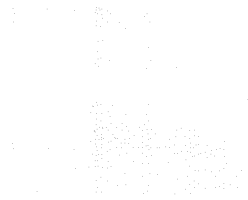
HIOKI

INSTRUCTION MANUAL

9518

GP-IB Interface

HIOKI E.E. CORPORATION



Contents

Introduction

Thank you for buying this optional 9518 GP-IB interface for the Hioki 3502 Capacitance HiTester or the Hioki 3510 Inductance HiTester. To get the maximum performance from this unit, and ensure trouble-free operation, read this manual first.

To ensure safe operation, and in order to exploit its functionality to the full, please follow the directions in this section carefully.

The following three levels of heading are also used in this manual to prioritize warnings:

⚠ **DANGER** .... This indicates a point where an error during use could pose serious danger to the operator.

⚠ **WARNING** ... This indicates a point where an error could damage the unit or pose a hazard to the operator.

NOTE ..... This indicates an important point relating to operation.

Chapter 1

Before Use

1-1 Check of External Appearance and Accessories

When the 9518 GP-IB interface unit is delivered, check that the following items are included in the package:

(1) 9518 GP-IB interface
(2) This instruction manual

In the event of any damage, or if anything is missing, contact your nearest Hioki service representative without delay.

1-2 Shipping Precautions

If reshipping the unit, preferably use the original packing. If this is not available, use the following procedure.

(1) Wrap the unit in plastic sheeting.
(2) After wrapping cushioning material around the unit, pack it into a cardboard box, and then seal up the box with adhesive tape.

1-3 Points for Attention During Use

(1) If you change the device address of the 3502/3510 while using it, you should immediately turn the power off and on again. If you do not do so, the address change will not be registered by the bus, and problems will occur.
(2) Always be sure to secure the GP-IB cable to the 9518 unit by tightening up the fixing screws provided.
(3) Program messages sent just after the power has been turned on are executed after the self test has terminated.
(4) It is vital that the proper data format is used when inputting commands with data values to the 3502 or 3510 unit.
(5) The values given for command processing times are the values measured when the long form of the command and the designated data format are used, and headers for response messages are enabled.
(6) Commands specific to the 3502 or 3510 unit are all sequential commands.
(7) For details of the various functions, refer to the instruction manual for the 3502 or 3510 unit.

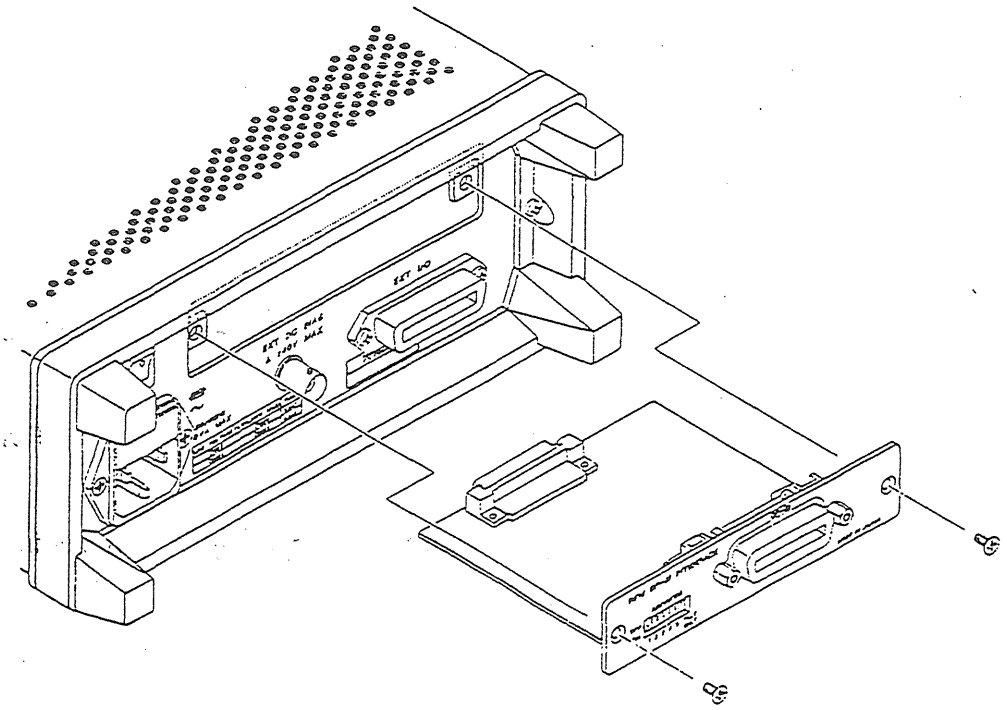
1-4 Installing and Removing the GP-IB Interface

⚠ DANGER

When installing the 9518 GP-IB interface unit in the 3502 Capacitance HiTester or the 3510 Inductance HiTester, be absolutely sure that the power cable and the connectors to the 3502 or 3510 have been removed first, in order to prevent electric shock to the operator and also damage to the unit.

The space for fitting the 9518 GP-IB interface in the rear panel of the 3502 or 3510 is covered with a blanking plate. Follow these three steps to install the 9518 interface:

(1) Remove the fixing screws, and take off the blanking plate.
(2) Insert the 9518 GP-IB interface into the exposed slot in the rear of the 3502 or 3510 by sliding its guide rails along the rails provided on the sides of the inside of the slot. (See the figure below)
(3) Push the 9518 firmly into place, and fix with the screws removed in step (1).



Note: On the rear panel of the 3510 there is no EXT DC BIAS terminal (BNC connector).

⚠ DANGER

To prevent electric shock to the operator, never use the 3502 or 3510 without either the GP-IB interface or the blanking plate in place.

Chapter 2

Overview

2-1 Introduction to the GP-IB Interface

By connecting the 9518 GP-IB interface to the 3502 Capacitance HiTester or the 3510 Inductance HiTester, it is possible to control all the functions of the 3502 or 3510 (except for powering on) via the GP-IB bus.

IEEE Standard 488.1-1987

Further, the 9518 is designed with reference to the following standard:

IEEE Standard 488.2-1987

NOTE

On the 9518, if the output queue becomes full, it is cleared and a query error is generated. This differs from the IEEE 488.2 specification, which only stipulates the clearing of the output queue and the outputting of a query error when a deadlock state occurs, that is, when both the input buffer and the output queue have become full, and continuation of processing has become impossible.

2-2 Features

(1) All of the functions of the 3502 or 3510 main unit, except for powering on, can be controlled via the GP-IB interface.
(2) When the comparator function is being used, the "D" setting can be made to an accuracy of five decimal places (only when averaging is being performed).
(3) The beeper sound can be turned on and off.
(4) The unit can be reset.
(5) IEEE 488.2-1987 standard (essential) commands can be used.
(6) With the talk-only function test values can be printed out without using any controller, by connecting the 9518 to a printer equipped with a GP-IB interface having a listen-only function.

2-3 Specifications

Interface functions provided

SH1	All source handshake functions
AH1	All accepter handshake functions
T5	Basic talk functions Serial poll function Talk-only mode is not provided. The talker cancellation function with MLA (My Listen Address) is provided.
L4	Basic listener functions Listen-only mode is not provided. The listener cancellation function with MTA (My Talk Address) is provided.
SR1	All service request functions
RL1	All remote/local functions
PP0	Parallel polling is not provided.
DC1	All device clear functions
DT1	All device trigger functions
C0	The controller function is not provided.

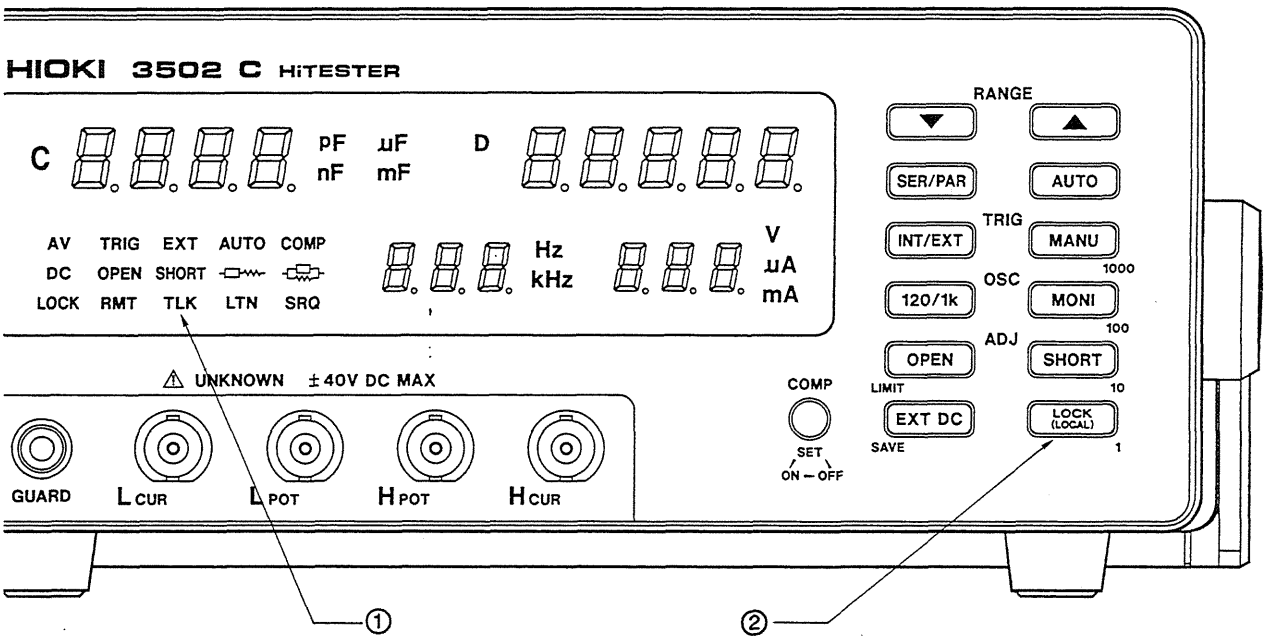
ASCII codes are used.

Chapter 3

Controls and Connections

3-1 Controls and Connections

(1) 3502 Front Panel



① GP-IB status display

These indicators show the GP-IB control state:

RMT: Remote	SRQ: Service request
TLK: Talker	LTN: Listener

② LOCAL key (remote release key)

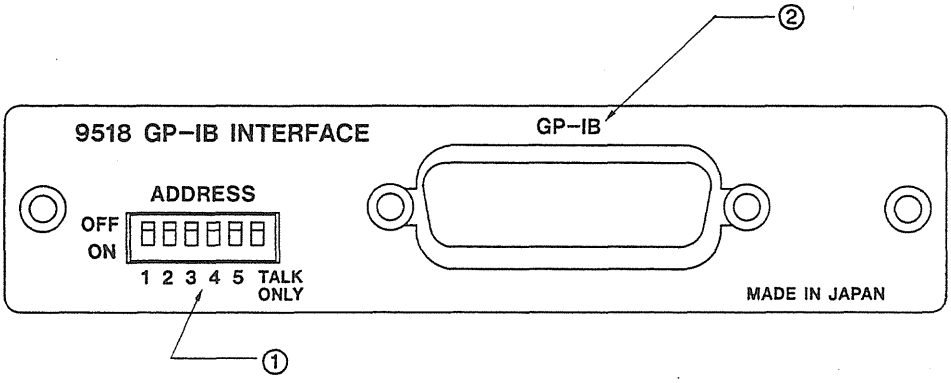
Press this LOCK(LOCAL) key to release the remote state of the GP-IB interface and to resume the local state.

However, this LOCK(LOCAL) key is disabled if the GP-IB controller has put the unit into the local lock out state. (Pressing the key has no effect.)

Note

The front panel of the 3510 is identical to that of the 3502 insofar as ① and ② are concerned.

(2) 9518 GP-IB interface outer panel



① Address switches

These are used to set the device address of this 3502/3510 unit on the GP-IB bus. For how to set these switches, refer to Section 4-1, "Setting the GP-IB Device Address" (page 4-2).

② GP-IB connector

Connect the GP-IB cable to this connector.

⚠ DANGER

In order to prevent any danger of electric shock to the operator, check carefully that the power cable and the connectors to the 3502/3510 have been removed first, before connecting the GP-IB cable to this connector.

Chapter 4

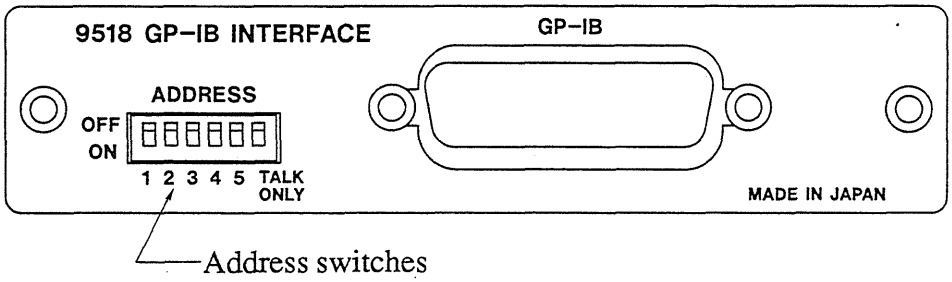
Operation

4-1 Setting the GP-IB Device Address

- The address of the 3502/3510 unit (called the device) on the GP-IB bus can be set to any number from 0 to 30.
- Use the DIP switches on the GP-IB panel to set the device address.
- On dispatch from the factory, this address is initially set to 1.
- If this address is (apparently) set to 31, i.e. if all the switches are in the ON position, then the bus lines of the 3502/3510 are disabled.

NOTE

If you change the bus address while the 3502/3510 is being used, then you should immediately turn the power off and on again. If this is not done, the address will not be changed to the new one.

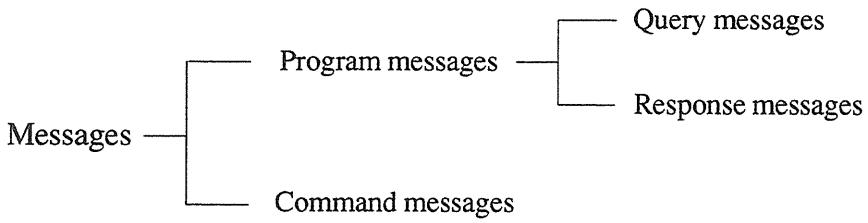


0 . . . OFF 1 . . . ON

Address	Switch settings 12345	Address	Switch settings 12345	Address	Switch settings 12345
0	00000	10	01010	20	00101
1	10000	11	11010	21	10101
2	01000	12	00110	22	01101
3	11000	13	10110	23	11101
4	00100	14	01110	24	00011
5	10100	15	11110	25	10011
6	01100	16	00001	26	01011
7	11100	17	10001	27	11011
8	00010	18	01001	28	00111
9	10010	19	11001	29	10111
				30	01111

4-2 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 device from the controller, while response messages are those sent from the device to the controller.

(1) Program messages

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.

(2) Response messages

After a query message has been received, a response message is produced the moment that its syntax has been checked. It is possible, when and only when headers are off, to change the message unit separator of response messages from the semicolon ";" to the comma "," by using the command "TRANsmit:SEParator". Initially this separator is set to the semicolon ";". (a space is represented by "␣" in the examples)

Headers on: "L␣+01.23E+00;D␣+0.0456E+00"

Headers off: "+01.23E+00;␣+0.0456E+00"

↓

Headers off: "+01.23E+00,␣+0.0456E+00"

A detailed explanation is given on page 5-74.

4-3 Command Syntax

The names of commands for the 3502 or 3510 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.

If the unabbreviated long form of the command consists of two or more words, then its abbreviated short form consists of the first letter of the first word (shown with a single upper case letter) followed by a header portion of the second word (the portion shown in upper case letters). For example, the abbreviated short form of the "COMParator: First_LIMit" command is as follows:

COMParator: FLIMit

Response messages generated by the 3502 or 3510 are in long form and in upper case letters.

