HIOKI

8840 MEMORY HiCORDER

9587 GP-IB INTERFACE

INSTRUCTION MANUAL

HIOKI E.E. CORPORATION

Contents

Safety Notes

⚠ DANGER

- To avoid the danger of electric shock or damage to the unit, before removing the 9587 GP-IB interface, turn off the power, and remove the power cable.
- Normally keep GP-IB interface installed permanently. If not fitted, it must be replaced by a blanking panel.

⚠ WARNING

- The GP-IB interface is not isolated from the 8840 system.
- Exercise caution, because the ground of the logic inputs and the GP-IB interface ground are connected.

Chapter Summary

Chapter 1 gives an overview of the GP-IB interface.

Chapter 2 contains the GP-IB specifications.

Chapter 3 describes the operation procedures.

Chapter 4 gives the GP-IB command list.

Chapter 5 describes the GP-IB operation following examples for command.

Chapter 6 describes the program to operate GP-IB interface.

Chapter 7 covers method of setting plotter out.

Chapter 8 contains the standard related to the GP-IB.

Chapter 9 describes method of installing the GP-IB interface.

Chapter 1

Outline

Outline

The GP-IB (General Purpose Interface Bus) was developed as an interface for general use by programmable instrumentation, and as an interface is rich in expandability and has many distinctive features.

There are various interfaces with specific names apart from the GP-IB, such as the IEEE-488 bus, the IEC bus, and the HP-IB which is an internal standard within the Hewlett-Packard Company. These are basically the same standard, but, because the number of connector pins and the arrangement of the signals and so on differ, much care should be exercised.

In this explanation of management and operation, only the GP-IB related resources of the 8840 will be described.

If more detailed knowledge of the GP-IB interface is required, reference should be made to the following literature:

The Institute of Electrical and Electronics Engineers, Inc.: "IEEE Standard Digital Interface for Programmable Instrumentation", IEEE Std 488.1-1987, IEEE Std 488.2-1987 (1987)

Chapter 2

GP-IB Specification

2-1. Standards

IEEE Standard 488.1-1987

IEEE Standard 488.2-1987

2-2. Interface Functions

Function	Implementation
SH1	SH (Source Handshake) - All Functions
AH1	AH (Acceptor Handshake) - All Functions
T5	Basic Talk Function, Serial Poll Function, Talk Only Function MLA (My Listen Address) Talk Release Function
L4	Basic Listener Function, MTA (My Talk Address) Listen Release Function
SR1	SR (Service Request) - All Functions
RL1	RL (Remote/Local) - All Functions
PP0	PP (Parallel Poll) - No Function
DC1	DC (Device Clear) - All Functions
DT0	DT (Device Trigger) - No Function
C0	C (Control) - No Function

2-3. GP-IB Signal Lines

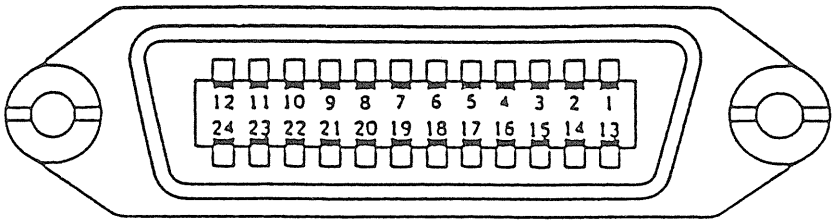
Bus Signal Lines		Remarks	
Data bus	DIO 1 (Data Input Output 1) DIO 2 (Data Input Output 2) DIO 3 (Data Input Output 3) DIO 4 (Data Input Output 4) DIO 5 (Data Input Output 5) DIO 6 (Data Input Output 6) DIO 7 (Data Input Output 7) DIO 8 (Data Input Output 8)	Apart from input and output of data, these are used for input and output of interface messages and device messages.	
Transfer bus	DAV (Data Valid) NRFD (Not Ready For Data) NDAC (Not Data Accepted)	Signal which indicates data bus information validity. Input preparation completed signal. Input completed signal.	These perform acceptor and source handshake.
Control bus	ATN (Attention) IFC (Interface Clear) SRQ (Service Request) REN (Remote Enable) EOI (End or Identify)	Signal which indicates that the information on the data bus is an interface message or a device message. Signal which sets the interface bus system to the initial condition. Signal which requests a non-synchronous service. Signal which performs changeover of remote and local control. Indicates the last byte of data.	

2-4. Connector Pin Assignment

On the 8840: RC10 (F) -24R-LNA (made by Hirose) or compatible.

On the cable: 57-10240 (made by DDK) or compatible.

Pin arrangement diagram for the GP-IB interface connector on the 8840:

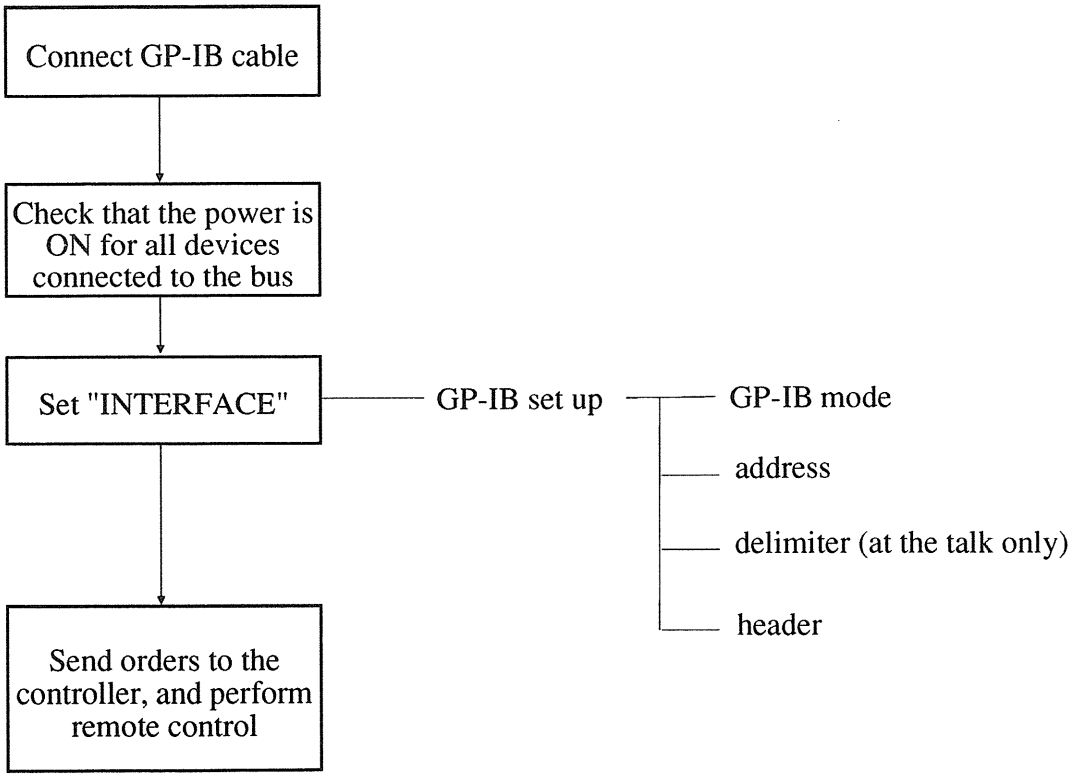


Pin number	Name of signal line	Pin number	Name of signal line
1	DIO 1	13	DIO 5
2	DIO 2	14	DIO 6
3	DIO 3	15	DIO 7
4	DIO 4	16	DIO 8
5	EOI	17	REN
6	DAV	18	GND
7	NRFD	19	GND
8	NDAC	20	GND
9	IFC	21	GND
10	SRQ	22	GND
11	ATN	23	GND
12	SHIELD	24	LOGIC GND

Chapter 3

Method of Operation

3-1. Basic Operational Procedure



⚠ WARNING

The GP-IB interface is not isolated from the 8840 system. Exercise caution, because the ground of the logic inputs and the GP-IB interface ground are connected.

3-2. Setup Procedure

On the 8840, use the interface setting screen, accessed from the "system" screen, to set the GP-IB address for the unit, and select whether or not to use headers, mode, and delimiter in messages output by the 8840.

Procedure

1. Press the SYSTEM key to display the "system" screen.

2. Press F5 (1of2), the press F2 (INTERFACE).

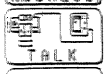
   Using the cursor keys, move the flashing cursor in order to the items shown in the figure below by the reference numbers, and make the settings using the function keys.

3. Set the GP-IB operation mode for this unit.

   Set the GP-IB address for this unit on the bus.

Function key indication *Meaning*

: ADDRESSABLE... Assign a device address, so this unit can be used both as talker and listener.



: TALK ONLY ... Use this unit as talker only. (Only when using a plotter)

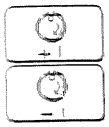
: DISABLE ... Do not use the GP-IB interface.

[If the TALK ONLY mode is select, set the delimiter. Refer to step 6.]

4. Set the GP-IB device address.

   Use F1 and F2, or the JOG control knob to adjust the numerical value.

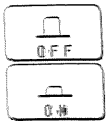
Function key indication *Meaning*


1 to 30

5. Enable or disable the headers.

   Select whether or not this unit as talker should output an identifying header at the beginning of each message it sends.

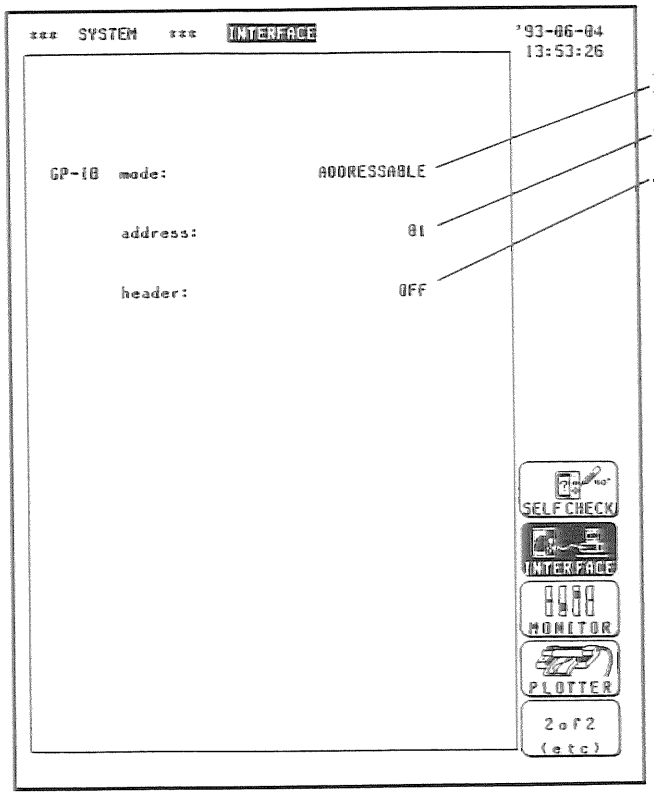
Function key indication *Meaning*



: No headers

: Output headers

SYSTEM screen (INTERFACE)



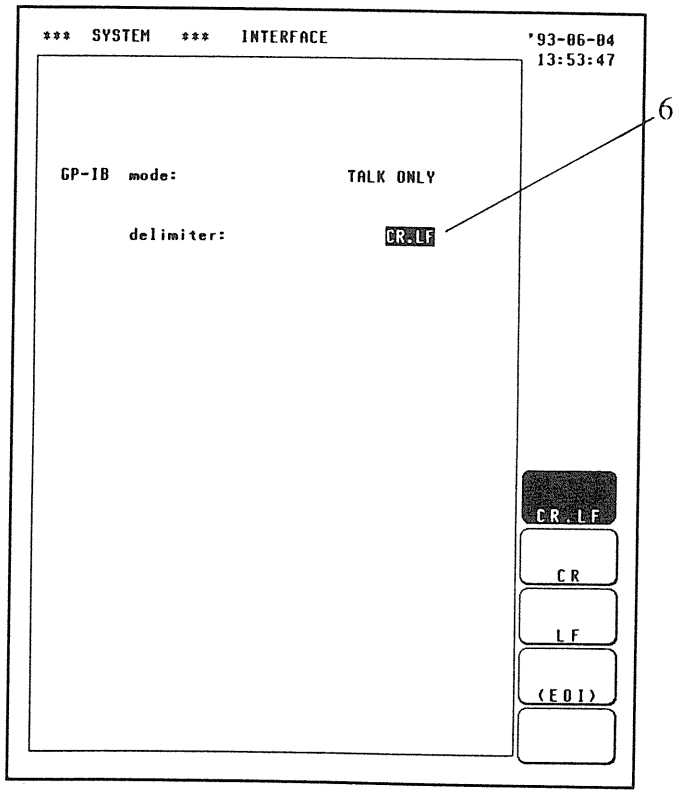
6. Select the GP-IB delimiter for talk-only mode.

   Select the appropriate delimiter sequence for the plotter being used.

Function key indication		*Meaning*
CR.LF	:	CR + LF (EOI)
CR	:	CR (EOI)
LF	:	LF (EOI)
(EOI)	:	(EOI)

[For TALK ONLY mode]

SYSTEM screen (INTERFACE)

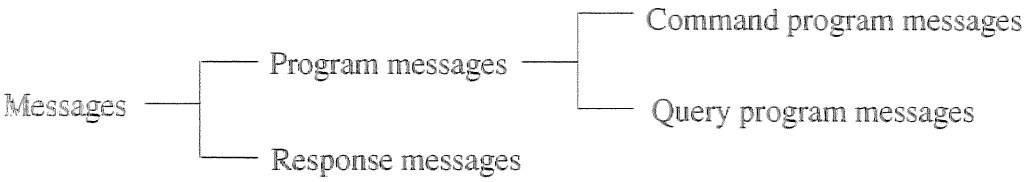


3-3. Receive and Send Protocols

(1) Messages

Data received or sent by the GP-IB interface is called a message.

The following are the message types:



Of these, program messages are those received by the unit from the controller, while response messages are those sent from the unit to the controller.

Program messages are command messages or query messages.

Command messages are orders for control of the device, such as for making settings or for reset or the like.

Query messages are orders for responses relating to results of operation, results of measurement, or the state of device settings.

Response messages are sent in response to query program messages.

After a query message has been received, a response message is produced the moment that its syntax has been checked.

(2) Command syntax

When no ambiguity would arise, the term "command" is henceforth used to refer to both command and query program messages.

The 8840 accepts commands without distinction between lower case and upper case letters. It generates response messages in the long form (when headers are enabled) and in upper case letters.

The names of commands for the 8840 are as far as possible mnemonic. Furthermore, all commands have a long form, and an abbreviated short form.

In command references in this manual, the short form is written in upper case letters, and then this is continued in lower case letters so as to constitute the long form. Either of these forms will be accepted during operation, but intermediate forms will not be accepted. Further, during operation both lower case letters and upper case letters will be accepted without distinction.

Example:
For "DISPLAY", either "DISPlay" (the long form) or "DISP" (the short form) will be accepted. However, any one of "DISPLA", "DISPL", or "DIS" is wrong and will generate an error.

(3) Command program headers

Commands must have a header, which identifies the command in question.

There are three kinds of header: the simple command type, the compound command type, and the standard command type.

- Simple command type header:

  The first word constitutes the header.

  Example :HEADer ON

  Simple command type header. / data

- Compound command type header:

  A header made up from a plurality of simple command type headers marked off by colons.

  Example :CONFigure : TDIV 1. E-3

  Simple command type header. / Simple command type header. / data

  Compound command type header.

- Standard command type header:

  A command beginning with an asterisk and stipulated by IEEE488.2

  Example *RST

(4) Query program headers

These are for commands used for interrogating the unit about the result of an operation or about a setting.

These can be recognized as queries by a question mark appearing after the program header. The structure of the header is identical to that of a command program header, with "?" always being affixed to the last command. There are queries possible in each of the three previously described types of command form.

Example :HEADer?

Query program header

(5) Response messages

Response messages relating to queries are made up from header portions (which also may be absent due to header disablement) and data portions identical to those of program messages, and as a general rule are sent in an identical format to the format of the program message corresponding to their originating query.

(6) Terminators and separators

① Message Terminator

A terminator is used in order to separate the transmission of one message from another, and this terminator is not itself included in the message. On the 8840, LF, EOI, or LF+EOI is used as the message terminator, and LF+EOI is also used as the response message terminator.

② Message Unit Separator

A semicolon ";" is used as a message unit separator when it is desired to set out several messages on a single line.

Example :CONFIGURE:TDIV 1. E3;:CONFIGURE:SHOT 25

Message unit separator

③ Header separator

With a message which has both a header and data, a space " ␣ " is used as a header separator to separate the header from the data. Actually, " ␣ " is not displayed on the screen.

Example :CONFIGURE:SHOT ␣ 25

Header separator

④ Data separator

Commas are used as data separators for separating several data items from one another.

Example :DISPLAY:DRAW CH1 , DARK

Simple command type header.

Data separator

Compound command type header.

Header separator

(7) The command tree

The rule when writing several messages of compound command form on the same line, when no colon is prefixed to the next header after the semicolon (the message unit separator), is that that header is considered as continuing on from the header before the last colon in the message directly preceding.

This corresponds to the general concept of the current directory in the directory structure of UNIX or MS-DOS, and this directly preceding header is called the "current path".

Example 1 :CONF:TDIV 1. E-3;:CONF:SHOT 25

Example 2 :CONF:TDIV 1. E-3;SHOT 25

Both Example 1 and Example 2 are messages setting TIME/DIV to 1 and recording length to 25 divisions.

With Example 1, because there is a colon directly after the semicolon, the current position is the "root". Accordingly the reference of the next command is performed from the root.

On the other hand, with Example 2, because with ":CONF:TDIV 1. E-3;" the current path has become ":CONF", it is now possible to omit the ":CONF:" before "SHOT".

To reiterate, the colon at the beginning of a command forces the search for the command to begin from the root. Thus in Example 1:

The first colon indicates that the "CONFIGURE" command is at the root level.

(8) Data format

The 8840 uses character data, decimal data and character string data.

- Character data

① The first character must be alphabetic.

② The characters after the first character can only be alphabetic characters, numerals, or underline characters (_).

③ As alphabetic characters, during sending only upper case letters are used, but during receiving both upper case and lower case letters are permitted.

- Decimal data

Decimal data values are represented in what is termed NR format.

There are three types of NR format from NR1 to NR3, and each of these can appear as either a signed number or an unsigned number. Unsigned numbers are taken as positive.

Further, if the accuracy of a numerical value exceeds the range with which the 8840 can deal, it is rounded off. (5 and above is rounded up; 4 and below is rounded down).

NR1 format - integer data.
Examples: +15, -20, 25
NR2 format - fixed point numbers.
Examples: +1.23, -4.56, 7.89
NR3 format - floating point numbers.
Examples: +10E-3, -2.3E+3

NRf format

The term "NRf format" includes all these three formats.

When the 8840 is receiving it accepts NRf format, but when it is sending it utilizes whichever one of the formats NR1 to NR3 is indicated in the particular command.

- Character string data

Character string data is enclosed within quotation marks.

① The data is composed of 7 bit ASCII characters.

② Characters which cannot be handled by the 8840 are replaced by spaces.

③ When the 8840 is sending, only the double quotation mark (") is used as a quotation mark, but when receiving both this double quotation mark and also the single quotation mark (') are accepted.

3-4. Remote Control

- **Local state**

  This is the state in which the 8840 is controlled by its keys. When the power is turned on, the 8840 always comes up in local state.

- **Remote state**

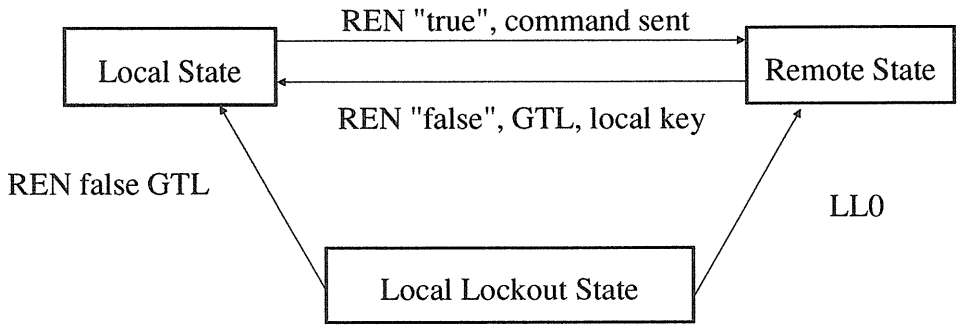
  In this state the 8840 is controlled from the GP-IB interface (the REN line is "true"), and its keys are disabled. When in the remote state, the 8840 returns to local state if the local key (F5 (LOCAL)) is pressed.

- **Local lockout state**

  When an LLO (Local Lockout) command (this is a GP-IB universal command) is received, even if the local key is pressed, the 8840 is prevented from returning to the local state. This state is called the local lockout state.

  In order to return the 8840 from the local lockout state to the local state, it is necessary either (a) to send a GTL (Go To Local) command (this is a GP-IB universal command), or (b) to turn the power to the 8840 temporarily off and then on again, or (c) to bring the line REN to "false".

  If a command is sent with REN in the "false" state, then the only way to return to the local state is with the local key.



Program example ..... HP-9816 (Hewlett-Packard)

local lockout	LOCAL LOCKOUT 7
local	LOCAL 7

3-5. Device Clear

When the 8840 receives the device clear command, it clears the input buffer and the output queue.

The device clear command is exemplified by the following:

HP 9816 (made by Hewlett-Packard)

```
CLEAR 7
```

3-6. The Status Byte and the Event Registers

(1) The status byte

Each bit of the status byte is a summary (logical OR) of the event register corresponding to that bit.

Further, the status byte and each event register has an enable register corresponding to it, and according to the setting of this enable register (which starts off at zero when the power is turned on) it is possible to mask the service requests originating from each event.

Status byte bit settings

bit 7:	Unused: 0
bit 6: rsv MSS	Set when a service request is issued.
bit 5: ESB	Event summary bit. Shows a summary of the standard event status register.
bit 4: MAV	Message available. Shows that a message is present in the output queue.
bit 3:	Unused: 0
bit 2:	Unused: 0
bit 1:	Unused: 0
bit 0: ESB0	Event summary bit 0 Shows a summary of event status register 0.

The following commands are used for reading the status byte, and for setting the service request enable register and for reading it.

Reading the status byte ..... *STB?

Setting the service request enable register ..... *SRE

Reading the service request enable register ..... *SRE?

(2) Standard event status register (SESR)

The summary of this register is set in bit 5 of the status byte.

Each bit is masked by setting the standard event status enable register (which starts off at zero when the power is turned on).

The circumstances when the contents of the standard event status register are cleared are as listed below.

1. When the *CLS command is received.
2. When the contents have been read by an *ESR? query.
3. When the power is turned off and turned on again.

Bit allocations in the standard event status register

bit 7: PON	The power has been turned on again. Since this register was last read, the unit has been powered off and on.
bit 6: URQ	User request: not used.
bit 5: CME	Command error. There is an error in a command that has been received; either an error in syntax, or an error in meaning.
bit 4: EXE	Execution error. An error has occurred while executing a command. Range error; Mode error.
bit 3: DDE	Device dependent error. It has been impossible to execute some command, due to an error other than a command error, a query error, or an execution error.
bit 2: QYE	Query error. The queue is empty, or data loss has occurred (queue overflow)
bit 1:	Request for controller right (not used) Unused: 0
bit 0: OPC	Operation finished. Only set for the *OPC command.

The following commands are used to read the standard event status register, and to set or read the standard event status enable register.

Read the standard event status register .....*ESR?

Set the standard event status enable register .....*ESE

Read the standard event status enable register .....*ESE?

(3) Event status register 0 (ESR0)

The summary of this register is set in bit 0 of the status byte.

Each bit is masked when the event status enable register 0 (which starts off at zero when the power is turned on) is set.

The circumstances when the contents of event status register 0 are cleared are as listed below.

1. When the *CLS command is received.
2. When the contents have been read by an :ESR0? query.
3. When the power is turned off and turned on again.

The bits of event status register 0

bit 7:	Waveform decision fail (NG).
bit 6:	Unused.
bit 5:	Unused.
bit 4:	Unused.
bit 3:	Printer operation finished (print, or copy, output).
bit 2:	Trigger wait finished (set when the trigger event occurs).
bit 1:	Measurement operation concluded (set by STOP).
bit 0:	Error not related to the GP-IB interface; printer error etc.

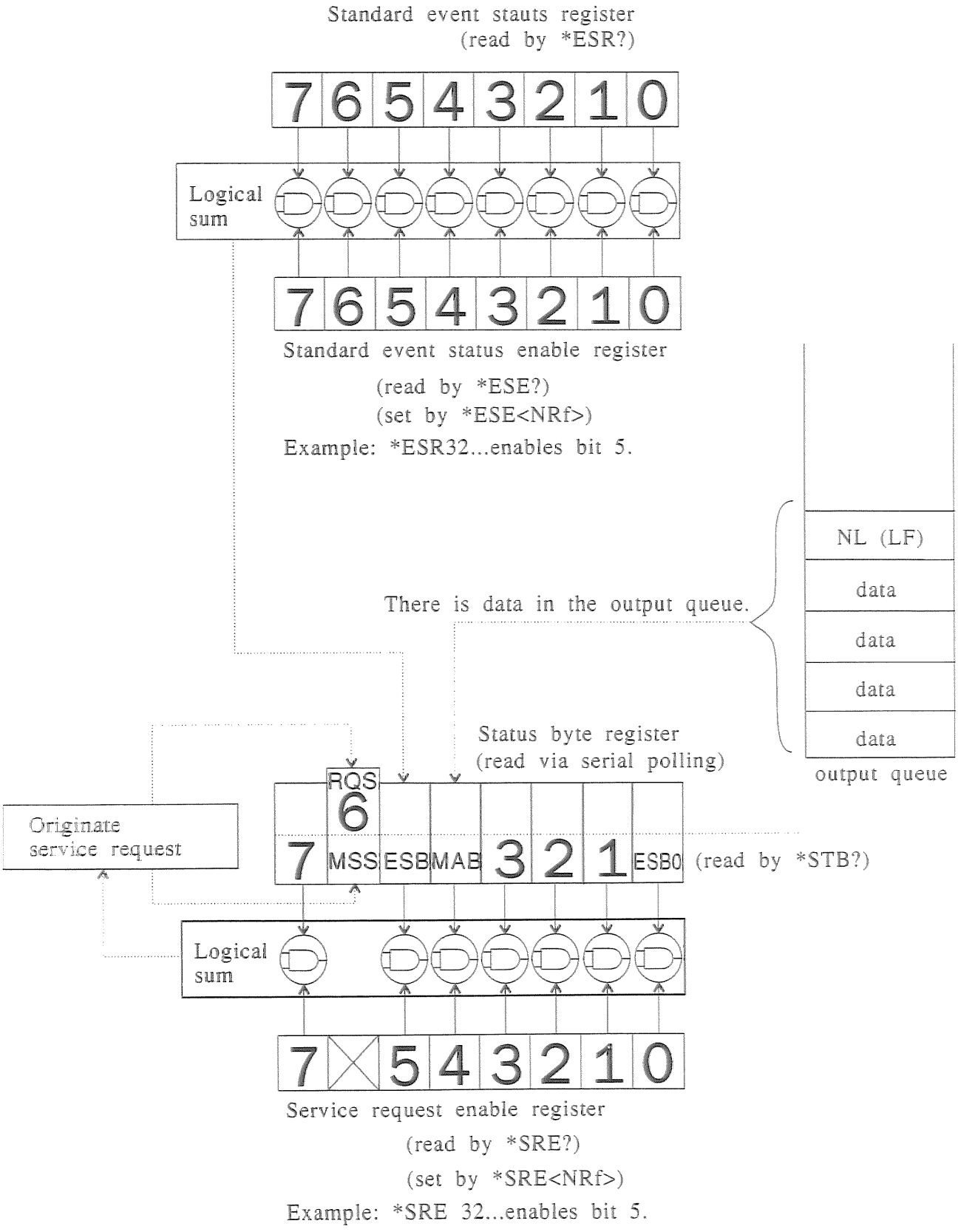
The following commands are used for reading the event status register 0, and for setting the event status enable register 0 and for reading it.

