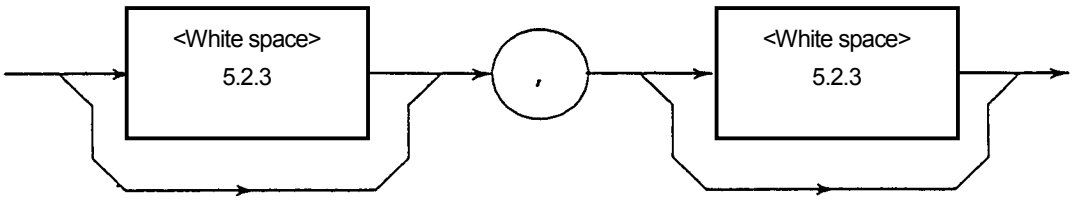
The <PROGRAM HEADER SEPARATOR> is used as the separator between a <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> and <PROGRAM DATA>.

When two or more <white space character> exist between a program header and program data, the first <white space character> is interpreted as the separator and the rest are ignored. The <white space character>, however, are effective to make a program easy to read. One header separator always exists between a header and data indicating the end of a program as well as beginning of a program.

<PROGRAM DATA SEPARATOR> in the figure above
→ 5.2.11 <PROGRAM DATA SEPARATOR>

5.2.11 <PROGRAM DATA SEPARATOR>

<PROGRAM DATA SEPARATOR> is defined as follows.

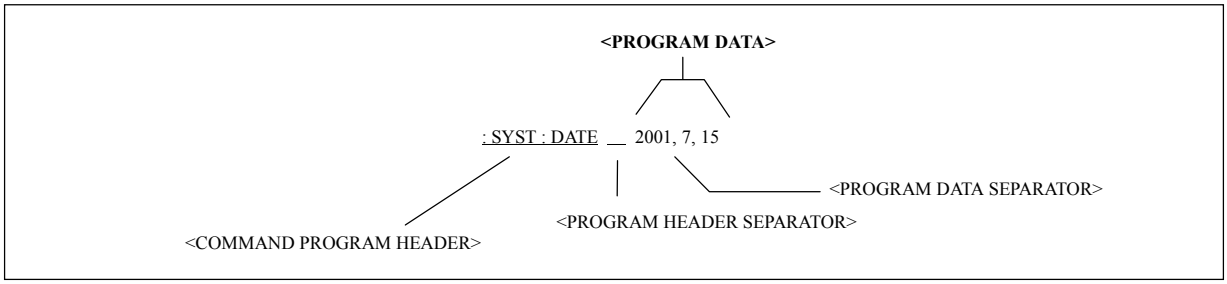


The <PROGRAM DATA SEPARATOR> is used to separate parameters when a <COMMAND PROGRAM HEADER> or <QUERY PROGRAM HEADER> has many parameters.

A comma is always needed when the data separated used, however, the <white space character> is not necessarily needed. The <white space character> before or after a comma is ignored. The <white space character>, however, are effective to make a program easy to read.

5.3 Program Data Format

Among the format system of program messages being terminated as explained before, the <PROGRAM DATA> format shown by the function grammar diagram in Section 5.2 is further explained here.



The <PROGRAM DATA> function elements are used for transmitting various parameters relating to the program header. Program data used by MP1580A are listed below.

<PROGRAM DATA>	Type
＜CHARACTER PROGRAM DATA＞	Short data consists of alphanumeric characters
＜DECIMAL NUMERIC PROGRAM DATA＞	Numeric constant in decimal notation
＜BOOLEAN PROGRAM DATA＞	Logical value (SCPI specification)
＜STRING PROGRAM DATA＞	Character string enclosed in " (double-quotation) or ' (single-quotation) marks.

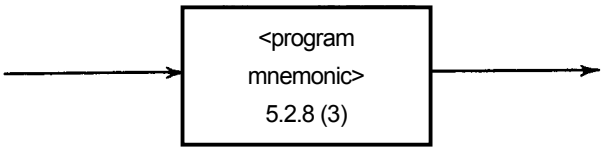
5.3.1 <CHARACTER PROGRAM DATA>

<CHARACTER PROGRAM DATA> represents a short data consists of alphanumeric characters.

< Example> : SENSe : TELecom : PATTern : TYPE PRBS11 (mnemonic data representing PRBS2^{11}-1)

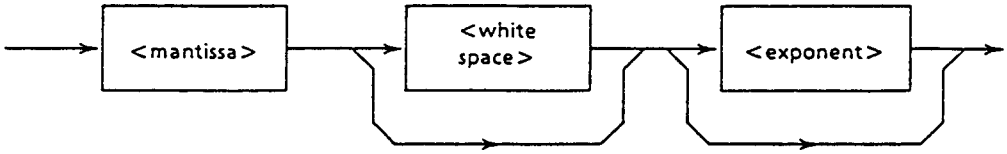
: SENSe : MEASure : TYPE SINGle (mnemonic data representing a single measurement)

The contents of the character data is the same as described in "5.2.8 (3) program mnemonic."

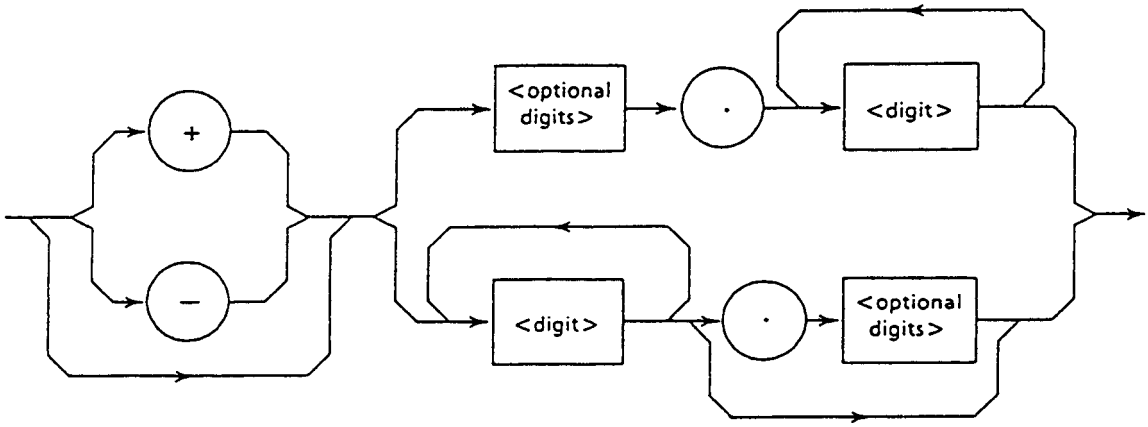


5.3.2 <DECIMAL NUMERIC PROGRAM DATA>

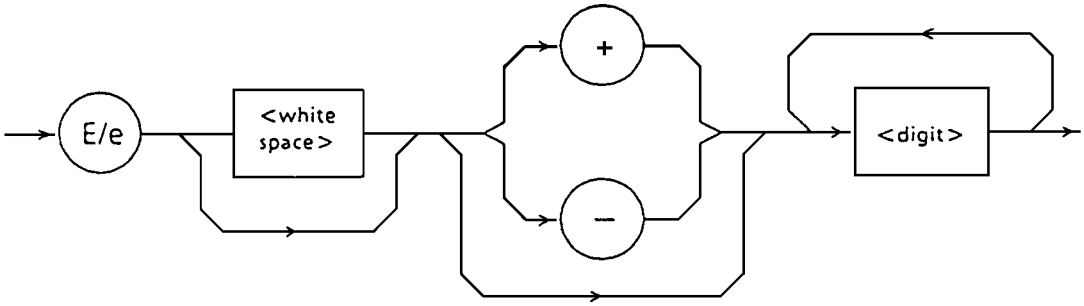
<DECIMAL NUMERIC PROGRAM DATA> represents a decimal value and is defined as follows.



<maintissa> is defined as follows.



<exponent> is defined as follows.



Values used by MP1580A are integers in decimal notation.

* Integer format

Represents an integer in decimal notation.

- "0" can be inserted at the beginning. → 005
- Space is not allowed between symbols and numerals. → +5 (○), +△5 (×)
- Spaces can be inserted after a numeral. → +5△△
- The "+" symbol may or may not be attached. → +5, 5
- No commas can be used to divide digits. → 1, 235 (×)

<Example> : SOURce : MEASure : PERiod 1 , M

5.3.3 <BOOLEAN PROGRAM DATA>

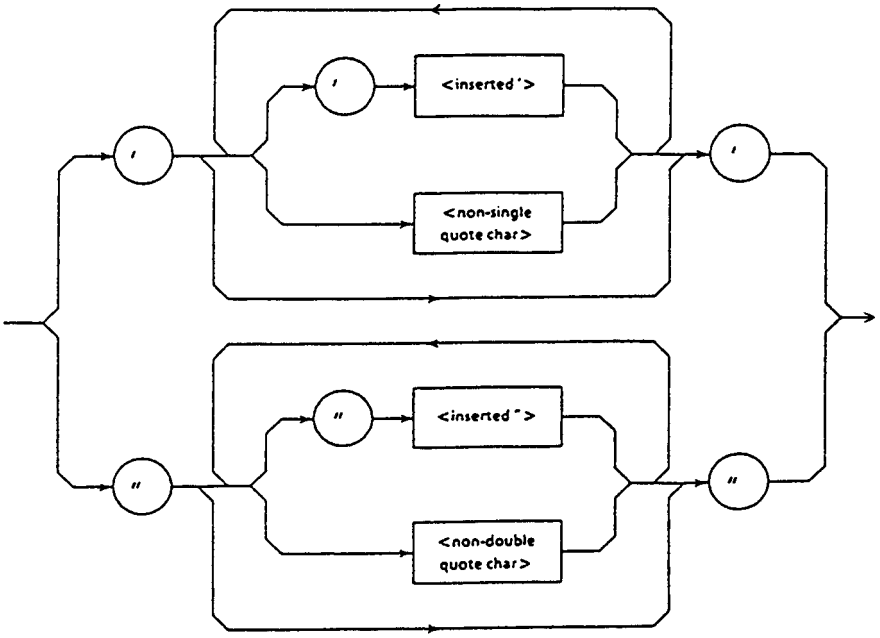
<BOOLEAN PROGRAM DATA> is program data which is specified by the SCPI and represents a logical value. As the value corresponding to true and false, ON and OFF of <CHARACTER PROGRAM DATA>, and 1 and 0 of <DECIMAL NUMERIC PROGRAM DATA> are defined.

<Example> : SYSTem : PRINt : JITT : SET ON
: SYSTem : PRINt : JITT : SET 1

5.3.4 <STRING PROGRAM DATA>

<STRING PROGRAM DATA> represents a character string. It is enclosed in " (double-quotation) or ' (single-quotation) marks.
When the character string contains the " (double-quotation) or ' (single-quotation) mark, the same mark should be used for the both ends of a character string.

It is defined as follows.



(1) <inserted> is defined by a single ASCII symbol of value 27 (decimal, 39 = ')

(2) <non-single quote char> is defined by a single ASCII symbol of value other than 27 (decimal, 39 = ')

(3) <inserted"> is defined by a single ASCII symbol of value 22 (decimal, 34 = ')

(4) <non-double quote char> is defined by a single ASCII symbol of value other than 22 (decimal, 34 = ')

<Description example>

Correct description examples are shown below.

When enclosed in ' (single quotation) marks:

'calculate '

'remote"control' (" is interpreted as a character)

'"Jan.""Feb."' (" is interpreted as a character)

When enclosed in " (double quotation) marks:

" calculate "

"It's a nice day." (' is interpreted as a character)

" 'program' 'data' " (' is interpreted as a character)

As shown above, the same marks should be used for the both ends of a character string.

Example in an actual command is shown below.

<Example> : SENS : TEL : PATT : UWOR "100110011001100"

(Character string expressing word data 1100110011001100)

: DISP : DSEL : NAME "SETup" (Character string expressing Setup main screen.)

: CALC : DATA? "JHIT:COUNt"

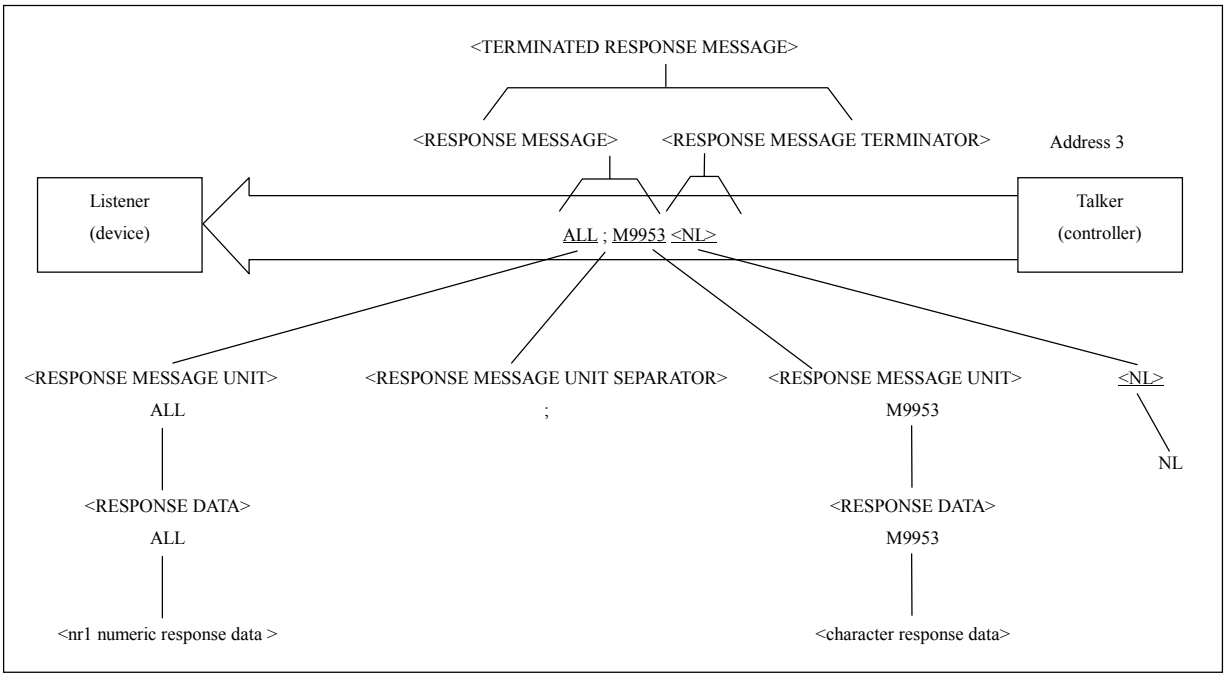
(Character string expressing hit count of manual jitter measurement.)

Section 6 Talker Output Format

This section describes formats of response messages returned from the talker (device, MP1580A) to the listener (controller).

6.1 Talker Output Response Message Format

Responses against the transmission/reception mode query :INST:COUP? and the receive signal bit rate query :SENS:TEL:BRAT? are shown below.
Since no headers are attached to the SCPI response, only data is returned as response.



The response message format consists of the function element sequence broken down to the minimum level unit that can express a function, as in the case of the program message. Upper case characters enclosed in parentheses < > are example functional elements in the figure above. A functional element can further be divided, which is called a coded element. Lower case characters enclosed in parentheses < > are example coded elements. Consequently, the grammar expression method is the same for the talker and listener.

Response messages

→ *6.2.1 <TERMINATED RESPONSE MESSAGE>*

Separator

→ *6.2.4 <RESPONSE MESSAGE UNIT SEPARATOR>*

→ *6.2.6 <RESPONSE DATA SEPARATOR>*

Terminator indicating end of the response

→ *6.2.2 <RESPONSE MESSAGE TERMINATOR>*

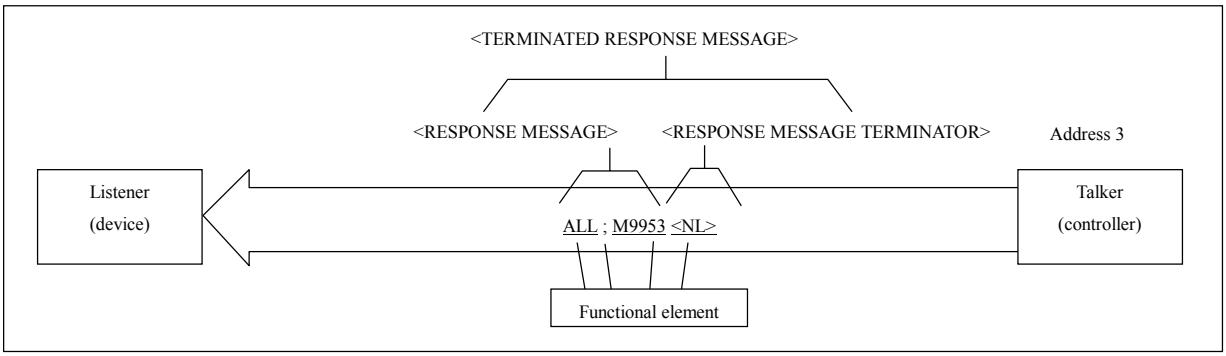
Response data

→ *6.2.7 <RESPONSE DATA>*

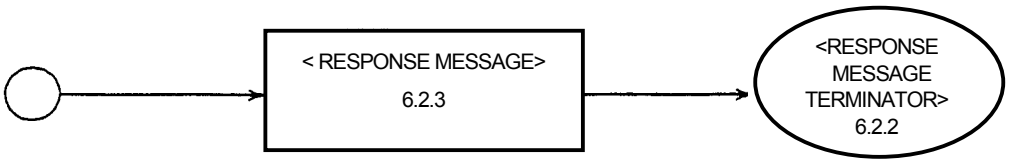
6.2 Functional Element of Response Message

6.2.1 <TERMINATED RESONSE MESSAGE>

A <TERMINATED RESONSE MESSAGE> linking two message units is shown below.



Definition
<TERMINATED RESONSE MESSAGE> is defined as follows.



<TERMINATED RESONSE MESSAGE> is a data message containing all function elements necessary for transmitting the message from the talker device to the controller.

For completing a <RESPONSE MESSAGE> transmission, a <RESPONSE MESSAGE TERMINATOR> is added at the end of a <RESPONSE MESSAGE>.

<RESPONSE MESSAGE>

→ 6.2.3 <RESPONSE MESSAGE>

NL, END

→ 5.2.2 <PROGRAM MESSAGE TERMINATOR>

<RESPONSE MESSAGE TERMINATOR>

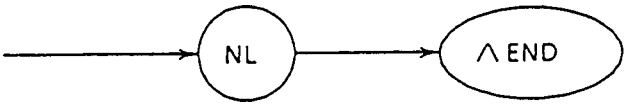
→ 6.2.2 <RESPONSE MESSAGE TERMINATOR>

6.2.2 <RESPONSE MESSAGE TERMINATOR>

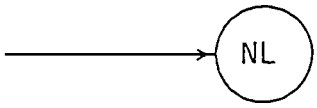
<RESPONSE MESSAGE TERMINATOR> is located after the <RESPONSE MESSAGE UNIT> and terminates one or more <RESPONSE MESSAGE UNIT> element sequences with constant length.

Definition of <RESPONSE MESSAGE TERMINATOR> varies depending on the interface, as follows.

(1) When the GPIB interface is used



(2) When the RS-232C or Ethernet interface is used



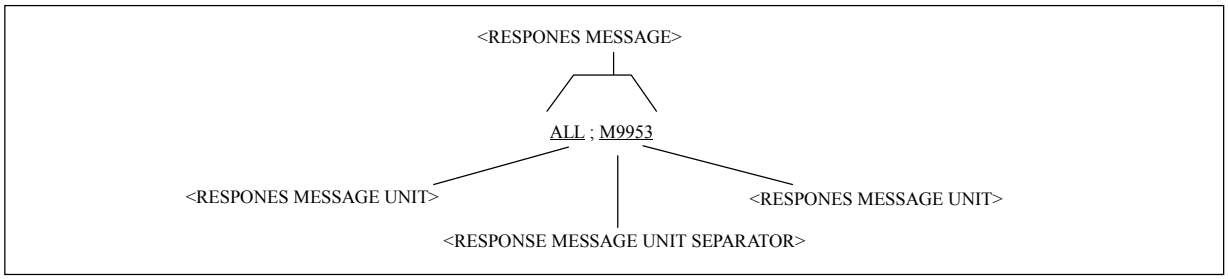
<Example> A program which reads the transmission/reception mode status.

```
10 WRITE @03: ":INST:COUP?"
20 READ @03:A$
30 PRINT A$
40 END
```

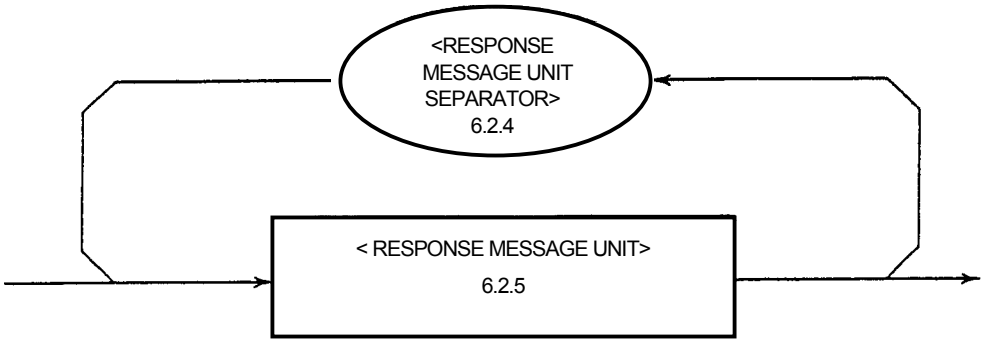
NL, END
→ 5.2.2

6.2.3 <RESPONSE MESSAGE>

An example response against input connector selection and receive signal bit rate query is shown below.



Definition
<RESPONSE MESSAGE> is defined as follows.



A <RESPONSE MESSAGE> is one or more <RESPONSE MESSAGE UNIT> element sequences.

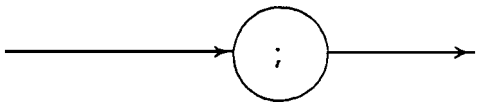
A <RESPONSE MESSAGE UNIT> element expresses a single message being sent from MP1580A to the controller. The <RESPONSE MESSAGE UNIT SEPARATOR> element is used as a separator which separates multiple <RESPONSE MESSAGE UNIT>.

<RESPONSE MESSAGE UNIT>
→ 6.2.5 <RESPONSE MESSAGE UNIT>

<RESPONSE MESSAGE UNIT SEPARATOR>
→ 6.2.4 <RESPONSE MESSAGE UNIT SEPARATOR>

6.2.4 <RESPONSE MESSAGE UNIT SEPARATOR>

<RESPONSE MESSAGE UNIT SEPARATOR> is defined as follows.

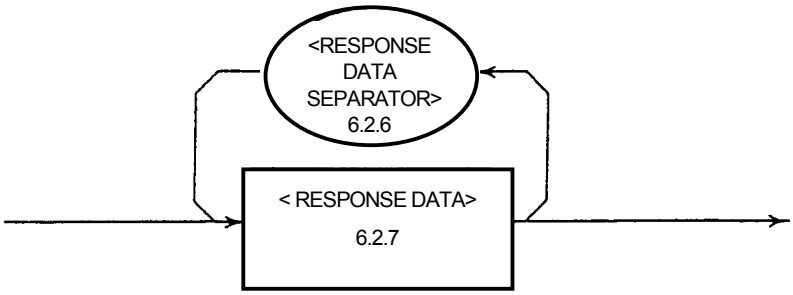


The <RESPONSE MESSAGE UNIT SEPARATOR> separates elements using the <UNIT SEPARATOR> semicolon ";" when multiple <RESPONSE MESSAGE UNIT> element sequences are output as a single <RESPONSE MESSAGE>.

6.2.5 <RESPONSE MESSAGE UNIT>

<RESPONSE MESSAGE UNIT> used in MP1580A is the response message unit without header, and only measured result data are responded.

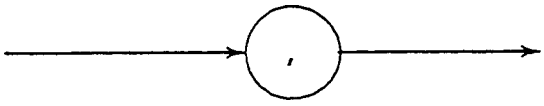
<RESPONSE MESSAGE UNIT> is defined as follows.



6.2.6 <RESPONSE DATA SEPARATOR>

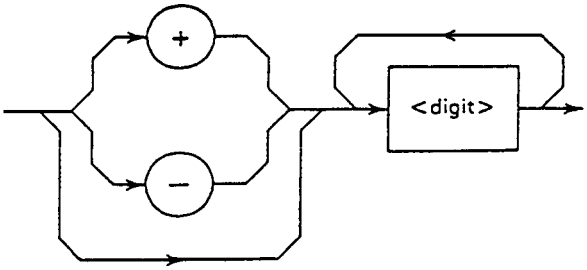
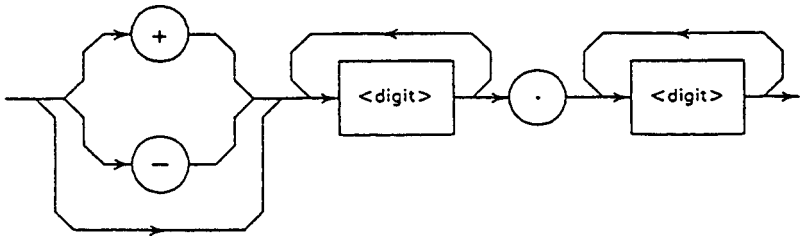
<RESPONSE DATA SEPARATOR> is used to separate data when multiple <RESPONSE DATA> are output.

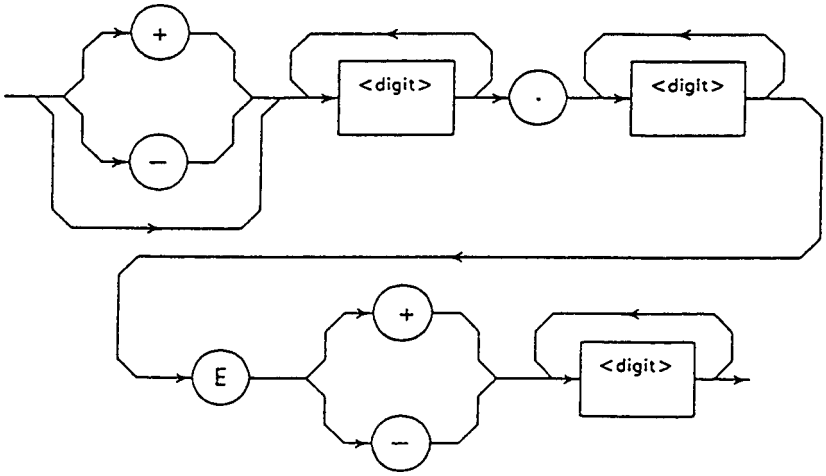
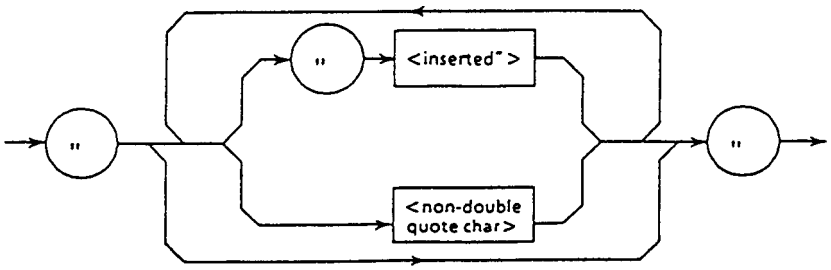
<RESPONSE DATA SEPARATOR> is defined as follows.

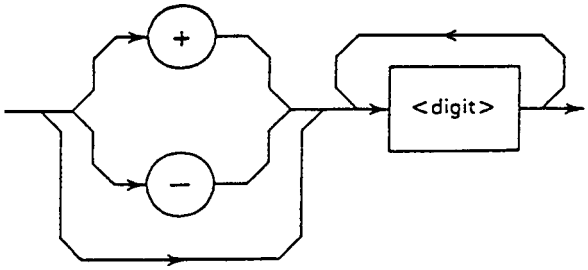
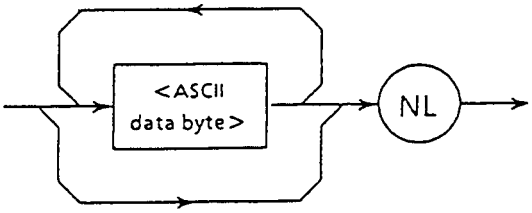


6.2.7 <RESPONSE DATA>

Which response data is returned is determined depending on the query message.

Element	Function
(1) CHARACTER RESPONSE DATA <Example> ABC DEFG	Express a short mnemonic data. 
(2) NR1 NUMERIC RESPONSE DATA <Example> 123 +123 -1234	Express a decimal integer. 
(3) NR2 NUMERIC RESPONSE DATA <Example> 12.3 +12.34 -12.345	Express numeric value with a fixed decimal point. 

Element	Function
(4) NR3 NUMERIC RESPONSE DATA <Example> 1.23E+45 -12.3E+45	Express a decimal real number with exponent. 
(5) STRING RESPONSE DATA <Example> "1234" "ABCD" " 1234.5"	Express a character string enclosed in double quotation marks. 

Element	Function
(6) ARBITARY ASCII RESPONSE DATA <Example> <ASCII Byte><ASCII Byte>NL^END	ASCII data bytes except NL characters are sent without separation. Since NL^END (or NL only) is inserted at the end, the message is terminated without exit. 1) For GPIB interface  2) For RS-232C or Ethernet interface 

6.2.8 Grammatical differences between the listener input and talker output formats

The grammatical differences between the listener input and talker output formats are shown below.

Listener input format: The program message format is so flexible that a device can easily receive a program message from the controller. Upper and lower case characters have the same meaning and any number of <white space> can be used as separator or terminator.
These enable you to write programs easy to read.

Talker outpour format: Contrary to the listener input format, the output messages are sent following the strict grammatical rules so that the controller can easily accept the response messages outputted form a device. There are only one response message for one function.

Grammatical Differences

Item	Listener input program message	Talker output program message
Characteristic	Flexible	Strict
Alphabetical characters	Not case-sensitive	Upper case only
Before/after "E" in NR3 exponent part	△ + E/e + △ (△ : 0 or more)	E (upper case) only
+ mark in NR3 exponent part	Can be omitted	Cannot be omitted
<white space> (: △) (Note 1)	Multiple <white space> can be inputted before/after the separator or before the terminator.	Not used
Unit separator	△ + semicolon (△ : 0 or more)	Semicolon only
Header affix space	△ + header (△ : 0 or more)	Header only
Header separator	header + △ (△ : 1 or more)	Header + one $20 (Note 2)
Data separator	△ + comma + △ (△ : 0 or more)	Comma only
Terminator	△ + {NL, EOI or NL+EOI} (△ : 0 or more)	NL + EOI

Note 1: △ represents a <white space>.

Note 2: ASCII code byte 20 (32 in decimal notation = ASCII character: SP, space)

Section 7 IEEE488.2 Common Commands

This section describes the IEEE488.2 common commands supported by MP1580A.

The common commands can be used commonly for the GPIB or RS-232C, Ethernet, interfaces.
All common commands supported by MP1580A are sequential commands.

The table below lists the IEEE488.2 common commands supported by MP1580A.

IEEE488.2 Common Command List

Mnemonic	Command's full spell
*IDN?	Identification Query
*RST	Reset Command
*TST?	Self Test Query
*OPC	Operation Complete Command
*OPC?	Operation Complete Query
*WAI	Wait Continue Command
*CLS	Clear Status Command
*ESE	Standard Event Status Enable Command
*ESE?	Standard Event Status Enable Query
*ESR?	Standard Event Status Register Query
*SRE	Service Request Enable Command
*SRE?	Service Request Enable Query
*STB?	Read Status Byte Query
*TRG	Trigger Command
*PSC	Power On Status Clear Command
*PSC?	Power On Status Clear Query
*SAV	Save Command
*RCL	Recall Command
*OPT?	Option Identification Query

*IDN? Identification Query

∇ Parameter None

- Response <ARBITRARY ASCII RESPONSE DATA>
  <Manufacturer>, <Model>, <Serial No. >, <Firmware revision No.>

  For MP1580A,
  ANRITSU,MP1580A,0123456789,****

  **** = 1 to 9999

- Function Reports the manufacturer name, mode, serial No. and firmware No.
  (Supplement) The lower 2 of 4 digits for the revision No. indicate decimal places.

- Reference item Screen Model type, Serial number

- Example use > *IDN?
  < ANRITSU,MP1580A,0123456789,1.00

*RST Reset Command

∇ Parameter None

- Function Returns MP1580A to the status immediately after power on.
  The measurement settings are maintained while measured result are cleared.
  To return MP1580A to the factory setting shipment, enter '0' to the SCPI command :SYSTem:MEMory:RECall0.

- Example use > *RST

*TST? Self Test Query

- Response <NR1 NUMERIC RESPONSE DATA>
  0 : No errors
  1 : Error detected
- Function Queries result of the self test (RAM check) carried out upon power on.
- Example use > *TST?
  < 1

*OPC Operation Complete Command

∇ Parameter None

- Function When execution of the preceding command completes, sets bit 0 (operation completion bit) of the standard event status register and then sets SRQ to ON.
- Example use > *RCL 1; *OPC

*OPC? Operation Complete

- Response <NR1 NUMERIC RESPONSE DATA>
  1
- Function Returns '1' when execution of the preceding command completes.
- Example use > *RCL 1; *OPC?
  < 1

*WAI Wait Continue Command

∇ Parameter None

- Function Makes execution of the succeeding commands wait until execution of the preceding command completes.
- Example use > *WAI

*CLS Clear Status Command

∇ Parameter None

- Function Clears all status structure except the output queues and their MAV summary messages. However, enable register and Transition filter are not cleared.
  When an *CLS is sent immediately after <PROGRAM MESSAGE TERMINATOR> and before <QUERY MESSAGE UNIT> element, the output queues and MAV bits are also cleared.
- Example use > *CLS

*ESE Standerd Event Status Enable Command

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>
An integer between 0 and 255.
The parameter represents the total of bit digit values when bits to be enabled form bits of the standard event status enable register.
The value for the bit to be enabled is set to 1. For the bit to be disabled, set to 0.
See "Section 7 Status Report" for details on the register configuration of MP1580A.

- Function Sets or clears the standard event status enable register.
- Example use Sets bits 2 and 4 for the standard event status enable register.
  > *ESE 20

*ESE? Standerd Event Status Enable Query

- Response <NR1 NUMERIC RESPONSE DATA>
  An integer between 0 and 255.
- Function Queries current value of the standard event status enable register.
- Example use > *ESE?
  < 20

*ESR? Standerd Event Status Register Query

- Response <NR1 NUMERIC RESPONSE DATA>
  An integer between 0 and 255.
  Sum of bit values of standard status register bits is responded.
  See "Section 7 Status Report" for details on the register configuration of MP1580A.
- Function Queries current value of the standard event status register.
- Example use When a command error exists.
  > *ESR?
  < 32

*SRE Service Request Enable Command

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>
An integer between 0 and 255.
The parameter represents the sum of bit values to be enabled from bits of the service request enable register.
The value for the bit to be enabled is set to 1. For the bit to be disabled, set to 0.
See "Section 7 Status Report" for details on the register configuration of MP1580A.

- Function Sets or clears the service request enable register.
- Example use Sets bit 4 for the service request enable register.
  > *SRE 16

*SRE? Service Request Enable Query

- Response <NR1 NUMERIC RESPONSE DATA>
  An integer between 0 and 255.
  Sum of bit values of service request enable register bits is responded.
  See "Section 7 Status Report" for details on the register configuration of MP1580A.
- Function Queries current value of the service request enable register.
- Example use > *SRE?
  < 16

*STB? Read Status Byte Query

- Response <NR1 NUMERIC RESPONSE DATA>
  An integer between 0 and 255.
  Sum of bit values of status byte register bits is responded.
  See "Section 7 Status Report" for details on the register configuration of MP1580A.
- Function Queries current value of the status byte register including MSS (Master Summary Status) bit.
- Example use When error occurred or the event queue is not empty.
  > *STB?
  < 4

*TRG Trigger Command

∇ Parameter None

- Function IOperation is the same as that for IEEE488.2 GET (Group Execute Trigger bus command).
  MP1580A starts or restarts measurement when received the *TRG command.
- Example use > *TRG

*PSC Power On Status Clear

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>
0: Sets power-on status clear flag to False.
Other than 0: Sets power-on status clear flag to True.

- Function Determines if service request, standard event status parallel pole enable registers of the status reporting section are cleared at power on.
When set to 0, enable registers is not cleared and the device can issue SRQ upon power on.
When set to 1, enable registers is cleared and the device cannot issue SRQ upon power on.

- Example use To issue SRQ without clearing power-on status flag:
> *PSC 0; *SRE 32; *ESE 128

*PSC? Power On Status Clear Query

- Response <NR1 NUMERIC RESPONSE DATA>
0: Power-on status flag is False
1: Power-on status flag is True.

- Function Queries true/false of the power-on status flag.

- Example use > *PSC?
< 0

*SAV Save Command

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No. 1 to 10

- Function Writes current setting data to the memory with the specified number.
(:SYSTem:MEMory:STORe has the same function.)
The state range saved is the same as that being affected by the *RST command.
The SCPI command :SYSTem:MEMory:STORe has the same function.

- Example use To write the data to memory No. 3:
> *SAV 3

*RCL Recall Command

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>
0 to 10

- Function Recalls the memory of the specified number and sets MP1580A to the previous settings accordingly.
The state range loaded is the same as that being affected by the *RST command.
The SCPI command :SYSTem:MEMory:RECall has the same function.
Parameter 0 has the same initialization function as that for the :SYSTem:MEMory:RECall0 command.

- Example use To recall the analyze data in memory No. 1:
> *RCL 1

*OPT? Option Identification Query

∇ Parameter None

- Response <ARBITRARY ASCII RESPONSE DATA>
  Character corresponding to the option or unit.
- Function Reports the list of the installed options.
  Each option number is reported being separated by comma.
  Options are reported in the order of the optional interface unit slot 1 to 2 for MP1580A.
  Each unit option is reported after the unit model.
- Example use > *OPT?< OPT01,OPT02,OPT03,OPT04, MU150018A,OPT02

Option Character Correspondence

Main body / Unit	Option No.	Character	Option Name
MP1580A/81A (Main body)	Option01	OPT01	RS-232C
	Option02	OPT02	GPIB
	Option03	OPT03	Ethernet
	Option04	OPT04	VGA output
MU150018A (10G/2.5G Jitter)	Option01	–	(N/A)
	Option02	OPT02	Wander
	Option03	OPT03	Wander ref. output

Section 8 Status Report

This section describes the status register configuration and bit definition for equipment-unique status register.

The status register configuration of MP1580A conforms to the SCPI (Standard Command for Programmable Instruments) standard. See "Section 9 SCPI Overview" for details.

8.1 Status Register Configuration for MP1580A

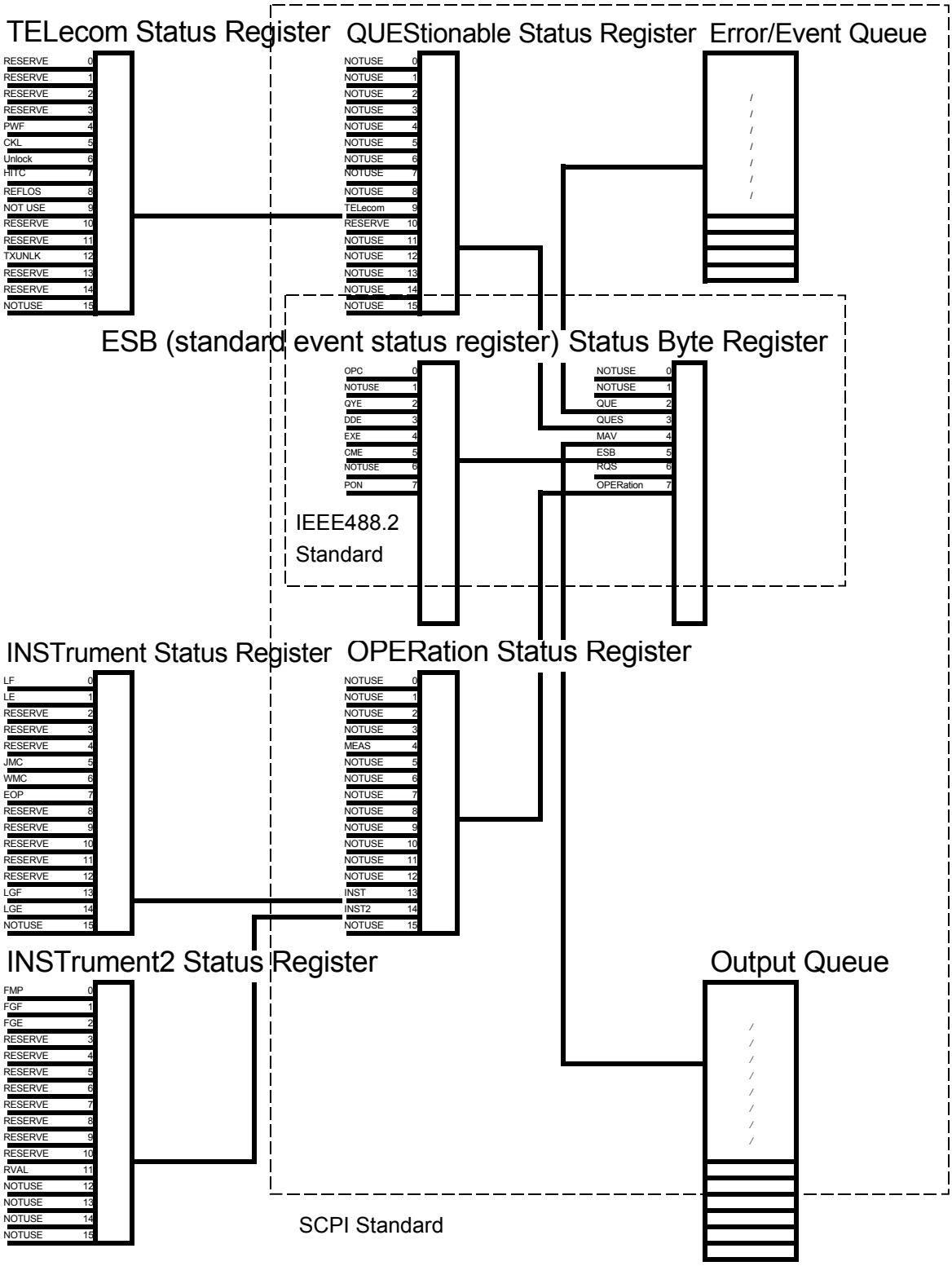
The SCPI requires to configure the status registers specified by the IEEE488.2, and SCPI-unique OPERation and QUEStionable status registers.

The status registers installed to MP1580A are shown below.

Registers specified by IEEE488.2	Standard event register, Status byte register
Registers specified by SCPI	QUEStionable status register, OPERation status register
MP1580A-unique registers	TELecom status register, INSTrument status register, INSTrument2 status register

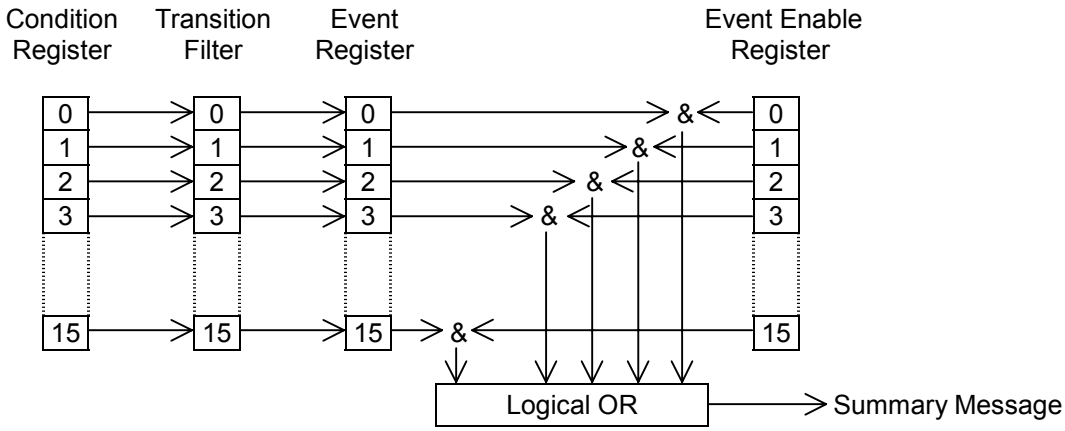
Simplified configuration drawing of status registers installed to MP1580A is shown in the next page. (Bit positions and width will be explained later.)

The status register configuration for MP1580A is shown below.



Status Register Configuration

Status registers configuration except one for the IEEE488.2 standard is shown below.



Status Register Configuration

Register/Filter Function

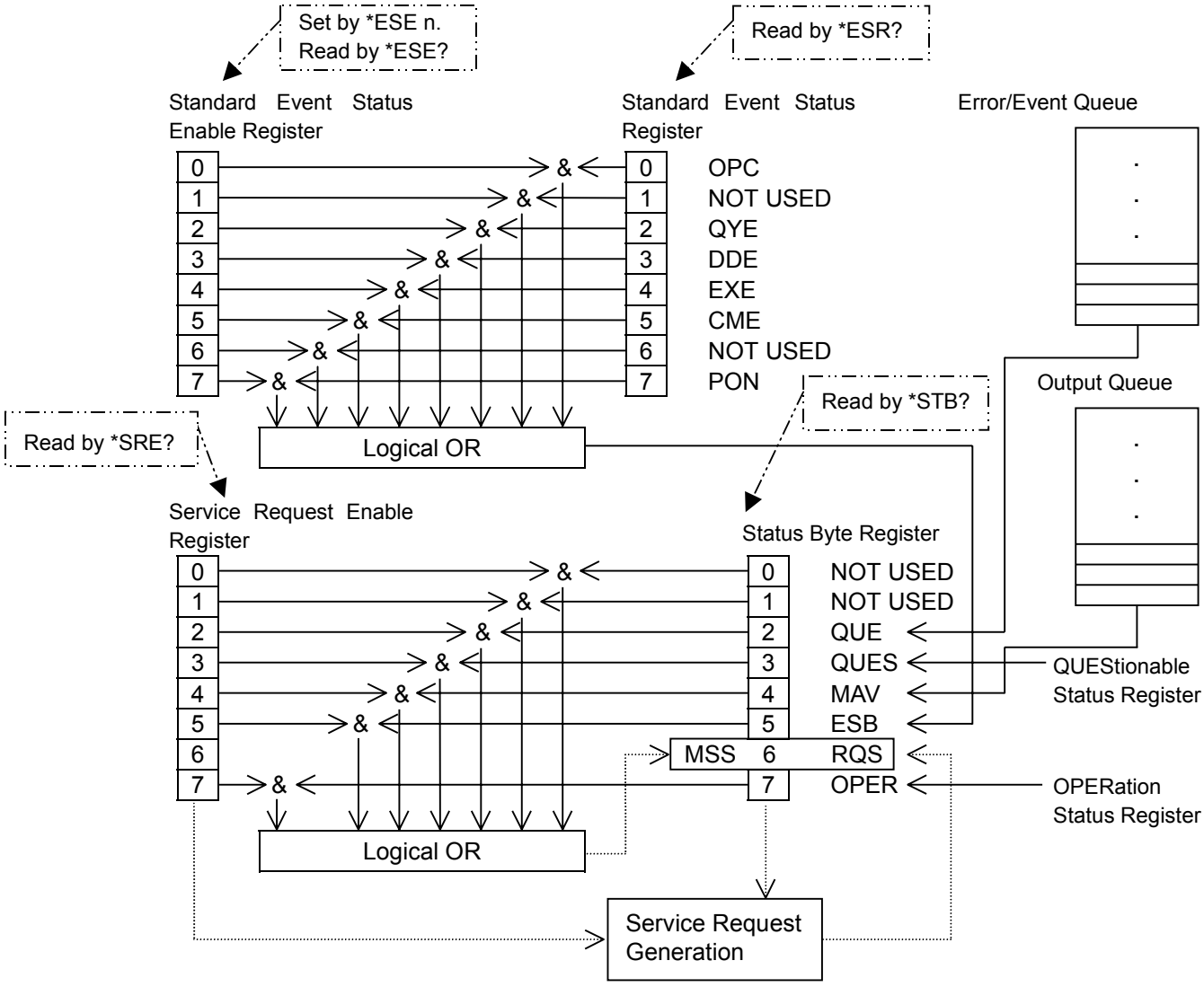
Register/Filter	Function
Condition Register	Monitors device status and changes in real time according to device status. This register therefore does not memorize status.
Transition Filter	Set contents of the Condition Register to Event Register. Transition Filter conducts one of the following 3 evaluations for changes of Condition Register: Positive direction change: Event becomes true only when the corresponding condition changes from false to true. Negative direction change: Event becomes true only when the corresponding condition changes from true to false. Bi-directional change: Event becomes true when positive or negative direction change occurs.
Event Register	Memorizes the Transition Filter output.
Event Enable Register	Selects the bit of corresponding Event Register to be set to make the summary message true.

8.2 IEEE488.2 Specified Status Register

The IEEE488.2 specifies the following two status registers:

Status Register Functions Specified by IEEE488.2

Status Byte Register	Capable of setting RQS and 7 summary message bits. This register is used in combination with service request enable register and sets SRQ to ON when logical-OR operation of both registers is other than 0. System reserves bit 6 for RQS, and reports service request to the external controller using this bit.
Standard Event Status Register	The 8 standard events that the device encounters are registered at this register. Logical OR output bit is indicated at bit 5 of status byte register as ESB (Event Status Bit) summary message.



Bit Definition for IEEE488.2 Standard Register

Bit definition for the status byte register and standard event status byte register is shown below.

Bit Definition for Status Byte Register

DB2	QUE (Error/Event QUEue)	Indicates that the error and event queues are not empty.
DB3	QUES (QUEStionable status register summary)	QUEStionable status register summary
DB4	MAV (Message Available)	Indicates that the output queue is not empty.
DB5	ESB (Event Summary Bit)	Standard event status register summary
DB6	RQS (ReQuest Service) MSS (Master Summary Status)	RQS message Indicates at least one cause for requesting service exists in device.
DB7	OPER (OPERation status register summary)	OPERation status register summary

Bit Definition for Standard Event Status Register

DB0	OPC (Operation Complete)	Indicates that all specified operations have completed.
DB3	DDE (Device-dependent Error)	Indicates that an error other than command and execution errors occurred.
DB4	EXE (Execution Error)	Indicates that an execution error occurred.
DB5	CME (Command Error)	Indicates that a command error occurred.
DB7	PON (Power on)	Indicates that power-on status has changed from OFF to ON.

