

TA720
Time Interval Analyzer
Communication Interface
U S E R ' S M A N U A L

Foreword

Thank you for purchasing the YOKOGAWA TA720 Time Interval Analyzer. This Communication Interface User's Manual describes the functions of the communication interface and command communications. To ensure correct use, please read this manual thoroughly before beginning operation. After reading the manual, keep it in a convenient location for quick reference whenever a question arises during operation of the software. The following manual is also provided in addition to this manual. Read them along with this manual.

Manual Title	Manual No.	Description
TA720 User's Manual	IM 704510-01E	Explains all functions and procedures of the TA720 excluding the command communication functions.

Notes

- The contents of this manual are subject to change without prior notice as a result of continuing improvements to the instrument's performance and functions.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA dealer as listed on the back cover of this manual.
- Copying or reproducing all or any part of the contents of this manual without the permission of Yokogawa Electric Corporation is strictly prohibited.
- The TCP/IP software of this product and the document concerning the TCP/IP software have been developed/created by YOKOGAWA based on the BSD Networking Software, Release 1 that has been licensed from California University.

Trademarks

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

Revisions

1st Edition: September 2002

How to Use This Manual

Structure of the Manual

This User's Manual consists of the following sections:

- Chapter 1 GP-IB Communication Interface**
Describes the setup procedures for using the GP-IB communication interface.
- Chapter 2 Ethernet Communication Interface (Option)**
Describes the setup procedures for using the Ethernet communication interface.
- Chapter 3 Program Syntax and Programming**
Describes the program syntax and points to note when creating the programs.
- Chapter 4 Commands**
Describes all the commands one by one.
- Chapter 5 Status Reports**
Describes the status byte, various registers, queues, and other information.
- Chapter 6 Sample Programs**
Introduces program examples made using Visual Basic.
- Appendix**
Describes reference material such as an ASCII character code table.
- Index**
Gives an index.

Conventions Used in This Manual

- **Conventions**

Type	Symbol	Meaning
Unit	k	1000 Example: 100 kHz
	K	1024 Example: 128 KB (memory size)
Symbols	Note	Describes useful information.

- **Symbols Used in the Syntax**

The following table indicates symbols that are used in the syntax mainly in chapter 3. These symbols are referred to as BNF (Backus-Naur Form) symbols. For a description of the symbols used with the data (conditions and values that are written following a space after the program header), see pages 3-5 and 3-6.

Symbol	Meaning	Example	Entry Example
<>	Defined value	WINDow<x> <x>=1 to 16	WINDOW2
{ }	Select from values given in { }	MODE {AUTO MANual}	MODE AUTO
	Exclusive OR	MODE {AUTO MANual}	MODE AUTO
[]	Can be omitted		:MEASure[:MODE]
...	Can be repeated		

Contents

Foreword	i
How to Use This Manual	ii
Chapter 1 GP-IB Communication Interface	
1.1 Names and Functions of the Parts Related to GP-IB Communications	1-1
1.2 Connecting the GP-IB Cable	1-2
1.3 GP-IB Communication Capabilities and Specifications	1-3
1.4 Settings on the TA720	1-5
1.5 Responses to Interface Messages	1-6
Chapter 2 Ethernet Communication Interface (Option)	
2.1 Names and Functions of the Parts Related to Ethernet Communications	2-1
2.2 Ethernet Communications Functions and Specifications	2-2
2.3 Connection Using the Ethernet Communication Interface	2-3
2.4 Setting the TA720	2-4
Chapter 3 Program Syntax and Programming	
3.1 Messages	3-1
3.2 Commands	3-3
3.3 Responses	3-4
3.4 Data	3-5
3.5 Synchronization with the Controller	3-7
Chapter 4 Commands	
4.1 List of Commands	4-1
4.2 ASCale Group	4-8
4.3 CALCulation Group	4-9
4.4 COMMunicate Group	4-25
4.5 DISPlay Group	4-27
4.6 FILE Group	4-31
4.7 HCOPy Group	4-34
4.8 HHIStoqram<x> and THIStoqram<x> Group	4-36
4.9 IHIStoqram Group	4-45
4.10 INPut Group	4-52
4.11 MEASure Group	4-55
4.12 MEMory Group	4-57
4.13 RECall Group	4-59
4.14 SAMPlE Group	4-60
4.15 SSTart Group	4-64
4.16 STARt Group	4-64
4.17 STATus Group	4-65
4.18 STOP Group	4-66
4.19 STORe Group	4-66
4.20 SYSTem Group	4-67
4.21 TVARiation<x> Group	4-69
4.22 UNIT Group	4-71
4.23 Common Command Group	4-72

Chapter 5 Status Report

5.1	Overview of the Status Report	5-1
5.2	Status Byte	5-2
5.3	Standard Event Register	5-3
5.4	Extended Event Register	5-4
5.5	Output Queue and Error Queue	5-5

Chapter 6 Sample Programs

6.1	Before Programming	6-1
6.2	Sample Program Image	6-1
6.3	Initialization, Error, and Functions for Execution	6-2
6.4	Setting Measurement Parameters or Querying the Settings (for GP-IB)	6-6
6.5	Querying the Measured Statistical Values of Period Measurement (for GP-IB)	6-8
6.6	Querying the Measured Statistical Values of Period A & Period B Measurement (for GP-IB)	6-11
6.7	Querying the Measured Data of A-to-B Time Interval Measurement (for GP-IB)	6-14
6.8	Setting Measurement Parameters or Querying the Settings (for Ethernet)	6-17
6.9	Querying the Measured Statistical Values of Pulse Width Measurement (for Ethernet)	6-19

Appendix

Appendix 1	ASCII Character Codes	App-1
Appendix 2	Error Messages	App-2
Appendix 3	Overview of IEEE.488.2-1992	App-4

Index

Chapter 1 GP-IB Communication Interface

1

1.1 Names and Functions of the Parts Related to GP-IB Communications

Front Panel

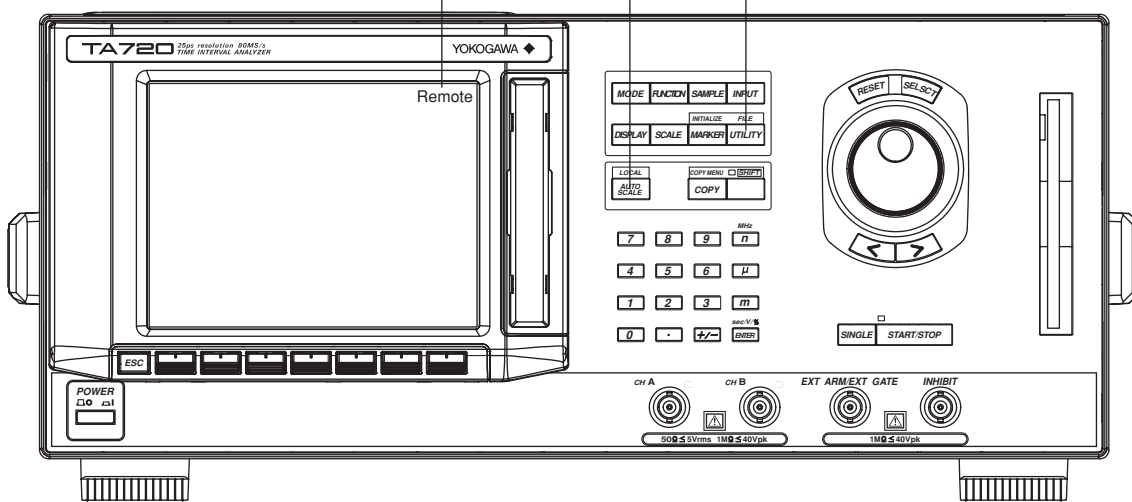
Remote indication appears at the upper right corner of the screen when in remote mode using communications.

LOCAL key (SHIFT+AUTO SCALE) key

Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled.

UTILITY key

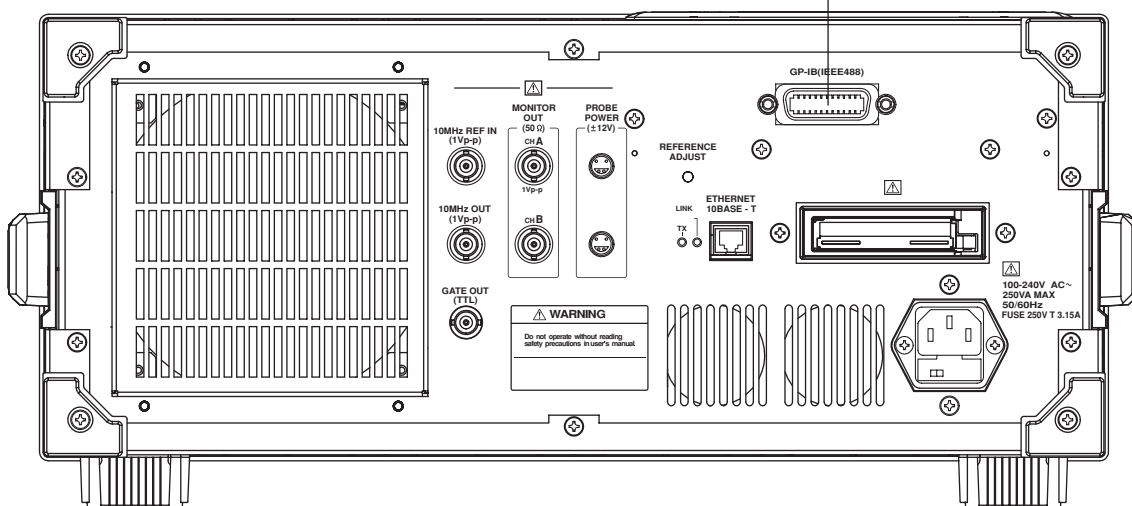
Press this key to configure communications.



Rear Panel

GP-IB connector

Connector used to connect the TA720 to the controller (PC) using a GP-IB cable.



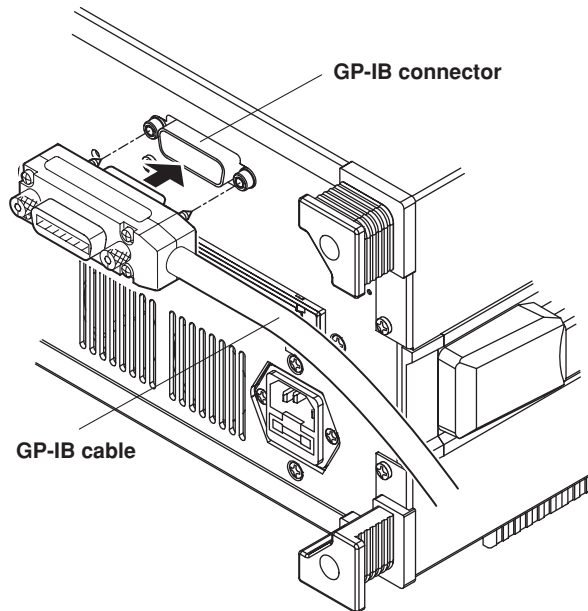
1.2 Connecting the GP-IB Cable

GP-IB Cable

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

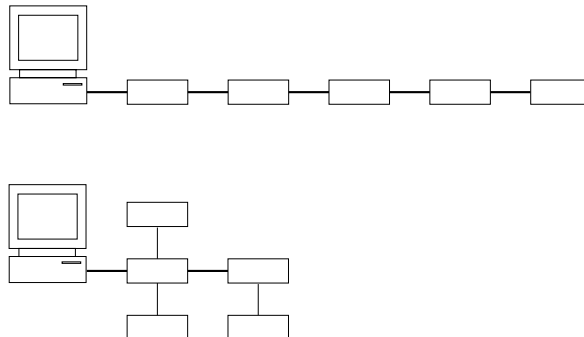
Connection Procedure

Connect the cable as shown below.



Precautions to Be Taken When Connecting Cables

- Firmly tighten the screws on the GP-IB cable connector.
- Multiple devices can be connected to a single GP-IB system. However, no more than 15 devices (including the controller) can be connected to a single system.
- When connecting multiple devices, each device must have its own unique address.
- Use a cable of length 2 m or less for connecting the devices.
- Make sure the total cable length does not exceed 20 m.
- When communicating, have at least two-thirds of the devices turned ON.
- When connecting multiple devices, connect them in a star or linear configuration (see the figure below). Loop and parallel configurations are not allowed.



1.3 GP-IB Communication Capabilities and Specifications

GP-IB Communication Capabilities

- **Listener capability**
 - All of the information that you can set with the panel keys can be set through the GP-IB interface except for turning ON/OFF the power and setting the communication parameters.
 - Receives commands from a controller requesting the output of setup information, measured data, and other information.
 - Also receives status report commands.
- **Talker capability**
Outputs setup information, measured data, and other information.

Note

Listen-only, talk-only, and controller capabilities are not available on this instrument.

Switching between Remote and Local Modes

- **When switching from local to remote mode**
Receiving a REN (Remote Enable) message from the controller when the instrument is in the local mode causes the instrument to switch to the remote mode.
 - REMOTE indication appears at the upper right corner of the screen (see page 1-1).
 - All keys other than the **LOCAL (SHIFT+AUTO SCALE)** key are locked.
 - The settings that existed in the local mode are maintained even when the instrument switches to the remote mode.
- **When switching from remote to local mode**
Pressing the **LOCAL (SHIFT+AUTO SCALE)** key when the instrument is in the remote mode causes the instrument to switch to the local mode. However, this act is invalid if the instrument has been set to Local Lockout mode (see page 1-6) by the controller.
 - REMOTE indication at the upper right corner of the screen disappears.
 - Key operations are enabled.
 - The settings that existed in the remote mode are maintained even when the instrument switches to the local mode.

1.3 GP-IB Communication Capabilities and Specifications

GP-IB Interface Specifications

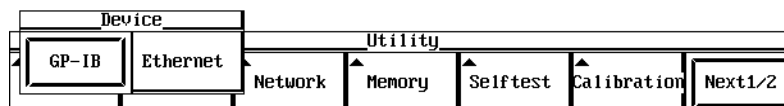
Electrical and mechanical specifications:	Conforms to IEEE St'd 488-1978 (JIS C1901-1987)
Functional specifications:	See table below.
Protocol:	Conforms to IEEE St'd 488.2-1992
Code:	ISO (ASCII) code
Mode:	Addressable mode
Address setting:	The address can be set in the range from 0 to 30 on the GP-IB setup menu that is played using the UTILITY key.
Clear remote mode:	Clear remote mode by pressing the LOCAL (SHIFT+AUTO SCALE) key. However, key operations are void when Local Lockout is enabled by the controller.

Function	Subset Name	Description
Source handshaking	SH1	Full source handshaking capability
Acceptor handshaking	AH1	Full acceptor handshaking capability
Talker	T6	Basic talker capability, serial polling, untalk on MLA (My Listen Address), and no talk-only capability
Listener	L4	Basic listener capability, unlisten on MTA (My Talk Address), and no listen-only capability.
Service request	SR1	Full service request capability
Remote local	RL1	Full remote/local capability
Parallel polling	PP0	No parallel polling capability
Device clear	DC1	Full device clear capability
Device trigger	DT1	Full device trigger capability
Controller	C0	No controller capability
Electrical characteristics	E1	Open collector

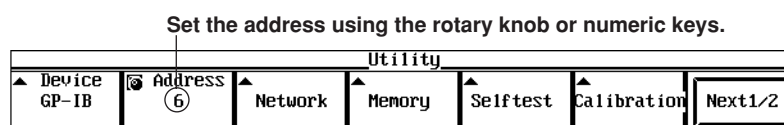
1.4 Settings on the TA720

Procedure

1. Press the **UTILITY** key to display the Utility menu.
2. Press the **Device** soft key to display the Device selection menu.
3. Select **GP-IB** using the soft key.



4. Set the address using the rotary knob or numeric keys.



Explanation

Setting the Address

Each device that can be connected via GP-IB has a unique address within the GP-IB system. This address is used to distinguish the device from others. Therefore, when you connect the TA720 to a PC, for example, make sure to assign a unique address to the TA720.

Select the address from the following: The initial value is 1.

Selectable range: 0 to 30

1.5 Responses to Interface Messages

What Is an Interface Message

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

- **Uni-line messages**

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

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

- **Multi-line messages**

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

- **Address commands**

These commands are valid when the instrument is designated as a listener or as a talker. The following five types are available.

- Commands that are valid on an instrument that is designated as a listener
GTL (Go To Local), SDC (Selected Device Clear), PPC (Parallel Poll Configure), and GET (Group Execute Trigger)

- Commands that are valid on an instrument that is designated as a talker
TCT (Take Control)

- **Universal commands**

These commands are valid on all instruments regardless of the listener and talker designations. The following five types are available.

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

- In addition, listener address, talker address, and secondary commands are also considered interface messages.
- The differences between SDC and DCL

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

Responses to Interface Messages

- **Responses to a uni-line message**

- IFC: Clears the talker and listener functions. Stops output if data are being output.
- REN: Switches between the remote and local modes.
- IDY: Not supported.

- **Responses to a multi-line message (address command)**

- GTL: Switches to the local mode.
- SDC: Clears the program message (command) being received and the output queue (see page 5-5).
- GET: Same operation as the *TRG command.
- The COMMunicate:WAIT command is immediately terminated.
- PPC or TCT: Not supported.

- **Responses to a multi-line message (universal command)**

- LLO: Disables the LOCAL key on the front panel to prohibit switching to the local mode.
- DCL: Same operation as the SDC message.
- SPE: Sets the talker function on all devices on the bus to serial polling mode. The controller polls the devices in order.
- SPD: Clears the serial polling mode of the talker function on all devices on the bus.
- PPU: Not supported.

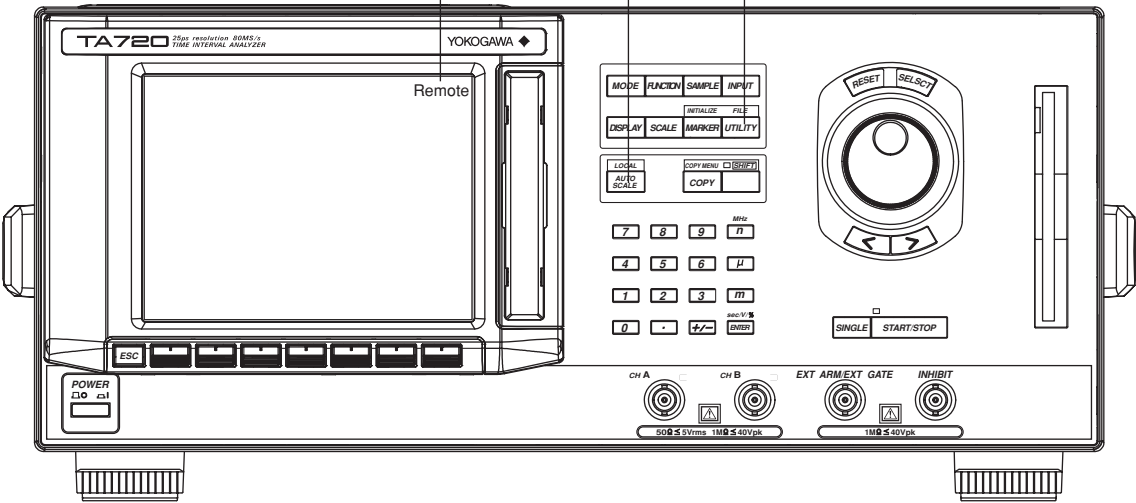
Chapter 2 Ethernet Communication Interface (Option)

2.1 Names and Functions of the Parts Related to Ethernet Communications Front Panel

Remote indication appears at the upper right corner of the screen when in remote mode using communications.

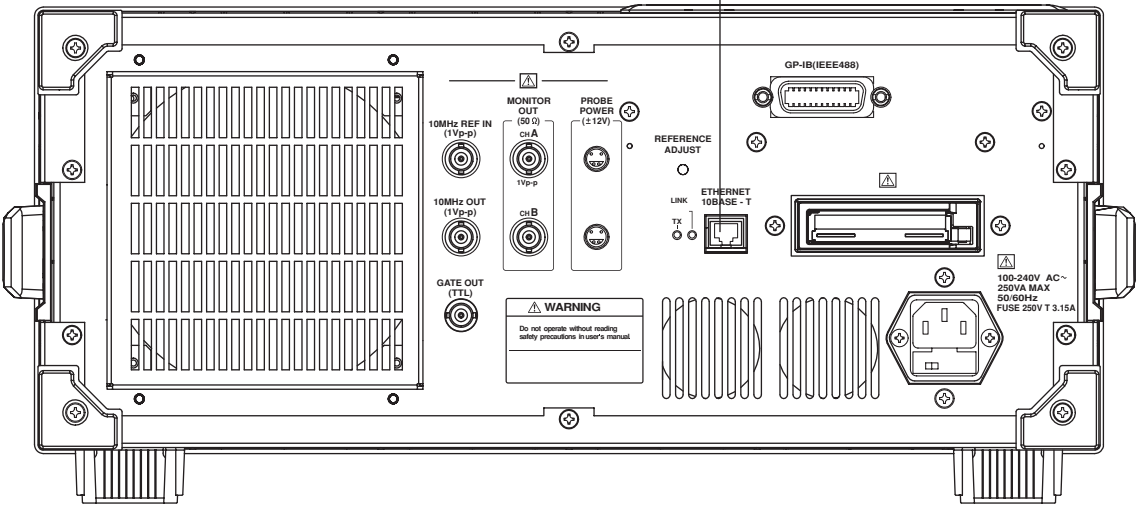
LOCAL key (SHIFT+AUTO SCALE) key
Press this key to clear the remote mode (controlled via communications) and enter the local mode in which key operations are enabled.

UTILITY key
Press this key to set the user name and password for user verification and TCP/IP parameters.



Rear Panel

Ethernet interface connector
Connector used to connect the TA720 to the controller (PC) using an Ethernet cable.
For details on how to connect the cable, see page 2-3.



2.2 Ethernet Communications Functions and Specifications

Reception Function

You can specify the same settings as those specified by front panel key operations.
Receives output requests for measured and computed data, setting parameters of the panel, and error codes.

Transmission Function

Outputs measured and computed data.
Outputs setting parameters of the panel and the status byte.
Outputs error codes that are generated.

Ethernet Communication Interface Specifications

Number of communication ports:	1
Electrical and mechanical specifications:	Conforms to IEEE802.3
Transmission system:	10BASE-T
Maximum transmission rate:	10 Mbps
Connector type:	RJ-45

Switching between Remote and Local Modes

- **When switching from local to remote mode**
If the TA720 receives a “:COMMunicate:REMOte ON” command from the PC when it is in the local mode, it switches to the remote mode.
 - REMOTE is displayed at the upper left corner of the screen.
 - All keys except the **LOCAL (SHIFT+AUTO SCALE)** key are disabled.
 - The settings that existed in the local mode are maintained even when the instrument switches to the remote mode.
- **When switching from remote to local mode**
Pressing the **LOCAL (SHIFT+AUTO SCALE)** key when the instrument is in the remote mode causes the instrument to switch to the local mode. However, this is void when the TA720 has received a “:COMMunicate:LOCKout ON” command from the PC (local lockout condition).
When the TA720 receives a “:COMMunicate:REMOte OFF” command from the PC, the TA720 switches to the local mode regardless of the local lock condition.
 - REMOTE indication at the upper right corner disappears.
 - Key operations are enabled.
 - The settings that existed in the remote mode are maintained even when the instrument switches to the local mode.

Note

The Ethernet communication interface cannot be used concurrently with the GP-IB interface.

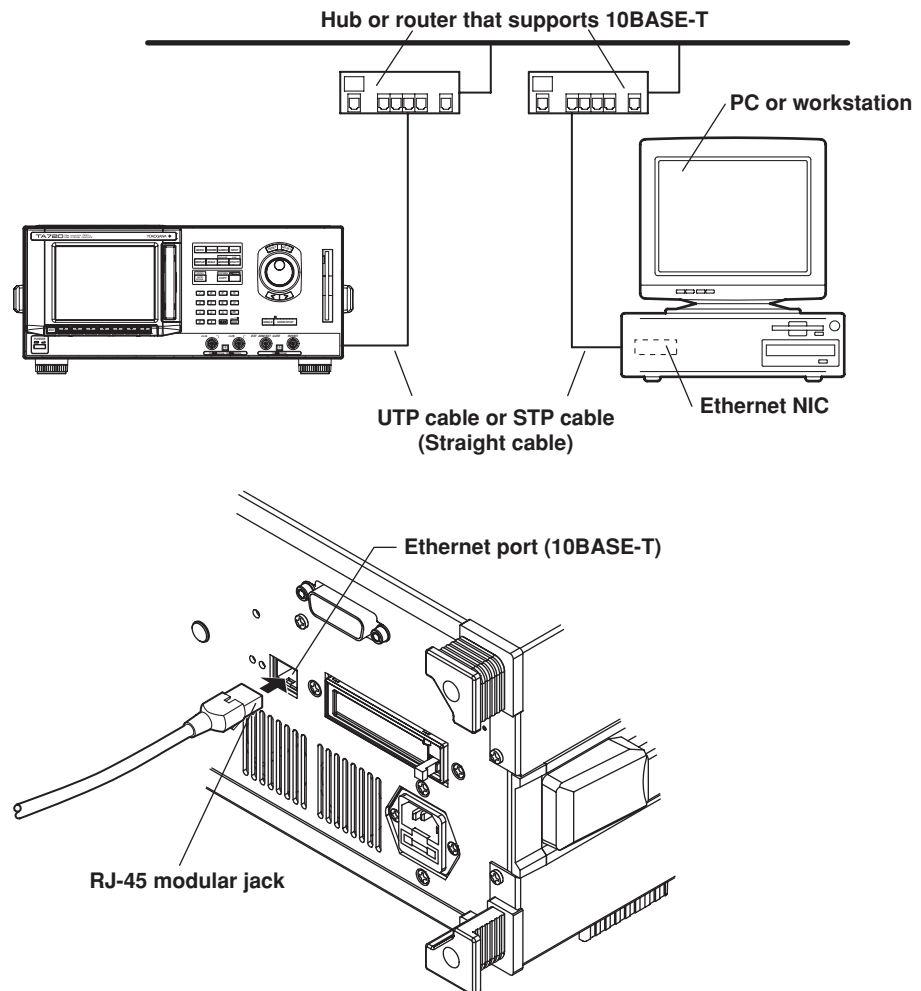
User Verification Function

When using the Ethernet communication interface, a user name and password are required when connecting to the network. The user name and password are set on the User Account pop-up window under the Utility menu of the TA720. For details, see section 2.4, “Setting the TA720.”

2.3 Connection Using the Ethernet Communication Interface

Connection Procedure

Connect a UTP (Unshielded Twisted-Pair) cable or an STP (Shielded Twisted-Pair) cable that is connected to a hub, for example, to the 10BASE-T port on the rear panel of the TA720.



Precautions to Be Taken When Connecting Cables

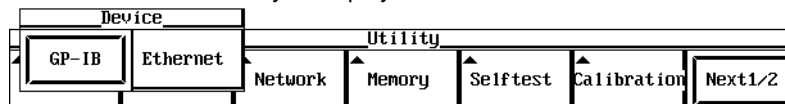
To connect the TA720 and a PC, be sure to use a straight cable via a hub. Operation is not guaranteed when the TA720 and the PC are connected one-to-one using a cross cable.

2.4 Setting the TA720

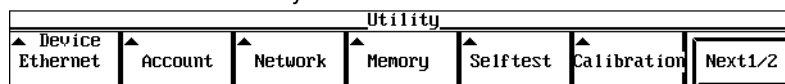
Procedure

- **Selecting the Communication Interface**

1. Press the **UTILITY** key to display the Utility menu.
2. Press the **Device** soft key to display the Device selection menu.



3. Press the **Ethernet** soft key.



Note

Only the communication interface selected by Device is enabled. The TA720 does not accept commands that are transmitted using an unselected GP-IB communication interface.

- **Setting the User Name, Password, and Timeout**

4. Press the **Account** soft key.
- The user verification dialog box opens.

5. Turn the rotary knob to move the cursor onto the **User Name**.
6. Press the SELECT key to display the keyboard.
7. Enter the user name. Specify anonymous if you wish to allow access to all users. To restrict access, enter the user name using up to 15 characters.

8. Turn the rotary knob to move the cursor to **Password**.
9. Press the SELECT key to display the keyboard.
10. Enter the password using up to 15 characters.
11. Turn the rotary knob to move the cursor onto the **Time Out**.
12. Press the SELECT key.
13. Set the timeout time using the rotary knob or numeric keys.

14. Press the SELECT key to confirm the new settings.

Note

- For details on the keyboard operation, see section 4.4, “Entering Values and Character Strings” in the *TA720 User's Manual IM704510-01E*.
- User name and password are case-sensitive.

- **Setting the TCP/IP**

To use the network interface function, the following TCP/IP settings must be entered.

- IP address
- Subnet mask
- Default gateway

For details on how to enter these parameters, see section 12.2, “Setting the TCP/IP” in the *TA720 User's Manual IM704510-01E*.

Explanation

Enter the following settings when using a controller to set information that can be specified through key operation on the TA720 or when outputting setting parameters or output waveform data to the controller.

- **Setting the User Name and Password**

The network interface has a user verification function. A user name and password for the TA720 are set in advance.

- Setting the User Name
Enter the password using up to 15 characters. The default setting is “anonymous.”
- Setting the password
Enter the password using up to 15 characters.

- **Setting the Timeout Time**

The connection to the TA720 is automatically dropped if there is no access to the TA720 for the specified time. The default setting is Infinite (no timeout).

- **Setting the TCP/IP**

For details, see section 12.2, “Setting the TCP/IP” in the *TA720 User's Manual IM704510-01E*.

Note

- If the user verification fails, connection to the TA720 is dropped.
- Password is not required if the user name is “anonymous.”
- If you change the user name, password, timeout time, or TCP/IP settings, power-cycle the TA720 to activate the new settings.

Chapter 3 Program Syntax and Programming

3.1 Messages

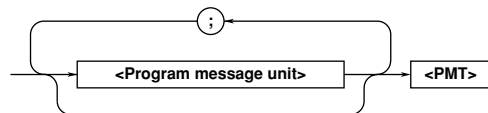
Message

Messages are used to exchange information between the controller and the instrument. Messages that are sent from the controller to the instrument are called program messages and messages that are sent back from the instrument to the controller are called response messages.

If a program message contains a message unit that requests a response (a query), the instrument returns a response message upon receiving the program message. A single response message is always returned in response to a single program message.

Program Message

Data that is sent from the controller to the instrument are called program messages. The program message format is shown below.



<Program Message Unit>

A program message consists of one or more program message units; each unit corresponds to one command. The instrument executes the received commands in order.

Each program message unit is separated by a semicolon (;).

For details regarding the format of the program message unit, see the next section.

Example

:MEASURE:MODE HHISTOGRAM;FUNCTION PERIOD,A<PMT>

Unit Unit

<PMT>

PMT is a program message terminator. The following three types are available.

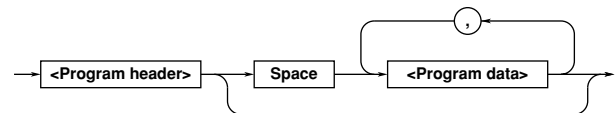
NL (New Line): Same as LF (Line Feed). ASCII code "0AH"

^END: END message that is defined in IEEE488.1 (EOI signal)
(The data byte that is sent simultaneously with the END message is the last data of the program message.)

NL ^END: NL with an END message added (NL is not included in the program message.)

• Program Message Unit Format

The program message unit format is shown below.



<Program header>

The program header indicates the command type. For details, see page 3-3.

<Program Data>

If certain conditions are required in executing a command, program data is added. A space (ASCII code "20H") separates the program data from the header. If there are multiple sets of program data, they are separated by commas (,).

For details, see page 3-5.

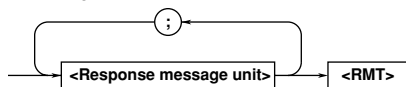
Example :SAMPLE:GATE:MODE EVENT<PMT>

Header Data

3.1 Messages

Response Message

Data that is sent from the instrument to the controller are called response messages. The response message format is shown below.



<Response Message Unit>

A response message consists of one or more response message units; each response message unit corresponds to one response.

Response message units are separated by a semicolon (;).

For details regarding the format of the response message unit, see page 3-4.

Example

```
:SAMPLE:GATE:MODE EXTERNAL;POLARITY POSITIVE<RMT>
```

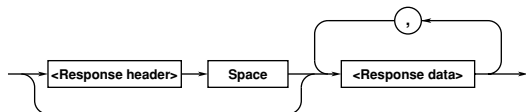
Unit Unit

<RMT>

A response message terminator. It is NL^END.

• Response Message Unit Format

The response message unit format is shown below.



<Response Header>

A response header sometimes precedes the response data. A space separates the data from the header.

For details, see page 3-4.

<Response Data>

Response data contains the content of the response.

If there are multiple sets of response data, they are separated by commas (,). For details, see page 3-5.

Example

```
500.0E-03<RMT> :SAMPLE:INTERVAL MINIMUM<RMT>
```

Data Header Data

If there are multiple queries in a program message, responses are made in the same order as the queries. In most cases, a single query returns a single response message unit, but there are a few queries that return multiple units. The first response message unit always corresponds to the first query, but the nth response unit may not necessarily correspond to the nth query. Therefore, if you want to make sure that every response is retrieved, divide the program messages into individual messages.

Precautions to Be Taken when Transferring Messages

- If a program message that does not contain a query is sent, the next program message can be sent at any time.
- If a program message that contains a query is sent, a response message must be received before the next program message can be sent. If the next program message is sent before the response message is received in its entirety, an error occurs. The response message that was not received is discarded.
- If the controller tries to receive a response message when there is none, an error occurs. If the controller tries to receive a response message before the transmission of the program message is complete, an error occurs.
- If a program message containing multiple message units is sent, and the message contains incomplete units, the instrument will attempt to execute the ones that are believed to be complete. However, these attempts may not always be successful. In addition, if the message contains queries, the responses may not be returned.

Deadlock

The instrument can store in its buffer program and response messages of length 1024 bytes or more (The number of available bytes varies depending on the operating conditions). When both the transmit and receive buffers become full at the same time, the instrument can no longer continue to operate. This state is called a deadlock. In this case, operation can be resumed by discarding the program message. Deadlock will not occur if the program message (including the <PMT>) is kept below 1024 bytes. Furthermore, deadlock never occurs if a program message does not contain a query.

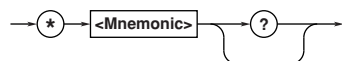
3.2 Commands

Command

There are three types of commands (program headers) that are sent from the controller to the instrument. They differ in their program header formats.

Common Command Header

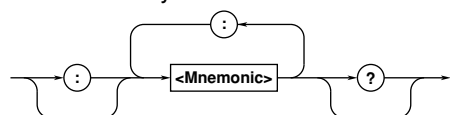
Commands that are defined in the IEEE 488.2-1992 are called common commands. The header format of a common command is shown below. An asterisk (*) is always placed in the beginning of a command.



An example of a common command: *CLS

Compound Header

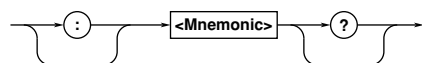
Dedicated commands used by the instrument are classified and arranged in a hierarchy according to their functions. The format of a compound header is shown below. A colon (:) must be used to specify a lower hierarchy.



An example of a compound header:
MEASURE:FUNCTION

Simple Header

These commands are functionally independent and do not have a hierarchy. The format of a simple header is shown below.



An example of a simple header: START

Note

A <mnemonic> is a character string made up of alphanumeric characters.

When Concatenating Commands

• Command Group

A command group is a group of commands that have common compound headers arranged in a hierarchy. A command group may contain sub-groups.

Example Group of commands related to sampling

```
:SAMPLE?
:SAMPLE:GATE?
:SAMPLE:GATE:MODE
:SAMPLE:GATE:EVENTSIZE
:SAMPLE:GATE:POLARITY
:SAMPLE:GATE:TIME
:SAMPLE:INTERVAL
:SAMPLE:ARMING:SOURCE
:SAMPLE:ARMING:DELAY:MODE
:SAMPLE:ARMING:DELAY:TIME
```

• When Concatenating Commands of the Same Group

The instrument stores the hierarchical level of the command that is currently being executed, and performs analysis on the assumption that the next command sent will also belong to the same level. Therefore, common header sections can be omitted for commands belonging to the same group.

Example :INPUT:ACHANNEL:COUPLING AC;
IMPEDANCE I50<PMT>

• When Concatenating Commands of Different Groups

If the following command does not belong to the same group, a colon (:) is placed in front of the header.

Example :MEASURE:MODE TSTAMP;;DISPLAY:
ITEM LIST<PMT>

• When Concatenating Simple Headers

If a simple header follows another command, a colon (:) is placed in front of the simple header.

Example :MEASURE:MODE TSTAMP;;
START<PMT>

• When Concatenating Common Commands

Common commands that are defined in the IEEE 488.2-1992 are independent of hierarchy. Colons (:) are not needed before a common command.

Example :MEASURE:MODE TSTAMP;*CLS;
FUNCTION PERIOD,A<PMT>

3.2 Commands/3.3 Responses

- **When Separating Commands with <PMT>**

If a terminator is used to separate two commands, each command is a separate message. Therefore, the common header must be specified for each command even when commands belonging to the same command group are being concatenated.

Example :MEASURE:MODE TSTAMP<PMT>:
 MEASURE:FUNCTION PERIOD,A<PMT>

Upper-level Query

An upper-level query is a query in which a question mark (?) is appended to the highest level command of a group. Execution of an upper-level query allows all settings that can be specified in the group to be received at once. Some query groups which are comprised of more than three hierarchical levels can output all the lower level settings.

Example :MEASURE?<PMT> -> :MEASURE:
 MODE HHISTOGRAM;
 FUNCTION PERIOD,A;SLOPE RISE

The response to an upper-level query can be transmitted as a program message back to the instrument. In this way, the settings that existed when the upper-level query was made can be restored. However, some upper-level queries will not return setup information that is not currently in use. It is important to remember that not all the group's information is necessarily returned as part of a response.

Header Interpretation Rules

The instrument interprets the header that is received according to the rules below.

- Mnemonics are not case sensitive.
Example MEASure can be also written as
 measure or Measure.
- The lower-case section of the header can be omitted.
Example MEASure can also be written as MEASU
 or MEAS
- The question mark (?) at the end of a header indicates that it is a query. The question mark (?) cannot be omitted.
Example The shortest abbreviation for
 "MEASure?" is "MEAS?."
- If the <x> (value) at the end of a mnemonic is omitted, it is interpreted as a 1.
Example If WINDow<x> is written as WIND, this
 represents WINDow1.
- The section enclosed by braces ([]) can be omitted.
Example :CALCulation[:WINDow1]:
 AVERage? can be written as
 :CALCulation:AVERage?.

However, the last section enclosed by braces ([]) cannot be omitted in an upper-level query.

3.3 Responses

When the controller sends a message unit that has a question mark (?) in its program header (query), the instrument returns a response message to the query. A response message is returned in one of the following two forms.

- Response consisting of a header and data
If the response can be used as a program message without any change, it is returned with a command header attached.
Example :SAMPLE:GATE:MODE?<PMT> ->
 :SAMPLE:GATE:MODE EVENT<RMT>
- Response consisting of data only
If the response cannot be used as a program message unless changes are made to it (query-only command), only the data section is returned. However, there are query-only commands that return responses with the header attached.
Example :STATUS:ERROR?<PMT> ->
 0,"NO ERROR"<RMT>
- **When You Wish to Return a Response without a Header**
Responses that return both header and data can be set so that only the data section is returned. The ":COMMunicate:HEADer" command is used to do this.
- **Abbreviated Form**
Normally, the lower-case section is removed from a response header before the response is returned to the controller. Naturally, the full form of the header can also be used. For this, the ":COMMunicate:VERBoSe" command is used.

3.4 Data

Data

A data section comes after the header. A space must be included between the header and the data. The data contains conditions and values. Data is classified as below.

Data	Description
<Decimal>	Value expressed as a decimal number (Example: The number of measurement samples -> :SAMPLE:GATE:EVENTsize 100)
<Voltage><Time> <Frequency> <Percentage>	Physical value (Example: Gate time -> :SAMPLE:GATE:TIME 1US)
<Register>	Register value expressed as either binary, octal, decimal or hexadecimal. (Example: Extended event register value -> :STATUS:EESE #HFE)
<Character data>	Predefined character string (mnemonic). Can be selected from { } (Example: Gate mode selection -> :SAMPLE:GATE:MODE {EVENT TIME EXTERNAL})
<Boolean>	Indicates ON and OFF. Set to ON, OFF or value (Example: Turn ON panorama display -> :DISPLAY:PANorama:STATE ON)
<Character string data>	Arbitrary character string (Example: Name of the file to be deleted -> :FILE:DELeTe:SETup "SETUP_1")
<Block data>	Arbitrary 8-bit data (Example: Response to acquired measured data -> #6000010ABCDEFGHIJ)

<Decimal>

<Decimal> indicates a value expressed as a decimal number, as shown in the table below. Decimal values are given in the NR form as specified in the ANSI X3.42-1975.

Symbol	Description	Example
<NR1>	Integer	125 -1 +1000
<NR2>	Fixed-point number	125.0 -90 +001.
<NR3>	Floating-point number	125.0E+0 -9E-1 +.1E4
<NRf>	Any of the forms <NR1> to <NR3> is allowed.	

- The instrument can receive decimal values that are sent from the controller in any of the forms, <NR1> to <NR3>. This is represented by <NRf>.
- For response messages that the instrument returns to the controller, the form (<NR1> to <NR3> to be used) is determined by the query. The same form is used regardless of the size of the value.
- In the case of <NR3>, the "+" after the "E" can be omitted, but the "-" cannot.
- If a value outside the setting range is entered, the value will be normalized so that it is just inside the range.
- If a value has more significant digits than the available resolution, the value is rounded.

<Voltage>, <Time>, <Frequency>, <Percent>

<Voltage>, <Time>, <Frequency>, and <Percent> indicate decimal values that have physical significance. Except for <Percent>, a <Multiplier> or <Unit> can be attached to <NRf>. They can be entered in any of the following forms.

Form	Example
<NRf><Multiplier><Unit>	5MV
<NRf><Unit>	5E-3V
<NRf><Multiplier>	5M
<NRf>	5E-3

• <Multiplier>

Multipliers which can be used are shown below.

Symbol	Word	Multiplier
EX	Exa	10 ¹⁸
PE	Peta	10 ¹⁵
T	Tera	10 ¹²
G	Giga	10 ⁹
MA	Mega	10 ⁶
K	Kilo	10 ³
M	Milli	10 ⁻³
U	Micro	10 ⁻⁶
N	Nano	10 ⁻⁹
P	Pico	10 ⁻¹²
F	Femto	10 ⁻¹⁵
A	Ato	10 ⁻¹⁸

• <Unit>

Units which can be used are shown below.

Symbol	Word	Description
V	Volt	Voltage
S	Second	Time
HZ	Hertz	Frequency
MHZ	Megahertz	Frequency
PCT	Percentage	Percentage

- <Multiplier> and <Unit> are not case sensitive.
- "U" should be used in place of "μ" in the data.
- "MA" is used for Mega to distinguish it from Milli.
- If both <Multiplier> and <Unit> are omitted, the default unit is used.
- Response messages are always expressed in the <NR3> form. Response messages are returned using the default unit without the <Multiplier> or <Unit>.

3.4 Data

<Register>

<Register> indicates an integer, and can be expressed in hexadecimal, octal, or binary as well as a decimal number. <Register> is used when each bit of the value has a particular meaning. <Register> is expressed in one of the following forms.

Form	Example
<NRf>	1
#H	#H0F
<Hexadecimal value made up of the digits 0 to 9 and A to F>	
#Q<Octal value made up of the digits 0 to 7>	#q777
#B<Binary value made up of the digits 0 and 1>	#B001100

- <Register> is not case sensitive.
- Response messages are always expressed as <NR1>.