Reading event status register 0 ..... :ESRO?

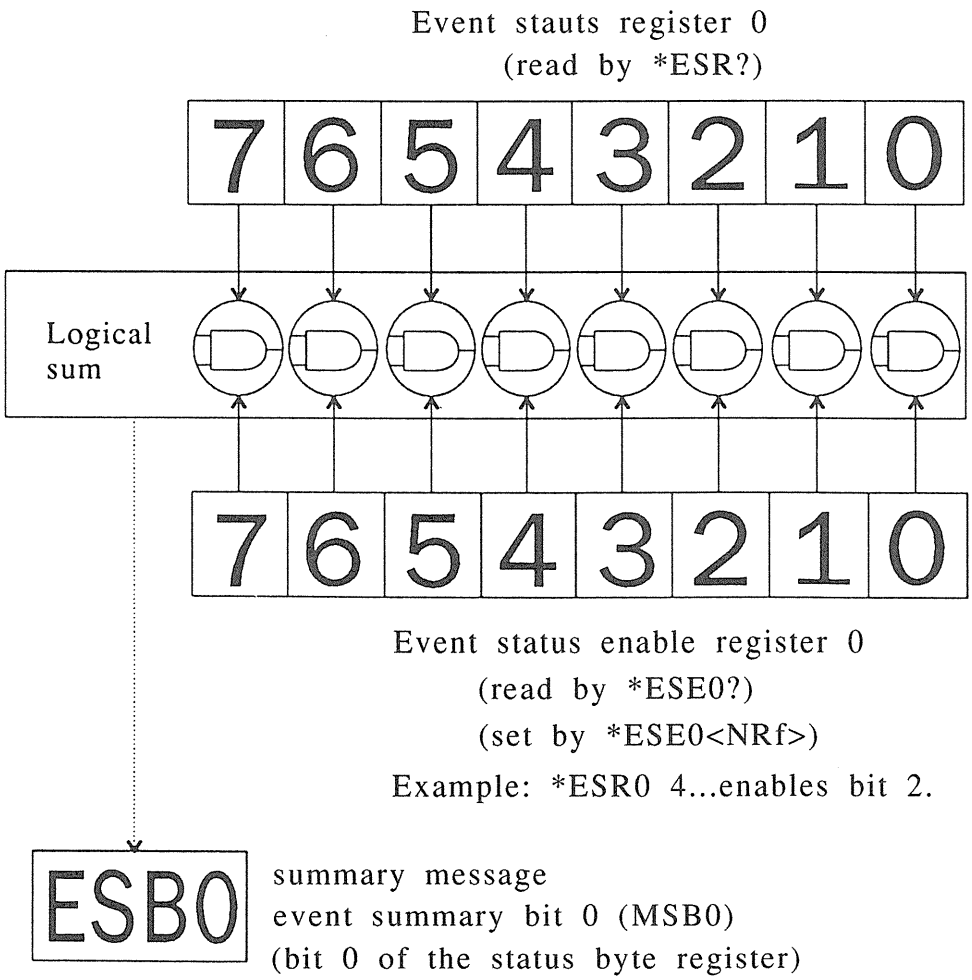
Setting event status enable register 0 ..... :ESE0

Reading event status enable register 0 ..... :ESE0?

Status byte data structure:



Event status register 0 data structure



3-7. The Input Buffer and the Output Queue

(1) Input buffer

The 8840 has an input buffer of 512 bytes capacity.

Messages which are received are put into this buffer and executed in order.

However, an :ABORT command is executed instantly as soon as it is received.

(2) Output queue

The 8840 has an output queue of 256 bytes capacity.

Response messages are accumulated in this queue and are read out from the controller.

The circumstances when the output queue is cleared are as listed below:

1. When the controller has read out its entire contents.
2. When a device clear is issued.
3. When the power is turned off and turned on again.
4. Upon receipt of the next message.

If the length of a response message has exceeded 256 bytes, a query error occurs.

3-8. GP-IB Errors

When a command which has been received contains an error, that one of bits 2 to 5 of the standard event status register which corresponds to the event which has occurred is set.

Further, if a command has given rise to an error (apart from an execution error), commands accumulated in the input buffer and waiting for execution after that command are ignored.

Bit allocations in the standard event status register

bit 7: PON	The power has been turned on again. Since this register was last read, the unit has been powered off and on.
bit 6: URQ	User request: not used.
bit 5: CME	Command error. There is an error in a command that has been received; either an error in syntax, or an error in meaning.
bit 4: EXE	Execution error. An error has occurred while executing a command. Range error; Mode error.
bit 3: DDE	Device dependent error. It has been impossible to execute some command, due to an error other than a command error, a query error, or an execution error.
bit 2: QYE	Query error. The queue is empty, or data loss has occurred (queue overflow)
bit 1:	Request for controller right (not used) Unused: 0
bit 0: OPC	Operation finished. Only set for the *OPC command.

Chapter 4

GP-IB Commands Summary

4-1. Standard Commands Stipulated by IEEE488.2

Command	Data (for a query, response data)	Explanation	Ref page
*IDN?	Maker's name, model number, serial number, software version (not used, zero)	Queries device ID.	5-4
*OPT?	Whether channel 1 input unit exists Whether channel 2 input unit exists . . . Whether channel 8 input unit exists	Queries device option provision.	5-4
*RST		Device initial setting.	5-4
*TST?	A<NR1> (0 = normal, 1 = failure)	Queries the result of the self-test.	5-4
*OPC		Sets the LSB of SESR after all action has been completed.	5-5
*OPC?	A<NR1>	Queries whether all action has been completed. ASCII [1] is the response.	5-5
*WAI		Wait until action fully completed.	5-5
*CLS		Clears the status byte and associated queues.	5-5
*ESE A	A=0 to 255	Sets SESER.	5-5
*ESE?	A<NR1> 0 to 255	Queries SESER.	
*ESR?	A<NR1> 0 to 255	Queries SESR.	5-6
*SRE A	A= 0 to 255	Sets SRER.	5-6
*SRE?	A<NR1> 0 to 63, 128 to 191	Queries SRER.	
*STB?	A<NR1> 0 to 255	Reads the STB and the MSS bit, without performing serial polling.	5-6
:ESE0 A	A= 0 to 255	Writes ESER0.	5-7
:ESE0?	A<NR1> 0 to 255	Reads ESER0.	5-7 #
:ESR0?	A<NR1> 0 to 255	Reads ESR0.	5-7

specific to the 8840.

4-2. Commands specific to the 8840

4-2-1. Execution control etc. (common to all functions)

Command	Data (for a query, response data)	Explanation	Ref page
:STARt		Same as the START key.	5-8
:STOP		Same as the STOP key.	5-8
:ABORT		Forced halt.	5-8
:PRINt		Same as the PRINT key.	5-8
:HCOPy		Same as the COPY key.	5-8
:FEED A	A=1 to 255 (unit: mm)	Feeds the paper the specified distance.	5-9
:AUTO		Sets the time axis and the voltage axis automatically. Only the memory recorder function	5-9
:LIGHt A$	A$=ON, OFF	Enables and disables LCD back light	5-9
:LIGHt?	A$	Enables and disables LCD back light	
:ERRor?	A <NR1> error number	Queries 8840 error number.	5-9
:HEADer A$	A$ = OFF,ON	Enables and disables headers.	5-10
:HEADer?	A$	Queries header enablement.	
:FUNCtion A$	A$ = MEM,REC,XYC,FFT	Changes the function.	5-10
:FUNCtion?	A$	Queries the function.	
:A4 PRint		Same as the FEED key + COPY key on the main unit.	5-10

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-2. Setting and querying the time axis range, the shot length, etc. (CONFigure commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CONFigure				
:TDIV A	A = time per division (unit: seconds)	Sets the time axis range	MEM REC	5-11
:TDIV?	A<NR 3> (unit: seconds)	Queries the time axis range		
:SHOT A	A = recording length (unit: DIV)	Sets the recording length	MEM REC	5-11
:SHOT?	A<NR 1> (unit: DIV)	Queries the recording length		
:FORMat A$	A$= SINGle, DUAL, QUAD, OCT, XYSingle, XYDual (MEM) SINGle, DUAL, QUAD, OCT, (REC) XYSingle , XYDual (XYC) SINGle, DUAL, NYQuist(FFT)	Sets the format.	ALL	5-11
:FORMat?	A$	Queries the format.		
:DOTLine A$	A$ = DOT, LINE	Sets the interpolation function.	MEM XYC FFT	5-12
:DOTLine?	A$	Queries the interpolation function.		
:ROLL A$	A$=OFF, ON	Enables and disables roll mode.	MEM	5-12
:ROLL?	A$	Queries roll mode enablement.		
:OVWRite A$	A$=OFF,ON	Enables and disables waveform superimposition.	MEM	5-12
:OVWRite?	A$	Queries waveform superimposition enablement.		
:PRKInd A$	A$ = WAVE, LOGGing	Specifies the printer output style.	MEM REC FFT	5-13
:PRKInd?	A$	Queries the printer output style.		
:LOGGing A	A = 0.01 to 100 (MEM) 1 to 100 (REC)	Specifies the logging output interval	MEM REC	5-13
:LOGGing?	A<NR 2>	Queries the logging output interval		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CONFigure				
:PRINt A$	A$ = OFF, ON	Sets printer output.	REC	5-13
:PRINt?	A$	Queries printer output.		
:SMOOth A$	A$ = OFF, ON	Enables and disables roll mode.	MEM	5-14
:SMOOth?	A$	Queries smooth printing enablement.		
:ATPRint A$	A$ = OFF, ON	Enables and disables auto print.	MEM FFT	5-14
:ATPRint?	A$	Queries auto print enablement.		
:ATSAve A$	A$ = OFF, ON	Enables and disables auto save.	MEM FFT	5-14
:ATSAve?	A$	Queries auto save enablement.		
:MEMDiv A$	A$ = OFF, SEQ, MULTI	Sets the memory division function.	MEM	5-15
:MEMDiv?	A$	Queries the memory division function.		
:MAXBlock A	A =3,7,15,31 (in multi-block function); A =2 to 31 (in sequential save function)	Sets the memory block number (in sequential save and multi-block function)	MEM	5-15
:MAXBlock?	A<NR 1>	Queries the memory block number		
:USEBlock A	A =1 to number of memory divisions	Sets the number of the memory block used (in sequential save and multi-block function).	MEM	5-15
:USEBlock?	A<NR 1>	Queries the number of the memory block used.		
:REFBlock A	A =OFF, 1 to number of memory divisions	Sets the reference block (in multi-block function).	MEM	5-16
:REFBlock?	A<NR 1>	Queries the reference block.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CONFigure				
:WVCOmp A$	A$ = OFF, OUT, ALLOUT	Sets the waveform decision mode.	MEM FFT	5-16
:WVCOmp?	A$	Queries the waveform decision mode.		
:CMPStop A$	A$ = GO, NG, G_N	Sets the waveform decision stop mode.	MEM FFT	5-16
:CMPStop?	A$	Queries the waveform decision stop mode.		
:AVERage A	A =0, 2, 4, 8, 16, 32, 64, 128, 256 (0 ; OFF)	Sets the count for averaging.	MEM	5-17
:AVERage?	A<NR 1>	Queries the current setting of the count for averaging.		
:FFT AVERage A	A =2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096	Sets the count for averaging in the FFT function.	FFT	5-17
:FFT AVERage?	A<NR 1>	Queries the current setting of the count for averaging in the FFT function.		
:FFTAVKind A$	A$ = OFF, T_EXP, F_EXP, T_LIN, F_LIN, F_PEAK	Sets the averaging method.	FFT	5-17
:FFTAVKind?	A$	Queries the currently set averaging method as character data.		
:FFTMode A, ch1$ (,ch2$)	A = 1, 2 ch1$, ch2$ = CH1 to CH8	Sets the FFT channel mode.	FFT	5-18
:FFTMode?	A<NR 1>, ch1$, ch2$	Queries the current FFT channel mode.		
:FFTWind A$ (,B)	A$ = RECTan, HANNing, EXPOnential B = 0 to 99 (%)	Sets the window function.	FFT	5-18
:FFTWind?	A$, B<NR 1>	Queries the current window function.		
:FFTFunction A$, B$	A$ =G1, G2 B$ =STR, LIN, RMS, PSP, ACR, HIS, OCT, TRF, COH, CSP, CCR, IMP	Sets the FFT analysis mode.	FFT	5-19
:FFTFunction? A$	A$, B$	Queries the current FFT analysis mode.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CONFigure				
:FFTRef A$	A$ = NEW, MEM	Designates the source for FFT data.	FFT	5-20
:FFTRef?	A$	Queries the current FFT data source.		
:FFTSCale A$, B$	A$ = G1, G2 B$ = AUTO, MANUal	Sets the display scaling method for a graph.	FFT	5-20
:FFTSCale? A$	A$, B$	Queries the current display scaling method for a graph.		
:FFTUp A$, B	A$ = G1, G2 B = -9.999E+29 to +9.999E+29	Sets the FFT vertical axis upper limit for a graph.	FFT	5-20
:FFTUp? A$	A$, B<NR3>	Queries the current FFT display scale vertical axis upper limit for a graph.		
:FFTLow A$, B	A$ = G1, G2 B = -9.999E+29 to +9.999E+29	Sets the FFT display scale vertical axis lower limit for a graph.	FFT	5-21
:FFTLow? A$	A$, B<NR3>	Queries the current FFT display scale vertical axis lower limit for a graph.		
:FFTXaxis A$, B$	A$ =G1, G2 B$ = 1_1oct, 1_3oct, (octave analysis) LINhz, LOGhz (otherwise)	Sets the x-axis.	FFT	5-21
:FFTXaxis? A$	A$, B$	Queries the present x-axis setting.		
:FFTYaxis A$, B$	A$ =G1, G2 B$ = LINMAg, LINREal, LINIMag, LOGMAg, PHASE	Sets the y-axis.	FFT	5-22
:FFTYaxis? A$	A$, B$	Queries the present y-axis setting.		
:FREQ A	A = 80000, 40000, 20000, 8000, 4000, 2000, 800, 400, 200, 80, 40, 20, 8, 4, 2, 0.667, 0.333, 0.133	Sets the frequency range.	FFT	5-24
:FREQ	A<NR3>	Queries the currently set frequency range.		
:OCTFilter A$	A$ = NORMal, SHARp	Sets the type of octave filter.	FFT	5-24
:OCTFilter?	A$	Queries the currently set type of octave filter.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-3. Setting and querying trigger source, level, etc.(TRIGger commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:TRIGger	(ch$ = CH1 to CH8)			
:SOURce A$	A$ = OR, AND	Sets trigger logical operator to AND or OR.	All	5-25
:SOURce?	A$	Queries trigger logical operator (AND or OR).		
:KIND ch$, A$	A$ = OFF, LEVEl, LOGIc, IN, OUT	Sets type of trigger for the indicated channel.	All	5-25
:KIND? ch$	ch$, A$	Queries type of trigger for the indicated channel.		
:LEVEl ch$, A	A$ = trigger level (unit: V)	Sets the trigger level of the indicated channel.	All	5-25
:LEVEl? ch$	ch$, A<NR 3>	Queries the trigger level of the indicated channel.		
:SLOPe ch$, A$	A$ = UP, DOWN	Sets the trigger direction (slope) of the indicated channel.	All	5-26
:SLOPe? ch$	ch$, A$	Queries the trigger direction (slope) of the indicated channel.		
:UPPEr ch$, A	A = upper limit level (unit: V)	Sets upper limit level of window trigger or out trigger.	All	5-26
:UPPEr? ch$	ch$, A<NR 3>	Queries upper limit level of window trigger or out trigger.		
:LOWEr ch$, A	A = lower limit level (unit: V)	Sets lower limit level of window trigger or out trigger.	All	5-27
:LOWEr? ch$	ch$ A <NR3>	Queries lower limit level of window trigger or out trigger.		
:LOGPat ch$, "A$"	A$ = XXXX trigger pattern (X, 0, 1)	Sets the trigger pattern for a logic trigger.	All	5-27
:LOGPat? ch$	ch$ "A$"	Queries the trigger pattern for a logic trigger.		
:LOGAnd ch$, A$	A$ = OR, AND	Sets AND/OR for the logic trigger pattern.	All	5-27
:LOGAnd? ch$	ch$ A$	Queries AND/OR for the logic trigger pattern.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:TRIGger	(ch$ = CH1 to CH8)			
:FILTer ch$, A	A =0 (OFF), 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 5.0, 10.0 (DIV)	Sets filter width of level or logic trigger.	All	5-28
:FILTer? ch$	ch$ A<NR2>	Queries filter width of level or logic trigger.		
:EXTErnal A$	A$ = OFF, ON	Enables and disables external trigger.	All	5-28
:EXTErnal?	A$	Queries external trigger enablement.		
:MODE A$	A$ = SINGle, REPEat (REC) SINGle, REPEat, AUTO (MEM, FFT)	Sets trigger mode.	MEM REC FFT	5-28
:MODE?	A$	Queries trigger mode.		
:PRETrig A	A =0,2, 5, 10, ...90, 95, 100, and -95 (%)	Sets pre-trigger.	MEM FFT	5-29
:PRETrig?	A<NR 1> (unit: %)	Queries pre-trigger		
:TIMIng A$	A$ = START, STOP, S_S	Sets trigger timing.	REC XYC	5-29
:TIMIng?	A$	Queries trigger timing.		
:TIMEr A$	A$ = OFF, ON	Sets timer trigger.	All	5-29
:TIMEr?	A$	Queries timer trigger.		
:TMSTArt month, day, hour, min	month = 1 to 12 day = 1 to 31 hour = 0 to 23 min = 0 to 59	Sets start time of timer trigger.	All	5-30
:TMSTArt?	month, day, hour, min (all <NR 1>)	Queries start time of timer trigger.		
:TMSTOp	Same as :TMSTArt	Sets stop time of timer trigger.	All	5-30
:TMSTOp?	Same as :TMSTArt?	Queries stop time of timer trigger.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:TRIGger	(ch$ = CH1 to CH8)			
:TMINTvl hour, min, sec	hour = 0 to 99 min = 0 to 59 sec = 0 to 59	Sets time interval for timer trigger.	All	5-30
:TMINTvl?	hour, min, sec (all <NR1>)	Queries time interval for timer trigger.		
:DETECTTime hour, min, sec	hour = 0 to 23 min = 0 to 59 sec = 0 to 59	Sets the time point for trigger detection.	All	5-31
:DETECTTime?	hour, min, sec (all <NR1>)	Queries the currently set time point for trigger detection.		
:DETECTDate year, month, day	year = 0 to 99 month= 1 to 12 day = 1 to 31	Sets the date for trigger detection.	All	5-31
:DETECTDate?	year, month, day (all <NR1>)	Queries the currently set date for trigger detection.		
:STOPTime hour, min, sec	hour = 0 to 23 min = 0 to 59 sec = 0 to 59	Sets termination time of start operation.	REC XYC	5-32
:STOPTime?	hour, min, sec (all<NR1>)	Queries termination time of start operation.		
:STOPDate year, month, day	year = 0 to 99 month= 1 to 12 day = 1 to 31	Sets the date of termination.	REC XYC	5-32
:STOPDate?	year, month, day (all <NR1>)	Queries the date of termination.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-4. Setting and querying input channel (UNIT commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:UNIT	(ch$ = CH1 to CH8)			
:RANGe ch$, A	A = voltage range (unit: V) temperature range (units: ˚C)	Sets input channel voltage range	All	5-33
:RANGe? ch$	ch$, A<NR 3>	Queries input channel voltage range.		
:POSItion ch$, A	A = Position value (unit: %)	Sets the origin position for an input channel.	All	5-33
:POSItion? ch$	ch$ A<NR 1>	Queries the origin position for an input channel.		
:COUPling ch$, A$	A$ =GND, DC (8916,8919 units) GND, DC, RMS, R_G (8917 unit)	Sets input channel coupling.	All	5-33
:COUPling? ch$	ch$, A$	Queries input channel coupling.		
:FILTer ch$, A$	A= cutoff frequency (unit Hz)	Sets input channel filter.	All	5-34
:FILTer?ch$	ch$, A<NR 2>	Queries input channel filter.		
:ADJUST		Carries out zero adjustment for the input units.	All	5-34
:SENSor ch$, A$	A$ = K, J, T	Sets the type of a temperature input unit sensor.	All	5-34
:SENSor? ch$	ch$, A$	Queries the type of a temperature input unit sensor.		
:AAFilter ch$, A$	ch$ A$ = OFF, ON	Turns on or off the FFT anti-aliasing filter.	MEM FFT	5-35
:AAFilter? ch$	ch$, A$	Queries the current on or off state of the FFT anti-aliasing filter.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-5. Setting and querying changeover of the screen mode, the waveform display (DISPlay commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:DISPlay	ch$ = CH1 to CH8			
:CHANge A$	A$ =SYSTem, STATus, CHANnel, DISPlay	Changes over the display screen.	All	5-36
:CHANge?	A$	Queries the display screen.		
:DRAWing ch$, A$	A$ =OFF, LIGHT, NORMal, DARK	Sets display and recording intensity for waveform.	MEM REC	5-36
:DRAWing? ch$	ch$, A$	Queries display and recording of a waveform.		
:GRAPh ch$, A	A = 1,2,3,4 (for DUAL format, 1, 2)	Sets waveform display graph in dual and quad format.	MEM REC	5-37
:GRAPh? ch$	ch$, A<NR1>	Queries waveform display graph in dual and quad format.		
:LOGDraw ch$, A$	A$ = OFF, ON In this command ch$ = CHA to CHD	Enables and disables display and recording of logic waveform.	MEM REC	5-37
:LOGDraw? ch$	A$	Queries display and recording enablement of logic waveform.		
:LOGPosi ch$, A	A=1, 2, 3, 4, 5, 6, 7, 8 In this command ch$ = CHA to CHD	Sets the display position of logic waveform.	MEM REC	5-37
:LOGPosi? ch$	A<NR1>	Queries the display position of logic waveform.		
:XMAG A$	(MEM) A$ = X 8, X 4, X 2, X 1, X 1_2, X 1_5, X 1_10, X 1_20 X 1_50, X 1_100, X 1_200, X 1_500, X 1_1000 (REC) A$ = X 1, X1_2, X 1_5, X 1_10 X1_50	Sets the zoom factor on the time axis.	MEM REC	5-38
:XMAG?	A$	Queries the zoom factor on the time axis.		

Command	Data (for a query, response data)	Explanation	Function	Ref page
:DISPlay	ch$ = CH1 to CH8			
:YMAG ch$, A$	A$ = X 1_2, X 1, X 2, X 4, X 8, (SINGLE format)	Sets the zoom factor on the voltage axis.	MEM REC	5-38
:YMAG? ch$	ch$, A$	Queries the zoom factor on the voltage axis.		
:XYDRawing A, B$	A = 1 to 4 B$ = OFF, LIGHT, NORMal, DARK	Sets the drawing level for an X-Y plot.	MEM XYC	5-38
:XYDRawing? A	A<NR1>, B$	Queries the drawing level for an X-Y plot.		
:XAXIs A, ch$	A = 1 to 4	In XY format, sets the X axis.	MEM XYC	5-39
:XAXIs? A	A<NR1>, ch$	In XY format, queries the X axis.		
:YAXIs A, ch$	A = 1 to 4	In XY format, set the Y axis	MEM XYC	5-39
:YAXIs? A	A<NR1>, ch$	In XY format, queries the Y axis		
:WAVE A$	A$ = ACUR (A-cursor), TRIG (trigger point), POINT (the point set with :MEMory:POINt)	Executes waveform display.	MEM	5-39
:XYCLr A$	A$ = OFF, ON	Sets the display clear function in the X-Y recorder function off or on.	XYC	5-40
:XYCLr?	A$	Queries the setting of the display clear function in the X-Y recorder function.		
:VARIable ch$, A$	A$ = ON, OFF	Sets the variable function.	All	5-40
:VARIable? ch$	A$	Queries the variable function.		
:VARIUPLOw ch$, B, C	B=C=-9.9999E+29 to +9.9999E+29	Sets the upper and lower limit values of the variable.	All	5-40
:VARIUPLOw? ch$	B<NR3>	Queries the upper and lower limit values of the variable.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-6. Cursor setting and reading (CURSor commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CURSor	(ch$ = CH1 to CH8)			
:MODE A$	A$ =OFF, TIME, VOLT, (REC) OFF, Xcur, Ycur, TRACe (XYSing, and XYQuad in MEM) OFF, TIME, VOLT, TRACe (MEM excluding XYSing and XYQuad) OFF, Xcur, Ycur (XYC) OFF, TRACe (FFT)	Sets the A and B cursor type.	All	5-40
:MODE?	A$	Queries the A and B cursor type.		
:ABCUrsor A$	A$ = A, B, A_B	Chooses between the A,B and the A&B cursors.	All	5-40
:ABCUrsor?	A$	Queries between the A,B and the A&B cursors.		
:ACHAnnel ch$	For XYformat, ch$ = X1 to X4	Sets the A cursor channel.	All	5-40
:ACHAnnel?	ch$	Queries the A cursor channel.		
:BCHAnnel ch$	For XYformat, ch$ = X1 to X4	Sets the B cursor channel.	All	5-42
:BCHAnnel?	ch$	Queries the B cursor channel.		
:APOSition A	(vertical cursor, trace cursor) A = 0 to amount of stored data: MEM. 0 to amount of stored data:REC 0 to 319:XYC 0 to 999:FFT (STR, ACR, CCR, IMP) 0 to 400:FFT (LIN, RMS, PSP, TRF, COM, CSP, HIS, OCT) (horizontal cursor) A = 0 to 639:MEM,REC 0 to 319:XYC	Sets the position of the A cursor.	All	5-42
:APOSition?	A<NR 1>	Queries the position of the A cursor.		
:BPOSition A	Same as: APOSition	Sets the position of the B cursor.	All	5-42
:BPOSition?	A<NR 1>	Queries the position of the B cursor.		