8.3 SCPI Specified Status Register

The SCPI requires that the following registers are installed in addition to those specified by the IEEE488.2.

Functions of SCPI Specified Status Registers

QUEStionable Status Register	Reports the signal status such as measurement result. Used to request service to the controller when an error occurred.
OPERation Status Register	Reports a part of the instrument status.

Bit definition for the above status registers is shown below.

Bit Definition for QUEStionable Status Register

DB9	TEL (TELecom status register summary)	TELecom status register summary

Bit Definition for OPERation Status Register

DB4	MEAS (MEASuring)	Indicates that a measurement is being in progress. (Note) (During calibration, this bit indicates that a calibration is being in progress.)
DB13	INST (INSTrument status register summary)	INSTrument status register summary
DB14	INST2(INSTrument2 status register summary)	INSTrument2 status register summary

Note: When a measurement is re-started during a measurement, MP1580A becomes Inactive and then becomes Active again.
DB4 bit is set to ON when a measurement is in progress.

8.4 MP1580A Unique Status Register

The table below lists status registers unique to MP1580A.

Functions of MP1580A Unique Registers

INSTrument Status Register	Reports a part of the instrument status.
INSTrument2 Status Register	Reports a part of the instrument status.
TELecom Status Register	Reports the signal status.

Bit definition for the above status registers are shown below.

Bit Definition for INSTrument Status Register

DB0	LF (Log Full)	Indicates that the graph data (Peak jitter graph) is full.
DB1	LE (Log Empty)	Indicates that the graph data (Peak jitter graph) is empty.
DB5	JMC (Jitter Measure Complete)	Indicates the end of a jitter measurement. (Note)
DB6	WMC (Wander Measure Complete)	Indicates the end of a wander measurement. (Note)
DB7	EOP (End Of Printer period)	Indicates the intermediate print timing for printer.
DB13	LGF (Logging memory Full)	Indicates that the logging memory is full.
DB14	LGE (Logging memory Empty)	Indicates that the logging memory is empty.

Note: Both of DB5 and DB6 are reported on completion of a jitter/wander measurement.

Bit Definition for INSTrument2 Status Register

DB0	FMP (Freq. Monitor Period)	Indicates the update timing for frequency monitor.
DB1	FGF (Freq. Graph Full)	Indicates that the Freq. graph data is full.
DB2	FGE (Freq. Graph Empty)	Indicates that the Freq. graph data is empty.
DB11	RVAL (Revaluation)	Indicates the end of re-evaluation.

Note: Status bits (FMP, FGF and FGE) for Frequency monitor are not changed excepting for manual measurement (Test menu: Manual.)

Bit Definition for TELecom Status Register

DB4	PWF (Power Fail)	Indicates that a power failure occurred.
DB5	CKL (Clock Loss)	Indicates that a clock loss has occurred on the Tx side
DB6	RXUNLK (Rx Unlock)	Indicates that Rx Unlock has occurred.
DB7	HITC (Hit count)	Indicates that Hit count is calculated during jitter measurement.
DB8	REFLOS (REF LOS)	Indicates that REF LOS has occurred during wander measurement.
DB12	TXUNLK (Tx Unlock)	Indicates that Tx Unlock has occurred.

Note: Bits not used in MP1580A are defined as "not used" even if they are specified in IEEE488.2 or SCPI standard.

8.5 Reading/Writing/Clearing the Status Register

Reading, writing and clearing procedures for the status register are described below.

(1) Read and write procedures

Reading and writing procedures for the status register are described below. ("Disabled" in the table below indicates that there are no means for writing.)

Table 7-6 Read/Write Procedures for the Status Register

Register	Read procedure	Write procedure
Status Byte Register	Serial pole (when using a GPIB interface bus) 7-bit status byte and RQS message bit are responded. At this time, status byte value is not changed. *STB? Status byte register contents and one numeric value from MSS summary message are responded. At this time, status byte value is not changed.	Disabled
Service Request Enable Register	*SRE? At this time, register contents are not changed.	*SRE
Standard Event Status Register	*ESR? At this time, register contents are cleared after reading.	Disabled
Standard Event Status Enable Register	*ESE? At this time, register contents are not changed.	*ESE
SCPI Event Register	:STATus:...:EVENt? At this time, register contents are cleared.	Disabled
SCPI Enable Register	:STATus:...:ENABle? At this time, register contents are not changed.	:STATus:...:ENABle
SCPI Transition Filter	:STATus:...:PTRansition? :STATus:...:NTRansition? At this time, register contents are not changed.	:STATus:...:PTRansition :STATus:...:NTRansition
Error/Event Queue	:SYSTem:ERRor?	Disabled

Note: The SCPI event register, SCPI enable register and SCPI Transition filter represent event register, enable register and transition filter of MP1580A unique status registers, in addition to the SCPI status register, respectively.

Common commands → Section 7
SCPI command details → Section 11

(2) Clear and reset procedures

*RST, *CLS, power-on or STATus:PRESet, etc. performs clearing/resetting the status registers, as described below.

The table below lists the status register status after executing these operations.

Table 7-7 Clear/Reset Procedures for Status Register

Register	*RST	*CLS	Power ON	STATus:PRESet	Other clear/reset procedure
Status Byte Register	Not changed	Cleared	Cleared	Not changed	
Service Request Enable Register	Not changed	Not changed	Cleared #	Not changed	Executing *SRE 0.
Standard Event Status Register	Not changed	Cleared	Cleared	Not changed	Cleared when the event is read by *ESR?
Standard Event Status Enable Register	Not changed	Not changed	Cleared #	Not changed	Executing *ESE 0.
SCPI Event Register	Not changed	Cleared	Cleared #	Not changed	Cleared when the event is read by :STATus:...:ENENt?
SCPI Enable Register	Not changed	Not changed	Reset #	Reset	Executing :STATus:...:ENABle 0.
SCPI Transition Filter	Not changed	Not changed	Reset #	Reset	Executing :STATus:...:PTRansition 0 and :STATus:...:NTRansition 0.
Error/Event Queue	Not changed	Cleared	Cleared	Not changed	All events are read by :SYSTem:ERRor?

Note: The SCPI event register, SCPI enable register and SCPI Transition filter represent the event register, enable register and transition filter of MP1580A unique status registers, in addition to the SCPI status register, respectively.

#: Cleared (or reset) when power is turned on by *PSC with the PSC (Power-ON Status Clear) flag of true.

The table below lists values after reset for registers and filters being affected by the :STATus:PRESet command.

Table 7-8 Values after Reset for Registers/ Filters being Affected by the :STATus:PRESet Command

Register	Enable/Filter	Value after reset
OPERational Status Register QUEStionable Status Register	Enable Register PTRansition Filter NTRansition Filter	All 0 All 1 All 0
INSTrument Status Register INSTrument2 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

Section 9 SCPI Overview

This section describes overview of the SCPI and the command system.

MP1580A adopts commands specified by the SCPI (Standard Commands for Programmable Instruments) as the remote control commands.

The following symbols are used in this and following sections for explaining command usage and response examples:

> *Program message (program command or query command)*

< *Response*

9.1 SCPI Overview

The SCPI is the equipment command language (defined by the SCPI Consortium) which does not depend upon the hardware.

Purpose of the SCPI lies in shortening the program development period for automatic measuring equipment (ATE) by unifying the programming environment including the equipment control and data handling. It also features in that the identical control can be used with controlling an equipment having the same functions, since interchangeability is ensured between the same models or with a different model having the same functions.

9.2 Command Configuration

:DISPlay	:SETup	[:NAME]	<sdisplay>
		[:NAME]?	
:SYSTem	:MEMory	:RECall	<numeric>
		:STORe	<numeric>

Example of SCPI Command Tree

(See Appendix B for the command list.)

The SCPI commands are constructed by the layer composition. Commands are grouped by related functions and each group forms a layer construction called a subsystem.

The same header is often used for different commands. However, the same header corresponds to different functions by its position. For this reason, a header needs to be described by the full path to the header used.

Example: SYSTem subsystem

SYSTem → The highest node
MEMory → The second level node
RECall,STORe → The third level nodes
<numeric> → Parameter types for RECall and STORe

9.3 Command Description Method

```
:DISPlay:SETup[:NAME]<pattern>
        <sdisplay> = "IFACe","JTRansfer","JWANder","PRINt","SYSTem","MEMory"
                     "FDISk","OREVision"
:DISPlay:SETup[:NAME]?
:SENSe:MEASure:PERiod <numeric>, <suffix>
                  <numeric> = 1 to 99, <suffix> = D,H,M,S
:SENSe:MEASure:PERiod?
```

Example of SCPI Commands

The command trees shown on the previous page are actually expressed as the above commands.
The rules on command description are explained below.

Command description → <Command Format>
[ ] for [:TYPE] → <Option Node>
Upper/Lower case characters → <Header Informative Form>

<Command Format>
A command always begins with ":".
Command headers are combined by ":".

<Header Informative Form>
Headers are classified into the short and long forms.
The short form is the informative form of the long form and only contains the portion of upper case characters. The same function expressed by a command, whichever the short or long form is used, is interpreted as the same command. (The short and long forms may also be mixed.)
Although upper and lower case characters are used for the command reference for identifying the short and long forms, upper and lower cases are not identified in the actual use (SENSE, Sense and sense are interpreted as the identical header.)

Example: Long form > :SENSe:MEASure:PERiod 1,M
Short form > :SENS:MEAS:PER 1,M
Long + Short > :SENse:MEAS:PERiod 1,M
Lower case characters only > :sense:measure:period 1,m

<Option Node>
[ ] represents an option node. The header enclosed in [ ] can be omitted.

Example: When the header is not omitted > :DISPlay:SETup[:NAME] "IFACe"
When the header is omitted > :DISPlay:SETup "IFACe"

<Command Separator>
At least one space is necessary between a command and a parameter. Two or more parameters need to be separated by ",".

9.4 Command Combination

Commands may be combined using ';'. The second command is referred to as the same level of the lower-most layer of the first command.
For this reason, the second command may be described in full path as shown in Example 1 or with the header upper than the TYPE omitted as shown in Example 2 (the bolded part in Example 1 are omitted in Example 2.)

Example 1: Describing two commands in full path.
>:SYSTem:DATE 2001,3,10;**:SYSTem**:TIME 14,0,10
Example 2: Set the lower-most level nodes, DATE and TIME, at the same time.
>:SYSTem:DATE 2001,3,10;TIME 14,10,10

9.5 Parameters

The table below lists the parameter types used for MP1580A.
The parameter types are shown by lower case characters enclosed in parentheses < > as shown in the table below.

The <PROGRAM DATA> types specified by IEEE488.2 (or SCPI) corresponding to parameter types are indicated by upper case characters.

Correspondence between each parameter type and IEEE488.2 (or SCPI) specified <PROGRAM DATA> type is described for each command.

Functions of Parameter Types

Parameter Type	Function
<numeric> <DECIMAL NUMERIC PROGRAM DATA>	Indicates a decimal value. <CHARACTER PROGRAM DATA> such as MINimum and MAXimum are also included as special numeric types. Values used by MP1580A are mainly the integer type. Decimal parts are rounded to the nearest natural number.
<boolean> <BOOLEAN PROGRAM DATA> (Defined by SCPI.)	Indicates a logical value. OFF or 0 indicates False while ON or 1 indicates True. Either 0/1 or OFF/ON can be used for setting. However, 0 or 1 is used in a response to an query.
<string> <STRING PROGRAM DATA>	Indicates a character string consisting of ASCII characters enclosed in single or double quotation marks. Either the long form or short form can be used. Example: 'LOF:M139' or "LOF: M139"
<brate>, <type>..etc <CHARACTER PROGRAM DATA>	Indicates character data expressed by a short character string corresponding to the setting content. Either long form or short form can be used.

9.6 Response

The response formats to query commands and response types used by MP1580A are listed in the table below. In this manual, response types are shown by lower case characters enclosed in parentheses < > in the table below.

The <RESPONSE DATA> types specified by IEEE488.2 (or SCPI) corresponding to response types are indicated by upper case characters.

Response Format (by Response Type)

Response Type	Format
<numeric>,<year>,... (<NR1 NUMERIC RESPONSE DATA>)	The number of response digits is made variable and the maximum number of digits for numeric range is matched with the maximum number of response digits. No spaces are inserted between sign and numerals. > :SYSTem:DATE? < 2001,7,14
<numeric>,<year>,... (<NR2 NUMERIC RESPONSE DATA>)	The number of response digits is made variable and the maximum number of digits for numeric range (including decimal place) is matched with the maximum number of response digits. No spaces are inserted between sign and numerals. > :SOURce:JITTer:MANual:FREQuency? < 100.0,HZ
<brate>,<type>..etc (<CHARACTER RESPONSE DATA>)	Short form characters are responded. Example: > :DISPlay:ANALysis:TGRaph:PRINt? < ALL
<string>,<display>..etc (<STRING RESPONSE DATA>)	String enclosed in " marks is responded. When both long and short forms can exist for contents of string corresponding to the program command, the short form string is responded. > :DISPlay:DSELect:NAME? < "SET" (Short form of "SETup") For response formats Form1 to Form14, see the following table.

Response Formats (Form1 to Form14)

Type	Format	Description
Form1 Integer type 1	" XXXXXX"	When 0 ≤ value ≤ 999,999, 6 among 8 characters with right justification. > :CALCulate:DATA? "HIT:COUNt" < " 892"
	" X.XE+XX"	When 1.0E+06 ≤ value ≤ 9.9E+15, 7 among 8 characters with right justification. > :CALCulate:DATA? "HIT:COUNt" < " 1.2E+07"
	">9.9E+15"	When 1.0E+16 ≤ value.
	"--------"	When no corresponding data is available for the query (8 characters)
Form2 Decimal type 1	" 0.0E-XX"	When value = 0, 7 among 8 characters with right justification.
	"<1.0E-15"	When value ≤ 9.9E-16.
	" X.XE-XX"	When 1.0E-15 ≤ value ≤ 9.9E-01, 7 among 8 characters with right justification. > :CALCulate:DATA? "EC:BIT" < " 1.0E-05"
	" 1.0E-00"	When value = 1, 7 among 8 characters with right justification.
	"--------"	When no corresponding data is available for the query (8 characters)
Form3 Percentage type	"XXX.XXXX"	0.0000 to 100.0000 > :CALCulate:DATA? "HIT:FS" < " 1.0010"
	"--------"	When no corresponding data is available for the query (8 characters)
Form3' Percentage type	"XXX.X"	0.0 to 100.0 > :CALCulate:LMONitor:TRAFfic:DATA? < ....," 45.2", ...
	"-----"	When no corresponding data is available for the query (5 characters)
Form4 Judgment type 1	" Acceptable" "Unacceptable" " Degraded"	Expressed by up to 12 characters with right justification. When the judgment result is OK: "Acceptable" When NG: "Unacceptable"
	"----------"	When no corresponding data is available for the query (12 characters)
Form5 UI type	" X.XXX" (Jitter Tx)	When 0.000 ≤ value ≤ 0.505 for the 0.5 UI range in UIp-p. When 0.000 ≤ value ≤ 0.252 for the 0.5 UI range in UI+p, UI-p. When 0.000 ≤ value ≤ 0.178 for the 0.5 UI range in UIrms. 5 among 6 characters with right justification > :CALCulate:DATA? "JAMPlitude:PTPeak" < " 0.234"
	" XX.XX"	When 0.00 ≤ value ≤ 20.20 for the 20 UI range in UIp-p. When 0.00 ≤ value ≤ 10.10 for the 20 UI range in UI+p, UI-p. When 0.00 ≤ value ≤ 7.14 for the 20 UI range in UIrms. 5 among 6 characters with right justification > :CALCulate:DATA? "JAMPlitude:RMS" < " 7.00"
	" XX.XX" (Jitter Tx)	When 0.00 ≤ value ≤ 80.80 for the 80 UI range in UIp-p. When 0.00 ≤ value ≤ 40.40 for the 80 UI range in UI+p, UI-p. When 0.00 ≤ value ≤ 28.56 for the 80 UI range in UIrms. 5 among 6 characters with right justification > :CALCulate:DATA? "JAMPlitude:RMS" < " 28.56"

Type	Format	Description
Form5 UI type	" XXXX"	When 0.00 ≤ value ≤ 1010 for the 1000 UI range in UIp-p. When 0.00 ≤ value ≤ 505 for the 1000 UI range in UI+p, UI-p. When 0.00 ≤ value ≤ 357 for the 1000 UI range in UIrms. When 0.00 ≤ value ≤ 4040 for the 4000 UI range in UIp-p. When 0.00 ≤ value ≤ 2020 for the 4000 UI range in UI+p, UI-p. When 0.00 ≤ value ≤ 1428 for the 4000 UI range in UIrms. 4 among 6 characters with right justification > :CALCulate:DATA? "JAMPlitude:RMS" < " 1000"
	">0.505"	When value > 0.505 (0.5UIrange, UIp-p)
	">2.020"	When value > 2.020 (2UIrange, UIp-p)
	">20.20"	When value > 20.20 (20UIrange, UIp-p)
	">80.80"	When value > 80.80 (80UIrange, UIp-p)
	"> 1010"	When value > 1010 (1000UIrange, UIp-p)
	"> 4040"	When value > 4040 (4000UIrange, UIp-p)
	">0.252"	When value > 0.252 (0.5UIrange, UI+p/UI-p)
	">1.010"	When value > 1.010 (2UIrange, UI+p/UI-p)
	">10.10"	When value > 10.10 (20UIrange, UI+p/UI-p)
	">40.40"	When value > 40.40 (80UIrange, UI+p/UI-p)
	"> 505"	When value > 505 (1000UIrange, UI+p/UI-p)
	"> 2020"	When value > 1010 (4000UIrange, UI+p/UI-p)
	">0.178"	When value > 0.178 (0.5UIrange, UIrms)
	">0.714"	When value > 0.714 (2UIrange, UIrms)
	"> 7.14"	When value > 7.14 (20UIrange, UIrms)
	">28.56"	When value > 28.56 (80UIrange, UIrms)
	"> 357"	When value > 357 (1000UIrange, UIrms)
	"> 1428"	When value > 1428 (4000UIrange, UIrms)
	"------"	When no corresponding data is available for the query (6 characters)
Form6 dB type	" XX.XX"	When value > 0, 5 among 7 characters with right justification. > :CALCulate:DATA? "JTRansfer:POINt1" < " 100, 10.00, 0.50, Acceptable"
	" 0.00"	When value = 0.
	" -XX.XX"	When value < 0, 6 among 7 characters with right justification. > :CALCulate:DATA? "JTRansfer:POINt1" < " 100, 10.00, -10.00, Acceptable"
	"> 10.00"	When value > 10.00
	"<-80.00"	When value < -80.00
	" ***.**"	When value = ***.** (parameter overflow)
	"UNLOCK"	When a time out occurred.
	"-------"	When no corresponding data is available for the query (7 characters)
Form7 Time type 1	" X.X"	When 0 ≤ value < 10, 3 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:TIE" < " 5.0"
	" XXXXXX"	When 10 ≤ value ≤ 999999, 6 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak" < " 1000"
	" -X.X"	When -10 < value < 0, 4 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:TIE" < " -5.0"

Type	Format	Description
Form7 Time type 1	" –XXXXXX"	-999999 ≤ value ≤ -10, 7 among 8 characters with right justification. The sign moves according to the number of digits. > :CALCulate:DATA? "WANDer:TIE" < " -1000"
	" X.XEXX"	1.0E6 ≤ value ≤ 1.0E10, 6 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:POINt3" < " 4.0E04"
	" -X.XEXX"	-1.0E10 ≤ value ≤ -1.0E6, 7 among 8 characters with right justification.
	"> 1.0E10"	When value > 1.0E10
	"<-1.0E10"	When value < -1.0E10
	"--------"	When no corresponding data is available for the query (8 characters)
	"CLKLOSS"	When a Clock loss occurred.
	"REFLOSS"	When a Ref. loss occurred.
	"UNLOCK"	When a hardware error occurred.
Form8 Time type 2	" X.X"	0 ≤ value < 10, 3 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak" < " 5.0"
	" XXXXXX"	10 ≤ value ≤ 999999, 6 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak" < " 5000"
	" X.XEXX"	1.1E6 ≤ value ≤ 1.0E10, 6 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak " < " 4.0E04"
	" -X.X"	-10 < value < 0, 3 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak" < " -5.0"
	" –XXXXXX"	-999999 ≤ value ≤ -10, 6 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak" < " -5000"
	" -X.XEXX"	-1.0E10 ≤ value ≤ -1.1E6, 6 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:PTPeak " < " -4.0E04"
	"--------"	When no corresponding data is available for the query (8 characters)
	">X.XEXX"	When overflowed, 7 among 8 characters with right justification. (>2.0E10, >1.0E10. one-side overflow)

Type	Format	Description
Form8' Time type 3	" X.XX"	0 ≤ value < 1, 4 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:POINt1" < " 0.10, 11"
	" X.X"	1 ≤ value < 10, 3 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:POINt1" < " 1.0, 19"
	" XXXX"	10 ≤ value < 1000, 4 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:POINt1" < " 12, 83"
	" XXXXXX" (for Linear)	1.0E3 ≤ value ≤ 1.2E5, 5 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:POINt1" < " 120000, 83"
	" X.XEXX" (for Log)	1.0E3 ≤ value ≤ 1.2E5, 5 among 8 characters with right justification. > :CALCulate:DATA? "WANDer:POINt1" < " 1.0E04, 12"
	"-------"	When no corresponding data is available for the query (8 characters)
Form9 Judgment type 2	"UNLOCK"	When an Unlock occurred. < :CALCulate:DATA? "JMANual:STATus" > "UNLOCK"
	"------"	When an Unlock does not occur (6 characters)
Form10 Integer type 2 (relative frequency)	" XXXXXXX.X"	When 9953M, 0 ≤ value ≤ 9,953,300.0 When 2488M, 0 ≤ value ≤ 2,488,300.0 9 among 11 characters with right justification. > :DISPlay:ANALysis:FMONitor:FREQuency? < "9953280000.0","9953280.0"," +1000.0"
	" -XXXXXXX.X"	When 9953M, -9,953,300.0 ≤ value < 0 When 2488M, -2,488,300.0 ≤ value < 0 10 among 11 characters with right justification.
	">9953300.0"	When 9953M, value > 9,953,300.0
	">2488300.0"	When 2488M, value > 2,488,300.0
	"<-9953300.0"	When 9953M, value < -9,953,300.0
	"<-2488325.0"	When 2488M, value < -2,488,325.0
	"----------"	When no corresponding data is available for the query (11 characters)
Form10' Integer type 4 (frequency)	" XXXXXXXXXX.X"	When 9953M, 9,943,326,700.0 ≤ value ≤ 9,963,233,300.0 When 2488M, 2,485,831,700.0 ≤ value ≤ 2,490,808,300.0 12 among 13 characters with right justification. > :DISPlay:ANALysis:FMONitor:FREQuency? < "9953280000.0","9953280.0"," +1000.0"
	">9963233300.0"	When 9953M, value > 9,963,233,300.0
	">2490808300.0"	When 2488M, value > 2,490,808,300.0
	"<9943326700.0"	When 9953M, value < 9,943,326,700.0
	"<2485831675.0"	When 2488M, value < 2,485,831,675.0
	"----------"	When no corresponding data is available for the query (13 characters)

Type	Format	Description
Form11 ppm type	" +XXXX.X"	When value > 0, 7 among 8 characters with right justification. > :DISPlay:ANALysis:FMONitor:FREQuency? < "9953280000.0","9953280.0"" +1000.0"
	" 0.0"	When value = 0
	" -XXXX.X"	When value < 0, 7 among 8 characters with right justification. The sign moves according to the number of digits. > :DISPlay:ANALysis:FMONitor:FREQuency? < "9953280000.0","-9953280.0"" -1000.0"
	">+1000.0"	When value > 1000.0
	"<-1000.0"	When value < -1000.0
	"--------"	When no corresponding data is available for the query.
Form13 Integer type 3	"+XXXXXXXX.X"	When 0 < value ≤ 990,000,000, 10 among 11 characters with right justification. > :DISPlay:ANALysis:FMONitor:FREQuency? < " 120000000.9"," +1000.0"," +1000.0"
	" X.X"	When value = 0, 10 among 11 characters with right justification. > :DISPlay:ANALysis:FMONitor:FREQuency? < " 120000000.9"," +1000.0"," 0.0"
	"-XXXXXXXX.X"	When -9,999,999.9 ≤ value < 0, 10 among 11 characters with right justification. > :DISPlay:ANALysis:FMONitor:FREQuency? < " 120000000.9"," +1000.0"," -1000.0"
	"----------"	When no corresponding data is available for the query.
Form14 Judgment type 3	“CLKLOSS”	Clock loss
	“-------“	When a Clock loss does not occur.

9.7 Buffer Size

Responses in the formats described in "Section 9.6 Response" are responded from MP1580A to the controller. So, the controller must have buffers sufficient to receive responses. The MP1580A command uses a maximum of 511 bytes.

Section 10 Command Execution

This section describes operations to execute device messages and details of commands. Read this section before using the device messages.

10.1 Manual and Remote Operations

This paragraph briefly describes the difference between manual operation and remote operation.

Manual operation is performed as follows:

1) Move the cursor to the item to be set.
2) Press Set (a window opens).
3) Select a parameter.
4) Press Set to close window and complete the setting.

Remote operation is performed as follows:

1) Send a program message.

Sending a program message can make the same setting as the operation above.

<Example>
Setting bit rate to 9,953 M:

Manual → Move the cursor to "Bit rate" on the Setup: Interface screen and select to specify "9953M."

Remote → Send :SOURce:TELecom:BRATe M9953.

10.2 Deciding Command

When determining the program message sent to MP1580A, search command details described in Section 10 for a command corresponding to each function and determine the command.

1) Searching the screen
   Determine a function to be sent and then search the corresponding screen.

2) Searching for the command
   Searching the screen in command details enables you to easily find the command details you want.

For details of functions and screens, refer to the separate manual of Vol. 1 Jitter and Wander Measurement.

When the objective screen is found, search for the command described in the Device Message Details in Section 11.

Section 10 describing the command details is configured as follows:

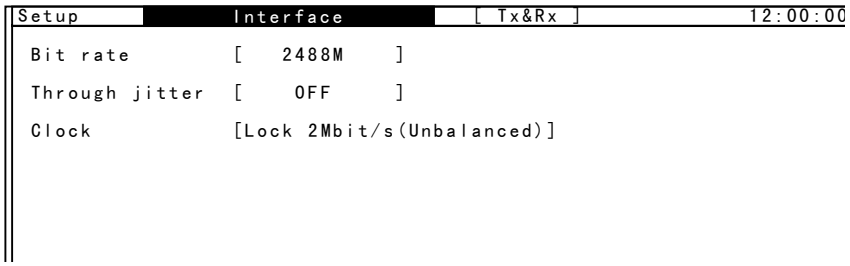
- MP1580A screens (Setup, Test menu, Result and Analyze screens) → See [(1)]
- Command list corresponding to screens → See [(2)]
- Command details shown in the list → See [(3)]

Device message details in Section 11 are configured as follows:

[(1)]
Figure of MP1580A screen (the right figure shows the Setup: Interface screen)

11.1.2 Interface subscreen

* For Tx&Rx



[(2)]
Command list corresponding to [(1)]

Omitted along the way.

Setting Item	Command
(Tx&Rx mode)	:INSTrument:COUPle

Setting Item	Command
Tx Bit Rate	:SOURce:TELecom:BRATe
Rx Bit Rate	:SENSe:TELecom:BRATe
Through jitter	:ROUTe:THRough:JITTer
Clock	:SOURce:TELecom:CLOCk:SOURce

[(3)]
Command details

:INSTrument:COUPle

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>
ALL Tx&Rx (Identical)
NONE Tx/Rx (Independent)

- Function Sets Tx and Rx sections to identical or independent status.

Omitted for the following.

10.3 Referring to Command Details

This paragraph describes how to search for a command in the following situations, taking examples.

- Setting MP1580A → See "10.3.1 Setting".
- Making measurement with MP1580A → See "10.3.2 Measurement".
- Reading a measured result → See "10.3.3 Reading measured result".
- Saving data → See "10.3.4 Saving data".

10.3.1 Setting

<Example> Set a bit rate on the transmission side.

1) Suppose the screen to be set.
   According to the separate manual of Vol. 1 (Jitter and Wander Measurement), it is found that the setting is allowed on the Setup: Interface screen.
   On the Setup: Interface screen, the following functions can be set:

   - Selecting the transmission and reception modes (Tx&Rx or Tx/Rx)
   - Setting the bit rate
   - Setting the jitter adding function
   - Setting a clock source

2) See the paragraph "Setup: Interface" screen of the "Command Details" section.

   (→ on the next page) (→ 11.1.2 Setup: Interface)

3) Refer to the command list for the item to be set. (→ (2) on the next page)
   The setting item is Tx Bit rate and the corresponding command is :SOURce:TELecom:BRATe.

4) See the command details shown on the reference page and determine the parameter to be set. Select M9953 here. (→ (3) on the next page)

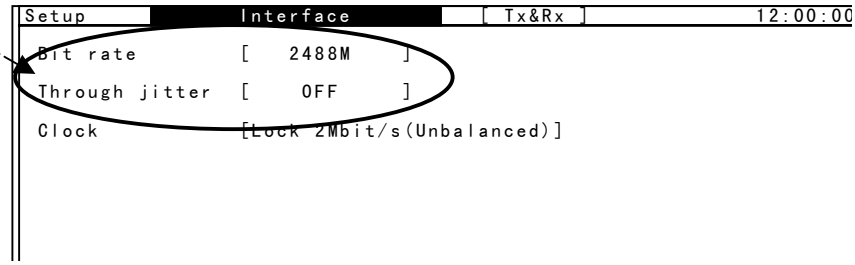
5) The following is the program message to be transmitted:

   :SOURce:TELecom:BRATe M9953

[(1)]
Figure of MP1580A screen. The setting item for the bit rate is Bit rate in the right figure.

11.1.2 Interface subscreen

* For Tx&Rx



[(2)]
The command is
:SOURce:TELecom:BRATe.

[(2)]
Command list corresponding to (1). Search for Bit rate on the transmission side (Tx).

Setting item	Command
(Tx&Rx mode)	:INSTrument:COUPle

Setting item	Command
Tx Bit Rate	:SOURce:TELecom:BRATe
Rx Bit Rate	:SENSe:TELecom:BRATe
Through jitter	:ROUTe:THRough:JITTer
Clock	:SOURce:TELecom:CLOCk:SOURce

Omitted along the way.

[(3)]
Command details. Select a parameter. Transmit a program message as shown in the usage example.

:SOURce:TELecom:BRATe

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>

M9953	9953Mbit/s
M2488	2488Mbit/s

-Function Sets the bit rate of TX signal.

-Affected item

Screen (when Tx&Rx) Bit rate
(when Tx/Rx) Tx Bit Rate.

-Example use Sets the bit rate of Tx signal to 9953 Mbit/s.
> :SOURce:TELecom:BRATe M9953

Omitted for the following.

10.3.2 Measurement

<Example> Performing a manual measurement.

The command to start measurement is :SENSe:MEASure:STARt.

(→ Refer to "Commands related to MP1580A" in the next section.)

This command provides the same function as pressing [• Start /Stop] to start measurement in manual operation.

To stop the measurement or to check that the measurement is completed, transmit to execute the following commands:

Event	Applicable command
Stops measurement.	:SENSe:MEASure:STOP
Checks the measurement completion.	:SENSe:MEASure:STATe?

10.3.3 Reading measured result

All the measured result displayed on the Result screen can be read by adding a parameter associated with the result to be read next to :CALC:DATA? (See the Result screen in the command details.)

<Example> Read an amplitude value (Peak to Peak) that is the result of manual measurement.

From the separate manual of Vol. 1 (Jitter and Wander measurement), it is found that the measured result is displayed on the Result: Manual screen (Unit: Peak/RMS).

1) See the paragraph on Result: Manual screen of the command details.
(→ See "11.3.2 Result: Manual")

2) See the command list for the item to be set.
Search for the desired item from the "Item" row.
The list provides :CALC:DATA? "JAMPlitude:PTPeak".

3) The program message to be transmitted is as follows:
:CALC:DATA? "JAMPlitude:PTPeak"

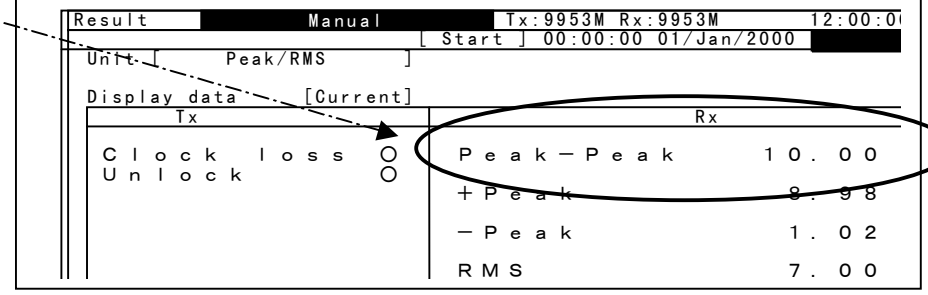
4) When the program message is transmitted, a response message is returned.

(Example) > :CALCulate:DATA? "JAMPlitude:PTPeak"
< " 0.234"
(For response message formats, see Section 9 SCPI Overview.)

11.3.2 Manual sub-screen

Unit: Peak/RMS display screen

[(1)]
Here in the screen an amplitude (Peak to Peak) is displayed.



[(2)]
The command is :CALCulate:DATA? and the parameter is "JAMPlitude:PTPeak".

[(2)]
Command list corresponding to (1). Search for an amplitude (Peak to Peak) here.

Item	Command	Parameter
Tx Jitter Clock loss	:CALCulate:DATA?	"<string> JMANual:TCLock"
Tx Jitter Unlock	:CALCulate:DATA?	"<string> JMANual:TSTatus"
Rx Jitter Peak-Peak	:CALCulate:DATA?	"<string> JAMPlitude:PTPeak"
Rx Jitter +Peak	:CALCulate:DATA?	"<string> JAMPlitude:PPEak"
Rx Jitter -Peak	:CALCulate:DATA?	"<string> JAMPlitude:MPEak"
Rx Jitter RMS	:CALCulate:DATA?	"<string> JAMPlitude:RMS"
Rx Jitter Unlock	:CALCulate:DATA?	"<string> JMANual:RSTatus"

Omitted along the way.

[(3)]
Command details.
Transmit a program message as shown in "Example use."

:CALCulate:DATA?

▽ Parameter <string> = <STRING PROGRAM DATA>
"[CURRent:]<result>" Current measured results
"LAST:<result>" Last measured result

See the separate table on the next page for the <result> contents.

- Response <string> = <STRING RESPONSE DATA>
See the separate table on the next page for the response format.
* When Test menu screen state differs to the specified parameter; the format (for which the corresponding response data to the query of the specified parameter does not exist) is output.
- Function Queries the measured result corresponding to the parameter.
Even if the Result screen is not displayed, the objective measured results can be queried.
- Referencee item screen Refer to the measured result display field on each the Result screen.
- Example use Queries the measured results of Jitter transfer measurement.
> :CALCulate:DATA? "CURRent:JTRansfer:POINt1"
or
> :CALCulate:DATA? "JTRansfer:POINt1"
< "100, 10.00, 0.00, Acceptable"

Omitted for the following.

10.3.4 Saving data

<Example> Saving the Peak jitter data obtained in manual measurement.

1) Assume the screen to be set.
   According to the separate operation manual of Vol. 1 (Jitter and Wander Measurement), it is found that the Peak jitter data can be saved by executing the Store function on the Analyze: Peak jitter screen.

2) See the paragraph on Analyze: Peak jitter screen of the command details.
   (→ See the Analyze: Peak jitter screen)

3) See the command list for the item to be set.
   Referring to the screen, it is found that the command to be saved is the [Store] key.
   The corresponding command is :SYSTem:MEMory:ANALysis:STORe.

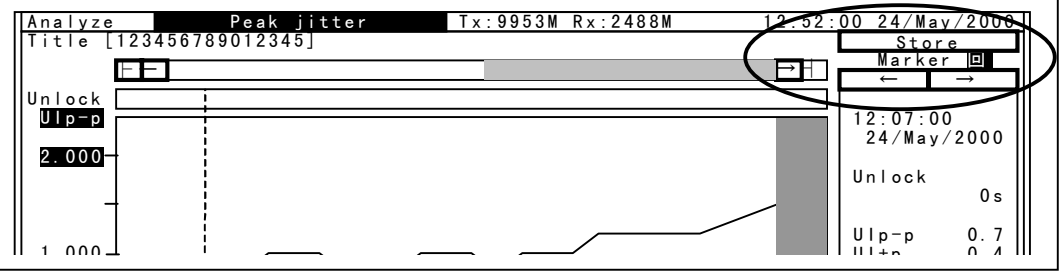
4) See command details shown on the reference page to determine the parameter to be set.

5) The program message to be transmitted is as follows:

   :SYSTem:MEMory:ANALysis:STORe PEAKjitter,"Demo1"

[(1)]
On the screen data is saved with the [Store] key.

11.4.3 Peak jitter sub-screen



[(2)]
The command is :SYSTem:MEMory:ANALysis:STORe.

[(2)]
Command list corresponding to (1). Search the [Store] key.

Setting item	Command
(Sub-screen)	:DISPlay:ANALysis[:NAME]
(Store key) Graph (Store key) Name	:SYSTem:MEMory:ANALysis:STORe
Title	:DISPlay:ANALysis:PEAK:TITLe

Omitted along the way.

[(3)]
Command details. Select a parameter. Send a program message as shown in "Example use."

:SYSTem:MEMory:ANALysis:STORe Setting command

∇ Parameter

<type> = <CHARACTER PROGRAM DATA>

~~LOGging~~	Logging screen
PEAKjitter	Peak jitter data
~~JTRansfer~~	Jitter transfer characteristics
WANDer	Wander data
FREQuency	Frequency graph data

<string> = <STRING PROGRAM DATA>

"ABCabc..." Data name (Max. 8 characters, excluding " " double quotation marks.)

- Function Writes the analysis data to memory.

- Affected item

Screen Store key, Graph, Name

- Example use Write the Analysis data (jitter transfer characteristics) to a memory named "Demo1."

> :SYSTem:MEMory:ANALysis:STORe JTRansfer,"Demo1"

Omitted for the following.

10.4 Details of Device Unique command

Details of device unique command are described here.

Note:

- Commands are divided and explained in accordance with MP1580A screens. See the command list in the appendix for commands categorized by subsystem.
- The device unique command supported by MP1580A is a sequential command.
- When a set value of the other item becomes inadmissible due to the setting of a program command, it is changed to an admissible value.
- When the setting condition is changed during measurement, the measurement is restarted.
- For the conditions for changing set values of other items or restarting measurement, refer to operating methods described in the separate operation manual of Vol. 1.

<*Configuration*>
Device message details described in the next section are configured as follows:

- MP1580A screens (Setup, Test menu, Result, Analyze screens)
- Command list associated with screens
- Command details shown in the list

(Execution example of commands → See "10.3 Referring to Command Details".)

<Description example of command details>

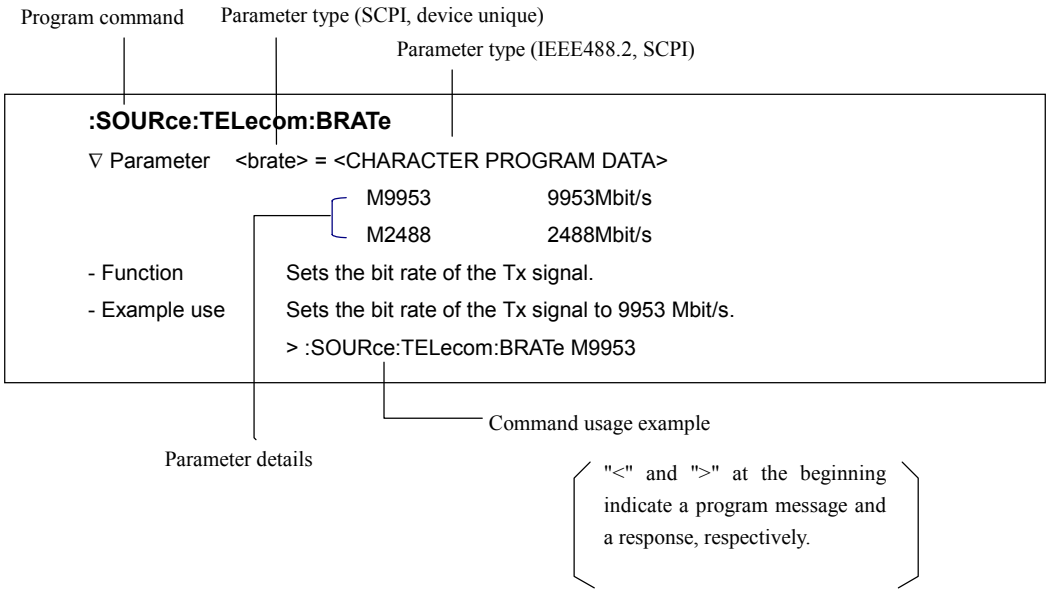
Command details are described as follows:

<Program command>

When restrictions are imposed for using a command, there is an item called "Restriction" along with the example below.

The usage example is described as follows:

> *program message (program command)*



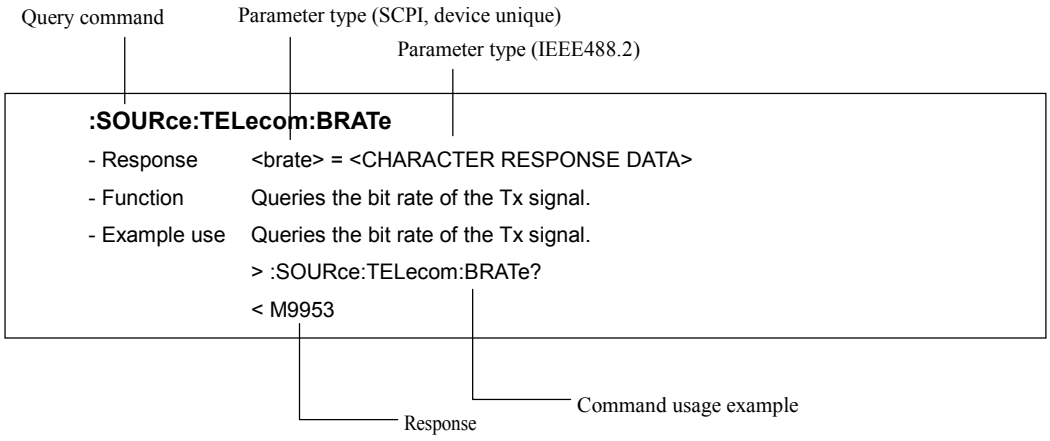
<Program query command>

When a program command corresponding to a query exists, response details are omitted (the parameter details of the program command represent the response).

The usage example is described as follows:

> *program message (query command)*

< *response*



Section 11 Device Message Details

This section describes the details of the device messages of the MP1580A, depending on the screen composition.

[Setup main screen]

Describes commands used on the Setup screen, depending on the Setup screen composition.

[Test menu main screen]

Describes commands used on the Test menu screen, depending on the Test menu screen composition.

[Result main screen]

Describes commands used on the Result screen, depending on the Result screen composition.
The screen displayed on the Result screen depends on the selection of the Test menu main screen.

[Analyze main screen]

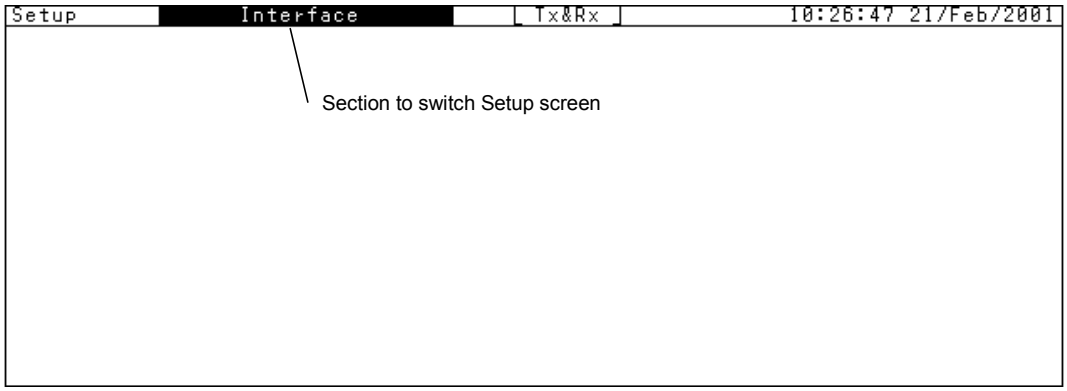
Describes commands used on the Analyze screen, depending on the Analyze screen composition.
The screen displayed on the Analyze screen depends on the selection of the Test menu main screen.

[Commands without related screen]

Describes commands for the equipment operation, without related display screen.
Commands for the MP1580A itself (mainly for panel operation)
Status register (STATus subsystem) commands

11.1 Setup Main Screen

11.1.1 Command commonly used on Setup screens



Setting Item	Command	Reference Page
Screen switching command commonly used on Setup screen	:DISPlay:SETup[:NAME]	11-5

:DISPlay:SETup[:NAME]

∇ Parameter <sdisplay> = <STRING PROGRAM DATA>

"IFACe"	The Interface screen
"JTRansfer"	The Jitter transfer screen
"JWANder"	The Jitter Wander screen
"PRINt"	The Print/Logging screen
"SYSTem"	The System screen
"MEMory"	The Memory screen
"FDISk"	The Floppy disk screen
"OREVision"	The Option/Revision screen

- Function Selects an item for display on the Setup screen.

- Example use When selecting the Memory screen as the item for display on the Setup screen.

```
> :DISPlay:SETup"MEMory"
        or
> :DISPlay:SETup:NAME"MEMory"
```

:DISPlay:SETup[:NAME]?

∇ Parameter None

- Response <sdisplay> = <STRING RESPONSE DATA>

"IFAC"	The Interface screen
"JTR"	The Jitter transfer screen
"JWAN"	The Jitter Wander screen
"PRIN"	The Print/Logging screen
"SYST"	The System screen
"MEM"	The Memory screen
"FDIS"	The Floppy screen
"OREV"	The Option/Revision screen

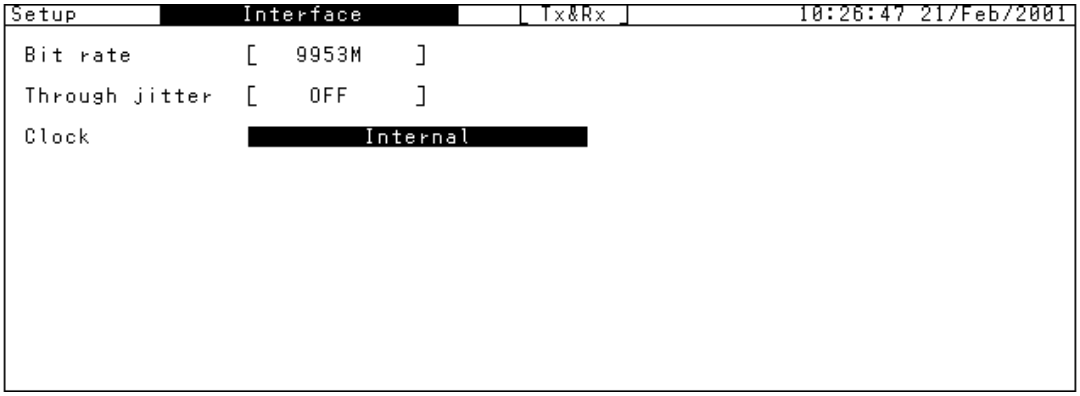
- Function Queries the item for display on the Setup screen.

- Example use

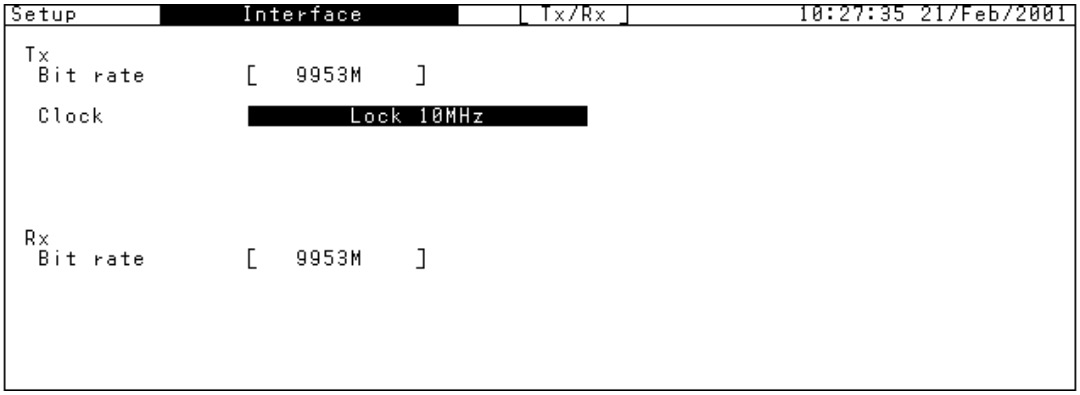
```
> :DISPlay:SETup?
        or
> :DISPlay:SETup:NAME?
< MEM
```

11.1.2 Interface sub-screen

* Tx&Rx



* Tx&Rx



Setting Item	Command	Reference Page
(Transmission/reception mode)	:INSTrument:COUPle	11-7

Setting Item	Command	Reference Page
Tx Bit Rate	:SOURce:TELecom:BRATe	11-8
Rx Bit Rate	:SENSe:TELecom:BRATe	11-9
Through jitter	:ROUTe:THRough:JITTer	11-10
Clock	:SOURce:TELecom:CLOCk:SOURce	11-11

:INSTrument:COUPle

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

ALL	Tx&Rx (Identical)
NONE	Tx/Rx (Separate)

- Function Specifies whether to make identical or separate setting for the transmitter and receiver.
  The commands shown below are set simultaneously for the transmitter and receiver when making identical setting.