<Character data>

<Character Data> is a specified string of character data (a mnemonic). It is mainly used to indicate options and is chosen from the character strings given in { }. For interpretation rules, refer to “Header Interpretation Rules” on page 3-4.

Form	Example
{EVENT TIME EXTERNAL}	EVENT

- As with the header, the “COMMunicate:VERBoSe” command can be used to select whether to return the response in the full form or in the abbreviated form.
- The “COMMunicate:HEADer” setting does not affect the <character data>.

<Boolean>

<Boolean> is data which indicates ON or OFF, and is expressed in one of the following forms.

Form	Example
{ON OFF <NRf>}	ON OFF 1 0

- When <Boolean> is expressed in the <NRf> form, “OFF” is selected if the rounded integer value is “0,” and ON for all other cases.
- A response message is always returned with a “1” if the value is ON and “0” if the value is OFF.

<Character String Data>

<Character string data> is not a specified character string like <Character data>. It is an arbitrary character string. The character string must be enclosed in single quotation marks (') or double quotation marks (").

Form	Example
<Character string data>	'ABC' "IEEE488.2-1987"

- If a character string contains a double quotation mark ("), the double quotation mark will be replaced by two concatenated double quotation marks (""). This rule also applies to a single quotation mark within a character string.
- A response message is always enclosed in double quotation marks (").
- <Character string data> is an arbitrary character string, therefore this instrument assumes that the remaining program message units are part of the character string if no single (') or double quotation mark (") is encountered. As a result, no error will be detected if a quotation mark is omitted.

<Block Data>

<Block data> is arbitrary 8-bit data. It is only used in response messages on the TA720 and is expressed in the following form.

Form	Example
#8<8-digit decimal number>	#800000010ABCDEFGHIJ
<data byte sequence>	

- #8
Indicates that the data is <Block data>.
- <8-digit decimal number>
Indicates the number of bytes of data (example: 00000010 = 10 bytes).
- <data byte sequence>
Expresses the actual data (example: ABCDEFGHIJ).
- Data is comprised of 8-bit values (0 to 255). This means that the ASCII code “0AH,” which stands for “NL,” can also be a code used for data. Hence, care must be taken when programming the controller.

3.5 Synchronization with the Controller

The TA720 does not support overlap commands, which allows the execution of the next command to start before the execution of the previous command is completed. If multiple sequential commands are sent consecutively, the execution of the next command is held until the execution of the previous command is completed.

Achieving Synchronization

Synchronization is sometimes required for reasons other than communications-related reasons, such as the activation of a trigger, even if a sequential command is used.

For example, if a “next program” message is transmitted to make an inquiry about the waveform data which has been acquired using single mode as the trigger mode, the “`MEMory:SEND?`” command is sometimes executed whether acquisition has been completed or not, causing a command execution error. `:SStart;MEMory:SEND?<PMT>`

In this case, the following method must be used to synchronize with the end of the acquisition.

- **Using the `STATus:CONDition?` query**

The “`STATus:CONDition?`” query is used to query the contents of the condition register (page 5-4). It is possible to judge whether acquisition is in progress or not by reading bit 0 of the condition register. Bit 0 is “1” if acquisition is in progress, and “0” if acquisition is stopped.

Example `:SStart<PMT>`
 `:STATus:CONDition?<PMT>`
 (Read the response. If bit 0 is 0, repeat this command until it becomes 1.)
 `:MEMory:SEND?<PMT>`

The `:MEMory:SEND?` command will not be executed until bit 0 of the condition register is set to “1.”

- **Using the extended event register**

The changes in the condition register can be reflected in the extended event register (page 5-4).

Example `:STATus:FILTer1 RISE;:STATus:EESE 1;EESR?;*SRE8;SStart<PMT>`
 (Wait for a service request occurrence)
 `:MEMory:SEND?<PMT>`

The “`STATus:FILTer1 RISE`” command sets the transition filter so that bit 0 (FILTer1) of the extended event register is set to “1” when bit 0 of the condition register changes from “0” to “1.”

The “`:STATus:EESE 1`” command is used to reflect only bit 0 of the extended event register to the status byte.

The “`STATus:EESR?`” command is used to clear the extended event register.

The “`*SRE`” command is used to generate a service request solely on the cause of the extended event register.

The “`:MEMory:SEND?`” command will not be executed until a service request is generated.

- **Using the `COMMunicate:WAIT` command**

The “`COMMunicate:WAIT`” command halts communications until a specific event is generated.

Example `:STATus:FILTer1 RISE;:STATus:EESR?;:SStart<PMT>`
 (Read the response to `STATus:EESR?`)
 `:COMMunicate:WAIT 1;:MEMory:SEND?<PMT>`

For a description of “`STATus:FILTer1 RISE`” and “`STATus:EESR?`” see the previous section regarding the extended event register.

The “`COMMunicate:WAIT 1`” command indicates that the program will wait for bit 0 of the extended event register to be set to “1.”

The `:MEMory:SEND?` command will not be executed until bit 0 of the extended event register is set to “1.”

Chapter 4 Commands

4.1 List of Commands

Command	Function	Page
AScale Group		
:AScale	Executes auto scaling.	4-8
CALCulation Group		
:CALCulation?	Queries all settings related to the statistical value.	4-10
:CALCulation:AREA	Queries the computation range or queries the current setting.	4-10
:CALCulation[:MEAS<x>]:AUTot?	Queries auto window T.	4-11
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TAverage?	Queries the average value when using time variation.	4-11
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TFLutter?	Queries the σ /AVE value (flutter) when using time variation.	4-11
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TJITter?	Queries the P-P/AVE value (jitter) when using time variation.	4-12
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TMAXimum?	Queries the maximum value when using time variation.	4-12
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TMINimum?	Queries the minimum value when using time variation.	4-13
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TPToPeak?	Queries the P-P value when using time variation.	4-13
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TRF?	Queries the RF value when using time variation.	4-14
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TSDeviation?	Queries the standard deviation (σ) when using time variation.	4-14
:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TSNumber?	Queries the number of samples for the statistical computation when using time variation.	4-15
:CALCulation[:MEAS<x>]:CONStt	Sets constant T or queries the current setting.	4-15
:CALCulation[:MEAS<x>][:WINDow<x>]:AVERage?	Queries the average value when using histogram.	4-16
:CALCulation[:MEAS<x>][:WINDow<x>]:DEViation?	Queries the deviation when using histogram.	4-16
:CALCulation[:MEAS<x>][:WINDow<x>]:DEVT?	Queries the deviation/T value when using histogram.	4-17
:CALCulation[:MEAS<x>][:WINDow<x>]:FLUTter?	Queries the σ /AVE value (flutter) when using histogram.	4-17
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}] :JITter?	Queries the σ /T value (jitter) when using histogram.	4-18
:CALCulation[:MEAS<x>][:WINDow<x>]:MAXimum?	Queries the maximum value when using histogram.	4-18
:CALCulation[:MEAS<x>][:WINDow<x>]:MEDian?	Queries the median value when using histogram.	4-19
:CALCulation[:MEAS<x>][:WINDow<x>]:MINimum?	Queries the minimum value when using histogram.	4-19
:CALCulation[:MEAS<x>][:WINDow<x>]:MODE?	Queries the most frequent value when using histogram.	4-20
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}] :PTOPeak?	Queries the P-P value when using histogram.	4-20
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}] :SDEViation?	Queries the standard deviation (σ) when using histogram.	4-21
:CALCulation[:MEAS<x>][:{WINDow<x> SUMMation}] :SNUMber?	Queries the number of samples for the statistical computation when using histogram.	4-21
:CALCulation:PARAmeter?	Queries whether all the statistical computations are turn ON or OFF.	4-22
:CALCulation:PARAmeter:CLEAr	Turns OFF all the statistical computation values.	4-22
:CALCulation:PARAmeter:AVERage	Turns ON/OFF the average computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAmeter:DEViation	Turns ON/OFF the deviation computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAmeter:DEVT	Turns ON/OFF the deviation/T computation when using histogram or queries the current setting.	4-22

4.1 List of Commands

Command	Function	Page
:CALCulation:PARAMeter:FLUTter	Turns ON/OFF the σ /AVE (flutter) computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAMeter:JITter	Turns ON/OFF the σ /T (jitter) computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAMeter:MAXimum	Turns ON/OFF the maximum value computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAMeter:MEDian	Turns ON/OFF the median value computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAMeter:MINimum	Turns ON/OFF the minimum value computation when using histogram or queries the current setting.	4-22
:CALCulation:PARAMeter:MODE	Turns ON/OFF the most frequent value computation when using histogram or queries the current setting.	4-23
:CALCulation:PARAMeter:PTOPeak	Turns ON/OFF the P-P value computation when using histogram or queries the current setting.	4-23
:CALCulation:PARAMeter:SDEviation	Turns ON/OFF the standard deviation (σ) computation when using histogram or queries the current setting.	4-23
:CALCulation:PARAMeter:TAVerage	Turns ON/OFF the average computation when using time variation or queries the current setting.	4-23
:CALCulation:PARAMeter:TFLutter	Turns ON/OFF the σ /AVE (flutter) computation when using time variation or queries the current setting.	4-23
:CALCulation:PARAMeter:TJITter	Turns ON/OFF the P-P/AVE (jitter) computation when using time variation or queries the current setting.	4-23
:CALCulation:PARAMeter:TMAXimum	Turns ON/OFF the maximum value computation when using time variation or queries the current setting.	4-23
:CALCulation:PARAMeter:TMINimum	Turns ON/OFF the minimum value computation when using time variation or queries the current setting.	4-23
:CALCulation:PARAMeter:TPTOpeak	Turns ON/OFF the P-P value computation when using time variation or queries the current setting.	4-24
:CALCulation:PARAMeter:TRF	Turns ON/OFF the RF value computation when using time variation or queries the current setting.	4-24
:CALCulation:PARAMeter:TSDeviation	Turns ON/OFF the standard deviation (σ) computation when using time variation or queries the current setting.	4-24
:CALCulation:POLarity	Sets the polarity to be analyzed when measuring both pulse widths or both edges or queries the current setting.	4-24
COMMunicate Group		
:COMMunicate?	Queries all settings related to communications.	4-25
:COMMunicate:HEADer	Sets whether to attach a header to the response data or queries the current setting.	4-25
:COMMunicate:LOCKout	Sets or clears local lockout.	4-25
:COMMunicate:REMote	Switches between remote and local.	4-25
:COMMunicate:VERBose	Sets whether to use abbreviated or unabbreviated form for response data.	4-25
:COMMunicate:WAIT	Waits for an extended event to occur.	4-26
:COMMunicate:WAIT?	Creates a response for the specified extended event.	4-26
DISPlay Group		
:DISPlay?	Queries all settings related to the display.	4-28
:DISPlay:BGRaph	Turns ON/OFF the BOTH Graph display when measuring both pulse widths or both edges or queries the current setting.	4-28
:DISPlay:BLOCK	Sets the displayed block or queries the current setting.	4-28
:DISPlay:DOTConnect	Turns ON/OFF dot connect on the time variation display or queries the current setting.	4-28
:DISPlay:DOTType	Sets the display method of the measured point on the time variation display or queries the current setting.	4-28
:DISPlay:GRAPhsize	Sets the size of the graph display or queries the current setting.	4-29
:DISPlay:GRID	Turns ON/OFF the grid display on the time variation display or queries the current setting.	4-29
:DISPlay:ITEM	Sets the display format or queries the current setting.	4-29
:DISPlay:OVERlap	Sets whether to display waveforms overlapped (ON/OFF) or queries the current setting.	4-29
:DISPlay:PANorama[:STATe]	Turns ON/OFF the panorama display or queries the current setting.	4-29
:DISPlay:SGRaph	Sets the display at the bottom section of the screen when using ALL display on the multi window or auto window or queries the current setting.	4-29
:DISPlay:SITem	Sets the type of statistical values to be displayed on the statistical display or queries the current setting.	4-30

Command	Function	Page
:DISPlay:SSTyle	Sets the display format on the statistical display or queries the current setting.	4-30
:DISPlay:STATistic	Turns ON/OFF the statistical display when using histogram or time variation display or queries the current setting.	4-30
:DISPlay:WINDow	Sets the window to be displayed or queries the current setting.	4-30
FILE Group		
:FILE?	Queries all settings related to files.	4-31
:FILE:CDIRectory	Changes the current directory.	4-31
:FILE:DELeTe:{BINary BMP POSTscript SETup STATistic TIFF TEXT}	Deletes various types of files.	4-32
:FILE:DRIVE	Sets the target drive.	4-32
:FILE:FORMat	Executes the floppy disk format.	4-32
:FILE:FREE?	Queries the free disk space in bytes.	4-32
:FILE:LOAD:{BINary SETup}	Recalls various types of files.	4-32
:FILE:MDIRectory	Creates a directory.	4-32
:FILE:PATH?	Queries the current directory.	4-32
:FILE:SAVE?	Queries all settings related to file saving.	4-32
:FILE:SAVE:ANAMing	Turns ON/OFF the auto naming function of saved file names or queries the current setting.	4-32
:FILE:SAVE:{BINary SETup STATistic TEXT}	Saves various types of files.	4-33
:FILE:SAVE:COMMeNt	Sets the comment at the top left section of the screen or queries the current setting.	4-33
:FILE:SAVE:SITem	Sets the type of statistical data file to be saved or queries the current setting.	4-33
HCOPY Group		
:HCOPY?	Queries all settings related to the output of screen image data.	4-34
:HCOPY:ABORt	Aborts the printout of the screen image.	4-34
:HCOPY:ANAMing	Turns ON/OFF the auto naming function of file names when saving screen images to files or queries the current setting.	4-35
:HCOPY:CDIRectory	Changes the current directory.	4-35
:HCOPY:COMMeNt	Sets the comment at the top left section of the screen or queries the current setting.	4-35
:HCOPY:COMPreSSion	Turns ON/OFF the compression when saving the screen image in BMP format or queries the current setting.	4-35
:HCOPY:DEVIce	Sets the output destination of the screen image or queries the current setting.	4-35
:HCOPY:DRIVE	Sets the target drive.	4-35
:HCOPY[:EXECute]	Executes the printout of the screen image.	4-35
:HCOPY:FILEname	Sets the name of the file for saving the screen image or queries the current setting.	4-35
:HCOPY:FORMat	Sets the format for saving the screen image or queries the current setting.	4-35
:HCOPY:TONE	Sets the color/gradation for saving the screen image or queries the current setting.	4-35
HHistogram and THISTogram Group		
:{HHistogram<x> THISTogram<x>}?	Queries all settings related to the histogram display.	4-38
:{HHistogram<x> THISTogram<x>}:AUTO?	Queries all settings related to the auto window.	4-38
:{HHistogram<x> THISTogram<x>}:AUTO:MODulation	Sets the modulation type on the auto window or queries the current setting.	4-38
:{HHistogram<x> THISTogram<x>}:AUTO:TTYPE	Sets how to determine constant T on the auto window or queries the current setting.	4-38
:{HHistogram<x> THISTogram<x>}:AUTO:WINDow1:STATE	Turns ON/OFF window 1 on the auto window or queries the current setting.	4-38
:{HHistogram<x> THISTogram<x>}:MARKer?	Queries all settings related to the marker.	4-38
:{HHistogram<x> THISTogram<x>}:MARKer:LOW	Sets the low marker value or queries the current setting.	4-38
:{HHistogram<x> THISTogram<x>}:MARKer[:STATE]	Turns ON/OFF the marker or queries the current setting.	4-39
:{HHistogram<x> THISTogram<x>}[:MODE]	Sets the window mode or queries the current setting.	4-39
:{HHistogram<x> THISTogram<x>}:MULTi?	Queries all settings related to the multi window.	4-39
:{HHistogram<x> THISTogram<x>}:MULTi:FREQuency	Sets constant T using the frequency format or queries the current setting.	4-39

4.1 List of Commands

Command	Function	Page
: {HHISTogram<x> THISTogram<x>} :MULTi:OFFSet	Sets the offset value on constant T or queries the current setting.	4-39
: {HHISTogram<x> THISTogram<x>} :MULTi:SIZE	Sets the number of windows or queries the current setting.	4-39
: {HHISTogram<x> THISTogram<x>} :MULTi:TVALue	Sets the constant T value or queries the current setting.	4-40
: {HHISTogram<x> THISTogram<x>} :MULTi:UPDate	Changes the window setting based on the constant T and offset values.	4-40
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>?	Queries all settings related to the specified window.	4-40
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:HORizontal?	Queries all settings related to the horizontal axis (X-axis) of the specified window.	4-40
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:HORizontal:CENTer	Sets the center position of the horizontal axis (X-axis) of the specified window.	4-40
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:HORizontal:SPAN	Sets the span of the horizontal axis (X-axis) of the specified window.	4-41
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:LABel	Sets the label of the specified window or queries the current setting.	4-41
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:MARKer?	Queries all settings related to the horizontal axis (X-axis) marker of the specified window.	4-41
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:MARKer: {LEFT RIGHT}	Sets the marker position of the specified window or queries the current setting.	4-41
: {HHISTogram<x> THISTogram<x>} :MULTi:WINDow<x>:MARKer: {LVALue? RVALue?}	Sets the frequency of the marker position of the specified window or queries the current setting.	4-42
: {HHISTogram<x> THISTogram<x>} :SINGle?	Queries all settings related to the single window.	4-42
: {HHISTogram<x> THISTogram<x>} :SINGle:FREQuency	Sets constant T using the frequency format or queries the current setting.	4-42
: {HHISTogram<x> THISTogram<x>} :SINGle:HORizontal?	Queries all settings related to the horizontal axis (X-axis) of the single window.	4-42
: {HHISTogram<x> THISTogram<x>} :SINGle:HORizontal:CENTer	Sets the center position of the horizontal axis (X-axis) of the single window.	4-42
: {HHISTogram<x> THISTogram<x>} :SINGle:HORizontal:SPAN	Sets the span of the horizontal axis (X-axis) of the single window.	4-43
: {HHISTogram<x> THISTogram<x>} :SINGle:MARKer?	Queries all settings related to the marker of the single window.	4-43
: {HHISTogram<x> THISTogram<x>} :SINGle:MARKer: {LEFT RIGHT}	Sets the marker position of the single window or queries the current setting.	4-43
: {HHISTogram<x> THISTogram<x>} :SINGle:MARKer: {LVALue? RVALue?}	Sets the frequency of the marker position of the single window or queries the current setting.	4-43
: {HHISTogram<x> THISTogram<x>} :SINGle:TVALue	Sets the constant T value of the single window or queries the current setting.	4-43
: {HHISTogram<x> THISTogram<x>} :VERTical?	Queries all settings related to the vertical axis (Y-axis).	4-43
: {HHISTogram<x> THISTogram<x>} :VERTical:AXIS	Queries the scale type setting of the vertical axis (Y-axis).	4-44
: {HHISTogram<x> THISTogram<x>} :VERTical:HIGh	Queries the upper limit of the vertical axis (Y-axis) scale.	4-44
IHISTogram Group		
: IHISTogram?	Queries all settings related to the histogram display for ISI mode.	4-47
: IHISTogram:AUTO?	Queries all settings related to the auto window.	4-47
: IHISTogram:AUTO:MODulation	Sets the modulation type on the auto window or queries the current setting.	4-47
: IHISTogram:AUTO:TTYPe	Sets how to determine constant T on the auto window or queries the current setting.	4-47
: IHISTogram:AUTO:WINDow1:STATe	Turns ON/OFF window 1 on the auto window or queries the current setting.	4-47
: IHISTogram: {MARK<x> SPACe<x>}	Sets mark/space or queries the current setting.	4-48
: IHISTogram[:MODE]	Sets the window mode or queries the current setting.	4-48
: IHISTogram:MULTi?	Queries all settings related to the multi window.	4-48

Command	Function	Page
:IHISTogram:MULTi:FREQuency	Sets constant T using the frequency format or queries the current setting.	4-48
:IHISTogram:MULTi:OFFSet	Sets the offset value on constant T or queries the current setting.	4-49
:IHISTogram:MULTi:SIZE	Sets the number of windows or queries the current setting.	4-49
:IHISTogram:MULTi:TVALue	Sets the constant T value or queries the current setting.	4-49
:IHISTogram:MULTi:UPDate	Changes the window setting based on the constant T and offset values.	4-49
:IHISTogram:MULTi:WINDow<x>?	Queries all settings related to the specified window.	4-49
:IHISTogram:MULTi:WINDow<x>:HORizontal?	Queries all settings related to the horizontal axis (X-axis) of the specified window.	4-49
:IHISTogram:MULTi:WINDow<x>:HORizontal:CENTer	Sets the center position of the horizontal axis (X-axis) of the specified window.	4-49
:IHISTogram:MULTi:WINDow<x>:HORizontal:SPAN	Sets the span of the horizontal axis (X-axis) of the specified window.	4-49
:IHISTogram:MULTi:WINDow<x>:LABel	Sets the window label or queries the current setting.	4-50
:IHISTogram:MULTi:WINDow<x>:MARKer?	Queries all settings related to the horizontal axis (X-axis) marker of the specified window.	4-50
:IHISTogram:MULTi:WINDow<x>:MARKer:{LEFT RIGHT}	Sets the marker position of the specified window or queries the current setting.	4-50
:IHISTogram:POLarity	Sets the polarity or queries the current setting.	4-50
:IHISTogram:SYNC	Turns ON/OFF the Sync function or queries the current setting.	4-50
:IHISTogram:TARGet	Sets the analysis target or queries the current setting.	4-50
:IHISTogram:TMODe	Sets the trigger mode or queries the current setting.	4-50
:IHISTogram:TRIGger	Sets the trigger condition or queries the current setting.	4-51
INPut Group		
:INPut?	Queries all settings related to the input.	4-52
:INPut:{ACHannel BCHannel}?	Queries all settings related to the specified channel.	4-52
:INPut:{ACHannel BCHannel}:COUPling	Sets the coupling of the specified channel or queries the current setting.	4-53
:INPut:{ACHannel BCHannel}:IMPedance	Sets the input impedance of the specified channel or queries the current setting.	4-53
:INPut:{ACHannel BCHannel}:TRIGger?	Queries all settings related to the trigger of the specified channel.	4-53
:INPut:{ACHannel BCHannel}:TRIGger:LEVel	Sets the trigger level of the specified channel or queries the current setting.	4-53
:INPut:{ACHannel BCHannel}:TRIGger:MODe	Sets the trigger mode of the specified channel or queries the current setting.	4-53
:INPut:AGATe?	Queries all settings related to arming and external gate.	4-53
:INPut:AGATe:LEVel	Sets the arming/gate level or queries the current setting.	4-53
:INPut:BCHannel:PHASe?	Queries all settings related to the CH B phase adjustment.	4-53
:INPut:BCHannel:PHASe:ADJust	Sets the phase adjustment time of CH B or queries the current setting.	4-53
:INPut:INHibit?	Queries all settings related to inhibit.	4-54
:INPut:INHibit:LEVel	Sets the inhibit level or queries the current setting.	4-54
MEASure Group		
:MEASure?	Queries all settings related to the measurement conditions.	4-55
:MEASure:FUNCTion	Sets the measurement function or queries the current setting.	4-55
:MEASure:MODe	Sets the sampling mode or queries the current setting.	4-56
:MEASure:POLarity	Sets the pulse width polarity or queries the current setting.	4-56
:MEASure:SLOPe	Sets the slope of the period/A-to-B time interval or queries the current setting.	4-56
MEMory Group		
:MEMory?	Queries all settings related the external transmission of the measured data.	4-57
:MEMory:BLOCK	Sets the target block for block sampling or queries the current setting.	4-57
:MEMory:BSIZe?	Queries the number of blocks in which the measurements are valid.	4-57
:MEMory:BYTeorder	Sets the transmission order of binary data or queries the current setting.	4-57
:MEMory:CLEar	Clears the measured data.	4-57
:MEMory:DATaselect	Sets the data to be transmitted or queries the current setting.	4-58
:MEMory:END	Sets the data position of transmission end or queries the current setting.	4-58
:MEMory:FORMat	Sets the format of the data to be transmitted or queries the current setting.	4-58
:MEMory:SEND<x>?	Executes the transmission of the measured data specified by "MEMory:DATaselect."	4-58
:MEMory:SIZE<x>?	Queries the number of data points that have been measured.	4-59

4.1 List of Commands

Command	Function	Page
:MEMory:START	Sets the data position of transmission start or queries the current setting.	4-59
RECall Group		
:RECall	Recalls the setup data.	4-59
SAMPlE Group		
:SAMPlE?	Queries all settings related to sampling.	4-61
:SAMPlE:ARMinG?	Queries all settings related to arming.	4-61
:SAMPlE:ARMinG:DELAy?	Queries all settings related to arming delay.	4-61
:SAMPlE:ARMinG:DELAy:{AEVentsize BEVentsize EVENTsize}	Sets the arming delay event of the specified channel or queries the current setting.	4-62
:SAMPlE:ARMinG:DELAy:{ATIME BTIME TIME}	Sets the arming delay time of the specified channel or queries the current setting.	4-62
:SAMPlE:ARMinG:DELAy[:MODE]	Sets the arming delay mode or queries the current setting.	4-62
:SAMPlE:ARMinG:SLOPe	Sets the arming slope or queries the current setting.	4-62
:SAMPlE:ARMinG:SOURCe	Sets the arming source or queries the current setting.	4-62
:SAMPlE:BLOCK?	Queries all settings related to block sampling.	4-62
:SAMPlE:BLOCK:REST?	Queries all settings related to block sampling rest.	4-62
:SAMPlE:BLOCK:REST:EVENT	Sets the rest time of block sampling in terms of the number of events or queries the current setting.	4-63
:SAMPlE:BLOCK:REST[:MODE]	Sets the block sampling rest mode or queries the current setting.	4-63
:SAMPlE:BLOCK:REST:TIME	Sets the block sampling rest time or queries the current setting.	4-63
:SAMPlE:BLOCK:SIZE	Sets the number of blocks of block sampling or queries the current setting.	4-63
:SAMPlE:BLOCK[:STATe]	Turns ON/OFF block sampling or queries the current setting.	4-63
:SAMPlE:GATE?	Queries all settings related to the gate.	4-63
:SAMPlE:GATE:EVENTsize	Sets the gate in terms of the number of events or queries the current setting.	4-63
:SAMPlE:GATE[:MODE]	Sets the gate type or queries the current setting.	4-64
:SAMPlE:GATE:POLarity	Sets the polarity of the external gate or queries the current setting.	4-64
:SAMPlE:GATE:TIME	Sets the gate time in terms of time or queries the current setting.	4-64
:SAMPlE:INHibit	Sets the polarity of inhibit input or queries the current setting.	4-64
:SAMPlE:INTerval	Sets the sampling interval or queries the current setting.	4-64
:SAMPlE:RCLock	Sets reference sampling clock or queries the current setting.	4-64
SStArt Group		
:SStArt	Executes single measurement.	4-64
StARt Group		
:StARt	Starts the measurement.	4-64
STATus Group		
:STATus?	Queries all settings related to the communication status.	4-65
:STATus:CONDition?	Queries the contents of the condition register.	4-65
:STATus:EESe	Sets the extended event enable register or queries the current setting.	4-65
:STATus:EESR?	Queries the content of the extended event register and clears the register.	4-65
:STATus:ERRor?	Queries the error code and message information.	4-65
:STATus:FILTer<x>	Sets the transition filter or queries the current setting.	4-65
:STATus:QMESsage	Sets whether or not to attach message information to the response to the “:STATus:ERRor?” query or queries the current setting.	4-65
StOP Group		
:StOP	Stops the measurement.	4-66
StORe Group		
:StORe	Stores the setup data.	4-66
SYSTem Group		
:SYSTem?	Queries all settings related to the SYSTem group.	4-67
:SYSTem:BEEP	Turns ON/OFF the beep sound or queries the current setting.	4-67
:SYSTem:CLICksound	Turns ON/OFF the click sound or queries the current setting.	4-68
:SYSTem:DATE	Sets the date or queries the current setting.	4-68
:SYSTem:LCD?	Queries all settings related to the LCD.	4-68
:SYSTem:LCD:AOff	Turns ON/OFF the auto off function of the LCD or queries the current setting.	4-68
:SYSTem:LCD:ATIME	Sets the auto off time of the LCD or queries the current setting.	4-68
:SYSTem:LCD:BRIGHtness	Sets the brightness of the LCD or queries the current setting.	4-68
:SYSTem:LCD[:STATe]	Turns ON/OFF the LCD backlight or queries the current setting.	4-68
:SYSTem:TIME	Sets the time or queries the current setting.	4-68
:SYSTem:WARnIng	Turns ON/OFF the warning display or queries the current setting.	4-68

Command	Function	Page
TVARiation Group		
:TVARiation<x>?	Queries all settings related to time variation.	4-69
:TVARiation<x>:HORizonatal?	Queries all settings related to the horizontal axis (X-axis) of the time variation display.	4-69
:TVARiation<x>:HORizonatal:MINimum	Sets the left end of the horizontal axis (X-axis) scale or queries the current setting.	4-70
:TVARiation<x>:HORizonatal:SPAN	Sets the span of the horizontal axis (X-axis) scale or queries the current setting.	4-70
:TVARiation<x>:MARKer?	Queries all settings related to the marker.	4-70
:TVARiation<x>:MARKer:{HIGH LOW}	Sets the position of the high/low marker or queries the current setting.	4-70
:TVARiation<x>:MARKer:{LEFT RIGHT}	Sets the position of the left/right marker or queries the current setting.	4-70
:TVARiation<x>:MARKer:{LVALue? RVALue?}	Queries the measured value at the left/right marker position.	4-70
:TVARiation<x>:MARKer[:STATE]	Turns ON/OFF the marker cursor or queries the current setting.	4-70
:TVARiation<x>:VERTical?	Queries all settings related to the vertical axis (Y-axis).	4-70
:TVARiation<x>:VERTical:CENTER	Queries the center value of the vertical axis (Y-axis).	4-70
:TVARiation<x>:VERTical:SPAN	Queries the span of the vertical axis (Y-axis).	4-71
UNIT Group		
:UNIT?	Queries the default unit of voltage, time and frequency.	4-71
:UNIT:VOLTage	Sets the default unit of voltage or queries the current setting.	4-71
:UNIT:TIME	Sets the default unit of time or queries the current setting.	4-71
:UNIT:FREQuency	Sets the default unit of frequency or queries the current setting.	4-71
Common Command Group		
*CAL?	Performs calibration and queries the result.	4-72
*CLS	Clears the standard event register, extended event register, and error queue.	4-72
*ESE	Sets the standard event enable register or queries the current setting.	4-72
*ESR?	Queries the standard event register and clears the register.	4-72
*IDN?	Queries the instrument model.	4-72
*OPC	Sets the bit 0 (OPC bit) of the standard event register to 1 upon the completion of the specified overlap command.	4-72
*OPC?	Creates a response upon the completion of the specified overlap command.	4-72
*RST	Resets the settings.	4-73
*SRE	Sets the service request enable register or queries the current setting.	4-73
*STB?	Queries the status byte register.	4-73
*TRG?	Executes single measurement.	4-73
*TST?	Performs a self-test and queries the result.	4-73
*WAI	Holds the subsequent command until the completion of the specified overlap operation.	4-73

4.2 AScale Group

The commands in this group deal with auto scale. You can execute the same operation as the AUTO SCALE key on the front panel.

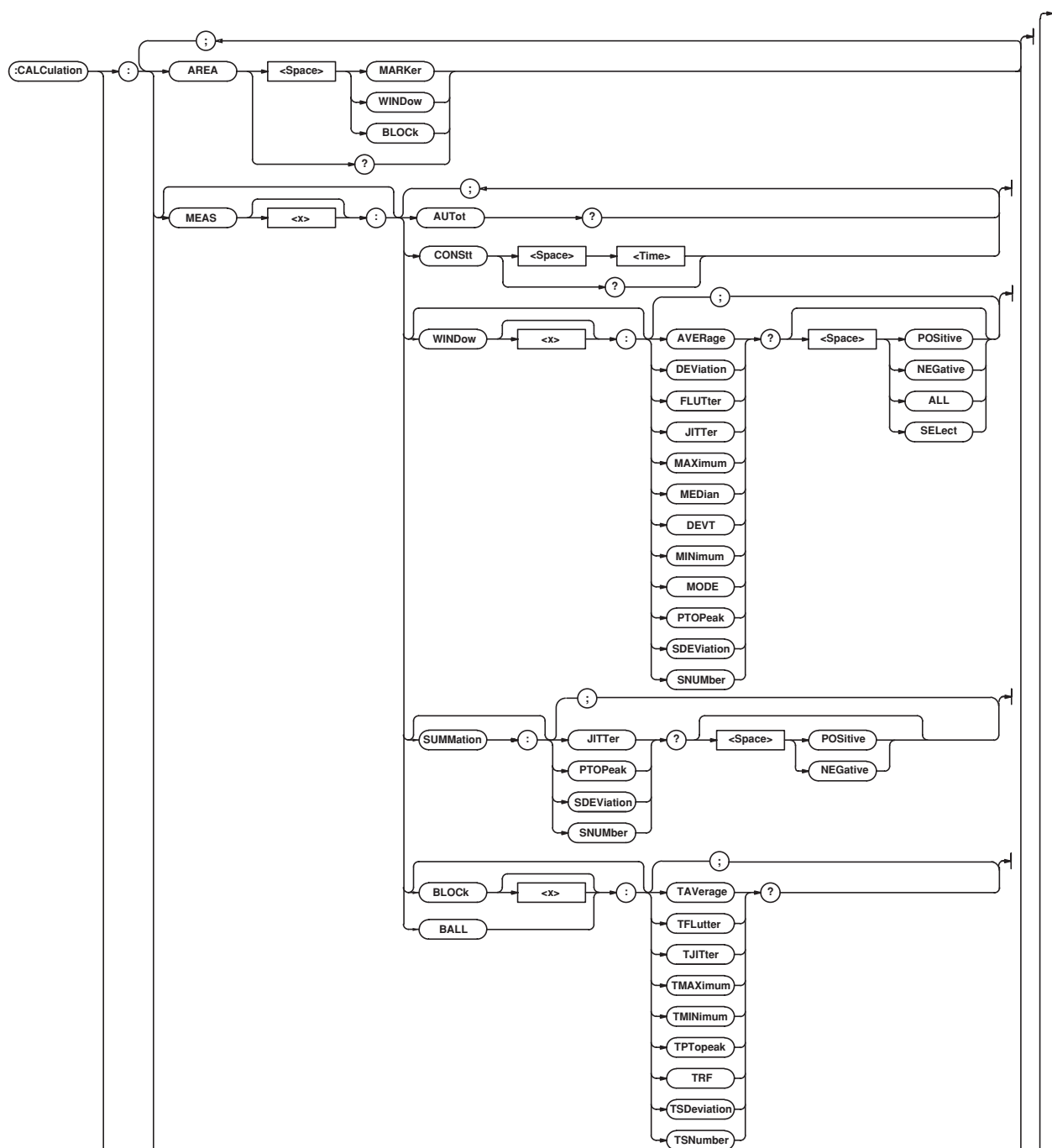


:AScale

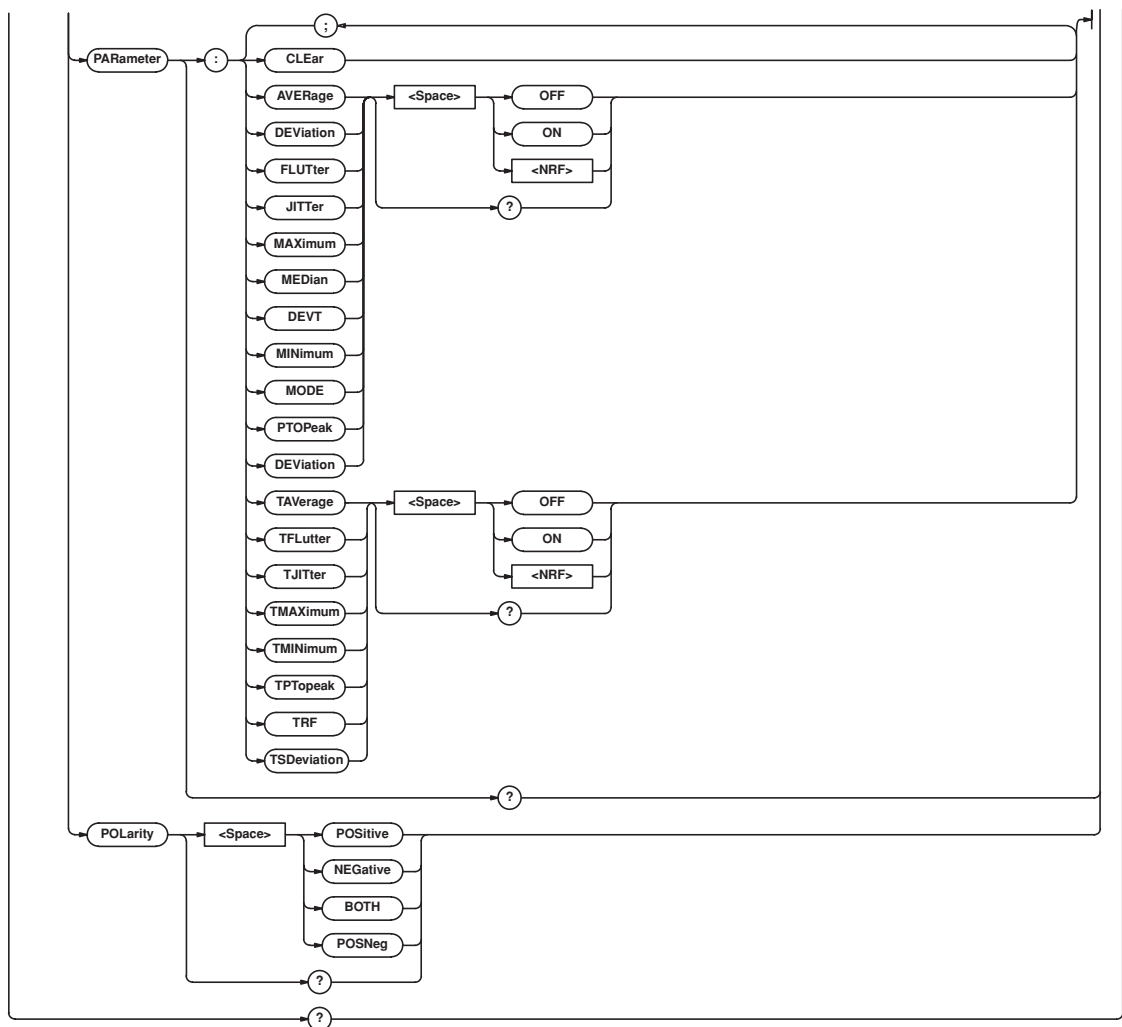
Function	Executes auto scaling.
Syntax	<code>:AScale</code>
Example	<code>:ASCALE</code>
Description	Auto scale cannot be executed when using multi window or auto window.

4.3 CALCulation Group

The commands in this group deal with statistical computation. You can make the same settings, make inquiries of settings, and make inquiries of the statistical values as when the Calc/Stat menu of the DISPLAY key on the front panel is used.



4.3 CALCulation Group



:CALCulation

Function Queries all settings related to the statistical value.

Syntax :CALCulation?

Example :CALCULATION? -> :CALCULATION:
AREA BLOCK;PARAMETER:AVERAGE 1;
DEVIATION 0;FLUTTER 0;JITTER 0;
MAXIMUM 0;MEDIAN 0;DEVT 0;
MINIMUM 0;MODE 0;PTOPEAK 1;
SDEVIATION 1;TAVERAGE 1;TFLUTTER 0;
TJITTER 0;TMAXIMUM 0;TMINIMUM 0;
TPTOPEAK 1;TRF 1;TSDEVIATION 1

:CALCulation:AREA

Function Queries the computation range of the statistical values or queries the current setting.

Syntax :CALCulation:AREA
{MARKer|WINDow|BLOCK}
:CALCulation:AREA?

Example :CALCULATION:AREA MARKER
:CALCULATION:AREA? -> :CALCULATION:
AREA MARKER

Description

- Cannot be specified when using multi window or auto window.
- BLOCK cannot be specified when using histogram.
- BLOCK cannot be specified when measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B.

:CALCulation[:MEAS<x>]:AUTot?

Function	Queries the value defined by measured T/ estimated T of the auto window.
Syntax	:CALCulation[:MEAS<x>]:AUTot? <x>=1 to 2
Example	- When measuring period, A-to-B time interval, or pulse width :CALCULATION:AUTOT? -> :CALCULATION:AUTOT 3.6750E-8 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:AUTOT? -> :CALCULATION:MEAS1: AUTOT 3.6750E-8
Description	- When measuring period, A-to-B time interval, or pulse width, omit “:MEAS<x>”. - Query is possible when auto window is enabled.

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TAverage?

Function	Queries the average value when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TAverage? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TAVERAGE? -> :CALCULATION: TAVERAGE 2.4035E-9 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TAVERAGE? -> :CALCULATION:BLOCK1: TAVERAGE 2.4035E-9 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TAVERAGE? -> :CALCULATION:MEAS1: TAVERAGE 2.4035E-9
Description	- If the statistical value is not valid, “NAN” is returned in response to a query. - When the computation range is a block and block sampling is used, specify “:{BLOCK<x> BALL}”. - When measuring period, A-to-B time interval, or pulse width, omit “:MEAS<x>”.

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TFLutter?

Function	Queries the α /AVE value (flutter) when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TFLutter? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TFLUTTER? -> :CALCULATION: TFLUTTER 1.2887E+1 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TFLUTTER? -> :CALCULATION:BLOCK1: TFLUTTER 1.2887E+1 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TFLUTTER? -> :CALCULATION:MEAS1: TFLUTTER 1.2887E+1
Description	- If the statistical value is not valid, “NAN” is returned in response to a query. - When the computation range is a block and block sampling is used, specify “:{BLOCK<x> BALL}”. - When measuring period, A-to-B time interval, or pulse width, omit “:MEAS<x>”.

4.3 CALCulation Group

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TJITter?

Function	Queries the P-P/AVE value (jitter) when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TJITter? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TJITTER? -> :CALCULATION:TJITTER 1.382E+00 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TJITTER? -> :CALCULATION:BLOCK1:TJITTER 1.382E+00 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TJITTER? -> :CALCULATION:MEAS1:TJITTER 1.382E+00
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When the computation range is a block and block sampling is used, specify ":{BLOCK<x> BALL}". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TMAXimum?

Function	Queries the maximum value when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TMAXimum? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TMAXIMUM? -> :CALCULATION:TMAXIMUM 1.1287E-7 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TMAXIMUM? -> :CALCULATION:BLOCK1:TMAXIMUM 1.1287E-7 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TMAXIMUM? -> :CALCULATION:MEAS1:TMAXIMUM 1.1287E-7
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When the computation range is a block and block sampling is used, specify ":{BLOCK<x> BALL}". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TMINimum?

Function	Queries the minimum value when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TMINimum? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TMINIMUM? -> :CALCULATION: TMINIMUM 9.99437E-8 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TMINIMUM? -> :CALCULATION:BLOCK1: TMINIMUM 9.99437E-8 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TMINIMUM? -> :CALCULATION:MEAS1: TMINIMUM 9.99437E-8
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When the computation range is a block and block sampling is used, specify "{BLOCK<x> BALL}". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TPTOpeak?

Function	Queries the P-P value when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TPTOpeak? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TPTOPEAK? -> :CALCULATION: TPTOPEAK 3.0245E-9 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TPTOPEAK? -> :CALCULATION:BLOCK1: TPTOPEAK 3.0245E-9 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TPTOPEAK? -> :CALCULATION:MEAS1: TPTOPEAK 3.0245E-9
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When the computation range is a block and block sampling is used, specify "{BLOCK<x> BALL}". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

4.3 CALCulation Group

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TRF?