4-4 Headers

Whether or not headers are prefixed to response messages is set by the "HEADer" command. It is essential to prefix headers to program messages.

(1) Command program headers

There are three types of command: simple commands, compound commands, and standard commands.

① Simple command header
This header is a sequence of letters and digits, starting with a letter.

Example: HEADer

② Compound command header
This header is made up from a plurality of simple command type headers marked off by colons.

Example: COMParator:RANGe

③ Standard command header
This header begins with an asterisk to indicate that it is a standard command, and continues with a standard command stipulated by IEEE 488.2.

Example: *RST

(2) Query program headers

These are for commands used for interrogating the device about the results of operations, about measured values, or about the current states of settings for the device.

As shown by the following example, they 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. Queries are possible in each of the three previously described types of command form.

Example: COMParator?

(3) Response messages

Response messages relating to queries are made up from header portions (which also may be absent if the headers are disabled) 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.

4-5 Message Terminators

The 3502/3510 recognizes either a linefeed character (0AH) or the EOI signal, or both, as message terminators.

To terminate a response message, the 3502/3510 always provides the appropriate EOI signal, and also sends a terminating character sequence. By the use of the "TRANsmit:TERMinator" command either of the following can be selected as response message terminator sequence:

(1) LF (linefeed only)

(2) CR + LF (carriage return plus linefeed)

The initial selection is (1).

A detailed explanation of the "TRANsmit:TERMinator" command is given on page 5-76.

4-6 Separators

(1) 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: ":AUTO␣ON [;] :DISPlay␣LD [;] *IDN?"

(2) Header separator

In a message which has a header and data, a space (represented by "␣" in the examples) is used as the header separator to separate the header from the data.

Example: ":DISPlay:MONitor [␣] VOLTage"

(3) Data separator

If a message has several data items, commas are required as data separators for separating these data items from one another.

Example: ": COMParator: SLIMit␣<lower limit>[,]<upper limit>"

4-7 Data Formats

The 3502/3510 uses character string data and decimal numeric data, and the type used varies according to the command in question.

(1) Character data

Character string data must always begin with an alphabetic character, and the characters following can be either alphabetic characters or numerals. Although in character data either upper case letters or lower case letters are accepted, response messages output by the 3502/3510 are always in upper case letters.

Example: ":TRIGger␣int ERNAL"

(2) Decimal data

The numeric data values are all represented in decimal, in three formats identified as NR1, NR2 and 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 limit which the 3502 or 3510 can deal, it is rounded off. (5 and above is rounded up; 4 and below is rounded down).

NR1 format - integer data.

Examples: +12, -23, 34

NR2 format - fixed point numbers.

Examples: +1.23, -23.45, 3.456

NR3 format - floating point numbers.

Examples: +1.E-2, -2.3E+4

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

When the 3502 or 3510 is receiving it accepts NRf format, but when it is sending response messages it utilizes whichever one of the formats NR1 to NR3 is indicated in the particular command. Examples:

":COMParator:RANGe␣6"

":COMParator:RANGe␣+6.012"

":COMParator:RANGe␣0.0006E4"

4-8 Abbreviation of Compound Commands

When several compound commands have a common head portion, for example :COMP:RANG and :COMP:TYPE, then, when and only when writing them directly following on from one another, this common portion (:COMP: in this example) can be omitted from each command except for the first one. This common portion is called "the current path", by analogy with the general concept of the current directory in the directory structure of UNIX or MSDDOS, and until it is cleared the analysis of following commands is performed by deeming them to be preceded by the current path which has been curtailed in the interests of brevity. This manner of using the current path is shown in the following example:
Normal expression:

":COMParator:RANGe␣6;:COMParator:TYPE␣3"

Abbreviated expression:

" : COMParator: RANGe␣6;TYPE␣3"

└──This becomes the current path, and can be curtailed from the following commands.

The current path is cleared when the power is turned on, when a system reset is performed by key input, when a colon ":" appears at the start of a command, and when a message terminator is detected.
Messages of standard command form can be executed without relation to the current path. Further, they have no effect upon the current path.
It is not necessary to prefix a colon ":" at the start of headers of simple commands and compound commands. However, in order to prevent confusion with abbreviated forms and mistakes in operation, it is recommended practice always to prefix ":" to headers.
With the 3502 or 3510, there are four possible current paths:

":COMParator:", ":CORRection:", ":DISPlay:", and ":TRANsmit:"

4-9 Output Queue

Response messages accumulate in the output queue and are read out as data and cleared by the controller.
The output queue is also cleared in the following circumstances:

- When a device clear is issued.
- When the power is turned off and turned on again.
- When the 3502 or 3510 unit is reset by a key press.

The 3502 or 3510 has an output queue of 800 bytes capacity. If the response messages overflow this limit of 800 bytes, a query error is generated, and the output queue is cleared. Further, if a new message is received while the output queue still contains data, the output queue is cleared, and a query error is generated.

4-10 Input Buffer

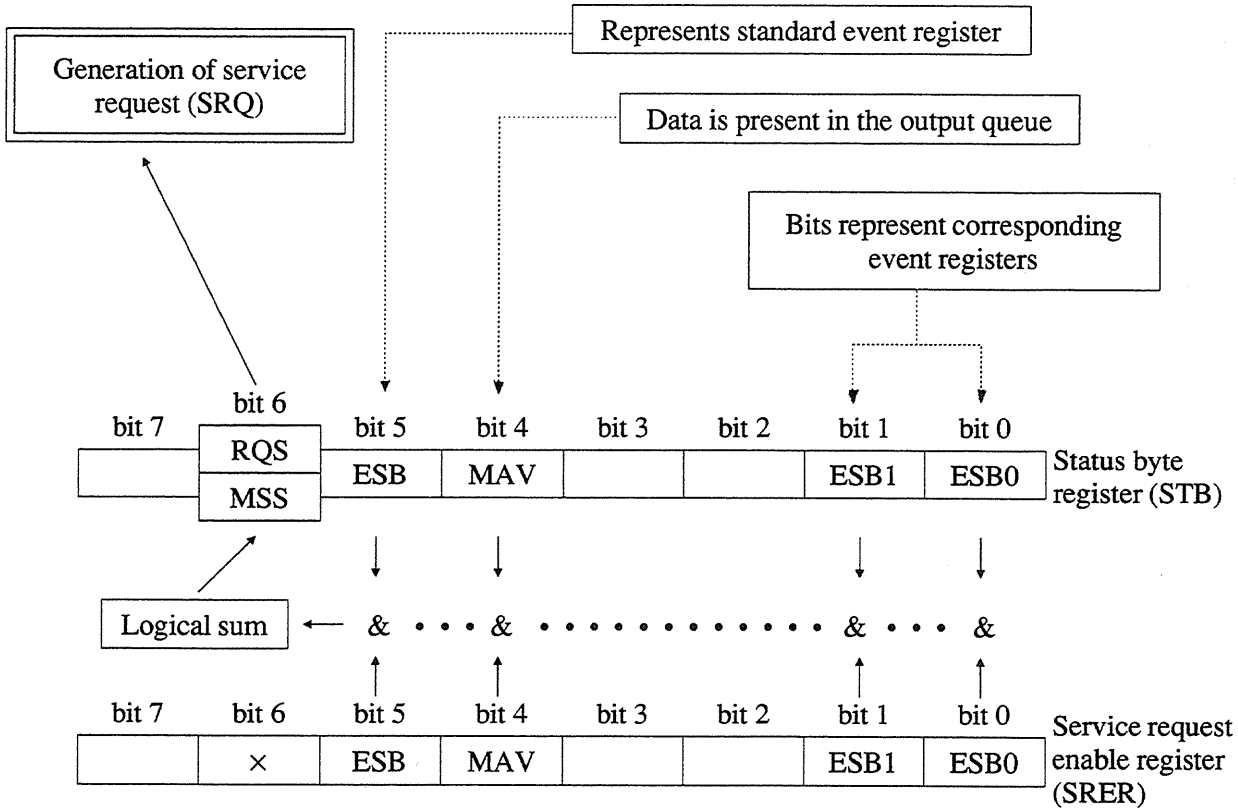
The 3502 or 3510 has an input buffer of 300 bytes capacity. Messages which are received are put into this buffer and executed in order. If the data accumulated in this buffer exceeds 300 bytes the buffer becomes full, and until a space again becomes available in the buffer the GP-IB interface bus goes into the waiting state.

4-11 Status Model

In its implementation of the serial polling function using service requests, the 3502 or 3510 employs the status model specified by IEEE 488.2.

The term "event" refers to any phenomenon which generates a service request.

Generation of service requests



The status byte register holds information relating to the event registers and the output queue. It is further possible to use the service request enable register as a mask to select the items required. If any of the bits selected by the mask becomes 1, bit 6 (the master summary status or MSS bit) is also set to 1, an RQS message is generated, and this generates a service request.

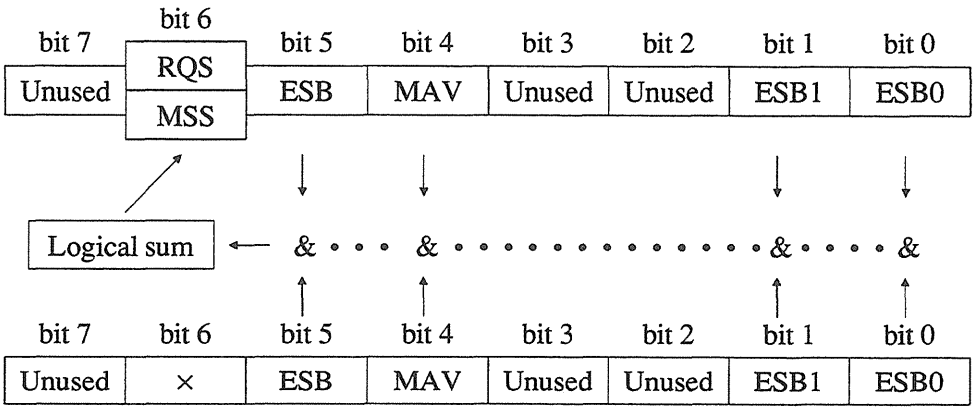
4-12 Status Byte Register

(1) Status byte register (STB)

The status byte register is an 8-bit register whose contents are output from the 3502 or 3510 to the controller, when serial polling is being performed.

If any bit in the status byte register has changed from 0 to 1 (provided that it is a bit which has been set in the service request enable register as a bit which can be used), then the MSS bit is set to 1. Simultaneously with this the RQS bit is also set to 1, and a service request is generated.

The status byte register (STB)



The service request enable register (SRER)

The RQS bit is synchronized with service requests, and is read out and simultaneously cleared when serial polling is being performed. Although the MSS bit is only read out on an "*STB?" query, on a "*CLS" command for example it is not cleared until the event is cleared.

Status byte register bit assignments

Bit 7	Unused
Bit 6 RQS	Set to 1 when a service request is dispatched
MSS	Logical sum of the other bits of the status byte register
Bit 5 ESB	Standard event summary (logical sum) bit Bitwise logical sum of the standard event status register
Bit 4 MAV	Message available Indicates that there is at least one message in the output queue
Bit 3	Unused
Bit 2	Unused
Bit 1 ESB1	Event summary bit 1 Bitwise logical sum of event status register 1
Bit 0 ESB0	Event summary bit 0 Bitwise logical sum of event status register 0

(2) Service request enable register (SRER)

This register masks the status byte register. Setting a bit of this register to 1 enables the corresponding bit of the status byte register to be used.

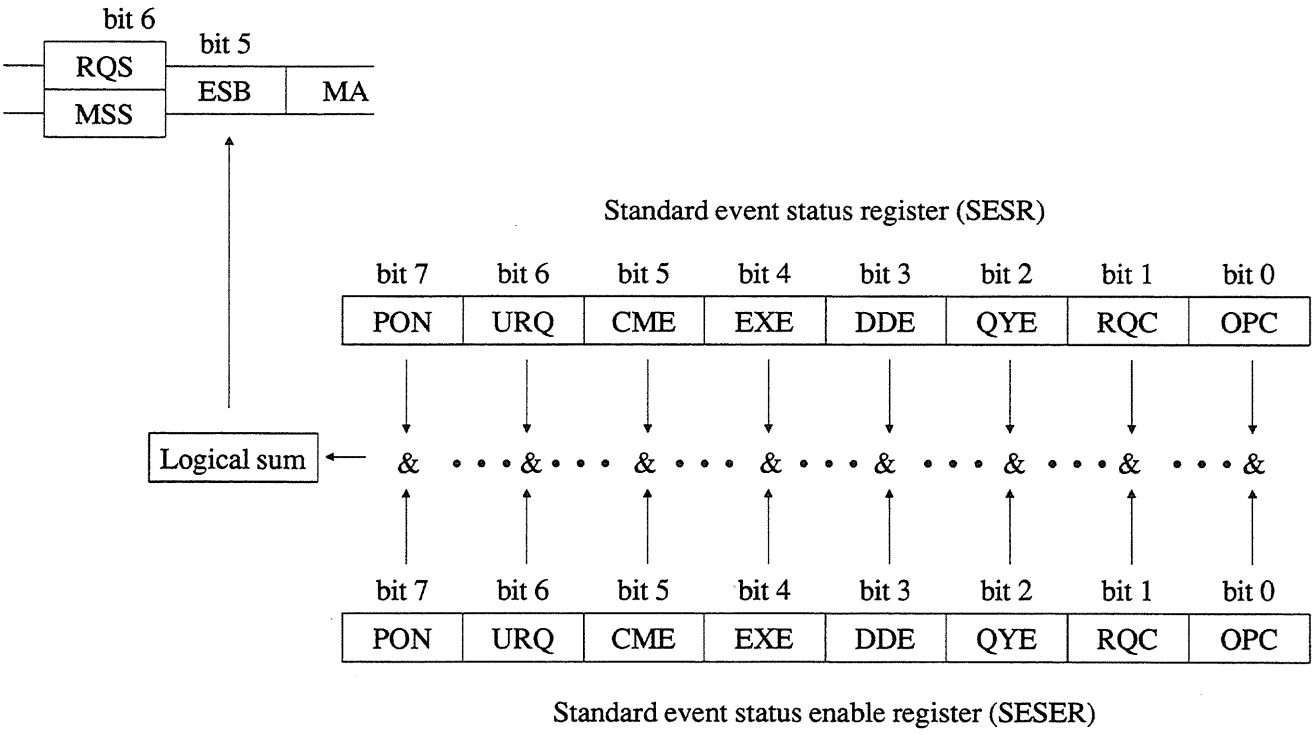
4-13 Event Registers

(1) Standard event status register (SESR)

The standard event status register is an 8-bit register. If any bit in the standard event status register is set to 1 (after masking by the standard event status enable register), bit 5 (ESB) of the status byte register is set to 1.

Status byte register (STB)

The standard event status register is cleared in the following three situations:



① When a "*CLS" command is received.

② When an "*ESR?" query is received.

③ When the unit is powered on.

(2) Standard event status enable register (SESER)

Setting any bit of the standard event status enable register to 1 enables the corresponding bit of the standard event status register to be accessed.