  :SOURce:TELecom:BRATe
  :SOURce:JITTer:MANual:RANGe
  :SENSe:TELecom:BRATe
  :SENSe:JITTer:MANual:RANGe
- Example use When specifying the transmission and reception modes separately
  > :INSTrument:COUPle NONE

:INSTrument:COUPle?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>
  Same as in the case of :INSTrument:COUPle.
- Function Queries whether transmission and reception mode settings are identical or separate.
- Example use > :INSTrument:COUPle?
  < NONE

:SOURce:TELecom:BRATe

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>

M9953	9953 Mbit/s
M2488	2488 Mbit/s

- Function Specifies the bit rate of the transmission signal.

- Example use When setting the bit rate of the transmission signal at 9953 Mbit/s.
  > :SOURce:TELecom:BRATe M9953

:SOURce:TELecom:BRATe?

∇ Parameter None

- Response <brate> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:TELecom:BRATe.

- Function Queries the bit rate of the transmission signal.

- Example use When querying the setting contents of the reception signal's bit rate.
  > :SOURce:TELecom:BRATe?
  < M9953

:SENSe:TELecom:BRATe

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>

M9953	9953 Mbit/s
M2488	2488 Mbit/s

- Function Specifies the bit rate of the reception signal.

- Example use When setting the bit rate of the reception signal at 9953 Mbit/s.
  > :SENSe:TELecom:BRATe M9953

:SENSe:TELecom:BRATe?

∇ Parameter None

- Response <brate> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:TELecom:BRATe.

- Function Queries the bit rate of the reception signal.

- Example use When querying the setting contents of the reception signal's bit rate.
  > :SENSe:TELecom:BRATe?
  < M9953

:ROUTe:THRough:JITTer

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets the Jitter adding function to OFF
ON or 1	Sets the Jitter adding function to ON

- Function Specifies whether to switch ON or OFF the function to add Jitter to the recoveried Clock.

- Example use When switching ON the function to add Jitter to the Clock.
  > :ROUTe:THRough:JITTer ON

:ROUTe:THRough:JITTer?

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Jitter adding function is set to OFF
1	Jitter adding function is set to ON

- Function Queries whether the function to add Jitter to the recoveried Clock is switched ON or OFF.

- Example use > :ROUTe:THRough:JITTer?
  < 1

:SOURce:TELecom:CLOCk:SOURce

∇ Parameter <csource> = <CHARACTER PROGRAM DATA>

INTernal	Internal clock
L10M	Lock 10M
L5M	Lock 5M
LUNB_2MHZ	Lock 2 MHz (Unbalanced)
LBAL_2MHZ	Lock 2 MHz (Balanced)
LUNB_2MBPS	Lock 2 Mbit/s (Unbalanced)
LBAL_2MBPS	Lock 2 Mbit/s (Balanced)
LBAL_1_5MHZ	Lock 1.5 MHz (Balanced)
LBAL_1_5MBPS	Lock 1.5 Mbit/s (Balanced)
LBAL_64 K	Lock 64 k+8 kHz
EXTernal	External signal source

- Function Specifies the clock source for the transmission signal.
  - Internal : Synchronous with the internal clock
  - Lock*** : Synchronous with the external DCS clock
  - External: Synchronous with the external signal source
    (The selection of External disables Jitter transfer and Wander measurements.)

- Example use When specifying the bit rate of the transmission signal.
  > :SOURce:TELecom:CLOCk:SOURce EXTernal

:SOURce:TELecom:CLOCk:SOURce?

∇ Parameter None

- Response <csource> = <CHARACTER RESPONSE DATA>

INT	Internal clock
L10M	Lock 10M
L5M	Lock 5M
LUNB_2MHZ	Lock 2 MHz (Unbalanced)
LBAL_2MHZ	Lock 2 MHz (Balanced)
LUNB_2MBPS	Lock 2 Mbit/s (Unbalanced)
LBAL_2MBPS	Lock 2 Mbit/s (Balanced)
LBAL_1_5MHZ	Lock 1.5 MHz (Balanced)
LBAL_1_5MBPS	Lock 1.5 Mbit/s (Balanced)
LBAL_64 K	Lock 64 k+8 kHz
EXT	External signal source

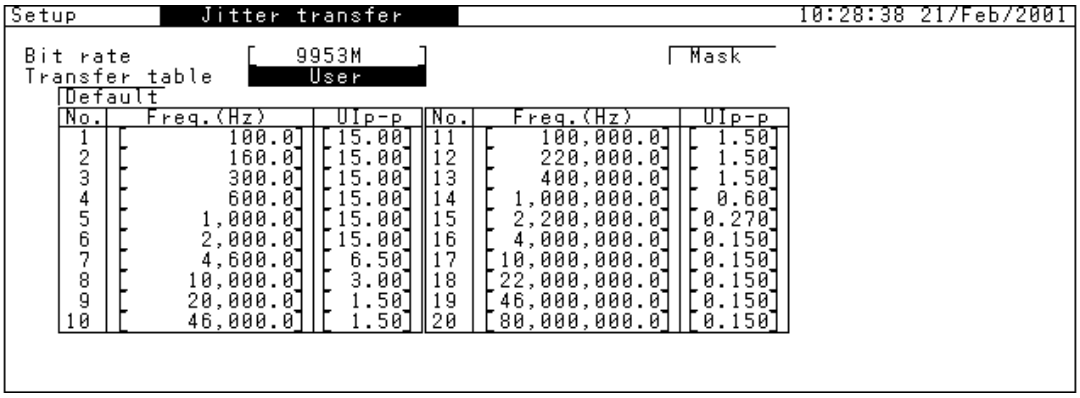
- Function Queries the setting status of the transmission signal's clock source.

- Example use When querying the setting contents of the transmission signal's bit rate.
  > :SOURce:TELecom:CLOCk:SOURce
  < EXT

11.1.3 Jitter transfer sub-screen

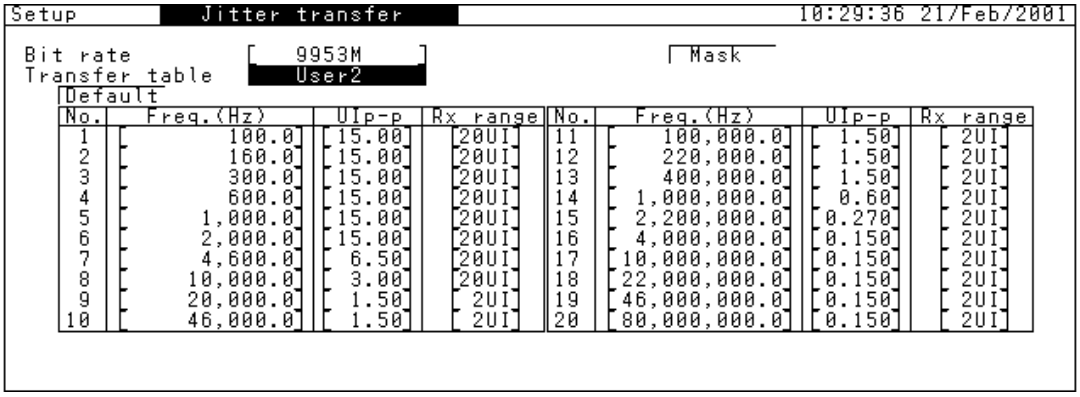
* When Table screen is displayed

Transfer table: User



No.	Freq.(Hz)	UIp-p	No.	Freq.(Hz)	UIp-p
1	100.0	15.00	11	100,000.0	1.50
2	160.0	15.00	12	220,000.0	1.50
3	300.0	15.00	13	400,000.0	1.50
4	600.0	15.00	14	1,000,000.0	0.60
5	1,000.0	15.00	15	2,200,000.0	0.270
6	2,000.0	15.00	16	4,000,000.0	0.150
7	4,600.0	6.50	17	10,000,000.0	0.150
8	10,000.0	3.00	18	22,000,000.0	0.150
9	20,000.0	1.50	19	46,000,000.0	0.150
10	46,000.0	1.50	20	80,000,000.0	0.150

Transfer table: User 2

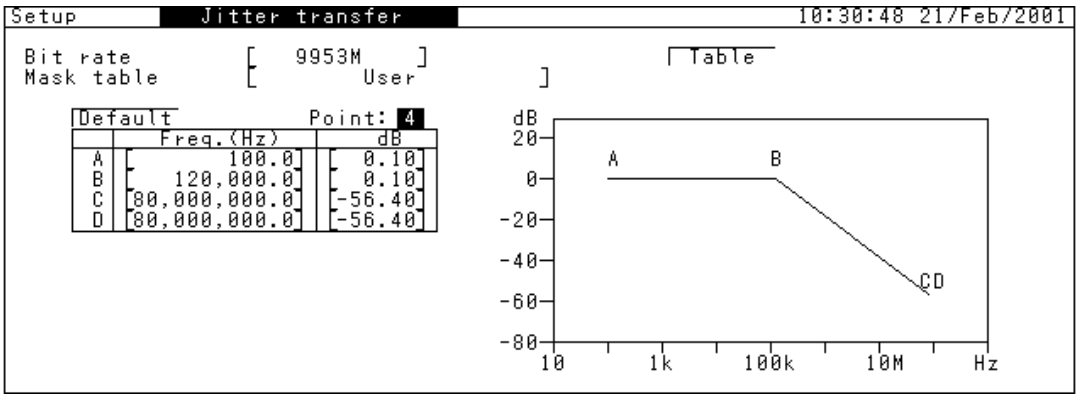


No.	Freq.(Hz)	UIp-p	Rx range	No.	Freq.(Hz)	UIp-p	Rx range
1	100.0	15.00	20UI	11	100,000.0	1.50	2UI
2	160.0	15.00	20UI	12	220,000.0	1.50	2UI
3	300.0	15.00	20UI	13	400,000.0	1.50	2UI
4	600.0	15.00	20UI	14	1,000,000.0	0.60	2UI
5	1,000.0	15.00	20UI	15	2,200,000.0	0.270	2UI
6	2,000.0	15.00	20UI	16	4,000,000.0	0.150	2UI
7	4,600.0	6.50	20UI	17	10,000,000.0	0.150	2UI
8	10,000.0	3.00	20UI	18	22,000,000.0	0.150	2UI
9	20,000.0	1.50	2UI	19	46,000,000.0	0.150	2UI
10	46,000.0	1.50	2UI	20	80,000,000.0	0.150	2UI

Setting Item	Command	Reference Page
Key to switch between the Table or Mask display	:DISPlay:SETup:JTRansfer:CONDition	11-14

Setting Item	Command	Reference Page
Bit Rate Transfer Table	:SOURce:JITTer:TRANsfer:PTABle:TYPE	11-15
Setting for Transfer table: User Bit rate Setting for Transfer table: User No. Setting for Transfer table: User Freq. (Numerical value) Setting for Transfer table: User Freq. (Unit) Setting for Transfer table: User UIp-p	:SOURce:JITTer:TRANsfer:PTABle:DATA	11-16
Setting for Transfer table: User Default (Table initialization) [Transfer=User]-[Default]-[<type>]-[set]	:SOURce:JITTer:TRANsfer:PTABle:DEFault	11-18
Setting for Transfer table: User 2 Bit rate Setting for Transfer table: User 2 No. Setting for Transfer table: User 2 Freq. (Numerical value) Setting for Transfer table: User 2 Freq. (Unit) Setting for Transfer table: User 2 UIp-p Setting for Transfer table: User 2 Rx range	:SOURce:JITTer:TRANsfer:PTABle:DATA2	11-19
Setting for Transfer table: User 2 Default (Table initialization) [Transfer=User2]-[Default]-[<type>]-[set]	:SOURce:JITTer:TRANsfer:PTABle:DEFault2	11-21

* When Mask is displayed



Setting Item	Command	Reference Page
Key to switch between the Table or Mask display	:DISPlay:SETup:JTRansfer:CONDition	11-14

The command above is the same as in the case of the Transfer table display.

Setting Item	Command	Reference Page
Brate Mask table	:SENSe:JITTer:TRANsfer:MTABle:TYPE	11-22
Default (Initialization of the Mask table)	:SENSe:JITTer:TRANsfer:MTABle:DEFault	11-23
Point (Number of points when Mask table: User)	:SENSe:JITTer:TRANsfer:MTABle:POINt	11-24
Setting for Mask table: User Bit rate Setting for Mask table: User Freq. (A to D) (Numerical value) Setting for Mask table: User Freq. (A to D) (Unit) Setting for Mask table: User dB (A to D)	:SENSe:JITTer:TRANsfer:MTABle:DATA	11-25

:DISPlay:SETup:JTRansfer:CONDition

∇ Parameter <type> = <CHARACTER PROGRAM DATA>
TABLe Table
MASK Mask

- Function Switches between the Table and Mask data display during jitter transfer characteristics measurement.

- Example use When specifying Table data as jitter transfer characteristic data.
> :DISPlay:SETup:JTRansfer:CONDition TABle

:DISPlay:SETup:JTRansfer:CONDition?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
TABL Table
MASK Mask

- Function Queries the display status of jitter transfer characteristic data (Table/Mask).

- Example use When querying the type of data display for jitter transfer measurement currently set.
> :DISPlay:SETup:JTRansfer:CONDition
< TAB

* The commands for Table setting are shown below.

:SOURce:JITTer:TRANsfer:PTABle:TYPE

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953
M2488

<type> = <CHARACTER PROGRAM DATA>

G958A	G.958 Type A
G958B	G.958 Type B
G825_2M	G.825 2M
G825_1_5M	G.825 1.5M
G813	G.813
B1377	Bell 1377
B253	Bell 253
USER	User
USER2	User2

- Function Specifies the edit table for jitter occurrence during jitter transfer characteristics measurement.

- Example use When setting the edit table for jitter transfer characteristics measurement, M2488, at G958 Type A.
> :SOURce:JITTer:TRANsfer:PTABle:TYPE M2488,G958A

:SOURce:JITTer:TRANsfer:PTABle:TYPE?

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:TYPE.

- Response <type> = <CHARACTER RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:TYPE.

- Function Queries the edit table's contents for jitter occurrence during jitter transfer characteristics measurement

- Example use When querying the edit table for jitter transfer characteristics measurement, 2488M.
> :SOURce:JITTer:TRANsfer:PTABle:TYPE? M2488
< G958A

:SOURce:JITTer:TRANsfer:PTABle:DATA

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 20 Step value: 1

<freq1> = <DECIMAL NUMERIC PROGRAM DATA>
1.0 to 990.0 Step value: 0.1

<freq2> = <CHARACTER PROGRAM DATA>
HZ Unit for <freq1>
KHZ
MHZ

The setting range is between 100.0 Hz and 80 MHz when the bit rate is set at 9953M.

The setting range is between 100.0 Hz and 20 MHz when the bit rate is set at 2488M.

Freq. setting range (Significant digit -- first two digits)

Freq.	Unit	Step
100.0 to 990.0	Hz	10
1.0 to 9.9	kHz	0.1
10.0 to 99.0		1
100.0 to 990.0		10
1.0 to 9.9	MHz	0.1
10.0 to 80.0 (9953M)		1
10.0 to 20.0 (2488M)		1

<amp1> = <DECIMAL NUMERIC PROGRAM DATA>
0.051 to 16.00 Step value: As per the table below

Ampl. Setting Range based on Freq

Bit rate	Freq (Hz)	Setting Range	Details	Step
9953M	100 to 99 k	0.10 to 16.00	0.10 to 16.00	0.05
	100 k to 2.0 M	0.051 to 80.80 × 10^(-1 × log ((Freq)/100000)) *1	0.051 to 0.500	0.001
			0.55 to 16.00	0.05
	2.1 M to 80 M	0.051 to 0.500	0.051 to 0.500	0.001
2488M	100 to 99 k	0.06 to 16.00	0.06 to 16.00	0.01
	100 k to 2.0 M	0.051 to 20.20 × 10^ (-1 × log ((Freq)/100000)) *1	0.051 to 0.500	0.001
			0.51 to 16.00	0.01
	2.1 M to 20 M	0.051 to 0.500	0.051 to 0.500	0.001

Note 1: Set to 16.00 when it is larger than 16.00.

- Function Specifies the contents of the User table for jitter occurrence during jitter transfer characteristics measurement.
  * On the screen, the value of the Freq. is displayed in "Hz" unit when it is set in "KHz" or "MHz" unit.
- Example use When setting 100.0 Hz, 0.051 and 20 UI for No.1 of the 2488M table.
  > :SOURce:JITTer:TRANsfer:PTABle:DATA M2488,1,100.0,HZ,0.051

:SOURce:JITTer:TRANsfer:PTABle:DATA?

∇ Parameter

<brate> = <CHARACTER PROGRAM DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA.

<point> = <DECIMAL NUMERIC PROGRAM DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA.

- Response

<freq1> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA.

<freq2> = <CHARACTER RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA.

<amp1> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA.

- Function

Queries the contents of the User setting table for jitter occurrence during jitter transfer characteristics measurement.

- Example use

Queries the contents of the No.1 User table of 2488M.
> :SOURce:JITTer:TRANsfer:PTABle:DATA? M2488,1
< 100.0,HZ,0.051

:SOURce:JITTer:TRANsfer:PTABle:DEFault

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<type> = <CHARACTER PROGRAM DATA>

G958A	G.958 Type A
G958B	G.958 Type B
G825_2M	G.825 2M
G825_1_5M	G.825 1.5M
G813	G.813
B1377	Bell 1377
B253	Bell 253

Default (initialization) operation setting for Transfer Tables
(○: Setting enabled, -: Setting disabled)

Transfer Table	Transfer (Table) Bit rate	
	9953M	2488M
G.958 Type A	-	○
G.958 Type B	-	○
G.825 2M	○	○
G.825 1.5M	○	○
G.813	-	○
Bell1377	○	-
Bell253	-	○

- Function Initializes the measurement table (User table) contents for jitter transfer characteristics measurement.

- Example use When initializing the contents of 2488M table of G.958 Type A.
> :SOURce:JITTer:TRANsfer:PTABle:DEFault M2488,G958A

:SOURce:JITTer:TRANsfer:PTABle:DATA2

∇ Parameter

<brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<point> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 20 Step value: 1

<freq1> = <DECIMAL NUMERIC PROGRAM DATA>
1.0 to 990.0 Step value: 0.1
* See :SOURce:JITTer:TRANsfer:PTABle:DATA for more information.
<freq2> = <CHARACTER PROGRAM DATA>
HZ Unit for <freq1>
KHZ
MHZ

<ampl> = <DECIMAL NUMERIC PROGRAM DATA>
0.050 to 16.000 Step value: 0.001
* Set to 20 UI when <ampl> is set at 1.71 or larger.

<rxrange> = <CHARACTER PROGRAM DATA>

UI20 20 UI
UI2 2 UI
* Set to 20 UI when <ampl> is set at 1.71 or larger.

- Function Specifies the contents of the User table for jitter occurrence during jitter transfer characteristics measurement, when the transfer table is set to User 2.

- Example use When setting 100.0 Hz, 0.051 and 20 UI in No.1 of the 2488M table.
> :SOURce:JITTer:TRANsfer:PTABle:DATA2 M2488,1,100.0,HZ,0.051,UI20

:SOURce:JITTer:TRANsfer:PTABle:DATA2?

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA2.

<point> = <DECIMAL NUMERIC PROGRAM DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA2.

- Response <freq1> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA2.

<freq2> = <CHARACTER RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA2.

<ampl> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA2.

<rxrange> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:DATA2.

- Function Queries the contents of the User setting table for jitter occurrence during jitter transfer characteristics measurement, when the transfer table is set to User 2.

- Example use Queries the frequency and rxrange set in No.1 of the 2488M table.
> :SOURce:JITTer:TRANsfer:PTABle:DATA? M2488,1
< 100.0,HZ,0.051

:SOURce:JITTer:TRANsfer:PTABle:DEFault2

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<type> = <CHARACTER PROGRAM DATA>

G958A	G.958 Type A
G958B	G.958 Type B
G825_2M	G.825 2M
G825_1_5M	G.825 1.5M
G813	G.813
B1377	Bell 1377
B253	Bell 253

* See :SOURce:JITTer:TRANsfer:PTABle:DEFault for more information.

- Function Initializes the measurement table (User 2 table) contents for jitter transfer characteristics measurement.

- Example use When initializing the contents of the table 2488M of G.958 Type A.
> :SOURce:JITTer:TRANsfer:PTABle:DEFault2 M2488,G958A

* The commands for Mask setting are shown below.

:SENSe:JITTer:TRANsfer:MTABle:TYPE

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<type> = <CHARACTER PROGRAM DATA>

G958A	G.958 Type A
G958B	G.958 Type B
ANSIT1A	ANSI T1.105.03 TYPE A
ANSIT1B	ANSI T1.105.03 TYPE B
B1377	Bell 1377
B253	Bell253
USER	

Transfer Mask setting for Bit rates (○: Setting enabled, -: Setting disabled)

Transfer Table	Transfer (Mask) Bit rate	
	9953M	2488M
G.958 Type A	-	○
G.958 Type B	-	○
ANSI T1.105.03 Type A	-	○
ANSI T1.105.03 Type B	-	○
Bell1377	○	-
Bell253	-	○
User	○	○

- Function Specifies the mask edit table for jitter transfer characteristics measurement.

- Example use When setting the mask edit table, M2488, to G.958 Type A.
> :SENSe:JITTer:TRANsfer:MTABle:TYPE M2488,G958A

:SENSe:JITTer:TRANsfer:MTABle:TYPE?

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:TYPE.

- Response <type> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:TYPE.

- Function Queries the mask edit table for jitter transfer characteristics measurement.

- Example use When querying the type set in the mask edit table, 2488M.
> :SENSe:JITTer:TRANsfer:MTABle:TYPE? M2488
< G958A

:SENSe:JITTer:TRANsfer:MTABle:DEFault

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<type> = <CHARACTER PROGRAM DATA>

G958A	G.958 Type A
G958B	G.958 Type B
ANSIT1A	ANSI T1.105.03 Type A
ANSIT1B	ANSI T1.105.03 Type B
B1377	Bell 1377
B253	Bell 253

Transfer Mask setting for Bit rates (○: Setting enabled, -: Setting disabled)

Transfer Table	Transfer (Mask) Bit rate	
	9953M	2488M
G.958 Type A	-	○
G.958 Type B	-	○
ANSI T1.105.03 Type A	-	○
ANSI T1.105.03 Type B	-	○
Bell1377	○	-
Bell253	-	○

- Function Initializes the contents of the mask table for jitter transfer characteristics measurement.

- Example use When initializing the contents of the 2488M User mask table to Transfer Mask Bell 253.
> :SENSe:JITTer:TRANsfer:MTABle:DEFault M2488,B253

:SENSe:JITTer:TRANsfer:MTABle:POINt

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<numeric> = <DECIMAL NUMERIC PROGRAM DATA>
2 to 4 Step value: 1

- Function Specifies the range of mask table output points for jitter transfer measurement.

- Example use When setting the range of output points for the 2488M table to 2.
> :SENSe:JITTer:TRANsfer:MTABle:POINt M2488,2

:SENSe:JITTer:TRANsfer:MTABle:POINt?

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:POINt.

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:POINt.

- Function Queries the range of mask table output points for jitter transfer measurement.

- Example use Queries the range of output points for the 2488M table.
> :SENSe:JITTer:TRANsfer:MTABle:POINt? M2488
< 2

:SENSe:JITTer:TRANsfer:MTABle:DATA

∇ Parameter <brate> = <CHARACTER PROGRAM DATA>
M9953, M2488

<point> = <CHARACTER PROGRAM DATA>
A, B, C, D

<freq1> = <DECIMAL NUMERIC PROGRAM DATA>
1.0 to 990.0 Step value: 0.1

Freq. setting range

Freq.	Unit	Step
100.0 to 990.0	Hz	10
1.0 to 9.9	kHz	0.1
10.0 to 99.0		1
100.0 to 990.0		10
1.0 to 9.9	MHz	0.1
10.0 to 80.0 (9953M)		1
10.0 to 20.0 (2488M)		1

<freq2> = <CHARACTER PROGRAM DATA>
HZ Unit for <freq1>
KHZ
MHZ

<ampl> = <DECIMAL NUMERIC PROGRAM DATA>
–80.00 to 10.00 Step value: 0.01

* When the set value for <freq1> exceeds the range, the device changes this value to one within the allowable range. (Digits following significant digits are truncated.)

- Function Specifies the contents of the User table for jitter occurrence during jitter transfer characteristics measurement, when the transfer table is set to User.

- Example use When setting 100.0 Hz and -3.00 dB for A of the 2488M mask table.
> :SENSe:JITTer:TRANsfer:MTABle:DATA M2488,A,100.0,HZ,-3.00

:SENSe:JITTer:TRANsfer:MTABle:DATA?

∇ Parameter

<brate> = <CHARACTER PROGRAM DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:DATA.

<point> = <NR1 NUMERIC PROGRAM DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:DATA.

- Response

<freq1> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:DATA.

<freq2> = <CHARACTER RESPONSE DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:DATA.

<ampl> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:MTABle:DATA.

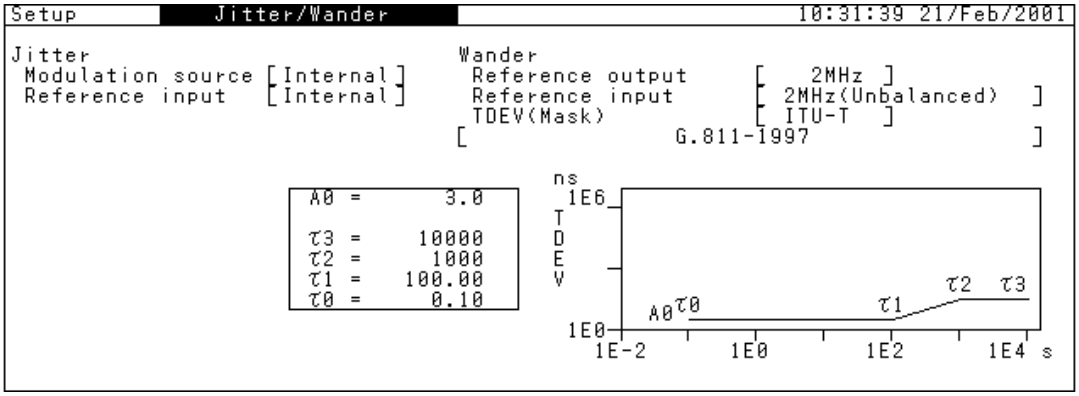
- Function

Queries the contents of the User table for jitter occurrence during jitter transfer characteristics measurement, when the transfer table is set to User.

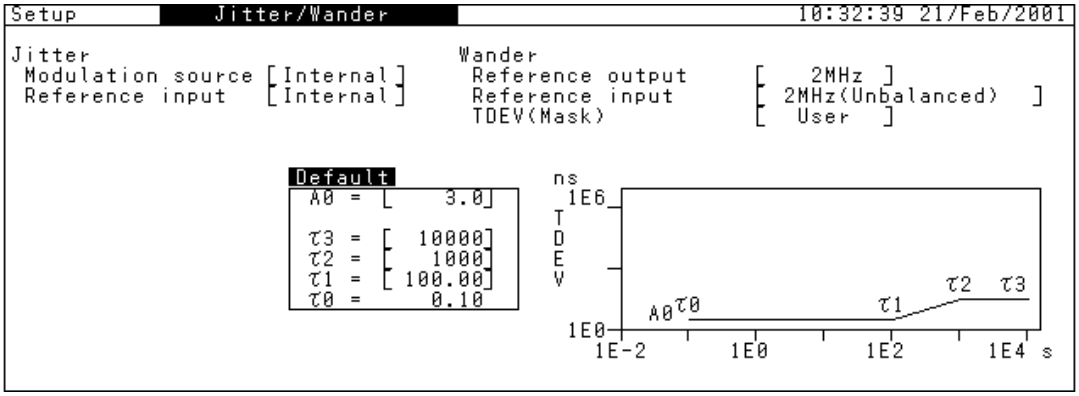
- Example use

When querying the setting contents for A of the 2488M mask table.
> :SENSe:JITTer:TRANsfer:MTABle:DATA? M2488,A
< 100.0,HZ,-3.00

11.1.4 Jitter/Wander sub-screen



The following screen appears when TDEV (Mask): User.



Item	Command	Reference Page
Jitter modulation source	:SOURce:JITTer:MANual:MODE	11-28
Jitter reference input	:SENSe:MEASure:JWANder:JITTer:MODE	11-29
Wander reference output	:SOURce:WANDer:MANual:MODE	11-30
Wander reference input	:SENSe:MEASure:JWANder:WANDer:MODE	11-31
Wander TDEV (Mask) (type)	:SOURce:JITTer:MANual:TDEV:DTYPe	11-32
Wander TDEV (Mask) (table)	:SOURce:JITTer:MANual:TDEV:ITYPe	11-33
Wander TDEV (Mask) (table)	:SOURce:JITTer:MANual:TDEV:ETYPe	11-34
Wander TDEV (Mask) (table)	:SOURce:JITTer:MANual:TDEV:ATYPe	11-35
Wander TDEV (Mask) (table)	:SOURce:JITTer:MANual:TDEV:BTYPe	11-36
A0 setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:A0	11-37
τ4 setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:VAT4?	11-38
τ3 setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:VAT3	11-39
τ2 setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:VAT2	11-40
τ1 setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:VAT1	11-41
τ0 setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:VAT0?	11-42
Default setting for Wander TDEV (Mask): User	:SOURce:JITTer:MANual:TDEV:DEFault	11-45

:SOURce:JITTer:MANual:MODE

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

INTernal	Internal signal
EXTernal	External signal

- Function Selects the source for generating the jitter modulating signal from internal or external one.

- Example use When setting the source for generating the jitter modulating signal at INTernal.
  > :SOURce:JITTer:MANual:MODE INTernal

:SOURce:JITTer:MANual:MODE?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>

INT	Internal signal
EXT	External signal

- Function Queries the status of the source for generating the jitter modulating signal.

- Example use > :SOURce:JITTer:MANual:MODE?
  < INT

:SENSe:MEASure:JWANder:JITTer:MODE

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

INTernal	Internal signal
EXTernal	External signal

- Function Specifies the input destination of the reference clock for jitter measurement.

- Example use When setting the input destination of reference clock to INTernal.
  > :SENSe:MEASure:JWANder:JITTer:MODE INT

:SENSe:MEASure:JWANder:JITTer:MODE?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>

INT	Internal signal
EXT	External signal

- Function Queries the setting status of the input destination of the reference clock for jitter measurement.

- Example use > :SENSe:MEASure:JWANder:JITTer:MODE?
  < INT

:SOURce:WANDer:MANual:MODE

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

M5	5 MHz
M2	2 MHz
M1_5	1.5 MHz

- Function Selects the output value of the reference clock generated by MP1580A unit for Wander measurement.

- Example use Specifies the wander reference output.
  > :SOURce:WANDer:MANual:MODE M2

:SOURce:WANDer:MANual:MODE?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:MANual:MODE.

- Function Queries the WANDer reference output.

- Example use Queries the wander reference output.
  > :SOURce:WANDer:MANual:MODE?
  < M2

:SENSe:MEASure:JWANder:WANDer:MODE

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

H10M	10 MHz
H5M	5 MHz
UNB_2MHZ	2 MHz (Unbalanced)
BAL_2MHZ	2 MHz (Balanced)
UNB_2MBPS	2 Mbit/s (Unbalanced)
BAL_2MBPS	2 Mbit/s (Balanced)
UNB_1_5MHZ	1.5 MHz (Unbalanced)
BAL_1_5MBPS	1.5 Mbit/s (Balanced)
H64_8	64 k+8 kHz

- Function Specifies the input value of the reference clock for Wander measurement.

- Example use When setting the reference clock for Wander measurement to 5 MHz signal.
  > :SENSe:MEASure:JWANder:WANDer:MODE H5M

:SENSe:MEASure:JWANder:WANDer:MODE?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:MEASure:JWANder:WANDer:MODE CLOCK.

- Function Queries the setting status of the input destination of the reference clock for Wander measurement.

- Example use > :SENSe:MEASure:JWANder:WANDer:MODE?
  < H5M

:SOURce:JITTer:MANual:TDEV:DTYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

ITUT	ITU-T
ETSI	ETSI
ANSI	ANSI
BELLcore	Bellcore
USER	User

- Function Specifies the TDEV Mask standard for manual jitter measurement.

- Example use When specifying ITU-T as the TDEV Mask standard for manual jitter measurement.
  > :SOURce:JITTer:MANual:TDEV:DTYPe ITUT

:SOURce:JITTer:MANual:TDEV:DTYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:TDEV:DTYPe.

- Function Queries the TDEV Mask standard for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:DTYPe?
  < ITUT

:SOURce:JITTer:MANual:TDEV:ITYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

G811	G.811-1997
S81T6	Section 8.1 Table 6 (G.812-1998)
S81T7	Section 8.1 Table 7 (G.812-1998)
S91T11	Section 9.1 Table 11 (G.812-1998)
S91T12	Section 9.1 Table 12 (G.812-1998)
SA31	Section A.3.1 (G.812-1998)
SA41	Section A.4.1 (G.812-1998)
SA5	Section A.5 (G.812-1998)
S71O1	Section 7.1 Option1 (G.813-1996)
S71O2	Section 7.1 Option2 (G.813-1996)

- Function Specifies the ITU-T TDEV Mask table for manual jitter measurement.

- Example use When specifying G.811-1997 as the ITU-T TDEV Mask table for manual jitter measurement.
  > :SOURce:JITTer:MANual:TDEV:ITYPe G811

:SOURce:JITTer:MANual:TDEV:ITYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:TDEV:ITYPe.

- Function Queries the ITU-T TDEV Mask table for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:ITYPe?
  < G811

:SOURce:JITTer:MANual:TDEV:ETYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

S721	Section 7.2.1 (ETS 300 462-3-1997)
S722	Section 7.2.2 (ETS 300 462-3-1997)
S723	Section 7.2.3 (ETS 300 462-3-1997)
S724	Section 7.2.4 (ETS 300 462-3-1997)
S61_4	Section 6.1 (ETS 300 462-4-1997)
S61_5	Section 6.1 (ETS 300 462-5-1996)
S72_4	Section 7.2 (ETS 300 462-4-1997)
S72_5	Section 7.2 (ETS 300 462-5-1996)
S8	Section 8 (ETS 300 462-4-1997)
ETS300_6	ETS 300 462-6-1997

- Function Specifies the ETSI TDEV Mask table for manual jitter measurement.

- Example use When specifying Section 7.2.1 (ETS 300 462-3-1997) as the ETSI TDEV Mask table for manual jitter measurement.
  > :SOURce:JITTer:MANual:TDEV:ETYPe S721

:SOURce:JITTer:MANual:TDEV:ETYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:TDEV:ETYPe.

- Function Queries the ETSI TDEV Mask table for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:ETYPe?
  < S721

:SOURce:JITTer:MANual:TDEV:ATYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

S722	Section 7.2.2 (ANSI TI.101-1994)
S732	Section 7.3.2 (ANSI TI.101-1994)
SD21	Section D.2.1 (ANSI TI.105.03-1994)
SD221	Section D.2.2.1 (ANSI TI.105.03-1994)
SD222	Section D.2.2.2 (ANSI TI.105.03-1994)
ANSIT1_9	ANSI T1.105.09-1996

- Function Specifies the ANSI TDEV Mask table for manual jitter measurement.

- Example use When specifying ANSI T1.105.09-1996 as the ANSI TDEV Mask table for manual jitter measurement.
  > :SOURce:JITTer:MANual:TDEV:ATYPe ANSIT1_9

:SOURce:JITTer:MANual:TDEV:ATYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:TDEV:ATYPe.

- Function Queries the ANSI TDEV Mask table for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:ATYPe?
  < ANSIT1_9

:SOURce:JITTer:MANual:TDEV:BTYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

GR2830	GR-2830-CORE-1995
S43	Section 4.3 (GR-1244-CORE-1995)
S53	Section 5.3 (GR-1244-CORE-1995)
S54S2	Section 5.4 Strarum 2&3E (GR-1244-CORE-1995)
S54S3	Section 5.4 Strarum 3 (GR-1244-CORE-1995)
S54424F515	Section 5.4.4.2.4 Figure 5-15 (GR-253-CORE-1999)
S54425F516	Section 5.4.4.2.4 Figure 5-16 (GR-253-CORE-1999)
S54432	Section 5.4.4.3.2 (GR-253-CORE-1999)
S545	Section 5.4.5 (GR-253-CORE-1999)

- Function Specifies the Bellcore TDEV Mask table for manual jitter measurement.

- Example use When specifying GR-2830-CORE-1995 as the Bellcore TDEV Mask table for manual jitter measurement.
  > :SOURce:JITTer:MANual:TDEV:BTYPe GR2830

:SOURce:JITTer:MANual:TDEV:BTYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:TDEV:BTYPe.

- Function Queries the Bellcore TDEV Mask table for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:BTYPe?
  < GR2830

:SOURce:JITTer:MANual:TDEV:A0

∇ Parameter <s> = <DECIMAL NUMERIC PROGRAM DATA>
1.0 to 200.0 Step value: 0.1
* The possible setting range varies depending on the mask standard.
For more information, see the table which comes after the detailed explanations on :SOURce:JITTer:MANual:TDEV:VAT0?.

- Function Specifies the TDEV A0 for manual jitter measurement.

- Example use When setting TDEV A0 for manual jitter measurement to 1.
> :SOURce:JITTer:MANual:TDEV:A0 1

:SOURce:JITTer:MANual:TDEV:A0?

∇ Parameter None

- Response <s> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:MANual:TDEV:A0.

- Function Queries the Bellcore TDEV Mask table for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:A0?
< 1

:SOURce:JITTer:MANual:TDEV:VAT4?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
  10000
- Function Queries TDEV τ4 for manual jitter measurement.
- Example use > :SOURce:JITTer:MANual:TDEV:VAT4?
  < 10000

:SOURce:JITTer:MANual:TDEV:VAT3

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
40 to 10000
* The possible setting range varies depending on the mask standard. For more information, see the table which comes after the detailed explanations on :SOURce:JITTer:MANual:TDEV:VAT0?.

- Function Specifies the TDEV τ3 for manual jitter measurement.

- Example use When setting TDEV τ3 for manual jitter measurement to 1000.
> :SOURce:JITTer:MANual:TDEV:VAT3 1000

:SOURce:JITTer:MANual:TDEV:VAT3?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:MANual:TDEV:VAT3.

- Function Queries TDEV τ3 for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:VAT3?
< 1000

:SOURce:JITTer:MANual:TDEV:VAT2

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
4 to 10000
* The possible setting range varies depending on the mask standard. For more information, see the table which comes after the detailed explanations on :SOURce:JITTer:MANual:TDEV:VAT0?.

- Function Specifies the TDEV τ2 for manual jitter measurement.

- Example use When setting TDEV τ2 for manual jitter measurement to 1000.
> :SOURce:JITTer:MANual:TDEV:VAT2 1000

:SOURce:JITTer:MANual:TDEV:VAT2?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:MANual:TDEV:VAT2.

- Function Queries TDEV τ2 for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:VAT2?
< 1000

:SOURce:JITTer:MANual:TDEV:VAT1

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.50 to 1000.00 Step value: 0.01
* The possible setting range varies depending on the mask standard. See limitations.
For more information, see the table which comes after the detailed explanations on :SOURce:JITTer:MANual:TDEV:VAT0?.

- Function Specifies the TDEV τ1 for manual jitter measurement.

- Example use When setting TDEV τ1 for manual jitter measurement to 20.
> :SOURce:JITTer:MANual:TDEV:VAT1 20

:SOURce:JITTer:MANual:TDEV:VAT1?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:MANual:TDEV:VAT1.

- Function Queries TDEV τ1 for manual jitter measurement.

- Example use > :SOURce:JITTer:MANual:TDEV:VAT1?
< 20

:SOURce:JITTer:MANual:TDEV:VAT0?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
  0.05 , 0.1
  * The response varies depending on the mask standard.

- Function Queries TDEV τ0 for manual jitter measurement.Manual jitter.

- Example use > :SOURce:JITTer:MANual:TDEV:VAT0?
  < 0.05

Table: Setting Range of Mask Values

Values Specified for Each Mask

Name of Mask Standard	Initial values for A0 and τ0 to τ4					
	A0	τ0	τ1	τ2	τ3	τ4
G.811-1997	3.0	0.10	100.00	1000	10000	-
G.812-1998 Section8.1 Table6	3.0	0.10	25.00	100	10000	-
G.812-1998 Section8.1 Table7	10.1	0.10	2.50	40	1000	10000
G.812-1998 Section9.1 Table11	34.0	0.10	20.00	100	1000	10000
G.812-1998 Section9.1 Table12	100.0	0.05	10.00	1000	-	-
G.812-1998 SectionA.3.1	10.1	0.10	2.50	40	1000	10000
G.812-1998 SectionA.4.1	100.0	0.05	10.00	1000	-	-
G.812-1998 SectionA.5	51.0	0.05	0.10	10	1000	-
G.813-1996 Section7.1 Option1	3.2	0.10	25.00	100	1000	-
G.813-1996 Section7.1 Option2	10.1	0.10	2.50	40	1000	10000
ETS 300 462-3-1997 Section7.2.1	3.0	0.10	100.00	1000	10000	-
ETS 300 462-3-1997 Section7.2.2	3.0	0.10	4.30	100	10000	-
ETS 300 462-3-1997 Section7.2.3	12.0	0.10	17.14	100	10000	-
ETS 300 462-3-1997 Section7.2.4	34.0	0.10	48.00	100	10000	-
ETS 300 462-4-1997 Section6.1	3.0	0.10	25.00	100	10000	-
ETS 300 462-4-1997 Section7.2	34.0	0.10	20.00	100	1000	10000
ETS 300 462-4-1997 Section8	3.0	0.10	13.30	100	1000	10000
ETS 300 462-5-1996 Section6.1	3.2	0.10	25.00	100	1000	-
ETS 300 462-5-1996 Section7.2	12.0	0.10	7.00	100	1000	-
ETS 300 462-6-1997	3.0	0.10	100.00	1000	10000	-
ANSI T1.101-1994 Section7.2.2	100.0	0.05	10.00	1000	-	-
ANSI T1.101-1994 Section7.3.2	10.0	0.05	1.73	30	1000	-
ANSI T1.105.03-1994 SectionD.2.1	10.0	0.05	1.73	30	1000	-
ANSI T1.105.03-1994 SectionD.2.2.1	100.0	0.05	10.00	1000	-	-
ANSI T1.105.03-1994 SectionD2.2.2	17.0	0.05	3.00	30	1000	-
ANSI T1.105.09-1996	10.0	0.05	10.00	1000	-	-
GR-2830-CORE-1995	10.1	0.10	2.50	40	10000	-
GR-1244-CORE-1995 Section4.3	100.0	0.05	10.00	1000	-	-
GR-1244-CORE-1995 Section5.3	10.1	0.10	2.50	40	1000	10000
GR-1244-CORE-1995 Section5.4 Stratum2&3E	2.0	0.05	1.10	300	1000	-
GR-1244-CORE-1995 Section5.4 Stratum3	50.0	0.05	0.10	10	1000	-
GR-253-CORE-1999 Section5.4.4.2.4 Fig5-15	10.0	0.10	1.70	30	1000	-
GR-253-CORE-1999 Section5.4.4.2.4 Fig5-16	17.0	0.10	3.00	30	1000	-
GR-253-CORE-1999 Section5.4.4.3.2	10.1	0.10	2.50	40	1000	10000
GR-253-CORE-1999 Section5.4.5	10.1	0.10	2.50	10000	-	-

Group No.of the range of each mask setting when TDEV (Mask): User

Name of ITU-T Mask Standard	No.
G.811-1997	5
G.812-1998 Section8.1 Table6	5
G.812-1998 Section8.1 Table7	11
G.812-1998 Section9.1 Table11	6
G.812-1998 Section9.1 Table12	1
G.812-1998 SectionA.3.1	11
G.812-1998 SectionA.4.1	1
G.812-1998 SectionA.5	12
G.813-1996 Section7.1 Option1	8
G.813-1996 Section7.1 Option2	11

Name of ETSI Mask Standard	No.
ETS 300 462-3-1997 Section7.2.1	10
ETS 300 462-3-1997 Section7.2.2	10
ETS 300 462-3-1997 Section7.2.3	10
ETS 300 462-3-1997 Section7.2.4	10
ETS 300 462-4-1997 Section6.1	5
ETS 300 462-4-1997 Section7.2	6
ETS 300 462-4-1997 Section8	7
ETS 300 462-5-1996 Section6.1	8
ETS 300 462-5-1996 Section7.2	5
ETS 300 462-6-1997	5

Name of Bellcore Mask Standard	No.
GR-2830-CORE-1995	3
GR-1244-CORE-1995 Section4.3	1
GR-1244-CORE-1995 Section5.3	11
GR-1244-CORE-1995 Section5.4 Stratum2&3E	2’
GR-1244-CORE-1995 Section5.4 Stratum3	12
GR-253-CORE-1999 Section5.4.4.2.4 Fig5-15	2
GR-253-CORE-1999 Section5.4.4.2.4 Fig5-16	2
GR-253-CORE-1999 Section5.4.4.3.2	11
GR-253-CORE-1999 Section5.4.5	4

Name of ANSI Mask Standard	No.
ANSI T1.101-1994 Section7.2.2	1
ANSI T1.101-1994 Section7.3.2	2
ANSI T1.105.03-1994 SectionD.2.1	2
ANSI T1.105.03-1994 SectionD.2.2.1	1
ANSI T1.105.03-1994 SectionD.2.2.2	2
ANSI T1.105.09-1996	1

Limitation of the Setting Range by Group No.

No.	Possible setting range for A0 and τ0 to τ4 when TDEV (Mask): User					
	A0	τ0	τ1	τ2	τ3	τ4
1	5.0 to 200.0	-	0.50 to 1000.00	τ1*10 to 10000	-	-
2	1.0 to 40.0	-	0.50 to 100.00	τ1*10 to 1000	τ2*10 to 10000	-
2'			0.50 to 333.00	τ1*10 to 3333	τ2*3 to 10000	
3	5.0 to 20.0	-	1.00 to 100.00	τ1*10 to 1000	τ2*10 to 10000	-
4	5.0 to 20.0	-	1.00 to 1000.00	τ1*10 to 10000	-	-
5	1.0 to 40.0	-	4.00 to 250.00	τ1*4 to 1000	τ2*10 to 10000	-
6	2.0 to 80.0	-	-	-	-	-
7	1.0 to 40.0	-	13.30	-	-	-
			13.10			
8	1.0 to 10.0	-	4.00 to 250.00	τ1* 4 to 1000	τ2*10 to 10000	-
9	5.0 to 20.0	-	1.00 to 333.00	τ1*10 to 3333	τ2* 3 to 10000	-
10	1.0 to 80.0	-	100.00	1000	-	-
			48.00	100		
			17.14	100		
			4.30	100		
11	5.0 to 20.0	-	-	-	-	-
12	25.0 to 100.0	-	0.10 to 100.00	τ1*10 to 1000	τ2*10 to 10000	-

Note: On-screen menu-based selections apply to the shaded sections in the table. However, input the numerical values in these shaded sections when using the remote commands.
The selection of A0 and τ1 is only enabled for No.10. Change τ2, depending on the set value for τ1.

:SOURce:JITTer:MANual:TDEV:DEFault

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

ITUT	ITU-T
ETS	ETSI
ANSI	ANSI
BELLcore	Bellcore

<table> = <CHARACTER PROGRAM DATA>

(When selecting ITU-T)

G811	G.811-1997
S81T6	Section 8.1 Table 6 (G.812-1998)
S81T7	Section 8.1 Table 7 (G.812-1998)
S91T11	Section 9.1 Table 11 (G.812-1998)
S91T12	Section 9.1 Table 12 (G.812-1998)
SA31	Section A.3.1 (G.812-1998)
SA41	Section A.4.1 (G.812-1998)
SA5	Section A.5 (G.812-1998)
S71O1	Section 7.1 Option1 (G.813-1996)
S71O2	Section 7.1 Option2 (G.813-1996)

(When selecting ETSI)

S721	Section 7.2.1 (ETS 300 462-3-1997)
S722	Section 7.2.2 (ETS 300 462-3-1997)
S723	Section 7.2.3 (ETS 300 462-3-1997)
S724	Section 7.2.4 (ETS 300 462-3-1997)
S61_4	Section 6.1 (ETS 300 462-4-1997)
S61_5	Section 6.1 (ETS 300 462-5-1996)
S72_4	Section 7.2 (ETS 300 462-4-1997)
S72_5	Section 7.2 (ETS 300 462-5-1996)
S8	Section 8 (ETS 300 462-4-1997)
ETS300_6	ETS 300 462-6-1997

(When selecting ANSI)

S722	Section 7.2.2 (ANSI TI.101-1994)
S732	Section 7.3.2 (ANSI TI.101-1994)
SD21	Section D.2.1 (ANSI TI.105.03-1994)
SD221	Section D.2.2.1 (ANSI TI.105.03-1994)
SD222	Section D.2.2.2 (ANSI TI.105.03-1994)
ANSIT1_9	ANSI T1.105.09-1996

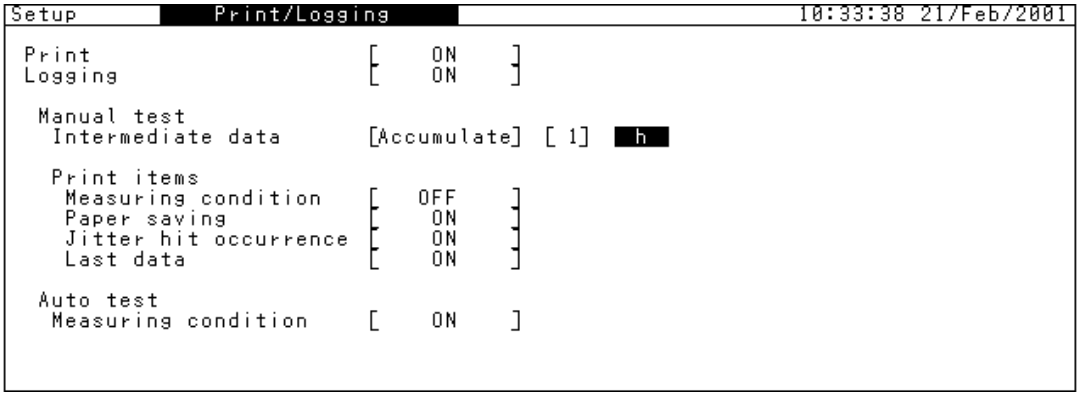
(When selecting Bellcore)

GR2830	GR-2830-CORE-1995
S43	Section 4.3 (GR-1244-CORE-1995)
S53	Section 5.3 (GR-1244-CORE-1995)
S54S2	Section 5.4 Strarum 2&3E (GR-1244-CORE-1995)
S54S3	Section 5.4 Strarum 3 (GR-1244-CORE-1995)
S54424F515	Section 5.4.4.2.4 Figure 5-15 (GR-253-CORE-1999)
S54425F516	Section 5.4.4.2.4 Figure 5-16 (GR-253-CORE-1999)
S54432	Section 5.4.4.3.2 (GR-253-CORE-1999)
S545	Section 5.4.5 (GR-253-CORE-1999)

- Function Initializes the Mask value when the mask standard selected for TDEV generation is User.