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CURSor	(ch$ = CH1 to CH8)			
:DTREad? A$	B$ A$=A, B, A-B B$ = readout value (t)	Queries the cursor readout value (t)	All	5-44
:DVREad? A$	B$ A$=A, B, A-B B$ = readout value (V)	Queries the cursor readout value (V)	All	5-44
:ABCHAnnel A$	A$ =G1, G2	Sets the graph for the A and B cursors.	FFT	5-44
:ABCHAnnel?	A$	Queries the current graph setting for the A and B cursors.		
:DFREad? A$	"B, C" A$ = A, B, A_B B = x-axis data C = y-axis data	Queries the current cursor readout position.	FFT	5-45

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-7. Setting and querying input and output, etc of stored data (MEMory commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:MEMory	(ch$ = CH1 to CH8)			
:POINt ch$, A	A = 0 to 1000000	Sets point in memory for input and output.	MEM	5-46
:POINt?	A<NR 1> = 0 to 1000000	Queries point in memory for input and output.		
:MAXPoint?	A<NR 1> = 0: not stored 2500 to 1000000(÷ 100 = number of divisions)	Queries the amount of data stored.	MEM	5-46
:PREPare		Prepares the memory for receipt of waveform data.	MEM	5-46
:ADATa B, C, ...	B, C,...= 0 to 4095	Input data to memory (ASCII).	MEM	5-47
:ADATa? A	A = 1 to 40 (number of output units) B, C,...<NR3> = 0 to 4095	Output data from memory. (ASCII)		
:VDATa B, C,...	B, C,...= voltage values (units: V)	Input data to memory (voltage values).	MEM	5-48
:VDATa? A	A = 1 to 10 (amount of data) B, C,...<NR3>= voltage value (units:V)	Output stored data. (votage values)		
:AREAl? ch$	A<NR 1> = 0 to 4095	Output stored data. Real time data output (ASCII)	MEM	5-49
:VREAl? ch$	A<NR 3> = voltage value (units:V)	Real time data output (voltage value)		
:LDATa A, B,...	A, B,...= 0 to 15	Input logic data to memory.	MEM	5-50
:LDATa? A	A = 1 to 50 (amount of output data) Response data <NR1>= 0 to 15	Output logic data from memory.		
:BDATa? A	A = 1 to 125 (amount od output data) Response data binary, integar.	Output binary data to memory.	MEM	5-51
:FFTData?	"A, B" A = x-axis data <NR3> B = y-axis data <NR3>	Queries FFT data at the output point.	FFT	5-52
:FFTPOint A$, B	A$ = G1, G2 B = 0 to 999 (STR, ACR, CCR, or IMP) 0 to 400 (LIN, RMS, PSP, TRF, COH, CSP, HIS, or OCT)	Sets the output point for FFT data.	FFT	5-52
:FFTPOint?	A$, B<NR3>	Queries the current output point for FFT data.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-8. Setting and querying the system screen (SYSTem commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:SYSTem				
:TIME hour, min, sec	hour = 0 to 23 min = 0 to 59 sec = 0 to 59	Sets the time.	All	5-53
:TIME?	hour, min, and sec (all<NR1>)	Queries the current time.		
:DATE year, month, day	year = 0 to 99 month= 1 to 12 day = 1 to 31	Sets the calendar.	All	5-53
:DATE?	year, month, day (all<NR 1>)	Queries the calendar.		
:DATAClear		Clear waveform data	ALL	5-53
:USECH A	A = 1, 2, 4, 8	Sets number of channels used.	ALL	5-53
:USECH?	A<NR 1>	Queries number of channels used.		
:STARt A$	A$ = ON, OFF	Enables and disables start key backup.	All	5-54
:STARt?	A$	Queries start key backup enablement.		
:GRID A$	A$ = OFF, STANdard, FINE	Sets the grid type.	All	5-54
:GRID?	A$	Queries the grid type.		
:CHMArk A$	A$ = ON, OFF	Enables and disables channel markers.	All	5-54
:CHMArk?	A$	Queries enablement of channel markers.		
:TMAXis A$	A$ =TIME, TIME (60), DIV	Sets the time axis display.	All	5-55
:TMAXis?	A$	Queries the time axis display.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:SYSTem				
:LIST A$	A$ = OFF, LIST, GAUGE, L_G	Sets list and gauge functions.	All	5-55
:LIST?	A$	Queries list and gauge functions.		
:CRTOff A$	A$ = ON, OFF	Enables and disables the screen saver function.	All	5-55
:CRTOff?	A$	Queries enablement of the screen saver function.		
:LCDDisp A$	A$ = NORMal, REVErse	Sets the LCD display.	All	5-56
:LCDDisp?	A$	Queries the LCD dispaly.		
:BEEPer A$	A$ = ON, OFF	Enables and disables the beep sound.	All	5-56
:BEEPer?	A$	Queries beep sound enablement.		
:THINout A$	A$ = ON, OFF	Sets the inter mittent print function.	All	5-56
:THINout?	A$	Queries the inter mittent print function.		
:COPY A$	A$ = PRINter, PLOTter FD_Mono, FD_Color	Sets the destination of hard copy output.	All	5-57
:COPY?	A$	Queries the destination of hard copy output.		
:BMPColor A$, B$, C$, D$	A$, B$, C$, D$ = BLACK, BLUE RED, MAGENTA, GREEN, CYAN, YELLOW, ORANGE	Sets colors of bit map file.	All	5-57
:BMPColor?	A$, B$, C$, D$	Queries colors of bit map file.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-9. Setting and querying scaling (SCALing commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:SCALing				
:KIND A$	A$ = RATIO, POINT	Sets the type of scaling.	All	5-58
:KIND?	A$	Queries the type of scaling.		
:SET ch$, A$	ch$ = CH1 to CH8 A$=OFF,SCI(or ON), ENG	Sets scaling type.	All	5-58
:SET? ch$	A$	Queries scaling type.		
:VOLT A$, B	A$ = CH1 to CH8 B = -9.9999E+9 to +9.9999E+9	Sets the scaling conversion value.	All	5-58
:VOLT? A$	B<NR3>	Queries the scaling conversion value.		
:OFFSet A$, B	A$ =CH1 to CH8 B =-9.9999E+9 to +9.9999E+9	Sets scaling offset.	All	5-59
:OFFSet? A$	B<NR3>	Queries scaling offset.		
:UNIT A$ "B$"	A$ =CH1 to CH8 B$ = scaling unit (7 characters)	Sets scaling unit.	All	5-59
:UNIT A$	"B$"	Queries scaling unit.		
:VOUPLOw ch$, B, C	B=C=-9.9999E+29 to +9.9999E+29	Sets the scaling VOLT UP, VOLT LOw.	All	5-59
:VOUPLOw ch$	B<NR3>	Queries the scaling VOLT UP, VOLT LOw.		
:SCUPLOw ch$, B, C	B=C=-9.9999E+29 to +9.9999E+29	Sets the scaling SC UP, SC LOw.	All	5-60
:SCUPLOw ch$	B<NR3>	Queries the scaling SC UP, SC LOw.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-10. Setting and querying comments (COMMent commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:COMMent				
:TITLe A$, "B$"	A$ = OFF SETTing, COMMent, S_C B$ = comment string (20 characters)	Sets a title comment.	All	5-61
:TITLe?	"B$"	Queries a title comment.		
:EACHch A$	A$ = ON,SETTing, COMMent, S_C	Enables or disables comments for all channels.	All	5-61
:EACHch?	A$	Queries whether comments for all channels are enabled or disabled.		
:CH ch$, "A$"	ch$ = CH1 to CH8; CHA to CHD A$ = comment string (20 characters)	Sets a comment for a particular channel	All	5-62
:CH? ch$	"A$"	Queries comment for a particular channel.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-11. Calculation setting and querying (CALCulate commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CALCulate				
:WVCALc A$	A$ = ON, OFF, EXEC (execute)	Enables and disables waveform processing calculation.	MEM	5-63
:WVCALc?	A$	Queries enablement of waveform processing calculation.		
:Z1 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z1	MEM	5-63
:Z1?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z1		
:Z2 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z2	MEM	5-64
:Z2?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z2		
:Z3 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z3	MEM	5-64
:Z3?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z3		
:Z4 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z4	MEM	5-64
:Z4?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z4		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CALCulate				
:Z5 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z5	MEM	5-65
:Z5?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z5		
:Z6 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z6	MEM	5-65
:Z6?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z6		
:Z7 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z7	MEM	5-65
:Z7?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z7		
:Z8 A$, B$, C$, D$	A$, B$, C$ = A to P D$ =PLUS, MINUS, MULT, DIVI	Sets the coefficients for the waveform processing calculation equation for Z8	MEM	5-66
:Z8?	A$, B$, C$, D$	Queries the coefficients for the waveform processing calculation equation for Z8		
:X1 A$, ch$, B$	A$ = OFF(ch$,B$ are disregarded) PAR, ABS, EXP, LOG, SQR, MOV, SLI, DIF, INT, DIF2, INT2 ch$ = CH1 to CH8 B$ = A to P (when A$ = MOV, a value from 1 to 4000;when SLI, a value from -4000 to 4000)	Sets calculation equation for X1.	MEM	5-66
:X1?	A$, ch$, B$	Queries calculation equation for X1.		
:X2 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1)	Sets calculation equation for X2.	MEM	5-67
:X2?	A$, ch$, B$	Queries calculation equation for X2.		
:X3 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1,Z2)	Sets calculation equation for X3.	MEM	5-67
:X3?	A$, ch$, B$	Queries calculation equation for X3.		
:X4 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1 to Z3)	Sets calculation equation for X4.	MEM	5-67
:X4?	A$, ch$, B$	Queries calculation equation for X4.		

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CALCulate				
:X5 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1 to Z4)	Sets calculation equation for X5.	MEM	5-68
:X5?	A$, ch$, B$	Queries calculation equation for X5.		
:X6 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1 to Z5)	Sets calculation equation for X6.	MEM	5-68
:X6?	A$, ch$, B$	Queries calculation equation for X6.		
:X7 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1 to Z6)	Sets calculation equation for X7.	MEM	5-68
:X7?	A$, ch$, B$	Queries calculation equation for X7.		
:X8 A$, ch$, B$	same as X1 (ch$ = CH1 to CH8, Z1 to Z7)	Sets calculation equation for X8.	MEM	5-69
:X8?	A$, ch$, B$	Queries calculation equation for X8.		
:Y1 A$, ch$, B$	A$ = OFF(ch$,B$ are disregarded) PAR, ABS, EXP, LOG, SQR, MOV, SLI, DIF, INT, DIF2, INT2, ch$ = CH1 to CH8 B$ = A to P(when A$ = MOV, a value from 1 to 4000;when SLI, a value from -4000 to 4000)	Sets calculation equation for Y1.	MEM	5-70
:Y1?	A$, ch$, B$	Queries calculation equation for Y1.		
:Y2 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1)	Sets calculation equation for Y2.	MEM	5-71
:Y2?	A$, ch$, B$	Queries calculation equation for Y2		
:Y3 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1, Z2)	Sets calculation equation for Y3.	MEM	5-71
:Y3?	A$, ch$, B$	Queries calculation equation for Y3.		
:Y4 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1 to Z3)	Sets calculation equation for Y4.	MEM	5-71
:Y4?	A$, ch$, B$	Queries calculation equation for Y4.		
:Y5 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1 to Z4)	Sets calculation equation for Y5.	MEM	5-72
:Y5?	A$, ch$, B$	Queries calculation equation for Y5.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL..... all MEM, REC, XYC and FFT function.

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CALCulate				
:Y6 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1 to Z5)	Sets calculation equation for Y6.	MEM	5-72
:Y6?	A$, ch$, B$	Queries calculation equation for Y6.		
:Y7 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1 to Z6)	Sets calculation equation for Y7.	MEM	5-72
:Y7?	A$, ch$, B$	Queries calculation equation for Y7.		
:Y8 A$, ch$, B$	same as Y1 (ch$ = CH1 to CH8, Z1 to Z7)	Sets calculation equation for Y8.	MEM	5-73
:Y8?	A$, ch$, B$	Queries calculation equation for Y8.		
:FACTor A$, B	A$ = A to P B= -9.999E+9 to +9.999E+9	Sets the value of calculation equation coefficient a to p.	MEM	5-73
:FACTor? A$	B<NR3>	Queries the value of calculation equation coefficient a to p.		
:Z1DIsplay ch$, A$, upper, lower	ch$ = CH1 to CH8 A$ = AUTO, MANUal (for manual) upper, lower = -9.999E+29 to +9.999E+29 (units: V)	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z1.	MEM	5-74
:Z1DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z1.		
:Z2DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z2.	MEM	5-74
:Z2DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z2.		
:Z3DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z3.	MEM	5-75
:Z3DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z3.		

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CALCulate				
:Z4DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z4.	MEM	5-75
:Z4DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z4.		
:Z5DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z5.	MEM	5-75
:Z5DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z5.		
:Z6DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z6.	MEM	5-76
:Z6DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z6.		
:Z7DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z7.	MEM	5-76
:Z7DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z7.		
:Z8DIsplay ch$, A$, upper, lower	Same as Z1DIsplay	Sets the channel for receipt of the calculated result of the waveform treatment calculation equation for Z8.	MEM	5-76
:Z8DIsplay?	ch$, A$, upper, lower	Queries the channel for receipt of the calculated result of the waveform treatment calculation equation for Z8.		

Command	Data (for a query, response data)	Explanation	Function	Ref page
:CALCulate				
:MEASure A$	A$ = ON, OFF, EXEC (execute)	Enables and disables waveform parameter calculation.	MEM	5-77
:MEASure?	A$	Queries enablement of waveform parameter calculation.		
:MEASPrint A$	A$ = OFF, ON	Enables and disables printing of waveform parameter calculation values.	MEM	5-77
:MEASPrint?	A$	Queries enablement of printing of waveform parameter calculation values.		
:MEASSet NO$, A$, ch$	NO$ = NO1 to NO4 A$ =OFF(not when NO$= NO1), MAX, MIN, MAXT, MINT, PP, AVE, RMS, AREA, PERI, FREQ, RISE, FALL, XYAREA ch$ = CH1 to CH8,ALL	Sets waveform parameter calculation.	MEM	5-78
:MEASSet? NO$	A$, ch$	Queries waveform parameter calculation.		
:ANSWer? A$,B$	A$ =NO 1 to NO 4 B$ =CH1 to CH8 C$ =NONE, MIN, MAX, MINT, MAXT, PP, AVE, RMS, AREA, PERI, FREQ, RISE, FALL, XYAREA D<NR3>=calculation result	Queries a waveform parameter calculation result. Waveform parameter calculation result response.	MEM	5-78
:COMP NO$, A$	NO$ = NO1 to NO4 A$ = ON, OFF	Enables or disables waveform parameter decision calculations.	MEM	5-79
:COMP? NO$	A$	Queries enablement of waveform parameter decision calculations.		
:COMPArea NO$, upper, lower	NO$ = NO1 to NO4 upper,lower = -9.999E+9 to +9.999E+9	Sets upper limit and lower limit values for waveform parameter calculation decision.	MEM	5-79
:COMPArea? NO$	upper<NR3>, lower<NR3>	Queries upper limit and lower limit values for waveform parameter calculation decision.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-12. Commands relating to the floppy disk drive (FDISK commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:FDISK				
:MODE A$	A$ = ON, OFF	Enables or disables the floppy disk mode.	All	5-80
:MODE?	A$	Queries enablement of the floppy disk mode		
:LOAD NO, A$	NO = file number A$ = FULL,MINImum,STARt, APPEnd	Executes a load from the floppy disk (in floppy disk mode)	All	5-80
:SAVE "NAME1$.NAME2$", A$, B$,	NAME1$ = file name (8 characters); NAME2$ = file extension (3 characters) A$ = type of data to save W : measurement data (MEM, REC, FFT) F : unit settings A : waveform decision area (MEM, FFT) (when A$ = W) B$ = channels to save ALL, CH1 to CH8 LOG A (logic channel CHA to D)	Performs a save to the floppy disk	All	5-81
:DELEte NO	NO = file number	Deletes a file from the floppy disk.	All	5-81
:FORMat A$	A$ = 2DD, 2HD, 2HC	Formats a floppy disk .	All	5-81
:FILE?	A<NR 1> = number of files	Queries how many files are saved on the floppy disk	All	5-82
:NINFor? NO	"NAME$" NO = file number NAME$ = file name	Queries filename on floppy disk	All	5-82
:INFOr? "NAME$"	NAME$ = file name (response) "NAME$", A, B$, "DATE$", "TIME$", B NAME$ = file name A = file number (if no file exists, then -1) B$ = type of data saved W : measurement data F : conditions of creation A : waveform decision area N : no such file DATE$ = year/month/day of save TIME$ = hour:min:sec of save B = file size	The response from the floppy disk	All	5-82

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

4-2-13. Commands relating to graphics (GRAPh commands)

Command	Data (for a query, response data)	Explanation	Function	Ref page
:GRAPh				
:EDIT A$	A$ = OFF, ON	Enables and disables the editor.	MEM FFT	5-83
:EDIT?	A$	Queries editor enablement.		
: LINE X1, Y1 X2, Y2	X 1, X2 = x-coordinates Y1, Y2 = y-coordinates	Draws a line from (X1, Y1) to (X2, Y2).	MEM FFT	5-83
: PARAllel high, low right, left	high = 0 to 19.97 (div) low = 0 to 19.97 (div) right = 0 to 14.96 (div) left = 0 to 14.96 (div)	Carries out a parallel movement of the drawing.	MEM FFT	5-84
: PAINT X, Y	X = x-coordinate, Y = y-coordinate	Begins solid fill from the point specified by (X, Y).	MEM FFT	5-84
: ERASe X1, Y1 X2, Y2	X 1, X2 = x-coordinates Y1, Y2 = y-coordinates	Erases from (X1, Y1) to (X2, Y2).	MEM FFT	5-84
: STORage		Loads a waveform into the editor.	MEM FFT	5-84
:REVErse		Reverses the drawing	MEM FFT	5-84
:ALLClear		Clears the entire drawing	MEM FFT	5-85
: CLEAr X1, Y1, X2, Y2	X 1, X2 = x-coordinate Y1, Y2 = y-coordinate	Clears the rectangle with the points (X1, Y1) and (X2, Y2) at diagonally opposite corners.	MEM FFT	5-85
: UNDO		Reverses the effect of the immediately previous editor command.	MEM FFT	5-85
: SAVE		Saves the decision area created with the editor.	MEM FFT	5-85
:POINT X,Y, A	X = x-coordinates, Y = y-coordinates A = 0,1	Sets waveform decision area data.	MEM FFT	5-85
:POINT ?	X, Y, A	Queries waveform decision area data.		

MEM ... memory recorder function REC ... recorder function
XYC ... XY recorder function FFT..... FFT recorder function
ALL.... all MEM, REC, XYC and FFT function.

Chapter 5 5

Command Reference

5-1. Command Reference organization

The following sections describe the format and functions of individual commands.

The following is an example of how the descriptions are organized.

❑ Changes and queries the function selection.

Syntax	(command)	: FUNCtion A$
	(query)	: FUNCtion?
	(response)	A$ = MEM : memory recorder function
		REC : recorder function
		XYC : XY recorder function
		FFT : FFT function
Explanation	(command) Switches to the function designated by A$.	
	(query) Returns the name of the current function as character data.	
Example	: FUNCTION MEM	
	The function is set to the memory recorder function.	
When allowed	In all functions.	

① Command function

② Command syntax

(command) gives the syntax of a command program message, (query) the syntax of a query program message, and (response) the format of the response message. The parameters, referred to as data, are shown as follows:

A, B, C,...	Numerical data (e.g. 1.5, 10E-3)
A$, B$, ...	Character data (e.g. A, A_B, C1)
"A", "A$",...	Character string data (e.g. "1.5", "mA")

The format of numerical data follows the formats <NR1>, <NR2>, and <NR3>, described Section 3-3 (8) "Data format". If no format is mentioned, <NRf> format (i.e. any of the above) is accepted.

A <NR1>	:	Numerical parameter in NR1 format
B	:	Numerical parameter in NRf format

NR1 format - integer data.	Examples: +15, -20, 25
NR2 format - fixed point numbers.	Examples: +1.23, -4.56, 7.89
NR3 format - floating point numbers.	Examples: +1.0E-3, -2.3E+3
The term "NRf format" includes all these three formats.	

When the 8840 is receiving a command or query program message, it accepts format, but when it is sending it utilizes whichever one of the formats <NR1> to <NR3> is indicated in the particular command.

Response messages may or may not have headers prefixed, according to the setting made by the :HEADER command.

③ Explanation of the command function.

④ Example of command use.

⑤ This lists the functions in which the command may be used.

MEM	:	memory recorder function
REC	:	recorder function
XYC	:	X-Y recorder function
FFT	:	FFT recorder function
All	:	Any of the MEM, REC, XYC, and FFT functions

Execution of commands

Commands are input into the input buffer and are executed in order.

However the :ABORT command is executed immediately, even if commands are waiting in the input buffer - more precisely, at the instant its terminator is received.

Commands other than those which can be handled by the 8840 in its current state are not executed but generate execution errors. This happens, for example, when in memory recorder function it is attempted to execute an recorder mode setting.

Further, almost all commands cannot be executed during measurement operation.

5-2. Standard Commands Stipulated by IEEE488.2

A. System data commands and queries

*IDN? command

❑ **Queries device ID.**

Syntax (query) *IDN?

(response) HIOKI, 8840, 0, V1.00

First field : Manufacturer's name
Second field: Model name
Third field : Serial number (not used: 0)
Fourth field : Software version

*OPT? command

❑ **Queries device option provision.**

Syntax (query) *OPT?

(response) Whether or not channel 1 input unit present; whether or not channel 2 input unit present; whether or not channel 8 input unit present; memory capacity.

0: not present.
1: analog input unit (8916 analog unit) present.
2: temperature input unit (8918 temperature unit) present.
3: DC/RMS input unit (8917 DC/RMS unit) present.
4: FFT analog unit (8919 FFT analog unit) present.

B. Internal operation commands and queries

*RST command

❑ **Device initial setting.**

Syntax (command) *RST

Explanation Initializes the 8840 (same as system reset).
However, it does not clear GP-IB related items.
(the event registers, the enable registers, etc.)
(the input buffer, the output queue, etc.)

*TST? command

❑ **Queries the result of the self-test.**

Syntax (query) *TST?

(response) A<NR1>
A = 0 : normal, 1: failure

Explanation The result of the self-test of the 8840 is returned as an NR1 numerical value.

C. Synchronous commands and queries

*OPC command

❑ **After all action has been completed during execution, sets the LSB (bit 0) of SESR (the standard event status register).**

Syntax (command) *OPC

Explanation When the command preceding the *OPC command completes execution, the LSB of SESR is set.

Example A$;B$;*OPC;C$
After the execution of the commands A$ and B$ is completed, the LSB of SESR is set.

*OPC? command

❑ **After execution is completed, replies with ASCII [1].**

Syntax (query) *OPC?
(response) 1

Explanation When the command preceding the *OPC command completes execution, the response of ASCII [1] is made.

*WAI command

❑ **After all execution is completed, subsequently performs the following command**

Syntax (command) *WAI

Example A$;B$;*WAI;C$
The command following *WAI is not executed until the execution of the commands A$ and B$ is completed.

D. Status and event control commands and queries

*CLS command

❑ **Clears the status byte and associated queues (except for the output queue).**

Syntax (command) *CLS

Explanation This instruction clears the event register associated with each bit of the status byte register. Accordingly, it also clears the status byte register. However, because it does not clear the output queue, it has no effect upon bit 4 (MAV) of the status byte.

*ESE command

❑ **Writes the standard event status enable register (SESER).**

Syntax (command) *ESE A
A= 0 to 255

Sets the mask pattern of SESER to a value in the range 0 to 255. Outside this range, an execution error occurs. The initial value (when the power is turned on) is 0.

Example *ESE 36
Bit 5 and bit 2 of SESER are set.

*ESE? command

❑ Reads the standard event status enable register (SESER).

Syntax (query) *ESE?
(response) A<NR 1>
A = 0 to 255

Explanation The contents of SESER as set by the *ESE command are returned as an integral value in the range 0 to 255.

*ESR? command

❑ Reads out and queries the contents of the standard event status register (SESR).

Syntax (query) *ESR?
(response) A<NR 1>
A = 0 to 255

Explanation The contents of SESR are returned as an NR1 numerical value.

*SRE command

❑ Writes the service request enable register (SRER).

Syntax (command) *SRE A
A = 0 to 255

Explanation Sets the mask pattern of SRER to a value in the range 0 to 255. Outside this range, an execution error occurs. However, the value of bit 6 is disregarded. The initial value (when the power is turned on) is 0.

Example *SRE 33
Bits 5 and 0 of SRER are set.

*SRE? command

❑ Reads the service request enable register (SRER).

Syntax (query) *SRE?
(responce) A<NR 1>
A = 0 to 63, 128 to 191

Explanation The contents of SRER as set by the *SRE command are returned as an NR1 numerical value in the range 0 to 63, 128 to 191. Bit 6 is always 0.

*STB? command

❑ Reads the status byte and the MSS bit, without performing serial polling.

Syntax (query) *STB?
(response) A<NR 1>
A = 0 to 255

Explanation This is the same as reading out the status byte with serial polling. However, bit 6 is not RQS, but is MSS. (Refer to the description of the status byte and the event register).

<Commands specific to the 8840>

:ESE0 command

❑ Writes event status enable register 0 (ESER0)

Syntax (command) : ESE0 A
A = 0 to 255

Explanation Sets the mask pattern of ESER0 to a value in the range 0 to 255. Outside this range, an execution error occurs. The initial value (when the power is turned on) is 0.

Example :ESE0 36
This sets bit 5 and bit 2 of ESER0.

:ESE0? command

❑ Reads event status enable register 0 (ESER0).

Syntax (query) : ESE0?
(response) A<NR 1>
A = 0 to 255

Explanation The contents of ESER0 are returned as an NR1 numerical value.

:ESR0? command

❑ Reads event status register 0 (ESR0).

Syntax (query) : ESR0?
(response) A<NR 1>
A = 0 to 255

Explanation The contents of ESR0 are returned as an NR1 numerical value, and ESR0 is cleared.

5-3. Commands Specific to the 8840

5-3-1. Execution control commands (common to all functions)

❑ Performs starting.

Syntax	(command) : STARt
Unit	Same as the START key. Starts waveform sampling operation.
When allowed	In all functions.

❑ Performs stopping.

Syntax	(command) : STOP
Unit	Same as the STOP key. Terminates at the instant that waveform sampling operation is completed. (with :STOP command, printer operation is not stopped, use :ABORT command to stop operation)
When allowed	In all functions.

❑ Aborts processing.

Syntax	(command) : ABORT
Unit	Same as the STOP key. Forced halt. Terminates even if waveform sampling operation is not yet completed. Also stops printer operation.
When allowed	In all functions.

❑ Performs printing.

Syntax	(command) : PRINt
Unit	Same as the PRINT key.
When allowed	In all functions.

❑ Screendump function.

Syntax	(command) : HCOPy
Unit	Same as the COPY key. Produces a hard copy of the screen.
When allowed	In all functions.

❑ Feeds printer paper.

Syntax	(command)	: FEED A A= 1 to 255
Explanation	Feeds the paper by a distance from 1 to 255 in millimeters determined by the numerical value in the data portion.	
When allowed	In all functions.	

❑ Performs automatic range setting.

Syntax	(command)	: AUTO
Unit	Same as the AUTO key. Sets the time axis and the voltage axis automatically.	
When allowed	In the memory recorder function.	

❑ Enables and disables LCD back light, and queries back light enablement.

Syntax	(command) (query) (response)	: LIGHt A$: LIGHt? A$ A$=OFF, ON
Explanation	(command) (query)	Sets the LCD back light key to on or off. Returns the setting of the current back light key as character data.
When allowed	In the memory recorder function.	

❑ Queries the 8840 error number.

Syntax	(query) (response)	: ERRor? A<NR 1> A = error number
Explanation	The type of error or the number of warning that has occurred on the 8840 is returned as NR1 numerical value. (Refer to 8840 manual appendix 1) If an error occurs during execution of :ERROR? then the error number is cleared.	
When allowed	In all functions.	

Execution control commands

❑ Enables and disables headers, and queries header enablement.

Syntax

(command) : HEADer A$
(query) : HEADer?
(response) A$
A$ = OFF,ON

Explanation

(command) Sets header enablement. When headers are enabled, responses to queries are prefixed by headers; when headers are disabled, responses are not so prefixed.

(query) Returns whether or not headers are prefixed to responses to queries. The initial toggle state for headers (when the power is turned on) is OFF.

Example

When headers are disabled:
response to :HEADER? is OFF
When headers are enabled:
response to :HEADER? is :HEADER ON

When allowed

In all functions.

❑ Changes and queries the function selection.

Syntax

(command) : FUNCtion A$
(query) : FUNCtion?
(response) A$
A$ = MEM : memory recorder function
REC : recorder function
XYC : XY recorder function
FFT : FFT recorder function

Explanation

(command) Switches to the function designated by A$.
(query) Returns the name of the current function as character data.

Example

:FUNCTION MEM
The function is set to the memory recorder function.

When allowed

In all functions.

❑ A4 print

Syntax

(command) : A4 PRint

Explanation

(command) Same as the FEED key + COPY key on the 8840 unit.

When allowed

In all functions.

5-3-2. CONFigure command (Sets and queries time axis range, recording length, etc)

❑ Sets and queries the time/div .

Syntax	(command)	: CONFigure : TDIV A
	(query)	: CONFigure : TDIV?
	(response)	A<NR3>
Explanation	(command)	Sets the time per division to a numerical value (unit seconds).
	(query)	Returns the currently set value of the time per division as an NR3numerical value. (If an attempt is made to set the time per division to a non-permitted value, it will be set to the next range above that value.)
Example	: CONFIGURE : TDIV +500.0E-6 Sets the time per division to 500 µs.	
When allowed	In the memory recorder function and the recorder function.	

❑ Sets and queries the recording length.

Syntax	(command)	: CONFigure : SHOT A
	(query)	: CONFigure : SHOT?
	(response)	A<NR1>
Explanation	(command)	Sets the numerical value of the recording length (unit divisions).
	(query)	Returns the currently set value of the recording length as an NR1 numerical value. (For the recorder function, 0 = CONT).
Example	: CONFIGURE: SHOT 25 Sets the recording length to 25 divisions.	
When allowed	In the memory recorder function and the recorder function.	

❑ Sets and queries the format.