Standard event status register (SESR) bit assignments

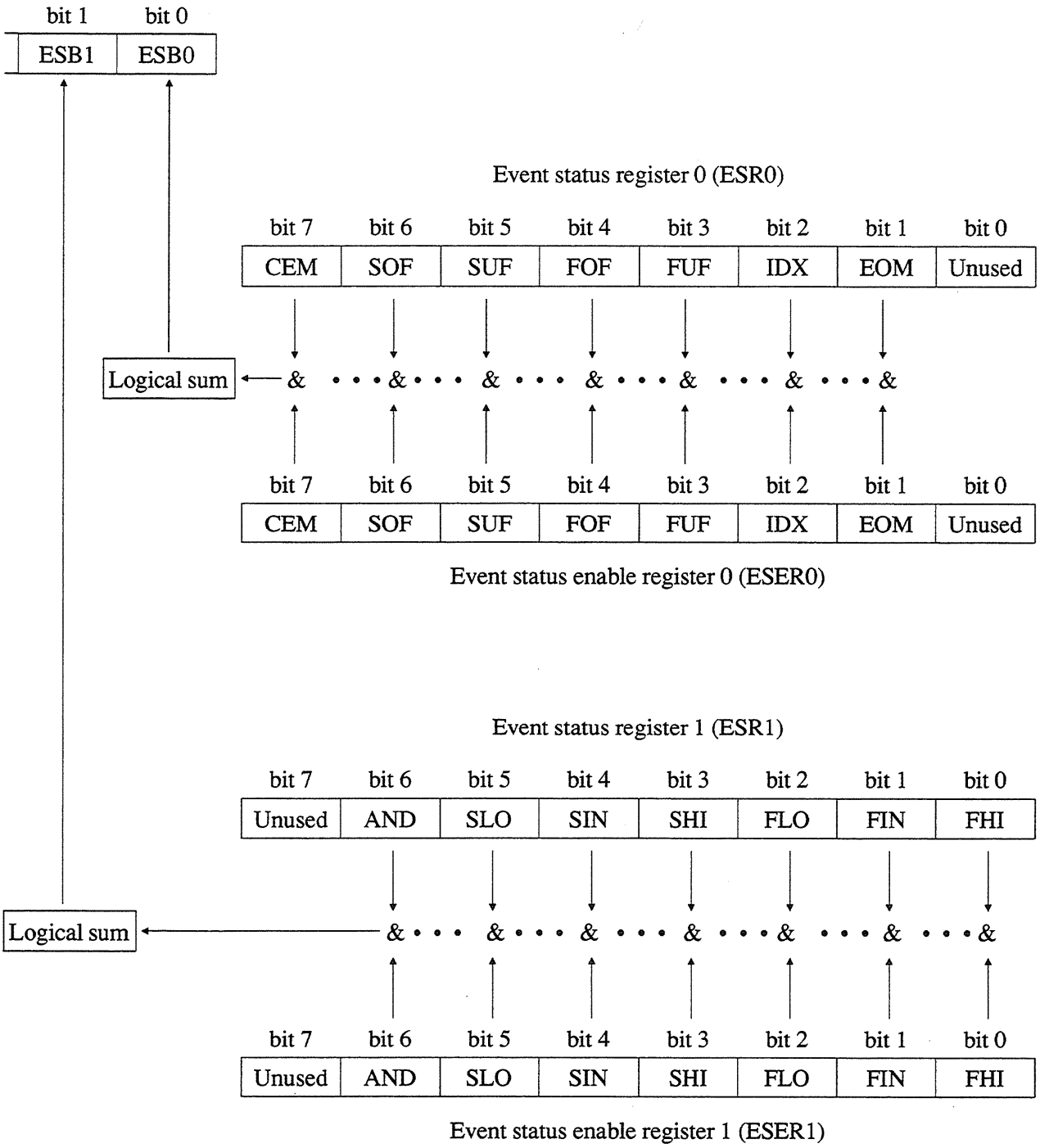
Bit 7 PON	Power on flag. When the power is turned on, or on recovery from a power cut, this bit is set to 1.
Bit 6 URQ	User request. Not used by the 3502 or 3510.
Bit 5 CME	Command error. When a command which has been received contains a syntactic or semantic error, this bit is set to 1. • The command is not supported by the 3502 or 3510. • There is a mistake in a program header. • The number of data parameters is wrong. • The format of the parameters is wrong.
Bit 4 EXE	Execution error When for some reason a command which has been received cannot be executed, this bit is set to 1. • The designated data value is outside the set range. • The designated data value is not acceptable.
Bit 3 DDE	Device dependent error. When a command cannot be executed due to some cause other than a command error, a query error, or an execution error, this bit is set to 1. • Execution is impossible due to an abnormality inside the 3502 or 3510. • Execution is impossible because some other function is being performed. • When COMParator:TYPE is set to "0" (all parameters off). • During open or short circuit compensation, valid data cannot be obtained.
Bit 2 QYE	Query error. This bit is set to 1 when a query error is detected by the output queue control. • When an attempt has been made to read the output queue when it is empty. • When the data overflows the output queue. • When data in the output queue has been lost. • When, on the same line, a query occurs after an "*IDN?" query.
Bit 1 RQC	Request for controller authority. Not used by the 3502 or 3510.
Bit 0 OPC	Operation terminated. This bit is set to 1 when an "*OPC" command is executed. • When the operation of all the messages up to the "*OPC" command has been completed.

(3) Event status registers specific to the 3502 and 3510 (ESR0 and ESR1)

Two 8-bit event status registers are provided for managing events on the 3502 or 3510. If any bit in one of these event status registers is set to 1 (after masking by the corresponding event status enable register), the following happens:

- For event status register 0, bit 0 of the status byte register (ESB0) is set to 1.
- For event status register 1, bit 1 of the status byte register (ESB1) is set to 1.

Status byte register (STB)



Event status register 0 and event status register 1 are cleared in the following three situations:

① When a "*CLS" command is received.

② When an "*ESR0?" query (for event status register 0) or "*ESR1?" query (for event status register 1) is received.

③ When the unit is powered on.

Event status register 0 (ESR0) bit assignments

Bit 7 CEM	Compensation data measurement completed
Bit 6 SOF	Second parameter overflow
Bit 5 SUF	Second parameter underflow
Bit 4 FOF	First parameter overflow
Bit 3 FUF	First parameter underflow
Bit 2 IDX	Data sampling completed
Bit 1 EOM	Measurement completed
Bit 0	Unused

Event status register 1 (ESR1) bit assignments

Bit 7	Unused
Bit 6 AND	Logical product (AND) of comparison results
Bit 5 SLO	Second parameter below lower limit
Bit 4 SIN	Second parameter within limits
Bit 3 SHI	Second parameter above upper limit
Bit 2 FLO	First parameter below lower limit
Bit 1 FIN	First parameter within limits
Bit 0 FHI	First parameter above upper limit

(4) Event status enable registers specific to the 3502 and 3510 (ESER0 and ESER1)

These event status enable registers mask the corresponding event status registers.

(5) Summary of commands for writing and reading each of the registers

Register	Read	Write
Status byte register	*STB?	———
Service request enable register	*SRE?	*SRE
Standard event status register	*ESR?	———
Standard event status enable register	*ESE?	*ESE
Event status register 0	ESR0?	———
Event status enable register 0	ESE0?	ESE0
Event status register 1	ESR1?	———
Event status enable register 1	ESE1?	ESE1

4-14 GP-IB Commands

The following commands are used for performing interface functions:

Command	Function
GTL	Go To Local The remote state is canceled, and the system goes into the local state.
LLO	Local Lock Out All keys, including the LOCAL key, become inoperable.
DCL	Device Clear Clears the input buffer and the output queue.
SDC	Selected Device Clear Clears the input buffer and the output queue.
GET	Group Execute Trigger The same as the "*TRG" standard command (refer to Section 5-14).

4-15 Talk-Only Mode

(1) Use

① A 3502 or 3510 unit, if equipped with a 9518 interface, can be connected via a GP-IB cable with a GP-IB printer having a listen-only function.
② Turn the "TALK ONLY" address switch on the 9518 outer panel to ON, set the printer to the listen-only mode, and turn on the power to both units.
③ When you require to cancel the talk-only function, turn the "TALK ONLY" address switch on the 9518 outer panel back to OFF, and turn the power off and on again.

(2) Output format

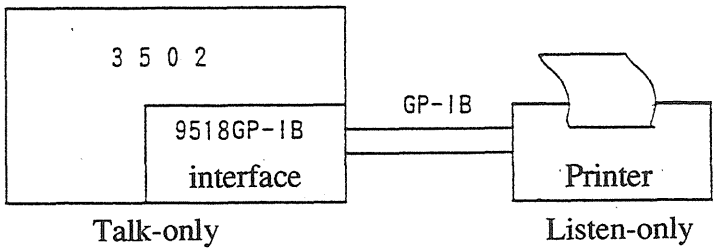
Because output is in the same format as that used for response messages to the ":MEASure?" query, refer to that response format (on page 5-65).

(3) Points for attention

① Do not connect several talk-only devices to the same bus.
② Basically only talk-only and listen-only devices should be connected to the bus. If a controller must unavoidably be connected on the same bus, be sure that its power supply is turned off.
③ Even if several listen-only devices are connected to the same bus, the bus will still operate, but the bus transmission speed may be reduced.

Example of connection (A 3510 unit may be connected in the same way)

[FIGURE]



Example of printout using the talk-only function:

```
C+10.24E-06;D+0.0012E+00
C+10.24E-06;D+0.0012E+00
C+10.24E-06;D+0.0012E+00
C+10.24E-06;D+0.0013E+00
C+10.24E-06;D+0.0012E+00
C+10.24E-06;D+0.0012E+00
C+10.24E-06;D+0.0012E+00
```

MEMO

Chapter 5

Command Reference

5-1 Format of Command Explanations

Syntax Specifies the syntax for the command.

Parameter (s) For a command that has parameters, specifies their format.

Function Specifies the function of the command.

Note Specifies points to which attention should be paid when using the command.

Response syntax Only appears for a command (query) to which a response message is returned. Specifies the syntax for the response message, both when headers are on and when headers are off.

Errors Specifies what types of error may occur. However of course all commands are susceptible to spelling mistakes.

Example These are simple examples of the use of the command. The examples all show commands in the short form.

Processing time Specifies the processing time used by the combination of the 3502 or 3510 and the 9518 to perform analysis and internal processing of the command, in its long form.

However, for commands which have data parameters, the processing time may depend on the number of parameters.
For query commands, this time is the time taken when headers are on.

NOTE

On the 3502 and the 3510, internal processing may involve a maximum delay of 60 ms from command receipt to start of analysis (70 ms when operating at 120 Hz with the 3502).
The commands which can be used on the 3502 and the 3510 are all of the sequential type.

3502/3510 This indicates a command which is available both for the 3502 Capacitance HiTester and for the 3510 Inductance HiTester.

3502 only This indicates means a command which is only available for the 3502 Capacitance HiTester.

3510 only This indicates a command which is only available for the 3510 Inductance HiTester.
If a command which is only available for one of these devices is used with the other, a command error occurs.

5-2 Standard Commands

*CLS

3502/3510

Clears the status byte register and the event registers.

Syntax *CLS

Function
- This instruction clears all the event registers associated with the bits of the status byte register (SESR, ESR0, ESR1, and the corresponding bits in STB).

Note
- This has no effect upon the output queue, the various enable registers, or bit 4 (the MAV bit) of the status byte register.

Errors
- This command is executed even in the event of a system error.

Processing time about 1.0 ms

*ESE

Sets the standard event status enable register.

Syntax *ESE␣<data>

Data <data>0 to 255 numerical data in NR1 format

Function

- This instruction sets the standard event status enable register (SESER) to a bit pattern which is used to mask the standard event status register (SESR).
- Any decimal fraction in the data value is rounded.

Standard event status enable register (SESER)

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Note

- When the power is turned on, and when a reset has taken place upon key input, the data is reinitialized to zero.

Errors

- If an attempt is made to make a setting outside the range of validity, an execution error is generated.
- If an attempt is made to set with <data> not of NRf type, a command error occurs.
- In the case of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: "*ESE␣36"

Bits 2 and 5 of SESER are set to 1.

Processing time about 3.1 ms

*ESE?

3502/3510

Reads the standard event status enable register (SESER).

Syntax *ESE?

Function
- The contents of SESER as set by the *ESE command are returned as an NR1 integral value (<data>) in the range 0 to 255.

Note
- If any error occurs, no response message to this query is produced.

Response syntax
- If headers are on, "*ESE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	"*ESE?"	"*ESE?"
Response:	"ESE␣36"	"36"

Processing time about 1.6 ms

*ESR? 3502/3510

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

Syntax *ESR?

Function

- The contents of the standard event status register SESR are returned as an numerical value (<data>) in NR1 format between 0 and 255.

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Notes

- No header is affixed to the response message.
- If any error occurs, no response message to this query is produced.
- Even in the event of a system error, this query is executed.

Response syntax

- Whether headers are on or off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.

Example

Transmission: "*ESR?"
Response: "32"

A command error (bit 5) has occurred.

Processing time about 1.6 ms

*IDN? 3502/3510

Queries manufacturer's name, model name, and software version.

Syntax *IDN?

Function
- The response consists of the name of the manufacturer of the unit, the model name, and the software version.

Notes
- The "IDN?" query is the last query message in the program messages. Accordingly, if another query is detected after this query, a query error is generated, and no response message after the "IDN?" query is produced.
- No header is affixed to the response message.
- If any error occurs, no response message to this query is produced.

Response syntax
- Whether headers are on or off, "HIOKI, 3510, 0, V01.00"

First field	:	Manufacturer's name
Second field	:	Model name (either 3502 or 3510)
Third field	:	Not used - always zero
Fourth field	:	Software version

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

Transmission:	"*IDN?"
Response:	"HIOKI, 3510, 0, V01.00"

Processing time about 1.9 ms

*OPC

3502/3510

After all action has been completed during execution, performs an SRQ request.

Syntax *OPC

Function
- When a number of commands are written on one line, the *OPC command sets bit 0 (the OPC bit) of the standard event status register (SESR) to 1 at the instant the previous commands have been completed,

Errors
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: "*RST;:MEAS?;*OPC;:HEAD␣OFF"

When the entire action of the commands *RST and MEAS? has been completed, the designated bit is set to 1.

Processing time about 1.0 ms

*OPC?

3502/3510

Queries whether or not all action has been completed during execution.

Syntax *OPC?

Function
- The same as the *OPC command, except in that, at the instant that the previous commands have been completed, instead of bit 0 (the OPC bit) of the standard event status register (SESR) being set to 1, the response message "1" is returned.

Notes
- With this query, if any error occurs, no response message is produced.
- No header is affixed to the response message.

Response syntax
- Whether headers are on or off, "1"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Processing time about 1.2 ms

*RST

Performs device initial setting.

Syntax *RST

Function

- Resets the 3502 or 3510 unit. The parameters which are reset, the values to which they are reset, and those items which are not affected by this command, are listed below.
  (1) Parameters which are reset, and their new values:
    - First parameter display: capacitance (C) display (on the 3502) or inductance (L) display (on the 3510)
    - Second parameter display: dissipation (D) display
    - Trigger mode: internal trigger
    - Test range: auto ranging
    - Averaging: off
    - Frequency display (3502) ratio: 1 kHz
    - Test signal display (3510): 1 kHz
    - Monitor display: monitor voltage
    - Beep sound: off
    - Headers: on
    - Data separator: semicolon ";"

  (2) Items which are not affected by a reset:
    - Interface functions
    - GP-IB device address
    - The output queue
    - The input buffer
    - The terminator for query messages
    - The various event registers (SESR, ESR0, and ESR1)
    - The various enable registers (SRER, SESER, ESER0, and ESER1)

Errors

- In the event of system error 6 (backup error) only, this command is executed. In the event of other system errors, this command is not executed, and a device dependent error is generated.

Processing time about 1.6 ms

*SRE

Sets the service request enable register (SRER).

Syntax *SRE␣<data>

Data <data> 0 to 255 numerical data in NR1 format

Function

- Sets the service request enable register SRER to a pattern is used to mask the status byte register (STB).
- After the decimal point, 5 and above is rounded up, while 4 and below is rounded down.

Service request enable register (SRER)

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Unused	×	ESB	MAV	Unused	Unused	ESB1	ESB0

Notes

- The unused bits (bits 2, 3, and 7) and bit 6 are disregarded.
- When the power is turned on, and when a reset has taken place upon key input, the data is reinitialized to zero.

Errors

- If an attempt is made to make a setting outside the range of validity, an execution error is generated.
- If an attempt is made to use a <data> value which is not of NRf format, a command error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example Transmission: "*SRE␣34"
Bits 1 and 5 of SRER are set to 1.