- Example use When specifying ITU-T as the TDEV Mask standard and specifying G.811-1997 as the Mask table, for manual jitter measurement.
  >:SOURce:JITTer:MANual:TDEV:DEFault ITUT, G811

11.1.5 Print/Logging sub-screen



Setting Item	Command	Reference Page
Print	:SYSTem:PRINt:APRINt	11-47
Logging	:SYSTem:PRINt:LOGGing	11-48
Intermediate data (type)	:SYSTem:PRINt:IDATa[:TYPE]	11-49
Intermediate data (Numerical value) Intermediate data (Unit)	:SYSTem:PRINt:IDATa:PERiod	11-50
Measuring condition (Manual)	:SYSTem:PRINt:MANual:MCONdition	11-51
Paper saving	:SYSTem:PRINt:MANual:SQUelch	11-52
Jitter Hit occurrence	:SYSTem:PRINt:JITTer:SET	11-53
Last data	:SYSTem:PRINt:LDATa:SET	11-54
Measuring condition (Auto)	:SYSTem:PRINt:AUTO:MCONdition	11-55

:SYSTem:PRINt:APRINt

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0 Sets automatic printing to OFF
ON or 1 Sets automatic printing to ON

- Function Switches automatic printing between ON and OFF.

- Example use When switching automatic printing to ON.
> :SYSTem:PRINt:APRINt ON

:SYSTem:PRINt:APRINt?

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>
0 Automatic printing is set to OFF
1 Automatic printing is set to ON

- Function Queries the setting status of automatic printing.

- Example use > :SYSTem:PRINt:APRINt?
< 1

:SYSTem:PRINt:LOGGing

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>
OFF or 0 Does not obtain log data.
ON or 1 Obtains log data.

- Function Specifies whether or not to record log data on the Analyze:Logging screen.
- Example use When starting the readout of log data.
- Example use > :SYSTem:PRINt:LOGGing ON

:SYSTem:PRINt:LOGGing?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 Does not obtain log data.
1 Obtains log data.
- Function Queries the setting status of whether or not log data is obtained on the Analyze:Logging screen.
- Example use > :SYSTem:PRINt:LOGGing?
< 1

:SYSTem:PRINt:IDATa[:TYPE]

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

OFF	Outputs no intermediate data.
INDividual	Value taken from the preceding intermediate data
ACCumulative	Value accumulated from the start of measurement

- Function Specifies the automatic printing format for intermediate measurement results.

- Example use When printing the value accumulated from the start of measurement.
  > :SYSTem:PRINt:IDATa ACC

:SYSTem:PRINt:IDATa?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>

OFF	Outputs no intermediate data.
IND	Value taken from preceding intermediate data
ACC	Value accumulated from the start of measurement

- Function Queries the automatic printing format for intermediate measurement results.

- Example use > :SYSTem:PRINt:IDATa?
  < ACC

:SYSTem:PRINt:IDATa:PERiod

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 99 Step value: 1
<suffix> = <CHARACTER PROGRAM DATA>
D day
H hour
M minute
S second

- Function Specifies the measurement intermediate printing timing.

- Example use When setting intermediate data printing period to 1 hour.
> :SYSTem:PRINt:IDATa:PERiod 1,H

:SYSTem:PRINt:IDATa:PERiod?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SYSTem:PRINt:IDATa:PERiod.
<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :SYSTem:PRINt:IDATa:PERiod.

- Function Queries the measurement intermediate printing timing.

- Example use > :SYSTem:PRINt:IDATa:PERiod?
< 1,H

:SYSTem:PRINt:MANual:MCONdition

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not print measurement conditions.
ON or 1	Prints measurement conditions.

- Function Specifies whether or not to print measurement conditions on the Test menu Manual screen.

- Example use When printing measurement conditions during manual measurement.
  > :SYSTem:PRINt:MANual:MCONdition ON

:SYSTem:PRINt:MANual:MCONdition?

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Does not print measurement conditions.
1	Prints measurement conditions.

- Function Queries whether or not to print measurement conditions on the Test menu Manual screen.

- Example use > :SYSTem:PRINt:MANual:MCONdition?
  < 1

:SYSTem:PRINt:MANual:SQUelch

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not perform printing in paper save mode.
ON or 1	Perform printing in paper save mode.

- Function Specifies whether or not to perform printing in paper save mode.

- Example use When performing printing in paper save mode.
  > :SYSTem:PRINt:MANual:SQUelch ON

:SYSTem:PRINt:MANual:SQUelch?

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Does not perform printing in paper save mode.
1	Perform printing in paper save mode.

- Function Queries the setting status of whether or not printing in paper save mode is performed.

- Example use > :SYSTem:PRINt:MANual:SQUelch?
  < 1

:SYSTem:PRINt:JITTer:SET

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not print jitter data.
ON or 1	Prints jitter data.

- Function Specifies whether or not to perform automatic printing when a jitter hit is counted.

- Example use When performing printing at jitter hit.
  > :SYSTem:PRINt:JITTer:SET ON

:SYSTem:PRINt:JITTer:SET?

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Does not print jitter data.
1	Prints jitter data.

- Function Queries the setting status of automatic printing when jitter hit is counted.

- Example use > :SYSTem:PRINt:JITTer:SET?
  < 1

:SYSTem:PRINt:LDATa:SET

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not print measurement result data.
ON or 1	Prints measurement result data.

- Function Specifies whether or not to print measurement result data.

- Example use When printing measurement result data.
  > :SYSTem:PRINt:LDATa:SET ON

:SYSTem:PRINt:LDATa:SET?

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Does not print measurement result data.
1	Prints measurement result data.

- Function Queries whether or not to print measurement result data.

- Example use > :SYSTem:PRINt:LDATa:SET?
  < 1

:SYSTem:PRINt:AUTO:MCONdition

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Does not print measurement conditions.
ON or 1	Prints measurement conditions.

- Function Specifies whether or not to set measurement conditions during automatic print.

- Example use When printing measurement conditions during automatic print.
  > :SYSTem:PRINt:AUTO:MCONdition ON

:SYSTem:PRINt:AUTO:MCONdition?

∇ Parameter None

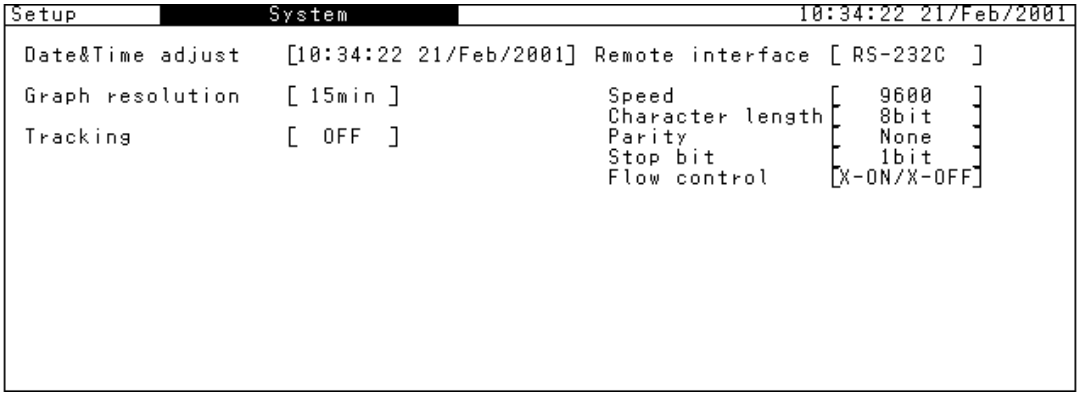
- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Does not print measurement conditions.
1	Prints measurement conditions.

- Function Queries whether or not to print measurement conditions during automatic print.

- Example use > :SYSTem:PRINt:AUTO:MCONdition?
  < 1

11.1.6 System sub-screen



Setting Item	Command	Reference Page
Graph resolution	:SENSe:MEASure:GRESolution	11-57
Date & Time adjust (Date)	:SYSTem:DATE	11-58
Date & Time adjust (Time)	:SYSTem:TIME	11-59

* The following items cannot be set, remotely.
Remote Interface
Tracking

- When RS-232C interface is used:
  Speed
  Character length
  Parity
  Stop bit
  Flow control

- When GPIB interface is used:
  Address

- When Ethernet interface is used:
  IP address
  Subnet mask
  Gateway address
  Port number
  MAC address

:SENSe:MEASure:GRESolution

∇ Parameter <gres> = <CHARACTER PROGRAM DATA>

MIN60	60 min
MIN15	15 min
MIN1	1 min

- Function Specifies the resolution for a single peak jitter graph on the Analyze screen.

- Example use When setting the resolution for the peak jitter graph on the Analyze screen to 1 min.
  > :SENSe:MEASure:GRESolution MIN1

:SENSe:MEASure:GRESolution?

∇ Parameter None

- Response <gres> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:MEASure:GRESolution.

- Function Queries the resolution for the peak jitter graph on the Analyze screen.

- Example use > :SENSe:MEASure:GRESolution?
  < MIN1

:SYSTem:DATE

∇ Parameter <numeric1> = <DECIMAL NUMERIC PROGRAM DATA>
2000 to 2093 (year)

<numeric2> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 12 (month)

<numeric3> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 31 (day)

- Function Sets current date.

- Example use When setting current date to May 30, 2000.
> :SYSTem:DATE 2000,5,30

:SYSTem:DATE?

∇ Parameter None

- Response <numeric1>,<numeric2>,<numeric3> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SYSTem:DATE.

- Function Queries current date.

- Example use > :SYSTem:DATE?
< 2000,5,30

:SYSTem:TIME

∇ Parameter <numeric1> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 23 (hour)

<numeric2> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59 (minute)

<numeric3> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59 (second)

- Function Sets current time.

- Example use When setting current time to 14:00:00.
> :SYSTem:TIME 14,0,0

:SYSTem:TIME?

∇ Parameter None

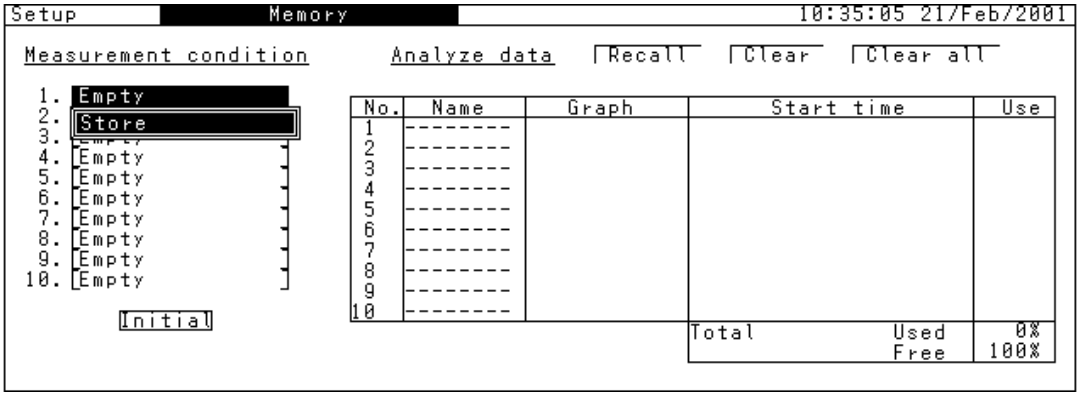
- Response <numeric1>,<numeric2>,<numeric3> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SYSTem:TIME.

- Function Queries current time.

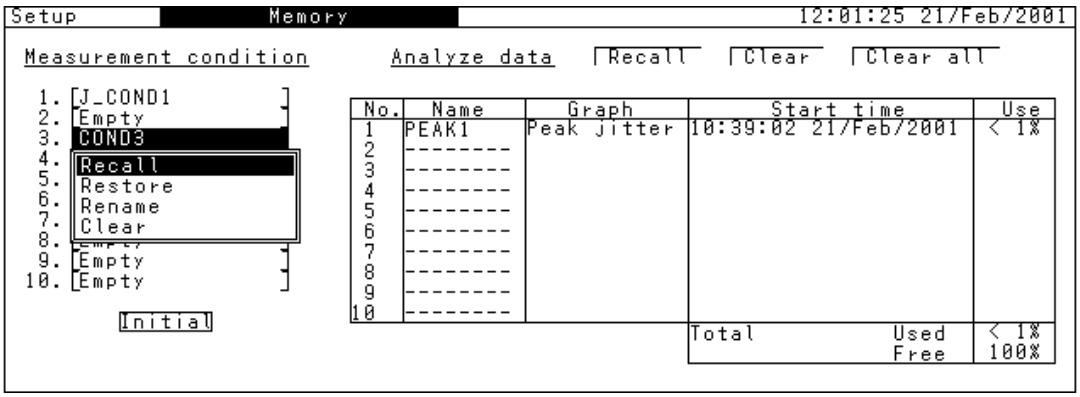
- Example use > :SYSTem:TIME?
< 14,0,0

11.1.7 Memory sub-screen

<When storing data>



<When displaying the function selection window>



Setting Item	Command	Reference Page
No.1 to 10 (Setting data) Recall	:SYSTem:MEMory:RECall	11-61
Initial key	:SYSTem:MEMory:RECall	11-61
No.1 to 10 (Setting data) Restore,Store	:SYSTem:MEMory:STORe	11-61
No.1 to 10 (Setting data) (Rename) (Name of setting data memory)	:SYSTem:MEMory:LABel	11-62
No.1 to 11 (Setting data) Clear	:SYSTem:MEMory:CLEar	11-63
No.1 to 10 (Measurement data) Name	:SYSTem:MEMory:ANALysis:LABel?	11-64
No.1 to 10 (Measurement data) Recall key	:SYSTem:MEMory:ANALysis:RECall	11-65
No.1 to 10 (Measurement data) Clear key	:SYSTem:MEMory:ANALysis:CLEar	11-66
Clear all key	:SYSTem:MEMory:ANALysis:ACLear	11-66

:SYSTem:MEMory:RECall

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 10 Memory No.0 to 10

- Function Recalls setting data from memory.
  The setting data is initialized when <numeric> = 0.

- Example use When recalling data from Memory No.1.
  > :SYSTem:MEMory:RECall 1

:SYSTem:MEMory:STORe

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No.1 to 10

- Function Writes the setting data to memory.
  (Same as the common command,"*SAV")

- Example use When writing data to Memory No.3.
  > :SYSTem:MEMory:STORe 3

:SYSTem:MEMory:LABel

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No.1 to 10

<title> = <STRING PROGRAM DATA>
Maximum 15 characters
Input characters: 0 to 9,A to Z,a to z, $&#%'.()-@_^{}^!
"Memory name"
Spaces are added when the number of characters is lower than 15.

- Function Writes the name of the setting data (renaming).

- Example use When naming the Memory No.1 file to "JIT TEST-1".
> :SYSTem:MEMory:LABel 1,"JIT TEST-1

:SYSTem:MEMory:LABel?

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No.1 to 10

- Response <title> = <STRING RESPONSE DATA>
"Memory name"

- Function Queries the name of the setting data.

- Example use When querying the file name of Memory No.1.
> :SYSTem:MEMory:LABel? 1
<"JIT TEST-1 "

:SYSTem:MEMory:CLEar

▽ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No.1 to 10

- Function Clears setting data from memory.

- Example use When deleting files in Memory No.3.
> :SYSTem:MEMory:CLEar 3

:SYSTem:MEMory:ANALysis:LABel?

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

1 to 10	Memory No.1 to 10

- Response

<title> = <STRING RESPONSE DATA>
Memory name (A fixed length of eight characters)
"abcdefgh"

<gtype> = <CHARACTER RESPONSE DATA>
Graph type

JTR	Jitter transfer characteristics
WAND	Wander data
FREQ	Frequency graph data
LOGG	Logging data

<stime> = <STRING RESPONSE DATA>
Time to start measurement (A fixed length of 19 characters)
"2000.12.25 18:40:30"

<use> = <STRING RESPONSE DATA>
Amount of used memory (Indication in percentage)
Form1
Note: If memory use is less than 1 %, it is output as 1 %.

- Function

Queries the registration status of analysis data memory.
* In the absence of data, an output is made as shown below.
"--------",----,"--------------------","--------"

- Example use

When querying the registration status of analysis data memory No.1.
> :SYSTem:MEMory:ANALysis:LABel? 1
<"JIT ANAL",JTR,"2000.12.25 18:40:30"," 30"

:SYSTem:MEMory:ANALysis:RECall

▽ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No.1 to 10

- Function Recalls analysis data from memory.

- Example use When recalling analysis data from Memory No.1.
> :SYSTem:MEMory:ANALysis:RECall 1

:SYSTem:MEMory:ANALysis:CLEar

∇ Parameter <numeric>= <DECIMAL NUMERIC PROGRAM DATA>
1 to 10 Memory No.1 to 10

- Function Clears analysis data from memory.

- Example use When deleting analysis data from Memory No.1.
> :SYSTem:MEMory:ANALysis:CLEar 1

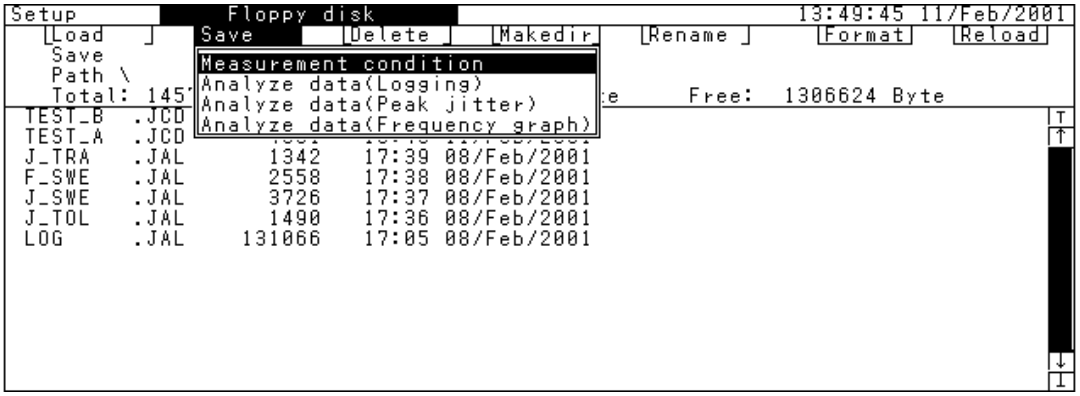
:SYSTem:MEMory:ANALysis:ACLear

∇ Parameter None

- Function Clears all analysis data from memory.

- Example use > :SYSTem:MEMory:ANALysis:ACLear

11.1.8 Floppy disk sub-screen



Setting Item	Command	Reference Page
Load key	:SYSTem:MMEMory:RECall	11-68
Save key (Name)	:SYSTem:MMEMory:STORe	11-69
Delete key	:SYSTem:MMEMory:DELete	11-70
Makedir key	:SYSTem:MMEMory:MDIRectory	11-70
Rename key	:SYSTem:MMEMory:REName	11-71
Format key	:SYSTem:MMEMory:INITialize	11-71
Reload key	:SYSTem:MMEMory:RELoad	11-72
Total Used Free (Name) (Size) (Date) Path	:SYSTem:MMEMory:CATalog?	11-73
Path	:SYSTem:MMEMory:CDIRectory	11-74

Note: When the floppy disk drive is in operation, only the query command can be executed.

:SYSTem:MMEMory:RECall

∇ Parameter <file_name> = <STRING PROGRAM DATA>
"File name" (Not case-sensitive. The extension is included.)
A file name can be from 1 to 12 characters long, with the use of "" disabled.

- Function Reads the setting (measurement) data stored in the current directory on the floppy disk.

- Example use When reading out the setting data to a file,"DEMO1.JCD".
> :SYSTem:MMEMory:RECall"DEMO1.JCD"

:SYSTem:MMEMory:STORe

∇ Parameter

<type> = <STRING PROGRAM DATA>

"CONDition"	Setting status data (binary format)
"LOGGomg:LOGGing"	Analysis data on the Logging screen (binary format)
"LOGGomg:LOGText"	Analysis data on the Logging screen (text format)
"PEAK:PEAK"	Analysis data on the Peak jitter screen (binary format)
"PEAK:PTEXt"	Analysis data on the Peak jitter screen (text format)
"JTRansfer:JTRansfer"	Analysis data on the Jitter Transfer screen (binary format)
"JTRansfer:JTRText"	Analysis data on the Jitter Transfer screen (text format)
"WANDer:WANDer"	Analysis data on the Wander screen (binary format)
"WANDer:WTEXt"	Analysis data on the Wander screen (text format)
"FREQuency:FREQuency"	Analysis data on the Frequency graph screen (binary format)
"FREQuency:FTEXt"	Analysis data on the Frequency graph screen (text format)
"RECall:RECall"	Analysis data on the Recall screen (binary format)
"RECall:RTEXt"	Analysis data on the Recall screen (text format)

<file_name> = <STRING PROGRAM DATA>
"File name" (Not case-sensitive. The extension is included.)
A file name can be from 1 to 12 characters long, with the use of "" disabled.
(Reference) The following extensions are added when files are saved, in accordance with the specified parameter.
Setting conditions (binary format): *.JCD
Analysis data (binary format): *.JAL
Text data (text format): *.TXT

- Function

Saves setting (measurement) data to the current directory on the floppy disk.
When the number of disks required for saving data is more than one, the operation cannot be executed using the remote command.
In this case, the operation results in an execution error, "Setting conflict".

- Example use

When writing analysis data (binary format) on the Jitter Transfer screen to a file named "DEMO1".
> :SYSTem:MMEMory:STORe"JTRansfer:JTRansfer","DEMO1"

A file named DEMO1.JAL is saved to the floppy disk when this command is sent.

:SYSTem:MMEMory:DELete

∇ Parameter <file_name> = <STRING PROGRAM DATA>
"File name" (Not case-sensitive. The extension is included.)
A file name can be from 1 to 12 characters long, with the use of "" disabled.

- Function Deletes a file or directory on the floppy disk.
(Only within the current directory)

- Example use When deleting a file named "DEMO1.JCD".
> :SYSTem:MMEMory:DELete"DEMO1.JCD"

:SYSTem:MMEMory:MDIRectory

∇ Parameter <dir_name> = <STRING PROGRAM DATA>
"Directory name" (Not case-sensitive. The extension is included.)
A directory name can be from 1 to 12 characters long, with the use of "" disabled.

- Function Creates a sub-directory on the floppy disk.
(Only within the current directory)

- Example use When creating a sub-directory named "ORIGINAL" under the current directory.
> :SYSTem:MMEMory:MDIRectory"ORIGINAL"

:SYSTem:MMEMory:REName

∇ Parameter <src_file> = <STRING PROGRAM DATA>
File name before renaming (Not case-sensitive.)
Maximum 12 characters

<dst_file> = <STRING PROGRAM DATA>
File name after renaming (Not case-sensitive.)
Maximum 12 characters

- Function Changes the file name on the floppy disk. (Only within the current directory)

- Example use When changing the file name "DEMO1.JCD" to "DEMO2.JCD".
> :SYSTem:MMEMory:REName"DEMO1.JCD","DEMO2.JCD"

:SYSTem:MMEMory:INITialize

∇ Parameter <None>

- Function Formats a 1.44-MB floppy disk.

- Example use When formatting the floppy disk.
:SYSTem:MMEMory:INITialize

:SYSTem:MMEMory:RELoad

∇ Parameter <None>

- Function Retries to read the information on the floppy disk.

- Example use When retrying to read the information on the floppy disk.
:SYSTem:MMEMory:RELoad

:SYSTem:MMEMory:CATalog?

▽ Parameter <numeric> = <STRING PROGRAM DATA>
1 to 100 Number for the first line of the displayed data
Allocates Line No.1 to a file or sub-directory of the most recent date.

- Response <use_byte> = <NR1 NUMERIC RESPONSE DATA>
0 to 1457664 Used space

<free_byte> = <NR1 NUMERIC RESPONSE DATA>
0 to 1457664 Free space

<current_dir> = <STRING RESPONSE DATA>
Current directory name

<current_file> = <NR1 NUMERIC RESPONSE DATA>
0 to 100 Total number of files in the current directory (including files in sub-directories.)

<file_entry> = <STRING RESPONSE DATA>
Information on files in the current directory
Outputs the following information for each file.
Outputs information in the order from the most recent date. Sub-directories take precedence over files.

<file_name>,<file_size>,<date_time>
<file_name> : Sub-directory or file name
Upper-case characters
(A file name includes the extension. The sub-directory name is enclosed with < >.)
<file_size> : File size (Form1)
<date_time> : File date and time

Note: - If disk information cannot be read out (disk not inserted),
0,0,"--------",0,"--------,---------,----------"
- If there are no disk information,
0,1457664,"¥",0,"--------,---------,----------"

- Function Queries the information on the floppy disk.
Maximum 4 files or sub-directories can be output at a time.

- Example use > :SYSTem:MMEMory:CATalog? 1
< 1024,1456640,"¥ORIGINAL",35,
"<SUB_ORG1>,---------,----------,
<SUB_ORG2>,---------,----------,
DEMO___1.JCD, 256,2000.12.31 10:30,
DEMO___2.JCD, 256,2000.12.31 10:22"

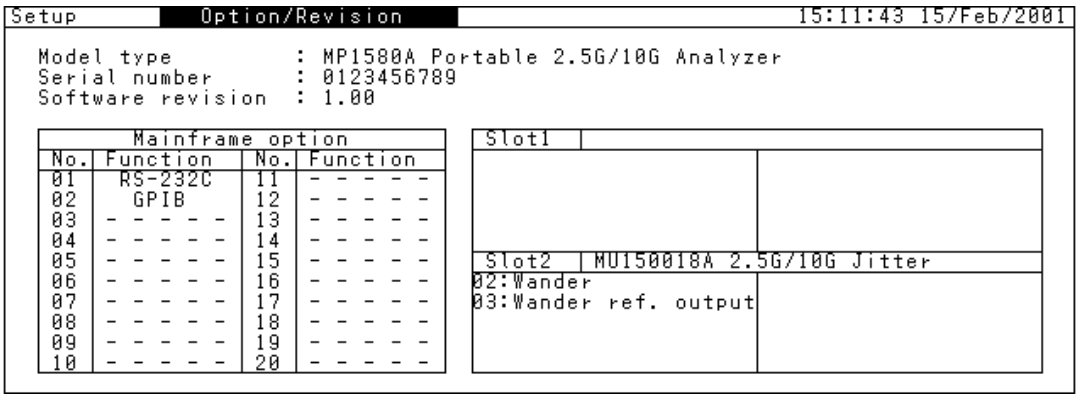
:SYSTem:MMEMory:CDIRectory

▽ Parameter <dir_name> = <STRING PROGRAM DATA>
Directory name (Not case-sensitive. Both the relative path and full path exist.)
A directory name can be from 1 to 63 characters long, with the use of "" disabled.
For the information on directory contents for the parameters,
use :SYSTem:MMEMory:CATalog? described on the previous page.

- Function Moves a directory on the floppy disk. (Only within the current directory)

- Example use When moving to a sub-directory named "ORIGINAL" located on the same level as the current sub-directory.
> :SYSTem:MMEMory:CDIRectory"..¥ORIGINAL"

11.1.9 Option/Revision sub-screen



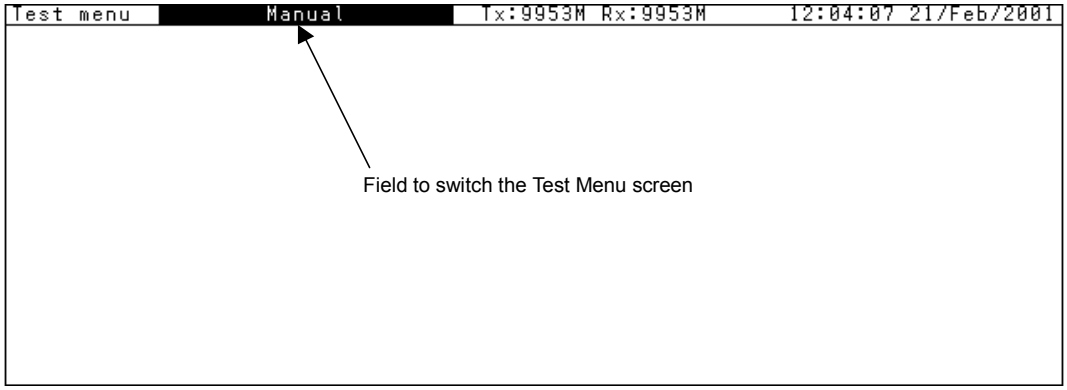
Setting Item	Command	Reference Page
Model type Serial number	*IDN?	6-4
Mainframe option No. Mainframe option Function Slot	*OPT?	6-11

* See the common commands for *IDN? and *OPT?.

11.2 Test Menu Main Screen

11.2.1 Manual sub-screen

a) Common command



A command commonly used is shown below.

Setting Item (Tx)	Command	Reference Page
Screen switching command commonly used on Test menu screen	:DISPlay:TMENu[:NAME]	11-77

:DISPlay:TMENu[:NAME]

∇ Parameter <tdisplay> = <STRING PROGRAM DATA>

"MANual"	The Manual screen
"JTRansfer"	The Jitter transfer screen
"WANDer"	The Wander screen

- Function: Selects an item for display on the Test Menu screen.
  The type of measurement is determined depending on which parameter of the command above is selected.

- Example use: When selecting on the MANual screen the item for display on the Test Menu screen.

```
> :DISPlay:TMENu:NAME"MANual"
                or
> :DISPlay:TMENu"MANual"
```

:DISPlay:TMENu[:NAME]?

∇ Parameter None

- Response <tdisplay> = <STRING RESPONSE DATA>

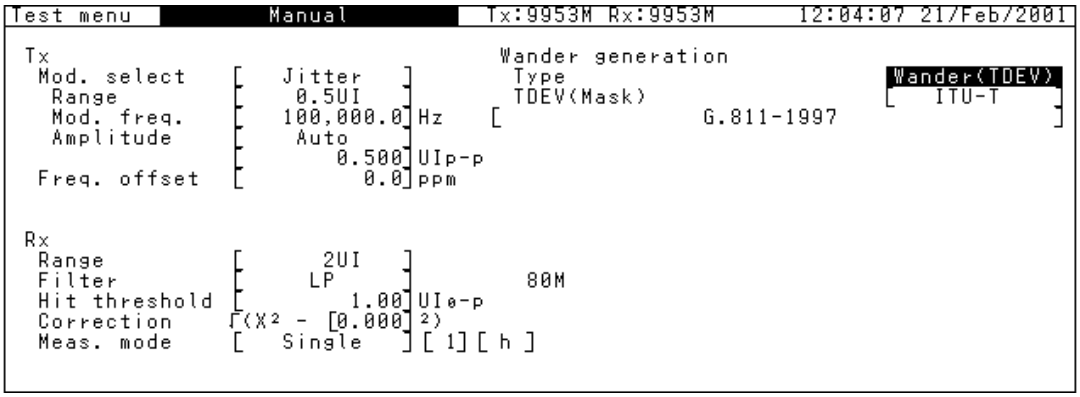
"MAN"	The Manual (STM) screen
"JTR"	The Jitter transfer screen
"WAND"	The Wander screen

- Function: Queries the item for display on the Test Menu screen.
  Example use

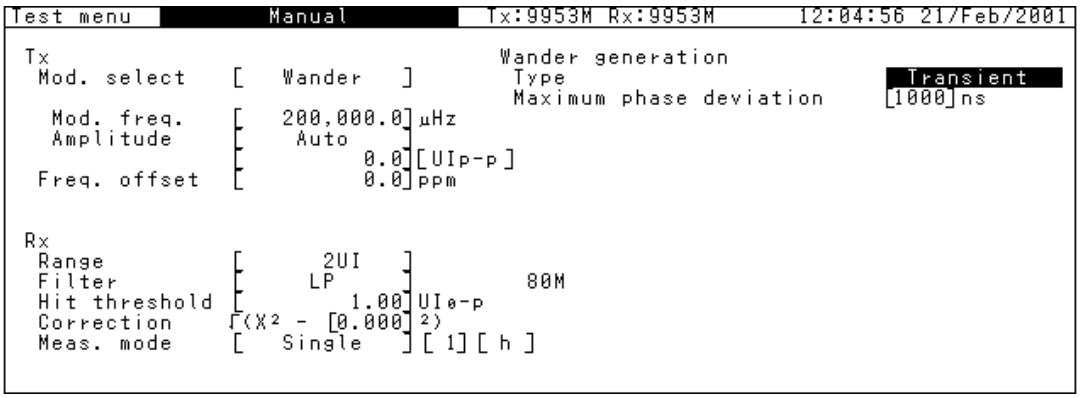
```
> :DISPlay:TMENu:NAME?
        or
> :DISPlay:TMENu?
<"MAN"
```

b) When Mod. Select: Jitter selected

- When Amplitude is set to "Auto"



c) When Mod. Select: Wander selected



- When Wander generation: Type is set to "Transient".

When Jitter Mod. Source: External on the Setup: Jitter/Wander screen selected, Mod. freq. (Jitter) appears as shown below.

- Tx

Setting Item (Tx)	Command	Reference Page
Mod. select	:SOURce:TELecom:JWANder:MSELect	11-80
Range	:SOURce:JITTer:MANual:RANGe	11-81
Mod. freq (Numerical value) Mod. freq (Unit)	:SOURce:JITTer:MANual:FREQuency	11-82
Amplitude (Type) when Mod. select: Jitter	:SOURce:JITTer:MANual:AMPLitude:TYPE	11-83
Amplitude (Numerical value in UIp-p) when Mod. select: Jitter	:SOURce:JITTer:MANual:AMPLitude:UIPP	11-84
When Mod. Select: Wander Mod. freq (Numerical value) Mod. freq (Unit)	:SOURce:WANDer:MANual:FREQuency	11-86
Amplitude (Type) when Mod. Select: Wander	:SOURce:WANDer:MANual:AMPLitude:TYPE	11-87
Amplitude (Numerical value in UIp-p) when Mod. Select: Wander	:SOURce:WANDer:MANual:AMPLitude:UIPP	11-88
Amplitude (Numerical value in ns) when Mod. Select: Wander	:SOURce:WANDer:MANual:AMPLitude:NSEC	11-89
Freq. offset	:SOURce:TELecom:OFFSet	11-90

- Rx

Setting Item (Rx)	Command	Reference Page
Range	:SENSe:JITTer:MANual:RANGe	11-91
Filter (Type)	:SENSe:JITTer:MANual:FILTer	11-92
Hit threshold	:SENSe:JITTer:MANual:THReshold	11-93
Correction	:SENSe:MEASure:JWANder:CORRection:OFFSet	11-94
Meas.mode (Type)	:SENSe:MEASure:TYPE	11-95
Meas.mode (Numerical value) Meas.mode (Unit)	:SENSe:MEASure:PERiod	11-96

- Wander Generation

Setting Item (Wander generation)	Item	Reference Page
Type	:SOURce:JITTer:WANDgen:TYPE	11-97
Mask (TDEV)	:SOURce:JITTer:WANDgen:TDEV:DTYPe	11-98
Mask (TDEV) table	:SOURce:JITTer:WANDgen:TDEV:ITYPe	11-99
Mask (TDEV) table	:SOURce:JITTer:WANDgen:TDEV:ETYPe	11-100
Mask (TDEV) table	:SOURce:JITTer:WANDgen:TDEV:ATYPe	11-101
Mask (TDEV) table	:SOURce:JITTer:WANDgen:TDEV:BTYPe	11-102
Maximum phase deviation	:SOURce:JITTer:WANDgen:MPDeviation	11-103

:SOURce:TELecom:JWANder:MSELect

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

OFF	Generates no jitter.
JITTer	Generates jitter.
WANDer	Generates wander.

- Function Specifies whether or not to generate jitter or wander.

- Example use When specifying generating jitter.
  > :SOURce:TELecom:JWANder:MSELect JITTer

:SOURce:TELecom:JWANder:MSELect?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>

OFF	Generates no jitter.
JITT	Generates jitter.
WAND	Generates wander.

- Function Queries whether or not to generate jitter or wander.

- Example use > :SOURce:TELecom:JWANder:MSELect?
  < JITT

:SOURce:JITTer:MANual:RANGe

∇ Parameter <numeric> = <CHARACTER PROGRAM DATA>

4000	4000 UI
1000	1000 UI
80	80 UI
20	20 UI
0_5	0.5 UI

- Function Specifies the range of jitter generation regarding jitter generation.

- Example use When setting the range of jitter generation for jitter measurement to 80 UI.
  > :SOURce:JITTer:MANual:RANGe 80

:SOURce:JITTer:MANual:RANGe?

∇ Parameter None

- Response <numeric> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:RANGe.

- Function Queries the range of jitter generation regarding jitter generation.

- Example use > :SOURce:JITTer:MANual:RANGe?
  < 80

:SOURce:JITTer:MANual:FREQuency

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.1 to 999.9 Step value: 0.1
The setting range and valid digits of Mod. freq are determined based on the Tx Range.

Bit rate	Tx Range	Setting Range (Hz)
9953M	4000 UI	0.1 to 600
	80 UI	0.1 to 2 M
	0.5 UI	100 k to 80 M
2488M	1000 UI	0.1 to 600
	20 UI	0.1 to 2 M
	0.5 UI	100 k to 20 M

Mod. freq	Unit	Step
0.1 to 0.9	Hz	0.1
1.0 to 99.0		1
100.0 to 990.0		10
1.0 to 9.9	kHz	0.1
10.0 to 99.0		1
100.0 to 990.0		10
1.0 to 9.9	MHz	0.1
10.0 to 80.0		1

<suffix> = <CHARACTER PROGRAM DATA>
HZ Hz
KHZ KHz
MHZ MHz

* The application changes <numeric> to a value within the allowable range.

- Function Sets the jitter modulating frequency regarding jitter generation.

- Example use When setting the modulating frequency to 100.0 Hz regarding jitter generation.
> :SOURce:JITTer:MANual:FREQuency 100.0,HZ

:SOURce:JITTer:MANual:FREQuency?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:MANual:FREQuency.

<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :SOURce:JITTer:MANual:FREQuency.

- Function Queries the jitter modulating frequency regarding jitter generation.

- Example use > :SOURce:JITTer:MANual:FREQuency?
< 100.0,HZ

:SOURce:JITTer:MANual:AMPLitude:TYPE

∇ Parameter <type> = <CHARACTER PROGRAM DATA>
MANual
AUTO

- Function Specifies how to set the value of jitter amplitude regarding jitter generation.

- Example use When setting the setting mode AUTO.
> :SOURce:JITTer:MANual:AMPLitude:TYPE AUTO

:SOURce:JITTer:MANual:AMPLitude:TYPE?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
MAN
AUTO

- Function Queries how to set the value of jitter amplitude regarding jitter generation.

- Example use When querying how to set the value of jitter amplitude regarding jitter generation.
> :SOURce:JITTer:MANual:AMPLitude:TYPE?
< AUTO

:SOURce:JITTer:MANual:AMPLitude:UIPP

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0.000 to 4040.000 Step value: 0.001 to 2

The setting range varies depending on the Tx Range.

In addition, when the set value falls outside the allowable range, the value is rounded off to a value which marks the limit of the allowable range.

Setting Range based on the Tx Range

Tx Range	Mod. freq (Hz)	Setting Range	Step
4000 UI	0.1 to 15	0 to 4040	2
	16 to 600	0 to 4040 × 10^ (-1 × log (f/15)) *1	
1000 UI	0.1 to 15	0 to 1010	1
	16 to 600	0 to 1010 × 10^ (-1 × log (f/15)) *2	
80 UI	0.1 to 100 k	0.00 to 80.80	0.05
	110 k to 2 M	0.00 to 80.80 × 10^ (-1 × log (f/100000)) *3	
20 UI	0.1 to 100 k	0.00 to 20.20	0.01
	110 k to 2 M	0.00 to 20.20 × 10^ (-1 × log (f/100000)) *4	
0.5 UI	100 k to 80 M	0.000 to 0.505	0.001

Note: *1: The maximum set value must be worked out by truncating decimal places from the value obtained through the equation.
In case an odd number is obtained, 1 must be subtracted from that number.

*2: The maximum set value must be worked out by truncating decimal places from the value obtained through the equation.

*3: The maximum set value must be worked out by truncating the third decimal places from the value obtained through the equation.
In case an odd number is obtained, 0.01 must be subtracted from the number.

*4: The maximum set value must be worked out by truncating the third decimal places from the value obtained through the equation.

- Function Specifies the value of jitter amplitude regarding jitter generation.
When this command is used, the :SOURce:JITTer:MANual:AMPLitude:TYPE setting is automatically set to AUTO.

- Example use When setting the value of jitter amplitude to 1.5.
> :SOURce:JITTer:MANual:AMPLitude:UIPP 1.5

:SOURce:JITTer:MANual:AMPLitude:UIPP?

▽ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:MANual:AMPLitude:UIPP

- Function Queries the value of jitter amplitude regarding jitter generation.

- Example use When querying the value of jitter amplitude regarding jitter generation.
  > :SOURce:JITTer:MANual:AMPLitude:UIPP?
  < 1.5

:SOURce:WANDer:MANual:FREQuency

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.01 to 999.99 Step value: 0.1
<suffix> = <CHARACTER PROGRAM DATA>
UHZ μHz
MLHZ mHz
HZ Hz

* The application changes <numeric> to a value within the allowable range.

Mod. Freq. Setting Range (Valid digit -- First three digits)

Mod. Freq.	Unit	Step
10.0 to 99.9	μHz	0.1
100 to 999		1
1.00 to 9.99	mHz	0.01
10.0 to 99.9		0.1
100 to 999		1
1.00 to 10.00	Hz	0.01

- Function Specifies the wander modulating frequency regarding wander generation.

- Example use When setting the modulating frequency to 100 mHz regarding wander generation.
> :SOURce:WANDer:MANual:FREQuency 100.0,MLHZ

:SOURce:WANDer:MANual:FREQuency?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:WANDer:MANual:FREQuency.

<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :SOURce:WANDer:MANual:FREQuency.

- Function Queries the wander modulating frequency regarding wander generation.

- Example use > :SOURce:WANDer:MANual:FREQuency?
< 100.0,MLHZ

:SOURce:WANDer:MANual:AMPLitude:TYPE

∇ Parameter <type> = <CHARACTER PROGRAM DATA>
MANual
AUTO

- Function Specifies how to set the value of wander amplitude regarding wander generation.

- Example use When setting the setting mode to AUTO.
> :SOURce:WANDer:MANual:AMPLitude:TYPE AUTO

:SOURce:WANDer:MANual:AMPLitude:TYPE?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
MAN
AUTO

- Function Queries the value of wander amplitude regarding wander generation.

- Example use When querying how to set the value of wander amplitude regarding wander generation.
> :SOURce:WANDer:MANual:AMPLitude:TYPE?
< AUTO

:SOURce:WANDer:MANual:AMPLitude:UIPP

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 400000 Step value: 1

- Function Specifies the value of wander amplitude regarding wander generation.
When this command is used, the parameter for
the :SOURce:WANDer:MANual:AMPLitude:TYPE command is automatically set to AUTO.

- Example use When setting the value of wander amplitude to 100 UIp-p.
> :SOURce:WANDer:MANual:AMPLitude:UIPP 100

:SOURce:WANDer:MANual:AMPLitude:UIPP?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:WANDer:MANual:AMPLitude:UIPP

- Function Queries the value of wander amplitude regarding wander generation.

- Example use When querying the value of wander amplitude regarding wander generation.
> :SOURce:WANDer:MANual:AMPLitude:UIPP?
< 100

:SOURce:WANDer:MANual:AMPLitude:NSEC

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 160751
Note: Since the conversion into the ns value is calculated based on UIp-p, a rounded value is set.

- Function Specifies the value of wander amplitude regarding wander generation.
When this command is used, the parameter for the :SOURce:WANDer:MANual:AMPLitude:TYPE command is automatically set to AUTO.

- Example use When setting the value of wander amplitude to 100 ns.
> :SOURce:WANDer:MANual:AMPLitude:NSEC 100

:SOURce:WANDer:MANual:AMPLitude:NSEC?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:WANDer:MANual:AMPLitude:NSEC.

- Function Queries the value of wander amplitude regarding wander generation.

- Example use When querying the value of wander amplitude regarding wander generation.
> :SOURce:WANDer:MANual:AMPLitude:NSEC?
< 100

:SOURce:TELecom:OFFSet

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-100.0 to 100.0 Step value: 0.1

- Function Specifies the frequency offset.

- Example use When setting the frequency offset to -50.3.
> :SOURce:TELecom:OFFSet –50.3

:SOURce:TELecom:OFFSet?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:TELecom:OFFSet.

- Function Queries the value of frequency offset.

- Example use > :SOURce:TELecom:OFFSet?
< -50.3

:SENSe:JITTer:MANual:RANGe

∇ Parameter <numeric> = <CHARACTER PROGRAM DATA>

4000	4000 UI range
1000	1000 UI range
20	20 UI range
2	2 UI range

- Function Specifies the measuring range.

- Example use When setting the measuring range to 20 UI.
  > :SENSe:JITTer:MANual:RANGe 20

:SENSe:JITTer:MANual:RANGe?

∇ Parameter None

- Response <numeric> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:JITTer:MANual:RANGe

- Function Queries the measuring range.

- Example use > :SENSe:JITTer:MANual:RANGe?
  < 20

:SENSe:JITTer:MANual:FILTer

∇ Parameter <filter> = <CHARACTER PROGRAM DATA>

LP	Low-pass filter
LPHP0	Low-pass filter + High-pass filter 0
LPHP1	Low-pass filter + High-pass filter 1
LPHP1S	Low-pass filter + High-pass filter 1'
LPHP2	Low-pass filter + High-pass filter 2
LPHP	Inserts Low-pass filter + High-pass filter.
LPHPS	Inserts Low-pass filter + High-pass filter'.
LPS	Inserts Low-pass filter'.
LPSHP0	Inserts Low-pass filter' + High-pass filter 0.

- Function Specifies a filter type.
  There is a limit to the type of filter that can be specified, depending on the set Rx Range.

- Example use When setting the filter type to LPHP0.
  > :SENSe:JITTer:MANual:FILTer LPHP0

:SENSe:JITTer:MANual:FILTer?

∇ Parameter None

- Response <filter> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:JITTer:MANual:FILTer.

- Function Queries the filter type.

- Example use > :SENSe:JITTer:MANual:FILTer?
  < LPHP0

:SENSe:JITTer:MANual:THReshold

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.05 to 10.0 Steps are shown below.

Limitation of the Setting Range based on the Rx Range

Rx Range	Setting Range	Step
2 UI	0.05 to 1.00	0.01
20 UI	0.5 to 10.0	0.1

Limitation Caused by Rx Range Switching

Range after switching	Before switching		The last digit is truncated.
	Range	Hit threshold	
2 UI	20 UI	0.5 to 1.0	0.50 to 1.00
		>1.0	1.00
20 UI	2 UI	< 0.50	0.5
		0.50 to 1.00	The last digit is truncated.

- Function Specifies the threshold value for hit measurement.

- Example use When setting the threshold value for hit measurement to 0.05.
> :SENSe:JITTer:MANual:THReshold 0.05

:SENSe:JITTer:MANual:THReshold?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:JITTer:MANual:THReshold.

- Function Queries the hit threshold.

- Example use > :SENSe:JITTer:MANual:THReshold?
< 0.05

:SENSe:MEASure:JWANder:CORRection:OFFSet

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0.000 to 0.714 Step value: 0.001
* Digits smaller than the resolution are truncated.

- Function Specifies the correction value of jitter measurement (RMS).

- Example use When setting the correction value of jitter measurement (RMS) to 0.1.
> :SENSe:MEASure:JWANder:CORRection:OFFSet 0.1

:SENSe:MEASure:JWANder:CORRection:OFFSet?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:MEASure:JWANder:CORRection:OFFSet.

- Function Queries the correction value of jitter measurement (RMS).

- Example use When querying the correction value of jitter measurement (RMS).
> :SENSe:MEASure:JWANder:CORRection:OFFSet?
< 0.1

:SENSe:MEASure:TYPE

∇ Parameter <mmode> = <CHARACTER PROGRAM DATA>

SINGle	Single measurement
REPeat	Repeat measurement
MANual	Manual measurement

- Function Specifies the measurement mode.

- Example use When setting the measurement mode to Repeat measurement.
  > :SENSe:MEASure:TYPE REPeat

:SENSe:MEASure:TYPE?

∇ Parameter None

- Response <mmode> = <CHARACTER RESPONSE DATA>

SING	Single measurement
REP	Repeat measurement
MAN	Manual measurement

- Function Queries the measurement mode.

- Example use > :SENSe:MEASure:TYPE?
  < REP

:SENSe:MEASure:PERiod

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 99 Step value: 1

<suffix> = <CHARACTER PROGRAM DATA>
S second
M minute
H hour
D day

- Function Specifies time and unit of the gating period for manual measurement.

- Example use When setting the measurement time to one hour.
> :SENSe:MEASure:PERiod 1,H

:SENSe:MEASure:PERiod?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:MEASure:PERiod.

<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :SENSe:MEASure:PERiod.

- Function Queries the measurement time.

- Example use > :SENSe:MEASure:PERiod?
< 1,H

:SOURce:JITTer:WANDgen:TYPE

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

OFF	OFF
WANDer	Wander (TDEV)
TRANsient	Transient
SIGNal	Signal off

- Function Specifies the type of wander generation.

- Example use When setting the type of wander generation to"Transient".
  > :SOURce:JITTer:WANDgen:TYPE TRANsient

:SOURce:JITTer:WANDgen:TYPE?

∇ Parameter None

- Response <type> = <CHARACTER PROGRAM DATA>
  Same as in the case of :SOURce:JITTer:WANDgen:TYPE.

- Function Queries the setting of type of wander generation.

- Example use When querying the setting of the type of wander generation.
  > :SOURce:JITTer:WANDgen:TYPE?
  < TRAN

:SOURce:JITTer:WANDgen:TDEV:DTYPE

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

ITUT	ITU-T
ETSI	ETSI
ANSI	ANSI
BELLcore	Bellcore
USER	User

- Function Specifies the TDEV Mask standard for Manual WandGen measurement.