Syntax	(command)	: CONFigure : FORMat A$
	(query)	: CONFigure : FORMat?
	(response)	A$ A$ = SINGle, DUAL, QUAD, OCT, XYSingle XYDual :MEM, SINGle, DUAL, QUAD, OCT, : REC, XYSingle, XYDual : XYC SINGle, DUAL, NYQuist : FFT
Explanation	(command)	Sets the format.
	(query)	Returns the current format as character data.
Example	:CONFIGURE :FORMAT SINGLE Sets the format to SINGLE.	
When allowed	In all functions.	

CONFigure

❑ Sets and queries the interpolation function.

Syntax	(command)	: CONFigure : DOTLine A$
	(query)	: CONFigure : DOTLine?
	(response)	A$ A$ = DOT, LINE
Explanation	(command)	Sets the interpolation function (DOT or LINE).
	(query)	Returns the currently set interpolation as character data.
Example	: CONFIGURE : DOTLINE DOT Sets the interpolation function to DOT.	
When allowed	In the memory recorder function, the XY recorder function, and the FFT function.	

❑ Enables and disables, and queries, the roll mode function.

Syntax	(command)	: CONFigure : ROLL A$
	(query)	: CONFigure : ROLL?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables and disables the roll mode function.
	(query)	Returns the current enablement state of the roll mode function as character data.
Example	: CONFIGURE : ROLL ON Sets the roll mode function to ON.	
When allowed	In the memory recorder function.	

❑ Sets and queries the waveform superimposition function.

Syntax	(command)	: CONFigure : OVWRite A$
	(query)	: CONFigure : OVWRite?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables and disables screen waveform superimposition.
	(query)	Returns the current setting of the waveform superimposition enablement as character data.
Example	: CONFIGURE : OVWRITE ON Sets the screen waveform superimposition to ON.	
When allowed	In the memory recorder function.	

❑ Sets and queries the printer output style.

Syntax
(command) : CONFigure : PRKInd A$
(query) : CONFigure : PRKInd?
(response) A$
A$ =WAVE (waveform)
LOGging (numerical)

Explanation
(command) Sets the printer output style to be waveform or logging (numerical data).
(query) Returns the current setting of the printer output style.

Example
: CONFIGURE : PRKIND WAVE
Sets the printer output style to be waveform.

When allowed
In the memory recorder function, the recorder function, and the FFT function.

❑ Sets and queries the logging output interval.

Syntax
(command) : CONFigure : LOGGing A
(query) : CONFigure : LOGGing?
(response) A< NR2 >
A = 0.01 to 100 (memory recorder function)
1 to 100 (recorder function)

Explanation
(command) Sets the numerical value of the logging output interval.
(query) Returns the numerical value of the logging output interval.
For the memory recorder function the values are 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100 (DIV)
For the recorder function the values are 1, 2, 5, 10, 20, 50, 100 (DIV)

Example
: CONFIGURE : LOGGING 10
Sets the logging output internal to 10 (DIV).

When allowed
In the memory recorder function and the recorder function.

❑ Sets and queries printer output.

Syntax
(command) : CONFigure : PRINt A$
(query) : CONFigure : PRINt?
(response) A$
A$ = OFF
ON

Explanation
(command) Sets the printer output.
(query) Returns the currently set state of the printer output as character data.

Example
: CONFIGURE : PRINT ON
Sets the printer output to ON.

When allowed
In the recorder function.

CONFigure

❑ Sets and queries the auto print function.

Syntax	(command)	: CONFigure : SMOOth A$
	(query)	: CONFigure : SMOOth?
	(response)	A$
		A$ = OFF,ON
Explanation	(command)	Enables and disables the smooth printing function.
	(query)	Returns the current enablement state of the smooth printing function as character data.
Example	: CONFIGURE : SMOOTH ON Sets the smooth printing function to ON.	
When allowed	In the memory recorder function.	

❑ Sets and queries the auto print function.

Syntax	(command)	: CONFigure : ATPRint A$
	(query)	: CONFigure : ATPRint?
	(response)	A$
		A$ = OFF,ON
Explanation	(command)	Toggles the auto print function on and off.
	(query)	Returns the current setting of the auto print function as character data.
Example	: CONFIGURE : ATPRINT ON Sets the auto print function to ON.	
When allowed	In the memory recorder function and the FFT function.	

❑ Sets and queries the auto save function.

Syntax	(command)	: CONFigure : ATSAve A$
	(query)	: CONFigure : ATSAve?
	(response)	A$
		A$ = OFF,ON
Explanation	(command)	Toggles the auto save function on and off.
	(query)	Returns the current setting of the auto save function as character data.
Example	: CONFIGURE : ATSAVE ON Sets the auto save function to ON.	
When allowed	In the memory recorder function and the FFT function.	

❑ Sets and queries memory division.

Syntax
(command) : CONFigure : MEMDiv A$
(query) : CONFigure : MEMDiv?
(response) A$
A$ = OFF
SEQ : sequential save
MULTI : multi-block

Explanation
(command) Sets the method of memory division recording.
(query) Returns the current setting for method of memory division recording as character data.

Example
: CONFIGURE : MEMDIV SEQ
Sets the method of memory division recording to the sequential save.

When allowed
In the memory recorder function.

❑ Sets and queries the number of memory blocks.

Syntax
(command) : CONFigure : MAXBlock A
(query) : CONFigure : MAXBlock?
(response) A<NR1>
A=2 to 31 (during multi-block operation, 3,7,15, or 31)

Explanation
(command) Sets the number of memory blocks.
(query) Returns the current number of memory blocks as an NR1 numerical value.

Example
: CONFIGURE : MAXBLOCK 15
Sets the number of memory blocks to 15.

When allowed
In the memory recorder function, when the memory division function is in use.

❑ Sets and queries the division block used.

Syntax
(command) : CONFigure : USEBlock A
(query) : CONFigure : USEBlock?
(response) A<NR1>
A = 1 to number of memory divisions

Explanation
(command) During memory division, sets the block used.
(query) Returns the currently used block as an NR1 numerical value.

Example
: CONFIGURE : USEBLOCK 15
Sets the block used to 15.

When allowed
In the memory recorder function, when the memory division function is in use.

CONFigure

❑ Sets and queries the reference block.

Syntax	(command)	: CONFigure : REFBlock A
	(query)	: CONFigure : REFBlock?
	(response)	A<NR 1> A = 1 to number of memory divisions 0 = OFF
Explanation	(command)	In multi-block mode, sets the reference block.
	(query)	Returns the current reference block as an NR1 numerical value.
Example	: CONFIGURE : REFBLOCK 15 Sets the reference block to 15.	
When allowed	In the memory recorder function, when the memory division multi-block function is in use.	

❑ Sets and queries the waveform decision mode.

Syntax	(command)	: CONFigure : WVCOmp A$
	(query)	: CONFigure : WVCOmp?
	(response)	A$ A$ = OFF, OUT, ALLOUT
Explanation	(command)	Sets the waveform decision mode.
	(query)	Returns the current waveform decision mode as character data.
Example	: CONFIGURE : WVCOMP OUT Sets the waveform decision mode to OUT.	
When allowed	In the memory recorder function and the FFT function.	

❑ Sets and queries the waveform decision stop mode.

Syntax	(command)	: CONFigure : CMPStop A$
	(query)	: CONFigure : CMPStop?
	(response)	A$ A$= GO, NG, G_N
Explanation	(command)	Sets the stop mode during waveform decision.
	(query)	Returns the current stop mode as character data.
Example	: CONFIGURE : CMPSTOP GO Sets the stop mode during waveform decision to GO.	
When allowed	In the memory recorder function and the FFT function.	

❑ Sets and queries the count for averaging.

Syntax
(command) : CONFigure : AVERage A
(query) : CONFigure : AVERage?
(response) A <NR1>
A = 0 : OFF
= 2, 4, 8, 16, 32, 64, 128, 256

Explanation
(command) Sets the count for averaging.
(query) Returns the current setting of the count for averaging as NR1 numerical value.

Example
: CONFIGURE : AVERAGE 32
Sets the count for averaging to 32.

When allowed
In the memory recorder function.

❑ Sets and queries the count for averaging in the FFT function.

Syntax
(command) : CONFigure : FFT AVERage A
(query) : CONFigure : FFT AVERage?
(response) A <NR1>
A = 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096

Explanation
(command) Sets the count for averaging in the FFT function.
(query) Returns the current setting of the count for averaging in the FFT function as NR1 numerical values.

Example
: CONFIGURE : FFT AVERAGE 2048
Sets the count for averaging to 2048.

When allowed
In the FFT function.

❑ Sets and queries the type of averaging in the FFT function.

Syntax
(command) : CONFigure : FFTAVKind A$
(query) : CONFigure : FFTAVKind?
(response) A$

A$ =	OFF	:off
	T_LIN	:simple time axis averaging
	T_EXP	:exponential time axis averaging
	F_LIN	:simple frequency axis averaging
	F_EXP	:exponential frequency axis averaging
	F_PEAK	:frequency axis peak hold

Explanation
(command) Sets the averaging method designated by A$.
(query) Returns the currently set averaging method as character data.

Example
: CONFIGURE : FFTAVKIND T_EXP
Sets time axis exponential averaging.

When allowed
In the FFT function.

CONFigure

❑ Sets and queries the FFT channel mode.

Syntax

(command) : CONFigure : FFTMode A, ch1$(,ch2$)
(query) : CONFigure : FFTMode?
(response) A<NR1>, ch1$, ch2$

A = 1	:one-channel FFT mode
A = 2	:two-channel FFT mode
ch1$ = CH1 to CH8	:analysis channel W1
ch2$ = CH1 to CH8	:analysis channel W2

Explanation

(command) Sets the FFT channel mode. I.e., designates the object channel or channels for FFT channel mode and the number thereof. In the one-channel FFT mode (only) the specification of channel 2 can be omitted, and if it is provided it is ignored. Transfer function, coherence function, cross power spectrum, cross correlation function, and impulse response are only effective in the two-channel FFT mode.

(query) Returns the current FFT channel mode as a numerical value in NR1 format, and the analysis channel as character data.

Example

: CONFIGURE:FFTMODE 2, CH3, CH5
The channel mode is set to the two-channel FFT mode, and the object channels for FFT mode are set to be channel 3 and channel 5.

When allowed In the FFT function.

❑ Sets and queries the FFT window function.

Syntax

(command) : CONFigure : FFTWind A$ (,B)
(query) : CONFigure : FFTWind?
(response) A$, B<NR1>

A$ =	RECTan	:rectangular window
	HANNing	:Hanning window
	EXPOnential	:exponential function window
B =0 to 99 (units %)		:coefficient for the exponential function

Explanation

(command) Sets the window function as indicated by A$. If the exponential window function is designated by A$, its exponential function coefficient can be set by using B

(query) Returns the current window function as character data, and the current exponential function coefficient as a numerical value in NR1 format. If the window function is currently rectangular window or Hanning window, B is returned as a dummy zero.

Example

: CONFIGURE : FFTWIND HANN
The window function is set to Hanning window.

When allowed In the FFT function.

❑ Sets and queries the FFT analysis mode.

Syntax

(command) : CONFigure : FFTFunction A$, B$
(query) : CONFigure : FFTFunction? A$
(response) A$, B$

A$ = G1, G2:graph number
B$ = STR :stored waveform
LIN : linear spectrum
RMS :RMS spectrum
PSP : power spectrum
ACR : auto-correlation function
HIS :histogram
TRF : transfer function
CSP :cross power spectrum
CCR :cross correlation function
IMP : impulse response
COH : coherence function
OCT :octave analysis

(TRF, CSP, CCR, IMP, COH: can only be used when the two-channel FFT mode is set.)

Explanation

(command) Sets the FFT analysis mode.
The FFT analysis mode can be set to transfer function, cross power spectrum, cross correlation function, or impulse response only in the two-channel FFT mode (FFTMODE 2, ch1$, ch2$). In this case, the corresponding function is calculated from channel 1 and channel 2. The result of the calculation is displayed on the graph designated by A$. G2 can be designated even if the display format is SINGLE, but this does not affect the display.

(query) Returns the current FFT analysis mode as character data.

Examples

```
: CONFIGURE : FORMAT DUAL
: CONFIGURE : FFTMODE 2, CH1, CH3
: CONFIGURE : FFTFUNCTION G1, IMP
: CONFIGURE : FFTFUNCTION G2, TRF
```

The impulse response calculated from channel 1 and channel 3 is displayed on G1, and the transfer function calculated from these channels is displayed on G2.

When allowed In the FFT function.

CONFigure

❑ Sets and queries the FFT data source.

Syntax
(command) : CONFigure : FFTRef A$
(query) : CONFigure : FFTRef?
(response) A$
A$ = NEW :new data
MEM :data stored in the memory

Explanation
(command) Designates the source for FFT data as specified by A$.
(query) Returns the current FFT data source as character data.

Example
: CONFIGURE : FFTREF NEW
New data is used as FFT data.

When allowed
In the FFT function.

❑ Sets and queries the FFT display scaling method.

Syntax
(command) : CONFigure : FFTSCale A$, B$
(query) : CONFigure : FFTSCale? A$
(response) A$, B$
A$ = G1, G2
B$ = AUTO, MANUal

Explanation
(command) Sets the display scaling method for the graph number designated by A$.
(query) Returns the current display scaling method for the graph number designated by A$ as character data.

Example
: CONFIGURE : FFTSCALE G1, AUTO
The scaling method for graph number 1 is set to automatic.

When allowed
In the FFT function.

❑ Sets and queries the FFT display scale vertical axis upper limit.

Syntax
(command) : CONFigure : FFTUp A$, B
(query) : CONFigure : FFTUp? A$
(response) A$, B <NR3>
A$ = G1, G2
B = -9.999E+29 to +9.999E+29

Explanation
(command) Sets the FFT display scale vertical axis upper limit for the graph number designated by A$ to the value designated by B.
(query) Returns the current FFT display scale vertical axis upper limit for the graph number designated by A$ as a numerical value in NR3 format.

Example
: CONFIGURE : FFTUP G2, 100
The FFT display scale vertical axis upper limit for graph 2 is set to 100.

When allowed
In the FFT function.

❑ Sets and queries the FFT display scale vertical axis lower limit.

Syntax
(command) : CONFigure : FFTLow A$, B
(query) : CONFigure : FFTLow? A$
(response) A$, B<NR3>
A$ = G1, G2
B = -9.999E+29 to +9.999E+29

Explanation
(command) Sets the FFT display scale vertical axis lower limit for the graph number designated by A$ to the value designated by B.
(query) Returns the current FFT display scale vertical axis lower limit for the graph number designated by A$ as a numerical value in NR3 format.

Example
: CONFIGURE : FFTLOW G2, 100
The FFT display scale vertical axis lower limit for display graph 2 is set to 100.

When allowed
In the FFT function.

❑ Sets and queries the FFT x-axis.

Syntax
(command) : CONFigure : FFTXaxis A$, B$
(query) : CONFigure : FFTXaxis? A$
(response) A$, B$
A$ = G1, G2
B$ = 1_1oct, 1_3oct :during octave analysis
LINhz, LOGhz :otherwise

Explanation
(command) Sets the x-axis of the graph number designated by A$. When the analysis mode is octave analysis, 1_1oct or 1_3oct can be set; otherwise, LINhz or LOGhz can be set. Some settings are not available for some analysis modes. If a setting is not available, an execution error is generated (see the table on the next page).
(query) Returns the present x-axis setting as character data.

Example
: CONFIGURE : FFTXAXIS G1, LINHZ
The setting for the x-axis of graph 1 is set to LINHZ.

When allowed
In the FFT function.

CONFigure

❑ Sets and queries the FFT y-axis.

Syntax

(command) : CONFigure : FFTYaxis A$, B$
(query) : CONFigure : FFTYaxis? A$
(response) A$, B$
A$= G1, G2
B$= LINMAg :linear magnitude
LINREal :linear real axis magnitude
LINIMag :linear imaginary axis magnitude
LOGMAg :logarithmic magnitude
PHASE :phase

Explanation

(command) Sets the y-axis of the graph number designated by A$. Some settings are not available for some analysis modes. If a setting is not available, an execution error is generated (see the table on the next page).

(query) Returns the present y-axis setting as character data.

Example

: CONFIGURE : FFTYAXIS G1,LINMAG
The setting for the y-axis of graph 1 is set to LINMAG.

When allowed In the FFT function.

Display settings available on the x-axis

Analysis mode	X-axis				
	Linear-Hz	Log-Hz	1/10 octave	1/3 octave	Fixed scale
STR					TIME
LIN	Yes	Yes			
RMS	Yes	Yes			
PSP	Yes	Yes			
ACR					TIME
HIS					VOLT
TRF	Yes	Yes			
CSP	Yes	Yes			
CCR					TIME
IMP					TIME
COH	Yes	Yes			
OCT			Yes	Yes	

Display settings available on the y-axis

Analysis mode	Y-axis					
	Linear-real	Linear-imaginary	Linear-magnitude	Log-magnitude	Phase	Fixed scale
STR						LINEAR
LIN	Yes	Yes	Yes	Yes	Yes	
RMS	Yes	Yes	Yes	Yes	Yes	
PSP			Yes	Yes		
ACR						LINEAR
HIS						LINEAR
TRF	Yes	Yes	Yes	Yes	Yes	
CSP	Yes	Yes	Yes	Yes	Yes	
CCR						LINEAR
IMP						LINEAR
COH						LINEAR
OCT			Yes	Yes		

CONFigure

❑ Sets and queries the FFT frequency range.

Syntax

(command) : CONFigure : FREQ A
(query) : CONFigure : FREQ?
(response) A<NR3>
A = 80000, 40000, 20000, 8000, 4000, 2000, 800, 400, 200, 80, 40, 20, 8, 4, 2, 0.667, 0.333, 0.133 (units: Hz)

Explanation

(command) Sets the frequency range. If an attempt is made to set an unacceptable value, i.e. a value which is not one of the above, then the frequency range is set to the next higher one of the above values.

(query) Returns the currently set frequency range as a numerical value in NR3 format.

Example

: CONFIGURE : FREQ 80
The frequency range is set to 80 Hz.

When allowed

In the FFT function.

❑ Sets and queries octave filter type.

Syntax

(command) : CONFigure : OCTFilter A$
(query) : CONFigure : OCTFilter?
(response) A$
A$ = NORMal, SHARp

Explanation

(command) Sets the type of octave filter.
(query) Returns the currently set type of octave filter as character data.

Example

: CONFIGURE : OCTFILTER NORMAL
Sets the octave filter type to NORMAL.

When allowed

In the FFT function.

5-3-3. TRIGger command (Sets and queries trigger source, level etc)

❑ Sets and queries trigger logical operator (AND/OR).

Syntax	(command)	: TRIGger : SOURce A$
	(query)	: TRIGger : SOURce?
	(response)	A$
		A$ = OR,AND
Explanation	(command) and (query)	Sets the logical operator determining whether the internal external triggers are ORed or ANDed. Returns the currently setting of the trigger logical operator (AND/OR) as character data.
Example	: TRIGGER : SOURCE OR Sets the trigger sourse to OR.	
When allowed	In all functions.	

❑ Sets and queries the kind of trigger.

Syntax	(command)	: TRIGger : KIND ch$, A$
	(query)	: TRIGger : KIND? ch$
	(response)	ch$, A$
		ch$ = CH1 to CH8 (for logic trigger, CH1 to CH4)
		A$ = OFF
		LEVEl: level trigger
		IN: window-in trigger
		OUT : window-out trigger
		LOGIc: logic trigger
Explanation	(command) (query)	Sets the type of trigger for the channel designated by ch$. Returns as character data the type of the current trigger for the channel designated by ch$.
Example	: TRIGGER : KIND CH1, LEVEL Sets channel 1 to level trigger.	
When allowed	In all functions.	

❑ Sets and queries trigger level.

Syntax	(command)	: TRIGger : LEVEl ch$, A
	(query)	: TRIGger : LEVEl? ch$
	(response)	ch$, A<NR3>
		ch$ = CH1 to CH8
		A = voltage value (unit V), temperature input unit (°C)
Explanation	(command)	Sets the trigger level of the level, of the channel designated by ch$.
	(query)	Returns the current trigger level as an NR3 numerical value.
Example	: TRIGGER : LEVEL CH1, 50E-3 Sets the trigger level of channel 1 to 50mV.	
When allowed	In all functions.	
Note	On the 8840 the trigger level can be set in units of 1% of full scale. For example, for 1 V full scale (50 mV/division), the steps are 10 mV, and if a command indicates a setting not corresponding to one of these steps (for example 12 mV), the 8840 automatically rounds the value (12 mV - 10 mV).	

TRIGger

❑ Sets and queries trigger direction (slope).

Syntax	(command)	: TRIGger : SLOPe ch$, A$
	(query)	: TRIGger : SLOPe? ch$
	(response)	ch$, A$ ch$ = CH1 to CH8 A$ = UP (rising) DOWN (falling)
Explanation	(command)	Sets the trigger direction of the level, of the channel designated by ch$.
	(query)	Returns the current trigger direction as a character value.
Example	: TRIGGER : SLOPE CH1, UP Sets the trigger direction of channel 1 to rising.	
When allowed	In all functions.	

❑ Sets and queries upper limit level for a window trigger.

Syntax	(command)	: TRIGger : UPPEr ch$, A
	(query)	: TRIGger : UPPEr? ch$
	(response)	ch$, A<NR3> ch$ = CH1 to CH8 A = voltage value unit (V), temperature input unit (°C)
Explanation	(command)	Sets the upper limit level of the window trigger of the channel designated by ch$ to a numerical value in the range from the upper limit level to voltage value.
	(query)	Returns the current upper limit level as an NR3 numerical value.
Example	: TRIGGER : UPPER CH1, +1. 0E-3 Sets the upper limit level for channel 1 to +1. 0mV.	
When allowed	In all functions.	
Note	On the 8840 the upper limit level of the window-in and out trigger can be set in units of 1% of full scale. Therefore the 8840 automatically rounds the value as note on the previous page.	

❑ Sets and queries lower limit level for a window trigger.

Syntax	(command)	: TRIGger : LOWEr ch$, A
	(query)	: TRIGger : LOWEr? ch$
	(response)	ch$, A<NR3> ch$ = CH1 to CH8 A = voltage value (unit V), temperature input unit (°C)
Explanation	(command)	Sets the lower limit level of the window trigger of the channel designated by ch$ to a numerical value in the range from zero to the lower limit level voltage value.
	(query)	Returns the current lower limit level as an NR3 numerical value.
Example	: TRIGGER : LOWER CH1, -1. 0E-3 Sets the lower limit level for channel 1 to -1. 0mV.	
When allowed	In all functions.	
Note	On the 8840 the lower limit level of the window-in and out trigger can be set in units of 1% of full scale. Therefore the 8840 automatically rounds the value as note on the previous page.	

❑ Sets and queries the trigger pattern for a logic trigger.

Syntax	(command)	: TRIGger : LOGPat ch$, "A$"
	(query)	: TRIGger : LOGPat? ch$
	(response)	ch$, "A$" ch$ = CH 1 to CH 4 A$ = XXXX :trigger pattern (X,0,1)
Explanation	(command)	Sets the trigger pattern for the logic trigger of the channel designated by ch$ to that specified by the given character data. Replacing double quotation marks (") in commands, single quotation marks (') can be also used.
Example	: TRIGGER : LOGPAT CH1, "X001" Sets the trigger pattern for channel 1 to "X001".	
When allowed	In all functions.	

❑ Sets and queries the logical operator (AND/OR) for the trigger pattern of a logic trigger.

Syntax	(command)	: TRIGger : LOGAnd ch$, A$
	(query)	: TRIGger : LOGAnd? ch$
	(response)	ch$, A$ ch$ = CH 1 to CH 4 A$ = OR,AND
Explanation	(command)	Sets the AND/OR logical operator for the trigger pattern of a logic trigger.
	(query)	Returns the present AND/OR setting as a character string.
Example	: TRIGGER : LOGAND CH1, OR Sets the AND/OR for the trigger pattern for channel 1 to OR.	
When allowed	In all functions.	

TRIGger

❑ Sets and queries filter.

Syntax	(command)	: TRIGger : FILTer ch$, A
	(query)	: TRIGger : FILTer? ch$
	(response)	ch$ A<NR2> ch$ = CH1 to CH8 A = 0 : OFF 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 5.0, 10.0
Explanation	(command)	Sets the filter width for a level trigger of the channel designated by ch$ to a numerical value from 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 5.0 and 10.0 0. In the recorder and XYC recorder modes, however, only an on/off setting for the filter is available. Setting any value of 0.1 or more enables the filter. If the value is zero, then the filter is disabled.
	(query)	Returns the current filter width as an NR1 numerical value.
Example	: TRIGGER : FILTER CH1, 0.1 Sets the filter width for a trigger of channel 1 to 0.1 (DIV).	
When allowed	In all functions.	

❑ Sets and queries external trigger.

Syntax	(command)	: TRIGger : EXTErnal A$
	(query)	: TRIGger : EXTErnal?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables and disables external trigger.
	(query)	Returns the current external trigger enablement state as character data.
Example	:TRIGGER:EXTERNAL OFF Sets the external trigger to OFF.	
When allowed	In all functions.	

❑ Sets and queries trigger mode.

Syntax	(command)	: TRIGger : MODE A$
	(query)	: TRIGger : MODE?
	(response)	A$ A$ = SINGle, REPEat, AUTO :MEM SINGle, REPEat :REC SINGle, REPEat, AUTO :FFT
Explanation	(command)	Sets the trigger mode.
	(query)	Returns the current trigger mode as character data.
Example	: TRIGGER : MODE REPEAT Sets the trigger mode to repeat.	
When allowed	In the memory recorder function, the recorder function, an the FFT function.	

❑ Sets and queries pre-trigger.

Syntax	(command)	: TRIGger : PRETrig A
	(query)	: TRIGger : PRETrig?
	(response)	A<NR1> A = 0, 2, 5, 10, 20.....80, 90, 100, -95 (unit %)
Explanation	(command)	Sets pre-trigger value to a numerical value (in percent). If an attempt is made to set a value which cannot be set on the 8840, setting is performed to the next higher permitted value.
	(query)	The currently set pre-trigger value is returned as an NR1 numerical value.
Example	: TRIGGER : PRETRIG 10 Pre-trigger value is set to 10%.	
When allowed	In the memory recorder function and the FFT function.	

❑ Sets and queries trigger timing.

Syntax	(command)	: TRIGger : TIMIng A$
	(query)	: TRIGger : TIMIng?
	(response)	A$ A$ = START STOP S_S (START&STOP)
Explanation	(command)	Sets the trigger timing.
	(query)	The currently set trigger timing is returned as character data.
Example	: TRIGGER : TIMING START Sets the trigger timing to START.	
When allowed	In the recorder function and the XYC recorder function.	

❑ Sets and queries whether the timer trigger is on or off.

Syntax	(command)	: TRIGger : TIMEr A$
	(query)	: TRIGger : TIMEr?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables or disables the timer trigger.
	(query)	Returns the current enablement state of the timer trigger as character data.
Example	: TRIGGER : TIMER ON Sets the timer trigger to ON.	
When allowed	In all functions.	

TRIGger

❑ Sets and queries the start instant for the timer trigger.

Syntax	(command)	: TRIGger : TMSTArt month, day, hour, min
	(query)	: TRIGger : TMSTArt?
	(response)	month, day, hour, min month = 1 to 12 day = 1 to 31 hour = 0 to 23 min = 0 to 59
Explanation	(command)	Sets the start instant for the timer trigger.
	(query)	Returns the current setting for the timer trigger start instant as NR1 numerical values.
Example	: TRIGGER : TMSTART 7, 5, 9, 30 Sets the start instant for the timer trigger to 9:30 on July 5th.	
When allowed	In all functions.	

❑ Sets and queries the stop instant for the timer trigger.

Syntax	(command)	: TRIGger : TMSTOp month, day, hour, min
	(query)	: TRIGger : TMSTOp?
	(response)	month, day, hour, min month = 1 to 12 day = 1 to 31 hour = 0 to 23 min = 0 to 59
Explanation	(command)	Sets the stop instant for the timer trigger.
	(query)	Returns the current setting for the timer trigger stop instant as NR1 numerical values.
Example	: TRIGGER : TMSTOP 7, 5, 10, 30 Sets the stop instant for the timer trigger to 10:30 on July 5th.	
When allowed	In all functions.	