Function Queries the RF value when using time variation.

Syntax :CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TRF?

<x> of MEAS<x>=1 to 2

<x> of BLOCK<x>=Block number (1 to number of blocks)

BALL: Queries the statistical value of the entire block.

- Example
- When measuring period, A-to-B time interval, or pulse width and block sampling is not used
:CALCULATION:TRF?
-> :CALCULATION:TRF 1.4775E-1
 - When measuring period, A-to-B time interval, or pulse width and block sampling is used
:CALCULATION:BLOCK1:TRF?
-> :CALCULATION:BLOCK1:TRF 1.4775E-1
 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B
:CALCULATION:MEAS1:TRF?
-> :CALCULATION:MEAS1:TRF 1.4775E-1

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When the computation range is a block and block sampling is used, specify "{BLOCK<x>|BALL}".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TSDeviation?

Function Queries the standard deviation (σ) when using time variation.

Syntax :CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TSDeviation?

<x> of MEAS<x>=1 to 2

<x> of BLOCK<x>=Block number (1 to number of blocks)

BALL: Queries the statistical value of the entire block.

- Example
- When measuring period, A-to-B time interval, or pulse width and block sampling is not used
:CALCULATION:TSDEVIATION?
-> :CALCULATION:TSDEVIATION 2.4035E-9
 - When measuring period, A-to-B time interval, or pulse width and block sampling is used
:CALCULATION:BLOCK1:TSDEVIATION?
-> :CALCULATION:BLOCK1:TSDEVIATION 2.4035E-9
 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B
:CALCULATION:MEAS1:TSDEVIATION?
-> :CALCULATION:MEAS1:TSDEVIATION 2.4035E-9
- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When the computation range is a block and block sampling is used, specify "{BLOCK<x>|BALL}".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

:CALCulation[:MEAS<x>][:{BLOCK<x>|BALL}] :TSNumber?

Function	Queries the number of samples for the statistical computation when using time variation.
Syntax	:CALCulation[:MEAS<x>][:{BLOCK<x> BALL}] :TSNumber? <x> of MEAS<x>=1 to 2 <x> of BLOCK<x>=Block number (1 to number of blocks) BALL: Queries the statistical value of the entire block.
Example	- When measuring period, A-to-B time interval, or pulse width and block sampling is not used :CALCULATION:TSNUMBER? -> :CALCULATION:TSNUMBER 1.000E+3 - When measuring period, A-to-B time interval, or pulse width and block sampling is used :CALCULATION:BLOCK1:TSNUMBER? -> :CALCULATION:BLOCK1:TSNUMBER 1.000E+3 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:TSNUMBER? -> :CALCULATION:MEAS1:TSNUMBER 1.000E+3
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When the computation range is a block and block sampling is used, specify "{BLOCK<x> BALL}". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

:CALCulation[:MEAS<x>]:CONSTt

Function	Sets constant T or queries the current setting.
Syntax	:CALCulation[:MEAS<x>]:CONSTt {<Time>} :CALCulation[:MEAS<x>]:CONSTt? <x>=1 to 2 <Time>=1 ns to 250 ns (25 ps steps)
Example	- When measuring period, A-to-B time interval, or pulse width :CALCULATION:CONSTT 100ns :CALCULATION:CONSTT? -> :CALCULATION:CONSTT 100.000E-9 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B :CALCULATION:MEAS1:CONSTT 100ns :CALCULATION:MEAS1:CONSTT? -> :CALCULATION:MEAS1:CONSTT 100.000E-9
Description	- Setting and query are possible when single window or multi window is enabled. - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".

4.3 CALCulation Group

:CALCulation[:MEAS<x>][:WINDow<x>]:

AVERage?

Function	Queries the average value when using histogram.
Syntax	:CALCulation[:MEAS<x>][:WINDow<x>]: AVERage? [{POSitive NEGative ALL SELEct}] <x> of MEAS<x>=1 to 2 <x> of WINDow<x>=Window number (1 to number of windows)
Example	When in time stamp/hardware histogram mode - When measuring period, A-to-B time interval, or pulse width and single window is used :CALCulation:AVERAGE? -> :CALCulation: AVERAGE 1.00035E-7 - When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used :CALCulation:WINDOW1:AVERAGE? -> :CALCulation:WINDOW1: AVERAGE 1.00035E-7 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used :CALCulation:MEAS1:AVERAGE? -> :CALCulation:MEAS1: AVERAGE 1.00035E-7 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used :CALCulation:MEAS1:WINDOW1:AVERAGE? -> :CALCulation:MEAS1:WINDOW1: AVERAGE 1.00035E-7 When in inter-symbol interference analysis mode :CALCulation:WINDOW1:AVERAGE? ALL -> :CALCulation:WINDOW1: AVERAGE 1.00035E-7
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When using single window, omit ":WINDow<x>". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>". - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive NEGative". - When in inter-symbol interference analysis mode, specify "ALL SELEct".

:CALCulation[:MEAS<x>][:WINDow<x>]:

DEVIation?

Function	Queries the deviation when using histogram.
Syntax	:CALCulation[:MEAS<x>][:WINDow<x>]: DEVIation? [{POSitive NEGative ALL SELEct}] <x> of MEAS<x>=1 to 2 <x> of WINDow<x>=Window number (1 to number of windows)
Example	When in time stamp/hardware histogram mode - When measuring period, A-to-B time interval, or pulse width and single window is used :CALCulation:DEVIATION? -> :CALCulation: DEVIATION 1.4425E+1 - When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used :CALCulation:WINDOW1:DEVIATION? -> :CALCulation:WINDOW1: DEVIATION 1.4425E+1 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used :CALCulation:MEAS1:DEVIATION? -> :CALCulation:MEAS1: DEVIATION 1.4425E+1 - When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used :CALCulation:MEAS1:WINDOW1:DEVIATION? -> :CALCulation:MEAS1:WINDOW1: DEVIATION 1.4425E+1 When in inter-symbol interference analysis mode :CALCulation:WINDOW1:DEVIATION? ALL -> :CALCulation:WINDOW1: DEVIATION 1.4425E+1
Description	- If the statistical value is not valid, "NAN" is returned in response to a query. - When using single window, omit ":WINDow<x>". - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>". - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive NEGative". - When in inter-symbol interference analysis mode, specify "ALL SELEct".

:CALCulation[:MEAS<x>][:WINDOW<x>]:DEVT?

Function Queries the deviation/T value when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDOW<x>]:
DEVT? [{POSitive|NEGative|ALL|
SElect}]
<x> of MEAS<x>=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)

Example **When in time stamp/hardware histogram mode**

- When measuring period, A-to-B time interval, or pulse width and single window is used
:CALCulation:DEVT?
-> :CALCulation:DEVT 9.99437E-8
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
:CALCulation:WINDOW1:DEVT?
-> :CALCulation:WINDOW1:
DEVT 9.99437E-8
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
:CALCulation:MEAS1:DEVT?
-> :CALCulation:MEAS1:
DEVT 9.99437E-8
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
:CALCulation:MEAS1:WINDOW1:DEVT?
-> :CALCulation:MEAS1:WINDOW1:
DEVT 9.99437E-8

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:DEVT? ALL
-> :CALCulation:WINDOW1:
DEVT 9.99437E-8
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":WINDOW<x>".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

:CALCulation[:MEAS<x>][:WINDOW<x>]:**FLUTter?**

Function Queries the σ /AVE value (flutter) when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDOW<x>]:
FLUTter? [{POSitive|NEGative|ALL|
SElect}]
<x> of MEAS<x>=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)

Example **When in time stamp/hardware histogram mode**

- When measuring period, A-to-B time interval, or pulse width and single window is used
:CALCulation:FLUTTER?
-> :CALCulation:FLUTTER 1.2887E+1
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
:CALCulation:WINDOW1:FLUTTER?
-> :CALCulation:WINDOW1:
FLUTTER 1.2887E+1
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
:CALCulation:MEAS1:FLUTTER?
-> :CALCulation:MEAS1:
FLUTTER 1.2887E+1
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
:CALCulation:MEAS1:WINDOW1:
FLUTTER?
-> :CALCulation:MEAS1:WINDOW1:
FLUTTER 1.2887E+1

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:FLUTTER? ALL
-> :CALCulation:WINDOW1:
FLUTTER 1.2887E+1
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":WINDOW<x>".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

4.3 CALCulation Group

:CALCulation[:MEAS<x>][:WINDow<x>|SUMMation]:JITTER?

Function Queries the σ/T value (jitter) when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDow<x>|SUMMation]:JITTER? [{POSitive|NEGative|ALL|SElect}]
 <x> of MEAS<x>=1 to 2
 <x> of WINDow<x>=Window number (1 to number of windows)
 SUMMation: Queries the statistical value when all windows are accumulated.

Example **When in time stamp/hardware histogram mode**

- When measuring period, A-to-B time interval, or pulse width and single window is used
 :CALCulation:JITTER?
 -> :CALCulation:JITTER 1.2366E+1
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
 :CALCulation:WINDOW1:JITTER?
 -> :CALCulation:WINDOW1:JITTER 1.2366E+1
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
 :CALCulation:MEAS1:JITTER?
 -> :CALCulation:MEAS1:JITTER 1.2366E+1
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
 :CALCulation:MEAS1:WINDOW1:JITTER?
 -> :CALCulation:MEAS1:WINDOW1:JITTER 1.2366E+1

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:JITTER? ALL
-> :CALCulation:WINDOW1:JITTER 1.2366E+1
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":{WINDow<x>|SUMMation}".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

:CALCulation[:MEAS<x>][:WINDow<x>]:MAXimum?

Function Queries the maximum value when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDow<x>]:MAXimum? [{POSitive|NEGative|ALL|SElect}]
 <x> of MEAS<x>=1 to 2
 <x> of WINDow<x>=Window number (1 to number of windows)

Example **When in time stamp/hardware histogram mode**

- When measuring period, A-to-B time interval, or pulse width and single window is used
 :CALCulation:MAXIMUM?
 -> :CALCulation:MAXIMUM 1.1287E-7
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
 :CALCulation:WINDOW1:MAXIMUM?
 -> :CALCulation:WINDOW1:MAXIMUM 1.1287E-7
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
 :CALCulation:MEAS1:MAXIMUM?
 -> :CALCulation:MEAS1:MAXIMUM 1.1287E-7
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
 :CALCulation:MEAS1:WINDOW1:MAXIMUM?
 -> :CALCulation:MEAS1:WINDOW1:MAXIMUM 1.1287E-7

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:MAXIMUM? ALL
-> :CALCulation:WINDOW1:MAXIMUM 1.1287E-7
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":WINDow<x>".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

:CALCulation[:MEAS<x>][:WINDOW<x>]:**MEDian?**

Function Queries the median value when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDOW<x>]:
MEDian? [{POSitive|NEGative|ALL|
SElect}]
<x> of MEAS<x>=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)

Example When in time stamp/hardware histogram mode

- When measuring period, A-to-B time interval, or pulse width and single window is used
:CALCulation:MEDIAN?
-> :CALCulation:MEDIAN 1.00145E-7
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
:CALCulation:WINDOW1:MEDIAN?
-> :CALCulation:WINDOW1:
MEDIAN 1.00145E-7
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
:CALCulation:MEAS1:MEDIAN?
-> :CALCulation:MEAS1:
MEDIAN 1.00145E-7
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
:CALCulation:MEAS1:WINDOW1:
MEDIAN?
-> :CALCulation:MEAS1:WINDOW1:
MEDIAN 1.00145E-7

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:MEDIAN? ALL
-> :CALCulation:WINDOW1:
MEDIAN 1.00145E-7
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":WINDOW<x>".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

:CALCulation[:MEAS<x>][:WINDOW<x>]:**MINimum?**

Function Queries the minimum value when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDOW<x>]:
MINimum? [{POSitive|NEGative|ALL|
SElect}]
<x> of MEAS<x>=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)

Example When in time stamp/hardware histogram mode

- When measuring period, A-to-B time interval, or pulse width and single window is used
:CALCulation:MINIMUM?
-> :CALCulation:
MINIMUM 9.99437E-8
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
:CALCulation:WINDOW1:MINIMUM?
-> :CALCulation:WINDOW1:
MINIMUM 9.99437E-8
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
:CALCulation:MEAS1:MINIMUM?
-> :CALCulation:MEAS1:
MINIMUM 9.99437E-8
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
:CALCulation:MEAS1:WINDOW1:
MINIMUM?
-> :CALCulation:MEAS1:WINDOW1:
MINIMUM 9.99437E-8

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:MINIMUM? ALL
-> :CALCulation:WINDOW1:
MINIMUM 9.99437E-8
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":WINDOW<x>".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

4.3 CALCulation Group

:CALCulation[:MEAS<x>][:WINDOW<x>]:MODE?

Function Queries the most frequent value when using histogram.

Syntax :CALCulation[:MEAS<x>][:WINDOW<x>]:MODE? [{POSitive|NEGative|ALL|SELEct}]
 <x> of MEAS<x>=1 to 2
 <x> of WINDOW<x>=Window number (1 to number of windows)

Example When in time stamp/hardware histogram mode

- When measuring period, A-to-B time interval, or pulse width and single window is used
 :CALCulation:MODE?
 -> :CALCulation:MODE 1.00025E-7
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
 :CALCulation:WINDOW1:MODE?
 -> :CALCulation:WINDOW1:MODE 1.00025E-7
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
 :CALCulation:MEAS1:MODE?
 -> :CALCulation:MEAS1:MODE 1.00025E-7
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
 :CALCulation:MEAS1:WINDOW1:MODE?
 -> :CALCulation:MEAS1:WINDOW1:MODE 1.00025E-7

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:MODE? ALL
-> :CALCulation:WINDOW1:MODE 1.00025E-7
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":WINDOW<x>".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SELEct".

:CALCulation[:MEAS<x>][:{WINDOW<x>|SUMMation}]:PTOPeak?

Function Queries the P-P value when using histogram.

Syntax :CALCulation[:MEAS<x>][:{WINDOW<x>|SUMMation}]:PTOPeak? [{POSitive|NEGative|ALL|SELEct}]
 <x> of MEAS<x>=1 to 2
 <x> of WINDOW<x>=Window number (1 to number of windows)
 SUMMation: Queries the statistical value when all windows are accumulated.

Example When in time stamp/hardware histogram mode

- When measuring period, A-to-B time interval, or pulse width and single window is used
 :CALCulation:PTOPeak?
 -> :CALCulation:PTOPeak 3.0245E-9
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
 :CALCulation:WINDOW1:PTOPeak?
 -> :CALCulation:WINDOW1:PTOPeak 3.0245E-9
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
 :CALCulation:MEAS1:PTOPeak?
 -> :CALCulation:MEAS1:PTOPeak 3.0245E-9
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
 :CALCulation:MEAS1:WINDOW1:PTOPeak?
 -> :CALCulation:MEAS1:WINDOW1:PTOPeak 3.0245E-9

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:PTOPeak? ALL
-> :CALCulation:WINDOW1:PTOPeak 3.0245E-9
```

- Description
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit ":{WINDOW<x>|SUMMation}".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SELEct".

:CALCulation[:MEAS<x>][:{WINDOW<x>|SUMMation}]:SDEVIation?

Function Queries the standard deviation (σ) when using histogram.

Syntax :CALCulation[:MEAS<x>][:{WINDOW<x>|SUMMation}]:SDEVIation? [{POSitive|NEGative|ALL|SElect}]
 <x> of MEAS<x>=1 to 2
 <x> of WINDOW<x>=Window number (1 to number of windows)
 SUMMation: Queries the statistical value when all windows are accumulated.

Example **When in time stamp/hardware histogram mode**

- When measuring period, A-to-B time interval, or pulse width and single window is used
 :CALCulation:SDEVIation?
 -> :CALCulation:
 SDEVIation 2.4035E-9
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
 :CALCulation:WINDOW1:SDEVIation?
 -> :CALCulation:WINDOW1:
 SDEVIation 2.4035E-9
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
 :CALCulation:MEAS1:SDEVIation?
 -> :CALCulation:MEAS1:
 SDEVIation 2.4035E-9
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
 :CALCulation:MEAS1:WINDOW1:
 SDEVIation?
 -> :CALCulation:MEAS1:WINDOW1:
 SDEVIation 2.4035E-9

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:SDEVIation?
ALL
-> :CALCulation:WINDOW1:
SDEVIation 2.4035E-9
```

- Description**
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit "{WINDOW<x>|SUMMation}".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

:CALCulation[:MEAS<x>][:{WINDOW<x>|SUMMation}]:SNUMBER?

Function Queries the number of samples for the statistical computation when using histogram.

Syntax :CALCulation[:MEAS<x>][:{WINDOW<x>|SUMMation}]:SNUMBER? [{POSitive|NEGative|ALL|SElect}]
 <x> of MEAS<x>=1 to 2
 <x> of WINDOW<x>=Window number (1 to number of windows)
 SUMMation: Queries the statistical value when all windows are accumulated.

Example **When in time stamp/hardware histogram mode**

- When measuring period, A-to-B time interval, or pulse width and single window is used
 :CALCulation:SNUMBER?
 -> :CALCulation:SNUMBER 6.000E+3
- When measuring period, A-to-B time interval, or pulse width and multi window or auto window is used
 :CALCulation:WINDOW1:SNUMBER?
 -> :CALCulation:WINDOW1:
 SNUMBER 6.000E+3
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and single window is used
 :CALCulation:MEAS1:SNUMBER?
 -> :CALCulation:MEAS1:
 SNUMBER 6.000E+3
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B and multi window or auto window is used
 :CALCulation:MEAS1:WINDOW1:
 SNUMBER?
 -> :CALCulation:MEAS1:WINDOW1:
 SNUMBER 6.000E+3

When in inter-symbol interference analysis mode

```
:CALCulation:WINDOW1:SNUMBER? ALL
-> :CALCulation:WINDOW1:
SNUMBER 6.000E+3
```

- Description**
- If the statistical value is not valid, "NAN" is returned in response to a query.
 - When using single window, omit "{WINDOW<x>|SUMMation}".
 - When measuring period, A-to-B time interval, or pulse width, omit ":MEAS<x>".
 - When ":CALCulation:POLarity" is set to "POSNeg", specify "POSitive|NEGative".
 - When in inter-symbol interference analysis mode, specify "ALL|SElect".

4.3 CALCulation Group

:CALCulation:PARAmeter?

Function Queries whether all the statistical computations are turn ON or OFF.

Syntax :CALCulation:PARAmeter?

Example :CALCULATION:PARAMETER?
-> :CALCULATION:PARAMETER:
AVERAGE 1;DEVIATION 1;DEVT 1;
FLUTTER 1;JITTER 1;MAXIMUM 1;
MEDIAN 1;MINIMUM 1;MODE 1;
PTOPEAK 1;SDEVIATION 1

:CALCulation:PARAmeter:CLEAr

Function Turns OFF all the statistical computation values.

Syntax :CALCulation:PARAmeter:CLEAr

Example :CALCULATION:PARAMETER:CLEAR

:CALCulation:PARAmeter:AVERAge

Function Turns ON/OFF the deviation computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:AVERAge
{<Boolean>}

Example :CALCulation:PARAmeter:AVERAge?
:CALCULATION:PARAMETER:AVERAGE ON
:CALCULATION:PARAMETER:AVERAGE?
-> :CALCULATION:PARAMETER:AVERAGE 1

:CALCulation:PARAmeter:DEVIation

Function Turns ON/OFF the deviation computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:DEVIation
{<Boolean>}

Example :CALCulation:PARAmeter:DEVIation?
:CALCULATION:PARAMETER:
DEVIATION ON
:CALCULATION:PARAMETER:DEVIATION?
-> :CALCULATION:PARAMETER:
DEVIATION 1

:CALCulation:PARAmeter:DEVT

Function Turns ON/OFF the deviation/T computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:DEVT
{<Boolean>}

Example :CALCulation:PARAmeter:DEVT?
:CALCULATION:PARAMETER:DEVT ON
:CALCULATION:PARAMETER:DEVT?
-> :CALCULATION:PARAMETER:DEVT 1

:CALCulation:PARAmeter:FLUTter

Function Turns ON/OFF the σ /AVE (flutter) computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:FLUTter
{<Boolean>}

Example :CALCulation:PARAmeter:FLUTter?
:CALCULATION:PARAMETER:FLUTTER ON
:CALCULATION:PARAMETER:FLUTTER?
-> :CALCULATION:PARAMETER:FLUTTER 1

:CALCulation:PARAmeter:JITter

Function Turns ON/OFF the σ /T (jitter) computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:JITter
{<Boolean>}

Example :CALCulation:PARAmeter:JITter?
:CALCULATION:PARAMETER:JITTER ON
:CALCULATION:PARAMETER:JITTER?
-> :CALCULATION:PARAMETER:JITTER 1

:CALCulation:PARAmeter:MAXimum

Function Turns ON/OFF the maximum value computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:MAXimum
{<Boolean>}

Example :CALCulation:PARAmeter:MAXimum?
:CALCULATION:PARAMETER:MAXIMUM ON
:CALCULATION:PARAMETER:MAXIMUM?
-> :CALCULATION:PARAMETER:MAXIMUM 1

:CALCulation:PARAmeter:MEDian

Function Turns ON/OFF the median value computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:MEDian
{<Boolean>}

Example :CALCulation:PARAmeter:MEDian?
:CALCULATION:PARAMETER:MEDIAN ON
:CALCULATION:PARAMETER:MEDIAN?
-> :CALCULATION:PARAMETER:MEDIAN 1

:CALCulation:PARAmeter:MINimum

Function Turns ON/OFF the minimum value computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:MINimum
{<Boolean>}

Example :CALCulation:PARAmeter:MINimum?
:CALCULATION:PARAMETER:MINIMUM ON
:CALCULATION:PARAMETER:MINIMUM?
-> :CALCULATION:PARAMETER:MINIMUM 1

:CALCulation:PARAmeter:MODE

Function Turns ON/OFF the most frequent value computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:MODE
{<Boolean>}

Example :CALCULATION:PARAMETER:MODE ON
:CALCULATION:PARAMETER:MODE?
-> :CALCULATION:PARAMETER:MODE 1

:CALCulation:PARAmeter:PTOPeak

Function Turns ON/OFF the P-P value computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:PTOPeak
{<Boolean>}

Example :CALCULATION:PARAMETER:PTOPEAK ON
:CALCULATION:PARAMETER:PTOPEAK?
-> :CALCULATION:PARAMETER:PTOPEAK 1

:CALCulation:PARAmeter:SDEVIation

Function Turns ON/OFF the standard deviation (σ) computation when using histogram or queries the current setting.

Syntax :CALCulation:PARAmeter:SDEVIation
{<Boolean>}

Example :CALCULATION:PARAMETER:
SDEVIATION ON
:CALCULATION:PARAMETER:SDEVIATION?
-> :CALCULATION:PARAMETER:
SDEVIATION 1

:CALCulation:PARAmeter:TAVerage

Function Turns ON/OFF the average computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TAVerage
{<Boolean>}

Example :CALCULATION:PARAMETER:TAVERAGE ON
:CALCULATION:PARAMETER:TAVERAGE?
-> :CALCULATION:PARAMETER:
TAVERAGE 1

:CALCulation:PARAmeter:TFLutter

Function Turns ON/OFF the σ /AVE (flutter) computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TFLutter
{<Boolean>}

Example :CALCULATION:PARAMETER:TFLUTTER ON
:CALCULATION:PARAMETER:TFLUTTER?
-> :CALCULATION:PARAMETER:
TFLUTTER 1

:CALCulation:PARAmeter:TJITter

Function Turns ON/OFF the P-P/AVE (jitter) computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TJITter
{<Boolean>}

Example :CALCULATION:PARAMETER:TJITTER ON
:CALCULATION:PARAMETER:TJITTER?
-> :CALCULATION:PARAMETER:TJITTER 1

:CALCulation:PARAmeter:TMAXimum

Function Turns ON/OFF the maximum value computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TMAXimum
{<Boolean>}

Example :CALCULATION:PARAMETER:TMAXIMUM ON
:CALCULATION:PARAMETER:TMAXIMUM?
-> :CALCULATION:PARAMETER:
TMAXIMUM 1

:CALCulation:PARAmeter:TMINimum

Function Turns ON/OFF the minimum value computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TMINimum
{<Boolean>}

Example :CALCULATION:PARAMETER:TMINIMUM ON
:CALCULATION:PARAMETER:TMINIMUM?
-> :CALCULATION:PARAMETER:
TMINIMUM 1

4.3 CALCulation Group

:CALCulation:PARAmeter:TPTopeak

Function Turns ON/OFF the P-P value computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TPTopeak
{<Boolean>}

:CALCulation:PARAmeter:TPTopeak?

Example :CALCULATION:PARAMETER:TPTOPEAK ON
:CALCULATION:PARAMETER:TPTOPEAK?
-> :CALCULATION:PARAMETER:
TPTOPEAK 1

:CALCulation:PARAmeter:TRF

Function Turns ON/OFF the RF value computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TRF
{<Boolean>}

:CALCulation:PARAmeter:TRF?

Example :CALCULATION:PARAMETER:TRF ON
:CALCULATION:PARAMETER:TRF?
-> :CALCULATION:PARAMETER:TRF 1

:CALCulation:PARAmeter:TSDeviation

Function Turns ON/OFF the standard deviation (σ) computation when using time variation or queries the current setting.

Syntax :CALCulation:PARAmeter:TSDeviation
{<Boolean>}

:CALCulation:PARAmeter:TSDeviation?

Example :CALCULATION:PARAMETER:
TSDEVIATION ON
:CALCULATION:PARAMETER:TSDEVIATION?
-> :CALCULATION:PARAMETER:
TSDEVIATION 1

:CALCulation:POLarity

Function Sets the polarity to be analyzed when measuring both pulse widths or both edges or queries the current setting.

Syntax :CALCulation:POLarity {POSitive|
NEGative|BOTH|POSNeg}
:CALCulation:POLarity?

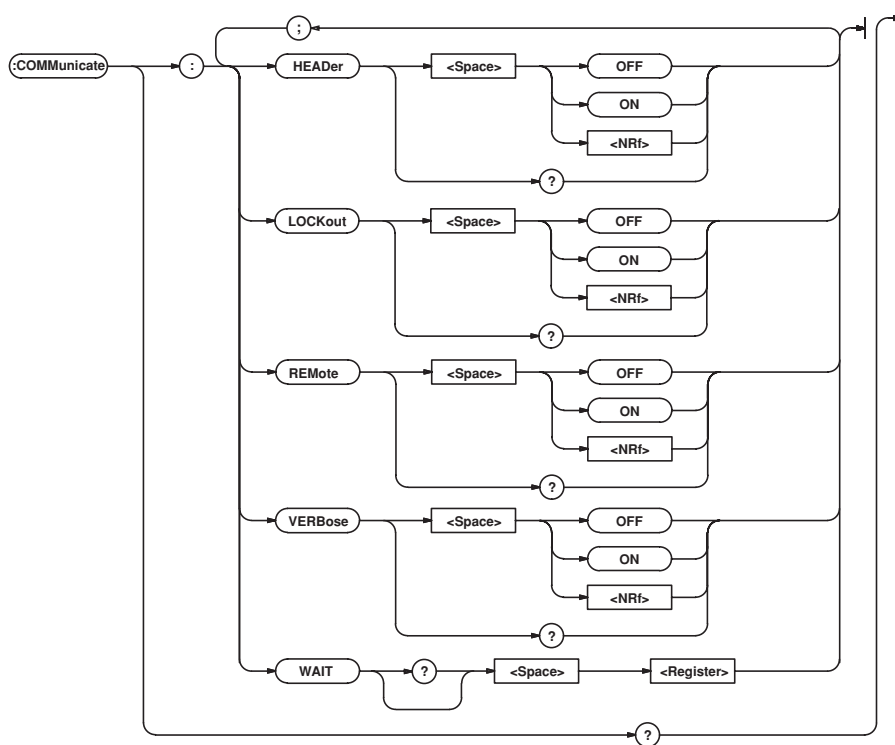
Example :CALCULATION:POLARITY POSITIVE
:CALCULATION:POLARITY?
-> :CALCULATION:POLARITY POSITIVE

Description

- When measuring both pulse widths or both edges, the statistical value corresponding to the polarity specified by this command is returned.
- When "POSNeg" is specified, specify "POSitive|NEGative" in the query inquiring the statistical value.

4.4 COMMunicate Group

The commands in this group deal with communications. There are no front panel keys that correspond to the commands in this group.



:COMMunicate?

Function Queries all settings related to communications.
 Syntax :COMMunicate?
 Example :COMMUNICATE?
 -> :COMMUNICATE:HEADER 1;VERBOSE 1

:COMMunicate:HEAdEr

Function Sets whether to attach a header to the response data or queries the current setting.
 Syntax :COMMunicate:HEAdEr {<Boolean>}
 :COMMunicate:HEAdEr?
 Example :COMMUNICATE:HEADER ON
 :COMMUNICATE:HEADER?
 -> :COMMUNICATE:HEADER 1

:COMMunicate:LOCKout

Function Sets or clears local lockout.
 Syntax :COMMunicate:LOCKout {<Boolean>}
 :COMMunicate:LOCKout?
 Example :COMMUNICATE:LOCKOUT ON
 :COMMUNICATE:LOCKOUT?
 -> :COMMUNICATE:LOCKOUT 1
 Description This command is dedicated to the Ethernet communication interface.

:COMMunicate:REMote

Function Sets remote or local. ON is remote mode.
 Syntax :COMMunicate:REMote {<Boolean>}
 :COMMunicate:REMote?
 Example :COMMUNICATE:REMOTE ON
 :COMMUNICATE:REMOTE?
 -> :COMMUNICATE:REMOTE 1
 Description This command is dedicated to the Ethernet communication interface.

:COMMunicate:VERBose

Function Sets the response messages to full form or abbreviated form or queries the current setting.
 Syntax :COMMunicate:VERBose {<Boolean>}
 :COMMunicate:VERBose?
 Example :COMMUNICATE:VERBOSE ON
 :COMMUNICATE:VERBOSE?
 -> :COMMUNICATE:HEADER ON

4.4 COMMunicate Group

:COMMunicate:WAIT

Function Waits for one of the specified extended events to occur.

Syntax :COMMunicate:WAIT {<Register>}
 <Register>=0 to 65535 (extended event register, see page 5-4)

Example :COMMUNICATE:WAIT 65535

Description For the description regarding how to synchronize the program using COMMunicate:WAIT, see page 3-7.

:COMMunicate:WAIT?

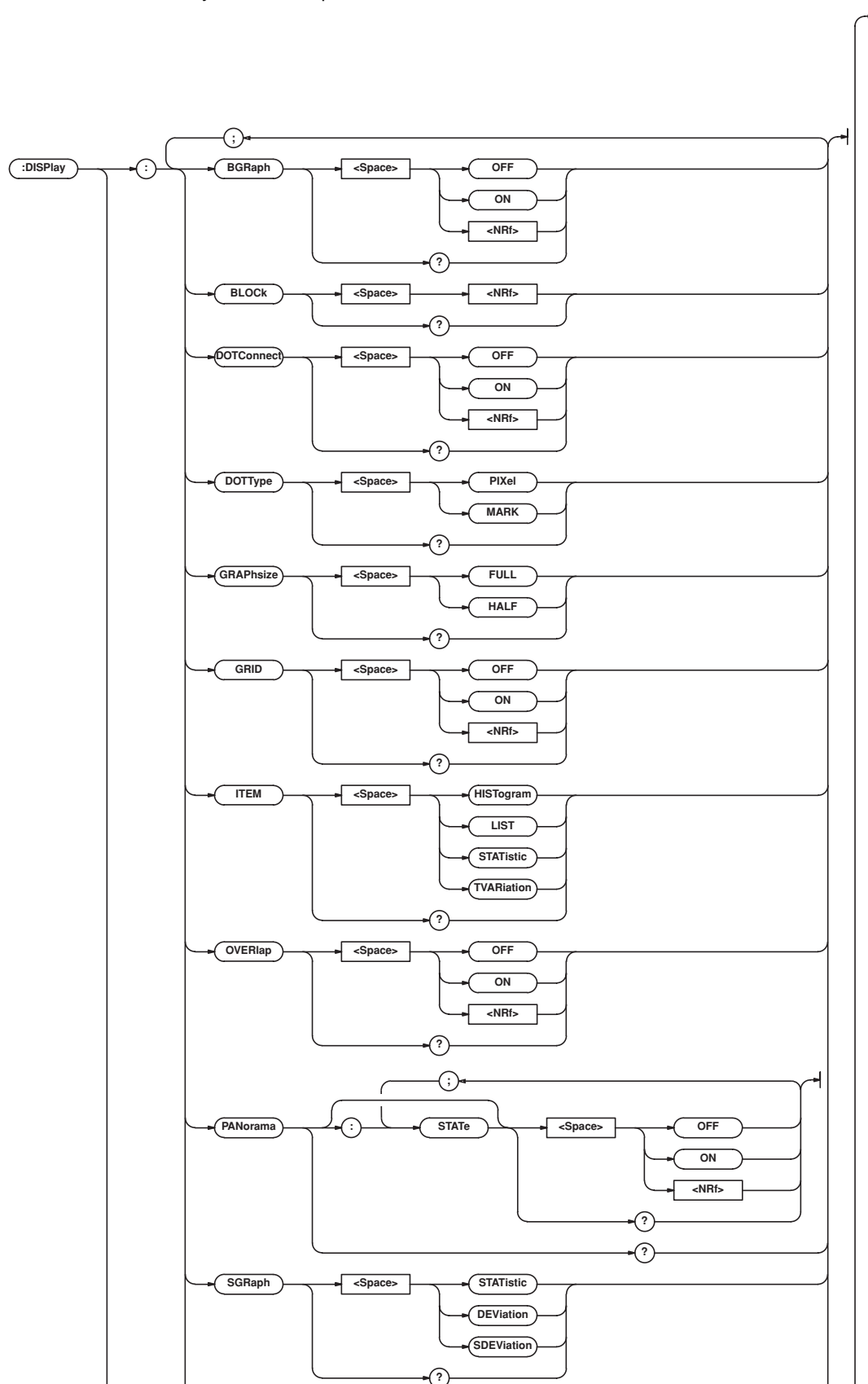
Function Creates the response that is returned when the specified event occurs.

Syntax :COMMunicate:WAIT? {<Register>}
 <Register>=0 to 65535 (extended event register, see page 5-4)

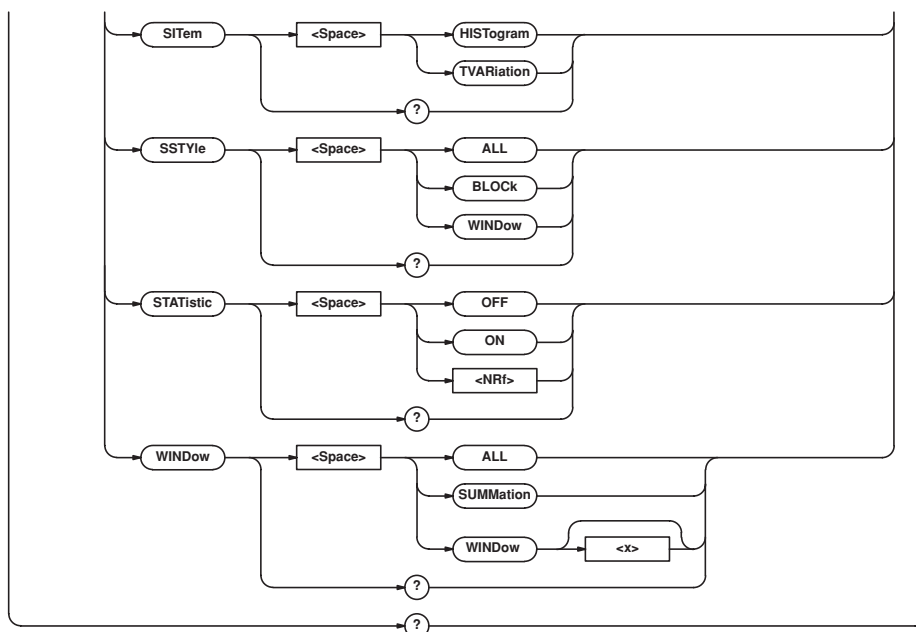
Example :COMMUNICATE:WAIT? 65535 -> 1

4.5 DISPlay Group

The commands in this group deal with the display settings. You can make the same settings and inquiries as when the DISPLAY key on the front panel is used.



4.5 DISPlay Group



:DISPlay?

Function Queries all settings related to the display.

Syntax :DISPlay?

Example :DISPLAY?

```

-> :DISPLAY:GRAPH SIZE FULL;
ITEM HISTOGRAM;PANORAMA:STATE 1;;
DISPLAY:STATISTIC 1
  
```

:DISPlay:BGRaph

Function Turns ON/OFF the BOTH Graph display when measuring both pulse widths or both edges or queries the current setting.

Syntax :DISPlay:BGRaph {<Boolean>}

:DISPLAY:BGRaph?

Example :DISPlay:BGRAPH OFF

:DISPlay:BGRAPH?

-> :DISPLAY:BGRAPH 0

Description

- Setting is possible when displaying histograms, measuring both pulse width or both edges, and
 - “:CALCulation:POLarity” is set to “POSitive|NEGative|POSNeg”.
- For details, see chapter 7 in the *TA720 User's Manual*.

:DISPlay:BLOCK

Function Sets the displayed block or queries the current setting.

Syntax :DISPlay:BLOCK {<Nrf>}

:DISPLAY:BLOCK?

<Nrf>=Block number (0 to number of blocks)

<Nrf>=0 indicates the all blocks.

Example :DISPlay:BLOCK 1

:DISPlay:BLOCK?

-> :DISPLAY:BLOCK 1

Description

- Setting is possible when the sampling mode is set to time stamp and
 - “:CALCulation:AREA” is set to “BLOCK”.
- Setting is possible when
 - “:SAMPle:BLOCK[:STATE]” is set to “ON”.

:DISPlay:DOTConnect

Function Turns ON/OFF dot connect on the time variation display or queries the current setting.

Syntax :DISPlay:DOTConnect {<Boolean>}

:DISPlay:DOTConnect?

Example :DISPLAY:DOTCONNECT ON

:DISPLAY:DOTCONNECT?

-> :DISPLAY:DOTCONNECT 1

:DISPlay:DOTType

Function Sets the display method of the measured point on the time variation display or queries the current setting.

Syntax :DISPlay:DOTType {PIXel|MARK}

:DISPlay:DOTType?

Example :DISPLAY:DOTTYPE PIXEL

:DISPLAY:DOTTYPE?

-> :DISPLAY:DOTTYPE PIXEL

:DISPlay:GRAPHsize

Function Sets the size of the graph display or queries the current setting.

Syntax :DISPlay:GRAPHsize {FULL|HALF}
:DISPlay:GRAPHsize?

Example :DISPLAY:GRAPHSIZE FULL
:DISPLAY:GRAPHSIZE?
-> :DISPLAY:GRAPHSIZE FULL

:DISPlay:GRID

Function Turns ON/OFF the grid display on the time variation display or queries the current setting.

Syntax :DISPlay:GRID {<Boolean>}
:DISPlay:GRID?

Example :DISPLAY:GRID ON
:DISPLAY:GRID?
-> :DISPLAY:GRID 1

:DISPlay:ITEM

Function Sets the display format or queries the current setting.

Syntax :DISPlay:ITEM {HISTogram|LIST|
STATistic|TVARiation}
:DISPlay:ITEM?

Example :DISPLAY:ITEM HISTOGRAM
:DISPLAY:ITEM?
-> :DISPLAY:ITEM HISTOGRAM

Description • "TVARiation" cannot be specified when the sampling mode is set to hardware histogram.
• When the sampling mode is set to inter-symbol interference analysis, only "HISTogram|LIST" can be specified.

:DISPlay:OVERlap

Function Sets whether to overlap the histograms of both polarities when measuring both pulse widths or both edges on the histogram display or queries the current setting.

Sets whether to overlap the waveforms of measurement 1 and measurement 2 on the time variation display or queries the current setting.

Syntax :DISPlay:OVERlap {<Boolean>}
:DISPlay:OVERlap?

Example :DISPLAY:OVERLAP ON
:DISPLAY:OVERLAP?
-> :DISPLAY:OVERLAP 1

Description • Setting is possible when displaying histograms, measuring both pulse width or both edges, and
"CALCulation:POLarity" is set to "POSitive|NEGative|POSNeg".
• Setting is possible when displaying time variation and measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B.
• For details, see chapter 7 in the *TA720 User's Manual*.

:DISPlay:PANorama[:STATE]

Function Turns ON/OFF the panorama display or queries the current setting.

Syntax :DISPlay:PANorama[:STATE]
{<Boolean>}
:DISPlay:PANorama:STATE?

Example :DISPLAY:PANORAMA:STATE ON
:DISPLAY:PANORAMA:STATE?
-> :DISPLAY:PANORAMA:STATE 1

Description Settings is possible when displaying histograms or time variation.

:DISPlay:SGRaph

Function Sets whether to display STATistic, DEViation, or SDEViation at the bottom section of the screen when using ALL display on the multi window or auto window or queries the current setting.

Syntax :DISPlay:SGRaph {STATistic|
DEViation|SDEViation}
:DISPlay:SGRaph?

Example :DISPLAY:SGRAPH DEVIATION
:DISPLAY:SGRAPH?
-> :DISPLAY:SGRAPH DEVIATION

Description Setting is possible when displaying histograms, multi window or auto window is enabled, and
":DISPlay:WINDow" is set to "ALL".

4.5 DISPlay Group

:DISPlay:SITem

Function Sets the type of statistical values to be displayed on the statistical display or queries the current setting.

Syntax :DISPlay:SITem {HISTogram|TVARiation}
:DISPlay:SITem?

Example :DISPLAY:SITEM HISTOGRAM
:DISPLAY:SITEM?
-> :DISPLAY:SITEM HISTOGRAM

Description Setting is possible when the sampling mode is set to time stamp and statistical values are being displayed.

:DISPlay:SSTyle

Function Sets the display format on the statistical display or queries the current setting.

Syntax :DISPlay:SSTyle {ALL|BLOCK|WINDow}
:DISPlay:SSTyle?

Example :DISPLAY:SSTYLE ALL
:DISPLAY:SSTYLE?
-> :DISPLAY:SSTYLE ALL

Description

- "ALL|WINDow" can be specified when the sampling mode is set to hardware histogram and multi window or auto window is enabled.
- "ALL|WINDow" can be specified when the sampling mode is time stamp, multi window or auto window is enabled, and
":DISPlay:SITem" is set to "HISTogram".
- "ALL|BLOCK" can be specified when the sampling mode is set to time stamp,
":SAMPle:BLOCK:STATe" is set to "ON",
":DISPlay:SITem" is set to "TVARiation",
and ":CALCulation:AREA" is set to "BLOCK".

:DISPlay:STATistic

Function Turns ON/OFF the statistical display when using histogram or time variation display or queries the current setting.

Syntax :DISPlay:STATistic {<Boolean>}
:DISPlay:STATistic?

Example :DISPLAY:STATISTIC ON
:DISPLAY:STATISTIC?
-> :DISPLAY:STATISTIC 1

:DISPlay:WINDow

Function Sets the window to be displayed or queries the current setting.