- Example use When specifying ITU-T as the TDEV Mask standard.
  > :SOURce:JITTer:WANDgen:TDEV:DTYPE ITUT

:SOURce:JITTer:WANDgen:TDEV:DTYPE?

∇ Parameter None

- Response <type> = <CHARACTER PROGRAM DATA>
  Same as in the case of :SOURce:JITTer:WANDgen:TDEV:DTYPE.

- Function Queries the TDEV Mask standard.

- Example use > :SOURce:JITTer:WANDgen:TDEV:DTYPE?
  < ITUT

:SOURce:JITTer:WANDgen:TDEV:ITYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

G811	G.811-1997
S81T6	Section 8.1 Table 6 (G.812-1998)
S81T7	Section 8.1 Table 7 (G.812-1998)
S91T11	Section 9.1 Table 11 (G.812-1998)
S91T12	Section 9.1 Table 12 (G.812-1998)
SA31	Section A.3.1 (G.812-1998)
SA41	Section A.4.1 (G.812-1998)
SA5	Section A.5 (G.812-1998)
S71O1	Section 7.1 Option1 (G.813-1996)
S71O2	Section 7.1 Option2 (G.813-1996)

* "O" in S71O1 and S71O2 represents an alphabetical character, not numeric.

- Function Specifies the ITU-T TDEV Mask table for Manual WanderGen. measurement.

- Example use When specifying G.811-1997 as the ITU-T TDEV Mask table.
  > :SOURce:JITTer:WANDgen:TDEV:ITYPe G811

:SOURce:JITTer:WANDgen:TDEV:ITYPe?

∇ Parameter None

- Response <type> = <CHARACTER PROGRAM DATA>
  Same as in the case of :SOURce:JITTer:WANDgen:TDEV:ITYPe.

- Function Queries the ITU-T TDEV Mask table.

- Example use >:SOURce:JITTer:WANDgen:TDEV:ITYPe?
  <G811

:SOURce:JITTer:WANDgen:TDEV:ETYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

S721	Section 7.2.1 (ETS 300 462-3-1997)
S722	Section 7.2.2 (ETS 300 462-3-1997)
S723	Section 7.2.3 (ETS 300 462-3-1997)
S724	Section 7.2.4 (ETS 300 462-3-1997)
S61_4	Section 6.1 (ETS 300 462-4-1997)
S61_5	Section 6.1 (ETS 300 462-5-1996)
S72_4	Section 7.2 (ETS 300 462-4-1997)
S72_5	Section 7.2 (ETS 300 462-5-1996)
S8	Section 8 (ETS 300 462-4-1997)
ETS300_6	ETS 300 462-6-1997

- Function Specifies the ETSI TDEV Mask table for Manual Wander measurement.

- Example use When specifying Section 7.2.1 (ETS 300 462-3-1997) as the ETSI TDEV Mask table.
  > :SOURce:JITTer:WANDgen:TDEV:ETYPe S721

:SOURce:JITTer:WANDgen:TDEV:ETYPe?

∇ Parameter None

- Response <type> = <CHARACTER PROGRAM DATA>
  : Same as in the case of :SOURce:JITTer:WANDgen:TDEV:ETYPe.

- Function Queries the ETSI TDEV Mask table.

- Example use >:SOURce:JITTer:WANDgen:TDEV:ETYPe?
  <S721

:SOURce:JITTer:WANDgen:TDEV:ATYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

S722	Section 7.2.2 (ANSI TI.101-1994)
S732	Section 7.3.2 (ANSI TI.101-1994)
SD21	Section D.2.1 (ANSI TI.105.03-1994)
SD221	Section D.2.2.1 (ANSI TI.105.03-1994)
SD222	Section D.2.2.2 (ANSI TI.105.03-1994)
ANSIT1_9	ANSI T1.105.09-1996

- Function Specifies the ANSI TDEV Mask table for Manual Wander measurement.

- Example use When specifying ANSI T1.105.09-1996 as the ANSI TDEV Mask table.
  > :SOURce:JITTer:WANDgen:TDEV:ATYPe ANSIT1_9

:SOURce:JITTer:WANDgen:TDEV:ATYPe?

∇ Parameter None

- Response <type> = <CHARACTER PROGRAM DATA>
  Same as in the case of :SOURce:JITTer:WANDgen:TDEV:ATYPe.

- Function Queries the ETSI TDEV Mask table.

- Example use >:SOURce:JITTer:WANDgen:TDEV:ATYPe?
  < ANSIT1_9

:SOURce:JITTer:WANDgen:TDEV:BTYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

GR2830	GR-2830-CORE-1995
S43	Section 4.3 (GR-1244-CORE-1995)
S53	Section 5.3 (GR-1244-CORE-1995)
S54S2	Section 5.4 Strarum 2&3E (GR-1244-CORE-1995)
S54S3	Section 5.4 Strarum 3 (GR-1244-CORE-1995)
S54424F515	Section 5.4.4.2.4 Figure 5-15 (GR-253-CORE-1999)
S54425F516	Section 5.4.4.2.4 Figure 5-16 (GR-253-CORE-1999)
S54432	Section 5.4.4.3.2 (GR-253-CORE-1999)
S545	Section 5.4.5 (GR-253-CORE-1999)

- Function Specifies the Bellcore TDEV Mask table for Manual Wander measurement.

- Example use When specifying GR-2830-CORE-1995 as the Bellcore TDEV Mask table.
  > :SOURce:JITTer:WANDgen:TDEV:BTYPe GR2830

:SOURce:JITTer:WANDgen:TDEV:BTYPe?

∇ Parameter None

- Response <type> = <CHARACTER PROGRAM DATA>
  Same as in the case of :SOURce:JITTer:WANDgen:TDEV:BTYPe.

- Function Queries the ETSI TDEV Mask table.

- Example use >:SOURce:JITTer:WANDgen:TDEV:BTYPe?
  < GR2830

:SOURce:JITTer:WANDgen:MPDeviation

∇ Parameter < deviation > = <DECIMAL NUMERIC PROGRAM DATA>
1000 to 2000 Step value: 100 (ns)

- Function Specifies the deviation value when the Wander generation: Type is set to "Transient" for Manual Wander measurement.

- Example use When setting the deviation value to 1000.
> :SOURce:JITTer:WANDgen:MPDeviation 1000

:SOURce:JITTer:WANDgen:MPDeviation?

∇ Parameter None

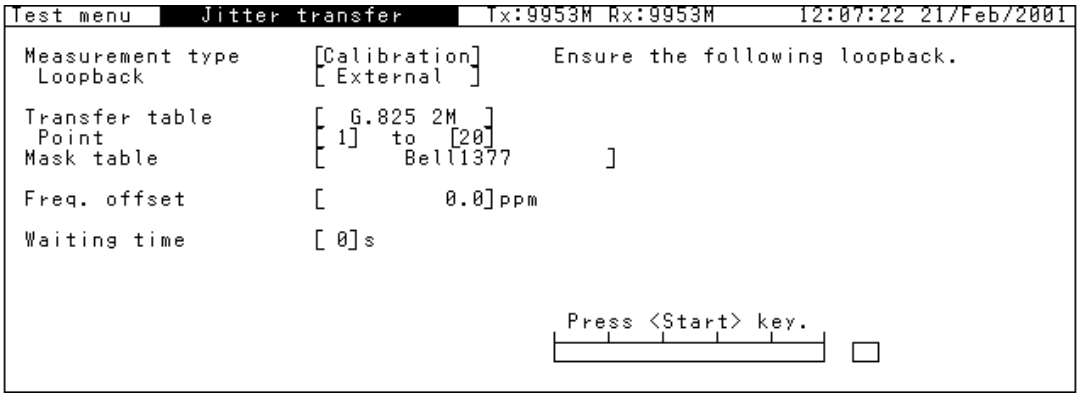
- Response <deviation> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:WANDgen:MPDeviation.

- Function Queries the deviation value when the Wander generation: Type is set to "Transient" for Manual Wander measurement.

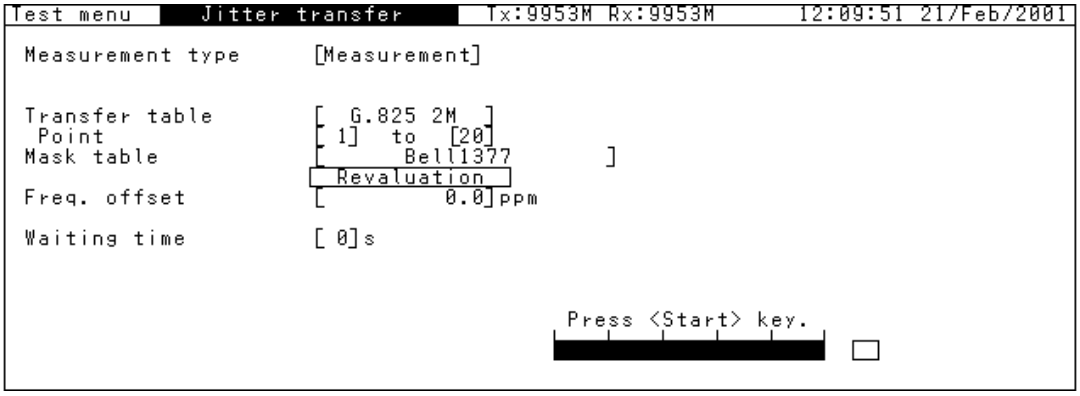
- Example use >:SOURce:JITTer:WANDgen:MPDeviation?
< 1000

11.2.2 Jitter transfer sub-screen

1) When Measurement type: Calibration selected.



2) When Measurement type: Measurement selected. (This screen appears when a measurement has completed.)



Setting Item	Command	Reference Page
Measurement type	:SENSe:JITTer:TRANsfer:MODE	11-105
Loopback	:SENSe:JITTer:TRANsfer:LOOPback	11-106
Transfer table	:SOURce:JITTer:TRANsfer:TYPE	11-107
Point (Start)	:SOURce:JITTer:TRANsfer:PTABle:COUNt	11-108
Point (End)		
Mask table	:SENSe:JITTer:TRANsfer:MASK	11-109
Revaluation	:SENSe:JITTer:TRANsfer:REValuation	11-110
Freq. offset	:SOURce:JITTer:TRANsfer:OFFSet	11-111
Waiting time	:SENSe:JITTer:TRANsfer:WTIMe	11-112

:SENSe:JITTer:TRANsfer:MODE

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

CAL	Calibration
MEAS	Measurement

- Function Specifies the measurement mode for the measurement of jitter transfer characteristics.
  Calibration starts when :SENSe:MEASure:STARt is transmitted following CAL setting.
  A measurement starts when :SENSe:MEASure:STARt is transmitted following MEAS setting.
  Be sure to perform calibration before measurement.

- Example use When setting the measurement mode to "CAL".
  > :SENSe:JITTer:TRANsfer:MODE CAL

:SENSe:JITTer:TRANsfer:MODE?

∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:JITTer:TRANsfer:MODE.

- Function Queries the measurement mode for the measurement of jitter transfer characteristics.

- Example use >:SENSe:JITTer:TRANsfer:MODE?
  < CAL

:SENSe:JITTer:TRANsfer:LOOPback

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

INTernal	Internal
EXTernal	External

- Function Switches between internal loopback and external connection for calibration measurement.

- Example use When specifying internal loopback for calibration measurement.
  > :SENSe:JITTer:TRANsfer:LOOPback INT

:SENSe:JITTer:TRANsfer:LOOPback?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:JITTer:TRANsfer:LOOPback.

- Function Queries switching between internal loopback and external connection for calibration measurement.

- Example use When querying switching between internal loopback and external connection for calibration measurement.
  > :SENSe:JITTer:TRANsfer:LOOPback?
  < INT

:SOURce:JITTer:TRANsfer:TYPE

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

G958A	G.958 Type A
G958B	G.958 Type B
G825_2M	G.825 2M
G825_1_5M	G.825 1.5M
G813	G.813
B1377	Bell 1377
B253	Bell 253
USER	User
USER2	User2

- Function Specifies the output table for jitter to occur during the measurement of jitter transfer characteristics.
  Whether or not a table can be set depends on the bit rate on the sending side.
  Refer to each standard for more information.

- Example use When setting the output table at G.958 Type A.
  > :SOURce:JITTer:TRANsfer:TYPE G958A

:SOURce:JITTer:TRANsfer:TYPE?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:TRANsfer:TYPE.

- Function Queries the output table for jitter to occur during the measurement of jitter transfer characteristics.

- Example use >:SOURce:JITTer:TRANsfer:TYPE?
  < G958A

:SOURce:JITTer:TRANsfer:PTABle:COUNt

∇ Parameter <numeric1>,<numeric2> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 20 Step value: 1

- Function Specifies the range of output points in the table for jitter to occur during the measurement of jitter transfer characteristics.

- Example use When setting the range of output points in the 2488M table between 2 and 10.
> :SOURce:JITTer:TRANsfer:PTABle:COUNt 2,10

:SOURce:JITTer:TRANsfer:PTABle:COUNt?

∇ Parameter None

- Response <numeric1>,<numeric2> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:JITTer:TRANsfer:PTABle:COUNt.

- Function Queries the range of output points in the table for jitter to occur during the measurement of jitter transfer characteristics.

- Example use > :SOURce:JITTer:TRANsfer:PTABle:COUNt?
< 2,10

:SENSe:JITTer:TRANsfer:MASK

∇ Parameter <mask> = <CHARACTER PROGRAM DATA>

G958A	G958 Type A
G958B	G958 Type B
ANSIT1A	ANSI T1.105.03 Type A
ANSIT1B	ANSI T1.105.03 Type B
B1377	Bell 1377
B253	Bell 253
USER	User

- Function Specifies the mask table for the measurement of jitter transfer characteristics. Whether or not a table can be set depends on the bit rate at the receiving end. Refer to each mask standard for more information.

- Example use When setting the mask table at G.958 Type A.
  > :SENSe:JITTer:TRANsfer:MASK G958A

:SENSe:JITTer:TRANsfer:MASK?

∇ Parameter None

- Response <mask> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:JITTer:TRANsfer:MASK.

- Function Queries the mask table for the measurement of jitter transfer characteristics.

- Example use > :SENSe:JITTer:TRANsfer:MASK?
  < G958A

:SENSe:JITTer:TRANsfer:REValuation

∇ Parameter	None
- Function	Re-evaluate the measurement of jitter transfer characteristics. When Test menu: Jitter transfer Masktable is set to "User", pressing this button performs re-evaluation when the table contents for Setup: Jitter transfer (Mask) Mask table: User are modified during the execution of a measurement. When the re-evaluation is completed, the process is automatically terminated. The re-evaluation can be specified to be performed only when required after modification of the set value. When no modification has been made to the user mask, re-evaluation is not performed even if this command is sent. Sending this command while no re-evaluation is required, a error "Setting conflict" is returned.
- Example use	When performing re-evaluation. > :SENSe:JITTer:TRANsfer:REValuation

:SOURce:JITTer:TRANsfer:OFFSet

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
-100.0 to 100.0 Step value: 0.1

- Function Specifies the frequency offset.
- Example use When setting the frequency offset to -50.3.
  > :SOURce:JITTer:TRANsfer:OFFSet –50.3

:SOURce:JITTer:TRANsfer:OFFSet?

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
  Same as in the case of :SOURce:JITTer:TRANsfer:OFFSet.
- Function Queries the value of frequency offset.
- Example use > :SOURce:JITTer:TRANsfer:OFFSet?
  < -50.3

:SENSe:JITTer:TRANsfer:WTIMe

∇ Parameter <wait> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 99 (s) Step value: 1

- Function Specifies the waiting time (waiting time before setting the next measurement point starts).

- Example use When setting the waiting time to 1 second.
> :SENSe:JITTer:TRANsfer:WTIMe 1

:SENSe:JITTer:TRANsfer:WTIMe?

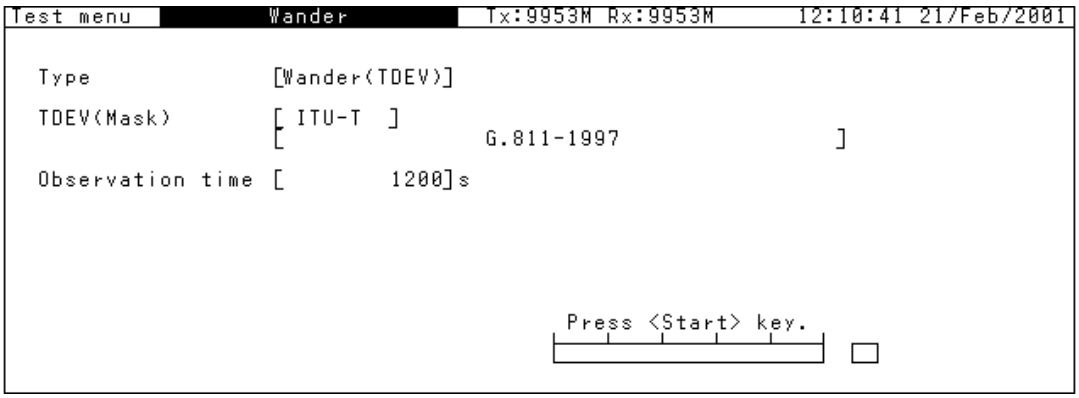
∇ Parameter None

- Response <wait> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:JITTer:TRANsfer:WTIMe.

- Function Queries the setting of the waiting time.

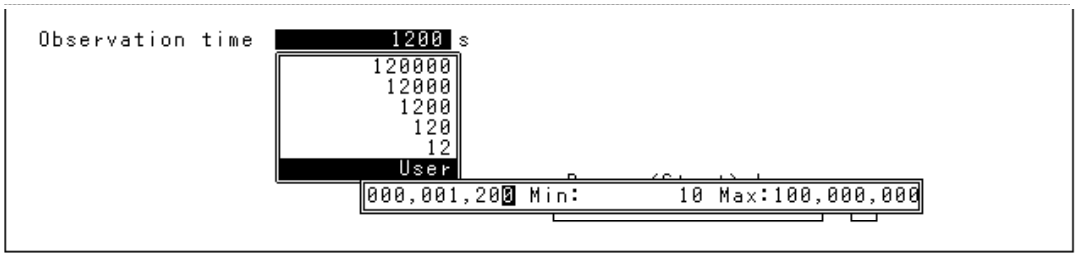
- Example use When querying the setting of the waiting time.
> :SENSe:JITTer:TRANsfer:WTIMe?
< 1

11.2.3 Wander sub-screen



The following screen appears when Type: Transient.

The following screen appears when Observation time: User.



Item	Command	Reference Page
Type	:SOURce:WANDer:AUTO:TYPE	11-114
Mask (TDEV)	:SOURce:WANDer:AUTO:TDEV:DTYPE	11-115
Mask (TDEV)table (ITU-T)	:SOURce:WANDer:AUTO:TDEV:ITYPE	11-116
Mask (TDEV)table (ETSI)	:SOURce:WANDer:AUTO:TDEV:ETYPE	11-117
Mask (TDEV)table (ANSI)	:SOURce:WANDer:AUTO:TDEV:ATYPE	11-118
Mask (TDEV)table (Bellcore)	:SOURce:WANDer:AUTO:TDEV:BTYPE	11-119
Maximum phase deviation	:SOURce:WANDer:AUTO:MPDeviation	11-120
Observation time	:SENSe:WANDer:AUTO:INTerval	11-121
Observation time: User	:SENSe:WANDer:AUTO:USER	11-122

:SOURce:WANDer:AUTO:TYPE

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

OFF	OFF
WANDer	Wander (TDEV)
TRANsient	Transient
SIGNal	Signal off

- Function Specifies the type of wander generation.

- Example use When setting the type of wander generation to "Wander (TDEV)".
  > :SOURce:WANDer:AUTO:TYPE WANDer

:SOURce:WANDer:AUTO:TYPE?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:AUTO:TYPE WANDer.

- Function Queries the setting of the type of wander generation.

- Example use When querying the setting of the type of wander generation.
  > :SOURce:WANDer:AUTO:TYPE WANDer?
  < WAND

:SOURce:WANDer:AUTO:TDEV:DTYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

ITUT	ITU-T
ETSI	ETSI
ANSI	ANSI
BELLcore	Bellcore
USER	User

- Function Selects the TDEV Mask standard for TDEV generation during automatic wander measurement.

- Example use When specifying ITU-T as the TDEV Mask standard.
  > :SOURce:WANDer:AUTO:TDEV:DTYPe ITUT

:SOURce:WANDer:AUTO:TDEV:DTYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:AUTO:TDEV:DTYPe.

- Function Queries the TDEV Mask standard.

- Example use > :SOURce:WANDer:AUTO:TDEV:DTYPe?
  < ITUT

:SOURce:WANDer:AUTO:TDEV:ITYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

G811	G.811-1997
S81T6	Section 8.1 Table 6 (G.812-1998)
S81T7	Section 8.1 Table 7 (G.812-1998)
S91T11	Section 9.1 Table 11 (G.812-1998)
S91T12	Section 9.1 Table 12 (G.812-1998)
SA31	Section A.3.1 (G.812-1998)
SA41	Section A.4.1 (G.812-1998)
SA5	Section A.5 (G.812-1998)
S71O1	Section 7.1 Option1 (G.813-1996)
S71O2	Section 7.1 Option2 (G.813-1996)

- Function Specifies the TDEV Mask table of the TDEV Mask standard (for ITU-T) during automatic wander measurement.

- Example use When specifying G.811-1997 as the ITU-T TDEV Mask table.
  > :SOURce:WANDer:AUTO:TDEV:ITYPe G811

:SOURce:WANDer:AUTO:TDEV:ITYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:AUTO:TDEV:ITYPe.

- Function Queries the TDEV Mask table of the TDEV Mask standard (for ITU-T) during automatic wander measurement.

- Example use > :SOURce:WANDer:AUTO:TDEV:ITYPe?
  < G811

:SOURce:WANDer:AUTO:TDEV:ETYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

S721	Section 7.2.1 (ETS 300 462-3-1997)
S722	Section 7.2.2 (ETS 300 462-3-1997)
S723	Section 7.2.3 (ETS 300 462-3-1997)
S724	Section 7.2.4 (ETS 300 462-3-1997)
S61_4	Section 6.1 (ETS 300 462-4-1997)
S61_5	Section 6.1 (ETS 300 462-5-1996)
S72_4	Section 7.2 (ETS 300 462-4-1997)
S72_5	Section 7.2 (ETS 300 462-5-1996)
S8	Section 8 (ETS 300 462-4-1997)
ETS300_6	ETS 300 462-6-1997

- Function Specifies the TDEV Mask table of the TDEV Mask standard (ETSI) during automatic wander measurement.

- Example use When specifying Section 7.2.1 (ETS 300 462-3-1997) as the ETSI TDEV Mask table .
  > :SOURce:WANDer:AUTO:TDEV:ETYPe S721

:SOURce:WANDer:AUTO:TDEV:ETYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:AUTO:TDEV:ETYPe.

- Function Queries the TDEV Mask table of the TDEV Mask standard (ETSI) during automatic wander measurement.

- Example use > :SOURce:WANDer:AUTO:TDEV:ETYPe?
  < S721

:SOURce:WANDer:AUTO:TDEV:ATYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

S722	Section 7.2.2 (ANSI TI.101-1994)
S732	Section 7.3.2 (ANSI TI.101-1994)
SD21	Section D.2.1 (ANSI TI.105.03-1994)
SD221	Section D.2.2.1 (ANSI TI.105.03-1994)
SD222	Section D.2.2.2 (ANSI TI.105.03-1994)
ANSIT1_9	ANSI T1.105.09-1996

- Function Specifies the TDEV Mask table of the TDEV Mask standard (for ANSI) during automatic wander measurement.

- Example use When specifying ANSI T1.105.09-1996 as the ANSI TDEV Mask table.
  > :SOURce:WANDer:AUTO:TDEV:ATYPe ANSIT1_9

:SOURce:WANDer:AUTO:TDEV:ATYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:AUTO:TDEV:ATYPe.

- Function Queries the TDEV Mask table of the TDEV Mask standard (for ANSI) during automatic wander measurement.

- Example use > :SOURce:WANDer:AUTO:TDEV:ATYPe?
  < ANSIT1_9

:SOURce:WANDer:AUTO:TDEV:BTYPe

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

GR2830	GR-2830-CORE-1995
S43	Section 4.3 (GR-1244-CORE-1995)
S53	Section 5.3 (GR-1244-CORE-1995)
S54S2	Section 5.4 Strarum 2&3E (GR-1244-CORE-1995)
S54S3	Section 5.4 Strarum 3 (GR-1244-CORE-1995)
S54424F515	Section 5.4.4.2.4 Figure 5-15 (GR-253-CORE-1999)
S54425F516	Section 5.4.4.2.4 Figure 5-16 (GR-253-CORE-1999)
S54432	Section 5.4.4.3.2 (GR-253-CORE-1999)
S545	Section 5.4.5 (GR-253-CORE-1999)

- Function Specifies the TDEV Mask table of the TDEV Mask standard (for Bellcore) during automatic wander measurement.

- Example use When specifying GR-2830-CORE-1995 as the Bellcore TDEV Mask table.
  > :SOURce:WANDer:AUTO:TDEV:BTYPe GR2830

:SOURce:WANDer:AUTO:TDEV:BTYPe?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SOURce:WANDer:AUTO:TDEV:BTYPe.

- Function Queries the TDEV Mask table of the TDEV Mask standard (for Bellcore) during automatic wander measurement.

- Example use > :SOURce:WANder:AUTO:TDEV:BTYPe?
  < GR2830

:SOURce:WANDer:AUTO:MPDeviation

∇ Parameter <deviation> = <DECIMAL NUMERIC PROGRAM DATA>
1000 to 2000 Step value: 100 (ns)

- Function Specifies the deviation value when the Wander generation: Type is set to "Transient" for automatic wander measurement.

- Example use When setting the deviation value to 1000.
> :SOURce:WANDer:AUTO:MPDeviation 1000

:SOURce:WANDer:AUTO:MPDeviation?

∇ Parameter None

- Response <deviation> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SOURce:WANDer:AUTO:MPDeviation.

- Function Queries the deviation value when the Wander generation: Type is set to "Transient" for automatic wander measurement.

- Example use >:SOURce:WANDer:AUTO:MPDeviation?
< 1000

:SENSe:WANDer:AUTO:INTerval

∇ Parameter <otime> = <CHARACTER PROGRAM DATA>

SEC120000	120000 sec.
SEC12000	12000 sec.
SEC1200	1200 sec.
SEC120	120 sec.
SEC12	12 sec.
USER	User

- Function Specifies the measurement time (observation time) for automatic wander measurement.

- Example use When setting the wander measurement time to 120 seconds.
  > :SENSe:WANDer:AUTO:INTerval SEC120

:SENSe:WANDer:AUTO:INTerval?

∇ Parameter None

- Response <otime> = <CHARACTER RESPONSE DATA>
  Same as in the case of :SENSe:WANDer:AUTO:INTerval.

- Function Queries the wander measurement time.

- Example use > :SENSe:WANDer:AUTO:INTerval?
  < SEC120

:SENSe:WANDer:AUTO:USER

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
10 to 100000000 Step value: 1

- Function Specifies the measurement time (observation time) for automatic wander measurement (When Interval is set to "User").
120,000 or more seconds can be set only by using application software.

- Example use When setting the wander measurement time to 120 seconds.
> :SENSe:WANDer:AUTO:USER120

:SENSe:WANDer:AUTO:USER?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :SENSe:WANDer:AUTO:INTerval.

- Function Queries the wander measurement time.

- Example use > :SENSe:WANDer:AUTO:USER?
< 120

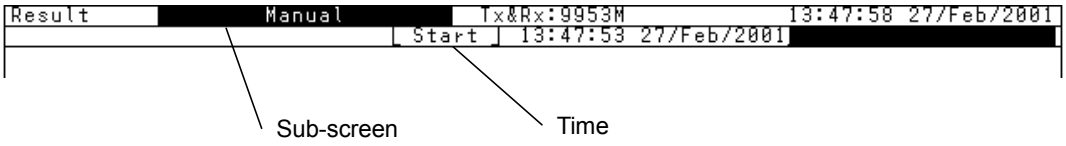
11.3 Result Main Screen

The Result screen displays the results of the measurement specified on the Test Menu screen.

The remote command can also be used to output the results of the measurement specified in the Test menu.

In case a setting which does not match the specified measurement is sent, an execution error "Setting conflict" is returned.

11.3.1 Command commonly used on Result screens



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:RESult[:NAME]	11-124
(Time) (Start)	:SENSe:MEASure:STIMe?	11-125
(Time) (Elapsed)	:SENSe:MEASure:ELAPsed?	11-125

Item	Command	Reference Page
Measurement results	:CALCulate:DATA?	11-126

:DISPlay:RESult[:NAME]

∇ Parameter <rdisplay> = <STRING PROGRAM DATA>

"MANual"	The Manual screen
"JTRansfer"	The Jitter Transfer screen
"WANDer"	The Wander screen

- Function Selects an item for display on the Result screen.
  The parameter to be selected is determined depending on which test is selected on the Test Menu screen.

- Example use When selecting "JTRansfer" as an item for display on the Result screen.
  (When Jitter transfer is selected on the Test Menu screen.)
  > :DISPlay:RESult:NAME"JTRansfer"
  or
  > :DISPlay:RESult"JTRansfer"

:DISPlay:RESult[:NAME]?

∇ Parameter None

- Response <rdisplay> = <CHARACTER RESPONSE DATA>

"MAN"	The Manual screen
"JTR"	The Jitter Transfer screen
"WAND"	The Wander screen

- Function Queries the measurement mode.

- Example use When querying the item for display on the Result screen.
  > :DISPlay:RESult:NAME?
  or
  > :DISPlay:RESult?
  <"JTR"

:SENSe:MEASure:STIMe?

∇ Parameter [<type>] = <CHARACTER PROGRAM DATA>

MANual	The Manual screen
JTRansfer	The Jitter Transfer screen
WANDer	The Wander screen

* When a parameter is omitted, queries the screen currently on display.

- Response <year>,<month>,<day>,<hour>,<minute>,<second>
  = <NR1 NUMERIC RESPONSE DATA>

  * In case no measurement has been performed, an output is made as shown below.
  < -,-,-,-,-,-

- Function Queries starting time of measurement.

- Example use > :SENSe:MEASure:STIMe?
  < 2000,10,5,19,50,34

:SENSe:MEASure:ELAPsed?

∇ Parameter [<type>] = <CHARACTER PROGRAM DATA>

MAN	The Manual screen
JTR	The Jitter Transfer screen
WAND	The Wander screen

* When a parameter is omitted, queries the screen currently on display.

- Response <day>,<hour>,<minute>,<second>
  = <NR1 NUMERIC RESPONSE DATA>

  * In case no measurement has been performed, an output is made as shown below.
  < -,-,-,-

- Function Queries the elapsed time of measurement.

- Example use When the time elapsed is 5 days and 19 hours, 50 minutes and 34 seconds.
  > :SENSe:MEASure:ELAPsed?
  < 5,19,50,34

:CALCulate:DATA?

∇ Parameter <string> = <STRING PROGRAM DATA>
"[CURRent:]<result>" Current measurement results
"LAST:<result>" Last measurement results

[CURRent:] can be omitted.
See the table on the following page for more information on <result>.

- Response <string> = <STRING RESPONSE DATA>
See the response format in the table on the following page.

* When the state of the Test Menu screen and the specified parameter do not match, a format indicating the "absence of relevant data" in response to the query made by the specified parameter is output.

- Function Queries the measurement results corresponding to the parameter.
This command is able to query the results of the target measurement even when the Result screen is not displayed.

- Example use When querying the results of Jitter transfer measurement.
> :CALCulate:DATA?"CURRent:JTRansfer:POINt1"
or
> :CALCulate:DATA?"JTRansfer:POINt1"
< "100, 10.00, 0.00, Acceptable"

When querying the results of Jitter transfer measurement during automatic wander measurement.
> :CALCulate:DATA?"JTRansfer:POINt1"
< "---, -----, ----, ----------"

The response format and the Example use regarding each measurement item are shown below.

Table Manual Jitter Measurement

Item	<result>	Response Format	Example use
Peak to Peak	"JAMPlitude:PTPeak"	Form5	> :CALCulate:DATA?"JAMPlitude:PTPeak" <" 0.234"
+Peak	"JAMPlitude:PPEak"	Form5	> :CALCulate:DATA?"JAMPlitude:PPEak" <" 7.00"
-Peak	"JAMPlitude:MPEak"	Form5	> :CALCulate:DATA?"JAMPlitude:MPEak" <" 22.23"
RMS	"JAMPlitude:RMS"	Form5	> :CALCulate:DATA?"JAMPlitude:RMS" <" 1224"
Hit Count	"JHIT:COUNt"	Form1	> :CALCulate:DATA?"JHIT:COUNt" <"1.0E+07"
Hit +Peak (Hit Second)	"JHIT:SECond"	Form1	> :CALCulate:DATA?"JHIT:SECond" <" 25453"
Hit %Free Second	"JHIT:FS"	Form3	> :CALCulate:DATA?"JHIT:FS" <" 20.0000"
Status (Tx Unlock)	"JMANual:TSTatus"	Form9	> :CALCulate:DATA?"JMANual:TSTatus" <"UNLOCK"
Status (Rx Unlock)	"JMANual:RSTatus"	Form9	> :CALCulate:DATA?"JMANual:RSTatus" <"UNLOCLK"
Status (Tx Clock loss)	"JMANual:TCLock"	Form14	> :CALCulate:DATA?"JMANual:TCLock" <"CLKLOSS"
Batch reading of data Peak to Peak +Peak -Peak RMS	"JAMPlitude:ADATa"	Form5, Form5, Form5, Form5	> :CALCulate:DATA?"JAMPlitude:ADATa" <" 0.234, 7.00, 22.23, 1224"
Batch reading of data Hit Count Hit +Peak (Hit Second) Hit %Free Second	"JHIT:ADATa"	Form1, Form1, Form3	> :CALCulate:DATA?"JHIT:ADATa" <"1.0E+07, 25453, 20.0000"
Batch reading of data Status (Tx Unlock) Status (Rx Unlock) Status (Tx Clock loss)	"JMANual:ADATa"	Form9, Form9, Form14	> :CALCulate:DATA?"JMANual:ADATa" <"UNLOCK,UNLOCK,CLKLOSS"

Table Manual Wander Measurement

Item		<result>	Response Format	Example use
Wander (DC-10 Hz)	Peak to Peak	"WANDer:PTPeak[:FULL]"	Form8	> :CALCulate:DATA?"WANDer:PTPeak[:FULL]" <" 5.0"
	+Peak	"WANDer:PPEak[:FULL]"	Form8	> :CALCulate:DATA?"WANDer:PPEak[:FULL]" <" 5000"
	-Peak	"WANDer:MPEak[:FULL]"	Form8	> :CALCulate:DATA?"WANDer:MPEak[:FULL]" <" 4.0E04"
	TIE	"WANDer:TIE[:FULL]"	Form7	> :CALCulate:DATA?"WANDer:TIE[:FULL]" <" 1.0E07"
Wander (DC-0.01 Hz)	Peak to Peak	"WANDer:PTPeak:LOW"	Form8	> :CALCulate:DATA?" WANDer:PTPeak:LOW" <" 5.0"
	+Peak	"WANDer:PPEak:LOW"	Form8	> :CALCulate:DATA?" WANDer:PPEak:LOW" <" 5000"
	-Peak	"WANDer:MPEak:LOW"	Form8	> :CALCulate:DATA?" WANDer:MPEak:LOW" <" 4.0E04"
	TIE	"WANDer:TIE:LOW"	Form7	> :CALCulate:DATA?" WANDer:TIE:LOW" <" 1.0E07"
Wander (0.01 Hz-10 Hz)	Peak to Peak	"WANDer:PTPeak:HIGH"	Form8	> :CALCulate:DATA?" WANDer:PTPeak:HIGH" <" 5.0"
	+Peak	"WANDer:PPEak:HIGH"	Form8	> :CALCulate:DATA?" WANDer:PPEak:HIGH" <" 5000"
	-Peak	"WANDer:MPEak:HIGH"	Form8	> :CALCulate:DATA?" WANDer:MPEak:HIGH" <" 4.0E04"
	TIE	"WANDer:TIE:HIGH"	Form7	> :CALCulate:DATA?" WANDer:TIE:HIGH" <" 1.0E07"
Reads out all the measurement results		"WANDer:ADATa"	Form8, Form8, Form8, Form7, Form8, Form8, Form8, Form7, Form8, Form8, Form8, Form7	> :CALCulate:DATA?" WANDer:ADATa" <" 5.0, 5000, 4.0E04, 1.0E07, 5.0, 5000, 4.0E04, 1.0E07, 5.0, 5000, 4.0E04, 1.0E07"

Table Jitter Transfer Measurement

Item	<result>	Response Format	Example use
Measurement results	"JTRansfer:POINt1" "JTRansfer:POINt2" : "JTRansfer:POINt19" "JTRansfer:POINt20"	(Frequency), (UIp-p), (dB), (Judgment) Form13, Form5, Form6, Form4	> :CALCulate:DATA?"JTRansfer:POINt1" <" 100, 10.00, 0.00, Acceptable"
Reads out all the obtained measurement results	"JTRansfer:POINt:ADATa"	(Point), (Frequency), (UIp-p), (dB), (Judgment) Form1, Form13, Form5, Form6, Form4	> :CALCulate:DATA?"JTRansfer:POINt:ADATa" <" 1,----------,------,-------,----------, 2,----------,------,-------,----------, 3,----------,------,-------,----------, 4,----------,------,-------,----------, 5,----------,------,-------,----------, 6,----------,------,-------,----------, 7,----------,------,-------,----------, 8,----------,------,-------,----------, 9,----------,------,-------,----------, 10,----------,------,-------,----------, 11,----------,------,-------,----------, 12, 100, 10.00, 0.00, Acceptable, 13, 200, 10.00, 0.00, Acceptable , 14, 240, 10.00, 0.00, Acceptable , 15, 10000, 10.00, 0.00, Acceptable , 16, 100, 10.00, 0.00, Acceptable , 17, 100, 10.00, 0.00, Acceptable, 18,----------,------,-------,----------, 19,----------,------,-------,----------, 20,----------,------,-------,----------"

Table Wander (TIE) Measurement (LOG)

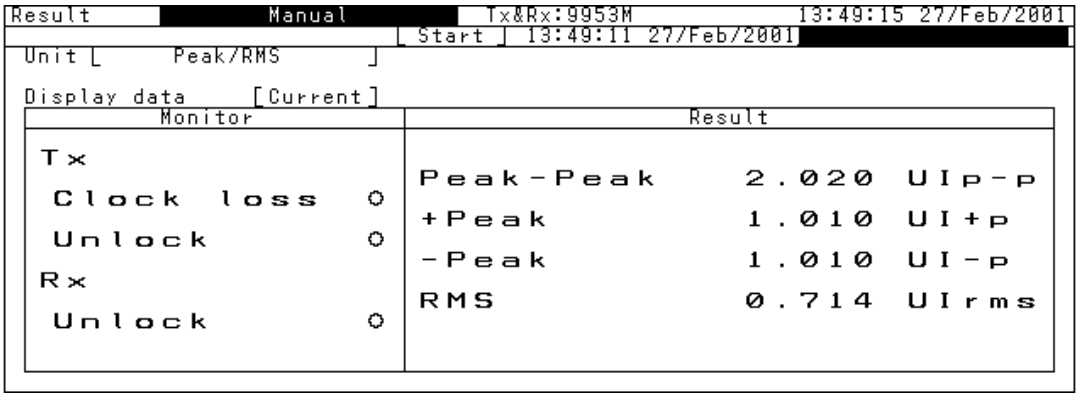
Item	<result>	Response Format	Example use
Measurement results	"WANDer:POINt1" "WANDer:POINt2" : "WANDer:POINt43" "WANDer:POINt44"	Form8’, Form8	> :CALCulate:DATA?"WANDer:POINt1" <" 0.10, 11"
Reads out all the measurement results	"WANDer:POINt:ADATa"	(No.) Form1, Form8’, Form8	>:CALCulate:DATA?"WANDer:POINt:ADATa" <" 1, 0.10, 11, 2, 0.15, 12, 3, 0.20, 13, 4, 0.25, 14, 5, 0.35, 15, 6, 0.50, 16, 7, 0.70, 19, 8, 1.0, 19, 9, 1.4, 19, 10, 1.9, 19, 11, 2.7, 20, 12, 3.7, 31, 13, 5.2, 50, 14, 7.2, 71, 15, 10, 83, 16, 12, 83"

Table Wander (TIE) Measurement (Linear)

Item	<result>	Response Format	Example use
Measurement results	"WANDer:LINear:POINt1" "WANDer:LINear:POINt2" : "WANDer:LINear:POINt119" "WANDer:LINear:POINt120"	Form8’, Form8	>:CALCulate:DATA?"WANDer:LINear:POINt1" <" 0.10, 11"
Reads out all the measurement results	"WANDer:LINear:POINt:ADATa"	(No.) Form1, Form8’, Form8	>:CALCulate:DATA?"WANDer:LINear:POINt:ADATa" <" 1, 0.10, 11, 2, 0.15, 12, 3, 0.20, 13, 4, 0.25, 14, 5, 0.35, 15, 6, 0.50, 16, 7, 0.70, 19, : : : 120, 120000, --------"

11.3.2 Manual sub-screen

a) Unit: Peak/RMS display screen



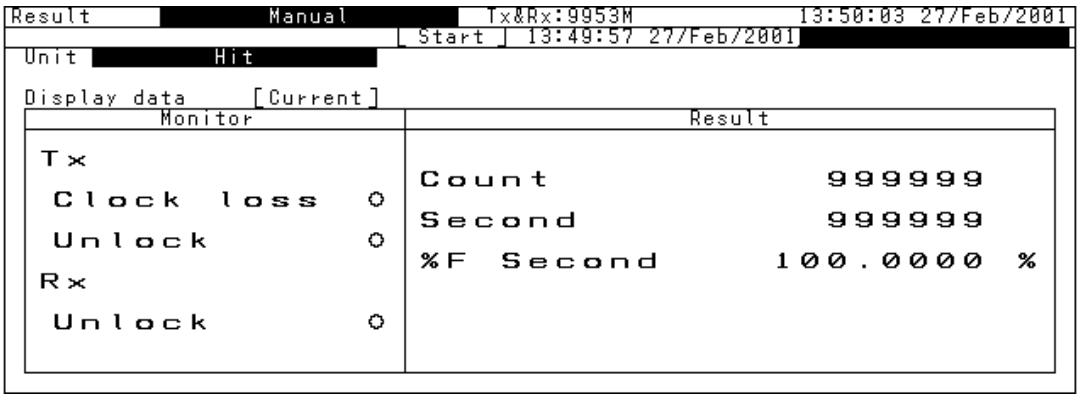
Setting Item	Command	Reference Page
Unit	:DISPlay:RESult:JWANder:UNIT	11-135
Display data	:DISPlay:RESult:JWANder:MODE	11-136

Setting Item	Command	Parameter	Reference Page
Monitor Tx Clock loss	:CALCulate:DATA?	"[CURRent:]JMANual:TCLock"	11-127
Monitor Tx Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:TSTatus"	11-127
Monitor Rx Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:RSTatus"	11-127
Result Peak-Peak	:CALCulate:DATA?	"[CURRent:]JAMPlitude:PTPeak" "LAST:JAMPlitude:PTPeak"	11-127
Result +Peak	:CALCulate:DATA?	"[CURRent:]JAMPlitude:PPEak" "LAST:JAMPlitude:PPEak"	11-127
Result -Peak	:CALCulate:DATA?	"[CURRent:]JAMPlitude:MPEak" "LAST:JAMPlitude:MPEak"	11-127
Result RMS	:CALCulate:DATA?	"[CURRent:]JAMPlitude:RMS" "LAST:JAMPlitude:RMS"	11-127

* The Unlock and Clock loss data show obtained monitored values.

* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

b) Unit: Hit display screen



Item	Command	Parameter	Reference Page
Monitor Tx Clock loss	:CALCulate:DATA?	"[CURRent:]JMANual:TCLock"	11-127
Monitor Tx Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:TSTatus"	11-127
Monitor Rx Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:RSTatus"	11-127
Rx Jitter Count	:CALCulate:DATA?	"[CURRent:]JHIT:COUNt" "LAST:JHIT:COUNt"	11-127
Rx Jitter +Peak	:CALCulate:DATA?	"[CURRent:]JHIT:SECond" "LAST:JHIT:SECond"	11-127
Rx Jitter %FSecond	:CALCulate:DATA?	"[CURRent:]JHIT:FS" "LAST:JHIT:FS"	11-127

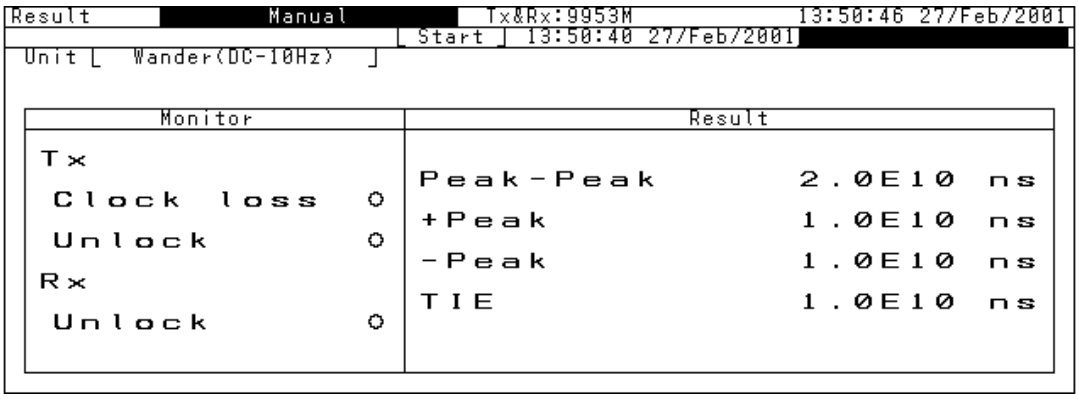
* Other commands are the same as those shown in "5.2.1 Unit: Peak/RMS display screen".

* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

- The Tx Jitter label, Clock loss lamp and Unlock lamp are not displayed when Clock on the Setup: Mapping screen is set to "Internal".

c) Unit: Wander ( ) display screen

1) When Unit is set to "Wander (DC-10 Hz)"



Setting Item		Command	<Parameter	Reference Page
Monitor Tx	Clock loss	:CALCulate:DATA?	"[CURRent:]JMANual:TCLock"	11-127
Monitor Tx	Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:TSTatus"	11-127
Monitor Rx	Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:RSTatus"	11-127
Rx Wander	Peak-Peak	:CALCulate:DATA?	"[CURRent:]WANDer:PTPeak[:FULL]"	11-128
Rx Wander	+Peak	:CALCulate:DATA?	"[CURRent:]WANDer:PPEak[:FULL]"	11-128
Rx Wander	-Peak	:CALCulate:DATA?	"[CURRent:]WANDer:MPEak[:FULL]"	11-128
Rx Wander	TIE	:CALCulate:DATA?	"[CURRent:]WANDer:TIE[:FULL]"	11-128

* Other commands are the same as those shown on the Unit: Peak/RMS display screen.

* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

2) When Unit is set to "Wander (DC-0.01 Hz)"

The same screen appears as in the case of Wander (DC-10 Hz).

Setting Item		Command	Parameter	Reference Page
Monitor Tx	Clock loss	:CALCulate:DATA?	"[CURRent:]JMANual:TCLock"	11-127
Monitor Tx	Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:TSTatus"	11-127
Monitor Rx	Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:RSTatus"	11-127
Rx Wander	Peak-Peak	:CALCulate:DATA?	"[CURRent:]WANDer:PTPeak:LOW"	11-128
Rx Wander	+Peak	:CALCulate:DATA?	"[CURRent:]WANDer:PPEak:LOW"	11-128
Rx Wander	-Peak	:CALCulate:DATA?	"[CURRent:]WANDer:MPEak:LOW"	11-128
Rx Wander	TIE	:CALCulate:DATA?	"[CURRent:]WANDer:TIE:LOW"	11-128

* Other commands are the same as those shown on the Unit: Peak/RMS display screen.

* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

3) When Unit is set to "Wander (0.01-10 Hz)"

The same screen appears as in the case of Wander (DC-10 Hz).

Setting Item	Command	Parameter	Reference Page
Monitor Tx Clock loss	:CALCulate:DATA?	"[CURRent:]JMANual:TCLock"	11-127
Monitor Tx Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:TSTatus"	11-127
Monitor Rx Unlock	:CALCulate:DATA?	"[CURRent:]JMANual:RSTatus"	11-127
Rx Wander Peak-Peak	:CALCulate:DATA?	"[CURRent:]WANDer:PTPeak:HIGH"	11-128
Rx Wander +Peak	:CALCulate:DATA?	"[CURRent:]WANDer:PPEak:HIGH"	11-128
Rx Wander -Peak	:CALCulate:DATA?	"[CURRent:]WANDer:MPEak:HIGH"	11-128
Rx Wander TIE	:CALCulate:DATA?	"[CURRent:]WANDer:TIE:HIGH"	11-128

* Other commands are the same as those shown on the Unit: Peak/RMS display screen.
* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

:DISPlay:RESult:JWANder:UNIT

∇ Parameter <unit> = <CHARACTER PROGRAM DATA>

PEAKRMS	Peak/RMS
HIT	Hit
WANDer1	Wander (DC-10 Hz)
WANDer2	Wander (DC-0.01 Hz)
WANDer3	Wander (0.01 Hz-10 Hz)

- Function Specifies the format for displaying the Jitter/Wander measurement results on the Result: Manual screen.

- Example use When setting the data to be displayed on the Result screen to "Peak/RMS."
  > :DISPlay:RESult:JWANder:UNIT PEAKRMS

:DISPlay:RESult:JWANder:UNIT?

∇ Parameter None

- Response <unit> = <CHARACTER RESPONSE DATA>

PEAKRMS	Peak/RMS
HIT	Hit
WAND1	Wander (DC-10 Hz)
WAND2	Wander (DC-0.01 Hz)
WAND3	Wander (0.01 Hz-10 Hz)

- Function Queries the format for displaying the Jitter/Wander measurement results on the Result: Manual screen.

- Example use When querying the data to be displayed on the Result: Manual screen.
  > :DISPlay:RESult:JWANder:UNIT?
  < PEAKRMS

:DISPlay:RESult:JWANder:MODE

∇ Parameter <mode> = <CHARACTER PROGRAM DATA>

CURRent	Current measurement results
LAST	Last measurement results

- Function Specifies the type (current-display/ended-measurement) of Manual (Jitter/Wander) measurement result display.