❑ Sets and queries the time interval for the timer trigger.

Syntax	(command)	: TRIGger : TMINTvl hour, min, sec
	(query)	: TRIGger : TMINTvl?
	(response)	hour, min, sec hour = 0 to 99 min = 0 to 59 sec = 0 to 59
Explanation	(command)	Sets the time interval for the timer trigger.
	(query)	Returns the current setting for the timer trigger time interval as NR1 numerical values.
Example	: TRIGGER : TMINTVL 1, 20, 30 Sets the time interval for the timer trigger to one hour, twenty minutes, and thirty seconds.	
When allowed	In all functions.	

❑ Sets and queries the time point for trigger detection.

Syntax	(command)	: TRIGger : DETECTTime hour, min, sec
	(query)	: TRIGger : DETECTTime?
	(response)	hour, min, sec hour = 0 to 23 min = 0 to 59 sec = 0 to 59
Explanation	(command)	Sets the time point for trigger detection.
	(query)	Returns the currently set time point for trigger detection as a numerical value in NR1 format. During memory partitioning, the time point for the memory block which is currently being displayed (the block in use) is the one referenced.
Example	: TRIGGER : DETECTTIME? The currently set time point for trigger detection is queried.	
When allowed	In all functions.	

❑ Sets and queries the date for trigger detection.

Syntax	(command)	: TRIGger : DETECTDate year, month, day
	(query)	: TRIGger : DETECTDate?
	(response)	year, month, day year = 0 to 99 month = 1 to 12 day = 1 to 31
Explanation	(command)	Sets the date for trigger detection.
	(query)	Returns the currently set date for trigger detection as a numerical value in NR1 format. During memory partitioning, the date for the memory block which is currently being displayed (the block in use) is the one referenced.
Example	: TRIGGER : DETECTDATE? The currently set date for trigger detection is queried.	
When allowed	In all functions.	

TRIGger

❑ Sets and queries the time for start operating termination.

Syntax	(command)	: TRIGger : STOPTime hour, min, sec
	(query)	: TRIGger : STOPTime?
	(response)	hour, min, sec hour = 0 to 23 min = 0 to 59 sec = 0 to 59
Explanation	(command)	Returns the time for start operating termination as a numetical value in NR1 format.
	(query)	Sets the currentry set time start operating termination.
Example	: TRIGGER : STOPTIME? The currently set time for start operating termination is queried.	
When allowed	In the recorder function and the XYC function.	

❑ Sets and queries the date for start operating termination.

Syntax	(command)	: TRIGger : STOPDate year, month, day
	(query)	: TRIGger : STOPDate?
	(response)	year, month, day year = 0 to 99 month = 1 to 12 day = 1 to 31
Explanation	(command)	Returns the date for start operating termination as a numetical value in NR1 format.
	(query)	Sets the currentry set time start operating termination.
Example	: TRIGGER : STOPDATE? The currently set date for start operating termination is queried.	
When allowed	In the recorder function and the XYC function.	

5-3-4. UNIT command

❑ Sets and queries the voltage range of an input channel.

Syntax	(command)	: UNIT : RANGe ch$, A
	(query)	: UNIT : RANGe? ch$
	(response)	ch$, A<NR3> ch$ = CH1 to CH8 A= voltage range (unit V) temperature range (unit °C)
Explanation	(command)	Sets the voltage range for the channel designated by ch$ to a numerical value (unit V). If the channel designated is for the temperature unit, set the temperature range to a numerical value.
	(query)	Returns the current voltage range or the temperature range for the channel designated by ch$ as an NR3 numerical value.
Example	: UNIT : RANGE CH1, +10.E-3 Sets the voltage range for channel 1 to 10 mV.	
When allowed	In all functions.	

❑ Sets and queries input channel origin position.

Syntax	(command)	: UNIT : POSItion ch$, A
	(query)	: UNIT : POSItion? ch$
	(response)	ch$, A<NR1> ch$ = CH1 to CH8 A = position (%)
Explanation	(command)	Sets the origin position for the channel designated by ch$ in the range.
	(query)	Returns the current origin position for the channel designated by ch$ as an NR1 numerical value (unit percent).
Example	: UNIT : POSITION CH1, 50 Sets the origin position for channel 1 to 50%.	
When allowed	In all functions.	

❑ Sets and queries input coupling for an input channel.

Syntax	(command)	: UNIT : COUPling ch$, A$
	(query)	: UNIT : COUPling? ch$
	(response)	ch$, A$ ch$ = CH1 to CH8 A$ = GND, DC GND, DC, RMS, R_G
Explanation	(command)	Sets the input coupling for the channel designated by ch$.
	(query)	Returns the current input coupling for the channel designated by ch$ as character data.
Example	: UNIT : COUPLING CH1, DC Sets the input coupling for channel 1 to DC.	
When allowed	In all functions.	

UNIT

❑ Sets and queries the filter for an input channel.

Syntax
(command) : UNIT : FILTer ch$, A
(query) : UNIT : FILTer? ch$
(response) ch$, A$
ch$ = CH1 to CH8
A = 0, 5, 50, 500, 5000 (8916 analog unit)
0, 5, 500 (8917 RMS unit)
0, 1.5, 5 (8918 temperature unit)
0, 5, 500 (8919 FFT analog unit)
(0 means OFF)

Explanation
(command) Sets the filter for the channel designated by ch$. If the channel designated is for the temperature unit, set the filter of the temperature unit.
(query) Returns the current filter setting for the channel designated by ch$ as character data.

Example
: UNIT : FILTER CH1, 500
Sets the filter for channel 1 to 500 Hz.

When allowed
In all functions.

❑ Carries out zero adjustment for the input units.

Syntax
(command) : UNIT : ADJUST

Explanation
(command) Carries out zero adjustment for the input units, however there is no zero adjustment function for the temperature input unit.

When allowed
In all functions.

❑ Sets and queries the type of the temperature input unit sensor.

Syntax
(command) : UNIT : SENSor ch$, A$
(query) : UNIT : SENSor? ch$
(response) ch$, A$
ch$ = CH1 to CH8
A$ = K, J, T

Explanation
(command) Sets the type of the temperature input unit sensor on the channel designated by ch$.
(query) Returns the type of the temperature input unit sensor currently on the channel designated by ch$ as character data.

Example
: UNIT : SENSOR CH1, K
The temperature input unit sensor on channel 1 is set to "K".

When allowed
In all functions.

❑ Sets and queries the FFT anti-aliasing filter.

Syntax	(command)	: UNIT : AAFilter ch$, A$
	(query)	: UNIT : AAFilter? ch$
	(response)	ch$, A$ ch$ = CH1 to CH8 A$ = OFF, ON
Explanation	(command)	Turns on or off the FFT anti-aliasing filter on the channel designated by A$.
	(query)	Returns the current on or off state of the FFT anti-aliasing filter on the channel designated by A$.
Example	: UNIT : AAFILTER CH3, ON Turns on the FFT anti-aliasing filter for channel 3.	
When allowed	In the memory recorder function and the FFT function.	

5-3-5. DISPlay command

❑ Sets and queries the screen mode.

Syntax	(command)	: DISPlay : CHANge A$
	(query)	: DISPlay : CHANge?
	(response)	A$ A$ = SYSTem STATus CHANnel DISPlay
Explanation	(command)	Changes the screen mode.
	(query)	Returns the current screen mode as character data.
Example	: DISPLAY : CHANGE DISPLAY Switches to the display mode.	
When allowed	In all functions.	

❑ Sets and queries waveform display style.

Syntax	(command)	: DISPlay : DRAWing ch$, A$
	(query)	: DISPlay : DRAWing? ch$
	(response)	ch$, A$ ch$ = CH1 to CH8 A$ = OFF, LIGHt, NORMal, DARK
Explanation	(command)	Sets the waveform display style for the channel designated by ch$ to OFF, LIGHT (high intensity), or DARK (low intensity).
	(query)	Returns the current waveform display style setting for the channel designated by ch$ as character data.
Example	: DISPLAY : DRAWING CH1, DARK Displays the channel 1 waveform the DARK.	
When allowed	In the memory recorder function and the recorder function.	

❑ Sets and queries waveform display graph in DUAL and QUAD format.

Syntax	(command)	: DISPlay : GRAPh ch$, A
	(query)	: DISPlay : GRAPh? ch$
	(response)	ch$, A<NR1> ch$ = CH1 to CH8 A = 1,2,3,4 (for DUAL, 1, 2)
Explanation	(command)	Sets the waveform display graph in the screen.
	(query)	In the screen, returns the current waveform display graph for a channel as an NR1 numerical value.
Example	: DISPLAY : GRAPH CH1, 1 Displays the channel 1 waveform in display graph 1.	
When allowed	In the memory recorder function and the recorder function.	

❑ Enables and disables, and queries, display of logic waveforms.

Syntax	(command)	: DISPlay : LOGDraw ch$, A$
	(query)	: DISPlay : LOGDraw? ch$
	(response)	ch$, A$ ch$ = CHA to CHD A$ = OFF, ON
Explanation	(command)	Enables and disables display of logic waveforms.
	(query)	Returns current enablement state of logic waveform display as character data.
Example	: DISPLAY : LOGDRAW CHA, ON Enables display of the channel A logic waveform.	
When allowed	In the memory recorder function and the recorder function.	

❑ Sets and queries operation of logic waveform display.

Syntax	(command)	: DISPlay : LOGPosi ch$, A
	(query)	: DISPlay : LOGPosi? ch$
	(response)	ch$, A ch$ = CHA to CHD A = 1 to 8
Explanation	(command)	Sets the position of logic waveform display.
	(query)	Returns the position of the current logic waveform display as character data.
Example	: DISPLAY : LOGPOSI CHA, 1 Sets the position of logic waveform display for channel A.	
When allowed	In the memory recorder function and the recorder function.	

DISPlay

❑ Sets and queries zoom factor on the time axis.

Syntax
(command) : DISPlay : XMAG A$
(query) : DISPlay : XMAG?
(response) A$
MEM: A$ = X8, X4, X2, X1, X1_2, X1_5, X1_10, X1_20, X1_50, X1_100, X1_200, X1_500, X1_1000
REC : A$ = X1, X1_2, X1_5, X1_10, X1_20, X1_50

Explanation
(command) Sets the zoom factor on the time axis according to character data.
(query) Returns the current zoom factor on the time axis as character data.

Example
: DISPLAY : XMAG X1_10
Sets the compression ratio along the time axis to be 1/10.

When allowed
In the memory recorder function and the recorder function.

❑ Sets and queries zoom factor on the voltage axis.

Syntax
(command) : DISPlay : YMAG ch$, A$
(query) : DISPlay : YMAG? ch$
(response) ch$, A$
ch$ = CH1 to CH8
A$ = X8, X4, X2, X1, X1_2 .....SINGLE format
X4, X2, X1, X1_2, X1_4 .....DUAL, X-Y dual, X-Ysing format
X2, X1, X1_2, X1_4, X1_8 ..QUAD, format
X1, X1_2, X1_4, X1_8, X1_16..OCT format

Explanation
(command) Sets the zoom factor on the voltage axis for the channel designated by ch$ according to the character data.
(query) Returns the current zoom factor on the voltage axis for the channel designated by ch$ as character data.

Example
: DISPLAY : YMAG X2
Sets the voltage axis to X2.

When allowed
In the memory recorder function and the recorder function.

❑ Sets and queries the drawing level for an X-Y plot.

Syntax
(command) : DISPlay : XYDRawing A, B$
(query) : DISPlay : XYDRawing? A
(response) A<NR1>, B$
A = 1 to 4
B$ = OFF
LIGHt
NORMal
DARK

Explanation
(command) Sets the drawing level for an X-Y plot.
(query) Queries the drawing level for an X-Y plot.

Example
: DISPLAY : XYDRAWING 1, DARK
Sets the graph 1 to DARK display.

When allowed
In the memory recorder function (in XY format) and the XYC recorder function.

❑ Sets and queries the X-axis, in the XY format.

Syntax	(command)	: DISPlay : XAXIs A, ch$
	(query)	: DISPlay : XAXIs? A
	(response)	A<NR1>, ch$ A = 1 to 4 (When XYsing, only 1) ch$ = CH1 to CH8
Explanation	(command)	Sets the Xaxis channel in the XY format.
	(query)	Returns the current Xaxis channel in the XY format.
Example	: DISPLAY : XAXIS 1, CH1 Sets graph 1 to channel 1.	
When allowed	In the memory recorder function (in XY format), and in the XYC recorder function.	

❑ Sets and queries the Y-axis, in the XY format

Syntax	(command)	: DISPlay : YAXIs A, ch$
	(query)	: DISPlay : YAXIs? A
	(response)	A<NR1>, ch$ A = 1 to 4 (When XYsing, only 1) ch$ = CH1 to CH8
Explanation	(command)	Sets the Y axis channel in the XY format.
	(query)	Returns the current Y axis channel in the XY format.
Example	: DISPLAY : YAXIS 1, CH2 Sets graph 1 to the Yaxis channel 2.	
When allowed	In the memory recorder function (in XY format) and the XYC recorder function.	

❑ Performs waveform display.

Syntax	(command)	: DISPlay : WAVE A$ A$= ACUR (the A-cursor) TRIG (the trigger point) POINT (the point set by :MEMory:POINt)
Explanation	Displays the waveform on the screen from the position indicated by A$.	
Example	: DISPLAY : WAVE ACUR Displays the waveform from the position of A cursor.	
When allowed	In the memory recorder function (when A$ = ACUR, the A-cursor must be displayed).	

DISPlay

❑ Sets and queries display clearing in the X-Y recorder function.

Syntax	(command)	: DISPlay : XYCLr A$
	(query)	: DISPlay : XYCLr?
	(response)	A$ A$ = OFF, ON
Explanation	(command)	Enables or disables display clearing in the X-Y recorder function.
	(query)	In the X-Y recorder function, returns the enablement of display clearing.
Example	: DISPLAY : XYCLR ON Sets the display clearing to ON.	
When allowed	In the XYC recorder function.	

❑ Enables and disables, and queries the variable function.

Syntax	(command)	: DISPlay : VARIable ch$, A$
	(query)	: DISPlay : VARIable? ch$
	(response)	ch$, A$ ch$ = CH1 to CH8 A$ = ON/OFF
Explanation	(command)	Enables or disables the variable function.
	(query)	Returns the current state of enablement of the variable function.
When allowed	In all function.	

❑ Sets and queries the upper and lower limit values of the variable function.

Syntax	(command)	: DISPlay : VARIUPLOw ch$, B, C
	(query)	: DISPlay : VARIUPLOw? ch$
	(response)	ch$, B, C ch$ = CH1 to CH8 B = C = -9.9999E+29 to +9.9999E+29
Explanation	(command)	Sets the upper and lower limit values of the waveform on the display screen for CH1 to CH8. B; upper limit value C; lower limit value
	(query)	Returns the current upper and lower limit values of the waveform on the display screen for CH1 to CH8 as an NR3 numerical value.
When allowed	In all functions.	

5-3-6. CURSor command

❑ Turns on and off, and queries, the A and B cursors.

Syntax	(command)	: CURSor : MODE A$
	(query)	: CURSor : MODE?
	(response)	A$
		A$ = OFF, TIME, VOLT, TRACe :MEM
		OFF, Xcur, Ycur, TRACe :MEM(XY format)
		OFF, TIME, VOLT : REC
		OFF, Xcur, Ycur :XYC
		OFF, TRACe : FFT
Explanation	(command)	Sets the A and B cursor type (vertical cursor, horizontal cursor, trace cursor).TIME and Xcur relate to the vertical cursor, VOLT and Ycur relate to the horizontal cursor, and TRACe relates to the trace cursor.
	(query)	Returns the current A and B cursor type as character data.
Example	: CURSOR : MODE TIME Sets vertical cursors.	
When allowed	In all functions.	

❑ Selects between, and queries, A, B only or A and B cursors.

Syntax	(command)	: CURSor : ABCUrsor A$
	(query)	: CURSor : ABCUrsor?
	(response)	A$
		A$ = A, B, A_B
Explanation	(command)	Selects A or B cursor or both (A_B).
	(query)	Returns whether currently the A or B cursor is in use, or both A and B cursors are in use, as character data.
Example	: CURSOR : ABCURSOR A Sets the cursor to A.	
When allowed	In all functions.	

❑ Sets and queries the channel for the A cursor.

Syntax	(command)	: CURSor : ACHAnnel ch$
	(query)	: CURSor : ACHAnnel?
	(response)	ch$
		ch$ = CH1 to CH8
		X1 to X4 (in X-Y recorder)
Explanation	(command)	Sets the channel for the A cursor.
	(query)	Returns the current A cursor channel as character data.
Example	: CURSOR : ACHANNEL CH1 Sets the channel for the A cursor to CH1.	
When allowed	During use of the cross-hair cursor or the horizontal cursor (excluding the FFT function).	

CURSor

❑ Sets and queries the channel for the B cursor.

Syntax	(command)	: CURSor : BCHAnnel ch$
	(query)	: CURSor : BCHAnnel?
	(response)	ch$ ch$ = CH1 to CH8 X1 to X4 (in X-Y recorder)
Explanation	(command)	Sets the channel for the B cursor.
	(query)	Returns the current B cursor channel as character data.
Example	: CURSOR: BCHANNEL CH1 Sets the channel for the B cursor to CH1.	
When allowed	During use of the trace cursor or the horizontal cursor (excluding the FFT function).	

❑ Sets and queries the position of the A cursor.

Syntax	(command)	: CURSor : APOSition A
	(query)	: CURSor : APOSition?
	(response)	A<NR1> [vertical cursor, trace cursor] A = 0 to number of stored data values (100 × recording length) : MEM A = 0 to number of stored data values : REC A = 0 to 319 :XYC analysis mode A = 0 to 999 (STR, ACR, CCR, IMP) : FFT A = 0 to 400 (LIN, RMS, PSP, TRF, COH, HIS, CSP, OCT) : FFT [horizontal cursor] A = 0 to 639 : MEM, REC 0 to 319 : XYC
Explanation	(command)	Sets the A cursor position (refer to next page).
	(query)	Returns the current A cursor position as an NR1 numerical value.
Example	: CURSOR : APOSITION 1000 Moves the position of the A cursor to 1000 point (10DIV) .	
When allowed	In all functions.	

❑ Sets and queries the position of the B cursor.

Syntax	(command)	: CURSor : BPOSition A
	(query)	: CURSor : BPOSition?
	(response)	A<NR1>
Explanation	(command)	Sets the B cursor position (refer to next page).
	(query)	Returns the current B cursor position as an NR1 numerical value.
Example	: CURSOR : BPOSITION 1000 Moves the position of the B cursor to 1000 point (10DIV) .	
When allowed	In all functions.	

The cursor position has the following meaning:

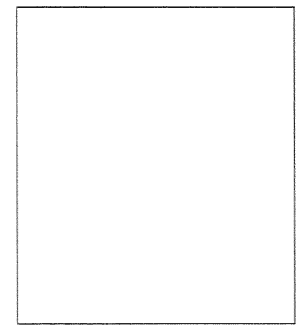
In the memory recorder function and the recorder function, when the vertical cursor or the trace cursor is in use, the cursor position is an indication of the current point in memory.

(When storage is performed on a basis of 25 divisions, the number of stored data values is 2500, and the cursor position indication lies in the range from 0 to 2500)

The horizontal cursor, and in the memory recorder function in XY format and in the XYC recorder function both the vertical cursor and the horizontal cursor, are shown to the lower left of the waveform area on the display in a standard position. (refer to next page).

In the memory recorder function and in the recorder function

[vertical cursor
trace cursor]



639 ← Upper end of the vertical axis

0 ← Lower end of the vertical axis

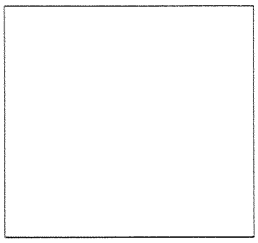
0

Left end of the horizontal axis

Recording length × 100

In the memory recorder function and in the XYC recorder function (in X-Y format)

[vertical cursor
horizontal cursor]



319 ← Upper end of the vertical axis

0 ← Lower end of the vertical axis

0

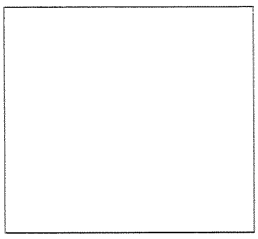
319

Left end of the horizontal axis

Right end of the horizontal axis

In the FFT function

[trace cursor]



0

999 (STR, ACR, CCR, IMP)
400 (LIN, RMS, PSP TRF, COH, HIS, CSP, OCT)

Left end of the horizontal axis

Right end of the horizontal axis

CURSor

❑ Queries the cursor readout value (t).

Syntax
(query) : CURSor : DTREad? A$
(response) "B unit"
A$ = A, B, A_B
B = the readout value (t, 1/t)

Explanation
(query) Returns the cursor readout value (t, 1/t) as a line of character data.

Example
(query) : CURSOR : DTREAD? A
(response) : CURSOR : DTREAD"5ms, 200Hz"
Queries the A cursor readout value.

When allowed
Provided that the cursor is not off, and that (t, 1/t) are being shown on the display.

❑ Sets and queries the cursor readout value (V).

Syntax
(query) : CURSor : DVREad? A$
(response) "B unit"
A$ = A, B, A_B
B$ = the readout value (V)

Explanation
(query) Returns the cursor readout value (V) as a line of character data.
For the temperature input unit, returns the temperature value (°C).

Example
(query) : CURSOR : DVREAD? A
(response) : CURSOR : DVREAD"385mV"
Queries the A cursor readout value.

When allowed
Provided that the cursor is not off, and that (V) is being shown on the display.

❑ Sets and queries the graph for the A and B cursors.

Syntax
(command) : CURSor : ABCHAnnel A$
(query) : CURSor : ABCHAnnel?
(response) A$
A$ = G1, G2

Explanation
(command) Sets the graph for the A and B cursors when the display format is DUAL. If the display format is SINGle or NYQuist, the cursor is displayed on graph 1, whatever setting is made with this command.

(query) Returns the current graph setting for the A and B cursors as character data.

Example
: CONFIGURE : FORMAT DUAL
: CURSOR : ABCHANNEL G1
: CURSOR : MODE TRACE
The A and B cursors are displayed on graph 1.

When allowed
In the FFT function.

❑ Queries the cursor readout position for FFT data.

Syntax	(query only)	: CURSor : DFREad? A$
	(response)	"Bunit , Cunit" A$ = A, B, A_B B = x-axis data C = y-axis data
Explanation	(query only)	Returns the current cursor readout position for FFT data as character data.
Example	: CURSOR : DFREAD? A The A cursor readout position is returned as character data.	
When allowed	In the FFT function.	

5-3-7. MEMory command

❑ Sets and queries the point in memory for input/output.

Syntax	(command)	: MEMory : POINt ch$, A
	(query)	: MEMory : POINt?
	(response)	ch$, A<NR1> ch$ = CH1 to CH8, CHA to CHD A = 0 to 1000000
Explanation	(command)	Sets the input/output point in memory.
	(query)	Returns the current input/output point in memory as an NR1 numerical value.
Example	: MEMORY : POINT CH1, 100 Sets the input/output point for channel 1 to the 100th location from the start of memory.	
When allowed	In the memory recorder function.	

❑ Queries the number of data samples stored.

Syntax	(query)	: MEMory : MAXPoint?
	(response)	: A <NR 1> A = 0 : no data stored 2500 to 1000000 (divided by 100 gives the number of divisions)
Explanation	Returns the number of data samples stored in the memory.	
Example	(query) : MEMORY : MAXPOINT? (response) : MEMORY : MAXPOINT 2500 (for header on) The number of data samples stored in the memory is 2500 (25 divisions).	
When allowed	In the memory recorder function.	

❑ Prepares the memory.

Syntax	(command only) : MEMory : PREPare
Explanation	(command only) If there is no waveform data in the 8840 unit, ensures that the memory is in a state ready and able to receive transmitted data. If waveform data is currently stored in the unit, clears it.
Example	: MEMORY : PREPARE Prepares the memory for receipt of waveform data.
When allowed	In the memory recorder function.

❑ Inputs data to memory, and outputs stored data.(in ASCII)

Syntax

(command) : MEMory : ADATa B, C,.....
(query) : MEMory : ADATa? A
(response) B, C, ...all<NR1>
B, C,..... = 0 to 4095 (data for storage)
A = 1 to 40 (number of data values to be output)

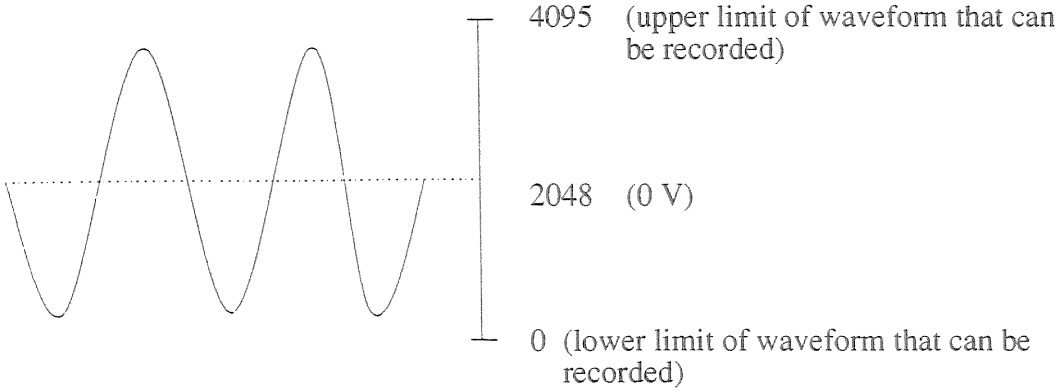
Explanation

(command) Puts the data of the data portion into the memory at the channel and point set by the MEMory:POINt command. If there are several data values, they are input in order from the point set by the MEMory:POINt command. The input/output point is incremented by the number of data values.

(query) The number of data values specified by A are output from the memory channel and point set by the MEMory:POINt command. The input/output point is incremented by the number of data values.
This cannot be executed during measurement operation.

✻ Relationship between data values in memory and measured voltages

The following figure illustrates the relationship between the data values (0 to 4095) input and output using the :MEMory:ADATa command and the measured voltage values. (2048 = 0V)



Measured voltage value = (data value - 2048) × (voltage range) / 80
Example: Voltage range = 1 (V/DIV), data value = 2500
The measured voltage = (2500 - 2048) × 1 / 80 = 5.65 (V)

Reference

The transfer rate by the :MEMORY :ADATA? 40 command is:
When recording length is 10,000DIV, 1051 seconds.
When recording length is 1,000DIV, 108 seconds.
(PC 9801 RA in use. receiving LINE INPUT @)

Example

: MEMORY : POINT CH1, 0
: MEMORY : ADATA? 10
Sets the input/output point to channel 1 and data value zero in memory, then outputs 10 stored data values.

When allowed

Provided that stored data is present, and provided that the input/output point is lower than the amount of data stored.

MEMory

❑ Input voltage data to memory, and output voltage data from memory.

Syntax

(command) : MEMory : VDATa B, C .....
(query) : MEMory : VDATa? A
(response) B, C,.....all<NR3>
B, C,..... = voltage values (unit volts) or temperature values (unit °C)
A = 1 to 10 (amount of data)

Explanation

(command) Puts the data values (voltage values) in the data portion into the memory at the channel and point set by the MEMory:POINt command. If there are several data values, they are input in order from the point set by the MEMory:POINt command.
The input/output point is incremented by the number of data values.

(query) The number of stored data values specified by A are output as voltage values from the memory channel and point set by the MEMory:POINt command.
The input/output point is incremented by the number of data values.
*When scaling, the scaled values are input and output.
When calculating the waveform, calculated results are input and output.
This cannot be executed during measurement operation.

Reference

The transfer rate by : MEMORY : VDATA? 10 command is:
When recording length is 10,000 DIV, 2418 seconds.
When recording length is 1,000 DIV, 242 seconds.
(PC 9801RA in use. recording LINE INPUT @)

Example

```
: MEMORY : POINT CH1, 0
: MEMORY : VDATa? 10
```

Sets the input/output point to channel 1 and data value zero in memory, then outputs 10 stored data values as voltage values.