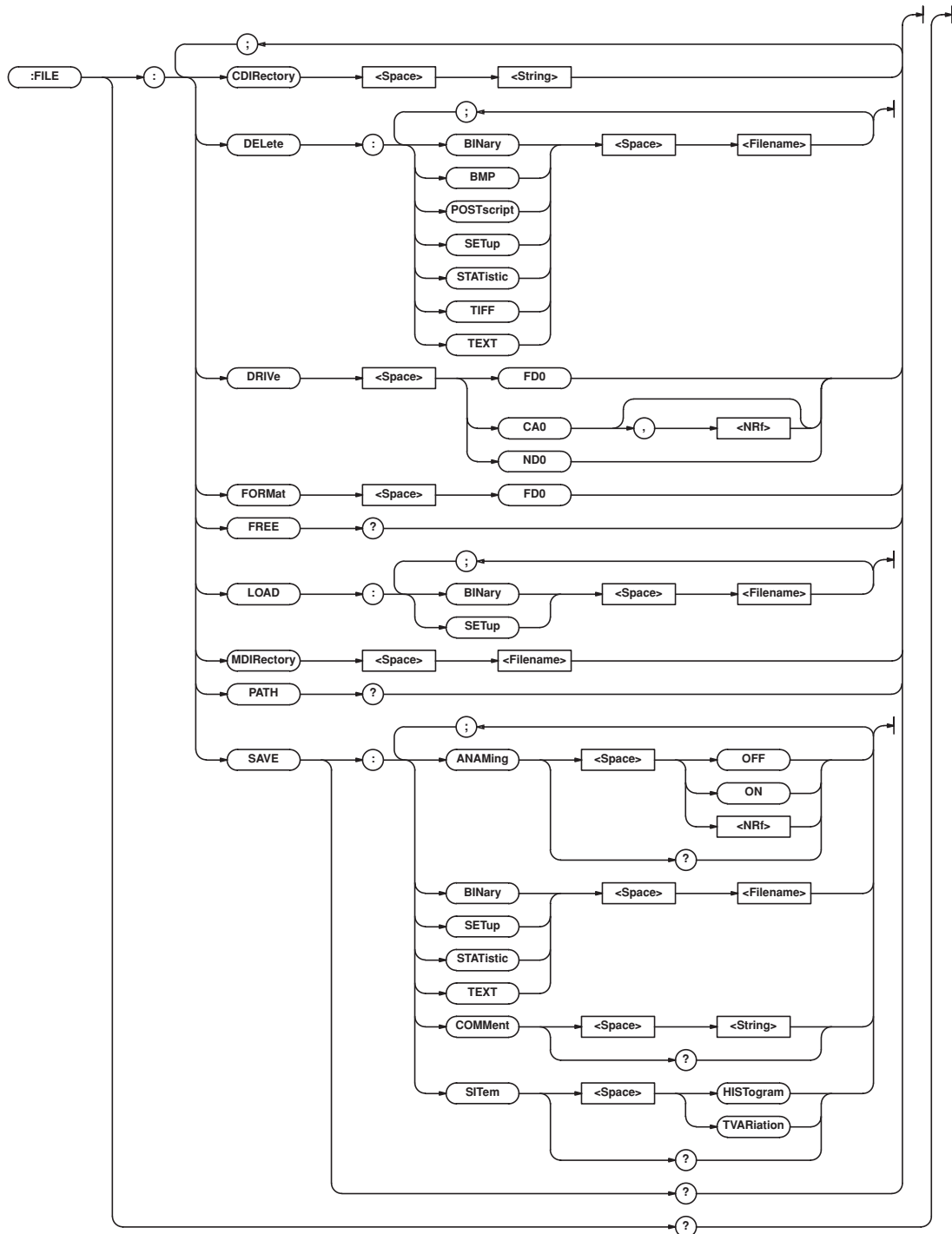
Syntax :DISPlay:WINDow {ALL|SUMMation|WINDow<x>}
:DISPlay:WINDow?
<x>=Window number (1 to number of windows)

Example :DISPLAY:WINDOW ALL
:DISPLAY:WINDOW?
-> :DISPLAY:WINDOW ALL

Description Setting is possible when displaying histograms and multi window or auto window is enabled.

4.6 FILE Group

The commands in this group deal with file operation. You can make the same settings, execute the same operations, and make the same inquiries as when the SHIFT+UTILITY key on the front panel is used.



:FILE?

Function Queries all settings related to files.

Syntax :FILE?

Example :FILE?

```
-> :FILE:SAVE:ANAMING 1;
COMMENT " ";SITEM TVARIATION
```

:FILE:CDIRectory

Function Changes the current directory.

Syntax :FILE:CDIRectory {<Character string>}

Example :FILE:CDIRECTORY "NO_1"

4.6 FILE Group

:FILE:DELeTe:{BINary|BMP|POSTscript|SETup|STATistic|TIFF|TEXT}

Function Deletes various types of files.

Syntax **:FILE:DELeTe:{BINary|BMP|POSTscript|SETup|STATistic|TIFF|TEXT} {<Character string>}**
<Character string>=File name (Up to 8 characters. See the TA720 User's Manual.)
BINary: Measurement data (binary data)
BMP: BMP screen image data
POSTscript: PostScript screen image data
SETup: Setup data
STATistic: Statistical data
TIFF: TIFF screen image data
TEXT: Measurement data (text data)

Example (The following is an example for the setup data.)
:FILE:DELeTe:SETup "TRASH"

Description "**:FILE:DRIVE**" is used to specify the target storage medium for deleting the files.

:FILE:DRIVE

Function Sets the target drive.

Syntax **:FILE:DRIVE {FD0|CA0[,<NRf>]|ND0}**
FD0: Floppy disk drive
CA0: PC card drive
<NRf>=Partition (0 to 3)
ND0: Network drive

Example **:FILE:DRIVE FD0**

Description

- PC card drive and network drive can be specified only on models with these options.
- If the PC card drive does not contain partitions, omit the <NRf> corresponding to partitions.

:FILE:FORMat

Function Executes the floppy disk format.

Syntax **:FILE:FORMat {FD0}**

Example **:FILE:FORMat FD0**

:FILE:FRee?

Function Queries the free disk space (bytes) on the drive.

Syntax **:FILE:FRee?**

Example **:FILE:FRee?**
-> **:FILE:FRee 163840**

Description

- "**:FILE:DRIVE**" is used to specify the target storage medium.
- If the target storage medium is set to network drive, query is not possible.

:FILE:LOAD:{BINary|SETup}

Function Recalls various types of data.

Syntax **:FILE:LOAD:{BINary|SETup} {<Character string>}**
<Character string>=File name (Up to 8 characters. See the TA720 User's Manual.)
BINary: Measurement data (binary data)
SETup: Setup data

Example (The following is an example of a binary measurement data.)
:FILE:LOAD:BINary "TARGET"

:FILE:MDIRectory

Function Creates a directory.

Syntax **:FILE:MDIRectory <Character string>**
<Character string>=File name (Up to 8 characters. See the TA720 User's Manual.)

Example **:FILE:MDIRECTORY "NEW"**

:FILE:PATH?

Function Queries the current directory.

Syntax **:FILE:PATH?**

Example **:FILE:PATH?**
-> **:FILE:PATH "Path=FD0"**

:FILE:SAVe?

Function Queries all settings related to file saving.

Syntax **:FILE:SAVe?**

Example **:FILE:SAVe?**
-> **:FILE:SAVe:ANAMing 0;**
COMMENT "SAMPLE"

:FILE:SAVe:ANAMing

Function Turns ON/OFF the auto naming function of saved file names or queries the current setting.

Syntax **:FILE:SAVe:ANAMing {<Boolean>}**
:FILE:SAVe:ANAMing?

Example **:FILE:SAVe:ANAMing ON**
:FILE:SAVe:ANAMing?
-> **:FILE:SAVe:ANAMing 1**

:FILE:SAVE:{BINary|SETup|STATistic|TEXT}

Function Saves various types of data.

Syntax :FILE:SAVE:{BINary|SETup|STATistic|TEXT} {<Character string>}
 <Character string>=File name (Up to 8 characters. See the TA720 User's Manual.)
 BINary: Measurement data (binary data)
 SETup: Setup data
 STATistic: Statistical data
 TEXT: Measurement data (text data)

Example (The following is an example of a statistical data file.)

```
:FILE:SAVE:STATISTIC "TARGET"
```

Description When the sampling mode is set to time stamp, the statistical data specified by ":FILE:SAVE:SITem" is saved.

:FILE:SAVE:COMment

Function Sets the comment at the top left section of the screen or queries the current setting.

Syntax :FILE:SAVE:COMment {<Character string>}
 :FILE:SAVE:COMment?
 <Character string>=Comment (Up to 25 characters. See the TA720 User's Manual.)

Example :FILE:SAVE:COMment "SAMPLE"
 :FILE:SAVE:COMment?
 -> :FILE:SAVE:COMment "SAMPLE"

:FILE:SAVE:SITem

Function Sets the type of statistical data file to be saved or queries the current setting.

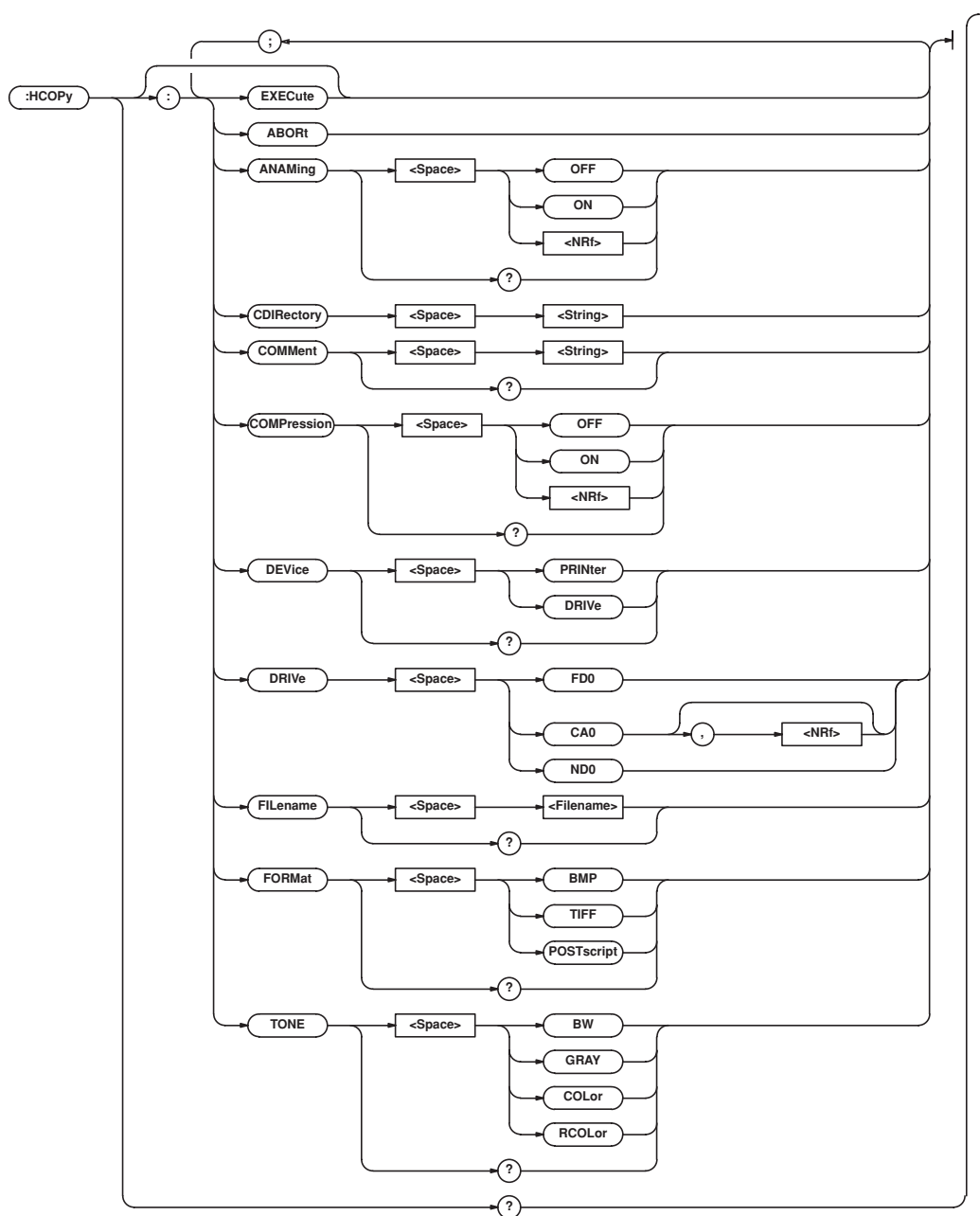
Syntax :FILE:SAVE:SITem {HISTogram|TVARiation}
 :FILE:SAVE:SITem?

Example :FILE:SAVE:SITEM TVARIATION
 :FILE:SAVE:SITEM?
 -> :FILE:SAVE:SITEM TVARIATION

Description Setting is possible when the sampling mode is set to time stamp.

4.7 HCOPy Group

The commands in this group deal with screen image data output.
You can make the same settings, execute the same operations, and make the same inquiries as when the (SHIFT+)COPY key on the front panel is used.



:HCOPY?

Function Queries all settings related to the output of
 screen image data.
Syntax :HCOPY?
Example :HCOPY?
 -> :HCOPY:DEVICE PRINTER;
 COMMENT "SAMPLE"

:HCOPY:ABORt

Function Aborts the printout of the screen image.
Syntax :HCOPY:ABORt
Example :HCOPY:ABORT

:HCOpy:ANAMing

Function Turns ON/OFF the auto naming function of file names when saving screen images to files or queries the current setting.

Syntax :HCOpy:ANAMing {<Boolean>}
:HCOpy:ANAMing?

Example :HCOpy:ANAMING ON
:HCOpy:ANAMING?
-> :HCOpy:ANAMING 1

:HCOpy:COMment

Function Sets the comment at the top left section of the screen or queries the current setting.

Syntax :HCOpy:COMment {<Character string>}
:HCOpy:COMment?
<Character string>=Comment (Up to 25 characters. See the TA720 User's Manual.)

Example :HCOpy:COMment "SAMPLE"
:HCOpy:COMment?
-> :HCOpy:COMment "SAMPLE"

:HCOpy:COMpression

Function Turns ON/OFF the compression when saving the screen image in BMP format or queries the current setting.

Syntax :HCOpy:COMpression {<Boolean>}
:HCOpy:COMpression?

Example :HCOpy:COMPRESSION ON
:HCOpy:COMPRESSION?
-> :HCOpy:COMPRESSION 1

:HCOpy:DEvice

Function Sets the output destination of the screen image or queries the current setting.

Syntax :HCOpy:DEvice {PRINter|DRIVE}
:HCOpy:DEvice?

Example :HCOpy:DEVICE PRINTER
:HCOpy:DEVICE?
-> :HCOpy:DEVICE PRINTER

Description If "DRIVE" is specified, the screen image is output to the drive specified by ":HCOpy:DRIVE".

:HCOpy:DRive

Function Sets the target drive.

Syntax :HCOpy:DRIVE {FD0|CA0[,<NRf>]|ND0}
FD0: Floppy disk drive
CA0: PC card drive
<NRf>=Partition (0 to 3)
ND0: Network drive

Example :HCOpy:DRIVE FD0

Description

- Sets the output destination drive when ":HCOpy:DEVICE" is set to "DRIVE".
- PC card drive and network drive can be specified only on models supporting them.
- If the PC card drive does not contain partitions, omit the <NRf> corresponding to partitions.

:HCOpy:CDIRectory

Function Changes the current directory.

Syntax :HCOpy:CDIRectory {<Character string>}

Example :HCOpy:CDIRECTORY "NO_1"

:HCOpy[:EXECute]

Function Executes the printout of the screen image.

Syntax :HCOpy[:EXECute]

Example :HCOpy:EXECUTE

:HCOpy:FILEname

Function Sets the name of the file for saving the screen image or queries the current setting.

Syntax :HCOpy:FILEname {<Character string>}
:HCOpy:FILEname?
<Character string>=File name (Up to 8 characters. See the TA720 User's Manual.)

Example :HCOpy:FILENAME "KEEP"
:HCOpy:FILENAME?
-> :HCOpy:FILENAME "KEEP"

:HCOpy:FORMat

Function Sets the format for saving the screen image or queries the current setting.

Syntax :HCOpy:FORMat {BMP|TIFF|POSTscript}
:HCOpy:FORMat?

Example :HCOpy:FORMAT BMP
:HCOpy:FORMAT?
-> :HCOpy:FORMAT BMP

:HCOpy:TONE

Function Sets the color/gradation for saving the screen image or queries the current setting.

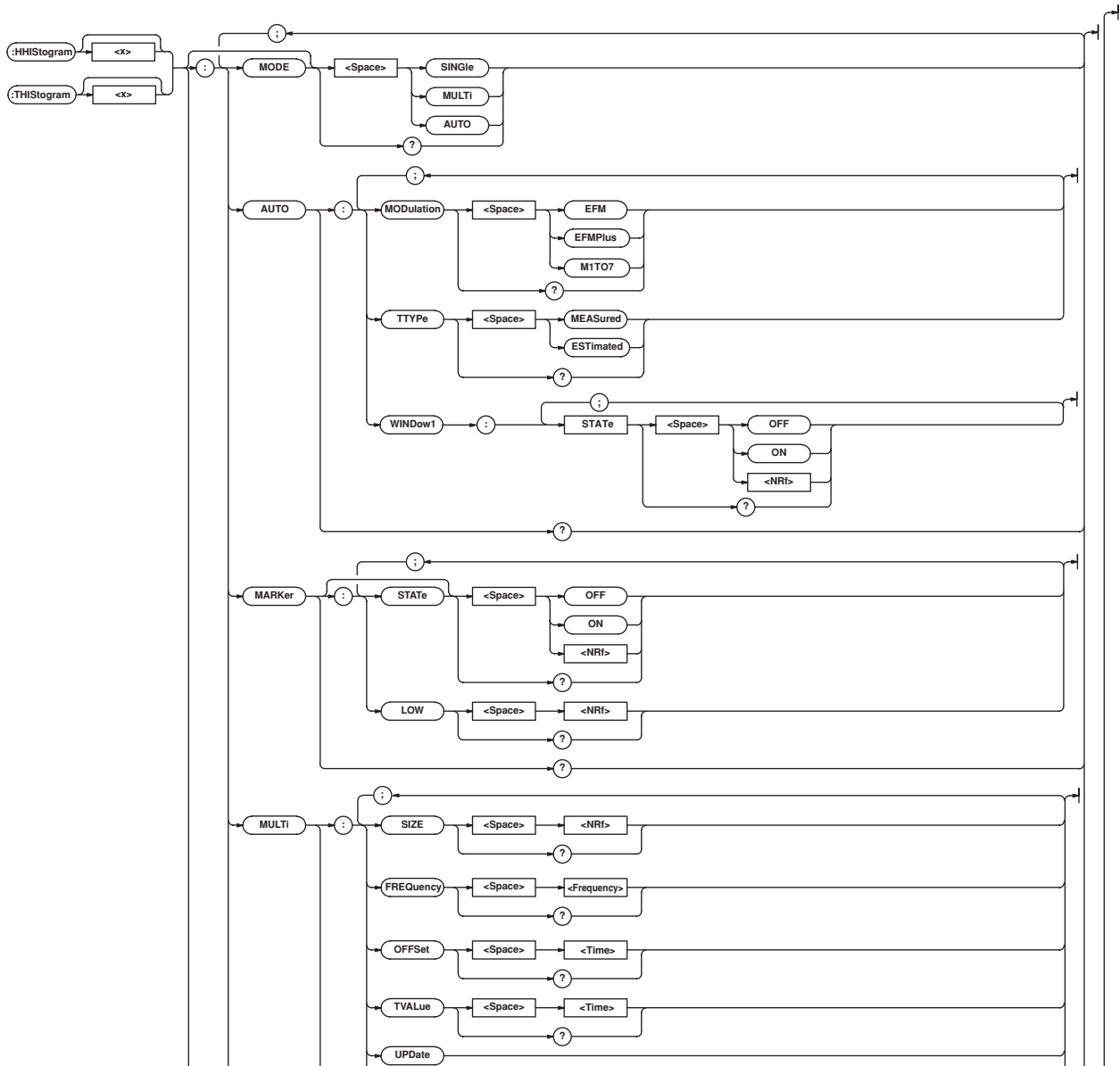
Syntax :HCOpy:TONE {BW|GRAY|COLor|RCOLor}

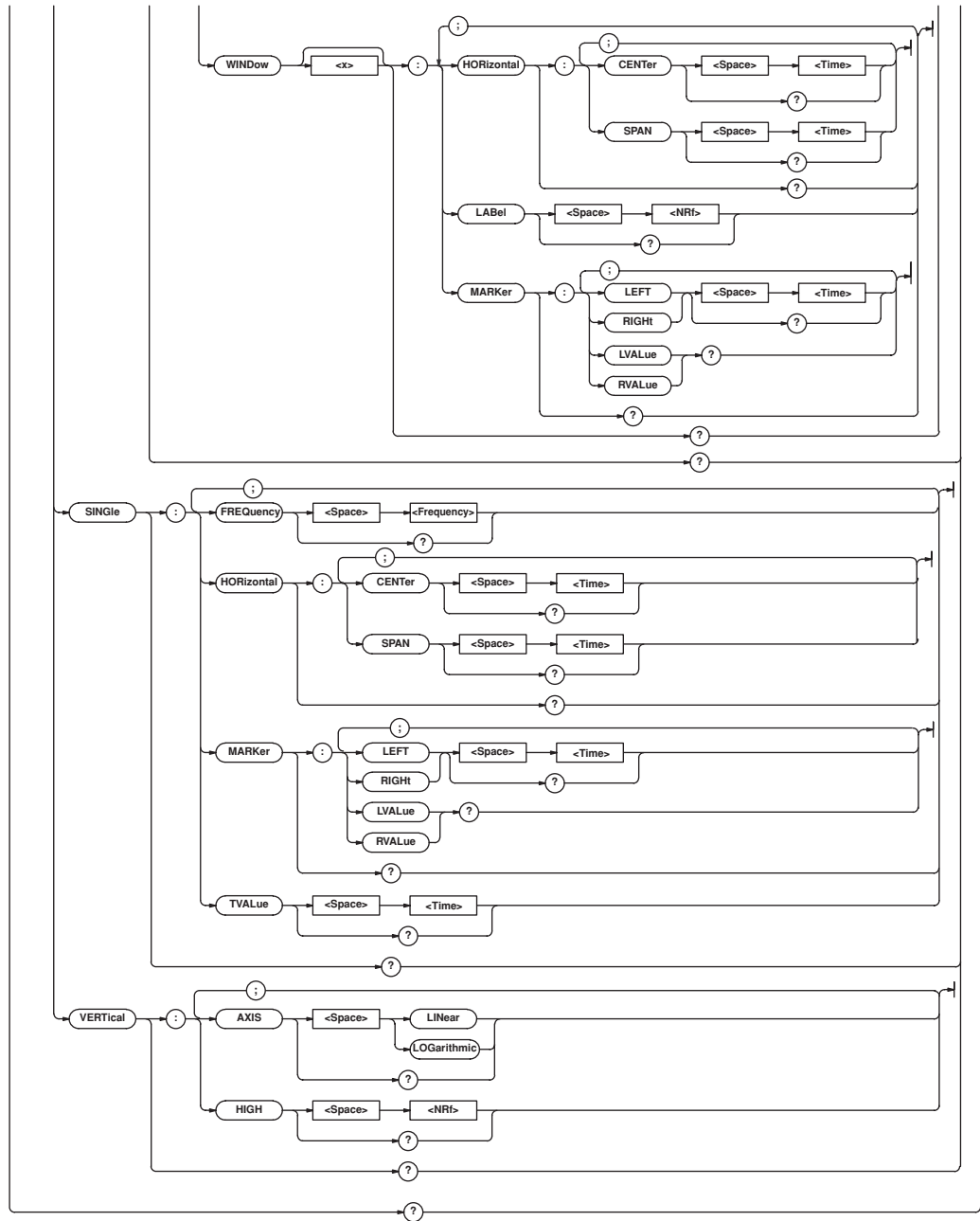
Example :HCOpy:TONE BW
:HCOpy:TONE?
-> :HCOpy:TONE BW

4.8 HHISistogram<x> and THISistogram<x> Group

The commands in this group deal with the histogram display of hardware histogram and time stamp modes. You can make the same settings and inquiries as when the DISPLAY, SCALE, and MARK keys on the front panel are used.

- When the sampling mode is set to hardware histogram, use “:HHISistogram<x>”.
- When the sampling mode is set to time stamp, use “:THISistogram<x>”.
- When measuring the period, A-to-B time interval, or pulse width or when measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B, use “HHISistogram1 | THISistogram1” or “HHISistogram | THISistogram” for the command for measurement 1. However, the query is “HHISistogram1 | THISistogram1”.
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B, use “HHISistogram2 | THISistogram2” for the command for measurement 2.





4.8 HHistogram<x> and THISTogram<x> Group

: {HHistogram<x> | THISTogram<x>}?

Function Queries all settings related to the histogram display in hardware histogram or time stamp mode.

Syntax : {HHistogram<x> | THISTogram<x>}?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM?
-> :HHISTOGRAM1:MARKER:LOW 0;
STATE 1;:HHISTOGRAM1:MODE SINGLE;
SINGLE:HORIZONTAL;
CENTER 150.000E-09;SPAN 0.30E-06;;
HHISTOGRAM1:SINGLE:MARKER:
LEFT 0.0E+00;RIGHT 300.000E-09;;
HHISTOGRAM1:SINGLE:
FREQUENCY 26.315E+06;;
TVALUE 38.000E-09;:HHISTOGRAM1:
VERTICAL:AXIS LOGARITHMIC;
HIGH 0.1E+03

: {HHistogram<x> | THISTogram<x>} :AUTO?

Function Queries all settings related to the auto window.

Syntax : {HHistogram<x> | THISTogram<x>} :
AUTO?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:AUTO?
-> :HHISTOGRAM1:AUTO:
MODULATION EFM;TTYPE ESTIMATED;
WINDOW1:STATE 1

: {HHistogram<x> | THISTogram<x>} :AUTO:

MODulation

Function Sets the modulation type on the auto window or queries the current setting.

Syntax : {HHistogram<x> | THISTogram<x>} :
AUTO:MODulation {EFM|EFMPlus|M1T07}
: {HHistogram<x> | THISTogram<x>} :
AUTO:MODulation?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:AUTO:MODULATION EFM
:HHISTOGRAM1:AUTO:MODULATION?
-> :HHISTOGRAM1:AUTO:MODULATION EFM

: {HHistogram<x> | THISTogram<x>} :AUTO:

TTYPE

Function Sets how to determine constant T on the auto window or queries the current setting.

Syntax : {HHistogram<x> | THISTogram<x>} :
AUTO:TTYPE {MEASured|ESTimated}
: {HHistogram<x> | THISTogram<x>} :
AUTO:TTYPE?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:AUTO:TTYPE MEASURED
:HHISTOGRAM1:AUTO:TTYPE?
-> :HHISTOGRAM1:AUTO:TTYPE MEASURED

: {HHistogram<x> | THISTogram<x>} :AUTO:

WINDow1:STATe

Function Turns ON/OFF window 1 on the auto window or queries the current setting.

Syntax : {HHistogram<x> | THISTogram<x>} :
AUTO:WINDow1:STATe {<Boolean>}
: {HHistogram<x> | THISTogram<x>} :
AUTO:WINDow1:STATe?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:AUTO:WINDOW1:STATE ON
:HHISTOGRAM1:AUTO:WINDOW1:STATE?
-> :HHISTOGRAM1:AUTO:WINDOW1:
STATE ON

: {HHistogram<x> | THISTogram<x>} :MARKer?

Function Queries all settings related to the marker.

Syntax : {HHistogram<x> | THISTogram<x>} :
MARKer?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:MARKER?
-> :HHISTOGRAM1:MARKER:LOW 0;
STATE 1

: {HHistogram<x> | THISTogram<x>} :MARKer:LOW

Function Sets the low marker value or queries the current setting.

Syntax : {HHistogram<x> | THISTogram<x>} :
MARKer:LOW {<Nrf>}
: {HHistogram<x> | THISTogram<x>} :
MARKer:LOW?
<x>=1 to 2
<Nrf>=Frequency

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:MARKER:LOW 100
:HHISTOGRAM1:MARKER:LOW?
-> :HHISTOGRAM1:MARKER:LOW 100

:{HHISistogram<x>|THISistogram<x>}:MARKer[:STATE]

Function	Turns ON/OFF the marker or queries the current setting.
Syntax	:{HHISistogram<x> THISistogram<x>}:MARKer[:STATE] {<Boolean>} :{HHISistogram<x> THISistogram<x>}:MARKer:STATE? <x>=1 to 2
Example	(The following is an example for the hardware histogram mode.) :HHISTOGRAM1:MARKER:STATE ON :HHISTOGRAM1:MARKER:STATE? -> :HHISTOGRAM1:MARKER:STATE 1

:{HHISistogram<x>|THISistogram<x>}[:MODE]

Function	Sets the window mode or queries the current setting.
Syntax	:{HHISistogram<x> THISistogram<x>}[:MODE] {SINGLE MULTI AUTO} :{HHISistogram<x> THISistogram<x>}:MODE? <x>=1 to 2
Example	(The following is an example for the hardware histogram mode.) :HHISTOGRAM1:MODE SINGLE :HHISTOGRAM1:MODE? -> :HHISTOGRAM1:MODE SINGLE

:{HHISistogram<x>|THISistogram<x>}:MULTi?

Function	Queries all settings related to the multi window.
Syntax	:{HHISistogram<x> THISistogram<x>}:MULTi? <x>=1 to 2
Example	(The following is an example for the hardware histogram mode.) :HHISTOGRAM1:MULTI? -> :HHISTOGRAM1:MULTI:SIZE 1; FREQUENCY 26.315E+06; TVALUE 38.000E-09; OFFSET 10.000-09E;:HHISTOGRAM1: MULTI:WINDOW1:HORIZONTAL: CENTER 114.000E-09;SPAN 0.060E-06;; HHISTOGRAM1:MULTI:WINDOW1:LABEL 3; MARKER:LEFT 95.000E-09; RIGHT 133.000E-09

:{HHISistogram<x>|THISistogram<x>}:MULTi:FREQUENCY

Function	Sets constant T using the frequency format or queries the current setting.
Syntax	:{HHISistogram<x> THISistogram<x>}:MULTi:FREQUENCY {<Frequency>} :{HHISistogram<x> THISistogram<x>}:MULTi:FREQUENCY? <x>=1 to 2 <Frequency>=4 MHz to 1000 MHz
Example	(The following is an example for the hardware histogram mode.) :HHISTOGRAM1:MULTI:FREQUENCY 50MHz :HHISTOGRAM1:MULTI:FREQUENCY? -> :HHISTOGRAM1:MULTI: FREQUENCY 50.000E+06
Description	Setting/Query is made using the frequency format of “:{HHISistogram<x> THISistogram<x>}:MULTi:TVALUE”.

:{HHISistogram<x>|THISistogram<x>}:MULTi:OFFSET

Function	Sets the offset value on constant T or queries the current setting.
Syntax	:{HHISistogram<x> THISistogram<x>}:MULTi:OFFSET {<Time>} :{HHISistogram<x> THISistogram<x>}:MULTi:OFFSET? <x>=1 to 2 <Time>=-100 ns to 300 ns (25 ps steps)
Example	(The following is an example for the hardware histogram mode.) :HHISTOGRAM1:MULTI:OFFSET 10ns :HHISTOGRAM1:MULTI:OFFSET? -> :HHISTOGRAM1:MULTI: OFFSET 10.000E-09

:{HHISistogram<x>|THISistogram<x>}:MULTi:SIZE

Function	Sets the number of windows or queries the current setting.
Syntax	:{HHISistogram<x> THISistogram<x>}:MULTi:SIZE {<NRF>} :{HHISistogram<x> THISistogram<x>}:MULTi:SIZE? <x>=1 to 2 <NRF>=The number of windows (1 to 14)
Example	(The following is an example for the hardware histogram mode.) :HHISTOGRAM1:MULTI:SIZE 10 :HHISTOGRAM1:MULTI:SIZE? -> :HHISTOGRAM1:MULTI:SIZE 10

4.8 HHISTogram<x> and THISTogram<x> Group

:{HHISTogram<x>|THISTogram<x>}:MULTi:TVALue

Function Sets the constant T value or queries the current setting.

Syntax :{HHISTogram<x>|THISTogram<x>}:
MULTi:TVALue {<Time>}
:{HHISTogram<x>|THISTogram<x>}:
MULTi:TVALue?
<x>=1 to 2
<Time>=1 ns to 250 ns (25 ps steps)

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:TVALUE 100ns
:HHISTOGRAM1:MULTI:TVALUE?
-> :HHISTOGRAM1:MULTI:
TVALUE 100.000E-09
```

:{HHISTogram<x>|THISTogram<x>}:MULTi:UPDate

Function Changes the window setting based on the constant T and offset values.

Syntax :{HHISTogram<x>|THISTogram<x>}:
MULTi:UPDate
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:UPDATE
```

:{HHISTogram<x>|THISTogram<x>}:MULTi:WINDow<x>?

Function Queries all settings related to the specified window.

Syntax :{HHISTogram<x>|THISTogram<x>}:
MULTi:WINDow<x>?
<x> of HHISTogram<x>|THISTogram<x>
=1 to 2
<x> of WINDow<x>=Window number (1
to number of windows)

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:WINDOW1?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 114.000E-09;
SPAN 0.060E-06;:HHISTOGRAM1:MULTI:
WINDOW1:LABEL 3;MARKER:
LEFT 95.000E-09;RIGHT 133.000E-09
```

:{HHISTogram<x>|THISTogram<x>}:MULTi:WINDow<x>:HORizontal?

Function Queries all settings related to the horizontal axis (X-axis) of the specified window.

Syntax :{HHISTogram<x>|THISTogram<x>}:
MULTi:WINDow<x>:HORizontal?
<x> of HHISTogram<x>|THISTogram<x>
=1 to 2
<x> of WINDow<x>=Window number (1
to number of windows)

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 150.000E-09;
SPAN 0.3E-06
```

:{HHISTogram<x>|THISTogram<x>}:MULTi:WINDow<x>:HORizontal:CENTer

Function Sets the center position of the horizontal axis (X-axis) of the specified window.

Syntax :{HHISTogram<x>|THISTogram<x>}:
WINDow<x>:HORizontal:CENTer <Time>
:{HHISTogram<x>|THISTogram<x>}:
WINDow<x>:HORizontal:CENTer?
<x> of HHISTogram<x>|THISTogram<x>
=1 to 2
<x> of WINDow<x>=Window number (1
to number of windows)
<Time>=-50 ns to 3.2 ms (25 ps
steps)

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 150ns
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:CENTER 150.000E-09
```

**:{HHistogram<x>|THISTogram<x>}:MULTI:
WINDOW<x>:HORIZONTAL:SPAN**

Function Queries all settings related to the horizontal axis (X-axis) of the specified window.

Syntax :{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:HORIZONTAL:SPAN
{<Time>}
:{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:HORIZONTAL:SPAN?
<x> of HHistogram<x>|THISTogram<x>
=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)
<Time>=1.5 ns, 3 ns, 7.5 ns, 15 ns,
30 ns, 60 ns, 150 ns, 300 ns, 600
ns, 1.5 μ s, 3 μ s, 6 μ s

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:SPAN 300ns
:HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:SPAN?
-> :HHISTOGRAM1:MULTI:WINDOW1:
HORIZONTAL:SPAN 0.3E-06
```

**:{HHistogram<x>|THISTogram<x>}:MULTI:
WINDOW<x>:LABEL**

Function Sets the label of the specified window or queries the current setting.

Syntax :{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:LABEL {<Nrf>}
:{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:LABEL?
<x> of HHistogram<x>|THISTogram<x>
=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)
<Nrf>=1 to 16

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:WINDOW1:LABEL 3
:HHISTOGRAM1:MULTI:WINDOW1:LABEL?
-> :HHISTOGRAM1:MULTI:WINDOW1:
LABEL 3
```

**:{HHistogram<x>|THISTogram<x>}:MULTI:
WINDOW<x>:MARKER?**

Function Queries all settings related to the horizontal axis (X-axis) marker of the specified window.

Syntax :{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:MARKER?
<x> of HHistogram<x>|THISTogram<x>
=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)

Example (The following is an example for the hardware histogram mode.)

```
:HHISTOGRAM1:MULTI:WINDOW1:MARKER?
-> :HHISTOGRAM1:MULTI:WINDOW1:
MARKER:LEFT 100.000E-09;
RIGHT 200.000E-09
```

**:{HHistogram<x>|THISTogram<x>}:MULTI:
WINDOW<x>:MARKER:{LEFT|RIGHT}**

Function Sets the marker position of the specified window or queries the current setting.

Syntax :{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:MARKER:{LEFT|RIGHT}
<Time>
:{HHistogram<x>|THISTogram<x>}:
MULTI:WINDOW<x>:MARKER:{LEFT|
RIGHT}?
<x> of HHistogram<x>|THISTogram<x>
=1 to 2
<x> of WINDOW<x>=Window number (1
to number of windows)
<Time>=Marker position (25 ps
steps)

Example (The following is an example for the hardware histogram mode and left marker.)