- Example use When setting the Result screen display mode to "CURRent."
  > :DISPlay:RESult:JWANder:MODE CURRent

:DISPlay:RESult:JWANder:MODE?

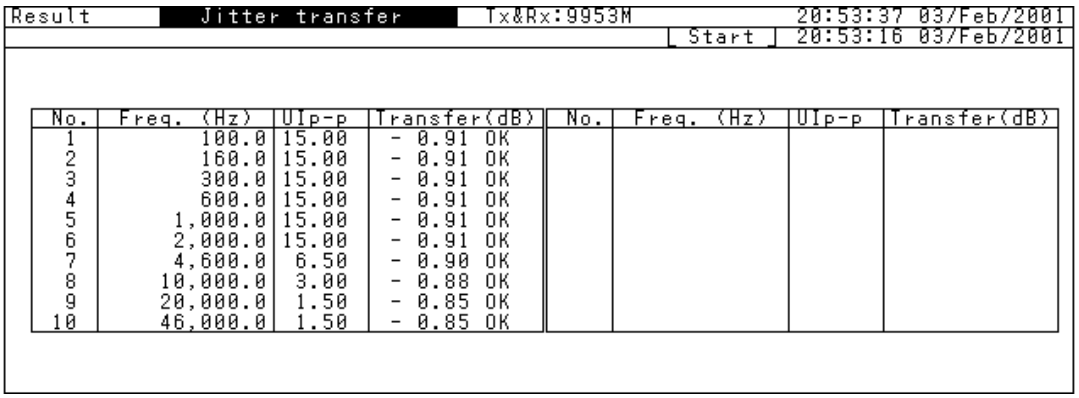
∇ Parameter None

- Response <mode> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:RESult:JWANder:MODE.

- Function Queries the type (current display/ended measurement) of Manual (Jitter/Wander) measurement result display.

- Example use When querying the type of result display on the Result screen.
  > :DISPlay:RESult:JWANder:MODE?
  < CURR

11.3.3 Jitter transfer sub-screen



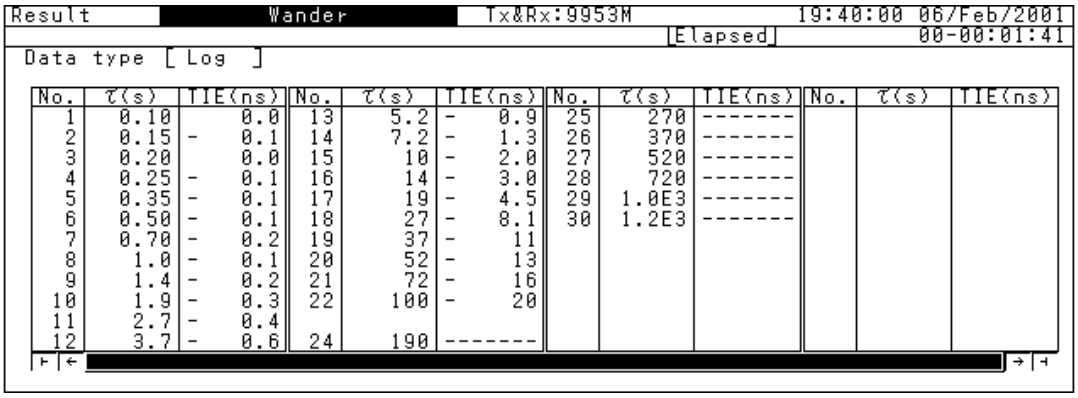
No.	Freq. (Hz)	UIp-p	Transfer(dB)	No.	Freq. (Hz)	UIp-p	Transfer(dB)
1	100.0	15.00	- 0.91 OK				
2	160.0	15.00	- 0.91 OK				
3	300.0	15.00	- 0.91 OK				
4	600.0	15.00	- 0.91 OK				
5	1,000.0	15.00	- 0.91 OK				
6	2,000.0	15.00	- 0.91 OK				
7	4,600.0	6.50	- 0.90 OK				
8	10,000.0	3.00	- 0.88 OK				
9	20,000.0	1.50	- 0.85 OK				
10	46,000.0	1.50	- 0.85 OK				

Item	Command	Parameter	Reference Page
No. Freq. UIp-p Transfer	:CALCulate:DATA?	"[CURRent:]JTRansfer:POINt1" to "[CURRent:]JTRansfer:POINt20"	11-129

* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

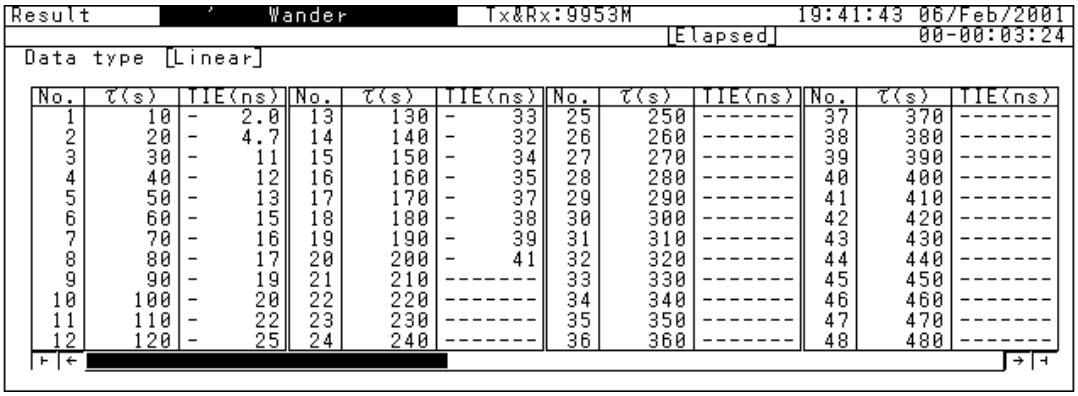
11.3.4 Wander sub-screen

d) When Data type is set to Log



No.	τ(s)	TIE(ns)	No.	τ(s)	TIE(ns)	No.	τ(s)	TIE(ns)	No.	τ(s)	TIE(ns)
1	0.10	0.0	13	5.2	- 0.9	25	270	-------			
2	0.15	- 0.1	14	7.2	- 1.3	26	370	-------			
3	0.20	0.0	15	10	- 2.0	27	520	-------			
4	0.25	- 0.1	16	14	- 3.0	28	720	-------			
5	0.35	- 0.1	17	19	- 4.5	29	1.0E3	-------			
6	0.50	- 0.1	18	27	- 8.1	30	1.2E3	-------			
7	0.70	- 0.2	19	37	- 11						
8	1.0	- 0.1	20	52	- 13						
9	1.4	- 0.2	21	72	- 16						
10	1.9	- 0.3	22	100	- 20						
11	2.7	- 0.4									
12	3.7	- 0.6	24	190	-------						

e) When Data type is set to Linear



No.	τ(s)	TIE(ns)	No.	τ(s)	TIE(ns)	No.	τ(s)	TIE(ns)	No.	τ(s)	TIE(ns)
1	10	- 2.0	13	130	- 33	25	250	-------	37	370	-------
2	20	- 4.7	14	140	- 32	26	260	-------	38	380	-------
3	30	- 11	15	150	- 34	27	270	-------	39	390	-------
4	40	- 12	16	160	- 35	28	280	-------	40	400	-------
5	50	- 13	17	170	- 37	29	290	-------	41	410	-------
6	60	- 15	18	180	- 38	30	300	-------	42	420	-------
7	70	- 16	19	190	- 39	31	310	-------	43	430	-------
8	80	- 17	20	200	- 41	32	320	-------	44	440	-------
9	90	- 19	21	210	-------	33	330	-------	45	450	-------
10	100	- 20	22	220	-------	34	340	-------	46	460	-------
11	110	- 22	23	230	-------	35	350	-------	47	470	-------
12	120	- 25	24	240	-------	36	360	-------	48	480	-------

Setting Item	Command	Reference Page
Data type	:DISPlay:RESult:WANDer:DISPtype	11-139

Item	Command	Parameter	Reference Page
No. τ TIE	:CALCulate:DATA? (When Data type is set to Log)	"[CURRent:]WANDer:POINt1" to "[CURRent:]WANDer:POINt44"	11-130
No. τ TIE	:CALCulate:DATA? (When Data type is set to Linear)	"[CURRent:]WANDer:LINear:POINt1" to "[CURRent:]WANDer:LINear:POINt120"	

* See "11.3.1 Command commonly used on Result screens" for more information on the :CALCulate:DATA? command.

:DISPlay:RESult:WANDer:DISPtype

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

LOG	log
LINear	linear

- Function Specifies the type of wander data.

- Example use When displaying the LOG results.
  > :DISPlay:RESult:WANDer LOG

:DISPlay:RESult:WANDer:DISPtype?

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>

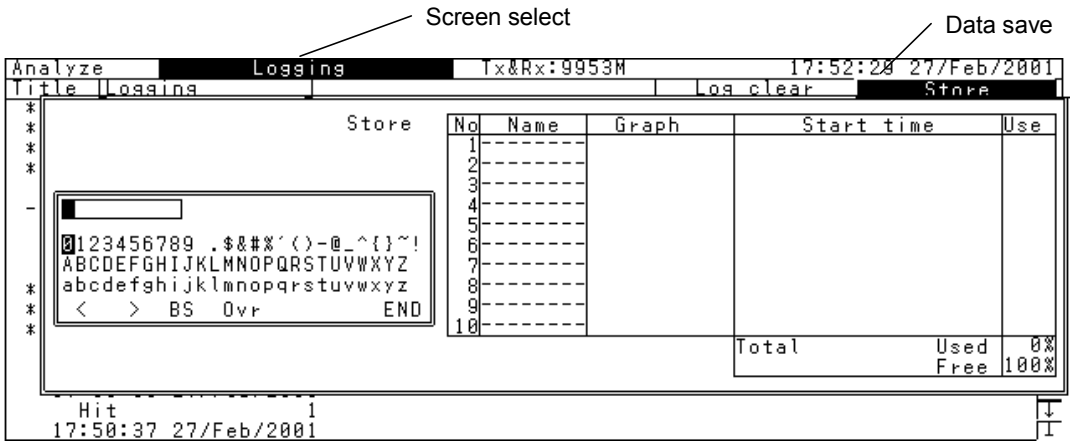
LOG	log
LIN	linear

- Function Queries the type of wander data display.
- Example use > :DISPlay:RESult:WANDer:DISPtype?
  < LIN

11.4 Analyze Main Screen

11.4.1 Analyze-screen common command

- The memory save window when Store is executed is shown as follows:



Analyze screen common command

Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph	:SYSTem:MEMory:ANALysis:STORe	11-142
(Store key) Name		

Note: :SYSTEM:MEMory:ANALysis:STORe can deal with the following data that contain graphs:

LOGGing	Logging data
PEAKjitter	Peak jitter data
JTRansfer	Jitter transfer characteristic
WANDer	Wander data
FREQuency	Frequency graph data

:DISPlay:ANALysis[:NAME]

Setting command

∇ Parameter <adisplay> = <STRING PROGRAM DATA>

"LOGGing"	The Logging screen
"PEAKjitter"	The Peak Jitter screen
"FMONitor"	The Frequency monitor screen
"FREQuency"	The Frequency graph screen
"JTRansfer"	The Jitter transfer screen
"WANDer"	The Wander screen
"RECall"	The Recall screen

- Function Selects a display item on the Analyze screen.

- Example use Select "FMONitor" as a display item on the Analyze screen.

```
> :DISPlay:ANALysis:NAME"FMONitor"
        or
> :DISPlay:ANALysis"FMONitor"
```

:DISPlay:ANALysis[:NAME]?

Query command

∇ Parameter None

- Response <adisplay> = <STRING RESPONSE DATA>

"LOGG"	The Logging screen
"PEAK"	The Peak Jitter screen
"FMON"	The Frequency monitor screen
"FGR"	The Frequency graph screen
"JTR"	The Jitter transfer screen
"WAND"	The Wander screen
"REC"	The Recall screen

- Function Queries a display item on the Analyze screen.

- Example use

```
> :DISPlay:ANALysis:NAME?
        or
> :DISPlay:ANALysis?
<"FMON"
```

:SYSTem:MEMory:ANALysis:STORe

Setting command

∇ Parameter

<type> = <CHARACTER PROGRAM DATA>

LOGGing	The LOGGing screen
PEAKjitter	The Peak jitter data
JTRansfer	The Jitter transfer characteristic
WANDer	The Wander data
FREQuency	The Frequency graph data

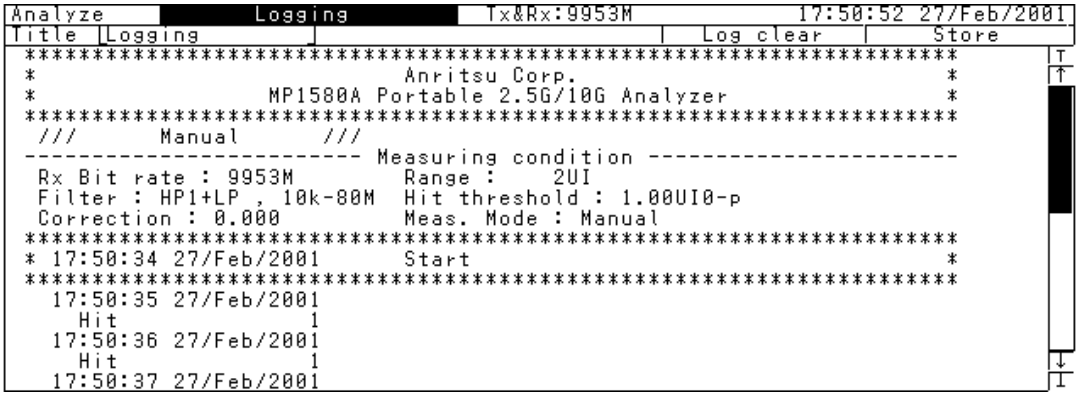
<string> = <STRING PROGRAM DATA>

"ABCabc..." Data name (max. 8 characters excluding "")

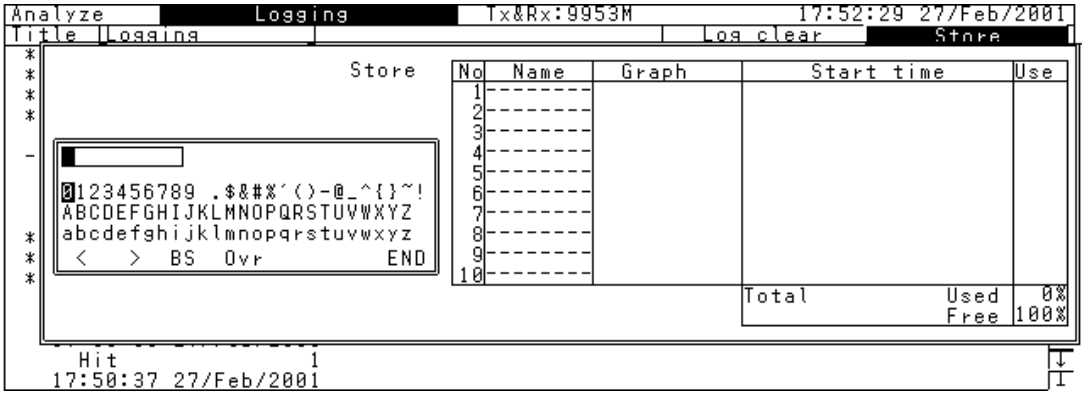
- Function Writes the analyze data in the memory.

- Example use Write the analyze data (jitter transfer characteristic) with a name "Demo1".
  > :SYSTem:MEMory:ANALysis:STORe JTRansfer,"Demo1"

11.4.2 Logging sub-screen



Store



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph (Store key) Name	:SYSTem:MEMory:ANALysis:STORe	11-142
Title	:DISPlay:ANALysis:LOGGing:TITLe	11-144
Log clear	:SENSe:LOGGing:CLEar	11-145
(Read Log data contents)	:CALCulate:LOGGing:DATA?	11-146
(Read all the Log data contents)	:CALCulate:LOGGing:ADATa?	11-147
Graph scroll ←, →, T, ⊥	:DISPlay:ANALysis:LOGGing:SCRoll	11-148

Refer to Analyze-screen common command for the following commands:
- :DISPlay:ANALysis[:NAME]
- :SYSTem:MEMory:ANALysis:STORe

:DISPlay:ANALysis:LOGGing:TITLe

Setting command

∇ Parameter <title> = <STRING PROGRAM DATA>
"Title string" Title string (max. 15 characters)
The string length is 0 to 15 characters.
Entry character: 0 to 9,A to Z,a to z, $&#%'.()-@_^{}^!

Spaces are added when the string is less than 15 characters.

- Function Sets a title on the Analyze: Logging screen.

- Example use Display "TITLE-LOG" as a title.
> :DISPlay:ANALysis:LOGGing:TITLe"TITLE-LOG"

:DISPlay:ANALysis:LOGGing:TITLe?

Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:LOGGing:TITLe

- Function Queries a title on the Analyze: Logging screen.
- Example use > :DISPlay:ANALysis:LOGGing:TITLe?
< "TITLE-LOG "

:DISPlay:ANALysis:LOGGing:CLEar

Setting command

∇ Parameter None

- Function Clear the log data displayed on the Analyze: Logging screen.

- Example use Start to read the log data.
  > :DISPlay:ANALysis:LOGGing:CLEar

:CALCulate:LOGGing:DATA?

Query command

∇ Parameter <start> = <DECIMAL NUMERIC PROGRAM DATA>
1 to the last line (step: 1): beginning line of data read
<end> = <DECIMAL NUMERIC PROGRAM DATA>
1 to the last line (step: 1) : beginning line of data read
-1 : obtains the log data up to the ending line.

- Response <logdata> = <STRING RESPONSE DATA>
"(log data contents of one line: 70 characters)"
When no data exists, "No data" is returned.

- Function Queries contents of the log data on the Analyze: Logging screen.

- Restriction When an <end> value is smaller than a <start> value, a numeric of <end> is assumed to be the same as the <start> value.
When an <end> value is larger than data that can be obtained, END is returned next to the last line of log data.

- Example use Specifies data from 1st line to 15th line, and read out them.
(Since the last data is 12th line, "END" is appended at the last.)
> :CALCulate:LOGGing:DATA? 1,15

```
<"************************************************************",
"*                        Anritsu Corp.                       *",
"*             MP1580A Portable 2.5G/10G Analyzer             *",
"************************************************************",
" ///                          Manual                        ///",
"--------------------- Measuring condition ------------------",
" Rx Bit rate : 9953M        Range :    2UI                    ",
" Filter : HP1+LP , 10k-80M  Hit threshold : 1.00UI0-p         ",
" Correction : 0.000          Meas. Mode : Repeat  1s          ",
"************************************************************",
"* 06:57:37 10/Feb/2000     Start                            *",
"************************************************************",
"END"
```

:CALCulate:LOGGing:ADATa? Query command

∇ Parameter <all> = <DECIMAL NUMERIC PROGRAM DATA>
-1 : Obtains the log data up to the ending line.

- Response <logdata> = <STRING RESPONSE DATA>
"(log data contents of one line)"
When no data exists, "No data" is returned.
END is returned next to the last line of the obtained data.

- Function Queries contents of all the log data on the Analyze: Logging screen.
- Example use Queries data contents.

> :CALCulate:LOGGing:ADATa? -1

```
<"*****************************************************",
"*                    Anritsu Corp.                    *",
"*        MP1580A Portable 2.5G/10G Analyzer           *",
"*****************************************************",
" ///                    Manual                     ///",
"----------------- Measuring condition ----------------",
" Rx Bit rate : 9953M       Range :   2UI              ",
" Filter : HP1+LP , 10k-80M  Hit threshold : 1.00UI0-p ",
" Correction : 0.000        Meas. Mode : Repeat  1s    ",
"*****************************************************",
"* 06:57:37 10/Feb/2000   Start                      *",
"*****************************************************",
"END"
```

:DISPlay:ANALysis:LOGGing:SCRoll

Setting command

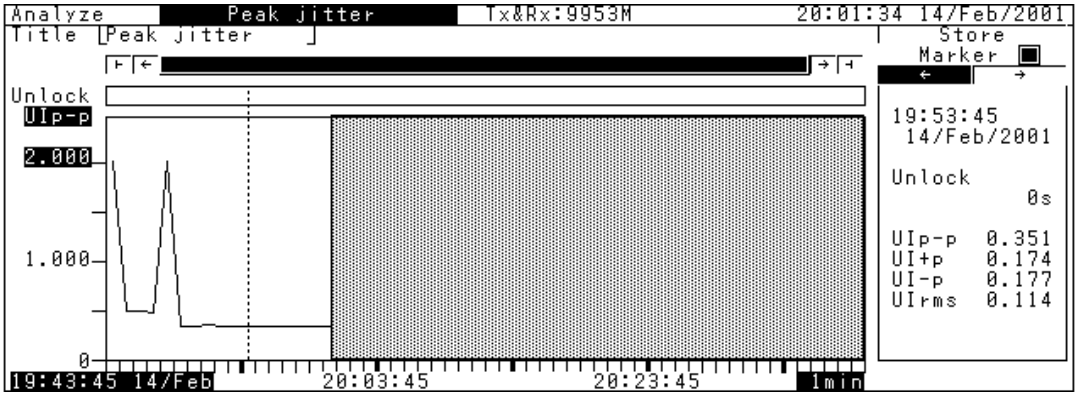
▽ Parameter <type> = <CHARACTER PROGRAM DATA>

TOP	Jumps to the data position of logging start.
END	Jumps to the data position of logging end.
NEXT	Scrolls down one screen.
BEFOR	Scrolls up one screen.

- Function Specifies graph scroll on the Analyze: Logging screen.

- Example use Scroll up one screen.
  > :DISPlay:ANALysis:LOGGing:SCRoll BEFOR

11.4.3 Peak jitter sub-screen



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph (Store key) Name	:SYSTem:MEMory:ANALysis:STORe	11-142
Title	:DISPlay:ANALysis:PEAK:TITLe	11-150
Graph scroll ←, →, -|, |-	:DISPlay:ANALysis:PEAK:SCRoll	11-151
Result display type	:DISPlay:ANALysis:PEAK:DTYPe	11-152
Scale	:DISPlay:ANALysis:PEAK:SCALe	11-153
Display start position (time)	:DISPlay:ANALysis:PEAK:FROM	11-154
Interval (time) Interval (unit)	:DISPlay:ANALysis:PEAK:INTerval	11-155
Marker key	:DISPlay:ANALysis:PEAK:MDISplay	11-156
In a marker field <,>	:DISPlay:ANALysis:PEAK:MARKer	11-157
(Data in a marker field) time (Data in a marker field) Unlock (Data in a marker field) UIp-p (Data in a marker field) UI+p (Data in a marker field) UI-p (Data in a marker field) UIrms	:DISPlay:ANALysis:PEAK:DATA?	11-158
(Peak jitter data)	:CALCulate:PEAK:DATA?	11-158

See the section on the common commands to Analyze screens for the following commands:
-:DISPlay:ANALysis[:NAME]
-:SYSTem:MEMory:ANALysis:STORe

:DISPlay:ANALysis:PEAK:TITLe

Setting command

∇ Parameter <title> = <STRING PROGRAM DATA>
"Title string" Title string (max. 15 characters)
The string length is 0 to 15 characters.
Entry character: 0 to 9,A to Z,a to z, $&#%'.()-@_^{}^!

Spaces are added when the string is less than15 characters.

- Function Sets a title on the Analyze: Peak jitter screen.

- Example use Display "TITLE-DISP" as a title of a jitter transfer characteristic measurement.
> :DISPlay:ANALysis:PEAK:TITLe"TITLE-DISP"

:DISPlay:ANALysis:PEAK:TITLe?

Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:TITLe.

- Function Queries a title on the Analyze: Peak jitter screen.

- Example use > :DISPlay:ANALysis:PEAK:TITLe?
< "TITLE-DISP "

:DISPlay:ANALysis:PEAK:SCRoll

Setting command

▽ Parameter <type> = <CHARACTER PROGRAM DATA>

LEFT	Scrolls left.
RIGHt	Scrolls right.
TOP	Moves to the top.
BOTTom	Moves to the bottom.

- Function Specifies graph scroll on the Analyze: Peak jitter screen.

- Example use Scroll in the right direction.
  > :DISPlay:ANALysis:PEAK:SCRoll RIGHt

:DISPlay:ANALysis:PEAK:DTYPE

Setting command

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

UIPTp	UIp-p
UIPP	UI+p
UIMP	UI-p
UIRMs	UIrms

- Function Sets a display type of a graph on the Analyze: Peak jitter screen.

- Example use Set the screen type to UIp-p.
  > :DISPlay:ANALysis:PEAK:DTYPe UIPTp

:DISPlay:ANALysis:PEAK:DTYPE?

Query command

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>

UIPT	UIp-p
UIPP	UI+p
UIMP	UI-p
UIRM	UIrms

- Function Queries the display type of a graph on the Analyze: Analyze: Peak jitter screen.

- Example use > :DISPlay:ANALysis:PEAK:DTYPe?
  < UIRM

:DISPlay:ANALysis:PEAK:SCALe

Setting command

∇ Parameter <scale> = <DECIMAL NUMERIC PROGRAM DATA>
0.002 to 4000.0 Refer to the following for a step.

Setting range by Rx Range

Rx Range	UIp-p	UI+p	UI-p	UIrms
4000 UI	4 to 4000 / 4 (step)	4 to 2000 / 4 (step)	4 to 2000 / 4 (step)	Setting disabled
1000 UI	2 to 1000 / 2 (step)	2 to 500 / 2 (step)	2 to 500 / 2 (step)	Setting disabled
20 UI	0.02 to 20.00 / 0.02 (step)	0.02 to 10.00 / 0.02 (step)	0.02 to 10.00 / 0.02 (step)	0.02 to 7.00 / 0.02 (step)
2 UI	0.002 to 2.000 / 0.002 (step)	0.002 to 1.000 / 0.002 (step)	0.002 to 1.000 / 0.002 (step)	0.002 to 0.700 / 0.002 (step)

- Function Sets the y axis scale of a graph on the Analyze: Peak jitter screen.

- Example use Set the scale to 0.02.
> :DISPlay:ANALysis:PEAK:SCALe 0.02

:DISPlay:ANALysis:PEAK:SCALe?

Query command

∇ Parameter None

- Response <scale> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:SCALe.

- Function Queries the y axis scale of a graph on the Analyze: Peak jitter screen.
- Example use > :DISPlay:ANALysis:PEAK:SCALe?
< 100.0

:DISPlay:ANALysis:PEAK:FROM

Setting command

∇ Parameter <numeric1>,<numeric2>,<numeric3>,<numeric4>,<numeric5> [,<numeric6>]
<DECIMAL NUMERIC PROGRAM DATA>
<numeric1> = 2000 to 2093 (year)
<numeric2> = 1 to 12 (month)
<numeric3> = 1 to 31 (day)
<numeric4> = 0 to 23 (hour)
<numeric5> = 0 to 59 (minute)
<numeric6> = 0 to 59 (second)

* When measured data exists:
If the time specified with a parameter does not exist, the time next to the specified time is set.
Specifying the time before the measurement start time sets the measurement start time.
Specifying the time after the log end time sets the log end time.
When <numeric6> is not specified, 0 is set.
* When measured data does not exist (when the screen displays "--:--:-- --/--"), setting the time issues an execution error "Setting conflict."

- Function Sets the display start position of a graph on the Analyze: Peak jitter screen.

- Example use Start display from 11:30:40, July 28, 2000.
> :DISPlay:ANALysis:PEAK:FROM 2000,7,28,11,30,40

:DISPlay:ANALysis:PEAK:FROM?

Query command

∇ Parameter None

- Response <numeric1>,<numeric2>,<numeric3>,<numeric4>,<numeric5>,<numeric6>
= <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:FROM

* When the Peak jitter analyze data does not exist, the following is output:
< -,-,-,-,-,-

- Function Queries the time of the display start position of a graph on the Analyze: Peak jitter screen.
- Example use > :DISPlay:ANALysis:PEAK:FROM?
< 2000,7,28,11,30,40

:DISPlay:ANALysis:PEAK:INTerval

Setting command

∇ Parameter <numeric> = <CHARACTER PROGRAM DATA>
1, 15, 60
<suffix> = <CHARACTER PROGRAM DATA>
M minute

- Function Sets the width of one time axis scale of a graph on the Analyze: Peak jitter screen.

- Example use Set the width for one scale to 1 minute.
> :DISPlay:ANALysis:PEAK:INTerval 1,M

:DISPlay:ANALysis:PEAK:INTerval?

Query command

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:INTerval.
<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:INTerval.

- Function Queries the width of one time axis scale of a graph on the Analyze: Peak jitter screen.

- Example use > :DISPlay:ANALysis:PEAK:INTerval?
< 1,M

:DISPlay:ANALysis:PEAK:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets the marker to OFF
ON or 1	Sets the marker to ON

- Function Sets the marker display on the Analyze: Peak jitter screen.
- Example use Set the marker display of a jitter transfer characteristic measurement to ON.
  > :DISPlay:ANALysis:PEAK:MDISplay ON

:DISPlay:ANALysis:PEAK:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Marker is set to OFF
1	Marker is set to ON

- Function Queries the marker display on the Analyze: Peak jitter screen.
- Example use > :DISPlay:ANALysis:PEAK:MDISplay?
  < 1

:DISPlay:ANALysis:PEAK:MARKer

Setting command

∇ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Peak jitter screen.

- Example use Move in the right direction.
  > :DISPlay:ANALysis:PEAK:MARKer RIGHt

:DISPlay:ANALysis:PEAK:DATA? Query command

∇ Parameter None

- Response <time>,<unlock>,<ui>,<uipp>,<uimp>,<uirms>
  <time> = <year>,<month>,<day>,<hour>,<minute>,<second>
  Time indicated with the marker
  <year> = <NR1 NUMERIC RESPONSE DATA>
  0, 2000 to 2093 (year)
  <month> = <NR1 NUMERIC RESPONSE DATA>
  0, 1 to 12 (month)
  <day> = <NR1 NUMERIC RESPONSE DATA>
  0, 1 to 31 (day)
  <hour> = <NR1 NUMERIC RESPONSE DATA>
  0 to 23 (hour)
  <minute> = <NR1 NUMERIC RESPONSE DATA>
  0 to 59 (minute)
  <second> = <NR1 NUMERIC RESPONSE DATA>
  0 to 59 (in seconds)
  <unlock> = <STRING RESPONSE DATA>
  Time when the marker-indicated data is unlocked
  Form1
  <ui> = <STRING RESPONSE DATA>
  Jitter value of the marker-indicated data (UIp-p)
  Form1 or Form2 (depending on the display scale)
  <uipp> = <STRING RESPONSE DATA>
  Jitter value of the marker-indicated data (UI+p)
  Form1 or Form2 (depending on the display scale)
  <uimp> = <STRING RESPONSE DATA>
  Jitter value of the marker-indicated data (UI-p)
  Form1 or Form2 (depending on the display scale)
  <uirms> = <STRING RESPONSE DATA>
  Jitter value of the marker-indicated data (UIrms)
  Form1 or Form2 (depending on the display scale)
  * When the Peak jitter analyze data does not exist or when the marker is OFF, the following is output:
  < 0,0,0,0,0,0,"--------","--------","--------","--------","--------"

- Function Queries the data indicated with the marker on the Analyze: Peak jitter screen.

- Example use > :DISPlay:ANALysis:PEAK:DATA?
  < 2000,10,12,20,10,30," 0"," 0.8"," 0.3"," 0.3"," 0.7"

:CALCulate:PEAK:DATA?

Query command

∇ Parameter
<time> = <year>,<month>,<day>,<hour>,<minute>
<year> = <DECIMAL NUMERIC PROGRAM DATA>
2000 to 2093 (year)
<month> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 12 (month)
<day> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 31 (day)
<hour> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 23 (hour)
<minute> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59 (minute)
[<number>] = <DECIMAL NUMEIC PROGRAM DATA>
The number of read data
0 to 3600

Note: When the number of data is not specified, all of the retained data is read.

- Response
<time> = <year>,<month>,<day>,<hour>,<minute>,<second>
Year, month, day, hour, minute, second of the read data

* If no data exists on the specified time when obtaining data, the data on the time next to the specified time is output.
Setting the time before the measurement start time outputs the data at the measurement start time.
Setting the time after the log end time outputs the data at the log end time.

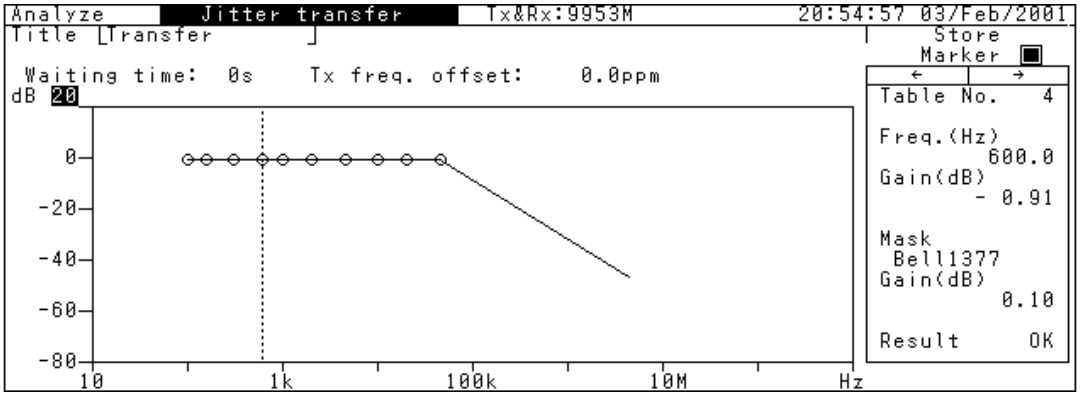
<year> = <NR1 NUMERIC RESPONSE DATA>
0,2000 to 2093 (year)
<month> = <NR1 NUMERIC RESPONSE DATA>
0,1 to 12 (month)
<day> = <NR1 NUMERIC RESPONSE DATA>
0,1 to 31 (day)
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 (hour)
<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 (minute)
<unlock> = <STRING RESPONSE DATA>
Occurrence time of unlock at the specified time (sec)
Form1
<ui> = <STRING RESPONSE DATA>
Jitter value (UIp-p) at the specified time
Form1 or 2
<uipp> = <STRING RESPONSE DATA>
Jitter value at the specified time (UI+p)
Form1 or 2
<uimp> = <STRING RESPONSE DATA>
Jitter value at the specified time (UI-p)
Form1 or 2
<uirms> = <STRING RESPONSE DATA>
Jitter value at the specified time (UIrms)
Form1 or 2

* When no data exists (when the screen displays "----"), the following is output:
< 0,0,0,0,0,0,"--------","--------","--------","--------","--------"

- Function
Queries the data of the Peak jitter graph.
(This query is also allowed when Marker is ON.)

- Example use
Read two data pieces from 10:30, June 10, 2000.
> :CALCulate:PEAK:DATA? 2000,6,10,10,30,0,2
or
> :CALCulate:PEAK:DATA? 2000,6,10,10,30,2
< 2000,10,12,20,10,30," 0"," 0.8"," 0.3"," 0.3"," 0.7",
" 1"," 0.8"," 0.3"," 0.3"," 0.7"

11.4.4 Jitter transfer sub-screen



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph (Store key) Name	:SYSTem:MEMory:ANALysis:STORe	11-142
Title	:DISPlay:ANALysis:JTRansfer:TITLe	11-161
Waitingtime	:SENSe:JITTer:Transfer:WTIMe?	11-112
Tx Freq. offset	:SOURce:JITTer:Transfer:OFFSet?	11-111
Vertical scale (Scale)	:DISPlay:ANALysis:JTRansfer:SCALe	11-162
Marker key	:DISPlay:ANALysis:JTRansfer:MDISplay	11-163
In a marker field <,>	:DISPlay:ANALysis:JTRansfer:MARKer	11-164
In a marker field <,> TableNo. In a marker field <,> Freq (numeric) In a marker field <,> Freq (unit) In a marker field <,> Gain (dB) In a marker field <,> Mask (Type) In a marker field <,> Mask Gain In a marker field <,> Mask Result	:DISPlay:ANALysis:JTRansfer:DATA?	11-165

See the section on the common commands to Analyze screens for the following commands:

- :DISPlay:ANALysis[:NAME]
- :SYSTem:MEMory:ANALysis:STORe

* The displays of the followings reflect the settings on the Test menu: Jitter transfer screen.
For details of associated commands, see the explanation for each screen.
Associated commands
:SENSe:JITTer:Transfer:WTIMe?
:SOURce:JITTer:Transfer:OFFSet?

** Test menu: Jitter transfer screen **

- Waitingtime
- Freq. offset

:DISPlay:ANALysis:JTRansfer:TITLe

Setting command

∇ Parameter <title> = <STRING PROGRAM DATA>
"Title string" Title string (max. 15 characters)
The string length is 0 to 15 characters.
Entry character: 0 to 9,A to Z,a to z, $&#%'.()-@_^{}^!

Spaces are added when the string is less than 15 characters.

- Function Sets a title on the Analyze: Jitter transfer screen.

- Example use Display "TITLE-DISP" as a title of a jitter transfer characteristic measurement.
> :DISPlay:ANALysis:JTRansfer:TITLe"TITLE-DISP"

:DISPlay:ANALysis:JTRansfer:TITLe?

Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:JTRansfer:TITLe.

- Function Queries a title on the Analyze: Jitter transfer screen.

- Example use > :DISPlay:ANALysis:JTRansfer:TITLe?
< "TITLE-DISP "

:DISPlay:ANALysis:JTRansfer:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

20	20 dB
10	10 dB
1	1 dB

- Function Sets the y axis scale of a graph on the Analyze: Jitter transfer screen.

- Example use Set 10 for the y axis scale of a graph for a jitter transfer characteristic measurement.
  > :DISPlay:ANALysis:JTRansfer:SCALe 10

:DISPlay:ANALysis:JTRansfer:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:JTRansfer:SCALe.

- Function Queries the value of the y axis scale of a graph on the Analyze: Jitter transfer screen.

- Example use > :DISPlay:ANALysis:JTRansfer:SCALe?
  < 10

:DISPlay:ANALysis:JTRansfer:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Jitter transfer screen.

- Example use Set the marker display of a jitter transfer measurement to ON
  > :DISPlay:ANALysis:JTRansfer:MDISplay ON

:DISPlay:ANALysis:JTRansfer:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Marker is set to OFF
1	Marker is set to ON

- Function Queries the display status of the marker on the Analyze: Jitter transfer screen.

- Example use > :DISPlay:ANALysis:JTRansfer:MDISplay?
  < 1

:DISPlay:ANALysis:JTRansfer:MARKer

Setting command

▽ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Jitter transfer screen.

- Example use Move in the right direction.
  > :DISPlay:ANALysis:JTRansfer:MARKer RIGHt

:DISPlay:ANALysis:JTRansfer:DATA? Query command

▽ Parameter None

- Response <point>,<freq1>,<freq2>,<amp1>,<mask>,<mamp1>,<result>

  <point> = <NR1 NUMERIC RESPONSE DATA>
  1 to 20 Measurement point
  <freq1> = <NR2 NUMERIC RESPONSE DATA>
  1.0 to 990.0 Modulated frequency (numeric)
  <freq2> = <CHARACTER RESPONSE DATA>
  HZ,KHZ,MHZ Modulated frequency (unit)
  <amp1>= <STRING RESPONSE DATA>
  Form6 Result of a jitter transfer characteristic measurement (dB)
  <mask> = <CHARACTER PROGRAM DATA>
  G958A G958 Type A
  ANSIT1A ANSI T1.105.03 Type A
  ANSIT1B ANSI T1.105.03 Type B
  B1377 Bell 1377
  B253 Bell 253
  USER User
  <mamp1> = <STRING RESPONSE DATA>
  Form6 Jitter transfer characteristic mask data (dB)
  <result> = <CHARACTER PROGRAM DATA>
  OK
  NG

  * When the jitter transfer data does not exist or when the marker is OFF, the following is output:
  < 0,0.0,HZ,"-------",0,0,0,"--"

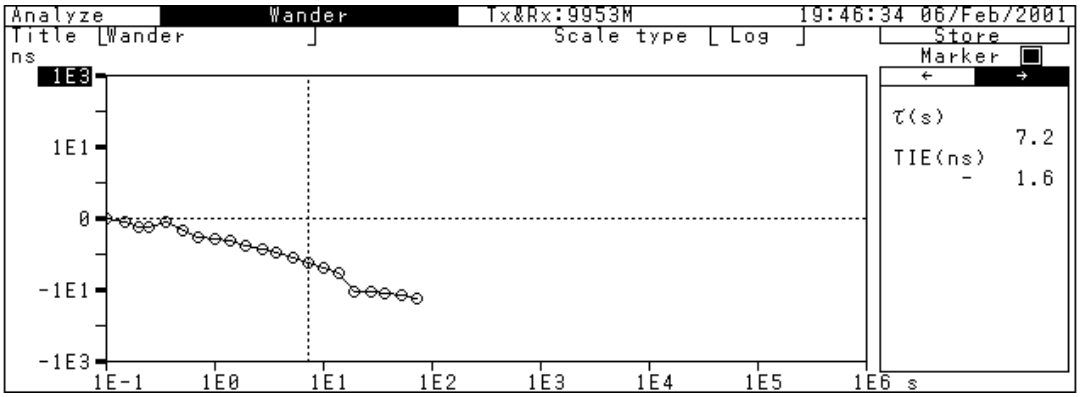
- Function Queries the maker-specified data on the Analyze: Jitter transfer screen.

- Example use > :DISPlay:ANALysis:JTRansfer:DATA?
  < 15,100,HZ," -15.00",G958A,"-12.00",OK

11.4.5 Wander sub-screen

Scale: Log

- Marker ON



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph (Store key) Name	:SYSTem:MEMory:ANALysis:STORe	11-142
Title	:DISPlay:ANALysis:WANDer:TITLe	11-168
Scale type	:DISPlay:ANALysis:WANDer:STYPe	11-169
Vertical scale (Scale)	:DISPlay:ANALysis:WANDer:LOG:SCALe	11-170
Marker key	:DISPlay:ANALysis:WANDer:MDISplay	11-174
In Marker field <,>	:DISPlay:ANALysis:WANDer:MARKer	11-175
In Marker field <,> Time (s) In Marker field <,> TIE (ns)	:DISPlay:ANALysis:WANDer:DATA?	11-175

See the section on the common commands to Analyze screens for the following commands:

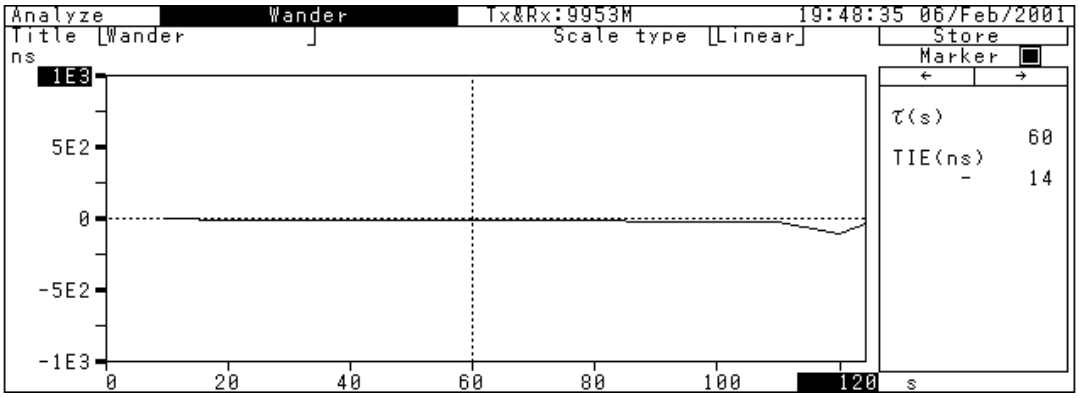
- :DISPlay:ANALysis[:NAME]
- :SYSTem:MEMory:ANALysis:STORe

The following is the specific command for Scale: Log.
Others are common with Scale: Linear.

- :DISPlay:ANALysis:WANDer:LOG:SCALe

Scale: Linear

- Marker ON



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph (Store key) Name	SYSTem:MEMory:ANALysis:STORe	11-142
Title	:DISPlay:ANALysis:WANDer:TITLe	11-168
Scaletype	:DISPlay:ANALysis:WANDer:STYPe	11-169
Vertical scale (Scale)	:DISPlay:ANALysis:WANDer:LINear:SCALe	11-171
Horizontal scale (Scale)	:DISPlay:ANALysis:WANDer:MEAStime	11-172
Horizontal scale	:DISPlay:ANALysis:WANDer:USER	11-173
Marker key	:DISPlay:ANALysis:WANDer:MDISplay	11-174
In Marker field <,>	:DISPlay:ANALysis:WANDer:MARKer	11-175
In Marker field <,> Time (s) In Marker field <,> TIE (ns)	:DISPlay:ANALysis:WANDer:DATA?	11-175

See the section on the common commands to Analyze screens for the following commands:

- :DISPlay:ANALysis[:NAME]
- :SYSTem:MEMory:ANALysis:STORe

The followings are the specific command for Scale: Linear.
Others are common with Scale: Log.

- :DISPlay:ANALysis:WANDer:LINear:SCALe
- :DISPlay:ANALysis:WANDer:MEAStime
- :DISPlay:ANALysis:WANDer:USER

:DISPlay:ANALysis:WANDer:TITLe

Setting command

∇ Parameter <title> = <STRING PROGRAM DATA>
"Title string" Title string (max. 15 characters)
The string length is 0 to 15 characters.
Entry character: 0 to 9,A to Z,a to z, $&#%'.()-@_^{}^!

When the string length is less than 15 characters, spaces are added.

- Function Sets a title on the Analyze: Wander screen.

- Example use Display "TITLE-DISP" as a title of a wander measurement.
> :DISPlay:ANALysis:WANDer:TITLe"TITLE-DISP"

:DISPlay:ANALysis:WANDer:TITLe?

Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:WANDer:TITLe.

- Function Queries a title on the Analyze: Wander screen.

- Example use > :DISPlay:ANALysis:WANDer:TITLe?
< "TITLE-DISP "

:DISPlay:ANALysis:WANDer:STYPe

Setting command

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

LOG	Log display
LINear	Linear display

- Function Sets the graph scale type of the y axis on the Analyze: Wander screen.

- Example use Set the scale type of the y axis on the Analyze: Wander screen to Log.
  > :DISPlay:ANALysis:WANDer:STYPe LOG

:DISPlay:ANALysis:WANDer:STYPe?

Query command

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>

LOG	Log display
LIN	Linear display

- Function Queries the scale type of the y axis on the Analyze: Wander screen

- Example use > :DISPlay:ANALysis:WANDer:STYPe?
  < LOG

:DISPlay:ANALysis:WANDer:LOG:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>
1E12 (unit: ns)
1E9
1E6
1E3

- Function Sets the maximum value of the y axis scale of the wander data (Log type).

- Example use Set the maximum value of the y axis scale of a graph to 106 ns.
> :DISPlay:ANALysis:WANDer:LOG:SCALe 1E6

:DISPlay:ANALysis:WANDer:LOG:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:WANDer:LOG:SCALe.

- Function Queries the maximum value of the y axis scale of the wander data (Log type) graph.

- Example use Query the maximum value of the y axis scale of the Wander data (Log type) graph.
> :DISPlay:ANALysis:WANDer:LOG:SCALe?
< 1E6

:DISPlay:ANALysis:WANDer:LINear:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

1E12 (unit: dB)
1E11
1E10
1E9
1E8
1E7
1E6
1E5
1E4
1E3
1E2
1E1
1E0

- Function Sets the maximum value of the y axis scale of the wander data (Linear type).

- Example use Set the maximum value of the y axis scale of a graph to 10^6.
  > :DISPlay:ANALysis:WANDer:LINear:SCALe 1E6

:DISPlay:ANALysis:WANDer:LINear:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:WANDer:LINear:SCALe.

- Function Queries the maximum value of the y axis scale of the wander data (Linear type) graph.

- Example use Query the maximum value of the y axis scale of the wander data (Linear type) graph.
  > :DISPlay:ANALysis:WANDer:Linear:SCALe?
  < 1E6

:DISPlay:ANALysis:WANDer:MEAStime

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

SEC120000	120000 sec
SEC12000	12000 sec
SEC1200	1200 sec
SEC120	120 sec
SEC12	12 sec
USER	User

- Function Sets the maximum value of the x axis of a wander data (Linear type) graph.

- Example use Set the maximum value of the x axis on a wander data (Linear type) graph to 1,200.
  > :DISPlay:ANALysis:WANDer:MEAStime 1200

:DISPlay:ANALysis:WANDer:MEAStime?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:WANDer:MEAStime.

- Function Queries the maximum value of the x axis on a wander data (Linear type) graph.

- Example use Query the maximum value of the x axis of a graph.
  > :DISPlay:ANALysis:WANDer:MEAStime?
  < 1200

:DISPlay:ANALysis:WANDer:USER

Setting command

∇ Parameter <scale> = <DECIMAL NUMERIC PROGRAM DATA>
From 10 to the value of observation time set in the Test menu: Wander screen.

* When a fraction is entered, it is rounded down:
  1280 becomes 1200
  and 12006 becomes 12000.
* When the observation time is 120,000 seconds or more, the upper limit is 120,000 seconds.

- Function Sets the maximum value of the x axis of a wander data (Linear type) graph.
  (Only when :DISPlay:ANALysis:WANDer:MEAStime is set to USER.)

- Example use Set the maximum value of the x axis on a wander data (Linear type) graph to 12,000.
  > :DISPlay:ANALysis:WANDer:USER 12000

  (Reference)
  Even if you specify >:DISPlay:ANALysis:WANDer:USER 12008, the value is set to 12,000.

:DISPlay:ANALysis:WANDer:USER?

Query command

∇ Parameter None

- Response <scale> = <NR1 NUMERIC RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:WANDer:USER.

- Function Queries the maximum value of the x axis on a wander data (Linear type) graph.

- Example use Query the maximum value of the x axis of a graph.
  > :DISPlay:ANALysis:WANDer:USER?
  < 12000

:DISPlay:ANALysis:WANDer:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Wander screen.