Processing time about 3.2 ms

*SRE? 3502/3510

Reads the service request enable register (SRER).

Syntax *SRE?

Function
- Returns the value of the service request enable register (SRER) as a numerical data value in NR1 format taken from the set: 0 to 3, 16 to 19, 32 to 35, and 48 to 51.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, "*SRE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	"*SRE?"	"*SRE?"
Response:	"*SRE␣34"	"34"

Processing time about 1.7 ms

*STB? 3502/3510

Queries the status byte register.

Syntax *STB?

Function

- Returns the set contents of the status byte register (STB) as a numerical data value (<data>) in NR1 format taken from the set: 0 to 3, 16 to 19, 32 to 35, 48 to 51, 64 to 67, 80 to 83, 96 to 99, and 112 to 115.

  The status byte register (STB):

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Unused	MSS	ESB	MAV	Unused	Unused	ESB1	ESB0

Notes

- Bit 6 indicates the value of the MSS bit.
- Even if service requests are cleared by serial polling, the MSS bit is not cleared.
- No header is affixed to the response message.
- With this query, if any error occurs, no response message is produced.
- Even if a system error occurs, this query is executed.

Response syntax

- Whether headers are on or off, "<data>"

Error

- If the response message is longer than 800 bytes, a query error is generated.

Example

Transmission: "*STB?"
Response: "32"

Some event has been generated in bit 5 of the standard event status register (ESB).

Processing time about 1.6 ms

*TRG

Lssues external trigger.

Syntax *TRG

Function
- In external trigger mode, performs measurement once.

Errors
- Executing this command in internal trigger mode generates an execution error.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example
Transmission: ":DISP␣LD;:TRIG␣EXT;*TRG;:MEAS?"
Response: "L␣+18.19E+00;D␣+0.0010E+00"

Processing time

	1 kHz	120 Hz (3502 only)
High speed testing	55 ms	70 ms
Testing with averaging	84 ms	200 ms

*TST? 3502/3510

Requests execution of, and queries the result of, the self test.

Syntax *TST?

Function

- Causes the 3502 or 3510 to perform the self test, and returns the result thereof as a numerical data value (<data>) in NR1 format in the range 0 through 31. The various bits of the result have the following meanings:

  bit 0 : a ROM error occurred
  bit 1 : a ROM error occurred
  bit 2 : display error occurred
  bit 3 : an I/O error occurred
  bit 4 : an interrupt error occurred
  bit 5 : unused
  bit 6 : unused
  bit 7 : unused

Notes

- No header is affixed to the response message.
- With this query, if any error occurs, no response message is produced.
- Even in the event of a system error, this query is still executed.
- A backup error (only) can be cleared with the *RST command.

Response syntax

- Whether headers are on or off, "<data>"

Error

- If the response message is longer than 800 bytes, a query error is generated.

Example

Transmission: "*TST?"
Response: "10"

A RAM error (bit 1) and an I/O error (bit 3) have occurred.

Processing time about 336 ms

*WAI

3502/3510

Waits until sampling is fully completed.

Syntax *WAI

Function
- The unit goes into waiting state until the operation of the previous command has been completed.

Notes
- The "*WAI" command is accepted by the 9518 interface because it is a standard command as specified by IEEE-488.2 1987. However, since all of the commands specific to the 3502 and 3510 are in any case sequential commands, using this "*WAI" command never has any effect.

Error
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Processing time about 1.0 ms

5-3 Commands Specific to the 3502 and/or 3510

AUTO

3502/3510

Sets automatic or manual setting of test range and equivalent circuit mode.

Syntax AUTO␣<data>

Data <data> ON/OFF character data

Function
- Switches between automatic and manual setting of test range and equivalent circuit mode.

Errors
- If <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Examples Transmission: ":AUTO␣ON"

The test range and the equivalent circuit mode are switched to automatic selection (auto-ranging).

Processing time about 2.7 ms

AUTO?

3502/3510

Queries whether automatic setting in effect.

Syntax

AUTO?

Function

- Queries whether test range and equivalent circuit mode are currently being automatically set, or not, and returns the result as "ON" or "OFF" (<data>).

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, "AUTO␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":AUTO?"	":AUTO?"
Response:	":AUTO␣ON"	"ON"

Processing time

about 1.4 ms

AVERaging

Turns averaging processing on or off.

Syntax AVERaging␣<data>

Data <data> ON/OFF character data

Function
- Starts, and stops, averaging of test data.

Notes
- If this command is used during execution of the comparator function, although the comparator setting conditions are changed temporarily, they will not be preserved.

Errors
- If <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":AVER␣ON"

Averaging processing is initiated.

Processing time about 3.0 ms

AVERaging? 3502/3510

Queries whether averaging is being performed.

Syntax AVERaging?

Function
- Queries whether averaging is currently being performed, or not, and returns the result as "ON" or "OFF" (<data>).

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":AVERAGING␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":AVER?"	":AVER?"
Response:	":AVERAGING␣ON"	"ON"

Processing time about 2.2 ms

BEEPer

3502/3510

Sets the beep sound on or off.

Syntax BEEPer␣<data>

Parameter <data> ON/OFF character data

Function
- Sets the beep sound on or off.

Errors
- If <data> is set to character data other than "ON" or "OFF", an execution error is generated.
- If <data> is other than character data, a command error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Examples Transmission: ":BEEP␣OFF"

This turns off the beep sound.

Processing time about 2.8 ms

BEEPer?

3502/3510

Queries whether the beep sound is on or off.

Syntax

BEEPer?

Function

- Returns the current on/off setting of the beep sound (as <data>).

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":BEEPER␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":BEEP?"	":BEEP?"
Response:	":BEEPER␣OFF"	"OFF"

Processing time

about 1.7 ms

BIAS

3502 only

Sets DC biasing on and off.

Syntax BIAS␣<data>

Parameter <data> ON/OFF character data

Function
- Sets DC biasing on or off.

Notes
- To use the comparator function with DC biasing applied, the DC biasing should be turned on before activating the comparator function.

Errors
- If <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Examples Transmission: ":BIAS␣ON"

DC biasing is turned on.

Processing time about 2.9 ms

BIAS? 3502 only

Queries whether DC biasing is on or off.

Syntax BIAS?

Function
- Returns whether DC biasing is on or off as character data.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":BIAS␣ON"
- If headers are off, "ON"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":BIAS?"	":BIAS?"
Response:	":BIAS␣ON"	"ON"

Processing time about 1.6 ms

COMParator

Turns the comparator function on and off.

Syntax COMParator␣<data>

Parameter <data> ON/OFF character data

Function
- Turns the comparator function on and off. (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)

Errors
- If COMParator:TYPE has been set to "0" (all parameters are set off), or if <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":COMP␣ON"

The comparator is turned on.

Processing time about 3.8 ms

COMParator? 3502/3510

Queries whether the comparator function is on or off.

Syntax COMParator?

Function
- Returns (in <data>) the current setting of the comparator function as "ON" or "OFF".

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":COMPARATOR␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP?"	":COMP?"
Response:	":COMPARATOR␣ON"	"ON"

Processing time about 2.4 ms

COMParator:AVERaging 3502/3510

Turns averaging processing for the comparator function on or off.

Syntax COMParator:AVERaging␣<data>

Parameter <data> ON/OFF character data

Function
- Turns averaging processing for the comparator function on or off. (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)

Errors
- If <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Examples Transmission: ":COMP:AVER␣OFF"

Averaging processing for the comparator function is turned off.

Processing time about 4.7 ms

COMParator:AVERaging? 3502/3510

Queries whether averaging processing for the comparator function is on or off.

Syntax COMParator:AVERaging?

Function
- Returns (in <data>) whether averaging processing for the comparator function is currently on or off.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, "COMPARATOR:AVERAGING␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:AVER?"	":COMP:AVER?"
Response:	":COMPARATOR :AVERAGING␣ON"	"ON"

Processing time about 4.0 ms

COMParator:First LIMit 3502/3510

Sets the lower and upper limit values for the first comparator parameter.

Syntax COMParator:FLIMit␣<data1>,<data2>

Parameter

<data1> is the lower limit value	(Example: 5.22)	Both are numerical
<data2> is the upper limit value	(Example: 6.11)	data in NR2 format

Function

- Sets the lower and upper limit values for the first parameter (i.e. the principal measured value) for the comparator function. (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)
- If the lower limit value and the upper limit value are set to be the same, then deviation testing is performed.
- <data> can be in NRf format, but rounding is performed for figures beyond the last valid decimal place.
- Refer to "7-6 Test Range Settings" to select valid upper and lower limit values for a given ranges setting.

Notes

- The long form and the short form are as shown below. Any variation will cause a command error.
  Long form: FLIMit Short form: FLIM
- The lower and upper limit values must be in the same range.

Errors

- If <data> is in other than NRf format, a command error occurs.
- If an attempt is made to set the lower and upper limit values straddling two or more ranges, or to set the upper limit to a value lower than the lower limit value, an execution error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Examples Transmission: ":COMP:FLIM␣0.987,1.234"

This sets the lower limit value for the first comparator parameter to 0.987 nF, and sets its upper limit value to 1.234 nF. (For the 3502 in range 5)

Transmission: ":COMP:FLIM␣0.876,0.987"

The lower limit value for the first comparator parameter is set to 0.876 mH, and its upper limit value is set to 0.987 mH. (For the 3510 in range 3)

Processing time about 6.4 ms

COMParator:First LIMit? 3502/3510

Queries the lower and upper limit values for the first comparator parameter.

Syntax COMParator:FLIMit?

Function
- Returns the current lower limit value (in <data1>) and upper limit value (in <data2>) for the first parameter for the comparator function in NR2 format.

Note
- With this query, if any error occurs, no response message is produced.
- The long form and the short form are as shown below. Any variation will cause a command error.
  Long form: FLIMit Short form: FLIM

Response syntax
- If headers are on, "COMPARATOR:FLIMIT␣<data1>,<data2>"
- If headers are off, "<data1>,<data2>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:FLIM?"	":COMP:FLIM?"
Response:	":COMPARATOR: FLIMIT␣0.876,0.987"	"0.876,0.987"

Processing time about 6.4 ms

COMParator:FREQuency

3502 only

Sets the test frequency for the comparator function.

Syntax COMParator:FREQuency␣<data>

Parameter <data> 120/1000 numerical data in NR1 format
This is the test frequency in Hertz (i.e. either 120 Hz or 1 kHz.)

Function

- Sets the test frequency for the comparator function. (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)
- <data> can be in NRf format, but rounding is performed for figures beyond the decimal point.

Errors

- If <data> has a value other than the two values described above, an execution error occurs.
- If this command is issued with <data> not in NRf format, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":COMP:FREQ␣120"

This sets the test frequency for the comparator function to 120 Hz.

Processing time about 5.4 ms

COMParator:FREQuency? 3502 only

Queries the test frequency for the comparator function.

Syntax COMParator:FREQuency?

Function
- Returns (in <data>) a numerical value in NR1 format representing the currently set test frequency (in Hertz) for the comparator function. The value of <data> is either 120 (120 Hz) or 1000 (1 kHz).

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":COMPARATOR:FREQUENCY␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:FREQ?"	":COMP:FREQ?"
Response:	":COMPARATOR: FREQUENCY␣120"	"120"

Processing time about 4.1 ms

MEMO

COMParator:MODE? 3502/3510

Queries the equivalent circuit mode for the comparator function.

Syntax COMParator:MODE?

Function
- Returns (in <data>) the currently set equivalent circuit mode for the comparator function as "SERIAL" or "PARALLEL".
- SERIAL: series equivalent circuit mode.
- PARALLEL: parallel equivalent circuit mode.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":COMPARATOR:MODE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:MODE?"	":COMP:MODE?"
Response:	":COMPARATOR :MODE␣PARALLEL"	"PARALLEL"

Processing time about 3.4 ms

COMParator:PARameter 3510 only

Sets the parameters for the comparator function.

Syntax COMParator:PARameter␣<data>

Parameter <data> LD/LQ character data
LD: inductance and dissipation (L-D display)
LQ: inductance and quality factor (L-Q display)

Function
- Sets the values to be displayed and used for the comparator function.(When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)

Notes
- The values for D (dissipation) and Q (quality factor) cannot be displayed simultaneously.

Errors
- If <data> consists of character <data> other than "LD" or "LQ", an execution error occurs.
- If <data> is other than character <data>, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":COMP:PAR␣LD"

The parameters L and D are set to be displayed and used for the comparator function.

Processing time about 4.7 ms

COMParator:PARameter? 3510 only

Queries the parameters for the comparator function.

Syntax COMParator:PARameter?

Function
- Returns (in <data>) the parameters being currently used and displayed for the comparator function.
- LD: inductance and dissipation (L-D display)
- LQ: inductance and quality factor (L-Q display)

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":COMPARATOR:PARAMETER␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:PAR?"	":COMP:PAR?"
Response:	":COMPARATOR :PARAMETER␣LD"	"LD"

Processing time about 4.1 ms

COMParator:RANGe 3502/3510

Sets the range for the comparator function.

Syntax COMParator:RANGe␣<data>

Parameter <data> can be from 1 to 22 with the 3502, or from 1 to 10 with the 3510.

- Refer to "7-6 Test Range Settings" for the relationship between the range data value and the displayed values.
- <data> can be in NRf format, but any digits after the decimal point will be rounded.

Function

- Sets the range for the comparator function.
  (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)

Notes

- When the range is set, the equivalent circuit mode is determined automatically.

Errors

- If <data> is of other than NRf format, a command error occurs.
- If <data> has a value other than those listed above, an execution error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":COMP:RANG␣5"

Set the comparator function range to range 5.

Processing time about 4.5 ms

COMParator:RANGe? 3502/3510

Queries the range for the comparator function.

Syntax COMParator:RANGe?

Function
- Returns (in <data>) the currently set range for the comparator function as a numerical data value in NR1 format, from 1 to 22 for the 3502, and from 1 to 10 for the 3510.
- Refer to "7-6 Test Range Settings" for the relationship between the range data value and the displayed values.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":COMPARATOR:RANGE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:RANG?"	":COMP:RANG?"
Response:	":COMPARATOR :RANGE␣5"	"5"

Processing time about 3.4 ms

COMParator:Secondary LIMit 3502/3510

Sets the lower and upper limit values for the second comparator parameter.

Syntax COMParator: SLIMit␣<data1><data2>

Parameter <data1> is the lower limit value (Example: 0.0005) Both are numerical data in NR2 format
<data2> is the upper limit value (Example: 0.0015) (during averaging)

Function

- Sets the lower and upper limit values for the second parameter for the comparator function. (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)
- <data> can be in NRf format, but rounding is performed for figures beyond the last valid decimal place.
- The valid ranges within which the upper and lower limit values can be set are as follows for each parameter: for high speed dissipation (D) testing, from 0.000 to 1.999; for dissipation (D) testing with averaging, from 0.0000 to 1.9999; and for quality factor (Q) testing (only on the 3510), 0.5 to 999.9.

Notes

- The long form and the short form are as shown below. Any variation will cause a command error.
  Long form: SLIMit Short form: SLIM
- Deviation testing cannot be performed for the second comparison parameter.
- For averaging, the dissipation value can be set to four decimal places. However, when manually setting the comparator parameters from the front panel, it is important to note that the dissipation value becomes invalid.

Errors If <data> is in other than NRf format, a command error occurs.
If an attempt is made to set the upper limit value to a value which is lower than the lower limit value, an execution error occurs.
In the event of a system error, this command is not executed, and a device dependent error is generated.

Example

Transmission: ":COMP:SLIM␣0.0026,0.0046"
This sets the lower limit value for the second comparator parameter (D) to 0.0026, and its upper limit value to 0.0046. (For testing with averaging).

Transmission: ":COMP:SLIM␣0.026,0.046"
This sets the lower limit value for the second comparator parameter (D - dissipation) to 0.026, and its upper limit value to 0.046. (For high speed testing).