```
:HHISTOGRAM1:MULTI:WINDOW1:MARKER:
LEFT 100ns
:HHISTOGRAM1:MULTI:WINDOW1:MARKER:
LEFT?
-> :HHISTOGRAM1:MULTI:WINDOW1:
MARKER:LEFT 100.000E-09
```

4.8 HHistogram<x> and THISTogram<x> Group

:{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LVALue|RVALue}?

Function Sets the frequency of the marker position of the specified window or queries the current setting.

Syntax :{HHistogram<x>|THISTogram<x>}:MULTI:WINDOW<x>:MARKER:{LVALue|RVALue}? [{POSitive|NEGative}]<x> of HHistogram<x>|THISTogram<x>=1 to 2<x> of WINDOW<x>=Window number (1 to number of windows)

Example (The following is an example for the hardware histogram mode.)
:HHISTGRAM1:MULTI:WINDOW1:MARKER:LVALE?
-> :HHISTOGRAM1:MULTI:WINDOW1:MARKER:LVALE 1234

Description When measuring both pulse widths or both edges and “:CALCulation:POLarity” is set to “POSNeg”, specify “POSitive|NEGative”.

:{HHistogram<x>|THISTogram<x>}:SINGLE?

Function Queries all settings related to the single window.

Syntax :{HHistogram<x>|THISTogram<x>}:SINGLE?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:SINGLE?
-> :HHISTOGRAM1:SINGLE:HORIZONTAL:CENTER 150.000E-09;SPAN 0.30E-06;;
HHISTOGRAM1:SINGLE:MARKER:
LEFT 0.0E+00;RIGHT 300.000E-09;
:HHISTOGRAM1:SINGLE:
FREQUENCY 26.315E+06;
TVALUE 38.000E-09

:{HHistogram<x>|THISTogram<x>}:SINGLE:FREQUENCY

Function Sets constant T using the frequency format or queries the current setting.

Syntax :{HHistogram<x>|THISTogram<x>}:SINGLE:FREQUENCY {<Frequency>}
:{HHistogram<x>|THISTogram<x>}:SINGLE:FREQUENCY?
<x>=1 to 2
<Frequency>=4 MHz to 1000 MHz

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:SINGLE:FREQUENCY 50MHz
:HHISTOGRAM1:SINGLE:FREQUENCY?
-> :HHISTOGRAM1:SINGLE:
FREQUENCY 50.000E+06

Description Setting/Query is made using the frequency format of
“:{HHistogram<x>|THISTogram<x>}:SINGLE:TVALUE”.

:{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL?

Function Queries all settings related to the horizontal axis (X-axis) of the single window.

Syntax :{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL?
<x>=1 to 2

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:SINGLE:HORIZONTAL?
-> :HHISTOGRAM:SINGLE:HORIZONTAL:
CENTER 150.000E-06;SPAN 0.3E-6

:{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL:CENTER

Function Sets the center position of the horizontal axis (X-axis) of the single window.

Syntax :{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL:CENTER {<Time>}
:{HHistogram<x>|THISTogram<x>}:SINGLE:HORIZONTAL:CENTER?
<x>=1 to 2

- When in hardware histogram mode
<Time>=-50 ns to 3.2 μs (25 ps steps)
- When in time stamp mode
<Time>=-50 ns to 20 ms (25 ps steps)

Example (The following is an example for the hardware histogram mode.)
:HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER 150ns
:HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER?
-> :HHISTOGRAM1:SINGLE:HORIZONTAL:
CENTER 150.000E-09

**:{HHISistogram<x>|THISistogram<x>}:SINGLE:
HORIZONTAL:SPAN**

Function	Sets the span of the horizontal axis (X-axis) of the single window.
Syntax	<pre> :{HHISistogram<x> THISistogram<x>}: SINGLE:HORIZONTAL:SPAN {<Time>} :{HHISistogram<x> THISistogram<x>}: SINGLE:HORIZONTAL:SPAN? <x>=1 to 2 • For hardware histogram mode <Time>=1.5 ns, 3 ns, 7.5 ns, 15 ns, 30 ns, 60 ns, 150 ns, 300 ns, 600 ns, 1.5 μs, 3 μs, 6 μs • For time stamp mode <Time>=1.5 ns, 3 ns, 7.5 ns, 15 ns, 30 ns, 60 ns, 150 ns, 300 ns, 600 ns, 1.5 μs, 3 μs, 6 μs, 15 μs, 30 μs, 60 μs, 150 μs, 300 μs, 600 μs, 1.5 ms, 3 ms, 6 ms, 15 ms, 30 ms </pre>
Example	(The following is an example for the hardware histogram mode.) <pre> :HHISTOGRAM1:SINGLE:HORIZONTAL: SPAN 300ns :HHISTOGRAM1:SINGLE:HORIZONTAL:SPAN? -> :HHISTOGRAM1:SINGLE:HORIZONTAL: SPAN 0.3E-06 </pre>

**:{HHISistogram<x>|THISistogram<x>}:SINGLE:
MARKER?**

Function	Queries all settings related to the marker of the single window.
Syntax	<pre> :{HHISistogram<x> THISistogram<x>}: SINGLE:MARKER? <x>=1 to 2 </pre>
Example	(The following is an example for the hardware histogram mode.) <pre> :HHISTOGRAM1:SINGLE:MARKER? -> :HHISTOGRAM1:SINGLE:MARKER: LEFT 100.000E-09;RIGHT 200.000E-09 </pre>

**:{HHISistogram<x>|THISistogram<x>}:SINGLE:
MARKER:{LEFT|RIGHT}**

Function	Sets the marker position of the single window or queries the current setting.
Syntax	<pre> :{HHISistogram<x> THISistogram<x>}: SINGLE:MARKER:{LEFT RIGHT} {<Time>} :{HHISistogram<x> THISistogram<x>}: SINGLE:MARKER:{LEFT RIGHT}? <x>=1 to 2 <Time>=Marker position (25 ps steps) </pre>
Example	(The following is an example for the hardware histogram mode and left marker.) <pre> :HHISTOGRAM1:SINGLE:MARKER: LEFT 100ns :HHISTOGRAM1:SINGLE:MARKER:LEFT? -> :HHISTOGRAM1:SINGLE:MARKER: LEFT 100.000E-09 </pre>

**:{HHISistogram<x>|THISistogram<x>}:SINGLE:
MARKER:{LVALue|RVALue}?**

Function	Sets the frequency of the left/right marker position of the single window or queries the current setting.
Syntax	<pre> :{HHISistogram<x> THISistogram<x>}: SINGLE:MARKER:{LVALue RVALue}? [{POSitive NEGative}] <x>=1 to 2 </pre>
Example	(The following is an example for the hardware histogram mode.) <pre> :HHISTOGRAM1:SINGLE:MARKER:LVALE? -> :HHISTOGRAM1:SINGLE:MARKER: LVALE 1234 </pre>
Description	When measuring both pulse widths or both edges and “:CALCulation:POLarity” is set to “POSNeg”, specify “POSitive NEGative”.

**:{HHISistogram<x>|THISistogram<x>}:SINGLE:
TVALUE**

Function	Sets the constant T value of the single window or queries the current setting.
Syntax	<pre> :{HHISistogram<x> THISistogram<x>}: SINGLE:TVALUE {<Time>} :{HHISistogram<x> THISistogram<x>}: SINGLE:TVALUE? <x>=1 to 2 <Time>=1 ns to 250 ns (25 ps steps) </pre>
Example	(The following is an example for the hardware histogram mode.) <pre> :HHISTOGRAM1:SINGLE:TVALUE 100ns :HHISTOGRAM1:SINGLE:TVALUE? </pre>

:{HHISistogram<x>|THISistogram<x>}:VERTICAL?

Function	Queries all settings related to the vertical axis (Y-axis) of the hardware histogram.
Syntax	<pre> :{HHISistogram<x> THISistogram<x>}: VERTICAL? <x>=1 to 2 </pre>
Example	(The following is an example for the hardware histogram mode.) <pre> :HHISTOGRAM1:VERTICAL? -> :HHISTOGRAM1:VERTICAL: AXIS LINEAR;HIGH 1.0E+03 </pre>

4.8 HHistogram<x> and THISTogram<x> Group

:{HHistogram<x>|THISTogram<x>}:VERTICAL:AXIS

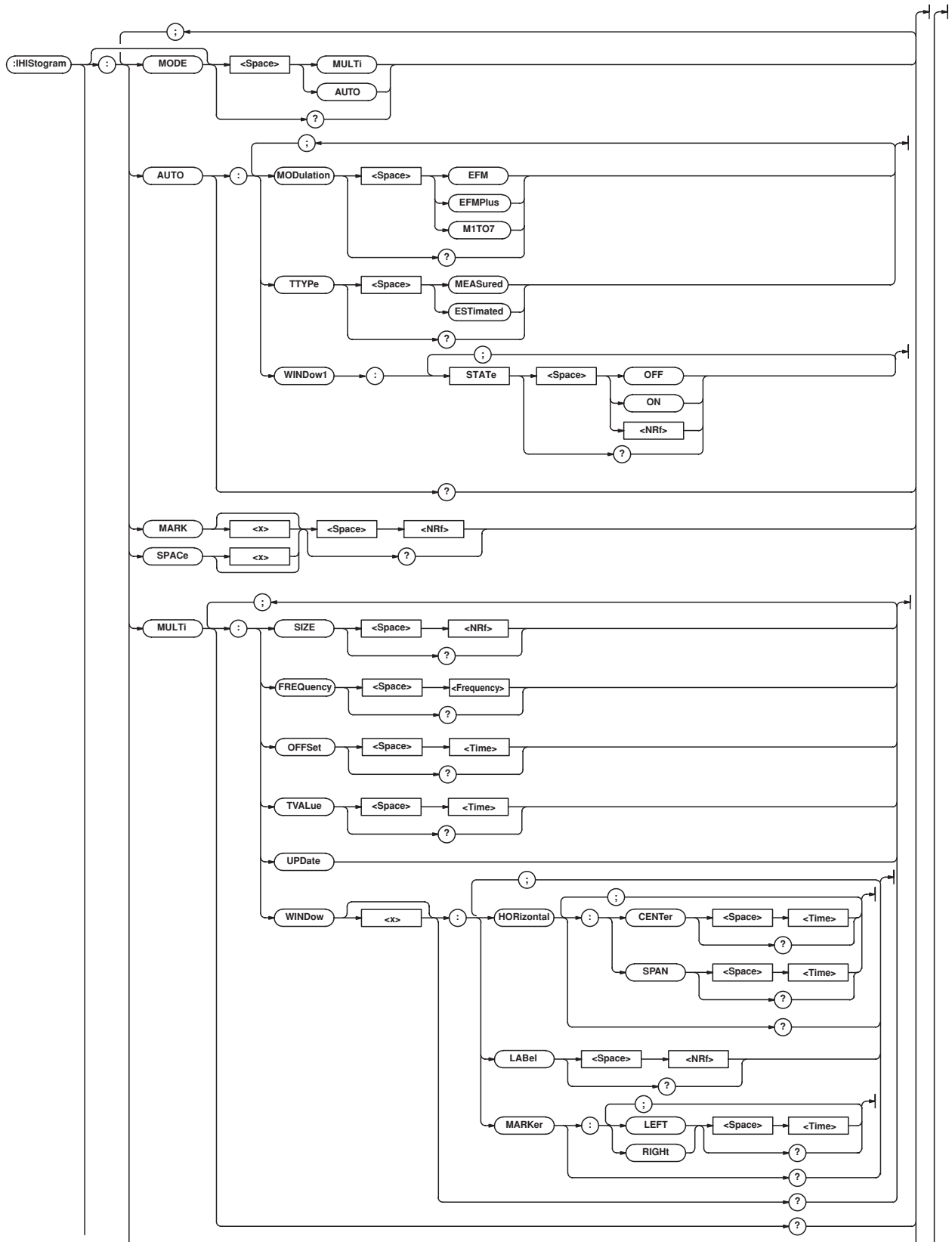
Function	Queries the scale type setting of the vertical axis (Y-axis).
Syntax	<pre>:{HHistogram<x> THISTogram<x>}: VERTICAL:AXIS {LINEar LOGarithmic} :{HHistogram<x> THISTogram<x>}: VERTICAL:AXIS? <x>=1 to 2</pre>
Example	(The following is an example for the hardware histogram mode.) <pre>:HHISTOGRAM1:VERTICAL:AXIS LINEAR :HHISTOGRAM1:VERTICAL:AXIS? -> :HHISTOGRAM1:VERTICAL: AXIS LINEAR</pre>

:{HHistogram<x>|THISTogram<x>}:VERTICAL:HIG

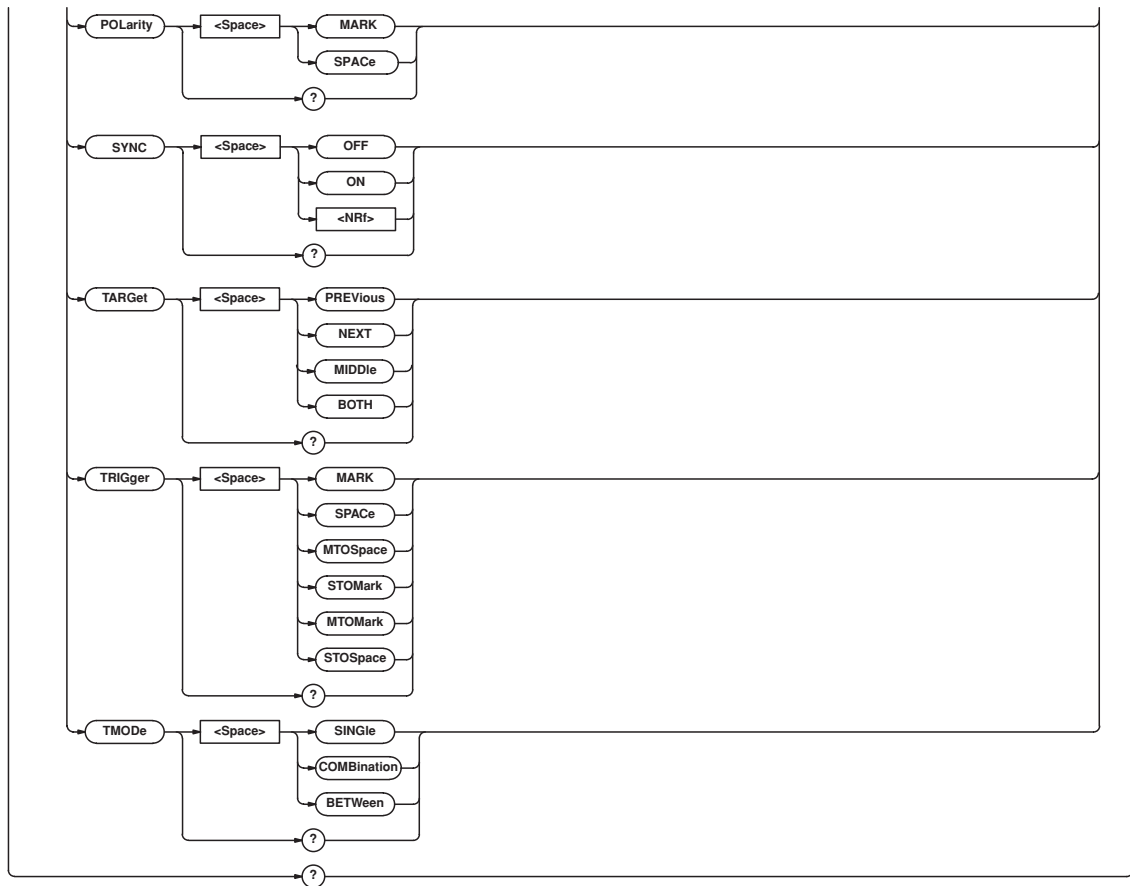
Function	Queries the upper limit of the vertical axis (Y-axis) scale.
Syntax	<pre>:{HHistogram<x> THISTogram<x>}: VERTICAL:HIG {<NRf>} :{HHistogram<x> THISTogram<x>}: VERTICAL:HIG? <x>=1 to 2 <NRf>=Frequency of the upper limit • When using linear scale 10, 20, 40, 100, 200, 400, 1000, 400000, 1E+6, 1E+7, 1E+8, 1E+9 • When using logarithmic scale 1E+1, 1E+2, ..., 1E+9</pre>
Example	(The following is an example for the hardware histogram mode.) <pre>:HHISTOGRAM1:VERTICAL:HIG 1000 :HHISTOGRAM1:VERTICAL:HIG? -> :HHISTOGRAM1:VERTICAL: HIG 1.0E+03</pre>

4.9 IHistogram Group

The commands in this group deal with the histogram display setting of the inter-symbol interference analysis mode. You can make the same settings and inquiries as when the DISPLAY, SCALE, and MARK keys on the front panel are used.



4.9 IHISTogram Group



:IHistogram?

Function Queries all settings related to the histogram display for inter-symbol interference analysis mode.

Syntax :IHistogram?

Example :IHISTGRAM?

```

-> :IHISTOGRAM:MARK1 4;MODE MULTI;
MULTI:SIZE 9;FREQUENCY 26.315E+06;
TVALUE 38.000E-09;
OFFSET 10.000E-09;WINDOW1:
HORIZONTAL:CENTER 86.000E-09;
SPAN 0.060E-06;:IHISTOGRAM:MULTI:
WINDOW1:LABEL 2;MARKER:
LEFT 67.000E-09;RIGHT 105.000E-09;:
IHISTOGRAM:MULTI:WINDOW2:HORIZONTAL:
CENTER 124.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW2:LABEL 3;
MARKER:LEFT 105.000E-09;
RIGHT 143.000E-09;:IHISTOGRAM:
MULTI:WINDOW3:HORIZONTAL:
CENTER 162.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW3:LABEL 4;
MARKER:LEFT 143.000E-09;
RIGHT 181.000E-09;:IHISTOGRAM:
MULTI:WINDOW4:HORIZONTAL:
CENTER 200.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW4:LABEL 5;
MARKER:LEFT 181.000E-09;
RIGHT 219.000E-09;:IHISTOGRAM:
MULTI:WINDOW5:HORIZONTAL:
CENTER 238.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW5:LABEL 6;
MARKER:LEFT 219.000E-09;
RIGHT 257.000E-09;:IHISTOGRAM:
MULTI:WINDOW6:HORIZONTAL:
CENTER 276.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW6:LABEL 7;
MARKER:LEFT 257.000E-09;
RIGHT 295.000E-09;:IHISTOGRAM:
MULTI:WINDOW7:HORIZONTAL:
CENTER 314.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW7:LABEL 8;
MARKER:LEFT 295.000E-09;
RIGHT 333.000E-09;:IHISTOGRAM:
MULTI:WINDOW8:HORIZONTAL:
CENTER 352.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW8:LABEL 9;
MARKER:LEFT 333.000E-09;
RIGHT 371.000E-09;:IHISTOGRAM:
MULTI:WINDOW9:HORIZONTAL:
CENTER 390.000E-09;SPAN 0.060E-06;:
IHISTOGRAM:MULTI:WINDOW9:LABEL 10;
MARKER:LEFT 371.000E-09;
RIGHT 409.000E-09;:IHISTOGRAM:
POLARITY SPACE;TADJUST:STATE 0;:
IHISTOGRAM:TARGET BOTH;
TRIGGER MARK;TMODE SINGLE

```

:IHistogram:AUTO?

Function Queries all settings related to the auto window.

Syntax :IHistogram:AUTO?

Example :IHISTOGRAM:AUTO?

```

-> :IHISTOGRAM:AUTO:MODULATION
EFMPLUS;TTYE ESTIMATED;WINDOW1:
STATE 1

```

:IHistogram:AUTO:MODulation

Function Sets the modulation type on the auto window or queries the current setting.

Syntax :IHistogram:AUTO:MODulation

```

{EFM|EFMplus|M1TO7}
:IHISTogram:AUTO:MODulation?

```

Example :IHISTOGRAM:AUTO:MODULATION EFMPLUS

```

:IHISTOGRAM:AUTO:MODULATION?
-> :IHISTOGRAM:AUTO:
MODULATION EFMPLUS

```

:IHistogram:AUTO:TTYe

Function Sets how to determine constant T on the auto window or queries the current setting.

Syntax :IHistogram:AUTO:TTYe

```

{MEASured|ESTimated}
:IHISTogram:AUTO:TTYe?

```

Example :IHISTOGRAM:AUTO:TTYE MEASURED

```

:IHISTOGRAM:AUTO:TTYE?
-> :IHISTOGRAM:AUTO:TTYE MEASURED

```

:IHistogram:AUTO:WINDow1:STATE

Function Turns ON/OFF window 1 on the auto window or queries the current setting.

Syntax :IHistogram:AUTO:WINDow1:STATE

```

{<Boolean>}
:IHISTogram:AUTO:WINDow1:STATE?

```

Example :IHISTOGRAM:AUTO:WINDOW1:STATE ON

```

:IHISTOGRAM:AUTO:WINDOW1:STATE?
-> :IHISTOGRAM:AUTO:WINDOW1:STATE
ON

```

4.9 IHistogram Group

:IHistogram:{MARK<x>|SPACE<x>}

Function Sets mark/space or queries the current setting.

Syntax :IHistogram:{MARK<x>|SPACE<x>}
{<Nrf>[, {LESS|MORE}]}
:IHistogram:{MARK<x>|SPACE<x>}?
<x> of MARK<x>=1 to 2
<x> of SPACE<x>=1 to 2
<Nrf>=Window label

Example :IHISTOGRAM:MARK1 3
:IHISTOGRAM:MARK1?
-> :IHISTOGRAM:MARK1 3

Description

- MARK1 can be specified when
":IHistogram:TRIGger" is set to "MARK".
- SPACE1 can be specified when
":IHistogram:TRIGger" is set to "SPACE".
- MARK1 and SPACE2 can be specified when
":IHistogram:TRIGger" is set to
"MTOSpace".
- SPACE1 and MARK2 can be specified when
":IHistogram:TRIGger" is set to
"STOMark".
- MARK1 and MARK2 can be specified when
":IHistogram:TRIGger" is set to
"MTOMark".
- SPACE1 and SPACE2 can be specified when
":IHistogram:TRIGger" is set to
"STOSpace".

:IHistogram[:MODE]

Function Sets the window mode or queries the current setting.

Syntax :IHistogram[:MODE] {MULTi|AUTO}
:IHistogram:MODE?

Example :IHISTOGRAM:MODE MULTI
:IHISTOGRAM:MODE?
-> :IHISTOGRAM:MODE MULTI

:IHistogram:MULTi?

Function Queries all settings related to the multi window.

Syntax :IHistogram:MULTi?

Example :IHISTOGRAM:MULTI?
-> :IHISTOGRAM:MULTI:SIZE 10;
FREQUENCY 26.315E+06;
TVALUE 38.000E-09;
OFFSET 10.000E-09;WINDOW1:
HORIZONTAL:CENTER 114.000E-09;
SPAN 0.060E-06;:IHISTOGRAM:MULTI:
WINDOW1:LABEL 1;MARKER:
LEFT 95.000E-09;RIGHT 133.000E-09;;
IHISTOGRAM:MULTI:WINDOW2:
HORIZONTAL:CENTER 152.000E-09;
SPAN 0.060E-06;:IHISTOGRAM:MULTI:
WINDOW2:LABEL 2;MARKER:
LEFT 133.000E-09;
RIGHT 171.000E-09;:IHISTOGRAM:
MULTI:WINDOW3:HORIZONTAL:

CENTER 190.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW3:LABEL 3;
MARKER:LEFT 171.000E-09;
RIGHT 209.000E-09;:IHISTOGRAM:
MULTI:WINDOW4:HORIZONTAL:
CENTER 228.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW4:LABEL 4;
MARKER:LEFT 209.000E-09;
RIGHT 247.000E-09;:IHISTOGRAM:
MULTI:WINDOW5:HORIZONTAL:
CENTER 266.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW5:LABEL 5;
MARKER:LEFT 247.000E-09;
RIGHT 285.000E-09;:IHISTOGRAM:
MULTI:WINDOW6:HORIZONTAL:
CENTER 304.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW6:LABEL 6;
MARKER:LEFT 285.000E-09;
RIGHT 323.000E-09;:IHISTOGRAM:
MULTI:WINDOW7:HORIZONTAL:
CENTER 342.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW7:LABEL 7;
MARKER:LEFT 323.000E-09;
RIGHT 361.000E-09;:IHISTOGRAM:
MULTI:WINDOW8:HORIZONTAL:
CENTER 380.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW8:LABEL 8;
MARKER:LEFT 361.000E-09;
RIGHT 399.000E-09;:IHISTOGRAM:
MULTI:WINDOW9:HORIZONTAL:
CENTER 418.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW9:LABEL 9;
MARKER:LEFT 399.000E-09;
RIGHT 437.000E-09;:IHISTOGRAM:
MULTI:WINDOW10:HORIZONTAL:
CENTER 532.000E-09;SPAN 0.060E-06;;
IHISTOGRAM:MULTI:WINDOW10:LABEL 10;
MARKER:LEFT 513.000E-09;
RIGHT 551.000E-09

:IHistogram:MULTi:FREQUENCY

Function Sets constant T using the frequency format or queries the current setting.

Syntax :IHistogram:MULTi:FREQUENCY
{<Frequency>}
:IHistogram:MULTi:FREQUENCY?
<Frequency>=4 MHz to 1000 MHz

Example :IHISTOGRAM:MULTI:FREQUENCY 50MHz
:IHISTOGRAM:MULTI:FREQUENCY?
-> :IHISTOGRAM:MULTI:
FREQUENCY 50.000E+06

Description Setting/Query is made using the frequency format of ":IHistogram:MULTi:TVALUE".

:IHISTogram:MULTi:OFFSet

Function Sets the offset value on constant T or queries the current setting.

Syntax :IHISTogram:MULTi:OFFSet {<Time>}
:IHISTogram:MULTi:OFFSet?
<Time>=-100 ns to 300 ns (25 ps steps)

Example :IHISTOGRAM:MULTI:OFFSET 10ns
:IHISTOGRAM:MULTI:OFFSET?
-> :IHISTOGRAM:MULTI:
OFFSET 10.000E-09

:IHISTogram:MULTi:SIZE

Function Sets the number of windows or queries the current setting.

Syntax :IHISTogram:MULTi:SIZE {<NRF>}
:IHISTogram:MULTi:SIZE?
<NRF>=The number of windows (1 to 14)

Example :IHISTOGRAM:MULTI:SIZE 10
:IHISTOGRAM:MULTI:SIZE?
-> :IHISTOGRAM:MULTI:SIZE 10

:IHISTogram:MULTi:TVALue

Function Sets the constant T value of the single window or queries the current setting.

Syntax :IHISTogram:MULTi:TVALue {<Time>}
:IHISTogram:MULTi:TVALue?
<Time>=1 ns to 250 ns (25 ps steps)

Example :IHISTOGRAM:MULTI:TVALUE 38ns
:IHISTOGRAM:MULTI:TVALUE?
-> :IHISTOGRAM:MULTI:
TVALUE 38.000E-09

:IHISTogram:MULTi:UPDate

Function Changes the window setting based on the constant T and offset values.

Syntax :IHISTogram:MULTi:UPDate

Example :IHISTOGRAM:MULTI:UPDATE

:IHISTogram:MULTi:WINDow<x>?

Function Queries all settings related to the specified window.

Syntax :IHISTogram:MULTi:WINDow<x>?
<x>=Window number (1 to number of windows)

Example :IHISTOGRAM:MULTI:WINDOW1?
-> :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:CENTER 114.000E-09;
SPAN 0.060E-06;:IHISTOGRAM:MULTI:
WINDOW1:LABEL 1;MARKER:
LEFT 95.000E-09;RIGHT 133.000E-09

:IHISTogram:MULTi:WINDow<x>:HORizontal?

Function Queries all settings related to the horizontal axis (X-axis) of the specified window.

Syntax :IHISTogram:MULTi:WINDow<x>:
HORizontal?
<x>=Window number (1 to number of windows)

Example :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:
-> :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:CENTER 114.000E-09;
SPAN 0.060E-06;

:IHISTogram:MULTi:WINDow<x>:HORizontal: CENTER

Function Sets the center position of the horizontal axis (X-axis) of a specific window.

Syntax :IHISTogram:MULTi:WINDow<x>:
HORizontal:CENTer {<Time>}
:IHISTogram:MULTi:WINDow<x>:
HORizontal:CENTer?
<x>=Window number (1 to number of windows)
<Time>=-50 ns to 3.2 μ s (25 ps steps)

Example :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:CENTER 114ns
:IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:CENTER?
-> :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:CENTER 114.000E-09

:IHISTogram:MULTi:WINDow<x>:HORizontal: SPAN

Function Queries all settings related to the horizontal axis (X-axis) of the specified window.

Syntax :IHISTogram:MULTi:WINDow<x>:
HORizontal:SPAN {<Time>}
:IHISTogram:MULTi:WINDow<x>:
HORizontal:SPAN?
<x>=Window number (1 to number of windows)
<Time>=1.5 ns, 3 ns, 7.5 ns, 15 ns, 30 ns, 60 ns, 150 ns, 300 ns, 600 ns, 1.5 μ s, 6 μ s

Example :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:SPAN 60ns
:IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:SPAN?
-> :IHISTOGRAM:MULTI:WINDOW1:
HORIZONTAL:SPAN 0.060E-06

4.9 IHistogram Group

:IHISTogram:MULTi:WINDow<x>:LABel

Function Sets the window label or queries the current setting.

Syntax **:IHISTogram:MULTi:WINDow<x>:LABel**
{<Nrf>}
:IHISTogram:MULTi:WINDow<x>:LABel?
<x>=Window number (1 to number of windows)
<Nrf>=1 to 16

Example **:IHISTOGRAM:MULTI:WINDOW1:LABEL 3**
:IHISTOGRAM:MULTI:WINDOW1:LABEL?
-> :IHISTOGRAM:MULTI:WINDOW1: LABEL 3

:IHISTogram:MULTi:WINDow<x>:MARKer?

Function Queries all settings related to the horizontal axis (X-axis) marker of the specified window.

Syntax **:IHISTogram:MULTi:WINDow<x>:MARKer?**
<x>=Window number (1 to number of windows)

Example **:IHISTOGRAM:MULTI:WINDOW1:MARKER?**
-> :IHISTOGRAM:MULTI:WINDOW1: MARKER:LEFT 95.000E-09; RIGHT 133.000E-09

:IHISTogram:MULTi:WINDow<x>:MARKer: {LEFT|RIGHT}

Function Sets the marker position of the specified window or queries the current setting.

Syntax **:IHISTogram:MULTi:WINDow<x>:MARKer: {LEFT|RIGHT} {<Time>}**
:IHISTogram:MULTi:WINDow<x>:MARKer: {LEFT|RIGHT}?
<x>=Window number (1 to number of windows)
<Time>=Marker position (25 ps steps)

Example **:IHISTOGRAM:MULTI:WINDOW1:MARKER: LEFT 100ns**
:IHISTOGRAM:MULTI:WINDOW1:MARKER: LEFT?
-> :IHISTOGRAM:MULTI:WINDOW1: MARKER:LEFT 100.000E-09

:IHISTogram:POLarity

Function Sets the polarity or queries the current setting.

Syntax **:IHISTogram:POLarity {MARK|SPACE}**
:IHISTogram:POLarity?

Example **:IHISTOGRAM:POLARITY SPACE**
:IHISTOGRAM:POLARITY?
-> :IHISTOGRAM:POLARITY SPACE

:IHISTogram:SYNC

Function Turns ON/OFF the Sync function or queries the current setting.

Syntax **:IHISTogram:SYNC {<Boolean>}**
:IHISTogram:SYNC?

Example **:IHISTOGRAM:SYNC ON**
:IHISTOGRAM:SYNC?
-> :IHISTOGRAM:SYNC 1

:IHISTogram:TARGet

Function Sets the analysis target or queries the current setting.

Syntax **:IHISTogram:TARGet**
{PREVIOUS|MIDDLE|NEXT|BOTH}
:IHISTogram:TARGet?

Example **:IHISTOGRAM:TARGET NEXT**
:IHISTOGRAM:TARGET?
-> :IHISTOGRAM:TARGET NEXT

Description

- “PREVIOUS|NEXT|BOTH” can be specified when measuring pulse width or pulse width A→pulse width B and “:IHISTogram:TMODE” is set to “SINGLE”.
- “PREVIOUS|NEXT” can be specified when measuring pulse width or pulse width A→pulse width B and “:IHISTogram:TMODE” is set to “COMBination”.
- “PREVIOUS|NEXT” can be specified when measuring pulse width A→A-to-B time interval of one edge and “:IHISTogram:TMODE” is set to “COMBination”.
- “PREVIOUS|NEXT” can be specified when measuring pulse width A→A-to-B time interval of both edges and “:IHISTogram:TMODE” is set to “SINGLE”.
- “PREVIOUS|MIDDLE|NEXT” can be specified when measuring pulse width A→A-to-B time interval of both edges and “:IHISTogram:TMODE” is set to “COMBination”.
- Setting is not possible when “:IHISTogram:TMODE” is set to “BETween”.

:IHISTogram:TMODe

Function Sets the trigger mode or queries the current setting.

Syntax **:IHISTogram:TMODe**
{SINGLE|COMBination|BETween}
:IHISTogram:TMODe?

Example **:IHISTOGRAM:TMODE COMBINATION**
:IHISTOGRAM:TMODE?
-> :IHISTOGRAM:TMODE COMBINATION

Description “BETween” cannot be specified when measuring pulse width A→A-to-B time interval.

:IHistogram:TRIGger

Function Sets the trigger condition or queries the current setting.

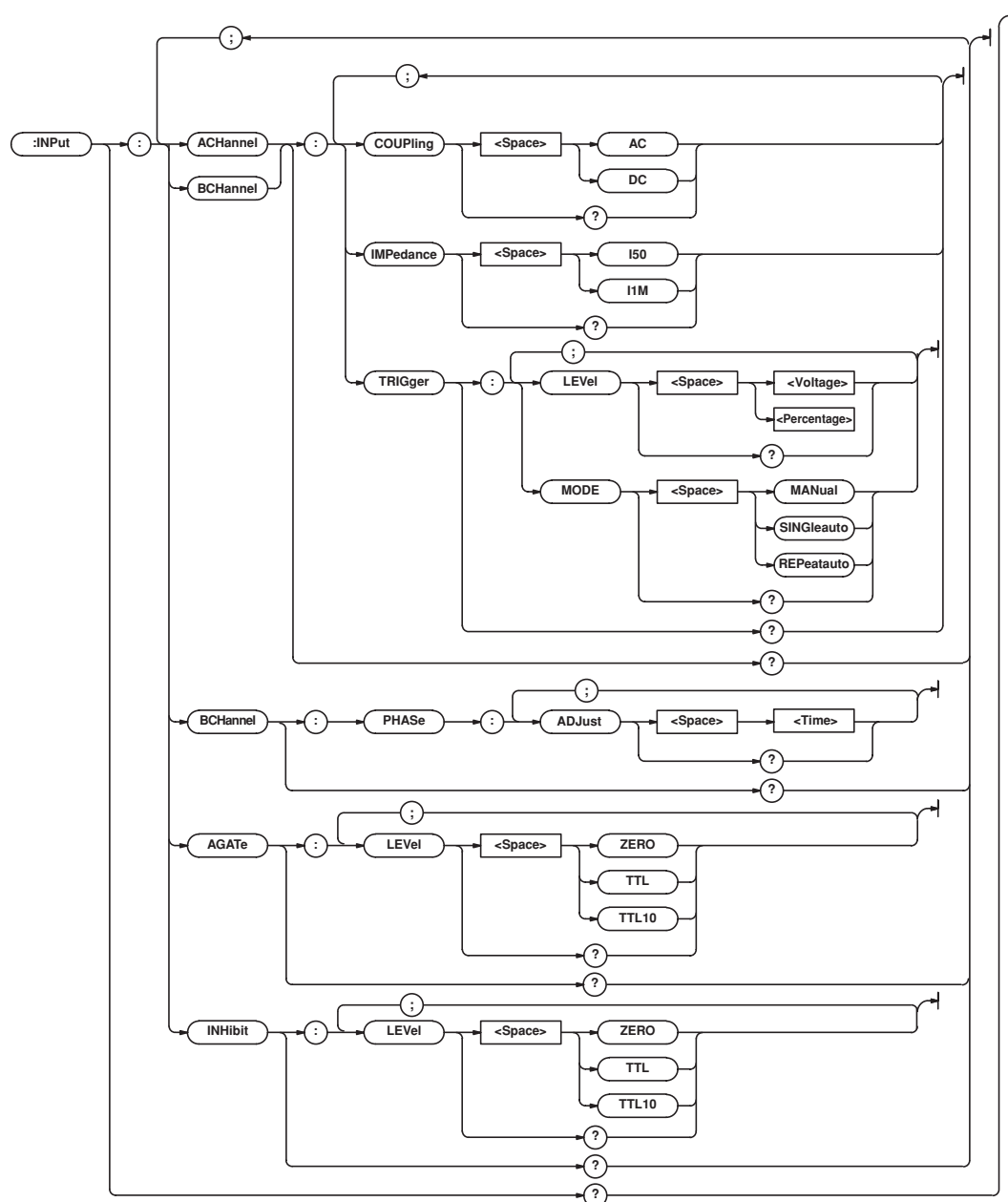
Syntax :IHistogram:TRIGger
 {MARK|SPACE|MTOSpace|STOMark|
 MTOMark|STOSpace}
 :IHistogram:TRIGger?

Example :IHISTOGRAM:TRIGGER MARK
 :IHISTOGRAM:TRIGGER?
 -> :IHISTOGRAM:TRIGGER MARK

Description • "MARK|SPACE" can be specified when
 ":IHistogram:TMODE" is set to "SINGLE".
 • "MTOSpace|STOMark" can be specified
 when ":IHistogram:TMODE" is set to
 "COMBination".
 • "MTOMark|STOSpace" can be specified
 when ":IHistogram:TMODE" is set to
 "BETween".

4.10 INPut Group

The commands in this group deal with the input section. You can make the same settings and inquiries as when the INPUT key on the front panel is used.



:INPut?

Function Queries all settings related to the input conditions.

Syntax :INPut?

Example :INPut?

```
-> :INPUT:ACHANNEL:COUPLING DC;
IMPEDANCE I1M;TRIGGER:
LEVEL 2.500E+00;MODE MANUAL;:INPUT:
AGATE:LEVEL TTL;:INPUT:BCHANNEL:
COUPLING DC;IMPEDANCE I1M;TRIGGER:
LEVEL 0.000E+00;MODEMANUAL;:INPUT:
INHIBIT:LEVEL TTL
```

:INPut:{ACHannel|BChannel}?

Function Queries all settings related to the specified channel.

Syntax :INPut:{ACHannel|BChannel}?

Example (The following is an example for CH A.)

```
:INPUT:ACHANNEL?
-> :INPUT:ACHANNEL:COUPLING DC;
IMPEDANCE I1M;TRIGGER:
LEVEL 2.500E+00;TRIGGER:MODE MANUAL
```

:INPut:{ACHannel|BCHannel}:COUPling

Function Sets the coupling of the specified channel or queries the current setting.

Syntax **:INPut:{ACHannel|BCHannel}:COUPling**
{AC|DC}
:INPut:{ACHannel|BCHannel}:
COUPling?

Example (The following is an example for CH A.)
:INPUT:ACHANNEL:COUPLING DC
:INPUT:ACHANNEL:COUPLING?
-> :INPUT:ACHANNEL:COUPLING DC

:INPut:{ACHannel|BCHannel}:IMPedance

Function Sets the input impedance of the specified channel or queries the current setting.

Syntax **:INPut:{ACHannel|BCHannel}:**
IMPedance {I50|I1M}
:INPut:{ACHannel|BCHannel}:
IMPedance?

Example (The following is an example for CH A.)
:INPUT:ACHANNEL:IMPEDANCE I1M
:INPUT:ACHANNEL:IMPEDANCE?
-> :INPUT:ACHANNEL:IMPEDANCE I1M

:INPut:{ACHannel|BCHannel}:TRIGger?

Function Queries all settings related to the trigger of the specified channel.

Syntax **:INPut:{ACHannel|BCHannel}:TRIGger?**

Example (The following is an example for CH A.)
:INPUT:ACHANNEL:TRIGGER?
-> :INPUT:ACHANNEL:TRIGGER:
LEVEL 2.500E+00;MODE MANUAL

:INPut:{ACHannel|BCHannel}:TRIGger:LEVel

Function Sets the trigger level of the specified channel or queries the current setting.

Syntax **:INPut:{ACHannel|BCHannel}:TRIGger:**
LEVel {<Voltage>|<Percent>}
:INPut:{ACHannel|BCHannel}:TRIGger:
LEVel?
 When in trigger manual mode <Voltage>=-5 V to +5 V (1-mV steps)
 When in trigger single/repeat auto <Percent>= 0 PCT to 100 PCT

Example (The following is an example for CH A in trigger manual mode)
:INPUT:ACHANNEL:TRIGGER:LEVEL 2.5V
:INPUT:ACHANNEL:TRIGGER:LEVEL?
-> :INPUT:ACHANNEL:TRIGGER:
LEVEL 2.500E+00

:INPut:{ACHannel|BCHannel}:TRIGger:MODE

Function Sets the trigger mode of the specified channel or queries the current setting.

Syntax **:INPut:{ACHannel|BCHannel}:TRIGger:**
MODE {MANual|SINGleauto|REPEatauto}
:INPut:{ACHannel|BCHannel}:TRIGger:
MODE?

Example (The following is an example for CH A.)
:INPUT:ACHANNEL:TRIGGER:MODE MANUAL
:INPUT:ACHANNEL:TRIGGER:MODE?
-> :INPUT:ACHANNEL:TRIGGER:
MODE MANUAL

:INPut:AGAtE?

Function Queries all settings related to arming and external gate.

Syntax **:INPut:AGAtE?**

Example **:INPUT:AGATE?**
-> :INPUT:AGATE:LEVEL ZERO

Description Since the external arming input terminal is shared with the external gate input terminal, the setting applies to both.

:INPut:AGAtE:LEVel

Function Sets the arming/gate level or queries the current setting.

Syntax **:INPut:AGAtE:LEVel {ZERO|TTL|TTL10}**
:INPut:AGAtE:LEVel?

Example **:INPUT:AGATE:LEVEL ZERO**
:INPUT:AGATE:LEVEL?
-> :INPUT:AGATE:LEVEL ZERO

:INPut:BCHannel:PHASe?

Function Queries all settings related to the CH B phase adjustment.

Syntax **:INPut:BCHannel:PHASe?**

Example **:INPUT:BCHANNEL:PHASE?**
-> :INPUT:BCHANNEL:PHASE:
ADJUST 6.0E-09

Description Querying is possible when measuring A-to-B time interval, period A & A-to-B time interval, or pulse width A & A-to-B time interval.

:INPut:BCHannel:PHASe:ADJust

Function Sets the phase adjustment time of CH B or queries the current setting.

Syntax **:INPut:BCHannel:PHASe:ADJust**
{<Time>}
:INPut:BCHannel:PHASe:ADJust?
 <Time>=0 s to 10 ns (100 ps steps)

Example **:INPUT:BCHANNEL:PHASE:**
ADJUST 100ps
:INPUT:BCHANNEL:PHASE:ADJUST?
-> :INPUT:BCHANNEL:PHASE:
ADJUST 100.0E-12

4.10 INPut Group

:INPut:INHibit?

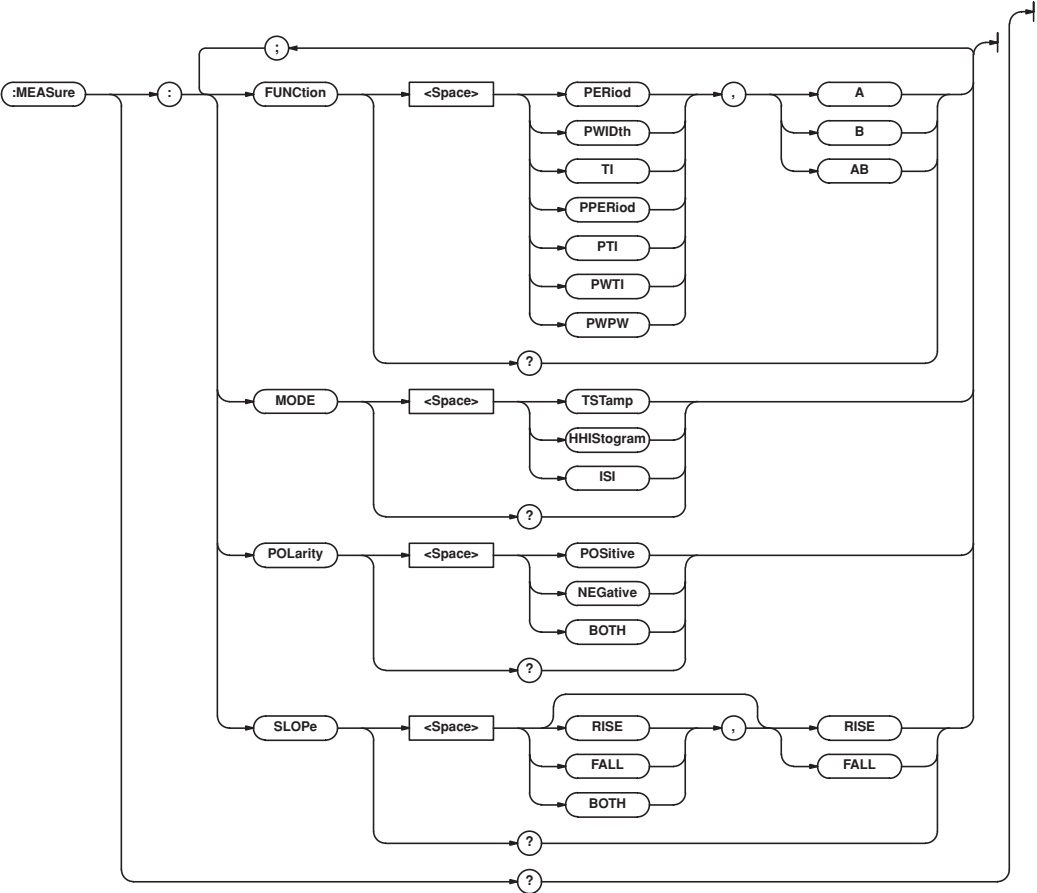
Function Queries all settings related to inhibit.
Syntax :INPut:INHibit?
Example :INPUT:INHIBIT?
 -> :INPUT:INHIBIT:LEVEL ZERO

:INPut:INHibit:LEVel

Function Sets the inhibit level or queries the current
 setting.
Syntax :INPut:INHibit:LEVel
 {ZERO|TTL|TTL10}
 :INPut:INHibit:LEVel?
Example :INPUT:INHIBIT:LEVEL TTL
 :INPUT:INHIBIT:LEVEL?
 -> :INPUT:INHIBIT:LEVEL TTL

4.11 MEASure Group

The commands in this group deal with measurement conditions. You can make the same settings and inquiries as when the MODE key on the front panel is used.



:MEASure?

Function Queries all settings related to the measurement conditions.

Syntax :MEASure?

Example :MEASURE?
 -> :MEASURE:MODE HHISTOGRAM;
 FUNCTION PERIOD,A;SLOPE RISE

:MEASure:FUNCTION

Function Sets the measurement function or queries the current setting.

Syntax :MEASure:FUNCTION
 {PERiod|PWIDth|TI|PPERiod|PTI|PWTI|PWPW},{A|B|AB}
 :MEASure:FUNCTION?

Example :MEASURE:FUNCTION PERIOD,A
 :MEASURE:FUNCTION?
 -> :MEASURE:FUNCTION PERIOD,A

Description

- “A | B” can be specified for the second parameter when measuring period or pulse width.
- Only AB can be specified for the second parameter when measuring A-to-B time interval (TI), period A & period B (PPERiod), period A & A-to-B time interval (PTI), pulse width A & A-to-B time interval (PWTI), pulse width A & pulse width B (PWPW).
- “PWIDth | PWTI | PWPW” can be specified when the sampling mode is set to inter-symbol interference analysis.

4.11 MEASure Group

:MEASure:MODE

Function Sets the sampling mode or queries the current setting.

Syntax **:MEASure:MODE**
{TSTamp|HHISTogram|ISI}
:MEASure:MODE?

Example **:MEASURE:MODE HHISTOGRAM**
:MEASURE:MODE?
-> :MEASURE:MODE HHISTOGRAM

:MEASure:POLarity

Function Sets the pulse width polarity when measuring pulse width, pulse width A & A-to-B time interval, or pulse width B or queries the current setting.

Syntax **:MEASure:POLarity**
{POSitive|NEGative|BOTH}
:MEASure:POLarity?

Example **:MEASURE:POLARITY POSITIVE**
:MEASURE:POLARITY?
-> :MEASURE:POLARITY POSITIVE

Description Querying is possible when measuring pulse width A & A-to-B time interval or pulse width A & pulse width B.

:MEASure:SLOPe

Function Sets the slope when measuring the period, A-to-B time interval, period A & period B, period A & A-to-B time interval, or pulse width A & A-to-B time interval or queries the current setting.

Syntax **:MEASure:SLOPe**
[{RISE|FALL|BOTH},] {RISE|FALL}
:MEASure:SLOPe?

Example **When measuring the period**
:MEASURE:SLOPE RISE
:MEASURE:SLOPE?
-> :MEASURE:SLOPE RISE

When measuring the A-to-B time interval
:MEASURE:SLOPE BOTH,RISE
:MEASURE:SLOPE?
-> :MEASURE:SLOPE BOTH,RISE

When measuring period A & period B
:MEASURE:SLOPE FALL,FALL
:MEASURE:SLOPE?
-> :MEASURE:SLOPE FALL,FALL

When measuring period A & A-to-B time interval
:MEASURE:SLOPE FALL,RISE
:MEASURE:SLOPE?
-> :MEASURE:SLOPE FALL,RISE

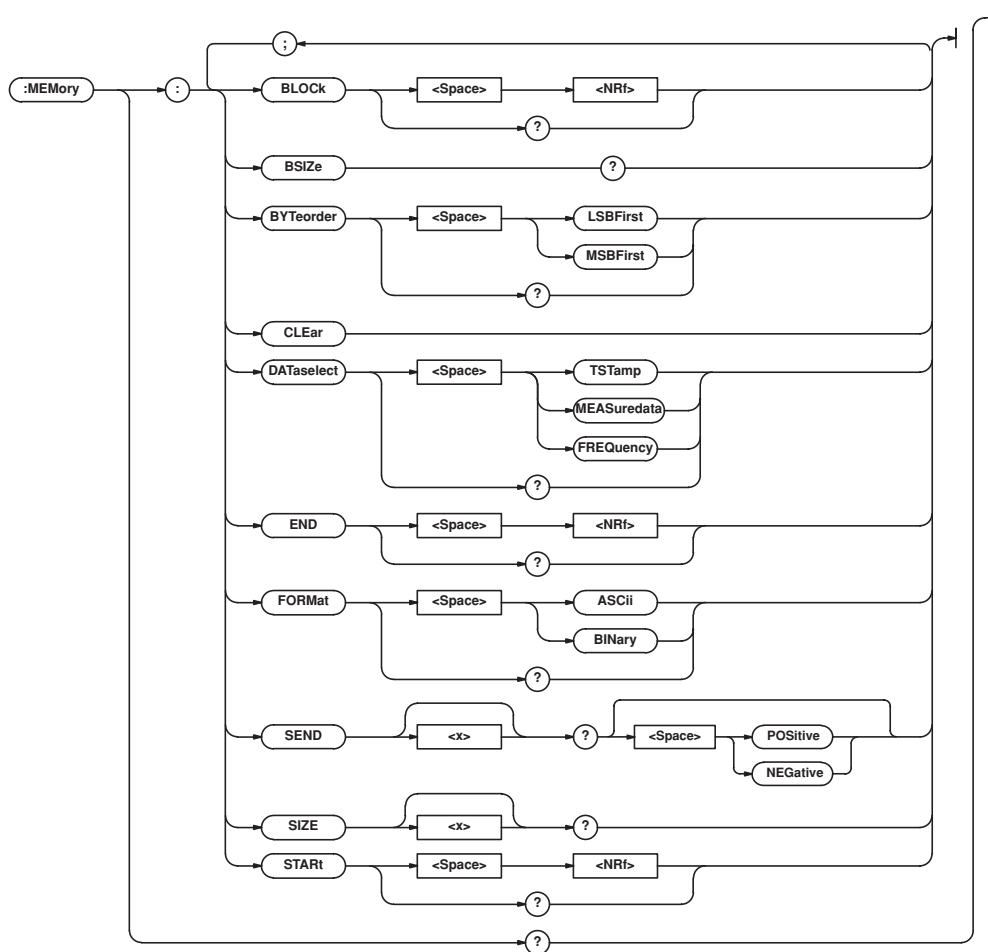
When measuring pulse width A & A-to-B time interval
:MEASURE:SLOPE?
-> :MEASURE:SLOPE BOTH,RISE

Description

- Specify only the second parameter when measuring the period.
- “RISE,RISE” or “FALL,FALL” can be specified when measuring period A & period B.
- “RISE,RISE” or “FALL,RISE” can be specified when measuring period A & A-to-B time interval.
- “BOTH,RISE” or “Both,FALL” can be specified when the sampling mode is set to time stamp or hardware histogram and pulse width A & A-to-B time interval is measured.

4.12 MEMory Group

The commands in this group deal with the external transmission of the measured data. There are no front panel keys that correspond to the commands in this group.



:MEMory?

Function Queries all settings related the external transmission of the measured data.

Syntax :MEMory?

Example :MEMory?