When allowed

Provided that stored data is present, and provided that the input/output point is lower than the amount of data stored.

❑ Outputs real time data (in ASCII).

Syntax	(query) (response)	: MEMory : AREAl? ch$ A <NR 1> ch$ = CH1 to CH8 A = 0 to 4095
Example	(query) (response)	: MEMORY : AREAL? CH1 : MEMORY : AREAL 3022 (for header on)
Explanation	(query)	Returns the value input on the channel designated by ch$.
When allowed	Providing that measurement operation is not taking place.	

❑ Outputs real time data (voltage values).

Syntax	(query) (response)	: MEMory : VREAl? ch$ A <NR 3> ch$ = CH1 to CH8 A= a voltage value (unit V) or temperature value (unit °C)
Example	(query) (response)	: MEMORY : VREAL CH1 : MEMORY : VREAL 5.5E-2 (for header on)
Explanation	(query)	Returns as a voltage value the value input on the channel designated by ch$.
When allowed	Providing that measurement operation is not taking place.	

MEMory

❑ Input logic data to memory, and output logic data from memory.

Syntax
(command) : MEMory : LDATa B,C.....
(query) : MEMory : LDATa? A
(response) B, C ....all<NR1>
B, C = 0 to 15 (logic data)
A = 1 to 50 (number of data values to be output)

Explanation
(command) Inputs the data values (logic values) in the data portion into the memory at the channel and point set by the : MEMory : POINt command. If there are several data values, they are input in order from the point set by the : MEMory : POINt command.
The input/output point is incremented by the number of data values.

(query) The number of stored data values specified by A are output as logic values from the memory channel and point set by the : MEMory : POINt command.
The input/output point is incremented by the number of data values.
This cannot be executed during measurement operation.

Note: The following is the correspondence between the channels set by the MEMory : POINt command and the logic channel groups:

CH1 ..... CHA 1 to CHA 4
CH2 ..... CHB 1 to CHB 4
CH3 ..... CHC 1 to CHC 4
CH4 ..... CHD 1 to CHD 4

The four logic channels in each group are encoded as binary bits in the NR1 data value, as shown in the following example.

7	6	5	4	3	2	1	0
0	0	0	0	A4	A3	A2	A1

LOW: 0
HIGH: 1

Example

```
: MEMORY : POINT CH1, 0
: MEMORY : LDATA? 1.....If the response is
: MEMORY : LDATA 10
```

then channels A1 to A4 are as follows;

7	6	5	4	3	2	1	0
0	0	0	0	1	0	1	0
				A4	A3	A2	A1

LOW: 0
HIGH: 1

When allowed In the memory function, provided that stored data is present, and provided that the input/output point is lower than the amount of data stored.

❑ Binary transfer of stored data.

Syntax

(query) : MEMory : BDAta? A

(response) # 0 ∗∗∗∗∗∗∗∗• • • •

A = 1 to 125

Explanation

Outputs the data stored by a :MEMory:POINt specification in binary format. The I/O point is also advanced by the number of data values. The format of the output data is as follows:

- Initially: '#0' (Indicates binary format.)
- After the '#0', the number of data values specified by A (each value is two bytes), is transmitted.
- The data is followed by LF (0AH) + EOI.

#0 ∗ ∗ ∗ ∗ ∗• • • • LF (EOI)

1 value

Number of values = A (A ∗ 2 bytes)

- The data consists if the unaltered binary codes of the data stored in memory. The bits are transmitted most significant bit first, and the first four bits are the data for the logic channels.

(Example)

Upper byte		Lower byte	
1100	0100	0010	1100
Logic channels		Analog channel data	

- The data obtained is the same as that for ADATa? and LDATa?; for details refer to these commands.

It is not possible to input data in binary format.

Reference

The transfer time in response to the command : MEMORY : BDATA? 125 is as follows:

When recording length is 10.000 DIV, 329 seconds.
When recording length is 1000 DIV, 33 seconds.
(PC 9801RA in use. recording LINE INPUT @)

Examples

```
: MEMORY : POINT CH1,0
: MEMORY : BDATA? 10
```

This sets the I/O point to channel 1, and stored data value to address 0 in memory, then outputs 10 data values in binary format.

When allowed

The unit must be in the memory recorder function mode, and there must be data stored in memory.
The I/O point must be before the end of the stored data.

MEMory

❑ Queries the FFT data at the output point.

Syntax (query only) : MEMory : FFTData?
(response) "A unit, B unit"
A = x-axis data (in <NR3> numerical format)
B = y-axis data (in <NR3> numerical format)

Explanation (query only) Returns the x-axis and y-axis FFT data at the output point specified by the instruction :MEMORY:FFTPOINT in <NR3> numerical format. When this command is executed, only one output point is calculated, and then the specified output point is increased by one. By executing this command repeatedly, a continuous set of data can be obtained.

Example
```
: MEMORY : FFTPOINT G1, 100
: MEMORY : FFTDATA?
: MEMORY : FFTDATA?
```
Returns the x-axis and y-axis FFT data at the points 100 and 101 on graph 1.

When allowed In the FFT function.

❑ Sets and queries the output point for FFT data.

Syntax (command) : MEMory : FFTPOint A$, B
(query) : MEMory : FFTPOint?
(response) A$, B<NR1>
A$ = G1, G2
B = 0 to 999... in analysis mode STR, ACR, CCR, or IMP
B = 0 to 400 ... in analysis mode LIN, RMS, PSP, TRF, COH, CSP, HIS, or OCT

Explanation (command) Sets the output point for FFT data on the graph number designated by A$.

(query) Returns the current output point for FFT data on the graph number designated by A$ as a numerical value in 1 format.

Example
```
: MEMORY : FFTPOINT G1, 100
```
Sets the output point for FFT data on the graph 1 to 100.

When allowed In the FFT function.

5-3-8. SYSTem command

❑ Sets the time, and queries the current time.

Syntax	(command)	: SYSTem : TIME hour, min, sec
	(query)	: SYSTem : TIME?
	(response)	hour, min, sec hour = 0 to 23 min = 0 to 59 sec = 0 to 59
Explanation	(command)	Sets the time.
	(query)	Returns the current time.
Example	: SYSTEM : TIME 10, 0, 0 Sets the internal clock to 10:00.	
When allowed	In all functions.	

❑ Sets the calendar date, and queries the current calendar date.

Syntax	(command)	: SYSTem : DATE year, month, day
	(query)	: SYSTem : DATE?
	(response)	year, month, day year = 0 to 99 month = 1 to 12 day = 1 to 31
Explanation	(command)	Sets the date on the internal calendar.
	(query)	Returns the current date.
Example	: SYSTEM : DATE 93, 6, 24 Sets the internal calendar to June 24th, 1993.	
When allowed	In all functions.	

❑ Cleaning waveform data

Syntax	(command)	: SYSTem : DATAClear
Explanation	Clear the waveform data	
When allowed	In all functions.	

❑ Sets and queries the number of channels used.

Syntax	(command)	: SYSTem : USECH A
	(query)	: SYSTem : USECH?
	(response)	A<NR1> A = 1, 2, 4, 8
Explanation	(command)	Sets the number of channels used to a numerical value.
	(query)	Returns the current number of channels used as an NR1 numerical value.
Example	: SYSTEM : USECH 8 Sets the number of channels to 8.	
When allowed	In the memory recorder function and the recorder function.	

SYSTem

❑ Enables and disables, and queries, the start backup function.

Syntax
(command) : SYSTem : STARt A$
(query) : SYSTem : STARt?
(response) A$
A$ = OFF, ON

Explanation
(command) Enables and disables the start backup function.
(query) Returns the current enablement state of the start backup function as character data.

Example
: SYSTEM : START ON
Sets the start backup function to ON.

When allowed
In all functions.

❑ Sets and queries the grid type.

Syntax
(command) : SYSTem : GRID A$
(query) : SYSTem : GRID?
(response) A$
A$ = OFF, STANdard, FINE

Explanation
(command) Sets the type of grid displayed.
(query) Returns the current grid setting as character data.

Example
: SYSTEM : GRID STANDARD
Sets the grid to STANDARD.

When allowed
In all functions.

❑ Enables and disables, and queries, the channel marker.

Syntax
(command) : SYSTem : CHMArk A$
(query) : SYSTem : CHMArk?
(response) A$
A$ = OFF, ON

Explanation
(command) Makes the corresponding channel marker setting.
(query) Returns the current channel marker setting as character data.

Example
: SYSTEM : CHMARK ON
Sets the channel marker to ON.

When allowed
In all functions.

❑ Sets and queries the time axis display.

Syntax
(command) : SYSTem : TMAXis A$
(query) : SYSTem : TMAXis?
(response) A$
A$ = TIME
TIME (60)
DIV

Explanation
(command) Sets the time axis display as character string data.
(query) Returns the current time axis display setting as character string data.

Example
: SYSTEM : TMAXIS TIME
Sets the time axis display to TIME.

When allowed In all functions.

❑ Sets and queries the list function and the gauge function.

Syntax
(command) : SYSTem : LIST A$
(query) : SYSTem : LIST?
(response) A$
A$ = OFF
LIST
GAUGE
L_G (LIST&GAUGE)

Explanation
(command) Sets the list function and the gauge function according to a character string.
(query) Returns the current settings for the list function and the gauge function as a character string.

Example
: SYSTEM : LIST LIST
Sets the list function.

When allowed In all functions.

❑ Enables and disables, and queries, the screen back light saver function.

Syntax
(command) : SYSTem : CRTOff A$
(query) : SYSTem : CRTOff?
(response) A$
A$ = OFF, ON

Explanation
(command) Enables or disables the screen saver function.
(query) Returns the current enablement state of the screen saver function as character data.

Example
: SYSTEM : CRTOFF ON
Sets the back light saver function to ON.

When allowed In all functions.

SYSTem

❑ Sets and queries the LCD display.

Syntax	(command)	: SYSTem : LCDDisp A$
	(query)	: SYSTem : LCDDisp?
	(response)	A$ A$ = NORMal, REVErse
Explanation	(command)	Sets the LCD display.
	(query)	Returns the current LCD display setting as character data.
Example	: SYSTEM : LCDDISP NORMAL Sets the LCD display to NORMAL.	
When allowed	In all functions.	

❑ Enables and disables, and queries, the sound of the beeper.

Syntax	(command)	: SYSTem : BEEPer A$
	(query)	: SYSTem : BEEPer?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables and disables the beeper sound.
	(query)	Returns the current enablement state of the beeper sound as character data.
Example	: SYSTEM : BEEPER ON Sets the beeper sound to ON.	
When allowed	In all functions.	

❑ Enables and disables, and queries intermittent printing function.

Syntax	(command)	: SYSTem : THINout A$
	(query)	: SYSTem : THINout?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables and disables intermittent printing function.
	(query)	Returns the current enablement state of intermittent printing as character data.
Example	: SYSTEM : THINOUT ON Sets the intermittent function to ON.	
When allowed	In all functions.	

❑ Sets and queries the destination of hard copy output.

Syntax
(command) : SYSTem : COPY A$
(query) : SYSTem : COPY?
(response) A$
A$ = PRINter, PLOTter
FD_Mono, FD_Color

Explanation
(command) Sets the output destination of hard copy.
(query) Returns the output destination of hard copy as character data.

Example
```
: SYSTEM : COPY PRINTER
: HCOPY
```
Prints out the hard copy of the screen display on the printer.

When allowed In all functions.

❑ Sets the colors of the bit map file.

Syntax
(command) : SYSTem : BMPColor A$, B$, C$, D$
(query) : SYSTem : BMPColor?
(response) A$, B$, C$, D$
A$, B$, C$, D$ = BLACK, BLUE, RED, MAGENTA, GREEN, CYAN, YELLOW, ORANGE

Explanation
(command) Sets the color of screen displays for using the bit map file. The character and frame can be designated by A$, and each waveforms; light, normal, dark, can be designated by B$, C$, and D$.
(query) Returns the setting screen colors for using bit map file as character data.

Example
```
: SYSTEM : COPY FD_COLOR
: SYSTem : BMPCOLOR BLACK, RED, YELLOW, ORANGE
: HCOPY
```
Outputs the hard copy with colored data on the floppy disk.

When allowed In all functions.

5-3-9. SCALing command

❑ Enables and disables, and queries, the scaling function.

Syntax	(command)	: SCALing : KIND A$
	(query)	: SCALing : KIND?
	(response)	A$ A$ = RATIO, POINT
Explanation	(command)	Sets the scaling type as character string data.
	(query)	Returns the current scaling type setting as character string data.
When allowed	In all functions.	

❑ Enables and disables, and queries, the scaling function.

Syntax	(command)	: SCALing : SET ch$, A$
	(query)	: SCALing : SET? ch$
	(response)	ch$, A$ ch$ = CH1 to CH8 A$ = OFF, SCI (or ON), and ENG
Explanation	(command)	Enables or disables the scaling function. A setting SCI or ON produces conventional scientific floating-point notation. The setting ENG produces floating-point notation using powers of 1000.
	(query)	Returns the current state of enablement of the scaling function as character data.
Example	: SCALING : SET CH1 SCI Sets the scaling to SCI.	
When allowed	In all functions.	

❑ Sets and queries the scaling conversion value.

Syntax	(command)	: SCALing : VOLT A$, B
	(query)	: SCALing : VOLT? A$
	(response)	B<NR3> A$ = CH1 to CH8 B = scaling conversion value (eu/volts) (-9.9999E+9 to +9.9999E+9)
Explanation	(command)	Sets the scaling conversion value for CH1 to CH8.
	(query)	Returns the current scaling conversion value for CH1 to CH8 as an NR3 numerical value.
Example	: SCALING : VOLT CH1, +2.0E-3 Sets the scaling conversion value for CH1 to +2.0E-3.	
When allowed	In all functions.	

❑ Sets and queries the scaling offset.

Syntax
(command) : SCALing : OFFSet A$, B
(query) : SCALing : OFFSet? A$
(response) A$, B<NR3>
A$ = CH1 to CH8
B= scaling offset (eu offset)
(-9.9999E+9 to +9.9999E+9)

Explanation
(command) Sets the scaling offset for CH1 to CH8
(query) Returns the current scaling offset for CH1 to CH8 as an NR3 numerical value.

Example : SCALING : OFFSET CH1, +1.0E-3
Sets the scaling offset for CH1 to +1.0E-3.

When allowed In all functions.

❑ Sets and queries the scaling unit.

Syntax
(command) : SCALing : UNIT A$, "B$"
(query) : SCALing : UNIT? A$
(response) A$, "B$"
A$ = CH1 to CH8
B$ = scaling unit (7 characters)

Explanation
(command) Sets the scaling unit for CH1 to CH8 (up to 7 characters allowed).

A to Z	a to z	+	−	*	/	%	•	=	space
^2 (= ^{2})		^3 (= ^{3})		~u (= μ)				~o (= Ω)	

Characters other than the above are replaced by spaces.
(query) Returns the current scaling unit for CH1 to CH8 as character data.
Replacing double quotation marks (") in the commands, single quotation marks (') can be also used.

Example : SCALING : UNIT CH1, "mA"
Sets the scaling unit for CH1 to milliamps.

When allowed In all functions.

❑ Sets and queries the scaling VOLT UP and LOW.

Syntax
(command) : SCALing : VOUPLOw ch$, B, C
(query) : SCALing : VOUPLOw? ch$
(response) ch$, B, C
ch$ = CH1 to CH8
B = C = -9.9999E+29 to +9.9999E+29

Explanation
(command) Sets the scaling VOLT UP and VOLT LOW values for CH1 to CH8.
(query) Returns the current scaling VOLT UP and VOLT LOW values for CH1 to CH8 as an NR3 numerical value.

When allowed In all functions.

SCALing

❑ Sets and queries the scaling SCALE UP and LOW.

Syntax	(command)	: SCALing : SCUPLOw ch$, B, C
	(query)	: SCALing : SCUPLOw? ch$
	(response)	ch$, B, C ch$ = CH1 to CH8 B = C = -9.9999E+29 to +9.9999E+29
Explanation	(command)	Sets the scaling SC UP and SC LOW values for CH1 to CH8.
	(query)	Returns the current scaling SC UP and SC LOW values for CH1 to CH8 as an NR3 numerical value.
When allowed	In all functions.	

5-3-10. COMMent command

❑ Enables and disables, and queries, title comments, and inputs comment characters.

Syntax
(command) : COMMent : TITLe A$, "B$"
(query) : COMMent : TITLe?
(response) A$, "B$"
A$ = OFF
SETTing
COMMent
S_C (SETTING&COMMENT)
B$ = comment characters (up to 20 characters)

Explanation
(command) Enables and disables comments, and inputs a string of comment characters. Characters that can be used are:

A to Z		a to z		0 to 9	+	−	*	/	%	=	(	)	
#	&	.	^	,	~u (= μ)			~c (=°)					space

Characters other than the above are replaced by spaces.
Comments may be omitted.
(query) Returns the current enablement state of title comments, and the characters of the comment if any, as character data.
Replacing double quotation marks (") in the commands, single quotation marks (') can be also used.

Example
: COMMENT : TITLE COMMENT, "HIOKI 8840"
Inputs "HIOKI 8840" as a title comment.

When allowed In all functions.

❑ Enables and disables, and queries, comments for all channels.

Syntax
(command) : COMMent : EACHch A$
(query) : COMMent : EACHch?
(response) A$
A$ = OFF
SETTing
COMMent
S_C (SETTING&COMMENT)

Explanation
(command) Enables or disables comments for all channels.
(query) Returns the current ON/OFF enablement state of comments for all channels as character data.

Example
: COMMENT: EACHCH COMMENT
Sets comments for all channels to COMMENT.

When allowed In all functions.

COMMent

❑ For each channel, sets and queries the inputs comment characters.

Syntax

(command) : COMMent : CH ch$, "A$,"
(query) : COMMent : CH? ch$
(response) ch$, "A$"
ch$ = CH1 to CH8, CHA to CHD
A$ = comment characters (up to 20 characters)

Explanation

(command) Enables and disables comment display for the channel specified by ch$, and inputs a string of comment characters. Characters that can be used are:

A to Z		a to z		0 to 9		+	−	*	/	%	=	(	)
#	&	.	^	,	~u (= μ)				~c (= °)			space	

Characters other than the above are replaced by spaces. Comments may be omitted.

(query) Returns the current enablement state of comment display for the channel specified by ch$, and the characters of the comment if any, as character data.
Replacing double quotation marks (") in the commands, single quotation marks (') can be also used.

Example

: COMMENT : CH CH1 , "ch1 = TEST"
Sets the comment display for channel 1 to "ch1 = TEST".

When allowed

In all functions.

5-3-11. CALCulate command

❑ Enables and disables, and queries, waveform processing calculation.

Syntax	(command)	: CALCulate : WVCALc A$
	(query)	: CALCulate : WVCALc?
	(response)	A$
		A$ = OFF, ON, EXEC (execute)
Explanation	(command)	Enables or disables, according to character data, the execution of waveform processing calculation.
	(query)	Returns, as character data, whether execution of waveform processing calculation is enabled or disabled. Only valid when execution (EXEC) is enabled.
Example	: CALCULATE : WVCALC ON Sets the waveform processing calculation to ON.	
When allowed	In the memory recorder function.	

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z1.

Syntax	(command)	: CALCulate : Z1 A$, B$, C$, D$
	(query)	: CALCulate : Z1?
	(response)	A$, B$, C$, D$
		A$, B$, C$ = A to P
		D$ = PLUS : +
		MINUS : -
		MULT: *
		DIVI: /
	[Syntax of :Z2 to :Z8 commands same as the :Z1 command]	
Explanation	(command)	Sets the coefficients for the waveform processing calculation equation for Z1 according to the character data.
	(query)	Returns the current coefficients for the waveform processing calculation equation for Z1 as character data. A$, B$, C$, D$ are used to set up the calculation equation for Z1 in the following way: Z1 = A$ X1 D$ B$ Y1 + C$
Example	: CALCULATE : Z1 A, B, C, PLUS Sets up the calculation equation for Z1 to be Z1 = aX1+bY1+c	
When allowed	In the memory recorder function.	

CALCulate

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z2.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax	(command)	: CALCulate : Z2 A$, B$, C$, D$
	(query)	: CALCulate : Z2?
	(response)	A$, B$, C$, D$
Explanation	(command)	Sets the coefficients for the waveform processing calculation equation for Z2 according to the character data.
	(query)	Returns the current coefficients for the waveform processing calculation equation for Z2 as character data.
When allowed	In the memory recorder function.	

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z3.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax	(command)	: CALCulate : Z3 A$, B$, C$, D$
	(query)	: CALCulate : Z3?
	(response)	A$, B$, C$, D$
Explanation	(command)	:Sets the coefficients for the waveform processing calculation equation for Z3 according to the character data.
	(query)	:Returns the current coefficients for the waveform processing calculation equation for Z3 as character data.
When allowed	In the memory recorder function.	

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z4.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax	(command)	: CALCulate : Z4 A$, B$, C$, D$
	(query)	: CALCulate : Z4?
	(response)	A$, B$, C$, D$
Explanation	(command)	Sets the coefficients for the waveform processing calculation equation for Z4 according to the character data.
	(query)	Returns the current coefficients for the waveform processing calculation equation for Z4 as character data.
When allowed	In the memory recorder function.	

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z5.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax	(command)	: CALCulate : Z5 A$, B$, C$, D$
	(query)	: CALCulate : Z5?
	(response)	A$, B$, C$, D$
Explanation	(command)	Sets the coefficients for the waveform processing calculation equation for Z5 according to the character data.
	(query)	Returns the current coefficients for the waveform processing calculation equation for Z5 as character data.
When allowed	In the memory recorder function.	

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z6.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax	(command)	: CALCulate : Z6 A$, B$, C$, D$
	(query)	: CALCulate : Z6?
	(response)	A$, B$, C$, D$
Explanation	(command)	:Sets the coefficients for the waveform processing calculation equation for Z6 according to the character data.
	(query)	:Returns the current coefficients for the waveform processing calculation equation for Z6 as character data.
When allowed	In the memory recorder function.	

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z7.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax	(command)	: CALCulate : Z7 A$, B$, C$, D$
	(query)	: CALCulate : Z7?
	(response)	A$, B$, C$, D$
Explanation	(command)	Sets the coefficients for the waveform processing calculation equation for Z7 according to the character data.
	(query)	Returns the current coefficients for the waveform processing calculation equation for Z7 as character data.
When allowed	In the memory recorder function.	

CALCulate

❑ Sets and queries the coefficients for the waveform processing calculation equation for Z8.

(For details, refer to the explanation for the :Z1 command on page 5-63.)

Syntax
(command) : CALCulate : Z8 A$, B$, C$, D$
(query) : CALCulate : Z8?
(response) A$, B$, C$, D$

Explanation
(command) Sets the coefficients for the waveform processing calculation equation for Z8 according to the character data.
(query) Returns the current coefficients for the waveform processing calculation equation for Z8 as character data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for X1.

Syntax
(command) : CALCulate : X1 A$, ch$, B$
(query) : CALCulate : X1?
(response) A$, ch$, B$

A$ = OFF(in this case, ch$ and B$ are disregarded)
PAR :(
ABS :Absolute value
EXP :Exponential
LOG :Common logarithm
SQR :Square root
MOV :Moving average
DIF :Differentiation once
INT :Intergration once
DIF2 :Differentiation twice
INT2 :Intergration twice
SLI :Parallel displacement
ch$ = CH1 to CH8
B$ = A to P;
when A$ is set to MOV, a numerical value from 1 to 4000;
when A$ is set to SLI, a numerical value from -4000 to 4000.

[Syntax of the :X2 to :X8 commands are same as :X1 command except "ch$"]

Explanation
(command) Sets the X1 calculation equation for the waveform processing calculation equation for Z1 according to the character or numerical data.
(query) Returns the current X1 calculation equation for the waveform processing calculation equation for Z1 as character or numerical data. A$, B$, and ch$ are used to set up the calculation equation in the following way:
X1 = A$(ch$+B$)
when A$ is set to MOV or SLI:
X1 = [MOV or SLI] (ch$,B$)
(Refer to Section 12 "Calculation Functions" of 8840 manual.)

Example
(1) : CALCULATE : X1 ABS, CH1, A
Sets up the calculation equation for X1 to be X1 = ABS(ch1+a)
(2) : CALCULATE : X1 MOV, CH1, 50
Sets up the calculation equation for X1 to be X1 = MOV(ch1,50)

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for X2.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax	(command)	: CALCulate : X2 A$, ch$, B$
	(query)	: CALCulate : X2?
	(response)	A$, ch$, B$ ch$ = CH1 to CH8, Z1
Explanation	(command)	Sets the X2 calculation equation for the waveform processing calculation equation for Z2 according to the character or numerical data.
	(query)	Returns the current X2 calculation equation for the waveform processing calculation equation for Z2 as character or numerical data.
When allowed	In the memory recorder function.	

❑ Sets up and queries the calculation equation for X3.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax	(command)	: CALCulate : X3 A$, ch$, B$
	(query)	: CALCulate : X3?
	(response)	A$, ch$, B$ ch$ = CH1 to CH8, Z1 to Z2
Explanation	(command)	Sets the X3 calculation equation for the waveform processing calculation equation for Z3 according to the character or numerical data.
	(query)	Returns the current X3 calculation equation for the waveform processing calculation equation for Z3 as character or numerical data.
When allowed	In the memory recorder function.	

❑ Sets up and queries the calculation equation for X4.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax	(command)	: CALCulate: X4 A$, ch$, B$
	(query)	: CALCulate: X4?
	(response)	A$, ch$, B$ ch$ = CH1 to CH8, Z1 to Z3
Explanation	(command)	Sets the X4 calculation equation for the waveform processing calculation equation for Z4 according to the character or numerical data.
	(query)	Returns the current X4 calculation equation for the waveform processing calculation equation for Z4 as character or numerical data.
When allowed	In the memory recorder function.	

CALCulate

❑ Sets up and queries the calculation equation for X5.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax
(command) : CALCulate : X5 A$, ch$, B$
(query) : CALCulate : X5?
(response) A$, ch$, B$
ch$ = CH1 to CH8, Z1 to Z4

Explanation
(command) Sets the X5 calculation equation for the waveform processing calculation equation for Z5 according to the character or numerical data.
(query) Returns the current X5 calculation equation for the waveform processing calculation equation for Z5 as character or numerical data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for X6.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax
(command) : CALCulate : X6 A$, ch$, B$
(query) : CALCulate : X6?
(response) A$, ch$, B$
ch$ = CH1 to CH8, Z1 to Z5

Explanation
(command) Sets the X6 calculation equation for the waveform processing calculation equation for Z6 according to the character or numerical data.
(query) Returns the current X6 calculation equation for the waveform processing calculation equation for Z6 as character or numerical data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for X7.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax
(command) : CALCulate : X7 A$, ch$, B$
(query) : CALCulate : X7?
(response) A$, ch$, B$
ch$ = CH1 to CH8, Z1 to Z6

Explanation
(command) Sets the X7 calculation equation for the waveform processing calculation equation for Z7 according to the character or numerical data.
(query) Returns the current X7 calculation equation for the waveform processing calculation equation for Z7 as character or numerical data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for X8.

(For details, refer to the explanation for the :X1 command on page 5-66.)

Syntax	(command)	: CALCulate : X8 A$, ch$, B$
	(query)	: CALCulate : X8?
	(response)	A$, ch$, B$ ch$ = CH1 to CH8, Z1 to Z7
Explanation	(command)	Sets the X8 calculation equation for the waveform processing calculation equation for Z8 according to the character or numerical data.
	(query)	Returns the current X8 calculation equation for the waveform processing calculation equation for Z8 as character or numerical data.
When allowed	In the memory recorder function.	

CALCulate

❑ Sets up and queries the calculation equation for Y1.

Syntax

(command) : CALCulate : Y1 A$, ch$, B$
(query) : CALCulate : Y1?
(response) A$, ch$, B$

A$ = OFF (in this case, ch$ and B$ are disregarded)
PAR : (
ABS : Absolute value
EXP : Exponential
LOG : Common logarithm
SQR : Square root
MOV: Moving average
DIF : Differentiation once
INT : Intergration once
DIF2 : Differentiation twice
INT2 : Intergration twice
SLI : Parallel displacement

ch$ = CH1 to CH8
B$ = A to P;
when A$ is set to MOV, a numerical value from 1 to 4000;
when A$ is set to SLI, a numerical value from -4000 to 4000.