Transmission: ":COMP:SLIM␣26.5,28.6"
This sets the lower limit value for the second comparator parameter (Q - quality factor) to 26.5, and its upper limit value to 28.6. (Only for the 3510)

Processing time about 7.8 ms

COMParator:Secondary LIMit? 3502/3510

Queries the lower and upper limit values for the second comparator parameter.

Syntax COMParator:SLIMit?

Function
- Returns the current lower limit value (in <data1>) and upper limit value (in <data2>) for the second parameter for the comparator function in NR2 format.

Note
- With this query, if any error occurs, no response message is produced.
- The long form and the short form are as shown below. Any variation will cause a command error.

  Long from: SLIMit Short from: SLIM

Response syntax
- If headers are on, ":COMPARATOR:SLIMIT␣<data1><data2>"
- If headers are off, "<data1><data2>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:SLIM?"	":COMP:SLIM?"
Response:	":COMPARATOR :SLIMIT␣0.0123,0.0077"	"0.0123,0.0077"

Processing time about 2.6 ms

COMParator:TRIGger 3502/3510

Sets the trigger mode for the comparator function.

Syntax COMParator:TRIGger_<data>

Parameter <data> INTernal/EXTernal character data

The parameter indicates internal or external trigger mode.

Function

- Sets the trigger mode for the comparator function.
  (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)

Errors

- If <data> consists of character data other than "INTernal" or "EXTernal", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":COMP:TRIG_EXT"

This sets the trigger mode for the comparator function to external triggering.

Processing time about 5.1 ms

COMParator:TRIGger? 3502/3510

Queries the trigger mode for the comparator function.

Syntax COMParator:TRIGger?

Function
- Returns (in <data>) the currently set trigger mode for the comparator function as "INTERNAL" or "EXTERNAL".

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":COMPARATOR:TRIGGER␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:TRIG?"	":COMP:TRIG?"
Response:	":COMPARATOR :TRIGGER␣EXTERNAL"	"EXTERNAL"

Processing time about 3.8 ms

COMParator:TYPE

Selects the parameters for which the comparison is performed.

Syntax COMParator:TYPE␣<data>

Parameter <data> 1, 2, or 3 numerical data in NR1 format

The value of <data> indicates which parameters the comparator function applies to, as follows:

data	First parameter comparison	Second parameter comparison
1	ON	OFF
2	OFF	ON
3	ON	ON

- <data> can be in NRf format, but any digits after the decimal point will be rounded.

Function

- Selects the parameters (i.e. display values) for which the comparison is performed.
  (When making settings for the comparator function, refer to Section 7-4, "Making Comparator Settings".)

Notes

- It is not possible to disable comparison for both parameters (e.g. by setting <data>=0).

Errors

- If <data> is of other than NRf format, a command error occurs.
- If the value of <data> is other than specified above, an execution error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":COMP:TYPE␣3"

This sets the comparator function to operate on both parameters.

Processing time about 4.0 ms

COMParator:TYPE? 3502/3510

Queries the parameters for which the comparison is performed.

Syntax COMParator:TYPE?

Function

- Returns (in <data>) a value indicating which parameters the comparator function currently applies to, as follows:

data	First parameter comparison	Second parameter comparison
0	OFF	OFF
1	ON	OFF
2	OFF	ON
3	ON	ON

Note

- Although <data>=0 may be returned as a response to this query, this setting cannot be made.
- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":COMPARATOR:TYPE␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":COMP:TYPE?"	":COMP:TYPE?"
Response:	":COMPARATOR:TYPE␣3"	"3"

Processing time about 3.2 ms

CORRection:OPEN 3502/3510

Executes or disables the open circuit compensation function.

Syntax CORRection:OPEN␣<data>

Parameter <data> ON/OFF character data

Function

- Executes or disables the open circuit compensation function.
- When the parameter "ON" is specified, the data for open circuit compensation is measured, and then the open circuit compensation is carried out.

Notes

- If valid open circuit compensation data cannot be obtained, the following steps take place:
  (1) The first parameter display on the unit displays "Err" for one second.
  (2) The open circuit compensation function is turned off.
  (3) The DDE bit (bit 3) of SESR is set to 1.
  (4) A device dependent error is generated.
- Other commands cannot be executed during measurement of the open circuit compensation data.

Errors

- If <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":CORR:OPEN␣ON"

This enables the open circuit compensation function, first measuring the data for open circuit compensation, and subsequently performing open circuit compensation.

Processing time for "ON", about 5.6 s; for "OFF", about 4.0 ms.

CORRection:OPEN? 3502/3510

Queries whether the open circuit compensation function is on or off.

Syntax CORRection:OPEN?

Function
- Returns (in <data1> as "ON" or "OFF") whether the open circuit compensation function is on or off.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":CORRECTION:OPEN␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":CORR:OPEN?"	":CORR:OPEN?"
Response:	":CORRECTION:OPEN␣ON"	"ON"

Processing time about 3.2 ms

CORRection:SHORt 3502/3510

Executes or disables the short circuit compensation function.

Syntax CORRection:SHORt␣<data>

Parameter <data> ON/OFF character data

Function
- Executes or disables the short circuit compensation function.
- When the parameter "ON" is specified, the data for short circuit compensation is measured, and then the short circuit compensation is carried out.

Notes
- If valid short circuit compensation data cannot be obtained, the following steps take place:
  (1) The short parameter display on the unit displays "Err" for one second.
  (2) The short circuit compensation function is turned off.
  (3) The DDE bit (bit 3) of SESR is set to 1.
  (4) A device dependent error is generated.
- Other commands cannot be executed during measurement of the short circuit compensation data.

Errors
- If <data> consists of character data other than "ON" or "OFF", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":CORR:SHOR␣ON"

This enables the short circuit compensation function, first measuring the data for short circuit compensation, and subsequently performing short circuit compensation.

Processing time for "ON", about 5.6 s; for "OFF", about 4.2 ms.

CORRection:SHORt? 3502/3510

Queries whether the short circuit compensation function is on or off.

Syntax CORRection:SHORt?

Function
- Returns (in <data1> as "ON" or "OFF") whether the short circuit compensation function is on or off.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":CORRECTION:SHORT␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":CORR:SHOR?"	":CORR:SHOR?"
Response:	":CORRECTION:SHORT␣ON"	"ON"

Processing time about 3.4 ms

DISPlay

3510 only

Sets the items to be displayed.

Syntax DISPlay␣<data>

Parameter <data> LD/LQ character data

LD: inductance and dissipation (L-D display)
LQ: inductance and quality factor (L-Q display)

Function

- Sets the items to be shown on the display.

Notes

- The values for D (dissipation) and Q (quality factor) cannot be displayed simultaneously.

Errors

- If <data> is set to character data other than "LD" or "LQ", an execution error is generated.
- If <data> is other than character data, a command error generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":DISP␣LD"

This sets the display to show inductance and dissipation.

Processing time about 3.0 ms

DISPlay?

3510 only

Queries the items currently being displayed.

Syntax

DISPlay?

Function

- Returns (as <data>; possible values are "LD" and "LQ") the presently set items which are being shown on the display.
- LD: inductance and dissipation (L-D display)
- LQ: inductance and quality factor (L-Q display)

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":DISPLAY␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":DISP?"	":DISP?"
Response:	":DISPLAY␣LD"	"LD"

Processing time

about 1.9 ms

DISPlay:MONitor 3502/3510

Sets the monitor parameter for display.

Syntax DISPlay:MONitor␣<data>

Parameter <data> VOLTage/CURRent/OFF character data

VOLTage: display of the monitored voltage value

CURRent: display of the monitored current value

OFF: no monitor display

Function

- Sets the test signal monitor display parameter, or stops monitor display.

Errors

- If <data> is set to character data other than "VOLTage" or "CURRent", or "OFF", an execution error is generated.
- If <data> is other than character data, a command error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":DISP:MON␣CURR"

This sets the display to show the monitored value of current.

Transmission: ":DISP:MON␣OFF"

This stops the display of monitored values.

Processing time about 4.5 ms

DISPlay:MONitor? 3502/3510

Queries the monitor parameter currently being displayed.

Syntax DISPlay:MONitor?

Function
- Returns (as <data>; possible values are "VOLTAGE", "CURRENT", and "OFF") the monitor parameter which is presently being shown on the display.
- VOLTAGE: denotes display of monitored voltage
- CURRENT: display of monitored current
- OFF: no monitor display

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":DISPLAY:MONITOR␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":DISP:MON?"	":DISP:MON?"
Response:	":DISPLAY :MONITOR␣CURRENT"	"CURRENT"

Processing time about 3.3 ms

DISPlay:SIGNal

3510 only

Sets the test signal data item for display.

Syntax DISPlay:SIGNal␣<data>

Parameter

- <data> VOLTage/FREQuency/OFF character data
- VOLTage: display of the test signal voltage value
- FREQuency: display of the test signal frequency value
- OFF: no test signal display

Function

- Sets the item of test signal data for display, or stops test signal data display.

Errors

- If <data> is set to character data other than "VOLTage", "FREQuency", or "OFF", an execution error is generated.
- If <data> is other than character data, a command error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Examples

Transmission: ":DISP:SIGN␣VOLT"

This sets the display to show the voltage value of the test signal.

Transmission: ":DISP:SIGN␣OFF"

No test signal data item is displayed.

Processing time about 4.0 ms

DISPlay:SIGNal?

3510 only

Queries the test signal data item currently being displayed.

Syntax

DISPlay:SIGNal?

Function

- Returns (as <data>; possible values are "VOLTAGE", "FREQUENCY", and "OFF") the test signal data item which is presently being shown on the display.
- VOLTAGE: display of the test signal voltage value
- FREQUENCY: display of the test signal frequency value
- OFF: no test signal data item display

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":DISPLAY:SIGNAL␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":DISP:SIGN?"	":DISP:SIGN?"
Response:	":DISPLAY :SIGNAL␣VOLTAGE"	"VOLTAGE"

Processing time

about 3.1 ms

ESE0

Sets event status enable register 0.

Syntax ESE0␣<data>

Data <data> 0 to 255 numerical data in NR1 format

Function

- Sets event status enable register 0 (ESER0) to the bitmask for controlling access to events in event status register 0 (ESR0).
- After the decimal point, 5 and above is rounded up, while 4 and below is rounded down.

Event status enable register 0 (ESER0)

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CEM	SOF	SUF	FOF	FUF	IDX	EOM	Unused

Notes

- Directly after the power has been switched on, or after a reset caused as a result of key operation, the data is initialized to 0.

Errors

- If an attempt is made to make a setting outside the range of validity, an execution error is generated.
- If <data> is other than NRf format data, a command error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":ESE0␣80"

Bit 6 and 4 of ESER0 is set to 1.

Processing time about 3.1 ms

ESE0?

3502/3510

Reads out event status enable register 0.

Syntax

ESE0?

Function

- Returns the value of event status enable register 0 in NR1 format as an even numerical value (<data>) in the range from 0 to 254.

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":ESE0␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":ESE0?"	":ESE0?"
Response:	":ESE0␣80"	"80"

Processing time

about 1.7 ms

ESE1

Sets event status enable register 1.

Syntax ESE1␣<data>

Data <data> 0 to 255 numerical data in NR1 format

Function

- Sets event status enable register 1 (ESER1) to the bitmask for controlling access to events in event status register 1 (ESR1).
- After the decimal point, 5 and above is rounded up, while 4 and below is rounded down.

Event status enable register 1 (ESER1)

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Unused	AND	SLO	SIN	SHI	FLO	FIN	FHI

Notes

- Directly after the power has been switched on, or after a reset caused as a result of key operation, the value is initialized to 0.

Errors

- If an attempt is made to make a setting outside the range of validity, an execution error is generated.
- If <data> is other than NRf format data, a command error is generated.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":ESE1␣48"

Bits 4, and 5 of ESER1 are set to 1.

Processing time about 3.2 ms

ESE1?

3502/3510

Reads out event status enable register 1.

Syntax ESE1?

Function
- Returns the value of event status enable register 1 (ESER1) in NR1 format as a numerical value (<data>) in the range from 0 to 127.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":ESE1␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":ESE1?"	":ESE1?"
Response:	":ESE1␣48"	"48"

Processing time about 1.7 ms

ESR0?

Reads out event status register 0.

Syntax ESR0?

Function

- Returns the value of event status register 0 (ESR0) in NR1 format as an even numerical value (<data>) in the range from 0 to 254, and then clears event status register 0.

Event status register 0, (ESR0)

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CEM	SOF	SUF	FOF	FUF	IDX	EOM	Unused

Note

- No header is prefixed to the response message.
- With this query, if any error occurs, no response message is produced.

Response syntax

- Whether headers are on or off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

Transmission: ":ESR0?"

Response: "2"

The measurement completed flag (bit 1) of ESR0 is raised.

Processing time about 1.7 ms

ESR1?

3502/3510

Reads out event status register 1.

Syntax ESR1?

Function

- Returns the value of event status register 1 (ESR1) in NR1 format as a numerical value (<data>) in the range from 0 to 127, and then clears event status register 1.

Event status register 1, (ESR1)

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Unused	AND	SLO	SIN	SHI	FLO	FIN	FHI

Note

- No header is prefixed to the response message.
- With this query, if any error occurs, no response message is produced.

Response syntax

- Whether headers are on or off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

Transmission: ":ESR1?"

Response: "36"

The first parameter below lower limit flag (bit 2) and the second parameter below lower limit flag (bit 5) are up.

Processing time about 1.7 ms

FREQuency

3502 only

Sets the testing frequency.

Syntax FREQuency␣<data>

Parameter <data> 120/1000 character data

This is the test frequency in Hertz (i.e. either 120 Hz or 1 kHz.)

Function

- Sets the testing frequency.
- <data> can be in NRf format, but rounding is performed for figures beyond the decimal point.

Errors

- If <data> has a value other than the two values described above, an execution error occurs.
- If an attempt is made to set with <data> not in NRf format, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":FREQ␣120"

This sets the testing frequency to 120 Hz.

Processing time about 4.2 ms

FREQuency?

3502 only

Queries the testing frequency.

Syntax

FREQuency?

Function

- Returns (in <data>) a numerical value in NR1 format representing the currently set test frequency (in Hertz).
- The value of <data> is either 120 (120 Hz) or 1000 (1 kHz).

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":FREQUENCY␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":FREQ?"	":FREQ?"
Response:	":FREQUENCY␣120"	"120"

Processing time

about 2.4 ms

HEADer

3502/3510

Turns response headers on or off.

Syntax HEADer␣<data>

Data <data> ON/OFF character data

Function
- This sets whether or not the 3502/3510 will prefix headers to its response messages.

Note
- In any case, responses to the queries *STB?, *TST?, ESR0?, and ESR1? are not prefixed with any headers.

Errors
- If <data> is set to character data other than "ON" or "OFF", an execution error is generated.
- If <data> is other than character data, a command error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example Transmission: ":HEAD␣OFF"

No headers will henceforward be prefixed to response messages.

Processing time about 3.0 ms

HEADer?

3502/3510

Queries whether or not headers on response messages are enabled.

Syntax

HEADer?

Function

- Returns (in <data>) as "ON" or "OFF" whether or not headers on response messages are enabled.

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":HEADER␣ON"
- If headers are off, "OFF"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":HEAD?"	":HEAD?"
Response:	":HEADER␣ON"	"OFF"

Processing time

about 1.9 ms

MEASure? 3502/3510

Queries measured data items.

Syntax MEASure?

Function

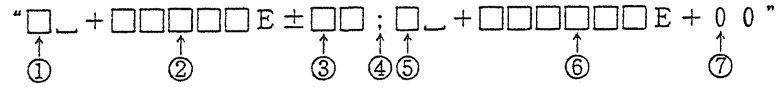
- Returns the measured values of test data items as numerical values in NR3 format.
- When the comparator function is in use, the comparison result is also returned as a numerical value in NR1 format.