```
-> :MEMORY:BLOCK 0;
    BYTEORDER LSBFIRST;
    DATASELECT TSTAMP;END 51000;
    FORMAT ASCII;START 1
```

:MEMory:BLOCK

Function Sets the target block for block sampling or queries the current setting.

Syntax :MEMory:BLOCK {<Nrf>}

:MEMory:BLOCK?

<Nrf>=Block number (0 to 1000)

Example :MEMory:BLOCK 1

:MEMory:BLOCK?

```
-> :MEMORY:BLOCK 1
```

Description <Nrf>=0 indicates the all blocks.

:MEMory:BSIZE?

Function Queries the number of blocks that have been actually measured using block sampling.

Syntax :MEMory:BSIZE?

Example :MEMory:BSIZE?

```
-> :MEMORY:BSIZE 100
```

:MEMory:BYTEorder

Function Sets the transmission order of binary data or queries the current setting.

Syntax :MEMory:BYTEorder

{LSBFirst|MSBFirst}

:MEMory:BYTEorder?

Example :MEMory:BYTEORDER LSBFIRST

:MEMory:BYTEORDER?

```
-> :MEMORY:BYTEORDER LSBFIRST
```

:MEMory:CLear

Function Clears the measured data.

Syntax :MEMory:CLear

Example :MEMory:CLEAR

4.12 MEMory Group

:MEMory:DATaselect

Function	Sets the data to be transmitted or queries the current setting.
Syntax	:MEMory:DATaselect {TSTamp MEASuredata FREQuency} :MEMory:DATaselect?
Example	:MEMORY:DATASELECT TSTAMP :MEMORY:DATASELECT? -> :MEMORY:DATASELECT TSTAMP
Description	• If time stamp (TSTamp) is specified and the “:MEMory:SEND<x>?” command is executed when the sampling mode is set to hardware histogram, an error occurs. • If FREQuency is specified and the “:MEMory:SEND<x>?” command is executed when the sampling mode is set to time stamp or inter-symbol interference analysis, an error occurs.

:MEMory:END

Function	Sets the data position of transmission end or queries the current setting.
Syntax	:MEMory:END {<Nrf> <Time>} :MEMory:END? When in hardware histogram mode <Time>=-2 ns to 3.2 μ s When in time stamp or inter-symbol interference analysis mode • Period, A-to-B time interval, or pulse width <Nrf>=Number of points (1 to 1,024,000) • Period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, pulse width A & pulse width B <Nrf>=Number of points (1 to 512,000)
Example	:MEMORY:END 1ns :MEORY:END? -> :MEMORY:END 1.000E-09

:MEMory:FORMat

Function	Sets the format of the data to be transmitted or queries the current setting.
Syntax	:MEMory:FORMat {ASCIi BINary} :MEMory:FORMat?
Example	:MEMORY:FORMAT ASCII :MEMORY:FORMAT? -> :MEMORY:FORMAT ASCII

:MEMory:SEND<x>?

Function	Executes the transmission of the measured data specified by “MEMory:DATaselect.”
Syntax	:MEMory:SEND<x>? [{POSitive NEGative}] <x>=1 to 2
Example	:MEMORY:SEND1? -> #800000016abcdabceabcfabcbg
Description	• When the transmission format is ASCII, the response data in <NR3> form is output for the number of data points with each data point separated by commas. When the format is BINARY, a single block data is output. • The converting equation to a real number in binary data is as follows. When in hardware histogram mode • When measuring both pulse widths or both edges and “:CALCulation:POLarity” is set to “POSNeg”, specify the output data using “POSitive NEGative”. • Frequency (FREQuency) The frequency is stored using a 4-byte unsigned integer. • Measured data (MEASuredata) Assumed to be a signed 4-byte integer; the measured value is the value that results by multiplying 25 ps to this number. • Time stamp value (TSTamp) Time stamp cannot be directly output in hardware histogram mode. When in tiem stamp or inter-symbol interference analysis mode • Frequency (FREQuency) The frequency cannot be output directly in time stamp or inter-symbol interference mode. • Measured data (MEASuredata) Assumed to be an unsigned 4-byte integer; the measured value is the value that results by multiplying 25 ps to this number. • Time stamp value (TSTamp) Assumed to be an unsigned 4-byte integer; the time stamp value is the value that results by multiplying 100 ns to this number. • For a description of the binary data format , see <Block data> on page 3-6.

:MEMory:SIZE<x>?

Function Queries the number of data points that have been measured.

Syntax :MEMory:SIZE<x>?
 <x>=1 to 2

Example :MEMory:SIZE1?
 -> 10000

Description "NAN" is returned when the sampling mode is set to hardware histogram.

:MEMory:START

Function Sets the data position of transmission start or queries the current setting.

Syntax :MEMory:START {<NRf>}
 :MEMory:START?

When in hardware histogram mode
<Time>=-2 ns to 3.2 μs

When in time stamp or inter-symbol interference analysis mode

- Period, A-to-B time interval, or pulse width
 <NRf>=Number of points (1 to 1,024,000)
- Period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, pulse width A & pulse width B
 <NRf>=Number of points (1 to 512,000)

Example :MEMory:START 1ns
 :MEMory:START?
 -> :MEMory:START 1.000E-09

4.13 RECall Group



:RECall

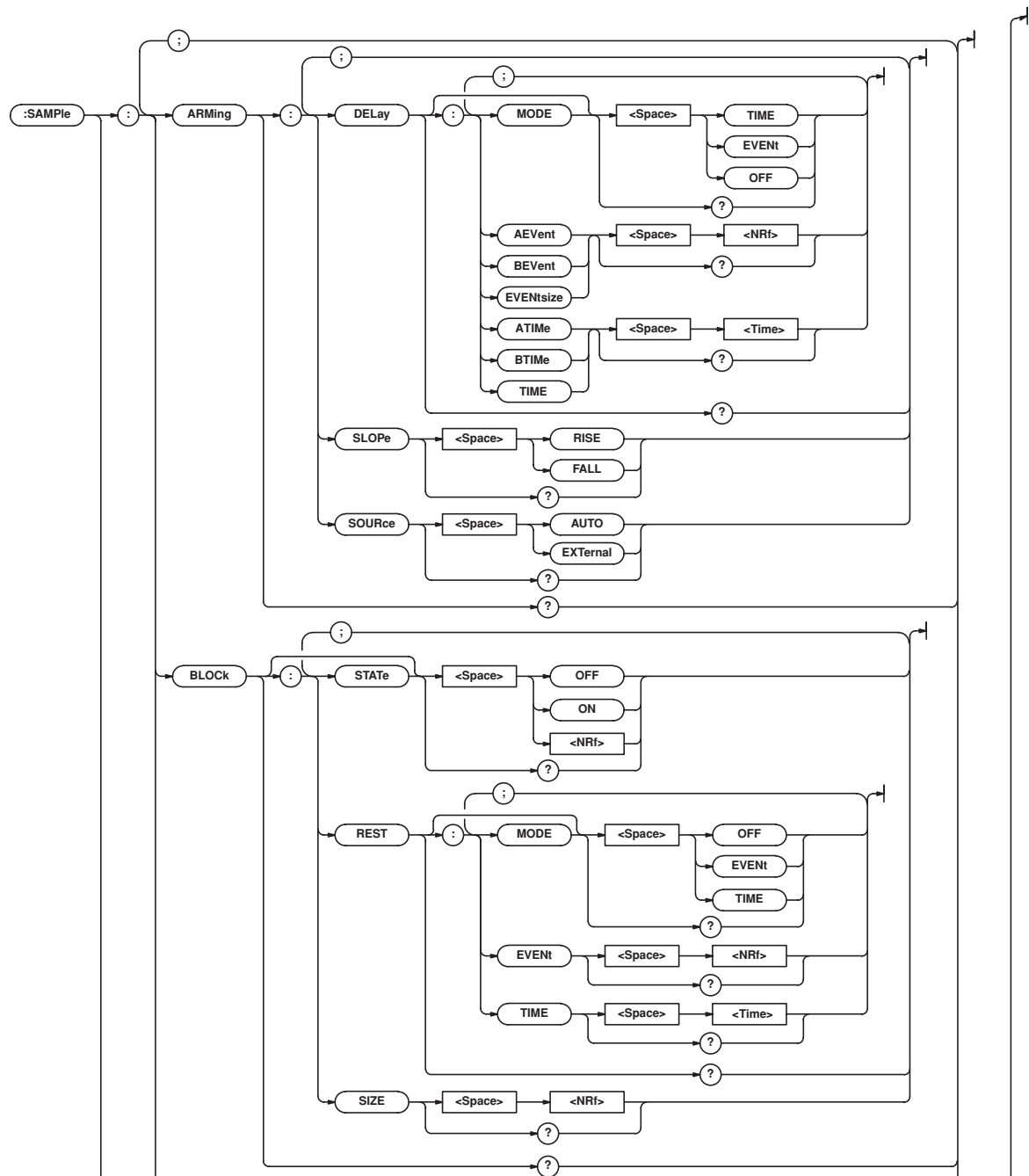
Function Recalls the setup data.

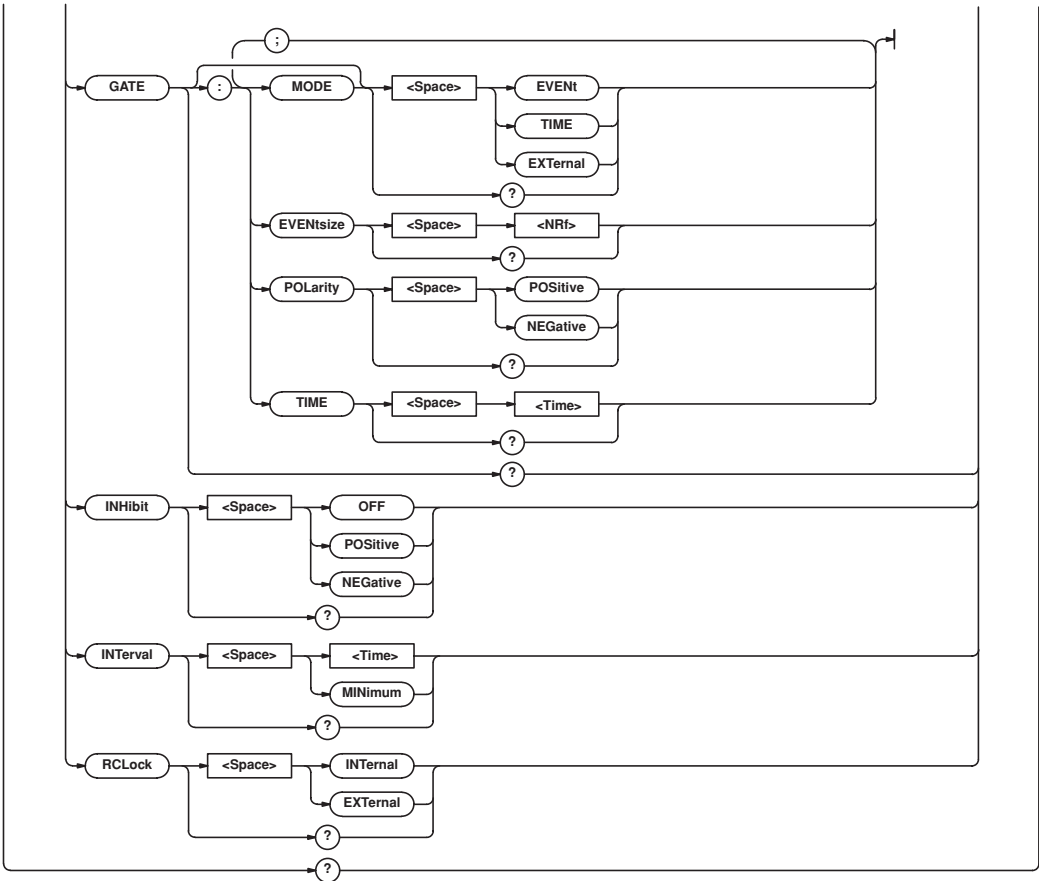
Syntax :RECall <NRf>
 <NRf>=0 to 31

Example :RECALL 0

4.14 SAMPlE Group

The commands in this group deal with sampling conditions. You can make the same settings and inquiries as when the SAMPLE key on the front panel is used.





:SAMPLE?
Function Queries all settings related to sampling.
Syntax :SAMPLE?
Example :SAMPLE?
-> :SAMPLE:ARMING:DELAY:
ATIME 1.0E-06;BTIME 1.0E-06;
MODE TIME;:SAMPLE:ARMING:
SLOPE FALL;SOURCE EXTERNAL;:SAMPLE:
GATE:EVENTSIZE 100;MODE EVENT;:
SAMPLE:INHIBIT OFF;RCLOCK INTERNAL

:SAMPLE:ARMing?
Function Queries all settings related to arming.
Syntax :SAMPLE:ARMing?
Example :SAMPLE:ARMING?
-> :SAMPLE:ARMING:DELAY:
ATIME 1.0E-06;BTIME 1.0E-06;
MODE TIME;:SAMPLE:ARMING:
SLOPE FALL;SOURCE EXTERNAL
Description Query is not possible when
":SAMPLE:GATE:MODE" is set to "EXTErnal".

:SAMPLE:ARMing:DElay?
Function Queries all settings related to arming delay.
Syntax :SAMPLE:ARMing:DElay?
Example :SAMPLE:ARMING:DELAY?
-> :SAMPLE:ARMING:DELAY:
ATIME 1.0E-06;BTIME 1.0E-06;
MODE TIME

4.14 SAMPlE Group

:SAMPlE:ARMinG:DELAy:{AEVEntsize|BEVEntsize|EVENtsize}

- Function Sets the arming delay event of the specified channel or queries the current setting.
- Syntax **:SAMPlE:ARMinG:DELAy:{AEVEntsize|BEVEntsize|EVENtsize} {<NRf>}**
:SAMPlE:ARMinG:DELAy:{AEVEntsize|BEVEntsize|EVENtsize}?
 <NRf>=Number of events (1 to 1000000)
- Example **:SAMPlE:ARMinG:DELAy:AEVENTSIZE 1000**
:SAMPlE:ARMinG:DELAy:AEVENTSIZE?
 -> **:SAMPlE:ARMinG:DELAy:AEVENTSIZE 1000**
- Description
- Setting is possible when “**:SAMPlE:ARMinG:DELAy:MODE**” is set to “**EVENT**”.
 - When measuring period A & period B or pulse width A & pulse width B, setting is possible for each channel {**AEVEntsize|BEVEntsize**}.
 - For all other cases, use {**EVENtsize**}.

:SAMPlE:ARMinG:DELAy:{ATIME|BTIME|TIME}

- Function Sets the arming delay time of the specified channel or queries the current setting.
- Syntax **:SAMPlE:ARMinG:DELAy:{ATIME|BTIME|TIME} {<Time>}**
:SAMPlE:ARMinG:DELAy:{ATIME|BTIME|TIME}?
 <Time>=1 μ s to 1 s (100 ns steps)
- Example **:SAMPlE:ARMinG:DELAy:BTIME 1us**
:SAMPlE:ARMinG:DELAy:BTIME?
 -> **:SAMPlE:ARMinG:DELAy:BTIME 1.0E-06**
- Description
- Setting is possible when “**:SAMPlE:ARMinG:DELAy:MODE**” is set to “**TIME**”.
 - When measuring period A & period B or pulse width A & pulse width B, setting is possible for each channel {**ATIME|BTIME**}.
 - For all other cases, use {**TIME**}.

:SAMPlE:ARMinG:DELAy[:MODE]

- Function Sets the arming delay mode or queries the current setting.
- Syntax **:SAMPlE:ARMinG:DELAy[:MODE]**
{TIME|EVENT|OFF}
:SAMPlE:ARMinG:DELAy:MODE?
- Example **:SAMPlE:ARMinG:DELAy:MODE OFF**
:SAMPlE:ARMinG:DELAy:MODE?
 -> **:SAMPlE:ARMinG:DELAy:MODE OFF**

:SAMPlE:ARMinG:SLOPe

- Function Sets the arming slope or queries the current setting.
- Syntax **:SAMPlE:ARMinG:SLOPe {RISE|FALL}**
:SAMPlE:ARMinG:SLOPe?
- Example **:SAMPlE:ARMinG:SLOPe RISE**
:SAMPlE:ARMinG:SLOPe?
 -> **:SAMPlE:ARMinG:SLOPe RISE**

:SAMPlE:ARMinG:SOURce

- Function Sets the arming source or queries the current setting.
- Syntax **:SAMPlE:ARMinG:SOURce**
{AUTO|EXTErnal}
:SAMPlE:ARMinG:SOURce?
- Example **:SAMPlE:ARMinG:SOURce AUTO**
:SAMPlE:ARMinG:SOURce?
 -> **:SAMPlE:ARMinG:SOURce AUTO**

:SAMPlE:BLOCK?

- Function Queries all settings related to block sampling.
- Syntax **:SAMPlE:BLOCK?**
- Example **:SAMPlE:BLOCK?**
 -> **:SAMPlE:BLOCK:REST:EVENT 1000;MODE EVENT;:SAMPlE:BLOCK:SIZE 10;STATE 1**
- Description
- Querying is not possible when measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B.
 - Query is not possible when “**:SAMPlE:GATE:MODE**” is set to “**EXTErnal**”.
 - Querying is not possible when in inter-symbol interference analysis mode.

:SAMPlE:BLOCK:REST?

- Function Queries all settings related to block sampling rest.
- Syntax **:SAMPlE:BLOCK:REST?**
- Example **:SAMPlE:BLOCK:REST?**
 -> **:SAMPlE:BLOCK:EVENT 1000;REST:MODE EVENT**

:SAMPLE:BLOCK:REST:EVENT

Function Sets the rest time of block sampling in terms of the number of events or queries the current setting.

Syntax :SAMPLE:BLOCK:REST:EVENT {<Nrf>}
:SAMPLE:BLOCK:REST:EVENT?
<Nrf>=Number of events (1 to 1,000,000)

Example :SAMPLE:BLOCK:REST:EVENT 1000
:SAMPLE:BLOCK:REST:EVENT?
-> :SAMPLE:BLOCK:REST:EVENT 1000

:SAMPLE:BLOCK:REST[:MODE]

Function Sets the block sampling rest mode or queries the current setting.

Syntax :SAMPLE:BLOCK:REST[:MODE]
{OFF|EVENT|TIME}
:SAMPLE:BLOCK:REST:MODE?

Example :SAMPLE:BLOCK:REST:MODE OFF
:SAMPLE:BLOCK:REST:MODE?
-> :SAMPLE:BLOCK:REST:MODE OFF

:SAMPLE:BLOCK:REST:TIME

Function Sets the block sampling rest time or queries the current setting.

Syntax :SAMPLE:BLOCK:REST:TIME {<Time>}
:SAMPLE:BLOCK:REST:TIME?
<Time>=1 μ s to 1 s (100 ns steps)

Example :SAMPLE:BLOCK:REST:TIME 1us
:SAMPLE:BLOCK:REST:TIME?
-> :SAMPLE:BLOCK:REST:TIME 1.0E-6

:SAMPLE:BLOCK:SIZE

Function Sets the number of blocks of block sampling or queries the current setting.

Syntax :SAMPLE:BLOCK:SIZE {<Nrf>}
:SAMPLE:BLOCK:SIZE?

When in time stamp or inter-symbol interference analysis mode

" :SAMPLE:BLOCK:REST:MODE" is set to "OFF"
<Nrf>=The number of blocks (2 to 1000)

When " :SAMPLE:BLOCK:REST:MODE" is set to "TIME|EVENT" or " :SAMPLE:ARMing:SOURce" is set to "EXTernal"

<Nrf>=The number of blocks (2 to 250)

When in hardware histogram mode

<Nrf>=The number of blocks (2 to 1000)

Example :SAMPLE:BLOCK:SIZE 10
:SAMPLE:BLOCK:SIZE?
-> :SAMPLE:BLOCK:SIZE 10

:SAMPLE:BLOCK[:STATE]

Function Turns ON/OFF block sampling or queries the current setting.

Syntax :SAMPLE:BLOCK[:STATE] {<Boolean>}
:SAMPLE:BLOCK:STATE?

Example :SAMPLE:BLOCK:STATE ON
:SAMPLE:BLOCK:STATE?
-> :SAMPLE:BLOCK:STATE 1

:SAMPLE:GATE?

Function Queries all settings related to the gate.

Syntax :SAMPLE:GATE?

Example :SAMPLE:GATE?
-> :SAMPLE:GATE:EVENTSIZE 1000;
MODE EVENT

:SAMPLE:GATE:EVENTsize

Function Sets the gate in terms of the number of events or queries the current setting.

Syntax :SAMPLE:GATE:EVENTsize {<Nrf>}
:SAMPLE:GATE:EVENTsize?

When in time stamp or inter-symbol interference analysis mode

When measuring period, A-to-B time interval, or pulse width

<Nrf>=Number of events (2 to 1,024,000)

When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B
<Nrf>=Number of events (1 to 512,000)

When in hardware histogram mode

When measuring period, A-to-B time interval, or pulse width

<Nrf>=Number of events (2 to 1,000,000,000)

When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B
<Nrf>=Number of events (1 to 1,000,000,000)

Example :SAMPLE:GATE:EVENTSIZE 1000
:SAMPLE:GATE:EVENTSIZE?
-> :SAMPLE:GATE:EVENTSIZE 1000

Description Setting is possible when "SAMPLE:GATE:MODE" is set to "EVENT".

4.14 SAMPlE Group/4.15 SStArt Group/4.16 STARt Group

:SAMPlE:GATE[:MODE]

Function Sets the gate type or queries the current setting.

Syntax **:SAMPlE:GATE[:MODE]**
{EVENT|TIME|EXTERNAL}
:SAMPlE:GATE:MODE?

Example **:SAMPlE:GATE:MODE EVENT**
:SAMPlE:GATE:MODE?
-> **:SAMPlE:GATE:MODE EVENT**

:SAMPlE:GATE:POLarity

Function Sets the polarity of the external gate or queries the current setting.

Syntax **:SAMPlE:GATE:POLarity**
{POSitive|NEGative}
:SAMPlE:GATE:POLarity?

Example **:SAMPlE:GATE:POLARITY POSITIVE**
:SAMPlE:GATE:POLARITY?
-> **:SAMPlE:GATE:POLARITY POSITIVE**

Description Setting is possible when "SAMPlE:GATE:MODE" is set to "EXTERNAL".

:SAMPlE:GATE:TIME

Function Sets the gate time in terms of time or queries the current setting.

Syntax **:SAMPlE:GATE:TIME {<Time>}**
:SAMPlE:GATE:TIME?
<Time>=1 μ s to 10 s (100 ns steps)

Example **:SAMPlE:GATE:TIME 1us**
:SAMPlE:GATE:TIME?
-> **:SAMPlE:GATE:TIME 1.0E-06**

Description Setting is possible when "SAMPlE:GATE:MODE" is set to "TIME".

:SAMPlE:INHibit

Function Sets the polarity of inhibit input or queries the current setting.

Syntax **:SAMPlE:INHibit**
{OFF|POSitive|NEGative}
:SAMPlE:INHibit?

Example **:SAMPlE:INHIBIT OFF**
:SAMPlE:INHIBIT?
-> **:SAMPlE:INHIBIT OFF**

Description When set to "OFF", the inhibit input is invalid.

:SAMPlE:INTERval

Function Sets the sampling interval or queries the current setting.

Syntax **:SAMPlE:INTERval {<Time>|MINimum}**
:SAMPlE:INTERval?
<Time>=0 to 1 s (1 μ s steps)

Example **:SAMPlE:INTERVAL MINIMUM**
:SAMPlE:INTERVAL?
-> **:SAMPlE:INTERVAL MINIMUM**

Description

- If "MINimum" is transmitted for the data or 0 is transmitted for <Time>, sampling is performed using the minimum interval.
- Setting is possible when the sampling mode is set to time stamp.

:SAMPlE:RClock

Function Sets reference sampling clock or queries the current setting.

Syntax **:SAMPlE:RClock {INTERNAL|EXTERNAL}**
:SAMPlE:RClock?

Example **:SAMPlE:RCLOCK INTERNAL**
:SAMPlE:RCLOCK?
-> **:SAMPlE:RCLOCK INTERNAL**

4.15 SStArt Group



:SStArt

Function Executes single measurement.

Syntax **:SStArt**

Example **:SSTART**

4.16 STARt Group



:STARt

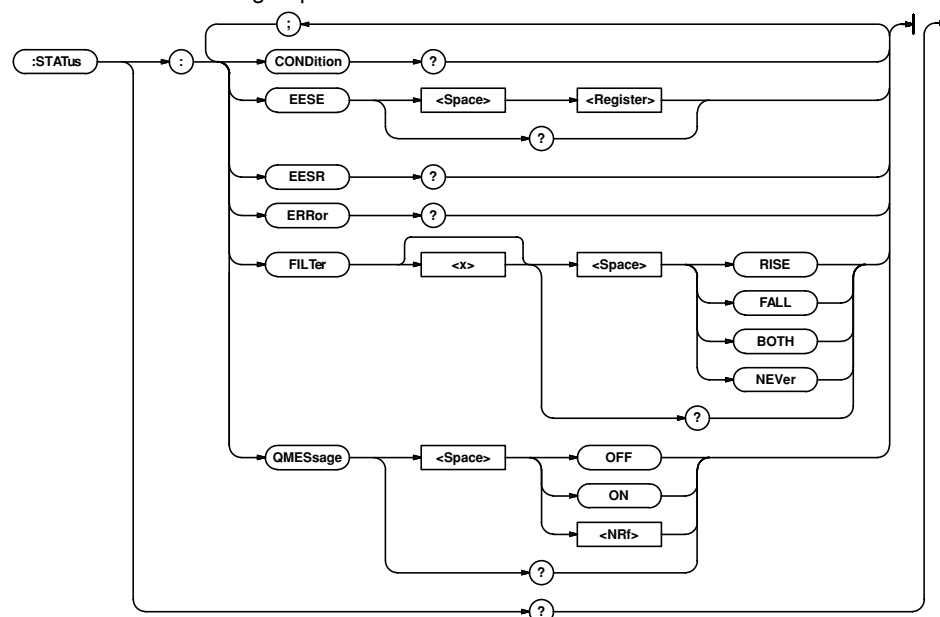
Function Starts the measurement.

Syntax **:STARt**

Example **:STARt**

4.17 STATUS Group

The commands in this group deal with communication status. For details on the status report, see chapter 5.



:STATUS?

Function Queries all settings related to the communication status.

Syntax :STATUS?

Example :STATUS?

```
-> :STATUS:EESr 0;FILTER1 RISE;
FILTER2 NEVER;FILTER3 NEVER;
FILTER4 NEVER;FILTER5 NEVER;
FILTER6 NEVER;FILTER7 NEVER;
FILTER8 NEVER;FILTER9 RISE;
FILTER10 RISE;FILTER11 RISE;
FILTER12 RISE;FILTER13 RISE;
FILTER14 NEVER;FILTER15 NEVER;
FILTER16 NEVER;QMESsage 1
```

:STATUS:CONDition?

Function Queries the contents of the condition register.

Syntax :STATUS:CONDition?

Example :STATUS:CONDITON?

```
-> 16
```

:STATUS:EESr

Function Sets the extended event enable register or queries the current setting.

Syntax :STATUS:EESr {<Register>}

:STATUS:EESr?

<Register>=0 to 65535

Example :STATUS:EESr 257

:STATUS:EESr?

```
-> :STATUS:EESr 257
```

:STATUS:EESR?

Function Queries the content of the extended event register and clears the register.

Syntax :STATUS:EESR?

Example :STATUS:EESR?

```
-> 1
```

:STATUS:ERRor?

Function Queries the error code and message information (top of the error queue).

Syntax :STATUS:ERRor?

Example :STATUS:ERRor?

```
-> 113,"Undefine header"
```

:STATUS:FILTer<x>

Function Sets the transition filter or queries the current setting.

Syntax :STATUS:FILTer<x>

{RISE | FALL | BOTH | NEVer}

:STATUS:FILTer<x>?

<x>=1 to 16

Example :STATUS:FILTer2 RISE

:STATUS:FILTer2?

```
-> :STATUS:FILTer2 RISE
```

:STATUS:QMESsage

Function Sets whether or not to attach message information to the response to the "STATUS:ERRor?" query or queries the current setting.

Syntax :STATUS:QMESsage {<Boolean>}

:STATUS:QMESsage?

Example :STATUS:QMESsage OFF

:STATUS:QMESsage?

```
-> :STATUS:QMESsage 0
```

4.18 STOP Group



:STOP

Function Stops the measurement.
Syntax :STOP
Example :STOP

4.19 STORe Group

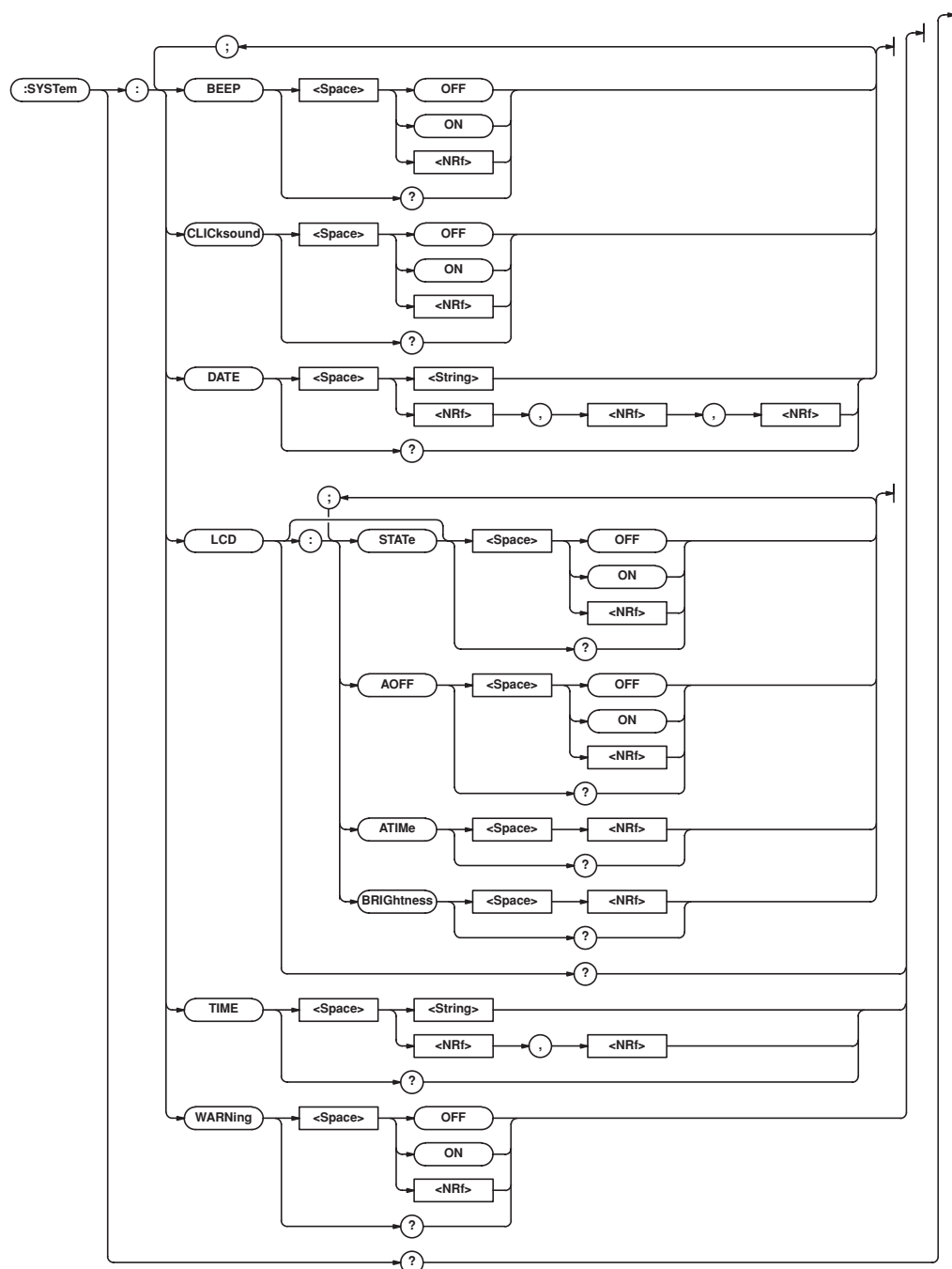


:STORe

Function Stores the setup data.
Syntax :STORe {<NRf>}
 <NRf>=0 to 31
Example :STORE 0

4.20 SYSTem Group

The commands in this group deal with basic system settings such as the date/time setting. You can make the same settings and inquiries as when the Config menu or LCD menu of the UTILITY key on the front panel is used.



:SYSTem?

Function Queries all settings related to the SYSTem group.

Syntax :SYSTem?

Example :SYSTem?

```

-> :SYSTem:BEEP 1:CLICKSOUND 1;LCD:
    AOFF 0;STATE 1;BRIGHTNESS 4;;
    SYSTem:WARNING 0
  
```

:SYSTem:BEEP

Function Turns ON/OFF the beep sound or queries the current setting.

Syntax :SYSTem:BEEP {<Boolean>}

:SYSTem:BEEP?

Example :SYSTem:BEEP ON

:SYSTem:BEEP?