[Syntax of the :Y2 to :Y8 commands are same as :Y1 command except "ch$"]

Explanation

(command) Sets the Y1 calculation equation for the waveform processing calculation equation for Z1 according to the character or numerical data.

(query) Returns the current Y1 calculation equation for the waveform processing calculation equation for Z1 as character or numerical data.
A$, B$, and ch$ are used to set up the calculation equation in the following way:
Y1 = A$(ch$+B$)
when A$ is set to MOV or SLI:
Y1 = [MOV or SLI] (ch$,B$)
(Refer to Section 12 "Calculation Functions" of the 8840 manual.)

Example

(1) : CALCULATE : Y1 ABS, CH1, A
Sets up the calculation equation for Y1 to be Y1 = ABS(ch1+a)

(2) : CALCULATE : Y1 MOV, CH1, 50
Sets up the calculation equation for Y1 to be Y1 = MOV(ch1,50)

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for Y2.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax	(command)	: CALCulate : Y2 A$, ch$, B$
	(query)	: CALCulate : Y2?
	(response)	A$, ch$, B$ ch$ = CH1 to CH8, Z1
Explanation	(command)	Sets the Y2 calculation equation for the waveform processing calculation equation for Z2 according to the character or numerical data.
	(query)	Returns the current Y2 calculation equation for the waveform processing calculation equation for Z2 as character or numerical data.
When allowed	In the memory recorder function.	

❑ Sets up and queries the calculation equation for Y3.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax	(command)	: CALCulate : Y3 A$, ch$, B$
	(query)	: CALCulate : Y3?
	(response)	A$, ch$, B$ ch$ = CH1 to CH8, Z1, Z2
Explanation	(command)	Sets the Y3 calculation equation for the waveform processing calculation equation for Z3 according to the character or numerical data.
	(query)	Returns the current Y3 calculation equation for the waveform processing calculation equation for Z3 as character or numerical data.
When allowed	In the memory recorder function.	

❑ Sets up and queries the calculation equation for Y4.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax	(command)	: CALCulate : Y4 A$, ch$, B$
	(query)	: CALCulate : Y4?
	(response)	A$, ch$, B$ ch$=CH1 to CH8, Z1 to Z3
Explanation	(command)	Sets the Y4 calculation equation for the waveform processing calculation equation for Z4 according to the character or numerical data.
	(query)	Returns the current Y4 calculation equation for the waveform processing calculation equation for Z4 as character or numerical data.
When allowed	In the memory recorder function.	

CALCulate

❑ Sets up and queries the calculation equation for Y5.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax
(command) : CALCulate : Y5 A$, ch$, B$
(query) : CALCulate : Y5?
(response) A$, ch$, B$
ch$=CH1 to CH8, Z1 to Z4

Explanation
(command) Sets the Y5 calculation equation for the waveform processing calculation equation for Z5 according to the character or numerical data.
(query) Returns the current Y5 calculation equation for the waveform processing calculation equation for Z5 as character or numerical data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for Y6.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax
(command) : CALCulate : Y6 A$, ch$, B$
(query) : CALCulate : Y6?
(response) A$, ch$, B$
ch$=CH1 to CH8, Z1 to Z5

Explanation
(command) Sets the Y6 calculation equation for the waveform processing calculation equation for Z6 according to the character or numerical data.
(query) Returns the current Y6 calculation equation for the waveform processing calculation equation for Z6 as character or numerical data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for Y7.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax
(command) : CALCulate : Y7 A$, ch$, B$
(query) : CALCulate : Y7?
(response) A$, ch$, B$
ch$=CH1 to CH8, Z1 to Z6

Explanation
(command) Sets the Y7 calculation equation for the waveform processing calculation equation for Z7 according to the character or numerical data.
(query) Returns the current Y7 calculation equation for the waveform processing calculation equation for Z7 as character or numerical data.

When allowed In the memory recorder function.

❑ Sets up and queries the calculation equation for Y8.

(For details, refer to the explanation for the :Y1 command on page 5-70.)

Syntax	(command)	: CALCulate : Y8 A$, ch$, B$
	(query)	: CALCulate : Y8?
	(response)	A$, ch$, B$ ch$=CH1 to CH8, Z1 to Z7
Explanation	(command)	Sets the Y8 calculation equation for the waveform processing calculation equation for Z8 according to the character or numerical data.
	(query)	Returns the current Y8 calculation equation for the waveform processing calculation equation for Z8 as character or numerical data.
When allowed	In the memory recorder function.	

❑ Sets and queries numerical values for coefficients a to p of the waveform processing calculation equation.

Syntax	(command)	: CALCulate : FACTor A$, B
	(query)	: CALCulate : FACTor? A$
	(response)	A$, B<NR3> A$ = A to P B = -9.999E+9 to +9.999E+9
Explanation	(command)	Sets to the given numerical value the one of the coefficients a to p which is designated in A$.
	(query)	Returns as an NR 3 numerical value the current value of that one of the coefficients a to p which is designated in A$. (Refer to Section 12 "Calculation Functions" of the 8840 manual.)
Example	: CALCULATE : FACTOR A, +1.234E+1 Sets the coefficient a to be equal to +1.234E+1	
When allowed	In the memory recorder function.	

CALCulate

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z1.

Syntax
(command) : CALCulate : Z1DIsplay ch$, A$, upper, lower
(query) : CALCulate : Z1DIsplay?
(response) ch$, A$, upper, lower
ch$ = CH1 to CH8, NONE
A$ = AUTO, MANUal
upper, lower = -9.999E+29 to +9.999E+29
(if A$ = AUTO, may be omitted)

[Syntax of :Z2DIsplay to :Z8 DIsplay commands are same as :Z1DIsplay.]

Explanation
(command) Displays the calculated result of the waveform processing calculation equation for Z1 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
(query) Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z1.

Example
: CALCULATE : Z1DISPLAY CH1, MANUAL, +5.000E+0, +0.000E+0
Displays the calculated result of the waveform processing calculation equation for Z1 on channel 1 within the range from 0 volts to 5 volts.

When allowed In the memory recorder function.

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z2.

(Refer to explanation for the :Z1DIsplay command.)

Syntax
(command) : CALCulate : Z2DIsplay ch$, A$, upper, lower
(query) : CALCulate : Z2DIsplay?
(response) ch$, A$, upper, lower

Explanation
(command) Displays the calculated result of the waveform processing calculation equation for Z2 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
(query) Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z2.

When allowed In the memory recorder function.

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z3.

(Refer to explanation for the :Z1DIsplay command on page 5-74.)

Syntax	(command)	: CALCulate : Z3DIsplay ch$, A$, upper, lower
	(query)	: CALCulate : Z3DIsplay?
	(response)	ch$, A$, upper, lower
Explanation	(command)	Displays the calculated result of the waveform processing calculation equation for Z3 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
	(query)	Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z3.
When allowed	In the memory recorder function.	

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z4.

(Refer to explanation for the :Z1DIsplay command on page 5-74.)

Syntax	(command)	: CALCulate : Z4DIsplay ch$, A$, upper, lower
	(query)	: CALCulate : Z4DIsplay?
	(response)	ch$, A$, upper, lower
Explanation	(command)	Displays the calculated result of the waveform processing calculation equation for Z4 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
	(query)	Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z4.
When allowed	In the memory recorder function.	

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z5.

(Refer to explanation for the :Z1DIsplay command on page 5-74.)

Syntax	(command)	: CALCulate : Z5DIsplay ch$, A$, upper, lower
	(query)	: CALCulate : Z5DIsplay?
	(response)	ch$, A$, upper, lower
Explanation	(command)	Displays the calculated result of the waveform processing calculation equation for Z5 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
	(query)	Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z5.
When allowed	In the memory recorder function.	

CALCulate

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z6.

(Refer to explanation for the :Z1DIsplay command on page 5-74.)

Syntax	(command)	: CALCulate : Z6DIsplay ch$, A$, upper, lower
	(query)	: CALCulate : Z6DIsplay?
	(response)	ch$, A$, upper, lower
Explanation	(command)	Displays the calculated result of the waveform processing calculation equation for Z6 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
	(query)	Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z6.
When allowed	In the memory recorder function.	

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z7.

(Refer to explanation for the :Z1DIsplay command on page 5-74.)

Syntax	(command)	: CALCulate : Z7DIsplay ch$, A$, upper, lower
	(query)	: CALCulate : Z7DIsplay?
	(response)	ch$, A$, upper, lower
Explanation	(command)	Displays the calculated result of the waveform processing calculation equation for Z7 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
	(query)	Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z7.
When allowed	In the memory recorder function.	

❑ Sets and queries the display channel for the calculated result of the waveform processing calculation equation for Z8.

(Refer to explanation for the :Z1DIsplay command on page 5-74.)

Syntax	(command)	: CALCulate : Z8DIsplay ch$, A$, upper, lower
	(query)	: CALCulate : Z8DIsplay?
	(response)	ch$, A$, upper, lower
Explanation	(command)	Displays the calculated result of the waveform processing calculation equation for Z8 on the channel designated by ch$ within the range from lower to upper (unit volts - however, if scaling is being performed, in those units).
	(query)	Returns the currently set display channel, scale setting lower limit, and upper limit for display of the calculated result of the waveform processing calculation equation for Z8.
When allowed	In the memory recorder function.	

❑ Enables and disables, and queries, waveform parameter calculation.

Syntax	(command)	: CALCulate : MEASure A$
	(query)	: CALCulate : MEASure?
	(response)	A$ A$ = OFF, ON, EXEC (execute)
Explanation	(command)	Enables or disables, according to character data, the execution of waveform parameter calculation.
	(query)	Returns, as character data, whether execution of waveform parameter calculation is enabled or disabled. Only valid when execution (EXEC) is enabled.
Example	:CALCULATE:MEASURE ON Sets the waveform parameter calculation to ON.	
When allowed	In the memory recorder function.	

❑ Enables and disables, and queries, waveform parameter calculation value Printing.

Syntax	(command)	: CALCulate : MEASPrint A$
	(query)	: CALCulate : MEASPrint?
	(response)	A$ A$ = OFF, ON
Explanation	(command)	Enables or disables, according to character data, print output of waveform parameter calculation values.
	(query)	Returns, as character data, whether execution of print output of waveform parameter calculation values is enabled or disabled.
Example	:CALCULATE:MEASPRINT ON Sets the waveform parameter calculation to ON.	
When allowed	In the memory recorder function.	

CALCulate

❑ Sets and queries waveform parameter calculations.

Syntax

(command) : CALCulate : MEASSet No$, A$, ch$
(query) : CALCulate : MEASSet? No$
(response) No$, A$, ch$

No$ = NO1 to NO4
A$ = OFF

MIN	: minimum value
MAX	: maximum value
MINT	: time to minimum value
MAXT	: time to maximum value
PP	: peak value
AVE	: average value
RMS	: effective value
AREA	:area value
PERI	: periodicity
FREQ	: frequency
RISE	:rising time
FALL	:fall time
XYAREA	:X-Y area value

ch$ = CH1 to CH8, ALL
[During XYAREA]
ch$ = x-axis channel, y-axis channel.

Explanation

(command) Sets the channel and the calculation item of the waveform parameter calculation designated by No$.
(query) Returns the the channel and the calculation item of the waveform parameter calculation designated by No$.

Example

(1) : CALCULATE : MEASSET NO1,MAX,CH1
Sets the calculation to be of the maximum value on channel 1.
(2) : CALC : MEASS NO2,XYAREA,CH1,CH2
If the X-axis is channel 1 and the Y-axis is channel 2, sets X-Y area value calculation.

When allowed In the memory recorder function.

❑ Queries parameter calculation result.

Syntax

(query) : CALCulate : ANSWer? A$, B$
(response) : C$, D<NR 3>

A$ = NO1 to NO4
B$ = NO1 to NO8
C$ = NONE, MIN, MAX, MINT, MAXT, PP, AVE, RMS, AREA, PERI, FREQ, RISE, FALL, XYAREA
D = calculation result

Explanation

(query) Returns the calculation result for the waveform parameter calculation item and result specified by A$ and B$.
When C$ is NONE, there is no calculation result.

Example

(query) : CALCULATE : ANSWER? NO1, CH1
(response) : CALCULATE : ANSWER MIN, -1.2345E-2
(for HEAD ON)
Queries the calculation result of NO1 for the channel1.

When allowed In the memory recorder function.

❑ Enables and disables, and queries, decision for waveform parameter calculation.

Syntax	(command)	: CALCulate : COMP NO$, A$
	(query)	: CALCulate : COMP? NO$
	(response)	NO$, A$ NO$ = NO1 to NO4 A$ = OFF, ON
Explanation	(command)	Enables and disables, according to the character data, the decision of the calculation result of waveform parameter calculation.
	(query)	Returns, as character data, the enablement state of the decision of the calculation result of waveform parameter calculation.
Example	: CALCULATE : COMP NO1, ON Sets the decision of calculation result of NO1 to ON.	
When allowed	In the memory recorder function.	

❑ Sets and queries upper and lower limits for the decision value for waveform parameter calculation.

Syntax	(command)	: CALCulate : COMPArea A$, upper, lower
	(query)	: CALCulate : COMPArea? A$
	(response)	A$, upper, lower A$ = NO1 to NO4 upper, lower = -9.999E+9 to +9.999E+9
Explanation	(command)	Sets, according to the numerical values supplied, the upper limit and the lower limit used when performing a decision on the waveform parameter calculated value designated by A$.
	(query)	Returns, as NR 3 numerical values, the upper limit and the lower limit used when performing a decision on the waveform parameter calculated value designated by A$.
Example	: CALCULATE : COMPAREA NO1, +1.000E+0,-1.000E+0 Sets the decision value for the waveform parameter calculation NO1 to be in the range -1.000E+0 < NO1 < +1.000E+0	
When allowed	In the memory recorder function.	

5-3-12. FDISK command

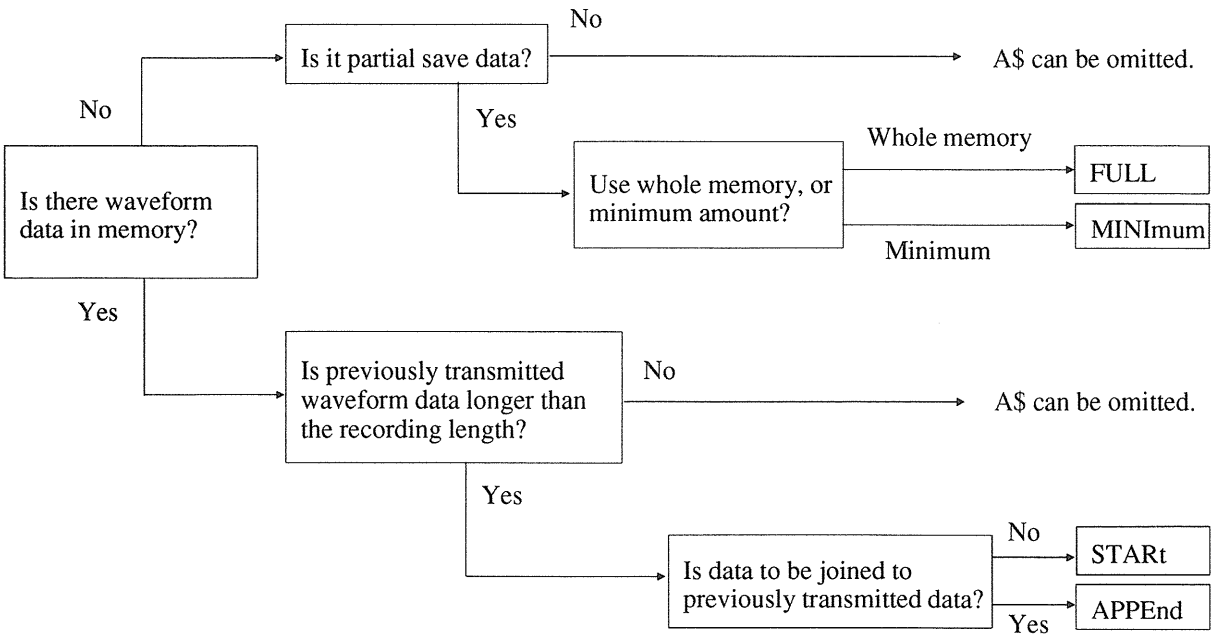
❑ Sets and queries the floppy disk mode.

Syntax	(command)	: FDISK : MODE A$
	(query)	: FDISK : MODE?
	(response)	A$ A$ = OFF,ON
Explanation	(command)	Enables and disables the floppy disk mode.
	(query)	Returns whether the floppy disk mode is currently enabled or disabled.
When allowed	In all functions.	

❑ Performs a load from the floppy disk.

Syntax	(command)	: FDISK : LOAD NO, A$ NO = file number A$ = FULL MINImum STARt APPEnd (A$ may be omitted)
Explanation	(command)	Loads the data in the file numbered No.
Example	: FDISK : LOAD 1 Loads the data of the file numbered 1 on the floppy disk as follows:	
When allowed	When the floppy disk control screen is displayed (after :FDISK:MODE ON).	

- Transmission methods and conditions



Note

- When a FULL/MINImum or STARt/APPEnd selection can be made, of the A$ specification is omitted, the defaults are MINImum and STARt.

❑ Performs a save to the floppy disk.

Syntax (command) : FDISK : SAVE "NAME1$.NAME2$", A$, B$
(when A$ = W)
: FDISK : SAVE "NAME1$.NAME2$", A$
(when A$ = F or A)
NAME1$ = file name (8 characters)
NAME2$ = extension (3 characters)
A$ = type of saved information
Wave: measurement data (MEM, REC, FFT only)
Fune: setting data
Area: waveform decision area (MEM, FFT only)
B$ =saved channels (only when A$ = W)
ALL
CH1 to CH8
LOG A (logic CHA to CHD)

Explanation (command) Saves on the floppy disk the information specified by A$. If an attempt is made to save to a filename that already exists, an execution error is generated.
Replacing double quotation marks (") in the commands, single quotation marks (') can be also used.

Example : FDISK : SAVE"TEST.DAT",W,ALL
Saves all channels of measurement data on the floppy disk under the file name "TEST.DAT"

When allowed When the floppy disk control screen is displayed (after :FDISK:MODE ON).

❑ Deletes a file from the floppy disk.

Syntax (command) : FDISK : DELEte NO
NO = file number

Explanation (command) Deletes from the floppy disk the file whose number is specified by No.

Example : FDISK : DELETE 1
Deletes from the floppy disk the file of No1.

When allowed When the floppy disk control screen is displayed (after :FDISK:MODE ON).

❑ Formats a floppy disk.

Syntax (command) : FDISK : FORMat A$
A$ = 2DD, 2HD, 2HC

Explanation (command) : Formats a floppy disk.

Example : FDISK : FORMAT 2DD
Formats a 2DD floppy disk.

When allowed When the floppy disk control screen is displayed (after :FDISK:MODE ON).

FDISK

❑ Queries how many files are saved on the floppy disk.

Syntax
(query) : FDISK : FILE?
(response) A<NR 1>
A = number of files

Explanation
(query) Returns the number of files which are currently saved on the floppy disk.

When allowed
When the floppy disk control screen is displayed (after :FDISK:MODE ON).

❑ Queries the name of a file saved on the floppy disk.

Syntax
(query) : FDISK : NINFor? NO
(response) "NAME$"
NO = file number
NAME$ = name of the file

Explanation
(query) Returns the filename of the file whose number is specified in No.

Example
(query) : FDISK : NINFOR? 1
(response) : FDISK : NINFOR 1, TEST. DAT

When allowed
When the floppy disk control screen is displayed (after :FDISK:MODE ON).

❑ Queries information about a file saved on the floppy disk.

Syntax
(query) : FDISK : INFOr? "NAME$"
(response) "NAME$", A,B$,"DATE$","TIME$", C
NAME$ = file name
A = file number (if no such file exists, 1)
B$ = type of information saved:
W : measurement data
F : conditions of creation
A : waveform decision area
N : no such file
DATE$ = date of save "year-month-day"
TIME$ = time of save "hour:minute:second"
C = size of file

Explanation
(query) Returns information about the file whose name is specified in NAME$. If no such file exists, returns:
-1, N, "----", "-:-:-", 0
Replacing double quotation marks (") in the commands, single quotation marks (') can be also used.

When allowed
When the floppy disk control screen is displayed (after :FDISK:MODE ON).

5-3-13. GRAPh command

❑ Enables and disables, and queries the enablement of, the graphics editor.

Syntax
(command) : GRAPh : EDIT A$
(query) : GRAPh : EDIT?
(response) A$
A$ = OFF,ON

Explanation
(command) Enables, and disables, the graphic editor mode.
(query) Returns whether or not the graphic editor mode is enabled as character data.

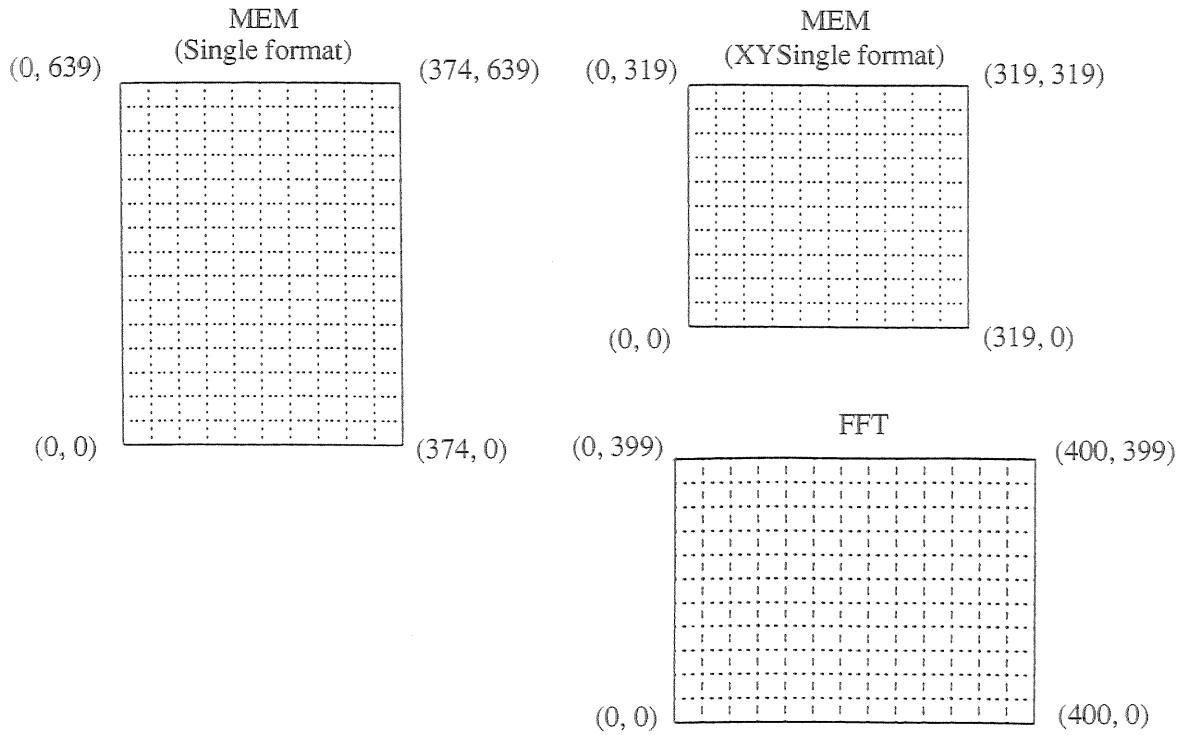
Example
: GRAPH : EDIT ON
Sets the graph editor mode to ON.

When allowed
In the memory recorder function (Single, XYSingle format) and the FFT function.

❑ Line command

Syntax
(command) : GRAPh : LINE X1, Y1, X2, Y2
X1, X2 = x-coordinates
Y1, Y2 = y-coordinates

Explanation
Draws a line from (X1, Y1) to (X2, Y2).
The relationship between the x- and y-coordinates is as follows.



Example
(query) : GRAPH : LINE 10, 20, 100, 200
Draws a line from (10, 20) to (100, 200).

When allowed
In the memory recorder function and the FFT function when in the graphic editor mode.

GRAPh

❑ Parallel Command

Syntax (command) : GRAPh : PARAllel high, low, right, left
high = 0 to 19.97 (div)
low = 0 to 19.97 (div)
right =0 to 14.96 (div)
left = 0 to 14.96 (div)

Explanation Carries out a parallel movement of the drawing.
The "high" and "low" parameters are set in units of 0.03125 steps, and the "right" and "left" parameters in units of 0.04 steps.

When allowed In the memory recorder function and the FFT function when in the graphics editor mode.

❑ Paint command

Syntax (command) : GRAPh : PAINT X, Y
X = x-coordinate
Y = y-coordinate

Explanation Begins solid fill from the point specified by (X, Y).
Refer to the :GRAPh:LINE command for details of X and Y.

When allowed In the memory recorder function and the FFT function when in the graphics editor mode.

❑ Erase command

Syntax (command) : GRAPh : ERASe X1, Y1, X2, Y2
X1, X2 = x-coordinates
Y1, Y2 = y-coordinates

Explanation Erases the line from (X1, Y1) to (X2, Y2).
Refer to the :GRAPh:LINE command for details of X and Y.

When allowed In the memory recorder function and the FFT function when in the graphics editor mode.

❑ Loads a waveform from storage.

Syntax (command) : GRAPh : STORage

Explanation Loads a waveform into the editor.

When allowed In the memory recorder function and the FFT function when in the graphics editor mode.

❑ The reverse command.

Syntax (command) : GRAPh : REVErse

Explanation (command) Reverses the video of the drawing.

When allowed In the memory recorder function and the FFT function when in the graphic editor mode.

❑ The all clear command.

Syntax	(command)	: GRAPh : ALLClear
Explanation	(command)	Clears the entire drawing.
When allowed	In the memory recorder function and the FFT function when in the graphic editor mode.	

❑ Clear command

Syntax	(command)	: GRAPh : CLEAr X1, Y1, X2, Y2 X1, X2 = x-coordinates Y1, Y2 = y-coordinates
Explanation	Clears the rectangle with the points (X1, Y1) and (X2, Y2) at diagonally opposite corners. Refer to the :GRAPh:LINE command for details of X and Y.	
When allowed	In the memory recorder function and the FFT function when in the graphics editor mode.	

❑ Undo command

Syntax	(command)	: GRAPh : UNDO
Explanation	Reverses the effect of the immediately previous editor command.	
When allowed	In the memory recorder function and the FFT function when in the graphics editor mode.	

❑ Saves the drawing (decision area)

Syntax	(command)	: GRAPh : SAVE
Explanation	Saves the decision area created with the editor.	
When allowed	In the memory recorder function and the FFT function when in the graphics editor mode.	

❑ Sets and queries decision area data points.

Syntax	(command) (query) (response)	: GRAPh : POINT X, Y, A : GRAPh : POINT? X, Y X, Y, A all <NR1> X = x-coordinate Y = y-coordinate A = 0, 1
Explanation	Writes the value A at the coordinates indicated by X and Y. Returns the value A at the coordinates indicated by X and Y. A is 1 for a point within the decision area, 0 for a point outside it.	
When allowed	In the memory recorder function and the FFT function when in the graphic editor mode.	

Chapter 6

Example Programs

Sample programs

These sample programs use a Hewlett-Packard HP9816 to control the 8840 on GP-IB device address 5.

Example 1 - Using setting commands

This program makes the following settings, then starts data capture.