Note

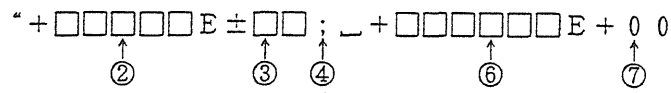
- With this query, if any error other than overflow or underflow occurs, no response message is produced.
- No test data is output until testing has been completed.
- By using the ":TRANsmit:SEParator" command, the data separator can be changed from the semicolon ";" to the comma ",".

Response syntax (1) During normal testing:

When headers are on:

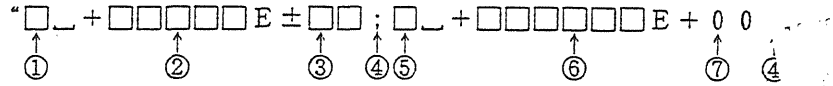


When headers are off:

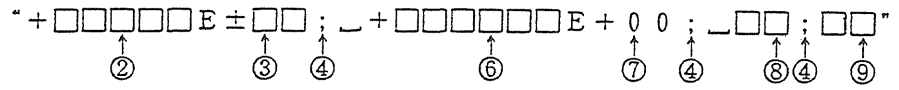


(2) When the comparator function is being used for comparison:

When headers are on:

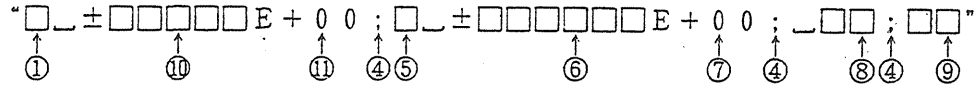


When headers are off:

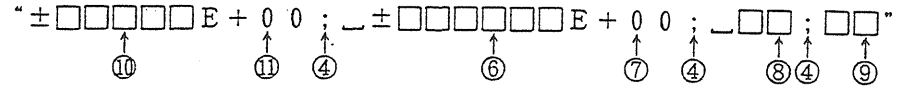


(3) When the comparator function is being used for deviation testing:

When headers are on:



When headers are off:



① First parameter header ("C" (capacitance) for the 3502; "L" (inductance) for the 3510)

② First parameter mantissa (four digits and decimal point) (position of the decimal point depends upon the set range)

③ First parameter exponent (two digits) (corresponding to the unit prefix p, n, μ, or m)

④ Semicolon (the data separator - can be changed to comma)

⑤ Second parameter header ("D" (dissipation) for the 3502; "D" (dissipation) or "Q" (quality factor) for the 3510)

⑥ Second parameter mantissa (for dissipation (D), five digits and decimal point; for quality factor (Q), four digits and decimal point)

For dissipation (D), the position of the decimal point is: □.□□□□

For quality factor (Q), the position of the decimal point is: □□□.□

When dissipation is being displayed during high speed testing, the last digit is always "0".

⑦ Second parameter exponent (two digits) (always "00")

⑧ First parameter comparator result (sign and one digit)

This represents the actual result as follows:

+0 ... In +1 ... Hi -1 ... Lo

With ":COMP:TYPE␣2", this is two spaces (refer to Example 3).

⑨ Second parameter comparator result (sign and one digit)

This represents the actual result as follows:

+0 ... In +1 ... Hi -1 ... Lo

With ":COMP:TYPE␣1", this does not appear (refer to Example 2).

⑩ First parameter deviation testing result (four digits and decimal point)

The decimal point is in the position □□□□. and the value represents the displayed value.

⑪ First parameter exponent (two digits) (always "00")

- First parameter overflow:
  All four digits of ② are "9" Example: +99.99E-03
- First parameter underflow:
  All four digits of ② are "0" Example: +00.00E-03
  For range 1, underflow cannot occur.
- Second parameter overflow:
  All five digits of ⑥ are "9" (only for dissipation D)
  Example: +9.9999E+00
- Second parameter underflow (only for the 3510):
  All four digits of ⑥ are "0" (only for quality factor Q)
  Example: +000.0E+00

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

(1) During normal testing (with the 3510, during averaging):

	When headers are on:
Transmission:	":MEAS?"
Response:	"L␣+18.19E-03;D␣+0.0011E+00"

When headers are off:

Transmission: ":MEAS?"
Response: "+18.19E-03;␣+0.0011E+00"

(2) When the comparator function is being used for comparison (:COMP:TYPE␣1):

When headers are on:

Transmission: ":MEAS?"
Response: "L␣+18.19E-03;D␣+0.0011E+00;␣+0"

When headers are off:

Transmission: ":MEAS?"
Response: "+18.19E-03;␣+0.0011E+00;␣+0"

(3) When the comparator function is being used for comparison (:COMP:TYPE␣2):

When headers are on:

Transmission: ":MEAS?"
Response: "L␣+18.19E-03;D␣+0.0011E+00;␣␣␣;-1"

When headers are off:

Transmission: ":MEAS?"
Response: "+18.19E-03;␣+0.0011E+00;␣␣␣;-1"

(4) When the comparator function is being used for comparison (:COMP:TYPE␣3):

When headers are on:

Transmission: ":MEAS?"
Response: "L␣+18.19E-03;D␣+0.0011E+00;␣+0;-1"

When headers are off:

Transmission: ":MEAS?"
Response: "+18.19E-03;␣+0.0011E+00;␣+0;-1"

(5) When the comparator function is being used for deviation testing (:COMP:TYPE␣3):

When headers are on:

Transmission: ":MEAS?"
Response: "L␣-0026.E+00;D␣+0.0011E+00;␣-1;-1"

When headers are off:

Transmission: ":MEAS?"
Response: "-0026.E+00;␣+0.0011E+00;␣-1;-1"

Processing time

- During normal testing, about 2.7 ms
  However, in the case of averaging with internal triggering, a maximum of 84 ms for 1 kHz testing and of 200 ms for 120 Hz testing (only for the 3502) is also required in order to produce the result after testing has been completed.
- When the comparator function is in use, about 3.1 ms

MEMO

MODE? 3502/3510

Queries the equivalent circuit mode.

Syntax MODE?

Function
- Returns (in <data>) the currently set equivalent circuit mode as "SERIAL" or "PARALLEL".
- SERIAL: series equivalent circuit mode
- PARALLEL: parallel equivalent circuit mode

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":MODE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":MODE?"	":MODE?"
Response:	":MODE␣PARALLEL"	"PARALLEL"

Processing time about 1.8 ms

MONitor:CURRent? 3502/3510

Queries the monitored test signal current value.

Syntax MONitor:CURRent?

Function
- Returns (in <data>) the monitored current value of the most recent test signal as a numerical data value in NR3 format.
- If the test circuit is saturated, all three digits are indicated as "0".

Note
- This query is valid even if the unit is displaying monitored voltage values, or the monitor display is off.
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":MONITOR:CURRENT␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":MON:CURR?"	":MON:CURR?"
Response:	":MONITOR: CURRENT␣+0.05E+0"	"+0.05E+0"

Processing time about 3.6 ms

MONitor:VOLTage? 3502/3510

Queries the monitored test signal voltage value.

Syntax MONitor:VOLTage?

Function
- Returns (in <data>) the monitored voltage value of the most recent test signal as a numerical data value in NR3 format.
- If the test circuit is saturated, all three digits are indicated as "0".

Note
- This query is valid even if the unit is displaying monitored voltage values, or the monitor display is off.
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":MONITOR:VOLTAGE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":MON:VOLT?"	":MON:VOLT?"
Response:	":MONITOR: VOLTAGE␣+0.78E+0"	"+0.78E+0"

Processing time about 3.5 ms

RANGe

3502/3510

Sets the test range.

Syntax RANGe␣<data>

Parameter <data> can be from 1 to 22 with the 3502, or from 1 to 10 with the 3510.

- Refer to "7-6 Test Range Settings" for the relationship between the range data value and the displayed values.
- <data> can be in NRf format, but any digits after the decimal point will be rounded.

Function

- Sets the test range.

Notes

- When the range is set, the equivalent circuit mode is determined automatically.

Errors If <data> is other than NRf format, a command error occurs.

If <data> consists of numerical data other than that specified above, an execution error occurs.

In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":RANG␣2"

Set the test range to range 2.

Processing time about 3.3 ms

:RANGe? 3502/3510

Queries the currently set range.

Syntax RANGe?

Function
- Returns (in <data>) the currently set range as a numerical data value in NR1 format, from 1 to 22 for the 3502, and from 1 to 10 for the 3510.
- Even if auto-ranging is currently enabled, this query will return the actual current range.
- Refer to "7-6 Test Range Settings" for the relationship between the range data value and the displayed values.

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":RANGE␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":RANG?"	":RANG?"
Response:	":RANGE␣1"	"1"

Processing time about 1.7 ms

TRANsmit:SEParator 3502/3510

Sets the data separator for response messages.

Syntax TRANsmit:SEParator␣<data>

Data <data> 0 or 1 through 255 numerical data in NR1 format

Function

- The data separator for response messages is set as follows:
- If <data> = 0, the separator is set to semicolon ";".
- If <data> = any value from 1 through 255, the separator is set to comma ",".
- <data> can be accepted in NRf format, but its effective value will be obtained by rounding off from the decimal point on the basis of 5 and above being rounded up and 4 and below being rounded down.

Notes

- Even if you set the data separator to the comma, it will appear as a semicolon when headers are on.

Errors

- If <data> is other than NRf format, a command error is generated.
- Even in the event of a system error, this command is executed.
- If the value of ata is outside the range 0 to 255, an execution error is generated.

Examples

Transmission: ":HOLD␣OFF ; : MEAS?"
Response: "+01.23E-03;␣+0.0045E+00"

Transmission: ":TRAN:SEP␣1;:MEAS?"
Response: "+01.23E-03,␣+0.0045E+00"

Transmission: ":HEAD␣ON;:MEAS?"
Response: "L␣+01.23E-03;D␣+0.0045E+00"

Transmission: ":HEAD␣OFF;:MEAS?"
Response: "+01.23E-03,␣+0.0045E+00"

Processing time about 4.7 ms

TRANsmit:SEParator? 3502/3510

Queries the data separator for response messages.

Syntax TRANsmit:SEParator?

Function

- The data separator for response messages is returned (in <data>) as 0 or 1.
- The returned numerical value corresponds to the setting state of the data separator as follows:
  (1) If <data> = 0, the separator is a semicolon ";".
  (2) If <data> = 1, the separator is a comma ",".

Note

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":TRANSMIT:SEPARATOR␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":TRAN:SEP?"	":TRAN:SEP?"
Response:	":TRANSMIT:SEPARATOR␣1"	"1"

Processing time about 3.6 ms

TRANsmit:TERMinator

3502/3510

Sets the data terminator for response messages.

Syntax TRANsmit:TERMinator␣<data>

Data <data> 0 or 1 through 255 numerical data in NR1 format

Function

- The data terminator for response messages is set as follows:

  (1)If <data> = 0, the terminator is set to LF. (line feed)

  (2)If <data> = 1 through 255, the terminator is set to CR (carriage return) + LF.

- <data> can be accepted in NRf format, but its effective value will be obtained by rounding off from the decimal point on the basis of 5 and above being rounded up and 4 and below being rounded down.

Notes

- Regardless of this setting, the EOI signal is always provided from 3502/3510.
- Even in the event of a system error, this command is executed.

Errors

- If <data> is of other than NRf format, a command error is generated.
- If, after rounding, the value of <data> is outside the range of validity, an execution error is generated.

Processing time about 4.8 ms

TRANsmit:TERMinator? 3502/3510

Queries the data terminator for response messages.

Syntax TRANsmit:TERMinator?

Function

- The data terminator for response messages is returned (in <data>) as 0 or 1.
- The returned numerical value corresponds to the setting state of the data terminator for response messages as follows:

  (1) If <data> = 0, the terminator is LF and EOI signal.

  (2) If <data> = 1, the terminator is CR + LF and EOI signal

Notes

- With this query, if any error occurs, no response message is produced.

Response syntax

- If headers are on, ":TRANSMIT:TERMINATOR␣<data>"
- If headers are off, "<data>"

Errors

- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Examples

	If headers are on	If headers are off
Transmission:	":TRAN:TERM?"	":TRAN:TERM?"
Response:	":TRANSMIT: TERMINATOR␣1"	"1"

Processing time about 3.8 ms

TRIGger 3502/3510

Sets the trigger mode.

Syntax TRIGger␣<data>

Parameter <data> INTernal/EXTernal character data

The parameter indicates internal or external trigger mode.

Function
- Sets the trigger mode.

Note
- If this command is used while the comparator function is being executed, although the comparator set conditions are changed, they will not be preserved.

Errors
- If <data> consists of character data other than "INTernal" or "EXTernal", an execution error occurs.
- If <data> is other than character data, a command error occurs.
- In the event of a system error, this command is not executed, and a device dependent error is generated.

Example Transmission: ":TRIG␣EXT"

This sets the trigger mode to external triggering.

Processing time about 3.6 ms

TRIGger? 3502/3510

Queries the trigger mode.

Syntax TRIGger?

Function
- Returns (in <data>) the currently set trigger mode as "INTERNAL" or "EXTERNAL".

Note
- With this query, if any error occurs, no response message is produced.

Response syntax
- If headers are on, ":TRIGGER␣<data>"
- If headers are off, "<data>"

Errors
- If the response message is longer than 800 bytes, a query error is generated.
- In the event of a system error, this query is not executed, and a device dependent error is generated.

Example

	If headers are on	If headers are off
Transmission:	":TRIG?"	":TRIG?"
Response:	":TRIGGER␣EXTERNAL"	"EXTERNAL"

Processing time about 2.2 ms

MEMO

Chapter 6

Sample Programs

The following sample programs are all written for the Hewlett-Packard HP9000 Series 300, using BASIC 4.0.

All commands in the sample programs are used in the short form, and the GP-IB device address of the 3502 or 3510 is taken as 1.

(1) Open- and short-circuit compensation

Summary

- This program carries out open- and short-circuit compensation on the 3502 or 3510.

Program List

```
10    DIM A$[40]
20    OUTPUT 701;":HEAD OFF"
30    OUTPUT 701;"*CLS"
40 Open1:    !
50    INPUT "Prepare unit for open circuit compensation, then press Enter",A$
60 Open2:    !
70    PRINT "Collecting open circuit compensation data"
80    OUTPUT 701;":CORR:OPEN ON"
90    OUTPUT 701;":ESR0?"
100   ENTER 701;A
110   IF BINAND(A,128) THEN
120                    GOTO 150
130                    ELSE
140                    GOTO 90
150                    END IF
160   OUTPUT 701:"*ESR?"
170   ENTER 701;A
180   IF BINAND(A,8) THEN
190                    PRINT "Open circuit compensation failed"
200                    GOSUB Open1
210                    ELSE
220                    GOSUB Short1
230                    END IF
240 Short1:    !
250   INPUT "Prepare unit for short circuit compensation, then press Enter",A$
260 Short2:    !
270   PRINT "Collecting short circuit compensation data"
280   OUTPUT 701;":CORR:SHOR ON"
290   OUTPUT 701;":ESR0?"
300   ENTER 701;A
310   IF BINAND(A,128) THEN
320                    GOTO 350
330                    ELSE
340                    GOTO 290
350                    END IF
```

Program Comments

10 Character array to hold messages

20 Response headers off

30 Clear registers

80 Execute

90-150 Wait until compensation ends (see p. 4-13)

160-230 Check whether compensation completed correctly (see p. 4-11)

280 Execute compensation

290-350 Wait until compensation ends (see p. 4-13)

360-430 Check whether compensation completed correctly (see p. 4-11)

```
360   OUTPUT 701;"*ESR?"
370   ENTER 701:A
380   IF BINAND(A.8) THEN
390                   PRINT "Short circuit compensation failed"
400                   GOSUB Short1
410                   ELSE
420                   GOSUB Last
430                   END IF
440 Last:   !
450   PRINT "Compensation operations completed"
460   END
```

(2) Basic settings and testing

Summary

- This program selects the test conditions for measurement on the 3502.
- It carries out a single test measurement, and displays the result on the screen.
- It also displays the monitored voltage and current values on the screen.