```

-> :SYSTem:BEEP 1
  
```

4.20 SYSTem Group

:SYSTem:CLICksound

Function Turns ON/OFF the click sound or queries the current setting.

Syntax **:SYSTem:CLICksound** {<Boolean>}
:SYSTem:CLICksound?

Example **:SYSTEM:CLICKSOUND ON**
:SYSTEM:CLICKSOUND?
-> **:SYSTEM:CLICKSOUND 1**

:SYSTem:DATE

Function Sets the date or queries the current setting.

Syntax **:SYSTem:DATE** {<Character string>|,<NRf>,<NRf>,<NRf>}
:SYSTem:DATE?

Example **:SYSTEM:DATE "2002/01/01"**
:SYSTEM:DATE?
-> **"2002/01/01"**

Description <Character string> must be in the following format: year (4 digits)/month (2 digits)/day (2 digits). When in the <NRf> form, specify the date in the order year, month, and day.

:SYSTem:LCD?

Function Queries all settings related to the LCD.

Syntax **:SYSTEM:LCD?**

Example **:SYSTEM:LCD?**
-> **:SYSTEM:LCD:AOFF 1;ATIME 0**

:SYSTem:LCD:AOFF

Function Turns ON/OFF the auto off function of the LCD or queries the current setting.

Syntax **:SYSTem:LCD:AOFF** {<Boolean>}
:SYSTem:LCD:AOFF?

Example **:SYSTEM:LCD:AOFF ON**
:SYSTEM:LCD:AOFF?
-> **:SYSTEM:LCD:AOFF 1**

:SYSTem:LCD:ATIME

Function Sets the auto off time of the LCD or queries the current setting.

Syntax **:SYSTem:LCD:ATIME** {<NRf>}
:SYSTem:LCD:ATIME?
<NRf>=1 to 60 (minutes)

Example **:SYSTEM:LCD:ATIME 15**
:SYSTEM:LCD:ATIME?
-> **:SYSTEM:LCD:ATIME 15**

Description Setting is possible when **":SYSTEM:LCD:AOFF"** is set to "ON".

:SYSTem:LCD:BRIGhtness

Function Sets the brightness of the LCD or queries the current setting.

Syntax **:SYSTem:LCD:BRIGhtness** {<NRf>}
:SYSTem:LCD:BRIGhtness?
<NRf>=0 to 7

Example **:SYSTEM:LCD:BRIGHTNESS**
:SYSTEM:LCD:BRIGHTNESS?
-> **:SYSTEM:LCD:BRIGHTNESS**

:SYSTem:LCD[:STATE]

Function Turns ON/OFF the LCD backlight or queries the current setting.

Syntax **:SYSTem:LCD[:STATE]** {<Boolean>}
:SYSTem:LCD:STATE?

Example **:SYSTEM:LCD:STATE ON**
:SYSTEM:LCD:STATE?
-> **:SYSTEM:LCD:STATE 1**

:SYSTem:TIME

Function Sets the time or queries the current setting.

Syntax **:SYSTem:TIME** {<Character string>|,<NRf>,<NRf>}
:SYSTem:TIME?

Example **:SYSTEM:TIME "12:00"**
:SYSTEM:TIME?
-> **"12:00:00"**

Description <Character string> must be in the following format: hour (2 digits):minutes (2 digits). When in the <NRf> form, specify the time in the order hour and minute. The response to a query includes seconds.

:SYSTem:WARNing

Function Turns ON/OFF the warning display or queries the current setting.

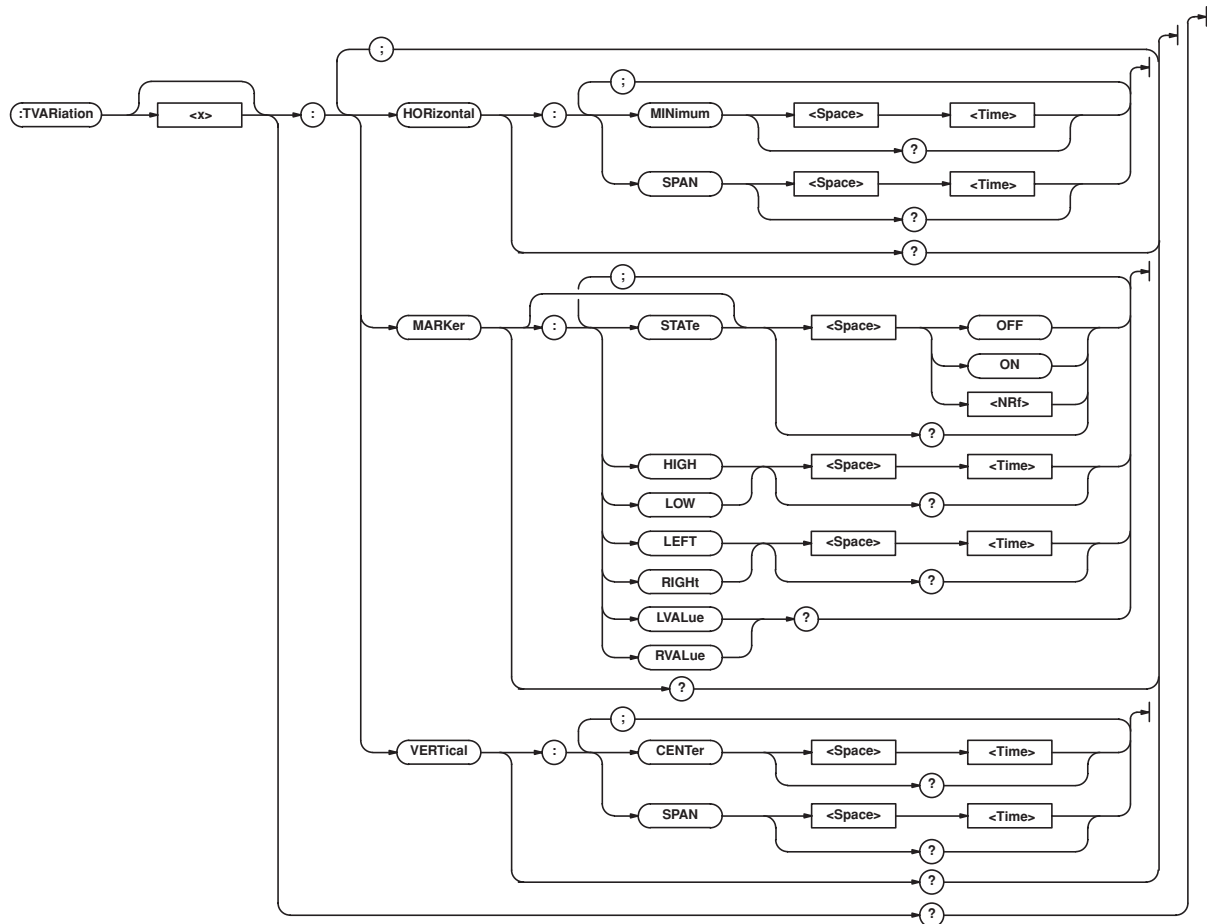
Syntax **:SYSTem:WARNing** <Boolean>
:SYSTem:WARNing?

Example **:SYSTEM:WARNING ON**
:SYSTEM:WARNING?
-> **:SYSTEM:WARNING 1**

4.21 TVARiation<x> Group

The commands in this group deal with the time variation display in time stamp mode. You can make the same settings and inquiries as when the DISPLAY, SCALE, and MARK keys on the front panel are used.

- When measuring the period, A-to-B time interval, or pulse width or when measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B, use "TVARiation1" or "TVARiation" for the command for measurement 1. However, the query is "TVARIATION1".
- When measuring period A & period B, period A & A-to-B time interval, pulse width A & A-to-B time interval, and pulse width A & pulse width B, use "TVARiation2" for the command for measurement 2.



:TVARiation<x>?

Function Queries all settings related to time variation.
Syntax :TVARiation<x>?
<x>=1 to 2
Example :TVARIATION?
-> :TVARIATION1:HORIZONTAL:
MINIMUM 100.0E-09;SPAN 0.06E-03;:
TVARIATION1:MARKER:
HIGH 150.000E-09;LEFT 1.0E-06;
LOW 100.000E-09;RIGHT 2.0E-06;
STATE 1:TVARIATION1:VERTICAL:
CENTER 100.000E-09;SPAN 0.2E-06

:TVARiation<x>:HORIZontal?

Function Queries all settings related to the horizontal axis (X-axis) of the time variation display.
Syntax :TVARiation<x>:HORIZontal?
<x>=1 to 2
Example :TVARIATION:HORIZONTAL?
-> :TVARIATION1:HORIZONTAL:
MINIMUM 100.0E-09;SPAN 0.06E-03

4.21 TVARiation<x> Group

:TVARiation<x>:HORizontal:MINimum

Function Sets the left end of the horizontal axis (X-axis) scale or queries the current setting.

Syntax :TVARiation<x>:HORizontal:MINimum {<Time>}
:TVARiation<x>:HORizontal:MINimum? <x>=1 to 2
<Time>=0 to 320 s (100 ns steps)

Example :TVARIATION1:HORIZONTAL:MINIMUM 100ns
:TVARIATION1:HORIZONTAL:MINIMUM? -> :TVARIATION1:HORIZONTAL:MINIMUM 100.0E-09

:TVARiation<x>:HORizontal:SPAN

Function Sets the span of the horizontal axis (X-axis) scale or queries the current setting.

Syntax :TVARiation<x>:HORizontal:SPAN {<Time>}
:TVARiation<x>:HORizontal:SPAN? <x>=1 to 2
<Time>=6 μ s, 12 μ s, 30 μ s, 60 μ s, 120 μ s, 300 μ s, 600 μ s, 1.2 ms, 3 ms, 6 ms, 12 ms, 30 ms, 60 ms, 120 ms, 300 ms, 600 ms, 1.2 s, 3 s, 6 s, 12 s, 30 s, 60 s, 120 s, 300 s, 600 s

Example :TVARIATION1:HORIZONTAL:SPAN 60us
:TVARIATION1:HORIZONTAL:SPAN? -> :TVARIATION1:HORIZONTAL:SPAN 0.06E-03

:TVARiation<x>:MARKer?

Function Queries all settings related to the marker.

Syntax :TVARiation<x>:MARKer?
<x>=1 to 2

Example :TVARIATION1:MARKer?
-> :TVARIATION1:MARKer:
HIGH 150.000E-09;LEFT 1.0E-06;
LOW 100.000E-09;RIGHT 2.0E-06;
STATE 1

:TVARiation<x>:MARKer:{HIGH|LOW}

Function Sets the position of the high/low marker or queries the current setting.

Syntax :TVARiation<x>:MARKer:{HIGH|LOW} {<Time>}
:TVARiation<x>:MARKer:{HIGH|LOW}? <x>=1 to 2
<Time>=Marker position (25 ps steps)

Example (The following is an example for the high marker.)
:TVARIATION1:MARKer:HIGH 150ns
:TVARIATION1:MARKer:HIGH? -> :TVARIATION1:MARKer:HIGH 150.000E-09

:TVARiation<x>:MARKer:{LEFT|RIGHT}

Function Sets the position of the left/right marker or queries the current setting.

Syntax :TVARiation:MARKer:{LEFT|RIGHT} {<Time>}
:TVARiation:MARKer:{LEFT|RIGHT}? <x>=1 to 2
<Time>=0 to 920 s (100 ns steps)

Example (The following is an example for the left marker.)
:TVARIATION1:MARKer:LEFT 150us
:TVARIATION1:MARKer:LEFT? -> :TVARIATION1:MARKer:LEFT 150.0E-03

:TVARiation<x>:MARKer:{LVALue|RVALue}?

Function Queries the measured value at the left/right marker position.

Syntax :TVARiation<x>:MARKer:{LVALue|RVALue}? <x>=1 to 2

Example (The following is an example for the left marker position.)
:TVARIATION1:MARKer:LVALUE? -> 1.2000E-06

:TVARiation<x>:MARKer[:STATe]

Function Turns ON/OFF the marker cursor or queries the current setting.

Syntax :TVARiation<x>:MARKer[:STATe] {<Boolean>}
:TVARiation<x>:MARKer:STATe? <x>=1 to 2

Example :TVARIATION1:MARKer:STATE ON
:TVARIATION1:MARKer:STATE? -> :TVARIATION1:MARKer:STATE 1

:TVARiation<x>:VERTical?

Function Queries all settings related to the vertical axis (Y-axis).

Syntax :TVARiation<x>:VERTical? <x>=1 to 2

Example :TVARIATION1:VERTical:CENTer? -> :TVARIATION1:VERTical:CENTer 100.000E-09;SPAN 0.2E-06

:TVARiation<x>:VERTical:CENTer

Function Queries the center value of the vertical axis (Y-axis).

Syntax :TVARiation<x>:VERTical:CENTer {<Time>}
:TVARiation<x>:VERTical:CENTer? <x>=1 to 2
<Time>=-50 ns to 20 ms (25 ps steps)

Example :TVARIATION1:VERTical:CENTer 100ns
:TVARIATION1:VERTical:CENTer? -> :TVARIATION1:VERTical:CENTer 100.000E-09

:TVARiation<x>:VERTical:SPAN

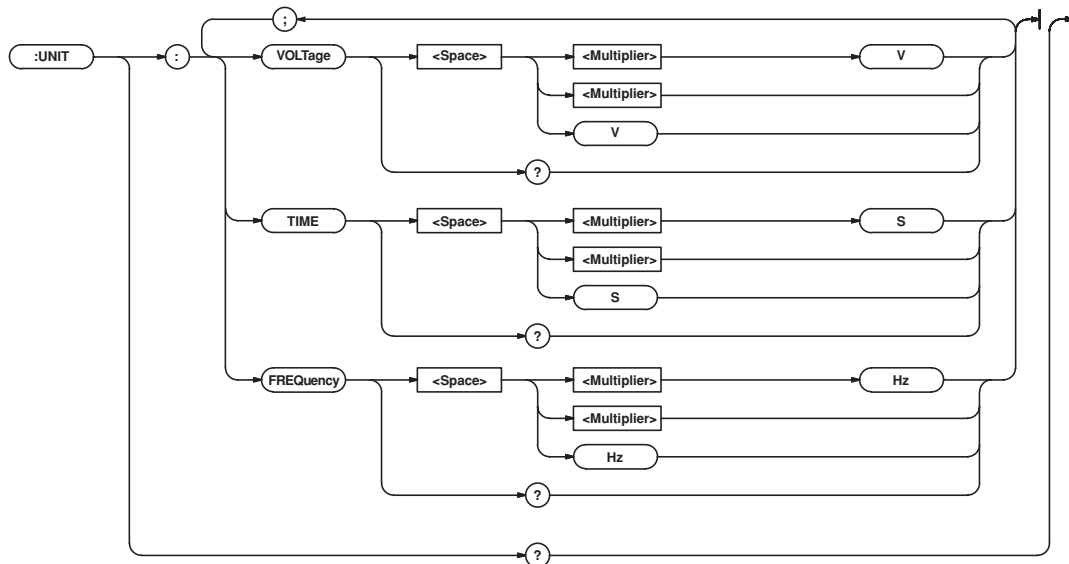
Function Queries the span of the vertical axis (Y-axis).

Syntax :TVARiation<x>:VERTical:SPAN
{<Time>}
:TVARiation<x>:VERTical:SPAN?
<x>=1 to 2
<Time>=500 ps, 1 ns, 2.5 ns, 5 ns,
10 ns, 20 ns, 50 ns, 100 ns, 200
ns, 500 ns, 1 μ s, 2 μ s, 5 μ s, 10 μ s,
20 μ s, 50 μ s, 100 μ s, 200 μ s, 500
 μ s, 1 ms, 2 ms, 5 ms, 10 ms, 20 ms

Example :TVARIATION1:VERTICAL:SPAN 200ns
:TVARIATION1:VERTICAL:SPAN?
-> :TVARIATION1:VERTICAL:
SPAN 0.2E-06

4.22 UNIT Group

The commands in this group deal with the default unit of voltage, time, and frequency data.

**:UNIT?**

Function Queries the default unit of voltage, time and frequency.

Syntax :UNIT?

Example :UNIT?
-> :UNIT:FREQUENCY HZ;VOLTAGE V;
TIME S

:UNIT:VOLTage

Function Sets the default unit of voltage or queries the current setting.

Syntax :UNIT:VOLTage {<Multiplier>}V
:UNIT:VOLTage?
<Multiplier> See section 3.4.

Example :UNIT:VOLTAGE V
:UNIT:VOLTAGE?
-> :UNIT:VOLTAGE V

:UNIT:TIME

Function Sets the default unit of time or queries the current setting.

Syntax :UNIT:TIME {<Multiplier>}S
:UNIT:TIME?
<Multiplier> See section 3.4.

Example :UNIT:TIME s
:UNIT:TIME?
-> :UNIT:TIME S

:UNIT:FREQuency

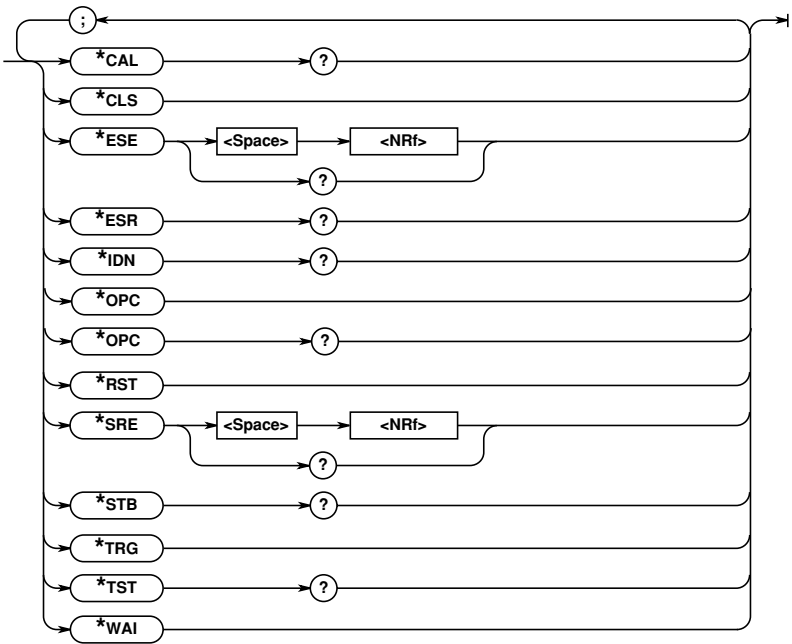
Function Sets the default unit of frequency or queries the current setting.

Syntax :UNIT:FREQuency {<Multiplier>}Hz
:UNIT:FREQuency?
<Multiplier> See section 3.4.

Example :UNIT:FREQuency HZ
:UNIT:FREQuency?
-> :UNIT:FREQuency HZ

4.23 Common Command Group

The commands in the common group are defined in the IEEE 488.2-1992 and are independent of the instrument's functions.



***CAL?**
Function Performs calibration and queries the result.
Syntax *CAL?
Example *CAL?
-> 0
Description If the calibration terminates normally, "0" is returned. Otherwise, a non-zero value is returned.

***CLS**
Function Clears the standard event register, extended event register, and error queue.
Syntax *CLS
Example *CLS
Description For details on the standard event register, extended event register, and error queue, see chapter 5, "Status Report."

***ESE**
Function Sets the standard event enable register or queries the current setting.
Syntax *ESE {<NRf>}
*ESE?
<NRf>=0 to 255
Example *ESE 253
*ESE?
-> 253
Description For details on the standard event enable register, see chapter 5, "Status Report."

***ESR?**
Function Queries the standard event register and clears the register.
Syntax *ESR?
Example *ESR?
-> 253
Description For details on the standard event register, see chapter 5, "Status Report."

***IDN?**
Function Queries the instrument model.
Syntax *IDN?
Example *IDN?
-> YOKOGAWA,704510,0,F1.01
Description A reply is returned in the following form: manufacturer, model, serial number (always 0), firmware version.

***OPC**
Function Sets a "1" to the standard event register bit upon the completion of the specified overlap command. Because the TA720 does not support overlap commands, the command is discarded.
Syntax *OPC

***OPC?**
Function Returns a "1" when the specified overlap command is finished. A "1" is always returned, because the TA720 does not support overlap commands.
Syntax *OPC?

***RST**

Function Initializes the setup data.

Syntax *RST

Example *RST

Description Executes the same operation as when the [Execute] soft key of the INIT (SHIFT+MARKER) key is pressed. For details on initialization, see the *TA720 User's Manual*.

***SRE**

Function Sets the service request enable register or queries the current setting.

Syntax *SRE {<NRf>}

*SRE?

<NRf>=0 to 255

Example *SRE 239

*SRE?

-> 239

***STB?**

Function Queries the status byte register.

Syntax *STB?

Example *STB?

-> 4

Description For details on the status byte register, see chapter 5, "Status Report."

***TRG?**

Function Executes the same operation as when the SINGLE key is pressed.

Syntax *TRG?

Description The multi-line message GET (Group Execute Trigger) also performs the same operation as this command.

***TST?**

Function Performs a self-test and queries the result.

Syntax *TST?

Example *TST?

-> 0

Description

- Executes the same operation as the [Board] test of the [Selftest] soft key.
- If the all the tests terminate normally, "0" is returned. Otherwise, a non-zero value is returned.

***WAI**

Function Holds the subsequent command until the completion of the specified overlap operation. Because the TA720 does not support overlap commands, the command is discarded.

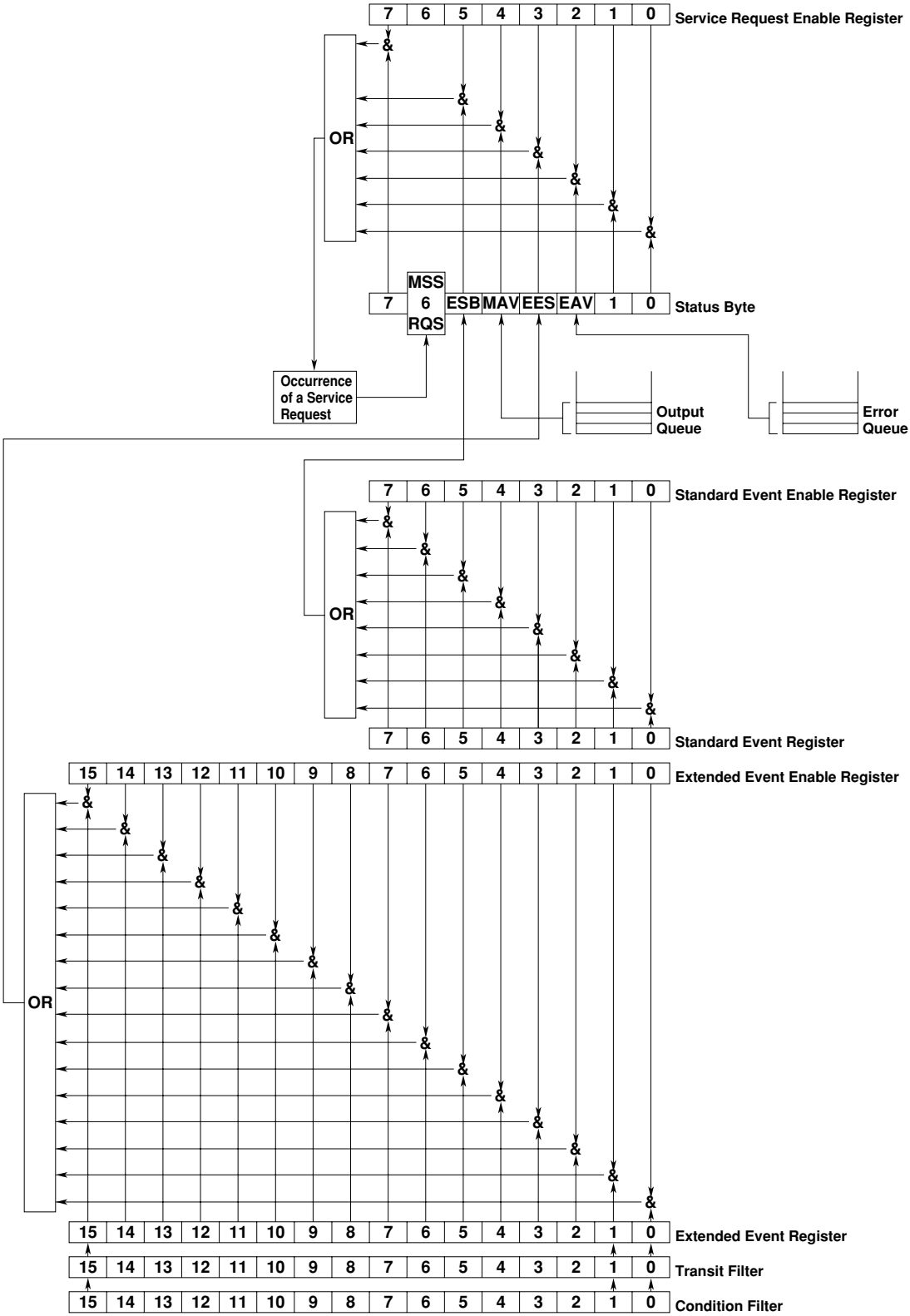
Syntax *WAI

Chapter 5 Status Report

5.1 Overview of the Status Report

Status Report

The figure below shows the status report that is read by serial polling. This status report is an extended version of the status report defined in IEEE 488.2-1992.



5.1 Overview of the Status Report/5.2 Status Byte

Overview of the Registers and Queues

Name	Function	Writing	Reading
Status byte		—	Serial polling (RQS), *STB?(MSS)
Service request enable register	Masks status byte	*SRE	*SRE?
Standard event register	Changes in device status	—	*ESR?
Standard event enable register	Masks standard event register	*ESE	*ESE?
Extended event register	Changes in device status	—	STATus: EESR?
Extended event enable register	Masks extended event register	STATus: EESE	STATus: EESE?
Condition register	Current instrument status	—	STATus: CONDition?
Transition filter	Conditions that change the extended event register	STATus: FILTER<x><x>?	STATus: FILTER
Output queue	Stores a response message to a query	All query commands	
Error queue	Stores the error No. and message	—	STATus: ERRor?

Registers and Queues That Affect the Status Byte

Registers that affect the bits of the status byte are shown below.

Standard event register:	Sets bit 5 (ESB) of the status byte to “1” or “0.”
Output queue:	Sets bit 4 (MAV) of the status byte to “1” or “0.”
Extended event register:	Sets bit 3 (EES) of the status byte to “1” or “0.”
Error queue:	Sets bit 2 (EAV) of the status byte to “1” or “0.”

Enable Registers

Registers that are used to mask a bit so that the bit will not affect the status byte even when it is set to 1, are shown below.

Status byte:	Mask the bits using the service request enable register.
Standard event register:	Mask the bits using the standard event enable register.
Extended event register:	Mask the bits using the extended event enable register.

Writing/Reading from Registers

The *ESE command is used to set the bits in the standard event enable register to 1's or 0's. The *ESE? command is used to query whether the bits in the standard event enable register are 1's or 0's. For details regarding these commands, see chapter 4.

5.2 Status Byte

Overview of Status Byte



- **Bits 0, 1, and 7**
Not used (always 0)
- **Bit 2 EAV (Error Available)**
Set to “1” when the error queue is not empty. In other words, this bit is set to “1” when an error occurs. See the page 5-5.
- **Bit 3 EES (Extend Event Summary Bit)**
Set to “1” when the logical product of the extended event register and the corresponding event register is “1.” —that is, when an event takes place inside the instrument. See the page 5-4.
- **Bit 4 MAV (Message Available)**
Set to “1” when the output queue is not empty. In other words, this bit is set to “1” when there are data to be transmitted. See the page 5-5.
- **Bit 5 ESB (Event Summary Bit)**
Set to “1” when the logical product of the standard event register and the corresponding event register is “1.” —that is, when an event takes place inside the instrument. See the page 5-3.
- **Bit 6 RQS (Request Service)/MSS (Master Status Summary)**
Set to “1” when the logical AND of the status byte excluding Bit 6 and the service request enable register is not “0.” In other words, this bit is set to “1” when the instrument is requesting service from the controller.
RQS is set to “1” when the MSS bit changes from “0” to “1,” and cleared when serial polling is carried out or when the MSS bit changes to “0.”

Bit Masking

To mask a bit in the status byte so that it does not cause an SRQ, set the corresponding bit of the service request enable register to “0.”

For example, to mask bit 2 (EAV) so that service is not requested when an error occurs, set bit 2 of the service request enable register to “0.” This can be done using the *SRE command. To query whether each bit of the service request enable register is “1” or “0,” use *SRE?. For details on the *SRE command, see chapter 4.

Operation of the Status Byte

A service request is issued when bit 6 of the status byte becomes “1.” Bit 6 is set to “1” when any of the other bits becomes a “1” (when the corresponding bit of the service request enable register is also set to “1”).

For example, if an event takes place and the logical OR of each bit of the standard event register and the corresponding bit in the enable register is “1,” bit 5 (ESB) will be set to “1.” In this case, if bit 5 of the service request enable register is “1,” bit 6 (MSS) will be set to “1,” thus requesting service from the controller.

It is also possible to check what type of event has occurred by reading the contents of the status byte.

Reading from the Status Byte

The following two methods are provided for reading the status byte.

- **Inquiry using the *STB? query**
Making an inquiry using the *STB? query sets bit 6 to MSS. This causes the MSS to be read. After completion of the read-out, none of the bits in the status byte will be cleared.
- **Serial polling**
Execution of a serial polling changes bit 6 to RQS. This causes RQS to be read. After completion of the read-out, only RQS is cleared. Using a serial polling, it is not possible to read MSS.

Clearing the Status Byte

No method is provided for forcibly clearing all the bits in the status byte. The bits that are cleared for each operation are shown below.

- **When an inquiry is made using the *STB? query**
No bit is cleared.
- **When a serial polling is executed**
Only the RQS bit is cleared.
- **When a *CLS command is received.**
When the *CLS command is received, the status byte itself is not cleared, but the contents of the standard event register (which affects the bits in the status byte) are cleared. As a result, the corresponding bits in the status byte are cleared, except bit 4 (MAV), since the output queue cannot be emptied by the *CLS command. However, the output queue will also be cleared if the *CLS command is received just after a program message terminator.

5.3 Standard Event Register

Overview of the Standard Event Register

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

- **Bit 7 PON (Power ON)**
Set to “1” when the power is turned ON.
- **Bit 6 URQ (User Request)**
Not used (always 0)
- **Bit 5 CME (Command Error)**
Set to “1” when the command syntax is incorrect.
Example Incorrectly spelled command name; “9” used in octal data.
- **Bit 4 EXE (Execution Error)**
Set to “1” when the command syntax is correct but the command cannot be executed in the current state.
Example Parameters are outside the range.
- **Bit 3 DDE (Device Dependent Error)**
Set to “1” when execution of the command is not possible due to an internal problem in the instrument that is not a command error or an execution error.
- **Bit 2 QYE (Query Error)**
Set to “1” if the output queue is empty or if the data is missing even after a query has been sent.
Example No response data; data is lost due to an overflow in the output queue.
- **Bit 1 RQC (Request Control)**
Not used (always 0)
- **Bit 0 OPC (Operation Complete)**
Set to “1” when the operation designated by the *OPC command (see chapter 4) has been completed.

Bit Masking

To mask a bit in the standard event register so that it does not cause bit 5 (ESB) of the status byte to change, set the corresponding bit in the standard event enable register to “0.” Refer to Chapter 4.

For example, to mask bit 2 (QYE) so that ESB will not be set to “1,” even if a query error occurs, set bit 2 of the standard event enable register to “0.” This can be done using the *ESE command. To inquire whether each bit of the standard event enable register is “1” or “0,” use the *ESE?. For details on the *ESE command, see chapter 4.

5.3 Standard Event Register/3.3 Responses

Operation of the Standard Event Register

The standard event register is provided for eight different kinds of event which can occur inside the instrument. Bit 5 (ESB) of the status byte is set to "1" when any of the bits in this register becomes "1" (or when the corresponding bit of the standard event enable register becomes "1").

Example

1. A query error occurs.
2. Bit 2 (QYE) is set to "1."
3. Bit 5 (ESB) of the status byte is set to "1" if bit 2 of the standard event enable register is "1."

It is also possible to check what type of event has occurred inside the instrument by reading the contents of the standard event register.

Reading from the Standard Event Register

The contents of the standard event register can be read by the *ESR command. After completion of the read-out, the register will be cleared.

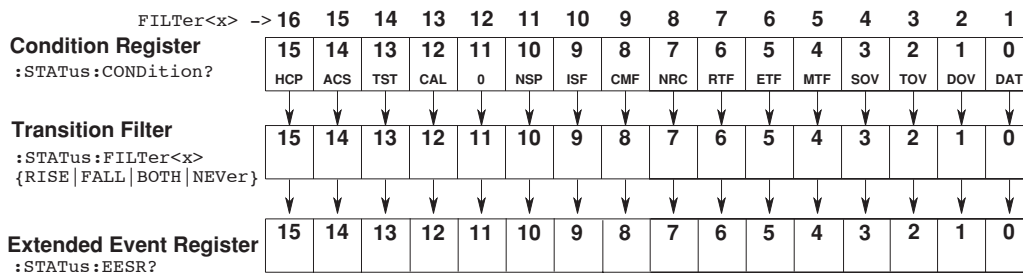
Clearing the Standard Event Register

The standard event register is cleared in the following three cases.

- When the contents of the standard event register are read using the *ESR command.
- When a *CLS command is received.
- When the instrument is power cycled.

5.4 Extended Event Register

Reading the extended event register tells you whether changes in the condition register (reflecting internal conditions) have occurred. A filter can be applied which allows you to decide which events are reported to the extended event register.



The meaning of each bit of the condition register is as follows.

Bit 0	DAT(DATA available)	Set to "1" when the measured data and the computation result are valid.
Bit 1	DOV(Data OVerflow)	Set to "1" when a measured data causes an overflow.
Bit 2	TOV(Time stamp OVerflow)	Set to "1" when a time stamp data causes an overflow.
Bit 3	SOV(Sample OVerflow)	Set to "1" when the number of measured samples causes an overflow.
Bit 4	MTF(Measured T Failure)	Set to "1" when the measurement of measured T fails.
Bit 5	ETF(Estimated T Failure)	Set to "1" when the measurement of estimated T fails.
Bit 6	RTF(Rest Time Failure)	Set to "1" when the pause time of block sampling falls short of 500 ns.
Bit 7	NRC(No Reference Clock)	Set to "1" when no reference signal is present.
Bit 8	CMF(Continuation Measurement Failure)	Set to "1" when continuous measurement fails.
Bit 9	ISF(ISI analysis Failure)	Set to "1" when inter-symbol interference analysis fails.
Bit 10	NSP(No Sync Pattern)	Set to "1" when simbol seach fails.
Bit 11	Reserved	Not used.
Bit 12	CAL(CALibration)	Set to "1" while calibration is in progress.
Bit 13	TST(TeSTing)	Set to "1" while self-test is in progress.
Bit 14	ACS(file ACceSsing)	Set to "1" while the storage medium is being accessed.
Bit 15	HCP(Hard-CoPYing)	Set to "1" while hard copy is in progress.

The transition filter parameters detect changes in the specified bit (numerical suffix, 1 to 16) of the condition register in the following manner and overwrite the extended event register.

RISE	The bit of the extended event register becomes "1" when the bit of the condition register changes from "0" to "1."
FALL	The bit of the extended event register becomes "1" when the bit of the condition register changes from "1" to "0."
BOTH	The bit of the extended event register becomes "1" when the bit of the condition register changes from "0" to "1," or from "1" to "0."
NEVer	Always 0.

5.5 Output Queue and Error Queue

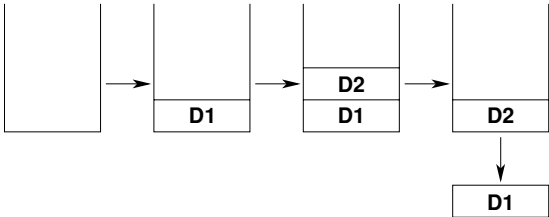
Overview of the Output Queue

The output queue is provided to store response messages to queries. For example, when the MEMory:SEND? query is sent to request output of the acquired waveform, the response data will be stored in the output queue until it is read out.

The example below shows that data is stored record by record in the output queue, and is read out oldest item first. The output queue is emptied in the following cases (in addition to when read-out is performed).

- When a new message is received from the controller.
- When a deadlock occurs (see page 3-2).
- When a device clear command (DCL or SDC) is received.
- When the instrument is power cycled.

The output queue cannot be emptied using the *CLS command. To see whether the output queue is empty or not, check bit 4 (MAV) of the status byte.



Overview of the Error Queue

The error queue stores the error No. and message when an error occurs. For example, if the controller sends an incorrect program message, the error number “113” and the message “Undefined header” are stored in the error queue when the error is displayed.

The contents of the error queue can be read using the STATus:ERRor? query. As with the output queue, the messages are read from the oldest ones first.

If an overflow occurs in the error queue, the final message will be replaced by message “350, “Queue overflow”.”

The error queue is emptied in the following cases (in addition to when read-out is performed).

- When a *CLS command is received.
- When the instrument is power cycled.

To see whether the error queue is empty or not, check bit 2 (EAV) of the status byte.

Chapter 6 Sample Programs

6.1 Before Programming

Environment

Model: IBM-AT compatible PC
 Language: Visual Basic Ver 6.0 Professional Edition or later.
 GPIB card: AT-GPIB/TNT IEEE-488.2 by National Instruments.

Setting Up Visual Basic

Standard modules used: Niglobal.bas
 Vbib-32.bas
 tmctl.bas
 tmval.bas

Note

Modules tmctl.bas and tmval.bas can be downloaded along with the sample programs from the YOKOGAWA Web site from the following URL.
<http://www.yokogawa.com/tm/Bu/software1.htm>

Setting Up the TA720

GP-IB

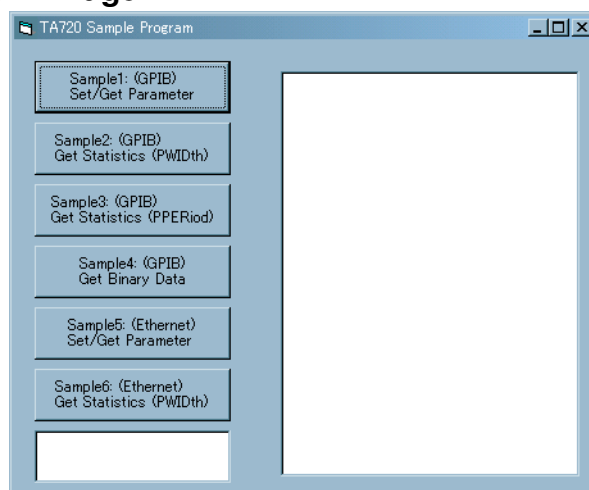
The sample programs given in this chapter use a GP-IB address of 1 for the TA720. Be sure to set the GP-IB address to 1 according to the procedures on page 1-4.

Ethernet

The sample programs given in this chapter use an IP address of 11.22.33.44, user name of anonymous, and no password.

Be sure to set the user name, password, timeout, and TCP/IP parameters according to the procedures on page 2-4.

6.2 Sample Program Image



6.3 Initialization, Error, and Functions for Execution

```

Option Explicit
Dim StartFlag As Integer      'Start Flag
Dim addr As Integer           'GPIB Address
Dim Timeout As Integer        'Timeout
Dim Dev As Integer            'Device ID(GPIB)
Dim eDev As Long              'Device ID(Ethernet)
Dim CtsFlag As Integer        'CTS Flag
Dim term As String            'Terminator
Dim Query(100) As String      'Query String
Dim Dummy As Integer

Private Function InitGpib() As Integer
    Dim eos As Integer         'EOS
    Dim eot As Integer         'EOI
    Dim brd As Integer         'GPIB Board ID
    Dim sts As Integer
    eos = &HCOA                'Terminator = LF
    eot = 1                    'EOI = Enable
    term = Chr(10)
    Timeout = T10s             'Timeout = 10s

    brd = ilfind("GPIB0")
    If (brd < 0) Then
        Call DisplayGPIBError(brd, "ilfind")
        InitGpib = 1
        Exit Function
    End If
    Dev = ildev(0, addr, 0, Timeout, eot, eos)
    If (Dev < 0) Then
        Call DisplayGPIBError(Dev, "ildev")
        InitGpib = 1
        Exit Function
    End If
    sts = ilsic(brd)            'Set IFC
    If (sts < 0) Then
        Call DisplayGPIBError(sts, "ilsic")
        InitGpib = 1
        Exit Function
    End If
    InitGpib = 0
End Function

Private Function InitEthernet() As Integer
    Dim ret As Long
    ret = TmInitialize(4, "11.22.33.44,anonymous,", eDev)
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmInitialize")
        InitEthernet = 1
        Exit Function
    End If

    ret = TmSetTerm(eDev, 2, 1)
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmSetTerm")
        InitEthernet = 1
        Exit Function
    End If

    ret = TmSetTimeout(eDev, 300)    'Timeout = 30s
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmSetTimeout")
        InitEthernet = 1
        Exit Function
    End If

    ret = TmSetRen(eDev, 1)
    If (ret <> 0) Then
        Call DisplayEthernetError(eDev, "TmSetRen")
        InitEthernet = 1
        Exit Function
    End If

    InitEthernet = 0
End Function

Private Sub DisplayGPIBError(ByVal sts As Integer, ByVal msg As String)
    Dim wrn As String
    Dim ers As String
    Dim ern As Integer

    If (sts And TIMO) Then
        wrn = "Time out" + Chr(13)
    Else
        wrn = ""
    End If
    If (sts And EERR) Then
        ern = iberr

```

```

    If (ern = EDVR) Then
        ers = "EDVR:System error"
    ElseIf (ern = ECIC) Then
        ers = "ECIC:Function requires GPIB board to be CIC"
    ElseIf (ern = ENOL) Then
        ers = "ENOL:No Listeners on the GPIB"
    ElseIf (ern = EADR) Then
        ers = "EADR:GPIB board not addressed correctly"
    ElseIf (ern = EARG) Then
        ers = "EARG:Invalid argument to function call"
    ElseIf (ern = ESAC) Then
        ers = "ESAC:GPIB board not System Controller as required"
    ElseIf (ern = EABO) Then
        ers = "EABO:I/O operation aborted(timeout)"
    ElseIf (ern = ENEB) Then
        ers = "ENEB:Nonexistent GPIB board"
    ElseIf (ern = EDMA) Then
        ers = "EDMA:DMA error"
    ElseIf (ern = EOIP) Then
        ers = "EOIP:I/O operation started before previous operation completed"
    ElseIf (ern = ECAP) Then
        ers = "ECAP:No capability for intended operation"
    ElseIf (ern = EFSO) Then
        ers = "EFSO:File system operation error"
    ElseIf (ern = EBUS) Then
        ers = "EBUS:GPIB bus error"
    ElseIf (ern = ESTB) Then
        ers = "ESTB:Serial poll status byte queue overflow"
    ElseIf (ern = ESRQ) Then
        ers = "ESRQ:SRQ remains asserted"
    ElseIf (ern = ETAB) Then
        ers = "ETAB:The return buffer is full"
    ElseIf (ern = ELCK) Then
        ers = "ELCK:Address or board is locked"
    Else
        ers = ""
    End If
Else
    ers = ""
End If
MsgBox ("Status No. " + Str(sts) + Chr(13) + wrn + "Error No. " + Str(ern) + Chr(13)
+ ers + Chr(13) + msg), vbExclamation, "Error!"
Call ibclr(Dev)
Call ibonl(Dev, 0)
Dev = -1
End Sub

Private Sub DisplayEthernetError(ByVal sts As Integer, ByVal msg As String)
    Dim wrn As String
    Dim ers As String
    Dim ern As Integer

    ern = TmGetLastError(eDev)

    If (ern = 2) Then
        ers = "Device not found"
    ElseIf (ern = 4) Then
        ers = "Connection to device failed"
    ElseIf (ern = 8) Then
        ers = "Device not connected"
    ElseIf (ern = 16) Then
        ers = "Device already connected"
    ElseIf (ern = 32) Then
        ers = "Incompatible PC"
    ElseIf (ern = 64) Then
        ers = "Illegal parameter"
    ElseIf (ern = 256) Then
        ers = "Send error"
    ElseIf (ern = 512) Then
        ers = "Receive error"
    ElseIf (ern = 1024) Then
        ers = "Received data not block data"
    ElseIf (ern = 4096) Then
        ers = "System error"
    ElseIf (ern = 8192) Then
        ers = "Illegal device ID"
    End If
    MsgBox ("Status No. " + Str(sts) + Chr(13) + wrn + "Error No. " + Str(ern) + Chr(13)
+ ers + Chr(13) + msg), vbExclamation, "Error!"
    'Call TmFinish(eDev)
    Dev = -1
End Sub

Private Sub Command1_Click()
    Dim sts As Integer
    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"

```

6.3 Initialization, Error, and Functions for Execution

```
List1.Clear
Dummy = DoEvents()
sts = SetParameter                                'Run Sample1 Set/Get Measure Parameter (GP-IB)
If (sts = 0) Then
    List1.AddItem Query(0)
    List1.AddItem Query(1)
End If
Text1.Text = "END"
StartFlag = 0
End Sub

Private Sub Command2_Click()
    Dim sts As Integer
    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetStatistics1                            'Run Sample2 GetStatistics 1 (GP-IB)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
        List1.AddItem Query(3)
        List1.AddItem Query(4)
        List1.AddItem Query(5)
        List1.AddItem Query(6)
        List1.AddItem Query(7)
        List1.AddItem Query(8)
        List1.AddItem Query(9)
        List1.AddItem Query(10)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub

Private Sub Command3_Click()
    Dim sts As Integer
    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetStatistics2                            'Run Sample3 Get Statistics 2 (GP-IB)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
        List1.AddItem Query(3)
        List1.AddItem Query(4)
        List1.AddItem Query(5)
        List1.AddItem Query(6)
        List1.AddItem Query(7)
        List1.AddItem Query(8)
        List1.AddItem Query(9)
        List1.AddItem Query(10)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub

Private Sub Command4_Click()
    Dim sts As Integer
    Dim i As Integer
    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetBinaryData                            'Run Sample4 Get Binary Data (GP-IB)
    If (sts = 0) Then
        For i = 0 To 99
            List1.AddItem Query(i)
        Next i
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub

Private Sub Command5_Click()
    Dim sts As Integer
    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
```

```

Text1.Text = "START"
List1.Clear
Dummy = DoEvents()
sts = SetParameterEthernet          'Run Sample5 Set/Get Measure Parameter (Ethernet)
If (sts = 0) Then
    List1.AddItem Query(0)
    List1.AddItem Query(1)
End If
Text1.Text = "END"
StartFlag = 0
End Sub

Private Sub Command6_Click()
    Dim sts As Integer
    If (StartFlag = 1) Then
        Exit Sub
    End If
    StartFlag = 1
    Text1.Text = "START"
    List1.Clear
    Dummy = DoEvents()
    sts = GetStatisticsEthernet      'Run Sample6 GetStatistics (Ethernet)
    If (sts = 0) Then
        List1.AddItem Query(0)
        List1.AddItem Query(1)
        List1.AddItem Query(2)
        List1.AddItem Query(3)
        List1.AddItem Query(4)
        List1.AddItem Query(5)
        List1.AddItem Query(6)
        List1.AddItem Query(7)
        List1.AddItem Query(8)
        List1.AddItem Query(9)
        List1.AddItem Query(10)
    End If
    Text1.Text = "END"
    StartFlag = 0
End Sub

Private Sub Form_Load()
    StartFlag = 0                  'Clear Start Flag
    Dev = -1                      'Clear device id
    addr = 1                      'GPIO Address = 1
    Command1.Caption = "Sample1: (GPIO)" + Chr(13) + "Set/Get Parameter"
    Command2.Caption = "Sample2: (GPIO)" + Chr(13) + "Get Statistics (PWIDth)"
    Command3.Caption = "Sample3: (GPIO)" + Chr(13) + "Get Statistics (PPERiod)"
    Command4.Caption = "Sample4: (GPIO)" + Chr(13) + "Get Binary Data"
    Command5.Caption = "Sample5: (Ethernet)" + Chr(13) + "Set/Get Parameter"
    Command6.Caption = "Sample6: (Ethernet)" + Chr(13) + "Get Statistics (PWIDth)"
    Text1.Text = ""
End Sub

```

6.4 Setting Measurement Parameters or Querying the Settings (for GP-IB)

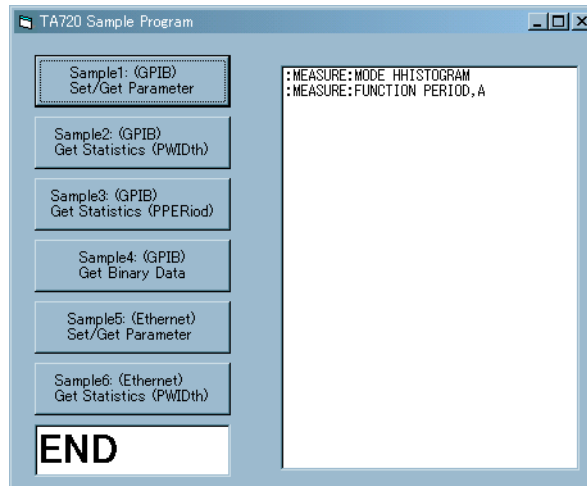
```

Sample1 (GP-IB)
Private Function SetParameter() As Integer
    Dim msg As String
    Dim qry As String
    Dim sts As Integer
    msg = Space$(100)
    qry = Space$(100)

    sts = InitGpib
    If (sts <> 0) Then
        SetParameter = 1
        Exit Function
    End If
    msg = "*RST" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    msg = ":COMMUNICATE:HEADER ON" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    msg = ":COMM:VERBOSE ON" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    msg = ":MEASURE:MODE HHISTOGRAM" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    msg = ":MEASURE:MODE?" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    Query(0) = Left$(qry, ibcntl - 1)
    msg = ":MEASURE:FUNCTION PERIOD,A" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    msg = ":MEASURE:FUNCTION?" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        SetParameter = 1
        Exit Function
    End If
    Query(1) = Left$(qry, ibcntl - 1)

    Call ibonl(Dev, 0)
    SetParameter = 0
End Function

```



6.5 Querying the Measured Statistical Values of Period Measurement (for GP-IB)

```

Sample2(GPIB)  Get Statistics
Private Function GetStatistics1() As Integer
    Dim msg As String          'Command buffer
    Dim qry As String          'Query buffer
    Dim sts As Integer
    Dim winlabel(10) As String 'Widow Label
    Dim i As Integer
    Dim temp As String

    msg = Space$(100)
    qry = Space$(100)

    sts = InitGpib              'Initialize GPIB
    If (sts <> 0) Then
        GetStatistics1 = 1
        Exit Function
    End If
    msg = "*RST" + term         'Initialize Setting Parameter
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    msg = ":MEAS:MODE TST" + term 'Mode = TimeStamp
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    msg = ":MEAS:FUNC PWID,A" + term 'Function = PulseWidth Ch = A
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    msg = ":SAMP:GATE:MODE EVEN;EVEN 100000" + term 'Gate = Event; EventSize 100000
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    msg = ":THIS:MODE MULT" + term 'WindowMode = MULTI WINDOW
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    msg = ":THIS:MULT:SIZE 10;TVAL 37ns" + term 'WINDOW SIZE 10; T=37ns
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    winlabel(0) = "3": winlabel(1) = "4": winlabel(2) = "5"
    winlabel(3) = "6": winlabel(4) = "7": winlabel(5) = "8"
    winlabel(6) = "9": winlabel(7) = "10": winlabel(8) = "11"
    winlabel(9) = "14"
    For i = 1 To 10
        msg = ":THIS:MULT:WIND" + LTrim(Str(i)) + ":LAB " + winlabel(i - 1) + term
        sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
        If (sts < 0) Then
            Call DisplayGPIBError(sts, msg)
            GetStatistics1 = 1
            Exit Function
        End If
    Next i
    msg = ":THIS:MULT:UPD" + term 'UPDATE Window Scale
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    msg = "CALC:PAR:AVER ON;SDEV ON" + term 'Calc/Stat On
    sts = ilwrt(Dev, msg, Len(msg)) 'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
End Function

```

6.5 Querying the Measured Statistical Values of Period Measurement (for GP-IB)