- Example use Set the marker display of a wander measurement to ON.
  > :DISPlay:ANALysis:WANDer:MDISplay ON

:DISPlay:ANALysis:WANDer:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Marker is set to OFF
1	Marker is set to ON

- Function Queries the marker display status on the Analyze: Wander screen.
- Example use > :DISPlay:ANALysis:WANDer:MDISplay?
  < 1

:DISPlay:ANALysis:WANDer:MARKer

Setting command

∇ Parameter <marker> = <CHARACTER PROGRAM DATA>
LEFT Moves left by 1 Div.
RIGHt Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Wander screen.

- Example use Move in the right direction.
> :DISPlay:ANALysis:WANDer:MARKer RIGHt

:DISPlay:ANALysis:WANDer:DATA?

Query command

∇ Parameter None

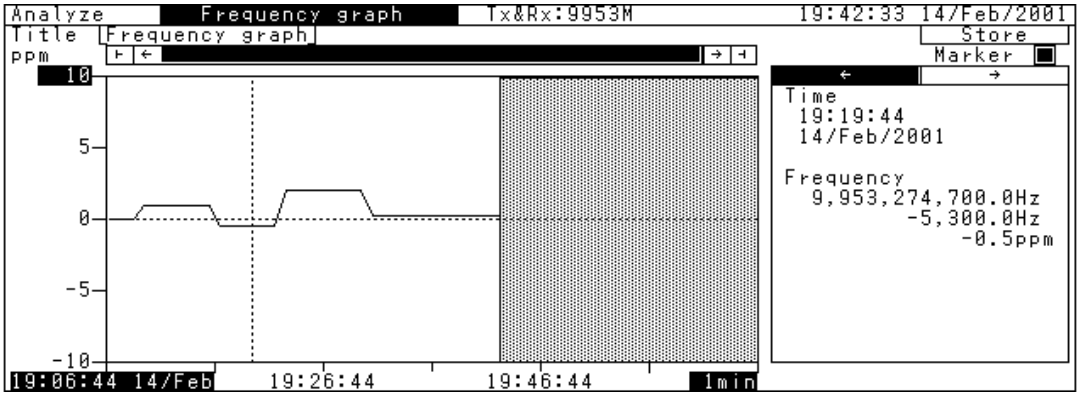
- Response <time1>,<time2>
<time1> = <STRING RESPONSE DATA>
Form8' τ (s)
<time2> = <STRING RESPONSE DATA>
Form8 (ns)

* When the wander data does not exist or the marker is OFF, the following is output:
< 0,0.0

- Function Queries the data contents indicated with the marker on the Analyze: Wander screen.

- Example use > :DISPlay:ANALysis:WANDer:DATA?
< " 0.35"," 3.0"

11.4.6 Frequency graph sub-screen



Setting Item	Command	Reference Page
(Sub-screen)	:DISPlay:ANALysis[:NAME]	11-141
(Store key) Graph (Store key) Name	:SYSTem:MEMory:ANALysis:STORe	11-142
Title	:DISPlay:ANALysis:FGRaph:TITLe	11-177
Graph scroll ←, →, -|, |-	:DISPlay:ANALysis:FGRaph:SCRoll	11-178
Vertical scale (Scale)	:DISPlay:ANALysis:FGRaph:SCALe	11-179
Marker key	:DISPlay:ANALysis:FGRaph:MDISplay	11-180
(Display start time) year (Display start time) month (Display start time) day (Display start time) hour (Display start time) minute (Display start time) second	:DISPlay:ANALysis:FGRaph:FROM	11-181
Horizontal scale (time) Horizontal scale (unit)	:DISPlay:ANALysis:FGRaph:INTerval	11-182
In Marker field <,>	:DISPlay:ANALysis:FGRaph:MARKer	11-183
Response item: (time) Response item: (Relative freq.) Response item: (ppm) Response item: (Freq.)	:DISPlay:ANALysis:FGRaph:DATA?	11-184
Response item: Time Response item: (Number of data)	:CALCulate:FGRaph:DATA?	11-185

See the section on the common commands to Analyze screens for the following commands:

- :DISPlay:ANALysis[:NAME]
- :SYSTem:MEMory:ANALysis:STORe

:DISPlay:ANALysis:FGRaph:TITLe

Setting command

∇ Parameter

<title> = <STRING PROGRAM DATA>
"Title string" Title string (max. 15 characters)
The string length is 0 to 15 characters.
Entry character: 0 to 9,A to Z,a to z, $&#%'.()-@_^{}^!

When the string length is less than 15 characters, spaces are added.

- Function Sets a title on the Analyze: Frequency graph screen.

- Example use Display "TITLE-DISP" as a title of Frequency.
> :DISPlay:ANALysis:FGRaph:TITLe"TITLE-DISP"

:DISPlay:ANALysis:FGRaph:TITLe?

Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:FGRaph:TITLe.

- Function Queries a title on the Analyze: Frequency graph screen.

- Example use > :DISPlay:ANALysis:FGRaph:TITLe?
< "TITLE-DISP "

:DISPlay:ANALysis:FGRaph:SCRoll

Setting command

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

LEFT	Scrolls left.
RIGHt	Scrolls right.
TOP	Moves to the top.
BOTTom	Moves to the bottom.

- Function Specifies the graph scroll on the Analyze: Frequency graph screen.

- Example use Scroll in the right direction.
  > :DISPlay:ANALysis:FGRaph:SCRoll RIGHt

:DISPlay:ANALysis:FGRaph:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

10	10 ppm
100	100 ppm
1000	1000 ppm

- Function Sets the y axis scale of a graph on the Analyze: Frequency graph screen.

- Example use Set the y axis scale of the Frequency graph to 10 ppm.
  > :DISPlay:ANALysis:FGRaph:SCALe 10

:DISPlay:ANALysis:FGRaph:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:FGRaph:SCALe.

- Function Queries the value of the y axis scale of a graph on the Analyze: Frequency graph screen.

- Example use > :DISPlay:ANALysis:FGRaph:SCALe?
  < 10

:DISPlay:ANALysis:FGRaph:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Frequency graph screen.

- Example use Set the marker display on Frequency to ON.
  > :DISPlay:ANALysis:FGRaph:MDISplay 1

:DISPlay:ANALysis:FGRaph:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Marker is set to OFF
1	Marker is set to ON

- Function Queries ON/OFF of the marker display on the Analyze: Frequency graph screen.

- Example use > :DISPlay:ANALysis:FGRaph:MDISplay?
  < 1

:DISPlay:ANALysis:FGRaph:FROM

Setting command

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>

<numeric1> =	2000 to 2093 (year)
<numeric2> =	1 to 12 (month)
<numeric3> =	1 to 31 (day)
<numeric4> =	0 to 23 (hour)
<numeric5> =	0 to 59 (minute)
<numeric6> =	0 to 59 (second)

* When the measured data exists:
  If the time specified with a parameter does not exist, the time next to the specified time is set.
  Specifying the time before the measurement start time sets the start time.
  Specifying the time after the log end time sets the log end time.
  When <numeric6> is not specified, 0 is set.
* When the measured data does not exist (when the screen displays "--:--:-- --/--"), setting the time causes an execution error "Setting conflict".

- Function Sets the display start position on the Frequency graph.

- Example use Start display from 11:30:40, July 28, 2000.
  > :DISPlay:ANALysis:FGRaph:FROM 2000,7,28,11,30,40

:DISPlay:ANALysis:FGRaph:FROM?

Query command

∇ Parameter None

- Response <numeric1>,<numeric2>,<numeric3>,<numeric4>,<numeric5>,<numeric6>
  Same as in the case of :DISPlay:ANALysis:FGRaph:FROM.

- Function Queries the display start position on the Frequency graph.

- Example use > :DISPlay:ANALysis:FGRaph:FROM?
  < 2000,7,28,11,30,40

:DISPlay:ANALysis:FGRaph:INTerval

Setting command

∇ Parameter

<numeric> = <CHARACTER PROGRAM DATA>
1, 15, 60 Interval Time

<suffix> = <CHARACTER PROGRAM DATA>
M Interval Time suffix minute only

- Function: Sets the width for one scale of the time axis on the Analyze: Frequency graph screen.
- Example use: Set the width for one scale to 1 minute.
  > :DISPlay:ANALysis:FGRaph:INTerval 1,M

:DISPlay:ANALysis:FGRaph:INTerval?

Query command

∇ Parameter: None

- Response:
  <numeric> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:FGRaph:INTerval.

  <suffix> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:FGRaph:INTerval.
- Function: Queries the width for one scale of the time axis on the Analyze: Frequency graph screen.
- Example use:
  > :DISPlay:ANALysis:FGRaph:INTerval?
  < 1,M

:DISPlay:ANALysis:FGRaph:MARKer

Setting command

∇ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Frequency graph screen.

- Example use Move in the right direction.
  > :DISPlay:ANALysis:FGRaph:MARKer RIGHt

:DISPlay:ANALysis:FGRaph:DATA? Query command

▽ Parameter None

- Response <time>,<frequency>,<ppm>
  - <time> = <year>,<month>,<day>,<hour>,<minute>,<second>
    Year, month, day, hour, minute, second of data
  - <year> = <NR1 NUMERIC RESPONSE DATA>
    0,2000 to 2093 year
  - <month> = <NR1 NUMERIC RESPONSE DATA>
    0,1 to 12 month
  - <day> = <NR1 NUMERIC RESPONSE DATA>
    0,1 to 31 day
  - <hour> = <NR1 NUMERIC RESPONSE DATA>
    0 to 23 hour
  - <minute> = <NR1 NUMERIC RESPONSE DATA>
    0 to 59 minute
  - <second> = <NR1 NUMERIC RESPONSE DATA>
    0 to 59 second
  - <frequency> = <STRING RESPONSE DATA>
    Monitor frequency (Hz) (relative)
    (Form10)
  - <ppm> = <STRING RESPONSE DATA>
    Monitor frequency deviation (ppm)
    (Form11)
  - <frequency> = <STRING RESPONSE DATA>
    Monitor frequency (Hz)
    (Form10')

  * When no data exists, the following is output:
    < 0,0,0,0,0,0,"----------","--------"

- Function Queries the data indicated with the marker on the Analyze: Frequency graph screen.

- Example use > :DISPlay:ANALysis:FGRaph:DATA?
  < 2000,10,23,1,20,30," 100.0"," +100"," 1100.0"

:CALCulate:FGRaph:DATA? Query command

∇ Parameter <time> = <year>,<month>,<day>,<hour>,<minute>,
<year> = <DECIMAL NUMERIC PROGRAM DATA>
2000 to 2093 (year)
<month> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 12 (month)
<day> = <DECIMAL NUMERIC PROGRAM DATA>
1 to 31 (day)
<hour> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 23 (hour)
<minute> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 59 (minute)

[<number>] = <DECIMAL NUMEIC PROGRAM DATA>
The number of read data
0 to 3600
Note: When the number of data pieces is not specified, all of the retained data is output.

- Response <time> = <year>,<month>,<day>,<hour>,<minute>,<second>
Year, month, day, hour, minute, second of the read data
  * If no data exists at the parameter-specified time when obtaining data, the data at the time next to the specified time is output.
  Setting the time before the measurement start time outputs the data at the measurement start time.
  Setting the time after the log end time outputs the data at the log end time.

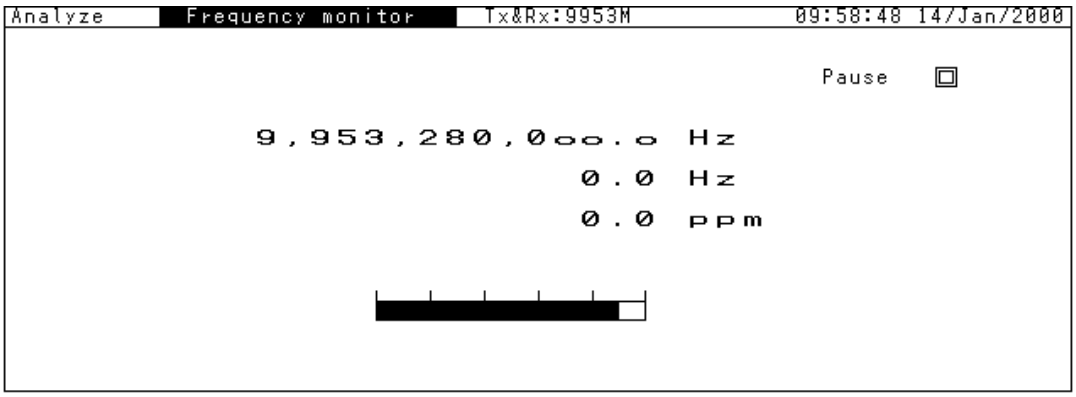
<year> = <NR1 NUMERIC RESPONSE DATA>
0,2000 to 2093 (year)
<month> = <NR1 NUMERIC RESPONSE DATA>
0,1 to 12 (month)
<day> = <NR1 NUMERIC RESPONSE DATA>
0,1 to 31 (day)
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 (hour)
<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 (minute)
<freq> = <STRING RESPONSE DATA>
Frequency value (Hz) (relative)
(From10)
<ppm> = <STRING RESPONSE DATA>
ppm value
(From11)

  * When no data exists (when the screen displays "----"), the following is output:
  < 0,0,0,0,0,0,"----------","--------"

- Function Queries the graph data of the frequency monitor.

- Example use Read two data pieces from 10:30, June 10, 2000.
> :CALCulate:FGRaph:DATA? 2000,6,10,10,30,0,2
or
> :CALCulate:FGRaph:DATA? 2000,6,10,10,30,2
< 2000, 6,10,10,30, 0," 300.9"," -10.0"," 300.9"," -10.0"

11.4.7 Frequency monitor sub-screen



Setting Item	Command	Reference Page
Pause key	:DISPlay:ANALysis:FMONitor:PAUSe	11-187
Response item: (relative Freq.) Response item: (ppm) Response item: (Freq.)	:DISPlay:ANALysis:FMONitor:FREQuency?	11-188

:DISPlay:ANALysis:FMONitor:PAUSe

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets Pause to OFF
ON or 1	Sets Pause to ON

- Function Sets Pause ON/OFF on the Analyze: Frequency monitor screen.

- Example use Set ON on the Frequency monitor to ON.
  > :DISPlay:ANALysis:FMONitor:PAUSe ON

:DISPlay:ANALysis:FMONitor:PAUSe?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Pause is set to OFF
1	Pause is set to ON

- Function Queries the Pause status on the Analyze: Frequency monitor screen.

- Example use > :DISPlay:ANALysis:FMONitor:PAUSe?
  < 1

:DISPlay:ANALysis:FMONitor:FREQuency? Query command

▽ Parameter None

- Response <relative>,<ppm>,<freq>
  <relative> = <STRING RESPONSE DATA>
  Form10 Relative frequency
  <ppm> = <STRING RESPONSE DATA>
  Form11 Monitor frequency deviation
  <freq> = <STRING RESPONSE DATA>
  Form10' Frequency

  * When no data exists, the following is output:
  < "----------","--------","----------"

- Function Queries the data on the Frequency monitor.

- Example use > :DISPlay:ANALysis:FMONitor:FREQuency?
  < " 1000.0"," +100.0"," 1100.0"

11.4.8 Recall sub-screen

Note:
- The Recall screen when each data is recalled is displayed in the same way as the individual screen.
- A title cannot be set on the Recall screen.
- Store function cannot be used.
- When a command not associated with the displayed Recall screen is sent, a command error "Setting conflict" is returned.

Commands that are associated with the Recall screen are described from the next page.

Recall screen common command list

Setting Item	Command	Parameter
(Recalltype)	:DISPlay:ANALysis:RECall:TYPE?	None

:DISPlay:ANALysis:RECall:TYPE?

Query command

∇ Parameter None

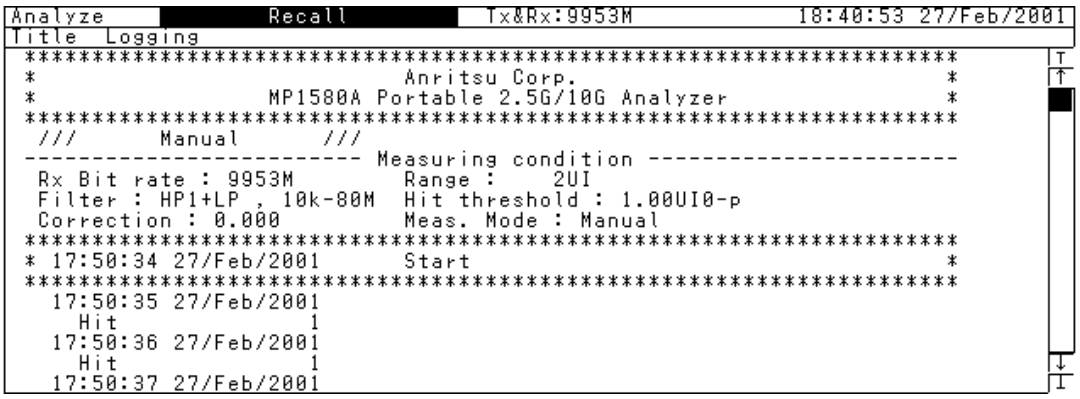
- Response <type> = <STRING RESPONSE DATA>
  "LOGG" Logging data
  "PEAK" Peak jitter data
  "JTOL" Jitter tolerance data
  "JSW" Jitter sweep data
  "FSW" Freq. sweep data
  "JTR" Jitter transfer data
  "WAND" Wander data
  "WSW" Wander sweep data
  "FGR" Frequency graph data

  * When the Recall data does not exist, the following is output:
  < "No data"

- Function Queries the data type displayed on the Analyze: Recall screen.

- Example use > :DISPlay:ANALysis:RECall:TYPE?
  < "LOGG"

a) Logging



Setting Item	Command	Reference Page
Title	:DISPlay:ANALysis:RECall:LOGGing:TITLe?	11-192
(Read the Log data contents)	:DISPlay:ANALysis:RECall:LOGGing:DATA?	11-193
(Read all the Log data contents)	:DISPlay:ANALysis:RECall:LOGGing:ADATa?	11-194
Graph scroll ←, →, T, ⊥	:DISPlay:ANALysis:RECall:LOGGing:SCRoll	11-195

:DISPlay:ANALysis:RECall:LOGGing:TITLe? Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:LOGGing:TITLe.

- Function Queries a title when the Logging data is obtained on the Analyze: Recall screen.

- Example use > :DISPlay:ANALysis:RECall:LOGGing:TITLe?
  < "TITLE-LOG "

:DISPlay:ANALysis:RECall:LOGGing:DATA? Query command

∇ Parameter <start> = <DECIMAL NUMERIC PROGRAM DATA>
1 to the last line (step: 1): beginning line of data read
<end> = <DECIMAL NUMERIC PROGRAM DATA>
1 to the last line (step: 1): beginning line of data read
-1: obtains the log data up to the ending line.

- Response <logdata> = <STRING RESPONSE DATA>
"(log data contents of one line)"
When no data exists, "No data" is output.

- Function Queries the data contents when the Logging data is obtained on the Analyze: Recall screen.

- Restriction When an <end> value is smaller than a <start> value, a numeric of <end> is assumed to be the same as the <start> value.
When an <end> value is larger than data that can be obtained, END is returned following the last line of the log data.

- Example use Specifies data from 1st line to 15th line, and read out them.
(Since the last data is 12th line, "END" is appended at the last.)

```
> :DISPlay:ANALysis:RECall:LOGGing:DATA? 1,15
<"*****************************************************",
"*                  Anritsu Corp.                     *",
"*          MP1580A Portable 2.5G/10G Analyzer        *",
"*****************************************************",
" ///                    Manual                    ///",
"----------------- Measuring condition ----------------",
" Rx Bit rate : 9953M       Range :   2UI              ",
" Filter : HP1+LP , 10k-80M  Hit threshold : 1.00UI0-p  ",
" Correction : 0.000        Meas. Mode : Repeat  1s    ",
"*****************************************************",
"* 06:57:37 10/Feb/2000     Start                    *",
"*****************************************************",
"END"
```

:DISPlay:ANALysis:RECall:LOGGing:ADATa? Query command

∇ Parameter <all> = <DECIMAL NUMERIC PROGRAM DATA>
-1: obtains the log data up to the last line.

- Response <logdata> = <STRING RESPONSE DATA>
"(log data contents of one line)"
When no data exists, "No data" is output.
END is returned following the last line of the log data.

- Function Queries all the log data contents when the Logging data is obtained on the Analyze: Recall screen.

- Example use Queries data contents.

```
> :DISPlay:ANALysis:RECall:LOGGing:ADATa? -1
        <"*****************************************************",
        "*                    Anritsu Corp.                    *",
        "*          MP1580A Portable 2.5G/10G Analyzer         *",
        "*****************************************************",
        " ///                   Manual                      ///",
        "----------------- Measuring condition ----------------",
        " Rx Bit rate : 9953M       Range :   2UI              ",
        " Filter : HP1+LP , 10k-80M  Hit threshold : 1.00UI0-p ",
        " Correction : 0.000         Meas. Mode : Repeat  1s   ",
        "*****************************************************",
        "* 06:57:37 10/Feb/2000   Start                      *",
        "*****************************************************",
        "END"
```

:DISPlay:ANALysis:RECall:LOGGing:SCRoll

Setting command

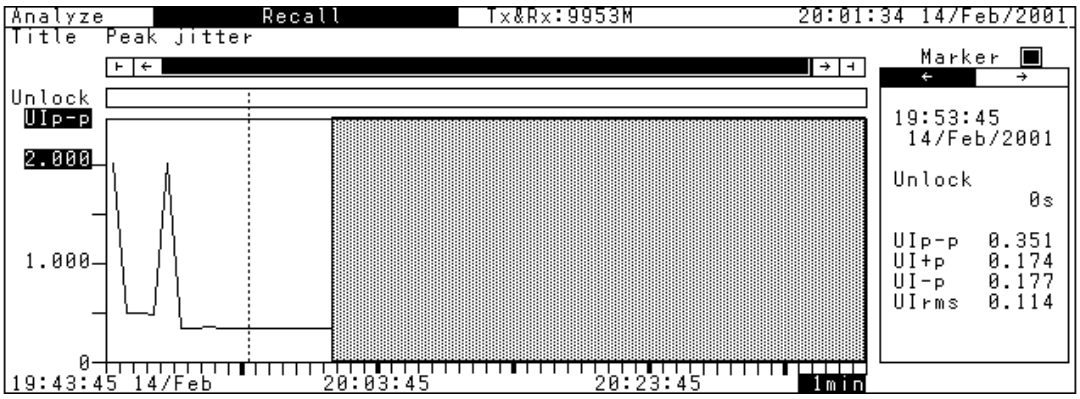
▽ Parameter <type> = <CHARACTER PROGRAM DATA>

TOP	Jumps to the data position of Logging start.
END	Jumps to the data position of Logging end.
NEXT	Scrolls down one screen.
BEFOR	Scrolls up one screen.

- Function Specifies the graph scroll when the Logging data is received on the Analyze: Recall screen.

- Example use Scroll up one screen.
  > :DISPlay:ANALysis:RECall:LOGGing:SCRoll BEFOR

b) Peak jitter



Setting Item	Command	Reference Page
Title	:DISPlay:ANALysis:RECall:PEAK:TITLe?	11-197
Graph scroll ←, →, -|, |-	:DISPlay:ANALysis:RECall:PEAK:SCRoll	11-198
Result display type	:DISPlay:ANALysis:RECall:PEAK:DTYPe	11-199
Scale	:DISPlay:ANALysis:RECall:PEAK:SCALe	11-200
Display start position (time)	:DISPlay:ANALysis:RECall:PEAK:FROM	11-201
Interval (time) Interval (unit)	:DISPlay:ANALysis:RECall:PEAK:INTerval	11-202
Marker key	:DISPlay:ANALysis:RECall:PEAK:MDISplay	11-203
In Marker field <,>	:DISPlay:ANALysis:RECall:PEAK:MARKer	11-204
(Data in a marker field) time (Data in a marker field) Unlock (Data in a marker field) UIp-p (Data in a marker field) UI+p (Data in a marker field) UI-p (Data in a marker field) UIrms	:DISPlay:ANALysis:RECall:PEAK:DATA?	11-205

:DISPlay:ANALysis:RECall:PEAK:TITLe?

Query command

∇ Parameter None

- Response <title> = <STRING RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:PEAK:TITLe.

- Function Queries a title on the Analyze: Recall screen (Peak jitter).

- Example use > :DISPlay:ANALysis:RECall:PEAK:TITLe?
  < "TITLE-DISP "

:DISPlay:ANALysis:RECall:PEAK:SCRoll

Setting command

▽ Parameter <type> = <CHARACTER PROGRAM DATA>

LEFT	Scrolls left.
RIGHt	Scrolls right.
TOP	Moves to the top.
BOTTom	Moves to the bottom.

- Function Specifies the graph scroll on the Analyze: Recall screen (Peak jitter).

- Example use Scroll in the right direction
  > :DISPlay:ANALysis:RECall:RECall:PEAK:K:SCRoll RIGHt

:DISPlay:ANALysis:RECall:PEAK:DTYPE

Setting command

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

UIPTp	UIp-p
UIPP	UI+p
UIMP	UI-p
UIRMs	UIrms

- Function Sets the display type of a graph on the Analyze: Recall screen (Peak jitter).

- Example use Set the screen type to UIp-p.
  > :DISPlay:ANALysis:RECall:PEAK:DTYPe UIPTp

:DISPlay:ANALysis:RECall:PEAK:DTYPE?

Query command

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>

UIPT	UIp-p
UIPP	UI+p
UIMP	UI-p
UIRM	UIrms

- Function Queries the display type of a graph on the Analyze: Recall screen (Peak jitter).
- Example use > :DISPlay:ANALysis:RECall:PEAK:DTYPe?
  < UIRM

:DISPlay:ANALysis:RECall:PEAK:SCALe

Setting command

∇ Parameter <scale> = <DECIMAL NUMERIC PROGRAM DATA>
0.002 to 4000.0 See the table below for a step.

Setting range by stored Rx Range

Rx Range	UIp-p	UI+p	UI-p	UIrms
4000 UI	4 to 4000 / 4 (step)	4 to 2000 / 4 (step)	4 to 2000 / 4 (step)	Setting disabled
1000 UI	2 to 1000 / 2 (step)	2 to 500 / 2 (step)	2 to 500 / 2 (step)	Setting disabled
20 UI	0.02 to 20.00 / 0.02 (step)	0.02 to 10.00 / 0.02 (step)	0.02 to 10.00 / 0.02 (step)	0.02 to 7.00 / 0.02 (step)
2 UI	0.002 to 2.000 / 0.002 (step)	0.002 to 1.000 / 0.002 (step)	0.002 to 1.000 / 0.002 (step)	0.002 to 0.700 / 0.002 (step)

- Function Sets the y axis scale of a graph on the Analyze Recall screen (Peak jitter).

- Example use Set the scale to 0.02.
> :DISPlay:ANALysis:RECall:PEAK:SCALe 0.02

:DISPlay:ANALysis:RECall:PEAK:SCALe?

Query command

∇ Parameter None

- Response <scale> = <NR2 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:SCALe.

- Function Queries the value of the y axis scale of a graph on the Analyze: Recall screen (Peak jitter).

- Example use > :DISPlay:ANALysis:RECall:PEAK:SCALe?
< 100.0

:DISPlay:ANALysis:RECall:PEAK:FROM

Setting command

∇ Parameter <numeric1>,<numeric2>,<numeric3>,<numeric4>,<numeric5> [,<numeric6>]
<DECIMAL NUMERIC PROGRAM DATA>
<numeric1> = 2000 to 2093 (year)
<numeric2> = 1 to 12 (month)
<numeric3> = 1 to 31 (day)
<numeric4> = 0 to 23 (hour)
<numeric5> = 0 to 59 (minute)
<numeric6> = 0 to 59 (second)

* When the time specified with a parameter does not exist, the time next to the specified time is set.
Specifying the time before the measurement start time sets the measurement start time.
Specifying the time after the log end time sets the log end time.
When <numeric6> is not specified, 0 is set.

- Function Sets the display start position of a graph on the Analyze Recall screen (Peak jitter).

- Example use Start display from 11:30:40, July 28, 2000.
> :DISPlay:ANALysis:RECall:PEAK:FROM 2000,7,28,11,30,40

:DISPlay:ANALysis:RECall:PEAK:FROM?

Query command

∇ Parameter None

- Response <numeric1>,<numeric2>,<numeric3>,<numeric4>,<numeric5>,<numeric6>
= <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:FROM.
* When the Peak jitter analyze data does not exist, the following is output:
< -,-,-,-,-,-

- Function Queries the time of the display start position of a graph on the Analyze: Recall screen (Peak jitter).

- Example use > :DISPlay:ANALysis:RECall:PEAK:FROM?
< 2000,7,28,11,30,40

:DISPlay:ANALysis:RECall:PEAK:INTerval

Setting command

∇ Parameter <numeric> = <CHARACTER PROGRAM DATA>
1, 15, 60
<suffix> = <CHARACTER PROGRAM DATA>
M minute

- Function Sets the width of one time axis scale of a graph on the Analyze: Recall screen (Peak jitter).

- Example use Set the width for one scale to 1 minute.
> :DISPlay:ANALysis:RECall:PEAK:INTerval 1,M

:DISPlay:ANALysis:RECall:PEAK:INTerval?

Query command

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:INTerval.
<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:PEAK:INTerval.

- Function Queries the width of one time axis scale of a graph on the Analyze: Recall screen (Peak jitter).

- Example use > :DISPlay:ANALysis:RECall:PEAK:INTerval?
< 1,M

:DISPlay:ANALysis:RECall:PEAK:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Recall screen (Peak jitter)

- Example use Set the marker display of a jitter transfer characteristic measurement to ON.
  > :DISPlay:ANALysis:RECall:PEAK:MDISplay ON

:DISPlay:ANALysis:RECall:PEAK:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>

0	Marker is set to OFF
1	Marker is set to ON

- Function Queries the display status of the marker on the Analyze: Recall screen (Peak jitter).

- Example use > :DISPlay:ANALysis:RECall:PEAK:MDISplay?
  < 1

:DISPlay:ANALysis:RECall:PEAK:MARKer

Setting command

▽ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Recall screen (Peak jitter).

- Example use Move in the right direction.
  > :DISPlay:ANALysis:RECall:PEAK:MARKer RIGHt

:DISPlay:ANALysis:RECall:PEAK:DATA? Query command

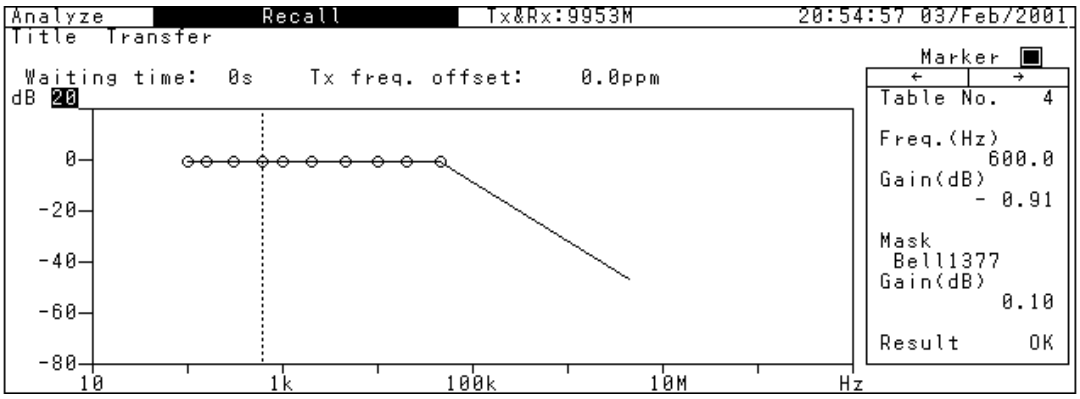
▽ Parameter None

- Response <time>,<unlock>,<ui>,<uipp>,<uimp>,<uirms>
  <time> = <year>,<month>,<day>,<hour>,<minute>,<second>
  Time indicated with the marker
  <year> = <NR1 NUMERIC RESPONSE DATA>
  0, 2000 to 2093 (year)
  <month> = <NR1 NUMERIC RESPONSE DATA>
  0, 1 to 12 (month)
  <day> = <NR1 NUMERIC RESPONSE DATA>
  0, 1 to 31 (day)
  <hour> = <NR1 NUMERIC RESPONSE DATA>
  0 to 23 (hour)
  <minute> = <NR1 NUMERIC RESPONSE DATA>
  0 to 59 (minute)
  <second> = <NR1 NUMERIC RESPONSE DATA>
  0 to 59 (second)
  <unlock> = <STRING RESPONSE DATA>
  Occurrence time when marker-indicated data is unlocked (sec).
  Form1
  <ui> = <STRING RESPONSE DATA>
  Jitter value of marker-indicated data (UI)
  Form1 or Form2 (depending on the display scale)
  <uipp> = <STRING RESPONSE DATA>
  Jitter value of marker-indicated data (UI)
  Form1 or Form2 (depending on the display scale)
  <uimp> = <STRING RESPONSE DATA>
  Jitter value of marker-indicated data (UI)
  Form1 or Form2 (depending on the display scale)
  <uirms> = <STRING RESPONSE DATA>
  Jitter value of marker-indicated data (UI)
  Form1 or Form2 (depending on the display scale)
  * When the Peak jitter analyze data does not exist or when the marker is OFF, the following is output:
  < 0,0,0,0,0,0,"--------","--------","--------","--------","--------"

- Function Queries the data indicated with the marker on the Analyze: Recall screen (Peak jitter).

- Example use > :DISPlay:ANALysis:RECall:PEAK:DATA?
  < 2000,10,12,20,10,30," 0"," 0.8"," 0.3"," 0.3"," 0.7"

c) Jitter transfer



Setting Item	Command	Reference Page
Title	:DISPlay:ANALysis:RECall:JTRansfer:TITLe?	11-207
Waiting time	:DISPlay:ANALysis:RECall:JTRansfer:WTIMe?	11-208
Freq.Offset	:DISPlay:ANALysis:RECall:JTRansfer:FREQuency?	11-208
Vertical scale (Scale)	:DISPlay:ANALysis:RECall:JTRansfer:SCALe	11-209
Marker key	:DISPlay:ANALysis:RECall:JTRansfer:MDISplay	11-210
In Marker field <,>	:DISPlay:ANALysis:RECall:JTRansfer:MARKer	11-211
In Marker field <,> TableNo. In Marker field <,> Freq. (Hz) In Marker field <,> Gain (dB) In Marker field <,> Mask (Type) In Marker field <,> Mask Gain In Marker field <,> Mask Result	:DISPlay:ANALysis:RECall:JTRansfer:DATA?	11-212

:DISPlay:ANALysis:RECall:JTRansfer:TITLe? Query command

▽ Parameter None

- Response <title> = <STRING RESPONSE DATA>
  "Title string" Title string (max. 15 characters)

- Function Queries a title on the Analyze: Recall screen (Jitter transfer)

- Example use > :DISPlay:ANALysis:RECall:JTRansfer:TITLe?
  < "TITLE-DISP "

:DISPlay:ANALysis:RECall:JTRansfer:WTIMe? Query command

∇ Parameter None

- Response <wait> = <NR1 NUMERIC RESPONSE DATA>
  0 to 99 (s)
- Function Queries the set value of Waiting time on the Analyze: Recall screen (Jitter transfer).
- Example use Query the setting of Waiting time.
  > :DISPlay:ANALysis:RECall:JTRansfer:WTIMe?
  < 10

:DISPlay:ANALysis:RECall:JTRansfer:FREQuency? Query command

∇ Parameter None

- Response <numeric> = <NR2 NUMERIC RESPONSE DATA>
  -100.0 to 100.0 (ppm)
- Function Queries the frequency offset value on the Analyze: Recall screen (Jitter transfer).
- Example use > :DISPlay:ANALysis:RECall:JTRansfer:FREQuency?
  < -50.0

:DISPlay:ANALysis:RECall:JTRansfer:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

20	20 dB
10	10 dB
1	1 dB

- Function Sets the y axis scale of a graph on the Analyze: Recall screen (Jitter transfer).

- Example use Set the y axis scale of a graph for a jitter transfer characteristic measurement to 10 dB.
  > :DISPlay:ANALysis:RECall:JTRansfer:SCALe 10

:DISPlay:ANALysis:RECall:JTRansfer:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:RECall:JTRansfer:SCALe.

- Function Queries the value of the y axis scale of a graph on the Analyze: Recall screen (Jitter transfer).

- Example use > :DISPlay:ANALysis:RECall:JTRansfer:SCALe?
  < 10

:DISPlay:ANALysis:RECall:JTRansfer:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Recall screen (Jitter transfer).

- Example use Set the marker display of a jitter transfer characteristic measurement to ON:
  > :DISPlay:ANALysis:RECall:JTRansfer:MDISplay ON

:DISPlay:ANALysis:RECall:JTRansfer:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>
  0 Marker is set to OFF
  1 Marker is set to ON

- Function Queries the display status of the marker on the Analyze: Recall screen (Jitter transfer).

- Example use > :DISPlay:ANALysis:RECall:JTRansfer:MDISplay?
  < 1

:DISPlay:ANALysis:RECall:JTRansfer:MARKer

Setting command

∇ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Recall screen (Jitter transfer).

- Example use Move in the right direction.
  > :DISPlay:ANALysis:RECall:JTRansfer:MARKer RIGHt

:DISPlay:ANALysis:RECall:JTRansfer:DATA? Query command

∇ Parameter None

- Response <point>,<freq1>,<freq2>,<amp1><mask>,<mamp1>,<result>

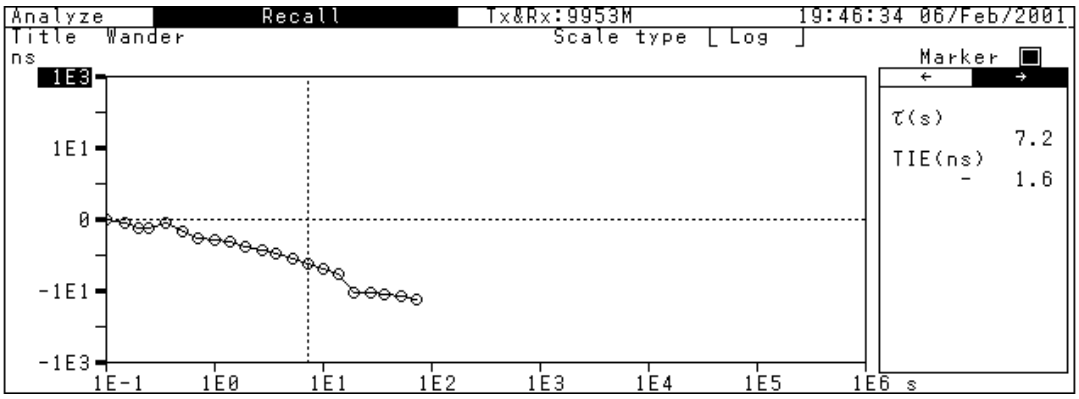
  <point> = <NR1 NUMERIC RESPONSE DATA>
  1 to 20 Measurement point
  <freq1> = <NR2 NUMERIC RESPONSE DATA>
  1.0 to 990.0 Modulated frequency (numeric)
  <freq2> = <CHARACTER RESPONSE DATA>
  HZ,KHZ,MHZ Modulated frequency (unit)
  <amp1>= <STRING RESPONSE DATA>
  Form6 Measurement result for jitter transfer characteristic (dB)
  <mask> = <CHARACTER PROGRAM DATA>
  G958A G958 Type A
  ANSIT1A ANSI T1.105.03 Type A
  ANSIT1B ANSI T1.105.03 Type B
  B1377 Bell 1377
  B253 Bell 253
  USER User
  <mamp1> = <STRING RESPONSE DATA>
  Form6 Measurement result for jitter transfer characteristic (dB)
  <result> = <CHARACTER PROGRAM DATA>
  OK
  NG

  * When the marker is OFF, the following is output:
  < 0,0.0,HZ,"-------",----,"-------",--

- Function Queries the data indicated with the marker on the Analyze: Recall screen (Jitter transfer).

- Example use > :DISPlay:ANALysis:RECall:JTRansfer:DATA?
  < 15,100,HZ," -15.00",G958A,"-12.00",OK

d) Wander



Setting Item	Command	Reference Page
Title	:DISPlay:ANALysis:RECall:WANDer:TITLe?	11-214
Scale type	:DISPlay:ANALysis:RECall:WANDer:STYPe	11-215
Vertical scale (Log Scale)	:DISPlay:ANALysis:RECall:WANDer:LOG:SCALe	11-216
Vertical scale (Linear Scale)	:DISPlay:ANALysis:RECall:WANDer:LINear:SCALe	11-217
Horizontal scale (Scale)	:DISPlay:ANALysis:RECall:WANDer:MEAStime	11-218
Horizontal scale (Scale)	:DISPlay:ANALysis:RECall:WANDer:USER	11-219
Marker key	:DISPlay:ANALysis:RECall:WANDer:MDISplay	11-220
In Marker field <,>	:DISPlay:ANALysis:RECall:WANDer:MARKer	11-221
In Marker field <,> τ In Marker field <,> TIE	:DISPlay:ANALysis:RECall:WANDer:DATA?	11-222

:DISPlay:ANALysis:RECall:WANDer:TITLe? Query command

▽ Parameter None

- Response <title> = <STRING RESPONSE DATA>
  "Title string" Title string (max. 15 characters)

- Function Queries a title on the Analyze: Recall screen (Wander).

- Example use > :DISPlay:ANALysis:RECall:WANDer:TITLe?
  < "TITLE-DISP "

:DISPlay:ANALysis:RECall:WANDer:STYPe

Setting command

∇ Parameter <type> = <CHARACTER PROGRAM DATA>

LOG	Log display
LINear	Linear display

- Function Specifies the y axis scale type of a graph on the Analyze: Recall screen (Wander).

- Example use Set the y axis scale type on the Wander screen to Log.
  > :DISPlay:ANALysis:RECall:WANDer:STYPe LOG

:DISPlay:ANALysis:RECall:WANDer:STYPe?

Query command

∇ Parameter None

- Response <type> = <CHARACTER RESPONSE DATA>

LOG	log
LIN	linear

- Function Queries the y axis scale type on the Analyze: Recall screen (Wander).
- Example use > :DISPlay:ANALysis:RECall:WANDer:STYPe?
  < LOG

:DISPlay:ANALysis:RECall:WANDer:LOG:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>
1E12
1E9
1E6
1E3

- Function Sets the maximum value of the y axis scale for the wander data (Log type) on the Analyze: Recall screen (Wander).

- Example use Set the maximum value of the y axis scale to 10^6 ns.
> :DISPlay:ANALysis:RECall:WANDer:LOG:SCALe 1E6

:DISPlay:ANALysis:RECall:WANDer:LOG:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:RECall:WANDer:LOG:SCALe.

- Function Queries the maximum value of the y axis scale of a wander data (Log type) graph on the Analyze: Recall screen (Wander).

- Example use Query the maximum value of the y axis scale of a wander data (Log type) graph.
> :DISPlay:ANALysis:RECall:WANDer:LOG:SCALe?
< 1E6

:DISPlay:ANALysis:RECall:WANDer:LINear:SCALe Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>
1E12 (unit: dB)
1E11
1E10
1E9
1E8
1E7
1E6
1E5
1E4
1E3
1E2
1E1
1E0

- Function Sets the maximum value of the y axis scale of a wander data (Linear type) on the Analyze: Recall screen (Wander).

- Example use Set the maximum value of the y axis scale of a graph to 10^6 ns.
> :DISPlay:ANALysis:RECall:WANDer:LINear:SCALe 1E6

:DISPlay:ANALysis:RECall:WANDer:LINear:SCALe? Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:RECall:WANDer:LINear:SCALe.

- Function Queries the maximum value of the y axis scale of a wander data (Linear type) graph on the Analyze: Recall screen (Wander).

- Example use Query the maximum value of the y axis scale of a wander data (Linear type) graph.
> :DISPlay:ANALysis:RECall:WANDer:Linear:SCALe?
< 1E6

:DISPlay:ANALysis:RECall:WANDer:MEAStime

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

SEC120000	120000 sec
SEC12000	12000 sec
SEC1200	1200 sec
SEC120	120 sec
SEC12	12 sec
USER	User

- Function Sets the maximum value of the x axis of a wander data (Linear type) graph on the Analyze: Recall screen (Wander).

- Example use Set the maximum value of the x axis of a graph (Linear type) to 1,200.
  > :DISPlay:ANALysis:RECall:WANDer:MEAStime 1200

:DISPlay:ANALysis:RECall:WANDer:MEAStime?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:RECall:WANDer:MEAStime.

- Function Queries the maximum value of the x axis of a wander data (Linear type) graph on the Analyze: Recall screen (Wander).

- Example use Query the maximum value of the x axis of a graph.
  > :DISPlay:ANALysis:RECall:WANDer:MEAStime?
  < 1200

:DISPlay:ANALysis:RECall:WANDer:USER

Setting command

∇ Parameter <scale> = <DECIMAL NUMERIC PROGRAM DATA>
From 10 to the value of observation time in the saved status

* When a fraction is entered, it is rounded down:
  1280 becomes to 1200,
  and 12006 becomes to 12000.
* When the observation time is 120,000 seconds or more, the upper limit is 120,000 seconds.

- Function Sets the maximum value of the x axis on a wander data (Linear type) graph on the Analyze: Recall screen (Wander).
  (Only when :DISPlay:ANALysis:RECall:WANDer:MEAStime is set to USER.)

- Example use Set the maximum value of the x axis of a graph (Linear type) to 12,000.
  > :DISPlay:ANALysis:RECall:WANDer:USER 12000

  (Reference)
  Even if you specify >:DISPlay:ANALysis:RECall:WANDer:USER 12008, the value is set to 12,000.

:DISPlay:ANALysis:RECall:WANDer:USER?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:RECall:WANDer:USER.

- Function Queries the maximum value of the x axis of a wander data (Linear type) graph on the Analyze: Recall screen (Wander).

- Example use Query the maximum value of the x axis of a graph.
  > :DISPlay:ANALysis:RECall:WANDer:USER?
  < 12000

:DISPlay:ANALysis:RECall:WANDer:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Recall screen (Wander).

- Example use Set the marker display of a wander measurement to ON.
  > :DISPlay:ANALysis:RECall:WANDer:MDISplay ON

:DISPlay:ANALysis:RECall:WANDer:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>
  0 Marker is set to OFF
  1 Marker is set to ON

- Function Queries the display status of the marker on the Analyze: Recall screen (Wander).
- Example use > :DISPlay:ANALysis:RECall:WANDer:MDISplay?
  < 1

:DISPlay:ANALysis:RECall:WANDer:MARKer

Setting command

∇ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Recall screen (Wander).

- Example use Move in the right direction.
  > :DISPlay:ANALysis:RECall:WANDer:MARKer RIGHt

:DISPlay:ANALysis:RECall:WANDer:DATA? Query command

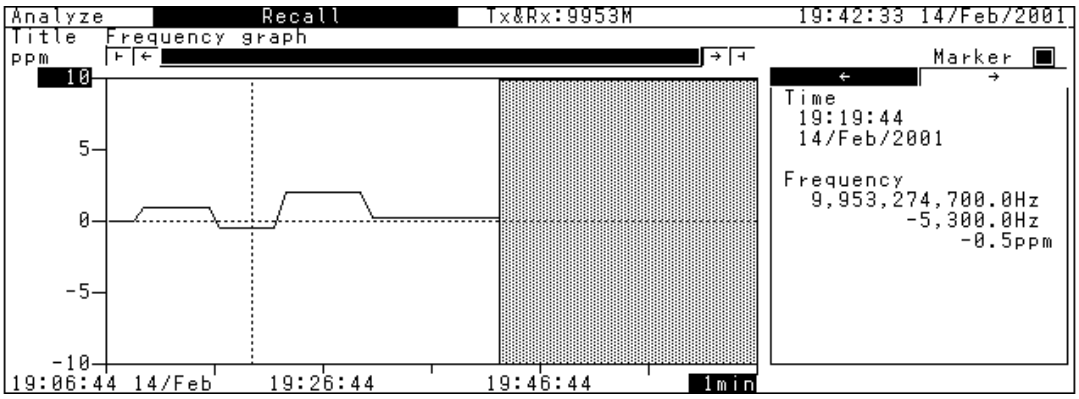
∇ Parameter None

- Response <time1>,<time2>
  <time1> = <STRING RESPONSE DATA>
  Form8' τ (s)
  <time2> = <STRING RESPONSE DATA>
  Form8 (ns)

  * When the marker is OFF, the following is output:
  < 0,0.0

- Function Queries data contents indicated with the marker on the Analyze: Recall screen (Wander).
- Example use > :DISPlay:ANALysis:RECall:WANDer:DATA?
  < " 0.35"," 3.0"

e) Frequency graph



Setting Item	Command	Reference Page
Title	:DISPlay:ANALysis:RECall:FGRaph:TITLe?	11-224
Graph scroll ←, →, -|, |-	:DISPlay:ANALysis:RECall:FGRaph:SCRoll	11-225
Vertical scale (Scale)	:DISPlay:ANALysis:RECall:FGRaph:SCALe	11-226
Marker key	:DISPlay:ANALysis:RECall:FGRaph:MDISplay	11-227
(Display start time)	:DISPlay:ANALysis:RECall:FGRaph:FROM	11-228
Horizontal scale (Scale) Horizontal scale (unit)	:DISPlay:ANALysis:RECall:FGRaph:INTerval	11-229
In Marker field <,>	:DISPlay:ANALysis:RECall:FGRaph:MARKer	11-230
In Marker field <,> (time) In Marker field <,> (Freq.) In Marker field <,> (relative Freq.) In Marker field <,> (ppm)	:DISPlay:ANALysis:RECall:FGRaph:DATA?	11-231

:DISPlay:ANALysis:RECall:FGRaph:TITLe?

Query command

▽ Parameter None

- Response <title> = <STRING RESPONSE DATA>
  "Title string" Title string (max. 15 characters)
- Function Queries a title on the Analyze: Recall screen (Frequency graph).
- Example use > :DISPlay:ANALysis:RECall:FGRaph:TITLe?
  < "TITLE-DISP "

:DISPlay:ANALysis:RECall:FGRaph:SCRoll

Setting command

▽ Parameter <type> = <CHARACTER PROGRAM DATA>

LEFT	Scrolls left.
RIGHt	Scrolls right.
TOP	Moves to the top.
BOTTom	Moves to the bottom.

- Function Specifies the graph scroll on the Analyze: Recall screen (Frequency graph).