Program List

```
10   DIM A$[60]
20   OUTPUT 701;":TRIG EXT"
30   OUTPUT 701;":AVER ON"
40   OUTPUT 701;":FREQ 120"
50   OUTPUT 701;":AUTO ON"
60   OUTPUT 701;":BIAS ON"
70   OUTPUT 701;":DISP:MON VOLT"
80   OUTPUT 701;"*TRG;:MEAS?"
90   ENTER 701;A$
100  PRINT A$
110  OUTPUT 701;":MON:VOLT?;:MON:CURR?"
120  ENTER 701;A$
130  PRINT A$
140  END
```

Program Comments

10 Character array to hold messages

20 Select external trigger mode

30 Enable averaging

40 Test frequency 120 Hz

50 Enable auto-ranging

60 Apply bias voltage

70 Switch monitor display to show voltage

80 Send trigger, and get measurement value

90-100 Read value and display

110 Query monitored voltage and current

Note that it is possible to get values for both voltage and current regardless of the monitor display.

120-130 Read value and display

Sample output

```
C +21.43E-09;D +0.0084E+00
:MONITOR:VOLTAGE +0.98E+00;:MONITOR:CURRENT +160E-06
```

(3) Basic comparator settings

Summary

- This program makes the comparator settings for the 3502.
- This program only makes the settings. It does not carry out any testing.

Note

- Before making the comparator settings, read the notes in Section 7-4 "Making Comparator Settings."

Program List

```
10   OUTPUT 701;":COMP OFF"
20   OUTPUT 701;":COMP:TRIG INT"
30   OUTPUT 701;":COMP:AVER ON"
40   OUTPUT 701;":COMP:FREQ 120"
50   OUTPUT 701;":COMP:RANG 6"
60   OUTPUT 701;":COMP:FLIM 20.00,25.00"
70   OUTPUT 701;":COMP:SLIM 0.0000,0.0200"
80   OUTPUT 701;":COMP:TYPE 3"
90   OUTPUT 701;":BIAS ON"
100  OUTPUT 701;":COMP ON"
110  END
```

Program Comments

10 Switch off comparator function

20 Select internal trigger mode

30 Enable averaging

40 Test frequency 120 Hz

50 Select range 6

Note that setting the range automatically determines the equivalent circuit mode. This range uses a series equivalent circuit.

60 Set the lower and upper limits for the first parameter (i.e. capacitance)

Setting both limits to the same value would enable deviation testing.

70 Set the lower and upper limits for the second parameter (i.e. dissipation)

80 Apply comparison to both capacitance and dissipation

90 Apply bias voltage

100 Switch on comparator function

(4) Saving the 3502 comparator settings to a file

Summary

- This program can be used together with sample program (5).
- It saves the 3502 comparator settings to the file "FILE1".
- It also displays the settings on the screen.

Note

- Before running this program for the first time, it is first necessary to create the data file, by executing the following command:
  CREATE BDAT "FILE1",7,40
- The settings saved by this program are restored in the 3502 by sample program (5). Refer to the notes in Section 7-4 "Making Comparator Settings."
- To hold the data saved in the file requires seven character arrays, in this case A$ through G$.

Program List

```
10    DIM A$[40]
20    DIM B$[40]
30    DIM C$[40]
40    DIM D$[40]
50    DIM E$[40]
60    DIM F$[40]
70    DIM G$[40]
80    OUTPUT 701;":HEAD ON"
90    ASSIGN @Path TO "FILE1"
100   OUTPUT 701;":COMP:TRIG?"
110   ENTER 701;A$
120   OUTPUT 701;":COMP:AVER?"
130   ENTER 701;B$
140   OUTPUT 701;":COMP:FREQ?"
150   ENTER 701;C$
160   OUTPUT 701;":COMP:RANG?"
170   ENTER 701;D$
180   OUTPUT 701;":COMP:FLIM?"
190   ENTER 701;E$
200   OUTPUT 701;":COMP:SLIM?"
210   ENTER 701;F$
220   OUTPUT 701;":COMP:TYPE?"
230   ENTER 701;G$
240   OUTPUT @Path;A$
250   OUTPUT @Path;B$
```

Program Comments

10-70 Character arrays to hold messages
80 Response headers on
90 Open settings file FILE1
100-230 Get comparator settings
240-300 Save to file
310 Close file FILE1
320-400 Display settings on screen
320 PRINT "Comparator settings:"
330 PRINT A$
340 PRINT B$
350 PRINT C$
360 PRINT D$
370 PRINT E$
380 PRINT F$
390 PRINT G$
400 PRINT " ... saved to FILE1"

```
260   OUTPUT @Path;C$
270   OUTPUT @Path;D$
280   OUTPUT @Path;E$
290   OUTPUT @Path;F$
300   OUTPUT @Path;G$
310   ASSIGN @Path TO *
320   PRINT "Comparator settings:"
330   PRINT A$
340   PRINT B$
350   PRINT C$
360   PRINT D$
370   PRINT E$
380   PRINT F$
390   PRINT G$
400   PRINT " ... saved to FILE1"
410   END
```

Sample output

Comparator settings:
:COMPARATOR:TRIGGER INTERNAL
:COMPARATOR:AVERAGING ON
:COMPARATOR:FREQUENCY 120
:COMPARATOR:RANGE 3
:COMPARATOR:FLIMIT 2.000,2.500
:COMPARATOR:SLIMIT 0.0000,0.0200
:COMPARATOR:TYPE 3
... saved to FILE1

(5) Loading 3502 comparator settings from a file

Summary

- This program loads the comparator settings from the file written using sample program (4).
- It then restores these settings on the 3502.
- It also displays the settings on the screen.

Notes

- As presented here, this program reads only the file with the fixed name FILE1.

Program List

```
10   DIM A$[40]
20   DIM B$[40]
30   DIM C$[40]
40   DIM D$[40]
50   DIM E$[40]
60   DIM F$[40]
70   DIM G$[40]
80   OUTPUT 701;"*CLS"
90   ASSIGN @Path TO "FILE1"
100  ENTER @Path;A$
110  ENTER @Path;B$
120  ENTER @Path;C$
130  ENTER @Path;D$
140  ENTER @Path;E$
150  ENTER @Path;F$
160  ENTER @Path;G$
170  OUTPUT 701;":COMP OFF"
180  OUTPUT 701;A$
190  OUTPUT 701;B$
200  OUTPUT 701;C$
210  OUTPUT 701;D$
220  OUTPUT 701;E$
230  OUTPUT 701;F$
240  OUTPUT 701;G$
250  OUTPUT 701;":COMP ON"
260  ASSIGN @Path TO *
270  PRINT A$
280  PRINT B$
290  PRINT C$
300  PRINT D$
```

Program Comments

10-70 Character arrays to hold messages

80 Clear registers

90 Open settings file FILE1

100-160 Read settings from file

170 Switch off comparator function

180-240 Restore settings on 3502

250 Switch on comparator function

260 Close file FILE1

270-340 Display settings on screen

```
310  PRINT E$
320  PRINT F$
330  PRINT G$
340  PRINT " ... settings restored from FILE1"
350  END
```

Sample output

```
:COMPARATOR:TRIGGER INTERNAL
:COMPARATOR:AVERAGING ON
:COMPARATOR:FREQUENCY 120
:COMPARATOR:RANGE 3
:COMPARATOR:FLIMIT 2.000,2.500
:COMPARATOR:SLIMIT 0.0000,0.0200
:COMPARATOR:TYPE 3
     ... settings restored from FILE1
```

(6) Carrying out comparator testing

Summary

- This program first makes the comparator settings. It then prompts for the number of samples to be tested, and starts testing.
- It uses an SRQ interrupt handling routine to count the occurrences of samples outside the comparator limits (either "Hi" or "Lo").
- At the end of testing, it displays the numbers of the samples which were outside the comparator limits.

Notes

- Before deciding on the comparator settings, read the notes in Section 7-4 "Making Comparator Settings."
- The program as presented here can only guarantee to handle 100 samples. If using more, adjust the array dimensions in lines 160 to 180.

Program List

```
10    OUTPUT 701;":COMP OFF"
20    OUTPUT 701;":COMP:TRIG EXT"
30    OUTPUT 701;":COMP:AVER ON"
40    OUTPUT 701;":COMP:FREQ 120"
50    OUTPUT 701;":COMP:RANG 5"
60    OUTPUT 701;":COMP:FLIM 20.00,25.00"
70    OUTPUT 701;":COMP:TYPE 1"
80    OUTPUT 701;":HEAD ON"
90    OUTPUT 701;":ESE1 5;*SRE 2"
100   ON INTR 7 GOSUB Interrupt
110   C=1
120   H=0
130   L=0
140   INPUT "Number of samples to measure: ",X
150   OPTION BASE 1
160   DIM A$(100)[40]
170   DIM Bh(100)
180   DIM Bl(100)
190   DIM F$[40]
200   OUTPUT 701;":COMP ON"
210   OUTPUT 701;"*CLS"
220   ENABLE INTR 7;2
230   IF C>=X+1 THEN
240             GOTO 330
250             ELSE
260             GOTO 270
```

Program Comments

10 Switch off comparator function

20 Select external trigger mode

30 Enable averaging

40 Test frequency 120 Hz

50 Select range 5

60 Set the lower and upper limits for the first parameter (capacitance)

70 Apply comparison to capacitance only

80 Response headers on

90 Set SRQ generation conditions (see page 4-12)

In this case ESE1 = 5, so an SRQ interrupt is produced whenever the comparator result is "Hi" or "Lo".

100 Attach interrupt handler to SRQ interrupt

110-130 Initialize variables

150-190 Arrays for saving comparator failures (The OPTION statement specifies that arrays start from 1)

200 Switch on comparator function

210 Clear registers

220 Enable SRQ interrupt

230-320 Measurement loop

```
270             END IF
280   OUTPUT 701;"*TRG::MEAS?"
290   ENTER 701;F$
300   A$(C)=F$
310   C=C+1
320   GOTO 220
330   PRINT "Number of samples high: ";H;" - sample nos.: ";
340      FOR Ih=1 TO H
350          PRINT Bh(Ih);
360      NEXT Ih
370   PRINT ""
380   PRINT "Number of samples low: ";L;" - sample nos.: ";
390      FOR I1=1 TO L
400          PRINT B1(I1);
410      NEXT I1
420   PRINT ""
430      FOR K=1 TO X
440          PRINT "Sample ";K;": ";A$(K)
450      NEXT K
460   OUTPUT 701;":ESE1 0;*SRE 0"
470   OUTPUT 701;":COMP OFF"
480   GOTO 650
490 Interrupt:  !
500   P=SPOLL(701)
510   OUTPUT 701;":ESR1?"
520   ENTER 701;G
530   IF BINAND(G,1) THEN
540                  H=H+1
550                  Bh(H)=C
560                  GOTO 620
570                  ELSE
580                  L=L+1
590                  B1(L)=C
600                  GOTO 620
610                  END IF
620   DISABLE INTR 7
630   OUTPUT 701;"*CLS"
640   RETURN
650   END
```

330-450 Display comparator results

460 Clear enable registers

470 Switch off comparator function

500 Carry out serial polling

510 Read ESR1 (see page 4-13)

When value is high, ESR1 bit 0 is set to 1, and when value is low, ESR1 bit 2 is set to 1.

520 Get value

530-610 Determine whether low or high, increment count, and save sample number

620 Disable SRQ interrupt

630 Clear registers

Sample output

Number of samples to measure: 5 ..... (key input)
Number of samples high: 0 - sample nos.:
Number of samples low: 2 - sample nos.: 2 5
Sample 1 :C +22.96E-09;D +0.0008; +0
Sample 2 :C +15.52E-09;D +0.0009; -1
Sample 3 :C +22.85E-09;D +0.0009; +0
Sample 4 :C +23.05E-09;D +0.0008; +0
Sample 5 :C +17.21E-09;D +0.0011; -1

(7) Basic settings and testing

Summary

- This program selects the test conditions for measurement on the 3510.
- It carries out a single test measurement, and displays the result on the screen.
- It also displays the monitored voltage and current values on the screen.

Notes

- It is not possible to display dissipation (D) and quality factor (Q) values simultaneously.

Program List

```
10   DIM A$[60]
20   OUTPUT 701;":TRIG EXT"
30   OUTPUT 701;":AVER ON"
40   OUTPUT 701;":AUTO ON"
50   OUTPUT 701;":DISP LD"
60   OUTPUT 701;":DISP:MON VOLT"
70   OUTPUT 701;"*TRG;:MEAS?"
80   ENTER 701;A$
90   PRINT A$
100  OUTPUT 701;":MON:VOLT?;:MON:CURR?"
110  ENTER 701;A$
120  PRINT A$
130  END
```

Program Comments

10 Character array to hold messages

20 Select external trigger mode

30 Enable averaging

40 Enable auto-ranging

50 Display inductance and dissipation

60 Switch monitor display to show voltage

70 Send trigger, and get measurement value

80-90 Read value and display

100 Query monitored voltage and current

Note that it is possible to get values for both voltage and current regardless of the monitor display.

110-120 Read value and display

Sample output

```
L +076.5E-06;D +0.1862E+00
:MONITOR:VOLTAGE +0.03E+00;:MONITOR:CURRENT +057.E-03
```

(8) Basic comparator settings

Summary

- This program makes the comparator settings for the 3510.

Notes

- This program only makes the settings. It does not carry out any testing.
- Before making the comparator settings, read the notes in Section 7-4 "Making Comparator Settings."

Program List

```
10   OUTPUT 701;":COMP OFF"
20   OUTPUT 701;":COMP:TRIG INT"
30   OUTPUT 701;":COMP:AVER ON"
40   OUTPUT 701;":COMP:RANG 5"
50   OUTPUT 701;":COMP:FLIM 20.00,25.00"
60   OUTPUT 701;":COMP:PAR LD"
70   OUTPUT 701;":COMP:SLIM 0.0000,0.0200"
80   OUTPUT 701;":COMP:TYPE 3"
90   OUTPUT 701;":COMP ON"
100  END
```

Program Comments

10 Switch off comparator function

20 Select internal trigger mode

30 Enable averaging

40 Select range 5

> Note that setting the range automatically determines the equivalent circuit mode. This range uses a series equivalent circuit.

50 Set the lower and upper limits for the first parameter (i.e. inductance)

> Setting both limits to the same value would enable deviation testing.

60 Display inductance and dissipation

70 Set the lower and upper limits for the second parameter (i.e. dissipation)

80 Apply comparison to both inductance and dissipation

90 Switch on comparator function

(9) Saving the 3510 comparator settings to a file

Summary

- This program can be used together with sample program (10).
- It saves the 3510 comparator settings to the file "FILE1".
- It also displays the settings on the screen.

Notes

- Before running this program for the first time, it is first necessary to create the data file, by executing the following command:
  CREATE BDAT "FILE1",7,40
- The settings saved by this program are restored in the 3510 by sample program (10). Refer to the notes in Section 7-4 "Making Comparator Settings."
- To hold the data saved in the file requires seven character arrays, in this case A$ through G$.