```

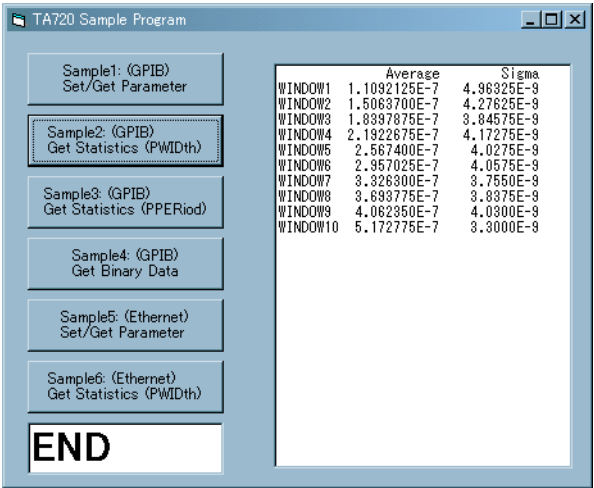
msg = "STAT:FILT1 RISE" + term           'Filter1 Rise(Data Available)
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If
msg = "STAT:EESR?" + term                'Clear Extended Event Register
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))           'Receive Query
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If
msg = "SST" + term                       'Single Measure Start
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If
msg = "COMM:WAIT 1" + term               'Wait until data available
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If
msg = ":COMM:HEAD OFF" + term            'Header = OFF
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If
Query(0) = Space$(14) + "Average" + Space$(8) + "Sigma"
For i = 1 To 10
    temp = LTrim(Str(i))
    Query(i) = "WINDOW" + temp + Space$(2 - Len(temp))
    'Get Average value
    msg = ":CALC:WIND" + LTrim(Str(i)) + ":AVER?" + term
    sts = ilwrt(Dev, msg, Len(msg))     'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))       'Receive Query
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    temp = Left$(qry, ibcntl - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
    'Get Standard Deviation value
    msg = ":CALC:WIND" + LTrim(Str(i)) + ":SDEV?" + term
    sts = ilwrt(Dev, msg, Len(msg))     'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))       'Receive Query
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics1 = 1
        Exit Function
    End If
    temp = Left$(qry, ibcntl - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
Next i

msg = ":COMM:HEAD ON" + term            'Header = ON
sts = ilwrt(Dev, msg, Len(msg))         'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics1 = 1
    Exit Function
End If

Call ibonl(Dev, 0)
GetStatistics1 = 0
End Function

```

6.5 Querying the Measured Statistical Values of Period Measurement (for GP-IB)



6.6 Querying the Measured Statistical Values of Period A & Period B Measurement (for GP-IB)

Sample3(GPIB) Get Statistics

```

Private Function GetStatistics2() As Integer
    Dim msg As String
    Dim qry As String
    Dim sts As Integer
    Dim winlabel(10) As String
    Dim i As Integer
    Dim temp As String

    msg = Space$(100)
    qry = Space$(100)

    sts = InitGpib
    If (sts <> 0) Then
        GetStatistics2 = 1
        Exit Function
    End If
    msg = "*RST" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If

    msg = ":MEASURE:MODE HHISTOGRAM" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    msg = ":MEASURE:FUNCTION PPERIOD,AB" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If

    msg = ":SAMPLE:GATE:MODE EVENT;EVENTSIZE 10000" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    msg = ":HHISTOGRAM:MODE MULTI" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    msg = ":HHISTOGRAM:MULTI:SIZE 10" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    msg = ":HHISTOGRAM1:MULTI:TVALUE 37ns" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    msg = ":HHISTOGRAM2:MULTI:TVALUE 37ns" + term
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPSError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If

    winlabel(0) = "3": winlabel(1) = "4": winlabel(2) = "5"
    winlabel(3) = "6": winlabel(4) = "7": winlabel(5) = "8"
    winlabel(6) = "9": winlabel(7) = "10": winlabel(8) = "11"
    winlabel(9) = "14"
    For i = 1 To 10
        msg = ":HHISTOGRAM:MULTI:WINDOW" + LTrim(Str(i)) _
            + ":LABEL " + winlabel(i - 1) + term
        sts = ilwrt(Dev, msg, Len(msg))
        If (sts < 0) Then
            Call DisplayGPSError(sts, msg)
            GetStatistics2 = 1
            Exit Function
        End If
    Next i
    GetStatistics2 = 1
    Exit Function
End Function

```

6.6 Querying the Measured Statistical Values of Period A & Period B Measurement (for GP-IB)

```
        Call DisplayGPIBError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
Next i
msg = ":HHISTOGRAM:MULTI:UPDATE" + term      'UPDATE Window Scale
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
msg = "CALCULATION:PARAMETER:AVERAGE ON" + term 'Calc/Stat On
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If

msg = "STATUS:FILTER1 RISE" + term            'Filter1 Rise(Data Available)
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
msg = "STATUS:EESR?" + term                  'Clear Extended Event Register
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))                'Receive Query
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
msg = "SSTART" + term                        'Single Measure Start
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
msg = "COMMUNICATE:WAIT 1" + term            'Wait until data available
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
msg = ":COMMUNICATE:HEADER OFF" + term       'Header = OFF
sts = ilwrt(Dev, msg, Len(msg))              'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If

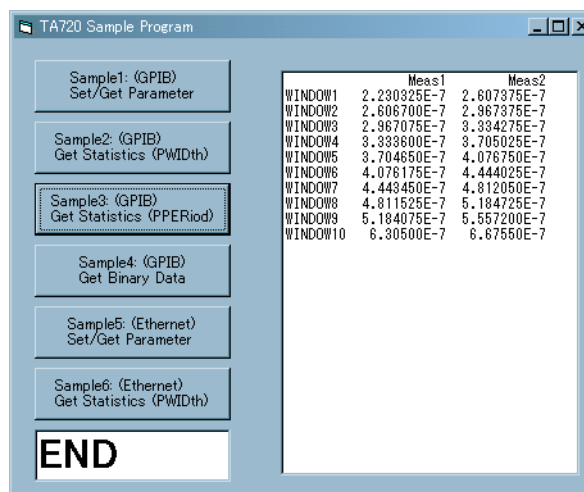
Query(0) = Space$(16) + "Meas1" + Space$(8) + "Meas2"
For i = 1 To 10
    temp = LTrim(Str(i))
    Query(i) = "WINDOW" + temp + Space$(2 - Len(temp))
    'Get Average value
    msg = ":CALCULATION:MEAS1:WINDOW" + LTrim(Str(i)) _
        + ":AVERAGE?" + term
    sts = ilwrt(Dev, msg, Len(msg))          'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))          'Receive Query
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics2 = 1
        Exit Function
    End If
    temp = Left$(qry, ibcntl - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
    'Get Standard Deviation value
    msg = ":CALCULATION:MEAS2:WINDOW" + LTrim(Str(i)) _
        + ":AVERAGE?" + term
    sts = ilwrt(Dev, msg, Len(msg))          'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetStatistics2 = 1
    End If
End For
```

6.6 Querying the Measured Statistical Values of Period A & Period B Measurement (for GP-IB)

```
Exit Function
End If
sts = ilrd(Dev, qry, Len(qry))           'Receive Query
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If
temp = Left$(qry, ibcntl - 1)
Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
Next i

msg = ":COMMUNICATE:HEADER ON" + term    'Header = ON
sts = ilwrt(Dev, msg, Len(msg))          'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatistics2 = 1
    Exit Function
End If

Call ibonl(Dev, 0)
GetStatistics2 = 0
End Function
```



6.7 Querying the Measured Data of A-to-B Time Interval Measurement (for GP-IB)

Sample4 (GP-IB)

```

Private Function GetBinaryData() As Integer
    Dim msg As String
    Dim qry As String
    Dim sts As Integer
    Dim idata(1000) As Integer
    Dim num As Integer
    Dim data_h As Long
    Dim data_l As Long
    Dim data As Variant
    Dim eos As Integer
    Dim i As Integer

    'Command buffer
    'Query buffer

    'Data buffer(Binary)

    'EOS

    msg = Space$(100)
    qry = Space$(1000)

    sts = InitGpib
    'Initialize GPIB
    If (sts <> 0) Then
        GetBinaryData = 1
        Exit Function
    End If
    msg = "*RST" + term
    'Initialize Setting Parameter
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = ":MEASURE:MODE TSTAMP" + term
    'Mode = Time Stamp
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = ":MEASURE:FUNCTION TI,AB" + term
    'Function = Time Interval, Ch = A,B
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = ":SAMPLE:GATE:MODE EVENT;EVENTSIZE 100" + term
    'Gate = Event; EventSize 100
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = "STATUS:FILTER1 RISE" + term
    'Filter1 Rise(Data Available)
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = "STATUS:EESR?" + term
    'Clear Extended Event Register
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    sts = ilrd(Dev, qry, Len(qry))
    'Receive Query
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = "SSTART" + term
    'Single Measure Start
    sts = ilwrt(Dev, msg, Len(msg))
    'Send Command
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If
    msg = "COMMUNICATE:WAIT 1" + term
    'Wait until data available
    sts = ilwrt(Dev, msg, Len(msg))
    If (sts < 0) Then
        Call DisplayGPIBError(sts, msg)
        GetBinaryData = 1
        Exit Function
    End If

    msg = "MEMORY:DASELECT MEASUREDATA" + term
    'Send Command
    sts = ilwrt(Dev, msg, Len(msg))

```

6.7 Querying the Measured Data of A-to-B Time Interval Measurement (for GP-IB)

```

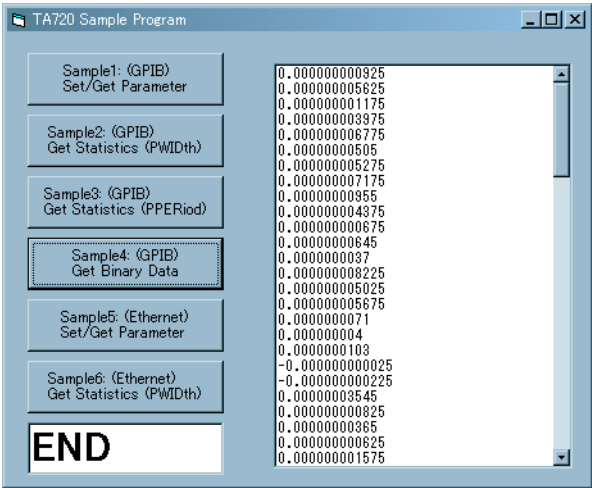
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
msg = "MEMORY:FORMAT BINARY" + term           'Data Format = Binary
sts = ilwrt(Dev, msg, Len(msg))                'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
msg = "MEMORY:BYTEORDER LSBFIRST" + term       'Data Byte order = LSB First(for
Little Endian)
sts = ilwrt(Dev, msg, Len(msg))                'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
msg = "MEMORY:START 1;END 100" + term          'START 1, END 100
sts = ilwrt(Dev, msg, Len(msg))                'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If

eos = 0
sts = ileos(Dev, eos)                          'Terminator = None(for Binary Data)
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
msg = "MEMORY:SEND?" + term
sts = ilwrt(Dev, msg, Len(msg))                'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
sts = ilrd(Dev, qry, 10)                       'Receive Header(#8*****#)
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
num = Val(Mid$(qry, 3, 8)) / 4
sts = ilrdi(Dev, idata(), 1000)                'Receive Binary Data + LF
For i = 0 To num - 1
    data_h = idata(i * 2 + 1)
    data_l = idata(i * 2)
    If data_h >= 0 And data_l >= 0 Then
        data = (data_h * 65536#) + data_l
    ElseIf data_h >= 0 And data_l < 0 Then
        data = (data_h * 65536#) + (data_l + 65536#)
    ElseIf data_h < 0 And data_l >= 0 Then
        data = (data_h + 65536#) * 65536# + data_l
        data = data - (65536# * 65536#)
    Else ' data_h < 0 And data_l < 0
        data = (data_h + 65536#) * 65536# + (data_l + 65536#)
        data = data - (65536# * 65536#)
    End If
    Query(i) = Format(data * 0.000000000025)
Next i

eos = &HC0A
sts = ileos(Dev, eos)                          'Terminator = LF
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetBinaryData = 1
    Exit Function
End If
Call ibonl(Dev, 0)
GetBinaryData = 0
End Function

```

6.7 Querying the Measured Data of A-to-B Time Interval Measurement (for GP-IB)



6.8 Setting Measurement Parameters or Querying the Settings (for Ethernet)

```

Sample5(Ethernet)
Private Function SetParameterEthernet() As Integer
    Dim msg As String
    Dim qry As String
    Dim sts As Integer
    Dim length As Long
    msg = Space$(100)
    qry = Space$(100)

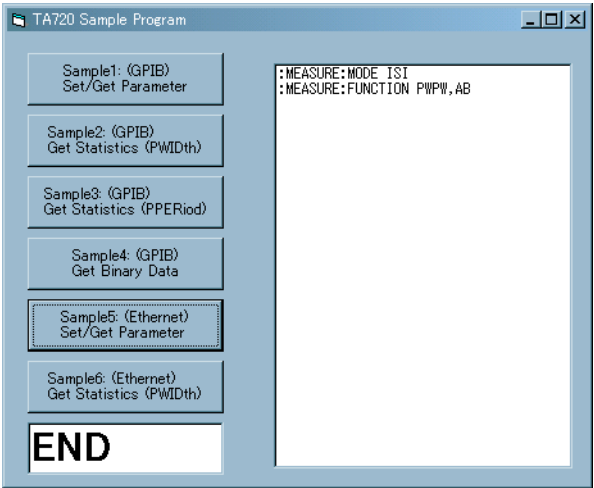
    sts = InitEthernet
    If (sts <> 0) Then
        SetParameterEthernet = 1
        Exit Function
    End If
    msg = "*RST" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    msg = ":COMMUNICATE:HEADER ON" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    msg = ":COMMUNICATE:VERBOSE ON" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    msg = ":MEASURE:MODE ISI" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    msg = ":MEASURE:MODE?" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    sts = TmReceive(eDev, qry, 1000, length)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    Query(0) = Left$(qry, length - 1)

    msg = ":MEASURE:FUNCTION PWPW,AB" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    msg = ":MEASURE:FUNCTION?" + term
    sts = TmSend(eDev, msg)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    sts = TmReceive(eDev, qry, 1000, length)
    If (sts <> 0) Then
        Call DisplayEthernetError(sts, msg)
        SetParameterEthernet = 1
        Exit Function
    End If
    Query(1) = Left$(qry, length - 1)

    Call TmFinish(eDev)
    SetParameterEthernet = 0
End Function

```

6.8 Setting Measurement Parameters or Querying the Settings (for Ethernet)



6.9 Querying the Measured Statistical Values of Pulse Width Measurement (for Ethernet)

Sample6(Ethernet)

```

Private Function GetStatisticsEthernet() As Integer
    Dim msg As String
    Dim qry As String
    Dim sts As Integer
    Dim length As Long
    Dim winlabel(10) As String
    Dim i As Integer
    Dim temp As String

    msg = Space$(100)
    qry = Space$(100)

    sts = InitEthernet
    If (sts <> 0) Then
        GetStatisticsEthernet = 1
        Exit Function
    End If
    msg = "*RST" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If

    msg = ":MEAS:MODE HHIS" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    msg = ":MEAS:FUNC PWID,A" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    msg = ":SAMP:GATE:MODE EVEN;EVEN 10000" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    msg = ":HHIS:MODE MULT" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    msg = ":HHIS:MULT:SIZE 10;TVAL 37ns" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If

    winlabel(0) = "3": winlabel(1) = "4": winlabel(2) = "5"
    winlabel(3) = "6": winlabel(4) = "7": winlabel(5) = "8"
    winlabel(6) = "9": winlabel(7) = "10": winlabel(8) = "11"
    winlabel(9) = "14"
    For i = 1 To 10
        msg = ":HHIS:MULT:WIND" + LTrim(Str(i)) + ":LAB " + winlabel(i - 1) + term
        sts = TmSend(eDev, msg)
        If (sts < 0) Then
            Call DisplayEthernetError(sts, msg)
            GetStatisticsEthernet = 1
            Exit Function
        End If
    Next i
    msg = ":HHIS:MULT:UPD" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    msg = "CALC:PAR:AVER ON;SDEV ON" + term
    sts = TmSend(eDev, msg)
    If (sts < 0) Then

```

6.9 Querying the Measured Statistical Values of Pulse Width Measurement (for Ethernet)

```
Call DisplayEthernetError(sts, msg)
GetStatisticsEthernet = 1
Exit Function
End If

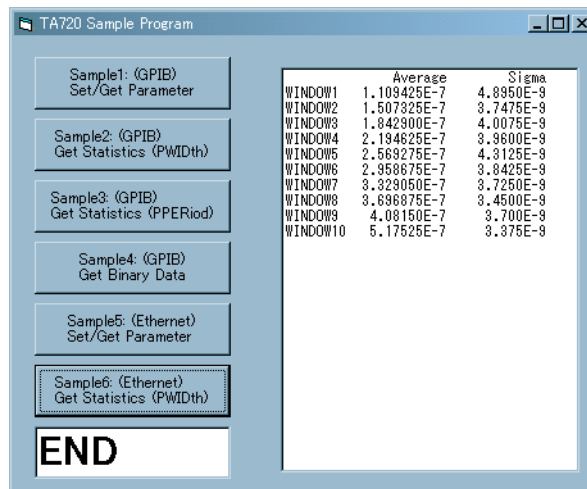
msg = "STAT:FILT1 RISE" + term           'Filter1 Rise(Data Available)
sts = TmSend(eDev, msg)                  'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = "STAT:EESR?" + term                'Clear Extended Event Register
sts = TmSend(eDev, msg)                  'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
sts = TmReceive(eDev, qry, 1000, length) 'Receive Query
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = "SST" + term                       'Single Measure Start
sts = TmSend(eDev, msg)                  'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = "COMM:WAIT 1" + term               'Wait until data available
sts = TmSend(eDev, msg)                  'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
msg = ":COMM:HEAD OFF" + term            'Header = OFF
sts = TmSend(eDev, msg)                  'Send Command
If (sts < 0) Then
    Call DisplayEthernetError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If

Query(0) = Space$(14) + "Average" + Space$(8) + "Sigma"
For i = 1 To 10
    temp = LTrim(Str(i))
    Query(i) = "WINDOW" + temp + Space$(2 - Len(temp))
    'Get Average value
    msg = ":CALC:WIND" + LTrim(Str(i)) + ":AVER?" + term
    sts = TmSend(eDev, msg)              'Send Command
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    sts = TmReceive(eDev, qry, 1000, length) 'Receive Query
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    temp = Left$(qry, length - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
    'Get Standard Deviation value
    msg = ":CALC:WIND" + LTrim(Str(i)) + ":SDEV?" + term
    sts = TmSend(eDev, msg)              'Send Command
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    sts = TmReceive(eDev, qry, 1000, length) 'Receive Query
    If (sts < 0) Then
        Call DisplayEthernetError(sts, msg)
        GetStatisticsEthernet = 1
        Exit Function
    End If
    temp = Left$(qry, length - 1)
    Query(i) = Query(i) + Space$(13 - Len(temp)) + temp
Next i

msg = ":COMM:HEAD ON" + term             'Header = ON
sts = TmSend(eDev, msg)                  'Send Command
If (sts < 0) Then
    Call DisplayGPIBError(sts, msg)
    GetStatisticsEthernet = 1
    Exit Function
End If
```

6.9 Querying the Measured Statistical Values of Pulse Width Measurement (for Ethernet)

```
End If  
  
Call TmFinish(eDev)  
GetStatisticsEthernet = 0  
End Function
```



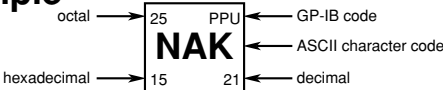
Appendix

Appendix 1 ASCII Character Codes

The following table shows the ASCII character codes.

	0	1	2	3	4	5	6	7
0	0 NUL	20 DEL	40 SP	60 0	100 @	120 P	140 '	160 p
1	1 SOH	21 DC1	41 !	61 1	101 A	121 Q	141 a	161 q
2	2 STX	22 DC2	42 "	62 2	102 B	122 R	142 b	162 r
3	3 ETX	23 DC3	43 #	63 3	103 C	123 S	143 c	163 s
4	4 EOT	24 DC4	44 \$	64 4	104 D	124 T	144 d	164 t
5	5 ENQ	25 NAK	45 %	65 5	105 E	125 U	145 e	165 u
6	6 ACK	26 SYN	46 &	66 6	106 F	126 V	146 f	166 v
7	7 BEL	27 ETB	47 ,	67 7	107 G	127 W	147 g	167 w
8	10 BS	30 CAN	50 (	70 8	110 H	130 X	150 h	170 x
9	11 HT	31 EM	51)	71 9	111 I	131 Y	151 i	171 y
A	12 LF	32 SUB	52 *	72 :	112 J	132 Z	152 j	172 z
B	13 VT	33 ESC	53 +	73 ;	113 K	133 [	153 k	173 {
C	14 FF	34 FS	54 ,	74 <	114 L	134 \ I	154 l	174
D	15 CR	35 GS	55 -	75 =	115 M	135] m	155 M	175 }
E	16 SO	36 RS	56 .	76 >	116 N	136 ^	156 n	176 ~
F	17 SI	37 US	57 /	77 ?	117 O	137 _	157 o	177 DEL (RUBOUT)
	Address Command	Universal Command	Listener Address		Talker Address		Secondary Command	

Example



Appendix 2 Error Messages

This section describes the error messages related to communications.

- If servicing is required, please contact your nearest YOKOGAWA dealer as listed on the back cover of this manual.
- For other error messages, see the *TA720 User's Manual IM704510-01E*.

Error in communication command (100-199)

Code	Message	Error Description	Page
102	Syntax error	Invalid syntax.	Chapters 3, 4
103	Invalid separator	Use a comma to separate the data.	3-1
104	Data type error	See pages 3-5 and 3-6 and enter using the correct data format.	3-5 to 3-6
108	Parameter not allowed	Check the number of parameters.	3-5, chapter 4
109	Missing parameter	Enter required parameters.	3-5, chapter 4
111	Header separator error	Insert a space between the header and data to separate them.	3-2
112	Program mnemonic too long	Check the mnemonic (alphanumeric character string).	Chapter 4
113	Undefined header	Check the header.	Chapter 4
114	Header suffix out of range	Check the header.	Chapter 4
120	Numeric data error	A mantissa is required before the exponent in the <NRf> form.	3-5
123	Exponent too large	Use a smaller exponent for <NR3> format.	3-5, chapter 4
124	Too many digits	Limit the number of digits to 255 or less.	3-5, chapter 4
128	Numeric data not allowed	Enter in a format other than <NRf> format.	3-5, chapter 4
131	Invalid suffix	Check the unit of the <Voltage>, <Time>, and <Frequency>.	3-5
134	Suffix too long	Check the unit of the <Voltage>, <Time>, and <Frequency>.	3-5
138	Suffix not allowed	No units are allowed other than <Voltage>, <Time>, and <Frequency>.	3-5
141	Invalid character data	Enter one of the character strings in {... }.	Chapter 4
144	Character data too long	Check the character strings in {... }.	Chapter 4
148	Character data not allowed	Enter in a format other than in {... }.	Chapter 4
150	String data error	Enclose <Character string> in double quotation or single quotation marks.	3-6
151	Invalid string data	<Character string> is too long or contains characters which cannot be used..	Chapter 4
158	String data not allowed	Enter in a data format other than <Character string>.	Chapter 4
161	Invalid block data	<Block data> is not allowed.	3-6, chapter 4
168	Block data not allowed	<Block data> is not allowed.	3-6, chapter 4
171	Invalid expression	Equation is not allowed	Chapter 4
178	Expression data not allowed	Equation is not allowed	Chapter 4
181	Invalid outside macro definition	Macro functions defined in IEEE488.2 are not supported.	—

Error in communication execution (200-299)

Code	Message	Error Description	Page
221	Setting conflict	Check the relevant settings.	Chapter 4
222	Data out of range	Check the range.	Chapter 4
223	Too much data	Check the data byte length.	Chapter 4
224	Illegal parameter value	Check the range.	Chapter 4
241	Hardware missing	Check the existence of options.	–
260	Expression error	Equation is not allowed	–
270	Macro error	Macro functions defined in IEEE488.2 are not supported.	–
272	Macro execution error	Macro functions defined in IEEE488.2 are not supported.	–
273	Illegal macro label	Macro functions defined in IEEE488.2 are not supported.	–
275	Macro definition too long	Macro functions defined in IEEE488.2 are not supported.	–
276	Macro recursion error	Macro functions defined in IEEE488.2 are not supported.	–
277	Macro redefinition not allowed	Macro functions defined in IEEE488.2 are not supported.	–
278	Macro header not found	Macro functions defined in IEEE488.2 are not supported.	–

Error in communication Query (400-499)

Code	Message	Error Description	Page
410	Query INTERRUPTED	Check transmission/reception order.	3-2
420	Query UNTERMINATED	Check transmission/reception order.	3-2
430	Query DEADLOCKED	Aborting transmission. Limit the length of the program message including <PMT> to 1024 bytes or less.	3-2
440	Query UNTERMINATED after indefinite response	Do not enter any query after *IDN? and *OPT?.	–

Error in Execution (600-899) (See the TA720 User's Manual)**Error in System Operation (912-915)**

Code	Message	Error Description	Page
912	Fatal error in Communication-driver	Servicing required.	–
914	A communication time-out has occurred.	Receive data within time-out time. The communication line may be faulty.	–

Other errors (350)

Code	Message	Corrective Action	Page
350	Queue overflow	Read the error queue.	5-5

Note

Code 350 indicates overflow of error queue. This code is returned as a response to the "STATus:ERRor?" query; it does not appear on the screen.

Appendix 3 Overview of IEEE.488.2-1992

The GP-IB interface provided with the TA720 conforms to the IEEE 488.3-1992 Standard. This standard specifies that the following 23 points be stated in the document. This section describes these points.

- 1 Subsets supported by IEEE 488.1 interface functions**
See "GP-IB Interface Specifications" on page 1-3.
- 2 Operation of device when the device is assigned to an address other than addresses 0 to 30**
The TA720 does not allow assignment to an address other than 0 to 30.
- 3 Reaction when the user changes the address**
The current address is changed when a new address is set using the [GP-IB] menu of the UTILITY key. The newly set address is valid until another new address is set.
- 4 Device set-up at power ON. Commands which can be used at power ON**
Basically, the previous settings (i.e. the settings which were valid when power was turned OFF) are valid.
All commands are available at power ON.
- 5 Message transmission options**
 - a Input buffer size**
1024 bytes.
 - b Queries that return multiple response messages**
See the example of the commands given in chapter 4.
 - c Queries which generate response data during analysis of the syntax**
Every query generates a response data when analysis of the syntax is completed.
 - d Queries which generate response data during reception**
No query generates response data when the query is received by the controller.
 - e Commands consisting of parameters which restrict one other**
See the example of the commands given in chapter 4.
- 6 Options included in command function elements and composite header elements**
See chapters 3 and 4.
- 7 Buffer size which affects transmission of block data**
The receive buffer size of block data is 64 KB.
- 8 List of program data elements which can be used in equations, and nesting limit**
No equations can be used.
- 9 Syntax of response to queries**
See the example of the commands given in chapter 4.
- 10 Communications between devices which do not follow the response syntax**
None.
- 11 Size of data block of response data**
0 to 4096000 bytes
- 12 List of supported common commands**
See section 4.23, "Common Command Group."
- 13 Condition of device when calibration is successfully completed**
The settings return to the conditions that existed before the calibration, measurements are terminated, and previous measured data are invalidated.
- 14 Maximum length of block data which can be used for definition of *DDT trigger macro**
Not supported.
- 15 Maximum length of macro label used in definition of macro, maximum length of block data which can be used for definition of macro, processing when recursion is used in definition of macro**
Macro functions are not supported.
- 16 Response to *IDN?**
See section 4.23, "Common Command Group."
- 17 Size of storage area for protected user data for PUD and *PUD?**
*PUD and *PUD? are not supported.
- 18 Length of *RDT and *RDT? resource name**
*RDT and *RDT? are not supported.
- 19 Change in status due to *RST, *LRN?, *RCL and *SAV**
*RST
See section 4.23, "Common Command Group."
*LRN?, *RCL, *SAV
These common commands are not supported.
- 20 Execution range of self-test using the *TST?**
This command executes the same Board test as the Selftest menu of the UTILITY key.
- 21 Structure of extended return status**
See chapter 4.
- 22 To find out whether each command is performed in parallel or sequentially**
See section 3.5, "Synchronization with the Controller" and chapter 4.
- 23 Description of execution of each command**
See the functions of each command in chapter 4 and TA720 User's Manual.

Index

Symbols

<Character data>	3-6
<Frequency>	3-5
<Multiplier>	3-5
<NRf>	3-5
<Percent>	3-5
<Register>	3-6
<Time>	3-5
<Unit>	3-5
<Voltage>	3-5

A

abbreviated form	3-4
Account	2-4
address	1-5
address, selection of	1-5
arming	4-53, 4-61
arming delay	4-61, 4-62
arming delay mode	4-62
arming level	4-53
arming slope	4-62
arming source	4-62
auto naming	4-32
auto scale	4-8
auto window	4-38, 4-47

B

backlight, ON/OFF of	4-68
beep sound	4-67
binary data	4-57
binary measurement data	4-32
block	4-28, 4-57
block data	3-6
block sampling	4-62
block sampling rest	4-62
block sampling rest time	4-63
blocks, the number of	4-57, 4-63
Boolean	3-6
BOTH Graph	4-28
brightness	4-68

C

Character string data	3-6
click sound	4-68
comment, setting and querying of	4-33, 4-35
communication status	4-65
communications, settings related to	4-25
compression, ON/OFF of	4-35
computation range	4-10
condition register	4-65, 5-4
constant T	4-15, 4-38, 4-39, 4-40, 4-42
constant T on the auto window	4-47
constant T using the frequency format	4-48
constant T value	4-43, 4-49

D

data	3-5
data points, number of	4-59
data recall	4-32
data save	4-32
date	4-68
deadlock	3-2
directory, creation of	4-32
display	4-28
display format	4-29
display size	4-29
dot	4-28
dot connect	4-28
drive	4-32, 4-35

E

error queue	4-65, 5-5
estimated T	4-11
events, the number of	4-63
extended event	4-26
extended event enable register	4-65
extended event register	4-65, 5-4
external gate	4-53

F

file	4-31
file, deletion of	4-32
format	4-32
free disk space	4-32

G

gate	4-63, 4-64
gate level	4-53
gate time	4-64
GP-IB cable	1-2
GP-IB connector	1-1
GP-IB interface specifications	1-4
graph parameter	4-28
graph setting	4-29
group	3-3

H

hardware histogram	4-38
header	3-3, 3-4, 4-25
histogram, statistical computation of	4-16, 4-22
horizontal axis (X-axis) of the window	4-49

I

Infinite	2-5
inhibit	4-54, 4-64
inhibit level	4-54
input condition	4-52
input coupling	4-53
input impedance	4-53
inter-symbol interference analysis mode	4-47
interface messages	1-6
ISI	4-47

Index

L

LCD	4-68
LOCAL key	1-1

M

manual scaling	4-69
mark	4-48
marker	4-38, 4-41, 4-42, 4-43, 4-70
measured point, display method of	4-28
measured T	4-11
measurement conditions	4-55
measurement function	4-55
measurement start	4-64
measurement stop	4-66
message	3-1
modulation type	4-38
modulation type on the auto window	4-47
multi window	4-39, 4-48
multi-line messages	1-6

O

offset value on constant T	4-39, 4-49
output queue	5-5
OVERlap	4-29
overlap	4-29

P

panorama display	4-29
Password	2-4
phase adjustment	4-53
polarity	4-56, 4-64
printout	4-34, 4-35
program message	3-1

R

recall	4-32
reference clock	4-64
remote/local switching	1-3
response message	3-2
rest mode	4-63
rest time	4-63

S

sampling interval	4-64
sampling mode	4-56
save	4-32
scale	4-28
scale upper limit	4-44
scale value	4-43
screen image, color/gradation of	4-35
screen image data	4-34
screen image, data format of	4-35
screen image, output destination of	4-35
screen image, storage file name	4-35
setup data	4-32
setup data, recalling of	4-59
setup data, storing of	4-66
single measurement	4-64
single window	4-42
slope	4-56
space	4-48

stat item	4-30
statistical computation items	4-10
statistical display, turning ON/OFF of	4-30
status byte	5-2
status report	5-1
Sync function, ON/OFF of	4-50

T

time	4-68
Time Out	2-4
time stamp mode	4-38
time variation	4-69
time variation, statistical computation of	4-11, 4-23
transition filter	4-65, 5-4
transmission of the measured data	4-58
trigger condition	4-51
trigger level	4-53
trigger mode	4-53
trigger mode of ISI	4-50

U

uni-line messages	1-6
User Name	2-4

W

window 1, ON/OFF of	4-38, 4-47
window label	4-50
window mode	4-39, 4-48
window, offset value of	4-40
windows, the number of	4-39, 4-49

X

X center, scale value	4-42
X span	4-43

Command List

{HHIStogram<x> THIStogram<x>} group	
:{HHIStogram<x> THIStogram<x>}?	4-38
:AUTO:MODulation	4-38
:AUTO:TTYPe	4-38
:AUTO:WINDow1:STATe	4-38
:AUTO?	4-38
:MARKer:LOW	4-38
:MARKer?	4-38
:MARKer[:STATe]	4-39
:MULTi:FREQuency	4-39
:MULTi:OFFSet	4-39
:MULTi:SIZE	4-39
:MULTi:TVALue	4-40
:MULTi:UPDate	4-40
:MULTi:WINDow<x>:HORizontal:CENTer	4-40
:MULTi:WINDow<x>:HORizontal:SPAN	4-41
:MULTi:WINDow<x>:HORizontal?	4-40
:MULTi:WINDow<x>:LABel	4-41
:MULTi:WINDow<x>:MARKer:{LEFT RIGHT}	4-41
:MULTi:WINDow<x>:MARKer:{LVALue RVALue}?	4-42
:MULTi:WINDow<x>:MARKer?	4-41
:MULTi:WINDow<x>?	4-40
:MULTi?	4-39
:SINGle:FREQuency	4-42
:SINGle:HORizontal:CENTer	4-42
:SINGle:HORizontal:SPAN	4-43

:SINgle:Horizontal?	4-42	*RST	4-73
:SINgle:MARKer:{LEFT RIGHT}	4-43	*SRE	4-73
:SINgle:MARKer:{LVALue RVALue}?	4-43	*STB?	4-73
:SINgle:MARKer?	4-43	*TRG	4-73
:SINgle:TVALue	4-43	*TST?	4-73
:SINgle?	4-42	*WAI	4-73
:VERTical:AXIS	4-44	COMMunicate group	
:VERTical:HIGH	4-44	:COMMunicate?	4-25
:VERTical?	4-43	:HEADer	4-25
[[:MODE]	4-39	:VERBose	4-25
AScale Group		:WAIT	4-26
:AScale	4-8	:WAIT?	4-26
CALCulation group		DISPlay group	
:AREA	4-10	:BGRaph	4-28
:CALCulation	4-10	:BLOCK	4-28
:PARAmeter:AVERage	4-22	:DISPlay?	4-28
:PARAmeter:CLEar	4-22	:DOTConnect	4-28
:PARAmeter:DEVIation	4-22	:DOTType	4-28
:PARAmeter:DEVT	4-22	:GRAPhsize	4-29
:PARAmeter:FLUTter	4-22	:GRID	4-29
:PARAmeter:JITter	4-22	:ITEM	4-29
:PARAmeter:MAXimum	4-22	:OVERlap	4-29
:PARAmeter:MEDian	4-22	:PANorama[:STATe]?	4-29
:PARAmeter:MINimum	4-22	:SGRaph	4-29
:PARAmeter:MODE	4-23	:SITem	4-30
:PARAmeter:PTOPeak	4-23	:SSTyle	4-30
:PARAmeter:SDEVIation	4-23	:STATistic	4-30
:PARAmeter:TAVerage	4-23	:WINDow	4-30
:PARAmeter:TFLutter	4-23	FILE group	
:PARAmeter:TJITter	4-23	:CDIRectory	4-31
:PARAmeter:TMAXimum	4-23	:DELete:{BINary BMP POSTscript SETup STATistic	
:PARAmeter:TMINimum	4-23	TIFF TEXT}	4-32
:PARAmeter:TPTopeak	4-24	:DRIVE	4-32
:PARAmeter:TRF	4-24	:FILE?	4-31
:PARAmeter:TSDeVIation	4-24	:FORMat	4-32
:PARAmeter?	4-22	:FREE?	4-32
:POLarity	4-24	:LOAD:{BINary SETup}	4-32
[[:MEAS<x>]:AUTot?	4-11	:MDIRectory	4-32
[[:MEAS<x>]:CONSTt	4-15	:PATH?	4-32
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TAVerage?	4-11	:SAVE:{BINary SETup STATistic TEXT}	4-33
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TFLutter?	4-11	:SAVE:ANAMing	4-32
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TJITter?	4-12	:SAVE:COMMent	4-33
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TMAXimum?	4-12	:SAVE:SITem	4-33
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TMINimum?	4-13	:SAVE?	4-32
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TPTopeak?	4-13	HCOPY group	
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TRF?	4-14	:ABORT	4-34
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TSDeVIation?	4-14	:ANAMing	4-35
[[:MEAS<x>]:[:{BLOCK<x> BALL}]:TSNumber?	4-15	:CDIRectory	4-35
[[:MEAS<x>]:[:{WINDow<x> SUMMation}]:JITter?	4-18	:COMMent	4-35
[[:MEAS<x>]:[:{WINDow<x> SUMMation}]:PTOPeak?	4-20	:COMPression	4-35
[[:MEAS<x>]:[:{WINDow<x> SUMMation}]:SDEVIation?	4-21	:DEVIce	4-35
[[:MEAS<x>]:[:{WINDow<x> SUMMation}]:SNUMber?	4-21	:DRIVE	4-35
[[:MEAS<x>]:[:{WINDow<x>]:AVERage?	4-16	:FILEname	4-35
[[:MEAS<x>]:[:{WINDow<x>]:DEVIation?	4-16	:FORMat	4-35
[[:MEAS<x>]:[:{WINDow<x>]:DEVT?	4-17	:HCOPY?	4-34
[[:MEAS<x>]:[:{WINDow<x>]:FLUTter?	4-17	:TONE	4-35
[[:MEAS<x>]:[:{WINDow<x>]:MAXimum?	4-18	[[:EXECute]	4-35
[[:MEAS<x>]:[:{WINDow<x>]:MEDian?	4-19	IHIStogram group	
[[:MEAS<x>]:[:{WINDow<x>]:MINimum?	4-19	{:MARK<x> SPACE<x>}	4-48
[[:MEAS<x>]:[:{WINDow<x>]:MODE?	4-20	:AUTO:MODulation	4-47
Common command group		:AUTO:TTYPe	4-47
*CAL?	4-72	:AUTO:WINDow1:STATe	4-47
*CLS	4-72	:AUTO?	4-47
*ESE	4-72	:IHIStogram?	4-47
*ESR?	4-72	:MULTi:FREQuency	4-48
*IDN?	4-72	:MULTi:OFFSet	4-49
*OPC	4-72	:MULTi:SIZE	4-49
*OPC?	4-72	:MULTi:TVALue	4-49

Index

:MULTi:UPDate	4-49	:GATE[:MODE]	4-64
:MULTi:WINDow<x>:HORizontal:CENTer	4-49	:INHibit	4-64
:MULTi:WINDow<x>:HORizontal:SPAN	4-49	:INTERval	4-64
:MULTi:WINDow<x>:HORizontal?	4-49	:RCLock	4-64
:MULTi:WINDow<x>:LABel	4-50	:SAMPle?	4-61
:MULTi:WINDow<x>:MARKer:{LEFT RIGHT}	4-50	SStart group	
:MULTi:WINDow<x>:MARKer?	4-50	:SStart	4-64
:MULTi:WINDow<x>?	4-49	StARt group	
:MULTi?	4-48	:StARt	4-64
:POLarity	4-50	STATus group	
:SYNC:STATe	4-50	:CONDition?	4-65
:TARGeT	4-50	:EESe	4-65
:TMODe	4-50	:EESR?	4-65
:TRIGger	4-51	:ERRor?	4-65
[:MODE]	4-48	:FILTer<x>	4-65
INPut group		:QMESsage	4-65
:{ACHannel BCHannel}:COUPling	4-53	:STATus?	4-65
:{ACHannel BCHannel}:IMPedance	4-53	StOP group	
:{ACHannel BCHannel}:TRIGger:LEVel	4-53	:StOP	4-66
:{ACHannel BCHannel}:TRIGger:MODE	4-53	StORe group	
:{ACHannel BCHannel}:TRIGger?	4-53	:StORe	4-66
:{ACHannel BCHannel}?	4-52	SYSTem group	
:AGATe:LEVel	4-53	:BEEP	4-67
:AGATe?	4-53	:CLICksound	4-68
:BCHannel:PHASe:ADJusT	4-53	:DATE	4-68
:BCHannel:PHASe?	4-53	:LCD:AOFF	4-68
:INHibit:LEVel	4-54	:LCD:ATime	4-68
:INHibit?	4-54	:LCD:BRIGHtness	4-68
:INPut?	4-52	:LCD?	4-68
MEASure group		:LCD[:STATe]	4-68
:FUNCTion	4-55	:SYSTem?	4-67
:MEASure?	4-55	:TIME	4-68
:MODE	4-56	:WARNing	4-68
:POLarity	4-56	TVARiation<x> group	
:SLOPe	4-56	:HORizontal:MINimum	4-70
MEMory group		:HORizontal:SPAN	4-70
:BLOCk	4-57	:HORizontal?	4-69
:BSIZe?	4-57	:MARKer:{HIGH LOW}	4-70
:BYTeorder	4-57	:MARKer:{LEFT RIGHT}	4-70
:CLEar	4-57	:MARKer:{LVALue RVALue}?	4-70
:DATaselect	4-58	:MARKer?	4-70
:END	4-58	:MARKer[:STATe]	4-70
:FORMat	4-58	:TVARiation<x>?	4-69
:MEMory?	4-57	:VERTical:CENTer	4-70
:SEND<x>?[{POSitive NEGative}]	4-58	:VERTical:SPAN	4-71
:SIZE<x>?	4-59	:VERTical?	4-70
:StARt	4-59	UNIT group	
RECall group		:FREQuency	4-71
:RECall	4-59	:TIME	4-71
SAMPle group		:UNIT?	4-71
:ARMinG:DELay:{AEVENTsize BEVENTsize EVENTsize}	4-62	:VOLTage	4-71
:ARMinG:DELay:{ATIME BTIME TIME}	4-62		
:ARMinG:DELay?	4-61		
:ARMinG:DELay[:MODE]	4-62		
:ARMinG:SLOPe	4-62		
:ARMinG:SOURce	4-62		
:ARMinG?	4-61		
:BLOCk:REST:EVENT	4-63		
:BLOCk:REST:TIME	4-63		
:BLOCk:REST?	4-62		
:BLOCk:REST[:MODE]	4-63		
:BLOCk:SIZE	4-63		
:BLOCk?	4-62		
:BLOCk[:STATe]	4-63		
:GATE:EVENTsize	4-63		
:GATE:POLarity	4-64		
:GATE:TIME	4-64		
:GATE?	4-63		