Function	:memory recorder
Format	:single
Time axis	:1 ms/division
Shot length	:25 divisions
Trigger source	:OR
Pre-trigger	:5%
Level trigger	:channel 1, at 0.5 V
Slope	:up

```
100  !----------------------------!
110  !  8840 Sample Program No.1  !
120  !----------------------------!
130  !
140  Adr=705                                          !Address=5
150  OUTPUT Adr;":FUNCTION MEM"
160  OUTPUT Adr;":CONFIGURE:FORMAT SINGLE"
170  OUTPUT Adr;":CONFIGURE:TDIV 1.E-3"
180  OUTPUT Adr;":CONFIGURE:SHOT 25"
190                                                   !Set trigger mode
200  OUTPUT Adr;":TRIGGER:MODE SINGLE"
210  OUTPUT Adr;":TRIGGER:SOURCE OR"
220  OUTPUT Adr;":TRIGGER:PRETRIG 5"
230  OUTPUT Adr;":TRIGGER:KIND CH1,LEVEL"
240  OUTPUT Adr;":TRIGGER:LEVEL CH1,0.5"
250  OUTPUT Adr;":TRIGGER:SLOPE CH1,UP"
260  OUTPUT Adr;":TRIGGER:KIND CH2,OFF"
270  OUTPUT Adr;":TRIGGER:KIND CH3,OFF"
280  OUTPUT Adr;":TRIGGER:KIND CH4,OFF"
290                                                   !Start
300  OUTPUT Adr;":START"
310  END
```

Notes on program

Line 140	:Select GP-IB device address 5.
Lines 150-280	:Make the settings.
Line 300	:Start data capture.

Example 2 - Using queries

This program asks the 8840 the current date and time, and what function it is in.

```
100  !----------------------------!
110  !  8840 Sample Program No.2  !
120  !----------------------------!
130  !
140  Adr=705                                    !Address=5
150  OUTPUT Adr;":HEADER OFF"
160  OUTPUT Adr;":SYSTEM:DATE?"                 !Read date
170  ENTER  Adr;Dy,Dm,Dd
180  OUTPUT Adr;":SYSTEM:TIME?"                 !Read time
190  ENTER  Adr;Th,Tm,Ts
200  OUTPUT Adr;":FUNCTION?"                    !Read function
210  ENTER  Adr;Fu$
220  !
230  PRINT Dy,Dm,Dd                             !Print date
240  PRINT Th,Tm,Ts                             !Print time
250  PRINT Fu$                                  !Print function
260  END
```

Example 3 Using service requests

This program reads data until an error occurs.

```
100  !----------------------------!
110  !  8840 Sample Program No.3  !
120  !----------------------------!
130  !
140  Adr=705                                    !Address=5
150  ON INTR 7 GOTO Srq_sub                     !Initialize
160  OUTPUT Adr;"*SRE 32"
170  OUTPUT Adr;"*ESE 60"
180  OUTPUT Adr;"*CLS"
190  ENABLE INTR 7;2
200  OUTPUT Adr;":CURSOR:MODE TRACE"
210  I=0
220 Loop:                                       !Main loop
230  OUTPUT Adr;":CURSOR:APOSITION ";I
240  OUTPUT Adr;":CURSOR:DFREAD? A"
250  ENTER Adr;A$
260  OUTPUT Adr;":CURSOR:DVREAD? A"
270  ENTER Adr;B$
280  PRINT A$,B$
290  I=I+100
300  GOTO Loop                                  !Loop end
310  !
320 Srq_sub:                                    !Service request routine
330  S=SPOLL(Adr)
340  IF BINAND(S,64)<>0 THEN Srq_sub
350  GOTO End_prc
360 Srq_prc: !
370  OUTPUT Adr;"*ESR?"
380  ENTER Adr;C
390  IF BINAND(C,4)<>0  THEN PRINT "Query Error"
400  IF BINAND(C,8)<>0  THEN PRINT "Machine Error"
410  IF BINAND(C,16)<>0 THEN PRINT "Execution Error"
420  IF BINAND(C,32)<>0 THEN PRINT "Command Error"
430 End_prc: !
440  PRINT "End"
450  END
```

Sample output

```
"0s"          "-1.25mV"
"500us"       "933.75mV"
"1ms"         "1.5262V"
              .
              .
"12.5ms"      "452.5mV"
Query Error
Execution Error
End
```

Notes on program

Line 140	:Select GP-IB device address 5.
Line 150	:Attach label Srq_sub for service request handling.
Line 160	:Set service request enable register. Value 32 means enable bit 5 (event summary) of status byte to cause a service request when it goes to 1.
Line 170	:Set standard event status enable register. Value 60 means enable bits 2, 3, 4 and 5 so that bit 5 of the status byte goes to 1 when there is a query error, machine error, execution error, or command error. Various other combinations of settings for *SRE, *ESE and :ESE0 are possible to vary the service request issuing conditions.
Line 180	:Clear the status byte, standard event status register, and event status register to zero.
Line 190	:Enable service request interrupt.
Line 200	:Display cursor in trace mode.
Lines 210-300	:Read the data values at the A cursor positions, moving the A cursor along by 100 points at a time. If an error occurs, call the service request subroutine Srq_sub. Line 230 sets the A cursor position to I, then lines 240 to 270 read the time (frequency) and voltage at the A cursor position. Line 290 advances the cursor position by 100.
Lines 320-450	:Service request handling routine
Line 330	:Poll device on address 5. Read status byte.
Lines 340-350	:If bit 6 of the status byte was 1 (i.e. if device on address 5 has generated a service request), go to label Src_prc. Otherwise (i.e. a different device generated the service request) go to label End_prc.
Lines 370-380	:Read standard event status register.
Lines 390-420	:Display a message indicating the type of error (corresponding to bits 2 to 5).
Lines 440-450	: Display ending message, and end.

Example 4 - Saving captured data to a disk

```
100  !----------------------------!
110  !  8840 Sample Program No.5  !
120  !----------------------------!
130  !
140  Adr=705                                    !Address=5
150  ON ERROR GOTO Err_prc                      !Initialize
160  !
170  PRINT "<Load Saved Data>"
180  OUTPUT Adr;":HEADER OFF"
190  INPUT "Channel(1-16)=",Ch$
200  LINPUT "File Name=",Na$
210  ASSIGN @File TO Na$&":HP8290X,700,1"
220  ENTER @File;N
230  !
240  OUTPUT Adr;":MEMORY:MAXPOINT?"
250  ENTER Adr;N
260  IF Mx<N THEN
270    Sh$=VAL$(N/100)
280    OUTPUT Adr;":CONFIGURE:SHOT "&Sh$
290    OUTPUT Adr;":MEMORY:PREPARE"
300  END IF
310  !
320  OUTPUT Adr;":MEMORY:POINT CH"&Ch$&",0"
330  FOR I=0 TO N
340    ENTER @File;Dt$
350    PRINT I,Dt$;
360    OUTPUT Adr;":MEMORY:VDATA "&Dt$
370  NEXT I
380  OUTPUT Adr;":DISPLAY:CHANGE DISPLAY"
390  PRINT "Completed"
400  STOP
410  !
420 Err_prc:                                    !Error routine
430  PRINT "Error !!"
440  END
```

Notes on program

Line 140	:Select GP-IB device address 5.
Line 150	:Allocate label Err_prc for error handling.
Line 180	:Switch headers off.
Lines 190-200	:Query number of data items stored, and read into Mx.
Lines 220-400	:If Mx 0, execute lines 230 to 370. Otherwise go to 390 to end.
Lines 240-250	:Input the channel number and filename.
Line 260	:Set up to read data from item 0 on the specified channel.
Lines 280-290	:Open file.
Line 300	:Write number of data items at beginning of file.
Lines 310-360	:Read voltage values one at a time from 0 to Mx, saving to file.
Lines 430-440	:If an error occurs, display a message and end.

Example 5 - Loading and displaying the data saved by example program 4

```
100  !----------------------------!
110  !  8840 Sample Program No.4  !
120  !----------------------------!
130  !
140  Adr=705                                    !Address=5
150  ON ERROR GOTO Err_prc                      !Initialize
160  !
170  PRINT "<Save Storage Data>"
180  OUTPUT Adr;":HEADER OFF"
190  OUTPUT Adr;":MEMORY:MAXPOINT?"
200  ENTER Adr;Mx
210  !
220  IF Mx>0 THEN
230    PRINT "Max Point=";Mx
240    INPUT "Channel(1-16)=",Ch$
250    LINPUT "File Name=",Na$
260    OUTPUT Adr;":MEMORY:POINT CH"&Ch$&",0"
270    !
280    CREATE ASCII Na$&":HP8290X,700,1",256
290    ASSIGN @File TO Na$&":HP8290X,700,1"
300    OUTPUT @File;Mx
310    FOR I=0 TO Mx
320      OUTPUT Adr;":MEMORY:VDATA? 1"
330      ENTER Adr;Dt
340      PRINT I,Dt
350      OUTPUT @File;Dt
360    NEXT I
370    PRINT "Completed"
380  ELSE
390    PRINT "No Storage Data !!"
400  END IF
410  STOP
420  !
430 Err_prc:                                    !Error routine
440  PRINT "Error !!"
450  END
```

Notes on program

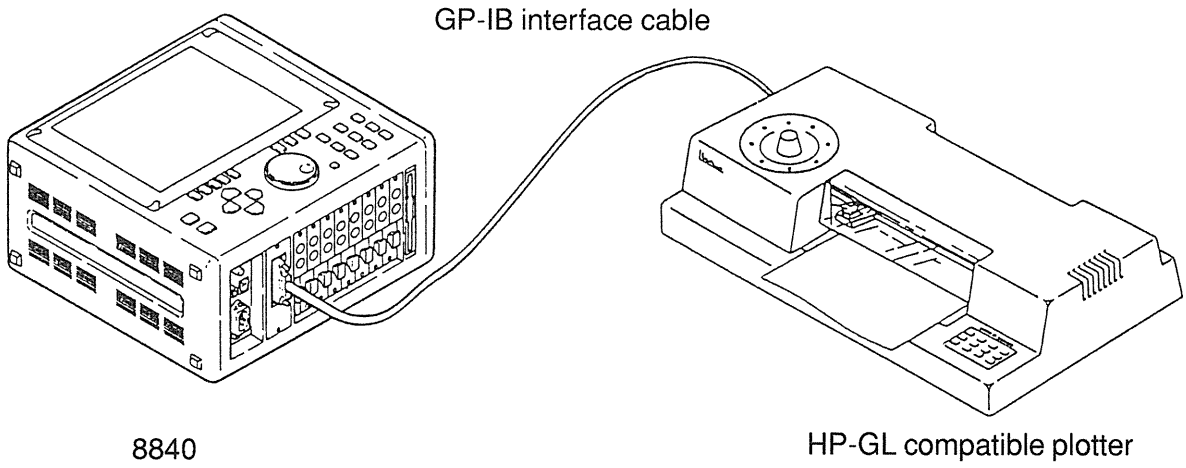
Line 140	:Select GP-IB device address 5.
Line 150	:Allocate label Err_prc for error handling.
Line 180	:Switch headers off.
Lines 190-200	:Input the channel number and filename.
Line 210	:Open file.
Line 220	:Read the value written at the beginning of the file into N.
Lines 240-250	:Read the available data memory into Mx.
Lines 260-300	:If Mx N, insufficient memory to hold data in 8840. Use :CONFIGURE:SHOT and :MEMORY:PREPARE to reserve required memory.
Line 320	:Set up to write data from item 0 on the specified channel.
Lines 330-370	:Write data values read from file to storage memory in 8825.
Line 380	:Redraw display, ready to display data values written.
Lines 390-400	:Display completion message, and end.
Lines 420-440	:If an error occurs, display a message and end.

Chapter 7

Plotter Output

Settings for plotter output

It is possible to use an HP-GL compatible plotter to output the waveform from the 8840 display screen.



Connect the plotter to the 8840 using the GP-IB interface cable.

The settings for the plotter are all carried out using the "system" screen (PLOTTER) on the 8840.

Method

1. Press the SYSTEM key, to display the "SYSTEM" screen.

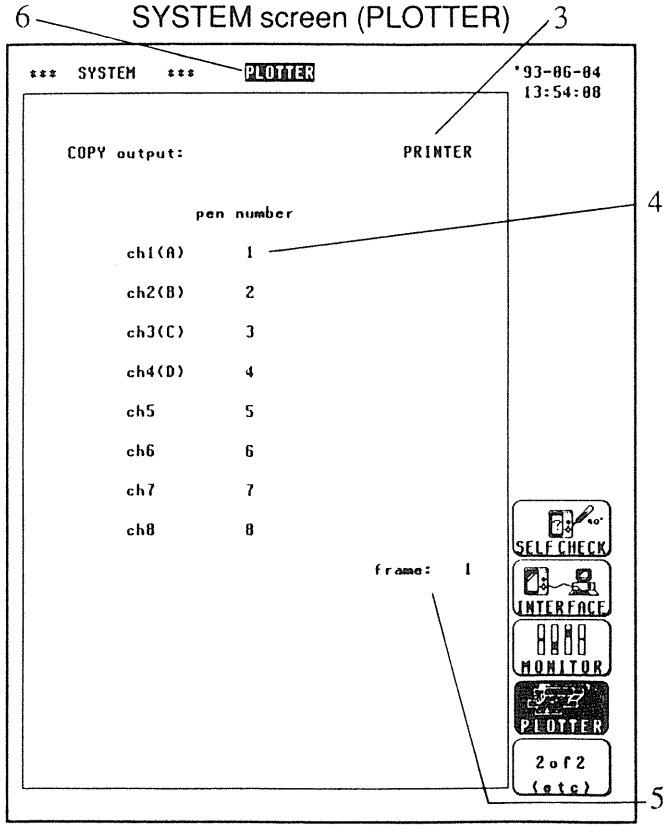
2. Press F5 (1of2), then press F4 (PLOTTER) to get the plotter settings screen.

   Next, use the cursor keys to move the flashing cursor through the items in the order listed on the right, and use the function keys to make the settings.

3. Plotter output settings.

   Select whether pressing the COPY key makes a normal hard copy of the screen on the printer, or makes a copy of the waveform on the plotter.

   According to the indications for the function keys, select the required option.



Function key indication *Meaning*

: Make a screen copy on the printer.

: Draw the waveform on the plotter.

: Save a screen displays on the floppy disk (monochrome)*

: Save a screen displays on the floppy disk (color)*

* : refer to the manual for the 8840 FFT analyzer.

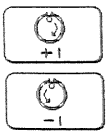
4. Pen number for each channel

   Set the pen number to be used by each channel.

   Choose from pens 1 through 8 or –, to assign a color for each waveform.

   Use F1 and F2 or the Jog control knob to make the settings.

Function key indication *Meaning*



1 to 8, – (no pen)

Notes: • Logic channel groups A through D and graph 1 through 4 in X-Y plot get the same pen assignments as the corresponding analog channels 1 through 4.

• If there is a comment appended to a channel, it will also appear on the plot. (For more details of comments, see Section 13-5 "Appending Comments" in the instruction manual for the 8840.)

5. Frame pen number

   Set the pen number to be used for the frame, for scales, and for indicating the time axis range, for example.

   The setting procedure is the same as in step 4.

   This completes settings in the SYSTEM screen (PLOTTER). Next carry out the GP-IB settings.

6. Press F2 (INTERFACE), to display the interface settings screen.

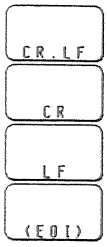
7. GP-IB operation mode

   Press F2 (TALK ONLY) to select talk only mode for the plotter.

8. Delimiter selection.

   The delimiter sequence required depends on the plotter being used. Consult the documentation accompanying the plotter, then use the function keys to make the selection.

Function key indication *Meaning*



: CR+LF (EOI)

: CR (EOI)

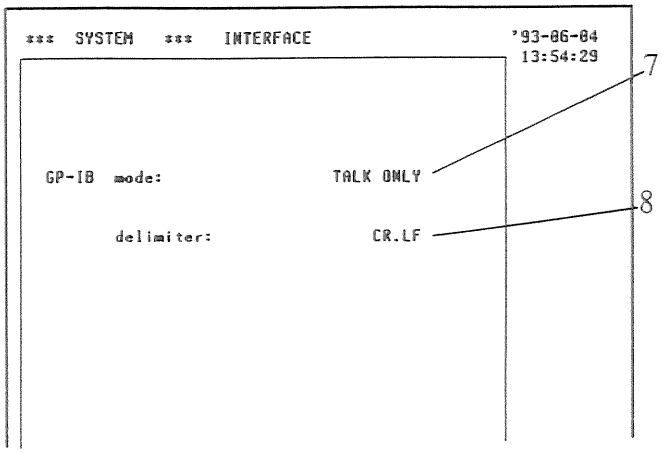
: LF (EOI)

: (EOI)

Set the plotter to listen-only mode.

This completes the 8840 settings.

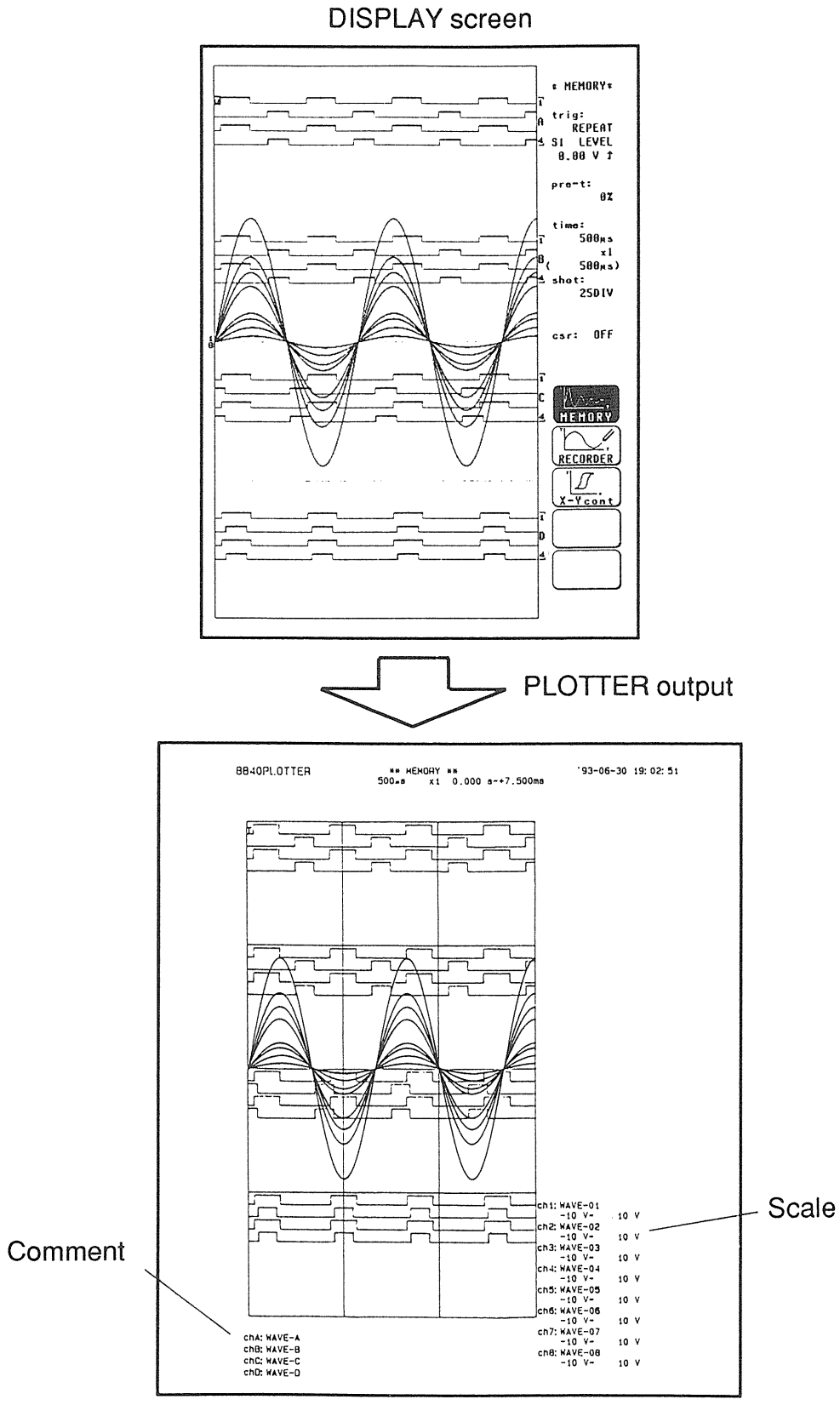
SYSTEM screen (INTERFACE)



9. Press the DISP key to show the DISPLAY screen.

10. Press the COPY key to begin plotter output.

The plotter output appears as shown in the following figure.



Notes

- In overwriting mode, only the last waveform captured will be printed out.
- The grid is drawn every 5 divisions at the "STANDARD" setting, and every 1 division at the "FINE" setting.
- The waveform in the recorder function and the compression waveform are traced twice.

Chapter 8

Device Compliance Statement

The following information relates to compliance with the IEEE 488.2 standard.

① IEEE 488.1 interface functions

- These are detailed in Section 2-2, "Interface functions".

② Operation with a device address other than 0 through 30

- It is not possible to set to other than 0 through 30.

③ Timing of changed device address recognition

- A change of address is recognized immediately after powering on.

④ Device settings at power on, including all commands which further restrict the initial setting.

- The status information is cleared. However, the command :MEMORY :POINT are all reinitialized, and all other items are preserved.

⑤ List of message exchange options

(a) Input buffer capacity and operation:

- The 8840 has an input buffer of 512 bytes capacity.
  If the data accumulated in this buffer exceeds 512 bytes the buffer full, and until a space again becomes available in the buffer the IEEE 488.1 bus goes into the waiting state.

(b) Queries to which multiple response message units are returned:

- There are no query to return multiple response message.

(c) Queries producing responses as syntax checking is performed:

- On the 8840, all queries produce responses when syntax checking is performed.

(d) Whether any queries produce responses when read:

- There are no queries which produce response messages at the instant they are read in by the controller.

(e) Whether any commands are coupled:

- There are no relevant commands.

⑥ Summary of functional elements for use when constructing device specific commands, and whether compound commands or program headers can be used:

- Program message, program message terminator, program message unit, program message unit separator, command message unit, query message unit, command program header, query program header, program data, character program data, decimal program data, chapter string program data, and compound commands program data.

⑦ Buffer capacity limitations for block data

- Block data is not used.

⑧ Summary of program data elements used in expressions, and deepest nesting level allowable in sub-expressions, including syntax restrictions imposed by the device.

- Sub-expressions are not used. Character data and decimal data are the only program data elements used.

⑨ Response syntax for queries

- Response syntax is detailed in Chapter 4 "GP-IB Commands Summary" and Chapter 5 "Command Reference".

⑩ Transmission congestion relating to device-to-device messages which do not conform to the general principles for basic response messages

- There are no device to device messages.

⑪ Response capacity for block data

- Block data does not appear in responses.

⑫ Summary of standard commands and queries used

- This appears in Chapter 4-1 "GP-IB Commands Summary".

⑬ Device state after a calibration query has been completed without any problem

- The "*CAL?" query is not used.

⑭ When using the "*DDT" command, the maximum length of block used in a trigger macro definition

- The "*DDT" command is not used.

⑮ When a macro command is being executed, the maximum length of macro label, the maximum length of block for defining a macro, and how echoing is managed when expanding a macro

- Macros are not used.

⑯ For queries related to identification, explanation of the response to the "*IDN?" query

- This is detailed in Chapter 5 "Command Reference".

⑰ Capacity of the user data storage area reserved for when the "*PUD" command and the "*PUD?" query are being executed

- The "*PUD" command and the "*PUD?" query are not used. Further, there is no user data storage area.

⑱ Resources when the "*RDT" command and the "*RDT?" query are being used

- The "*RDT" command and the "*RDT?" query are not used.

⑲ Conditions which are influenced when "*RST", "*LRN?", "*RCL", and "*SAV" are used

- "*LRN?", "*RCL", and "*SAV" are not used. The "*RST" command returns the 8840 to its initial state.

⑳ Scope of the self-testing executed as a result of the "*TST?" query

- Checks the internal ROM and RAMs.

㉑ Additional organization of the status data used in a device status report

- This is detailed in Section 3-6 "The Status Byte and the Event Registers".

㉒ Whether commands are overlap or sequential type

- All the commands are sequential commands except :ABORT command. An :ABORT command is executed instantly as soon as it is transmitted.

㉓ Criterion relating to the functions required at the instant that the termination message is produced, as a response to each command

- Termination occurs when the command has been parsed.

Chapter 9

Installation

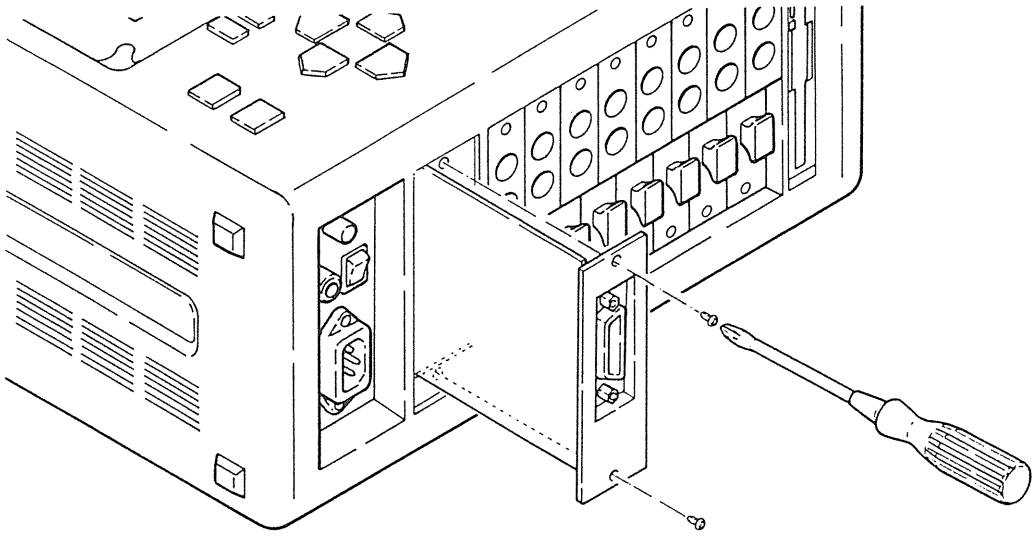
This section describes how to install the 9587 GP-IB interface.

① Power off the 8840 main unit and disconnect the power cord. When using the 8840-01, power off also the DC power supply and disconnect the cord for the DC power supply.

⚠ DANGER

To avoid the danger of electric shock or damage to the equipment, always disconnect the power supply cord and install the GP-IB interface.

② Fix the GP-IB interface with two screws, using Phillips screwdriver as shown in the figure below.



Note

If remove the GP-IB interface, remove the two fixing screws on the panel and connect the GP-IB cable again.

After fasten the fixing screws on the connector, pull it out of the main unit.

⚠ DANGER

- To avoid the danger of electric shock, never operate the 8840 unit with an input unit removed.
- If you should wish to use the unit after removing an input unit, fit a blanking panel over the opening of the removed unit.

Note

If you have no blanking panel, contact your nearest dealer.

Appendix

Appendix. Troubleshooting GP-IB faults

Check the items in the following table in the event of operating problems with the GP-IB interface.

Symptom	Likely causes and remedies
The GP-IB does not operate at all.	• Is the cable properly connected? • Is the GP-IB address of the 8840 unit correctly set? Does it clash the address of other equipment on the same bus? • Are all the devices that are connected powered on?
The 8840 keys stop working after using GP-IB communications.	• Press the LOCAL key on the front panel to end the remote operating state. • Has an LLO (local lock-out) command been sent to the 8840? Send a GTL command to return to the local state.
An attempt to read data using the INPUT@ (ENTER) statement causes the GP-IB bus to hang.	• Each and every INPUT@ (ENTER) statement must be preceded by a query. • Is the query transmitted incorrect?
Although a command was transmitted, the unit did not operate.	• Use the "*ESR?" query to check the standard event status register for anomalies.
Even though a number of queries were sent, only one response was received.	• Has an error occurred? • The response should be read immediately after each query. To read several responses in one operation, the corresponding queries must be combined into a single line using the message separator.
A service request is sometimes not issued.	• Are the service request enable register and the event status enable registers set correctly? • At the end of the SRQ handling routine, use a "*CLS" command to clear all of the event registers. If a bit in the event registers is not cleared, the same event occurring again will not generate a service request.

INDEX

C

D

E

I

L

M

O

Q

R

S

T

— Command —

Jul.	1993	Edition 1
Jul.	1993	Revised edition 1
Jan.	1995	Revised edition 2

HIOKI E.E. CORPORATION Printed in Japan

HIOKI E.E. CORPORATION

81 Koizumi, Ueda, Nagano 386-11, Japan
TEL 0268-28-0562 FAX 0268-28-0568

9587A981-02 95-01-001U 783201182