- Example use Scrolls in the right direction.
  > :DISPlay:ANALysis:RECall:FGRaph:SCRoll RIGHt

:DISPlay:ANALysis:RECall:FGRaph:SCALe

Setting command

∇ Parameter <scale> = <CHARACTER PROGRAM DATA>

10	10 ppm
100	100 ppm
1000	1000 ppm

- Function Sets the y axis scale of a graph on the Analyze: Recall screen (Frequency graph).

- Example use Set the y axis scale of a Frequency graph to 10 ppm.
  > :DISPlay:ANALysis:RECall:FGRaph:SCALe 10

:DISPlay:ANALysis:RECall:FGRaph:SCALe?

Query command

∇ Parameter None

- Response <scale> = <CHARACTER RESPONSE DATA>
  Same as in the case of :DISPlay:ANALysis:RECall:FGRaph:SCALe

- Function Queries the value of the y axis scale of a graph on the Analyze: Recall screen (Frequency graph).

- Example use > :DISPlay:ANALysis:RECall:FGRaph:SCALe?
  < 10

:DISPlay:ANALysis:RECall:FGRaph:MDISplay

Setting command

∇ Parameter <boolean> = <BOOLEAN PROGRAM DATA>

OFF or 0	Sets marker to OFF
ON or 1	Sets marker to ON

- Function Sets the marker display on the Analyze: Recall screen (Frequency graph).
- Example use Set the marker display on Frequency graph to ON.
  > :DISPlay:ANALysis:RECall:FGRaph:MDISplay 1

:DISPlay:ANALysis:RECall:FGRaph:MDISplay?

Query command

∇ Parameter None

- Response <boolean> = <NR1 NUMERIC RESPONSE DATA>
  0 Marker is set to OFF
  1 Marker is set to ON
- Function Queries ON/OFF of the marker display on the Analyze: Recall screen (Frequency graph).
- Example use > :DISPlay:ANALysis:RECall:FGRaph:MDISplay?
  < 1

:DISPlay:ANALysis:RECall:FGRaph:FROM

Setting command

∇ Parameter <DECIMAL NUMERIC PROGRAM DATA>
<numeric1> = 2000 to 2093 (year)
<numeric2> = 1 to 12 (month)
<numeric3> = 1 to 31 (day)
<numeric4> = 0 to 23 (hour)
<numeric5> = 0 to 59 (minute)
<numeric6> = 0 to 59 (second)

* When the time specified with a parameter does not exists, the time next to the specified time is set.
Specifying the time before the beginning data time sets the beginning data time.
Specifying the time after the latest data time sets the latest data time.

- Function Sets the display start position of Frequency graph on the Analyze: Recall screen.

- Example use Start display from 11:30:40, July 28, 2000.
> :DISPlay:ANALysis:RECall:FGRaph:FROM 2000,7,28,11,30,40

:DISPlay:ANALysis:RECall:FGRaph:FROM?

Query command

∇ Parameter None

- Response <numeric1>,<numeric2>,<numeric3>,<numeric4>,<numeric5>,<numeric6>
= <NR1 NUMERIC RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:RECall:FGRaph:FROM.

- Function Queries the display start position of Frequency graph on the Analyze: Recall screen.

- Example use > :DISPlay:ANALysis:RECall:FGRaph:FROM?
< 2000,7,28,11,30,40

:DISPlay:ANALysis:RECall:FGRaph:INTerval

Setting command

∇ Parameter

<numeric> = <CHARACTER PROGRAM DATA>
1, 15, 60
<suffix> = <CHARACTER PROGRAM DATA>
M minute

- Function

Sets the width for one scale of the time axis on the Analyze: Recall screen (Frequency graph).

- Example use

Set the width for one scale to 1 minute.
> :DISPlay:ANALysis:RECall:FGRaph:INTerval 1,M

:DISPlay:ANALysis:RECall:FGRaph:INTerval?

Query command

∇ Parameter

None

- Response

<numeric> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:RECall:FGRaph:INTerval.
<suffix> = <CHARACTER RESPONSE DATA>
Same as in the case of :DISPlay:ANALysis:RECall:FGRaph:INTerval.

- Function

Queries the width for one scale of the time axis on the Analyze: Recall screen (Frequency graph).

- Example use

> :DISPlay:ANALysis:RECall:FGRaph:INTerval?
< 1,M

:DISPlay:ANALysis:RECall:FGRaph:MARKer

Setting command

∇ Parameter <marker> = <CHARACTER PROGRAM DATA>

LEFT	Moves left by 1 Div.
RIGHt	Moves right by 1 Div.

- Function Specifies the marker movement on the Analyze: Recall screen (Frequency graph).

- Example use Moves in the right direction.
  > :DISPlay:ANALysis:RECall:FGRaph:MARKer RIGHt

:DISPlay:ANALysis:RECall:FGRaph:DATA? Query command

▽ Parameter None

- Response

<time>,<frequency>,<ppm>
<time> = <year>,<month>,<day>,<hour>,<minute>,<second>
Year, month, day, hour, minute, second of the read data
When no data exists on the time specified with a parameter, the data on the time next to the specified time is output.
Setting the time before the measurement start time outputs the data at the measurement start time.
Setting the time after the log end time outputs the data at the log end time.

<year> = <NR1 NUMERIC RESPONSE DATA>
0,2000 to 2093 year
<month> = <NR1 NUMERIC RESPONSE DATA>
0,1 to 12 month
<day> = <NR1 NUMERIC RESPONSE DATA>
0,1 to 31 day
<hour> = <NR1 NUMERIC RESPONSE DATA>
0 to 23 hour
<minute> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 minute
<second> = <NR1 NUMERIC RESPONSE DATA>
0 to 59 second

<frequency> = <STRING RESPONSE DATA>
Monitor frequency (Hz) (relative)
(Form10)

<ppm> = <STRING RESPONSE DATA>
Monitor frequency deviation (ppm)
(Form11)

<frequency> = <STRING RESPONSE DATA>
Monitor frequency (Hz)
(Form10)

* When the marker is OFF, the following is output:
< 0,0,0,0,0,0,"----------","--------"

- Function

Queries the data indicated with the marker on the Analyze: Recall screen (Frequency graph).

- Example use

> :DISPlay:ANALysis:RECall:FGRaph:DATA?
< 2000,12,25,12,54,30," 1000.0","+1000.0"," 2000.0"

11.5 Other Commands without Related Screens

This section describes the following commands that are not related to any screen.

- Commands relating to MP1580A itself (System)
- Status commands

11.5.1 Command relating to MP1580A itself (system)

Commands related to conditions of MP1580A main body and key operations are shown below.

List of commands not related to any screen

Function	Command	Reference Page
Switching display screens	:DISPlay:DSELect[:NAME]	11-233
Starting a measurement	:SENSe:MEASure:STARt	11-234
Stoping a measurement	:SENSe:MEASure:STOP	11-234
Querying the measurement status	:SENSe:MEASure:STATe?	11-235
Querying the error status	:SYSTem:ERRor?	11-236
Saving the screen information on FD	:SYSTem:DUMP	11-237
Querying the SCPI version	:SYSTem:VERSion?	11-237
Switching terminators	:SYSTem:TERMination	11-238

Note: See "Appendix C: Error Message" for details of :SYSTem:ERRor?

:DISPlay:DSELect[:NAME]

∇ Parameter <display> = <STRING PROGRAM DATA>

"SETup"	The Setup screen
"TMENu"	The Test menu screen
"RESult"	The Result screen
"ANALysis"	The Analyze screen

- Function Selects a screen to be displayed.

- Example use When selecting the Test menu screen display:

```
> :DISPlay:DSELect:NAME "TMENu"
        or
> :DISPlay:DSELect "TMEN"
```

:DISPlay:DSELect[:NAME]?

∇ Parameter None

- Response <display> = <STRING RESPONSE DATA>

"SET"	The Setup screen
"TMEN"	The Test menu screen
"RES"	The Result screen
"ANAL"	The Analyze screen

- Function Queries the screen currently displayed.

- Example use

```
> :DISPlay:DSELect:NAME?
        or
> :DISPlay:DSELect?
< "SET"
```

:SENSe:MEASure:STARt

∇ Parameter None

- Function Starts the measurement specified in the Test menu screen.

  * When a measurement is started, lamps on the front panel of the MP1580A come on.
  * When this command is resent during a measurement, the measurement is restarted.
  * A calibration performed during Jitter transfer is also stated with this command, as below.

  To start calibration, set parameter for :SENSe:JITTer:TRANsfer:MODE to CAL and then send this command.

- Example use When starting manual measurement:
  > :SENSe:MEASure:STARt

:SENSe:MEASure:STOP

∇ Parameter None

- Function Stops a measurement in progress.

- Example use > :SENSe:MEASure:STOP

:SENSe:MEASure:STATe?

▽ Parameter None

- Response <meastype> = <CHARACTER RESPONSE DATA>

MAN	Manual measurement
JTR	Jitter transfer measurement
WAND	Wander measurement

<numeric> = <NR1 NUMERIC RESPONSE DATA>

0	Measurement completed
1	Measurement in progress

- Function Queries the current measurement status.

- Example use
```
> :SENSe:MEASure:STATe?
> MAN,1
```

:SYSTem:ERRor?

∇ Parameter None

- Response <error/event_number>,<error/event_description>
  <error/event_number> = <NR1 NUMERIC RESPONSE DATA>
  -32768 to 32767
  <error/event_description> = <STRING RESPONSE DATA>
  (See the "Appendix C: ERROR MESSAGE" for details on error messages.)

- Function Queries the error status.

- Example use > :SYSTem:ERRor?
  < -102,"Syntax error"

:SYSTem:DUMP

∇ Parameter None

- Function Saves the displayed information on the floppy disk.
- Example use > :SYSTem:DUMP

:SYSTem:VERSion?

∇ Parameter None

- Response <version> = <NR2 NUMERIC RESPONSE DATA>
  YYYY.V YYYY: Year
  V: Revision number
- Function Queries the SCPI version that the MP1580A is compliant with.
- Example use > :SYSTem:VERSion?
  < 1995.0

:SYSTem:TERMination

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>

0	Switches the terminator to LF+EOI.
1	Switches the terminator to CR+LF+EOI.

The parameter is always switched to 0 (LF+EOI) at the power-on as the initial status.
The terminator described above is the one that is used when the interface is GPIB. When the interface is RS-232C or Ethernet, EOI is excluded from the terminator.

- Function Switches a terminator of the response data.

- Example use When setting a terminator to CR+LF+EOI:
  >:SYSTem:TERMination 1

:SYSTem:TERMination?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  Same as in the case of :SYSTem:TERMination.

- Function Queries the terminator of the response data.

- Example use >:SYSTem:TERMination?
  < 0

11.5.2 STATus subsystem

Commands for the status register are shown below:

Command				Reference Page
:STATus	:PRESet			11-240
	:OPERation	:CONDition?		11-249
		:ENABle		11-250
		:ENABle?		11-250
		:INSTrument	:CONDition?	11-253
			:ENABle	11-254
			:ENABle?	11-254
			:NTRansition	11-256
			:NTRansition?	11-257
			:PTRansition	11-255
			:PTRansition?	11-255
			[:EVENt]?	11-253
		:INSTrument2	:CONDition?	11-258
			:ENABle	11-259
			:ENABle?	11-259
			:NTRansition	11-261
			:NTRansition?	11-261
			:PTRansition	11-260
			:PTRansition?	11-260
			[:EVENt]?	11-258
		:NTRansition		11-252
		:NTRansition?		11-252
		:PTRansition		11-251
		:PTRansition?		11-251
		[:EVENt]?		11-249
	:QUEStionable	:CONDition?		11-241
		:ENABle		11-242
		:ENABle?		11-242
		:NTRansition		11-244
		:NTRansition?		11-244
		:PTRansition		11-243
		:PTRansition?		11-243
		:TELecom	:CONDition?	11-245
			:ENABle	11-246
			:ENABle?	11-246
			:NTRansition	11-248
			:NTRansition?	11-248
			:PTRansition	11-247
			:PTRansition?	11-247
			[:EVENt]?	11-245
		[:EVENt]?		11-241

:STATus:PRESet

∇ Parameter None

- Function Initializes the status register. Refer to the "Section 7: Status Report".

- Example use Initialize the status register
  > :STATus:PRESet

:STATus:QUEStionable[:EVENt]?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  Sum (decimal number) of bits of the event register
  Sum of set bits provides the response.

  Bit definition of the QUEStionable status register
  512 (Bit 9) TEL (TELecom status register summary)

- Function Queries the event register contents of the Questionable status register.

- Example use Query the event register of the Questionable status register.
  > :STATus:QUEStionable:EVENt?
  < 512

:STATus:QUEStionable:CONDition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  Sum (decimal number) of bits of the condition register

- Function Queries the condition register contents of the Questionable status register.

- Example use Query the condition register of the Questionable status register.
  > :STATus:QUEStionable:CONDition?
  < 512

:STATus:QUEStionable:ENABle

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

Valid bit
512 (Bit 9) TEL (TELecom status register summary)

- Function Sets a mask value for the event enable register of the Questionable status register. Setting “0” for a parameter masks all the bits.

- Example use Set 512 for a mask value for the event enable register of the Questionable status register.
> :STATus:QUEStionable:ENABle 512

:STATus:QUEStionable:ENABle?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
Sum (decimal number) of bits of the event enable register

- Function Queries the event enable register contents of the Questionable status register.

- Example use Query the event enable register of the Questionable status register.
> :STATus:QUEStionable:ENABle?
< 512

:STATus:QUEStionable:PTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit
512 (Bit 9) TEL (TELecom status register summary)

- Function Sets the transition filter (transition in the positive direction) of the Questionable status register.

- Example use Set 512 for the transition filter (transition in the positive direction) of the Questionable status register.
> :STATus:QUEStionable:PTRansition 512

:STATus:QUEStionable:PTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the positive direction) contents of the Questionable status register.

- Example use Query the transition filter (transition in the positive direction) of the Questionable status register.
> :STATus:QUEStionable:PTRansition?
< 32767

:STATus:QUEStionable:NTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit
512 (Bit 9) TEL (TELecom status register summary)

- Function Sets the transition filter (transition in the negative direction) of the Questionable status register.

- Example use Set 32767 for the transition filter (transition in the negative direction) of the Questionable status register.
> :STATus:QUEStionable:NTRansition 32767

:STATus:QUEStionable:NTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the negative direction) contents of the Questionable status register.

- Example use Query the transition filter (transition in the negative direction) of the Questionable status register.
> :STATus:QUEStionable:NTRansition?
< 32767

:STATus:QUEStionable:TELecom[:EVENt]?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the event register
- Function Queries the contents of the event register of the TELecom status register.
- Example use Query the event register of the TELecom status register.
  > :STATus:QUEStionable:TELecom:EVENt?
  < 32767

:STATus:QUEStionable:TELecom:CONDition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the condition register
- Function Queries the contents of the condition register of the TELecom status register.
- Example use Query the condition register of the TELecom status register.
  > :STATus:QUEStionable:TELecom:CONDition?
  < 32767

:STATus:QUEStionable:TELecom:ENABle

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

Valid bit		
16 (Bit4) PWF	(Power Fail)	Indicates power failure occurred.
32 (Bit5) CKL	(Clock Loss)	Indicates a clock loss occurred.
64 (Bit6) Unlock	(Unlock)	Indicates unlock occurred.
128 (Bit7) HITC	(Hit count)	Indicates that the Hit count was counted in a jitter measurement.
256 (Bit8) REFLOS	(REF LOS)	Indicates that REF loss occurred in a wander measurement.

- Function Sets a mask value for the event enable register of the TELecom status register. Setting "0" for a parameter masks all the bits.

- Example use Set 32767 for the event enable register of the TELecom status register.
> :STATus:QUEStionable:TELecom:ENABle 32767

:STATus:QUEStionable:TELecom:ENABle?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

- Function Queries the contents of the event enable register of the TELecom status register.

- Example use Query the event enable register of the TELecom status register.
> :STATus:QUEStionable:TELecom:ENABle?
< 32767

:STATus:QUEStionable:TELecom:PTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit

16 (Bit4) PWF	(Power Fail)	Indicates power failure occurred.
32 (Bit5) CKL	(Clock Loss)	Indicates a clock loss occurred.
64 (Bit6) Unlock	(Unlock)	Indicates unlock occurred.
128 (Bit7) HITC	(Hit count)	Indicates that the Hit count is counted in a jitter measurement.
256 (Bit8) REFLOS	(REF LOS)	Indicates that REF loss occurred in a wander measurement.

- Function Sets the transition filter (transition in the positive direction) of the TELecom status register.

- Example use Set 32767 for the transition filter (transition in the positive direction) of the TELecom status register.
> :STATus:QUEStionable:TELecom:PTRansition 32767

:STATus:QUEStionable:TELecom:PTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the contents of the transition filter (transition in the positive direction) of the TELecom status register.

- Example use Query the transition filter (transition in the positive direction) of the TELecom status register.
> :STATus:QUEStionable:TELecom:PTRansition?
< 32767

:STATus:QUEStionable:TELecom:NTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit		
16 (Bit4) PWF	(Power Fail)	Indicates power failure occurred.
32 (Bit5) CKL	(Clock Loss)	Indicates a clock loss occurred.
64 (Bit6) Unlock	(Unlock)	Indicates unlock occurred.
128 (Bit7) HITC	(Hit count)	Indicates that the Hit count is counted in a jitter measurement.
256 (Bit8) REFLOS	(REF LOS)	Indicates that REF loss occurred in a wander measurement.

- Function Sets the transition filter (transition in the negative direction) of the TELecom status register.

- Example use Set 32767 for the transition filter (transition in the negative direction) of the TELecom status register.
> :STATus:QUEStionable:TELecom:NTRansition 32767

:STATus:QUEStionable:TELecom:NTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the negative direction) contents of the TELecom status register.

- Example use Query the transition filter (transition in the negative direction) of the TELecom status register.
> :STATus:QUEStionable:TELecom:NTRansition?
< 32767

:STATus:OPERation[:EVENt]?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the event register
- Function Queries the contents of the event register of the OPERation status register.
- Example use Query the event register of the OPERation status register.
  > :STATus:OPERation:EVENt?
  < 32767

:STATus:OPERation:CONDition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the condition register
- Function Queries the condition register contents of the OPERation status register.
- Example use Query the condition register of the OPERation status register.
  > :STATus:OPERation:CONDition?

:STATus:OPERation:ENABle

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

Valid bit

16 (Bit4)	MEAS	(MEASuring) Indicates a measurement is in progress.
8192 (Bit13)	INST	(INSTrument status register summary) Indicates the INSTrument status register summary.
16384 (Bit14)	INST2	(INSTrument2 status register summary) Indicates the INSTrument2 status register summary.

- Function Sets a mask value for the event enable register of the OPERation status register.
Setting "0" for a parameter masks all the bits.

- Example use Set 32767 for the event enable register of the OPERation status register.
> :STATus:OPERation:ENABle 32767

:STATus:OPERation:ENABle?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

- Function Queries the contents of the event enable register of the OPERation status register.

- Example use Query the event enable register of the OPERation status register.
> :STATus:OPERation:ENABle?
< 32767

:STATus:OPERation:PTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit

16 (Bit4)	MEAS	(MEASuring) Indicates a measurement is in progress.
8192 (Bit13)	INST	(INSTrument status register summary) Indicates the INSTrument status register summary.
16384 (Bit14)	INST2	(INSTrument2 status register summary) Indicates the INSTrument2 status register summary.

- Function Sets the transition filter (transition in the positive direction) of the OPERation status register.

- Example use Set 32767 for the transition filter (transition in the positive direction) of the OPERation status register.
> :STATus:OPERation:PTRansition 32767

:STATus:OPERation:PTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the positive direction) contents of the OPERation status register.

- Example use Query the transition filter (transition in the positive direction) of the OPERation status register.
> :STATus:OPERation:PTRansition?
< 32767

:STATus:OPERation:NTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit		
16 (Bit4)	MEAS	(MEASuring) Indicates a measurement is in progress.
8192 (Bit13)	INST	(INSTrument status register summary) Indicates the INSTrument status register summary.
16384 (Bit14)	INST2	(INSTrument2 status register summary) Indicates the INSTrument2 status register summary.

- Function Sets the transition filter (transition in the negative direction) of the OPERation status register.

- Example use Set 32767 for the transition filter (transition in the negative direction) of the OPERation status register.
> :STATus:OPERation:NTRansition 32767

:STATus:OPERation:NTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the negative direction) contents of the OPERation status register.

- Example use Query the transition filter (transition in the negative direction) of the OPERation status register.
> :STATus:OPERation:NTRansition?
< 32767

:STATus:OPERation:INSTrument[:EVENt]?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the event register
- Function Queries the contents of the event register of the INSTrument status register.
- Example use Query the event register of the INSTrument status register.
  > :STATus:OPERation:INSTrument:EVENt?
  < 32767

:STATus:OPERation:INSTrument:CONDition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the condition register
- Function Queries the contents of the condition register of the INSTrument status register.
- Example use Query the condition register of the INSTrument status register.
  > :STATus:OPERation:INSTrument:CONDition?
  < 32767

:STATus:OPERation:INSTrument:ENABle

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

Valid bit

1 (Bit0)	LF	(Log Full) Indicates the log is fully occupied.
2 (Bit1)	LE	(Log Empty) Indicates the log is empty.
32 (Bit5)	JMC	(Jitter Measure Complete) Indicates a jitter measurement is completed.
64 (Bit6)	WMC	(Wander Measure Complete) Indicates a wander measurement is completed.
128 (Bit7)	EOP	(End Of Printer period) Indicates an intermediate print timing of a printer.
8192 (Bit13)	LGF	(Logging memory Full) Indicates that the Logging memory is fully occupied.
16384 (Bit14)	LGE	(Logging memory Empty) Indicates that the Logging memory is empty.

- Function Sets a mask value for the event enable register of the INSTrument status register.
Setting "0" for a parameter masks all the bits.

- Example use Set 32767 for the event enable register of the INSTrument status register.
> :STATus:OPERation:INSTrument:ENABle 32767

:STATus:OPERation:INSTrument:ENABle?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register.

- Function Queries the event enable register contents of the INSTrument status register.

- Example use Query the event enable register of the INSTrument status register.
> :STATus:OPERation:INSTrument:ENABle?
< 32767

:STATus:OPERation:INSTrument:PTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit

1 (Bit0)	LF	(Log Full) Indicates the log is fully occupied.
2 (Bit1)	LE	(Log Empty) Indicates the log is empty.
32 (Bit5)	JMC	(Jitter Measure Complete) Indicates a jitter measurement is completed.
64 (Bit6)	WMC	(Wander Measure Complete) Indicates a wander measurement is completed.
128 (Bit7)	EOP	(End Of Printer period) Indicates an intermediate print timing of a printer.
8192 (Bit13)	LGF	(Logging memory Full) Indicates that the Logging memory is fully occupied.
16384 (Bit14)	LGE	(Logging memory Empty) Indicates that the Logging memory is empty.

- Function Sets the transition filter (transition in the positive direction) of the INSTrument status register.

- Example use Set 32767 for the transition filter (transition in the positive direction) of the INSTrument status register.
> :STATus:OPERation:INSTrument:PTRansition 32767

:STATus:OPERation:INSTrument:PTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the positive direction) contents of the INStrument status register.

- Example use Query the transition filter (transition in the positive direction) of the INSTrument status register.
> :STATus:OPERation:INSTrument:PTRansition?
< 32767

:STATus:OPERation:INSTrument:NTRansition

▽ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit

1 (Bit0)	LF	(Log Full) Indicates the log is fully occupied.
2 (Bit1)	LE	(Log Empty) Indicates the log is empty.
32 (Bit5)	JMC	(Jitter Measure Complete) Indicates a jitter measurement is completed.
64 (Bit6)	WMC	(Wander Measure Complete) Indicates a wander measurement is completed.
128 (Bit7)	EOP	(End Of Printer period) Indicates an intermediate print timing of a printer.
8192 (Bit13)	LGF	(Logging memory Full) Indicates that the Logging memory is fully occupied.
16384 (Bit14)	LGE	(Logging memory Empty) Indicates that the Logging memory is empty.

- Function Sets the transition filter (transition in the negative direction) of the INSTrument status register.

- Example use Set 32767 for the transition filter (transition in the negative direction) of the INSTrument status register.
> :STATus:OPERation:INSTrument:NTRansition 32767

:STATus:OPERation:INSTrument:NTRansition?

▽ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the negative direction) contents of the INStrument status register.

- Example use Query the transition filter (transition in the negative direction) of the INSTrument status register.
  > :STATus:OPERation:INSTrument:NTRansition?
  < 32767

:STATus:OPERation:INSTrument2[:EVENt]?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the event register
- Function Queries the contents of the event register of the INStrument2 status register.
- Example use Query the event register of the INSTrument2 status register.
  > :STATus:OPERation:INSTrument2:EVENt?
  < 32767

:STATus:OPERation:INSTrument2:CONDition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
  0 to 32767 Sum (decimal number) of bits of the condition register
- Function Queries the contents of the condition register of the INStrument2 status register.
- Example use Query the condition register of the INSTrument2 status register.
  > :STATus:OPERation:INSTrument2:CONDition?
  < 32767

:STATus:OPERation:INSTrument2:ENABle

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

Valid bit

1 (Bit0)	FMP	(Freq. Monitor Period) Indicates an update timing of the frequency monitor.
2 (Bit1)	FGF	(Freq. Graph Full) Indicates that the frequency graph data is full.
4 (Bit2)	FGE	(Freq. Graph Empty) Indicates that the frequency graph data is empty.
2048 (Bit11)	RVAL	(Revaluation) Indicates that a re-evaluation is completed.

- Function Sets a mask value for the event enable register of the INSTrument2 status register.
Setting "0" for a parameter masks all the bits.

- Example use Set 32767 for the event enable register of the INSTrument2 status.
> :STATus:OPERation:INSTrument2:ENABle 32767

:STATus:OPERation:INSTrument2:ENABle?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the event enable register

- Function Queries the contents of the event enable register of the INSTrument2 status register.
Setting "0" for a parameter masks all the bits.

- Example use Query the event enable register of the INSTrument2 status register.
> :STATus:OPERation:INSTrument2:ENABle?
< 32767

:STATus:OPERation:INSTrument2:PTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit

1 (Bit0)	FMP	(Freq. Monitor Period) Indicates an update timing of the frequency monitor.
2 (Bit1)	FGF	(Freq. Graph Full) Indicates that the frequency graph data is full.
4 (Bit2)	FGE	(Freq. Graph Empty) Indicates that the frequency graph data is empty.
2048 (Bit11)	RVAL	(Revaluation) Indicates that a re-evaluation is completed.

- Function Sets the transition filter (transition in the positive direction) of the INSTrument2 status register.

- Example use Set 32767 for the transition filter (transition in the positive direction) of the INSTrument2 status register.
> :STATus:OPERation:INSTrument2:PTRansition 32767

:STATus:OPERation:INSTrument2:PTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the positive direction) contents of the INSTrument2 status register.

- Example use Query the transition filter (transition in the positive direction) of the INSTrument2 status register.
> :STATus:OPERation:INSTrument2:PTRansition?
< 32767

:STATus:OPERation:INSTrument2:NTRansition

∇ Parameter <numeric> = <DECIMAL NUMERIC PROGRAM DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

Valid bit

1 (Bit0)	FMP	(Freq. Monitor Period) Indicates an update timing of the frequency monitor.
2 (Bit1)	FGF	(Freq. Graph Full) Indicates that the frequency graph data is full.
4 (Bit2)	FGE	(Freq. Graph Empty) Indicates that the frequency graph data is empty.
2048 (Bit11)	RVAL	(Revaluation) Indicates that a re-evaluation is completed.

- Function Sets the transition filter (transition in the negative direction) of the INSTrument2 status register.

- Example use Set 32767 for the transition filter (transition in the negative direction) of the INSTrument2 status register.
> :STATus:OPERation:INSTrument2:NTRansition 32767

:STATus:OPERation:INSTrument2:NTRansition?

∇ Parameter None

- Response <numeric> = <NR1 NUMERIC RESPONSE DATA>
0 to 32767 Sum (decimal number) of bits of the transition filter

- Function Queries the transition filter (transition in the negative direction) contents of the INSTrument2 status register.

- Example use Query the transition filter (transition in the negative direction) of the INSTrument2 status register.
> :STATus:OPERation:INSTrument2:NTRansition?
< 32767

Appendix

Appendix A Conformance Information

A.1 SCPI Version

This instrument conforms to SCPI version 1995.0.

A.2 SCPI-Defined Commands

Remote commands of this instrument defined by SCPI 1993.0 are shown below:

:INSTrument:COUPle
:INSTrument:COUPle?
:CALCulate:DATA?
:SYSTem:DATE
:SYSTem:DATE?
:SYSTem:TIME
:SYSTem:TIME?
:SYSTem:ERRor?
:SYSTem:VERSion?
:STATus:PRESet
:STATus:QUEStionable[:EVENt]?
:STATus:QUEStionable:CONDition?
:STATus:QUEStionable:ENABle
:STATus:QUEStionable:ENABle?
:STATus:QUEStionable:PTRansition
:STATus:QUEStionable:PTRansition?
:STATus:QUEStionable:NTRansition
STATus:QUEStionable:NTRansition?
:STATus:OPERation[:EVENt]?
:STATus:OPERation:CONDition?
:STATus:OPERation:ENABle
:STATus:OPERation:ENABle?
:STATus:OPERation:PTRansition
:STATus:OPERation:PTRansition?
:STATus:OPERation:NTRansition
:STATus:OPERation:NTRansition?
:STATus:OPERation:INSTrument[:EVENt]?
:STATus:OPERation:INSTrument:CONDition?
:STATus:OPERation:INSTrument:ENABle
:STATus:OPERation:INSTrument:ENABle?
:STATus:OPERation:INSTrument:PTRansition
:STATus:OPERation:INSTrument:PTRansition?
:STATus:OPERation:INSTrument:NTRansition
:STATus:OPERation:INSTrument:NTRansition?

*IDN?
*RST
*TST?
*OPC
*OPC?
*WAI
*CLS
*ESE
*ESE?
*ESR?
*SRE
*SRE?
*STB?
*TRG
*PSC
*PSC?
*SAV
*RCL
*OPT?

* Commands other than those above are MP1580A specific commands that are not defined by SCPI.

This section describes the command tree by subsystem.

B.1 INSTrument subsystem

INSTrument subsystem selects the identical setting or the separate setting between transmission and reception.

Command		Reference screen
:INSTrument	:COUPle	Setup:Interface
	:COUPle?	

B.2 SOURce subsystem

SOURce subsystem handles control (setting and display) on the transmission side.

Command					Reference screen
:SOURce	:JITTer	:MANual	:AMPLitude	:TYPE	Test menu:Manual
				:TYPE?	
				:UIPP	Test menu:Manual
				:UIPP?	
			:FREQuency		Test menu:Manual
			:FREQuency?		
			:MODE		Setup:Jitter/Wander
			:MODE?		
			:RANGe		Test menu:Manual
			:RANGe?		
			:TDEV	:A0	Setup:Jitter/Wander
				:A0?	
				:DTYPe	Setup:Jitter/Wander
				:DTYPe?	
				:ATYPe	Setup:Jitter/Wander
				:ATYPe?	
				:BTYPe	Setup:Jitter/Wander
				:BTYPe?	
				:ETYPe	Setup:Jitter/Wander
				:ETYPe?	
				:ITYPe	Setup:Jitter/Wander
				:ITYPe?	
				:VAT0?	Setup:Jitter/Wander
				:VAT1	Setup:Jitter/Wander
				:VAT1?	
				:VAT2	Setup:Jitter/Wander
				:VAT2?	
				:VAT3	Setup:Jitter/Wander
				:VAT3?	
				:VAT4?	Setup:Jitter/Wander
				:DEFault	Setup:Jitter/Wander
		:TRANsfer	:PTABle	:COUNt	Test menu:Jitter transfer
				:COUNt?	
				:DATA	Setup:Jitter transfer
				:DATA?	
				:DATA2	Setup:Jitter transfer
				:DATA2?	
				:DEFault	Setup:Jitter transfer
				:DEFault2	Setup:Jitter transfer
				:TYPE	Setup:Jitter transfer
				:TYPE?	
			:TYPE		Test menu:Jitter transfer
			:TYPE?		
			:OFFSet		Test menu:Jitter transfer
			:OFFSet?		

Command					Reference screen
		:WANDgen	:MPDeviation		Test menu:Manual
			:MPDeviation?		
			:TDEV	:DTYPE	Test menu:Manual
				:DTYPE?	
				:ATYPe	Test menu:Manual
				:ATYPe?	
				:BTYPe	Test menu:Manual
				:BTYPe?	
				:ETYPe	Test menu:Manual
				:ETYPe?	
				:ITYPe	Test menu:Manual
				:ITYPe?	
			:TYPE		Test menu:Manual
			:TYPE?		
	:TELecom	:BRATe			Setup:Interface
		:BRATe?			
		:CLOCk	:SOURce		Setup:Interface
			:SOURce?		
		:JWANder	:MSELect		Test menu:Manual
			:MSELect?		
		:OFFSet			Test menu:Manual
		:OFFSet?			
	:WANDer	:AUTO	:MPDeviation		Test menu:Wander
			:MPDeviation?		
			:TDEV	:DTYPe	Test menu:Wander
				:DTYPe?	
				:ATYPe	Test menu:Wander
				:ATYPe?	
				:BTYPe	Test menu:Wander
				:BTYPe?	
				:ETYPe	Test menu:Wander
				:ETYPe?	
				:ITYPe	Test menu:Wander
				:ITYPe?	
			:TYPE		Test menu:Wander
			:TYPE?		
		:MANual	:AMPLitude	:NSEC	Test menu:Manual
				:NSEC?	
				:TYPE	Test menu:Manual
				:TYPE?	
				:UIPP	Test menu:Manual
				:UIPP?	
			:FREQuency		Test menu:Manual
			:FREQuency?		
			:MODE		Setup:Jitter/Wander
			:MODE?		

B.3 SENSe subsystem

SENSe subsystem handles measurement conditions and control (setting and display) on the reception side.

Command					Reference screen
:SENSe	:JITTer	:MANual	:FILTer		Test menu:Manual
			:FILTer?		
			:RANGe		Test menu:Manual
			:RANGe?		
			:THReshold		Test menu:Manual
			:THReshold?		
		:TRANsfer	:LOOPback		Test menu:Jitter transfer
			:LOOPback?		
			:MASK		Test menu:Jitter transfer
			:MASK?		
			:MODE		Test menu:Jitter transfer
			:MODE?		
			:MTABle	:DATA	Setup:Jitter transfer
				:DATA?	
				:DEFault	Setup:Jitter transfer
				:POINt	Setup:Jitter transfer
				:POINt?	
				:TYPE	Setup:Jitter transfer
				:TYPE?	
			:REValuation		Test menu:Jitter transfer
			:WTIMe		Test menu:Jitter transfer
			:WTIMe?		
	:TELecom	:BRATe			Setup:Interface
		:BRATe?			
	:MEASure	:STARt			Other command for Main body
		:STOP			Other command for Main body
		:STAte?			Other command for Main body
		:STIMe?			Result : Common
		:ELAPsed?			Result : Common
		:TYPE			Test menu:Manual
		:TYPE?			
		:PERiod			Test menu:Manual
		:PERiod?			
		:GRESolution			Setup:System
		:GRESolution?			
		:JWANder	:CORRection	:OFFSet	Test menu:Manual
				:OFFSet?	
			:JITTer	:MODE	Setup:Jitter/Wander
				:MODE?	
			:WANDer	:MODE	Setup:Jitter/Wander
				:MODE?	
	:WANDer	:AUTO	:INTerval		Test menu:Wander
			:INTerval?		
			:USER		Test menu:Wander
			:USER?		

B.4 DISPlay subsystem

DISPlay subsystem handles control (setting and display) on the screen.

Command				Reference screen
:DISPlay	:SETup	[:NAME]		Setup Common
		[:NAME]?		Setup Common
		:JTRansfer	:CONDition	Setup:Jitter transfer
			:CONDition?	
	:TMENu	[:NAME]		Test menu: Common
		[:NAME]?		
	:RESult	[:NAME]		Result : Common
		[:NAME]?		
		:JWANder	:MODE	Result :Manual
			:MODE?	
			:UNIT	Result :Manual
			:UNIT?	
		:WANDer	:DISPtype	Result :Wander
			:DISPtype?	
	:ANALysis	[:NAME]		Analyze: Common
		[:NAME]?		
		:LOGGing	:CLEar	Analyze:Logging
			:TITLe	Analyze:Logging
			:TITLe?	
			:SCRoll	Analyze:Logging
		:FGRaph	:DATA?	Analyze:Frequency graph
			:FROM	Analyze:Frequency graph
			:FROM?	
			:INTerval	Analyze:Frequency graph
			:INTerval?	
			:MARKer	Analyze:Frequency graph
			:MDISplay	Analyze:Frequency graph
			:MDISplay?	
			:SCALe	Analyze:Frequency graph
			:SCALe?	
			:SCRoll	Analyze:Frequency graph
			:TITLe	Analyze:Frequency graph
			:TITLe?	
		:FMONitor	:FREQuency?	Analyze:Frequency monitor
			:PAUSe	Analyze:Frequency monitor
			:PAUSe?	
		:JTRansfer	:DATA?	Analyze:Jitter transfer
			:MDISplay	Analyze:Jitter transfer
			:MDISplay?	
			:SCALe	Analyze:Jitter transfer
			:SCALe?	
			:MARKer	Analyze:Jitter transfer
			:TITLe	Analyze:Jitter transfer
			:TITLe?	

:DISPlay		:PEAK	:DATA?		Analyze:Peak jitter
			:DTYPE		Analyze:Peak jitter
			:DTYPE?		
			:FROM		Analyze:Peak jitter
			:FROM?		
			:INTerval		Analyze:Peak jitter
			:INTerval?		
			:MARKer		Analyze:Peak jitter
			:MDISplay		Analyze:Peak jitter
			:MDISpay?		
			:SCALe		Analyze:Peak jitter
			:SCALe?		
			:SCRoll		Analyze:Peak jitter
			:TITLe		Analyze:Peak jitter
			:TITLe?		
		:WANDer	:DATA?		Analyze:Wander
			:LINear	:SCALe	Analyze:Wander
				:SCALe?	
			:LOG	:SCALe	Analyze:Wander
				:SCALe?	
			:MDISplay		Analyze:Wander
			:MDISplay?		
			:MEAStime		Analyze:Wander
			:MEAStime?		
			:MARKer		Analyze:Wander
			:STYPe		Analyze:Wander
			:STYPe?		
			:TITLe		Analyze:Wander
			:TITLe?		
			:USER		Analyze:Wander
			:USER?		
		:RECall	:LOGGing	:DATA?	Analyze:Recall(Logging)
				:TITLe?	Analyze:Recall(Logging)
				:SCRoll	Analyze:Recall(Logging)
			:FGRaph	:DATA?	Analyze:Recall(Frequency graph)
				:FROM	Analyze:Recall(Frequency graph)
				:FROM?	
				:INTerval	Analyze:Recall(Frequency graph)
				:INTerval?	
				:MARKer	Analyze:Recall(Frequency graph)
				:MDISplay	Analyze:Recall(Frequency graph)
				:MDISplay?	
				:SCALe	Analyze:Recall(Frequency graph)
				:SCALe?	
				:SCRoll	Analyze:Recall(Frequency graph)
				:TITLe?	Analyze:Recall(Frequency graph)
			:JTRansfer	:DATA?	Analyze:Recall(Jitter transfer)
				:MDISplay	Analyze:Recall(Jitter transfer)
				:MDISplay?	
				:SCALe	Analyze:Recall(Jitter transfer)
				:SCALe?	
				:MARKer	Analyze:Recall(Jitter transfer)
				:TITLe?	Analyze:Recall(Jitter transfer)
				:WTIMe?	Analyze:Recall(Jitter transfer)
				:FREQuency?	Analyze:Recall(Jitter transfer)

:DISPlay			:PEAK	:DATA?		Analyze:Recall(Peak jitter)
				:DTYPE		Analyze:Recall(Peak jitter)
				:DTYPE?		
				:FROM		Analyze:Recall(Peak jitter)
				:FROM?		
				:INTerval		Analyze:Recall(Peak jitter)
				:INTerval?		
				:MARKer		Analyze:Recall(Peak jitter)
				:MDISplay		Analyze:Recall(Peak jitter)
				:MDISpay?		
				:SCALe		Analyze:Recall(Peak jitter)
				:SCALe?		
				:SCRoll		Analyze:Recall(Peak jitter)
				:TITLe?		Analyze:Recall(Peak jitter)
			:TYPE?			
			:OFFset?			
			:WANDer	:DATA?		Analyze:Recall(Wander)
				:LINear	:SCALe	Analyze:Recall(Wander)
					:SCALe?	
				:LOG	:SCALe	Analyze:Recall(Wander)
					:SCALe?	
				:MDISplay		Analyze:Recall(Wander)
				:MDISplay?		
				:MEAStime		Analyze:Recall(Wander)
				:MEAStime?		
				:MARKer		Analyze:Recall(Wander)
				:STYPe		Analyze:Recall(Wander)
				:STYPe?		
				:TITLe?		Analyze:Recall(Wander)
				:USER		Analyze:Recall(Wander)
				:USER?		
	:DSELect	[:NAME]				Other command for Main body
		[:NAME]?				

B.5 ROUTe subsystem

ROUTe subsystem handles control (setting and display) of the internal signal routes of the measuring instrument.

Command			Reference screen
:ROUTe	:THRough	:JITTer	Setup:Interface
		:JITTer?	

B.6 CALCulate subsystem

CALCulate subsystem displays measured results.

Command			Reference screen
:CALCulate	:DATA?		Result : Common
	:LOGGing	:DATA?	Analyze:Logging
		:ADATA?	Analyze:Logging
	:FGRaph	:DATA?	Analyze:Frequency graph

B.7 SYSTem subsystem

SYSTem subsystem handles control (setting and display) of a printer, memory, floppy disk, etc.

Command				Reference screen
:SYSTem	:DATE			Setup:System
	:DATE?			
	:TIME			Setup:System
	:TIME?			
	:DUMP			Other command for Main body
	:ERRor?			Other command for Main body
	:LED	:HISTory		Other command for Main body
		:HISTory?		
		:RESet		Other command for Main body
	:MEMory	:ANALysis	:ACLear	Setup:Memory
			:CLEar	Setup:Memory
			:LABel?	Setup:Memory
			:RECall	Setup:Memory
			:STORe	Analyze: Common
		:CLEar		Setup:Memory
		:LABel		Setup:Memory
		:LABel?		
		:RECall		Setup:Memory
		:STORe		Setup:Memory
	:MMEMory	:CATalog?		Setup:Floopy
		:CDIRectory		Setup:Floopy
		:DELete		Setup:Floopy
		:INITialize		Setup:Floopy
		:MDIRectory		Setup:Floopy
		:RECall		Setup:Floopy
		:RELoad		Setup:Floopy
		:REName		Setup:Floopy
		:STORe		Setup:Floopy
	:PRINt	:APRINt		Setup:Print/Logging
		:APRINt?		
		:AUTO	:MCONdition	Setup:Print/Logging
			:MCONdition?	
		:IDATa	:PERiod	Setup:Print/Logging
			:PERiod?	
			[:TYPE]	Setup:Print/Logging
		:IDATa?		Setup:Print/Logging
		:JITTer	:SET	Setup:Print/Logging
			:SET?	
		:LDATa	:SET	Setup:Print/Logging
			:SET?	
		:LOGGing		Setup:Print/Logging
		:MANual	:MCONdition	Setup:Print/Logging
			:MCONdition?	
			:SQUelch	Setup:Print/Logging
	:VERSion?			Other command for Main body
	:TERMination			Other command for Main body
	:TERMination?			

B.8 STATus subsystem

STATus subsystem handles control (setting and display) of a status register.
For details of the following commands, see to 10.5.2.

Command			
:STATus	:PRESet		
	:OPERation	:CONDition?	
		:ENABle	
		:ENABle?	
		:INSTrument	:CONDition?
			:ENABle
			:ENABle?
			:NTRansition
			:NTRansition?
			:PTRansition
			:PTRansition?
			[:EVENt]?
		:INSTrument2	:CONDition?
			:ENABle
			:ENABle?
			:NTRansition
			:NTRansition?
			:PTRansition
			:PTRansition?
			[:EVENt]?
		:NTRansition	
		:NTRansition?	
		:PTRansition	
		:PTRansition?	
		[:EVENt]?	
	:QUEStionable	:CONDition?	
		:ENABle	
		:ENABle?	
		:NTRansition	
		:NTRansition?	
		:PTRansition	
		:PTRansition?	
		:TELecom	:CONDition?
			:ENABle
			:ENABle?
			:NTRansition
			:NTRansition?
			:PTRansition
			:PTRansition?
			[:EVENt]?
		[:EVENt]?	

Appendix C SCPI Error Messages

SCPI defines codes and messages associated with errors as a response to the SCPI command, :SYSTem:ERRor? This section describes details of error messages supported by MP1580A.

C.1 Command Error

The error code [-199,-100] indicates that an IEEE488.2 syntax error has occurred. In this case, bit 5 for an event status register of MP1580A is set. The following event result an error:

- MP1580A receives a message that violates the IEEE488.2 standard.
- MP1580A receives a header not conforming to the device specific command and the common command definitions.
- GET (Group Execute Trigger) is transmitted into a program message.

Error messages for command error

Code	Message	Error detection condition
-101	Invalid character	Invalid characters are contained in the header or character data. Note: - Mnemonic always begins with an upper/lower case character, followed by any combination of characters A to Z, a to z, underline "_", and numeric 0 to 9. - The character data is defined in the same way as the program mnemonic. Example: When # is included in the header: :SENSe:JITTer:MANual:#ANGe 1000 When * is included in a character parameter: :SENSe:JITTer:MANual:FILTer LPHP*
-102	Syntax error	A command consists of only "*" excluding the white space.
-104	Data type error	The parameter type differs from the specified one. Note: - Character or string data is transmitted while numeric data is specified. - Numeric or string data is transmitted while character data is specified. Example: When character data is transmitted to a command with a parameter of numeric data: :SENSe:MEASure:PERiod 1A,D
-108	Parameter not allowed	The number of parameters differs from the specified one. Example: When three parameters are transmitted where two parameters are specified: :SENSe:MEASure:PERiod 1,S,2 When no parameter is added where a parameter is required: :SENSe:MEASure:TYPE
-110	Command header error	The beginning of a command excluding the white space is neither a command separator nor a character (including the case of the separator only).
-112	Program mnemonic too long	The program mnemonic contains 13 characters or more. Note: - The maximum mnemonic length is 12 characters and does not include space between characters. Example: When a colon is missed from the header: :SOURce:JITTerWANDgenTDEVDTYPE
-113	Undefined header	The header is not defined for this instrument although its syntax is correct. Example: When the wrong header is written: :SENSe:TELecom:ALARm:PYPE OFF
-120	Numeric data error	The numeric data has an error.
-130	Suffix error	The suffix has an error. Example: Alphabets are used in a part of data whose parameter is defined as the numeric data: :SYSTem:MEMory:STORe 1A
-140	Character data error	Abnormal character data (character not defined by SCPI)
-144	Character data too long	The character data contains 13 characters or more. Example: When characters are improperly typed: :SYSTem:MEMory:ANALysis:STORe JITTERTransfer

Code	Message	Error detection condition
-150	String data error	The string data is not defined. Example: When single and double quotation marks are mixed: :DISPlay:SETup:NAME 'SYSTem" When the string is not enclosed with two single quotation marks: :DISPlay:SETup:NAME 'SYSTem

C.2 Execution Error

When the following error occurs, bit 4 for a standard event register is set.
An error is caused when the following event occurs:

- <PROGRAM DATA> that follows the header is not defined for MP1580A.
- The program message cannot be executed for the reason of MP1580A's condition.

Execution error messages

Code	Message	Error detection condition
-220	Parameter error	Parameter error occurs.
-221	Setting conflict	The parameter is correct but cannot be set due to this MP1580A's condition. Example: When trying to change the setting of Margin during measurement. When a command that is not associated with the displayed Recall screen is transmitted.
-222	Data out of range	The numeric data exceeds the range set for MP1580A.
-224	Illegal parameter value	The filename is incorrect (wrong parameter).
-250	Mass storage error	FDD error occurred.
-252	Missing media	There is no disk.
-253	Corrupt media	The disk is not formatted.
-254	Media full	The disk has no free space.
-256	File name not found	There is no file with the specified name.
-257	File name error	A directory with the name already exits.
-258	Media protected	The media is write-protected.

C.3 Device Unique Error

The device unique error indicates that an error other than a command error, a query error or an execution error has occurred. These include failures of hardware and firmware and self test errors. When the device unique error occurs, bit 3 for a standard event register is set.

Error message for device unique error

Code	Message	Error detection condition
-310	System error	System error has occurred.
-314	Save/recall memory lost	Local memory is lost.
-315	Configuration memory lost	Resume memory (backup memory of the current status) is lost.
-330	Self-test failed	An error has occurred during the self test.(*1)

Note:

*1: Self test is performed immediately after power-on.

Appendix D Sample Program

This section describes the flow chart of sample program for using MP1580A and precautions for creating programs.

D.1 Precautions for Creating Programs

- When creating a program using remote commands to use MP1580A, the following cautions should be taken:

1. Each device should be always initialized.
   It is supposed that in many cases the individual device is always not under the optimal condition in an actual usage because of panel operations of its own or execution of other programs.
   Always initializing each device allows execution of a program under the constant condition.
   When using the GPIB interface, initialize the device with IFC, DCL or *RST command.
   For the RS-232C, and Ethernet interface, be sure to execute the *RST command before initialization.

2. The remote status of the device should not be changed while executing a program.
   Pressing [Local] key in the remote state turns MP1580A devices into the local state.
   Note that in this case automatic measurement of the device may not normally operate and measurement data may become unreliable.

3. Exceptional processing of protocol
   For an expected exception, provide an exceptional processing section in a program to avoid suspension by the program execution error.

4. Buffer overflow should be prevented. (RS-232C, Ethernet)
   The RS-232C, Ethernet interface of MP1580A has data area of 512 bytes as the internal reception buffer.
   However, some processing may cause buffer overflow.
   To prevent problems caused by overflow, be sure not to transmit massive data (control command) at a time when operating remote control with the RS-232C, Ethernet interface.
   One method would be to transmit a series of commands and then "OPC?" command, and wait until a response is returned and then send the next command, thereby synchronizing between transmission and reception.

D.2 Sample Program Flow Chart

A flow chart for creating a remote control program is shown below.

- Measured result read (manual measurement)

a) Reading measured result