Program List

```
10    DIM A$[40]
20    DIM B$[40]
30    DIM C$[40]
40    DIM D$[40]
50    DIM E$[40]
60    DIM F$[40]
70    DIM G$[40]
80    OUTPUT 701;":HEAD ON"
90    ASSIGN @Path TO "FILE1"
100   OUTPUT 701;":COMP:TRIG?"
110   ENTER 701;A$
120   OUTPUT 701;":COMP:AVER?"
130   ENTER 701;B$
140   OUTPUT 701;":COMP:RANG?"
150   ENTER 701;C$
160   OUTPUT 701;":COMP:FLIM?"
170   ENTER 701;D$
180   OUTPUT 701;":COMP:PAR?"
190   ENTER 701;E$
200   OUTPUT 701;":COMP:SLIM?"
210   ENTER 701;F$
220   OUTPUT 701;":COMP:TYPE?"
230   ENTER 701;G$
240   OUTPUT @Path;A$
250   OUTPUT @Path;B$
260   OUTPUT @Path;C$
```

Program Comments

10-70 Character arrays to hold messages
80 Response headers on
90 Open settings file FILE1
100-230 Get comparator settings
240-300 Save to file
310 Close file FILE1
320-400 Display settings on screen
320 PRINT "Comparator settings:"
330 PRINT A$
340 PRINT B$
350 PRINT C$
360 PRINT D$
370 PRINT E$
380 PRINT F$
390 PRINT G$
400 PRINT " ... saved to FILE1"

```
270   OUTPUT @Path;D$
280   OUTPUT @Path:E$
290   OUTPUT @Path:F$
300   OUTPUT @Path;G$
310   ASSIGN @Path TO *
320   PRINT "Comparator settings:"
330   PRINT A$
340   PRINT B$
350   PRINT C$
360   PRINT D$
370   PRINT E$
380   PRINT F$
390   PRINT G$
400   PRINT " ... saved to FILE1"
410   END
```

Sample output

Comparator settings:
:COMPARATOR:TRIGGER INTERNAL
:COMPARATOR:AVERAGING ON
:COMPARATOR:RANGE 4
:COMPARATOR:FLIMIT 11.00,14.00
:COMPARATOR:PARAMETER LD
:COMPARATOR:SLIMIT 0.0000,0.0200
:COMPARATOR:TYPE 3
... saved to FILE1

(10) Loading 3510 comparator settings from a file

Summary

- This program loads the comparator settings from the file written using sample program (9).
- It then restores these settings on the 3510.
- It also displays the settings on the screen.

Notes

- As presented here, this program reads only the file with the fixed name FILE1.

Program List

```
10   DIM A$[40]
20   DIM B$[40]
30   DIM C$[40]
40   DIM D$[40]
50   DIM E$[40]
60   DIM F$[40]
70   DIM G$[40]
80   OUTPUT 701;"*CLS"
90   ASSIGN @Path TO "FILE1"
100  ENTER @Path;A$
110  ENTER @Path;B$
120  ENTER @Path;C$
130  ENTER @Path;D$
140  ENTER @Path;E$
150  ENTER @Path;F$
160  ENTER @Path;G$
170  OUTPUT 701;":COMP OFF"
180  OUTPUT 701;A$
190  OUTPUT 701;B$
200  OUTPUT 701;C$
210  OUTPUT 701;D$
220  OUTPUT 701;E$
230  OUTPUT 701;F$
240  OUTPUT 701;G$
250  OUTPUT 701;":COMP ON"
260  ASSIGN @Path TO *
270  PRINT A$
280  PRINT B$
290  PRINT C$
300  PRINT D$
```

Program Comments

10-70 Character arrays to hold messages

80 Clear registers

90 Open settings file FILE1

100-160 Read settings from file

170 Switch off comparator function

180-240 Restore settings on 3510

250 Switch on comparator function

260 Close file FILE1

270-340 Display settings on screen

```
310  PRINT E$
320  PRINT F$
330  PRINT G$
340  PRINT " ... settings restored from FILE1"
350  END
```

Sample output

```
:COMPARATOR:TRIGGER INTERNAL
:COMPARATOR:AVERAGING ON
:COMPARATOR:RANGE 4
:COMPARATOR:FLIMIT 11.00,14.00
:COMPARATOR:PARAMETER LD
:COMPARATOR:SLIMIT 0.0000,0.0200
:COMPARATOR:TYPE 3
 ... settings restored from FILE1
```

(11) Carrying out comparator testing

Summary

- This program first makes the comparator settings. It then prompts for the number of samples to be tested, and starts testing.
- It uses an SRQ interrupt handling routine to count the occurrences of samples outside the comparator limits (either "Hi" or "Lo").
- At the end of testing, it displays the numbers of the samples which were outside the comparator limits.

Notes

- Before deciding on the comparator settings, read the notes in Section 7-4 "Making Comparator Settings."
- The program as presented here can only guarantee to handle 100 samples. If using more, adjust the array dimensions in lines 160 to 180.

Program List

```
10    OUTPUT 701;":COMP OFF"
20    OUTPUT 701;":COMP:TRIG EXT"
30    OUTPUT 701;":COMP:AVER ON"
40    OUTPUT 701;":COMP:RANG 4"
50    OUTPUT 701;":COMP:FLIM 11.00,14.00"
60    OUTPUT 701;":COMP:PAR LD"
70    OUTPUT 701;":COMP:TYPE 1"
80    OUTPUT 701;":HEAD ON"
90    OUTPUT 701;":ESE1 5;*SRE 2"
100   ON INTR 7 GOSUB Interrupt
110   C=1
120   H=0
130   L=0
140   INPUT "Number of samples to measure: ",X
150   OPTION BASE 1
160   DIM A$(100)[40]
170   DIM Bh(100)
180   DIM Bl(100)
190   DIM F$[40]
200   OUTPUT 701;":COMP ON"
210   OUTPUT 701;"*CLS"
220   ENABLE INTR 7;2
230   IF C>=X+1 THEN
240             GOTO 330
250             ELSE
260             GOTO 270
```

Program Comments

10 Switch off comparator function

20 Select external trigger mode

30 Enable averaging

40 Select range 4

50 Set the lower and upper limits for the first parameter (inductance)

60 Display inductance and dissipation

70 Apply comparison to inductance only

80 Response headers on

90 Set SRQ generation conditions (see page 4-12)

In this case ESE1 = 5, so an SRQ interrupt is produced whenever the comparator result is "Hi" or "Lo".

100 Attach interrupt handler to SRQ interrupt

110-130 Initialize variables

140 INPUT "Number of samples to measure: ",X

150-190 Arrays for saving comparator failures (The OPTION statement specifies that arrays start from 1)

200 Switch on comparator function

210 Clear registers

```
270             END IF
280   OUTPUT 701;"*TRG;:MEAS?"
290   ENTER 701;F$
300   A$(C)=F$
310   C=C+1
320   GOTO 220
330   PRINT "Number of samples high: ";H;" - sample nos.: ";
340      FOR Ih=1 TO H
350          PRINT Bh(Ih);
360      NEXT Ih
370   PRINT ""
380   PRINT "Number of samples low: ";L;" - sample nos.: ";
390      FOR I1=1 TO L
400          PRINT B1(I1);
410      NEXT I1
420   PRINT ""
430      FOR K=1 TO X
440          PRINT "Sample ";K;": ";A$(K)
450      NEXT K
460   OUTPUT 701;":ESE1 0;*SRE 0"
470   OUTPUT 701;":COMP OFF"
480   GOTO 650
490 Interrupt:  !
500   P=SPOLL(701)
510   OUTPUT 701;":ESR1?"
520   ENTER 701;G
530   IF BINAND(G,1) THEN
540                  H=H+1
550                  Bh(H)=C
560                  GOTO 620
570                  ELSE
580                  L=L+1
590                  B1(L)=C
600                  GOTO 620
610                  END IF
620   DISABLE INTR 7
630   OUTPUT 701;"*CLS"
640   RETURN
650   END
```

220 Enable SRQ interrupt

230-320 Measurement loop

330-450 Display comparator results

460 Clear enable registers

470 Switch off comparator function

500 Carry out serial polling

500 Carry out serial polling

510 Read ESR1 (see page 4-13)

When value is high, ESR1 bit 0 is set to 1, and when value is low, ESR1 bit 2 is set to 1.

520 Get value

530-610 Determine whether low or high, increment count, and save sample number

620 Disable SRQ interrupt

630 Clear registers

Sample output

Number of samples to measure: 5 ..... (key input)
Number of samples high: 0 - sample nos.:
Number of samples low: 2 - sample nos.: 2 5
Sample 1 :L +12.05E-03;D +0.0008; +0
Sample 2 :L +10.68E-03;D +0.0009; -1
Sample 3 :L +12.27E-03;D +0.0009; +0
Sample 4 :L +11.95E-03;D +0.0008; +0
Sample 5 :L +10.54E-03;D +0.0011; -1

Chapter 7

3502/3510 GP-IB Command Summary

7-1 Standard Commands

Command	Data format (number of data items)	Function	Page
*CLS		Clears STB and ESR	5-3
*ESE	NR1 numerical data (1)	Sets bitmask for ESR	5-4
*ESE?		Queries bitmask for ESR	5-5
*ESR?		Queries ESR	5-6
*IDN?		Queries device ID	5-7
*OPC		Issues service request after execution completion	5-8
*OPC?		Queries execution completion	5-9
*RST		Device initialization	5-10
*SRE	NR1 numerical data (1)	Sets bitmask for STB	5-11
*SRE?		Queries bitmask for STB	5-12
*STB?		Queries STB	5-13
*TRG		Performs sampling once	5-14
*TST?		Queries the result of the self-test	5-15
*WAI		Waits until testing is fully completed.	5-16

7-2 Commands Specific to the 3502 and/or 3510

Command	Data format (number of data items)	Function	Page
ESE0	NR1 numerical data (1)	Sets bitmask for ESR0	5-55
ESE0?		Queries bitmask for ESR0	5-56
ESE1	NR1 numerical data (1)	Sets bitmask for ESR1	5-57
ESE1?		Queries bitmask for ESR1	5-58
ESR0?		Queries ESR0	5-59
ESR1?		Queries ESR1	5-60
BEEper	ON/OFF (1)	Sets the beep sound	5-21
BEEper?		Queries the beep setting	5-22
HEADer	ON/OFF (1)	Sets headers	5-63
HEADer?		Queries headers	5-64
AUTO	ON/OFF (1)	Sets auto ranging	5-17
AUTO?		Queries auto range	5-18
RANGe	NR1 numerical data (1)	Sets current range	5-72
RANGe?		Queries current range	5-73
MODE?		Queries equivalent circuit mode	5-69
TRIGger	INTernal/EXTernal (1)	Sets trigger mode	5-78
TRIGger?		Queries trigger mode	5-79

Command	Data format (number of data items)	Explanation	Page
AVERaging	ON/OFF (1)	Sets averaging	5-19
AVERaging?		Queries averaging	5-20
FREQuency	NR1 numerical data (1)	Sets test frequency	5-61
FREQuency?		Queries test frequency	5-62
BIAS	ON/OFF (1)	Sets DC biasing	5-23
BIAS?		Queries DC biasing	5-24
DISPlay	LD/LQ (1)	Sets parameters displayed	5-49
DISPlay?		Queries parameters displayed	5-50
DISPlay: MONitor	VOLTage/CURRent/ OFF (1)	Sets monitor item displayed	5-51
DISPlay: MONitor?		Queries monitor item displayed	5-52
DISPlay: SIGNal	VOLTage/ FREQuency/OFF (1)	Sets test signal displayed	5-53
DISPlay: SIGNal?		Queries test signal displayed	5-54
CORRection: OPEN	ON/OFF (1)	Sets open circuit compensation function	5-45
CORRection: OPEN?		Queries open circuit compensation function	5-46
CORRection: SHORt	ON/OFF (1)	Sets short circuit compensation function	5-47
CORRection: SHORt?		Queries short circuit compensation function	5-48
COMParator	ON/OFF (1)	Enables comparison	5-25
COMParator?		Queries comparison enablement	5-26

Command	Data format (number of data items)	Explanation	Page
COMParator: PARAmeter	LD/LQ (1)	Sets parameters for comparison	5-35
COMParator: PARAmeter?		Queries parameters for comparison	5-36
COMParator: RANGe	NR1 numerical data (1)	Sets range for comparison	5-37
COMParator: RANGe?		Queries range for comparison	5-38
COMParator: MODE?		Queries equivalent circuit mode for comparison	5-34
COMParator: TRIGger	INTernal/EXTernal (1)	Sets trigger mode for comparison	5-41
COMParator: TRIGger?		Queries trigger mode for comparison	5-42
COMParator: AVERaging	ON/OFF (1)	Sets averaging for comparison	5-27
COMParator: AVERaging?		Queries averaging for comparison	5-28
COMParator: FREQuency	NR1 numerical data (1)	Sets frequency for comparison	5-31
COMParator: FREQuency?		Queries frequency for comparison	5-32
COMParator: TYPE	NR1 numerical data (1)	Selects parameters subject to comparison	5-43
COMParator: TYPE?		Queries parameters subject to comparison	5-44
COMParator: First LIMit	NR2 numerical data (2)	Sets limit values for first parameter	5-29
COMParator: First LIMit?		Queries limit values for first parameter	5-30
COMParator: Secondary LIMit	NR2 numerical data (2)	Sets limit values for second parameter	5-39
COMParator: Secondary LIMit?		Queries limit values for second parameter	5-40

TRANsmit: SEParator	NR1 numerical data (1)	Sets the separator	5-74
TRANsmit: SEParator?		Queries the separator	5-75
TRANsmit: TERMinator	NR1 numerical data (1)	Sets the terminator	5-76
TRANsmit: TERMinator?		Queries the terminator	5-77
MEASure?	NR3 numerical data (2) * NR1 numerical data (1 or 2)	Gets measurement data * Comparator result	5-65
MONitor: CURRent?	NR3 numerical data (1)	Queries monitored current	5-70
MONitor: VOLTage?	NR3 numerical data (1)	Queries monitored voltage	5-71

7-3 Initialization

The following table shows which items are initialized and which not, under various conditions.

Item	Power on	*RST command	Device clear	*CLS command
GP-IB device address	No	No	No	No
Device specific functions (ranges etc.)	No	Yes	No	No
Output queue	Yes	No	Yes	No
Input buffer	Yes	No	Yes	No
Status byte register	Yes	No	No *1	Yes *2
Event registers	Yes *3	No	No	Yes
Enable registers	Yes	No	No	No
Current path	Yes	No	Yes	No
Headers on/off	Yes	Yes	No	No
Terminator for response messages	Yes	No	No	No
Separator for response messages	Yes	Yes	No	No

*1 Only the MAV bit (bit 4) is cleared.

*2 All bits except the MAV bit are cleared.

*3 Except the PON bit (bit 7).

7-4 Making Comparator Settings

Often errors occur due to making comparator settings improperly or in the wrong order. In order to forestall this, keep to the following specified order when constructing your program.

Further, when making comparator settings, either check first to make sure that the comparator is turned off, or be sure first to turn it off by using the ":COMP" command.

(1) Example procedure for setting the comparator on the 3502

```
① ":COMP:TRIG INT"
② ":COMP:AVER ON"
③ ":COMP:FREQ 120"
④ ":COMP:RANG 5"
⑤ ":COMP:FLIM 10.00, 11.00"
⑥ ":COMP:SLIM 0.0012, 0.0015"
⑦ ":COMP:TYPE 3"
```

<Points for attention>

- Step ①:
  Because the ":COMP:TRIG" command exerts no influence on the other setting conditions, it can be located anywhere in this sequence.

- Steps ② and ⑥:
  When the upper and lower limit values for D (dissipation) are to be set using 4 1/2 digit accuracy, be sure to execute the ":COMP:SLIM" command to set the second parameter upper and lower limit values only after turning averaging on with the ":COMP:AVER" command. If the second parameter upper and lower limit values are set with averaging off, then the fifth decimal places of the values set will be rounded up or down from the values input. For example, if Step ⑥ were executed with averaging still turned off, then the second parameter upper limit value would be set to 0.002 and its lower limit value would be set to 0.001.

- Steps ③, ④, and ⑤:
  The range within which the upper and lower limit values for capacitance (C) can be set differs depending upon the test frequency and the test range. Accordingly, be sure to execute the ":COMP:FLIM" command to set the first parameter upper and lower limit values only after setting the test frequency and the test range.

- Step ⑦:
  Use the ":COMP:TYPE" command to designate the test parameters for comparison only after setting all the comparison conditions.

(2) Example procedure for setting the comparator on the 3510

```
① ":COMP:TRIG INT"
② ":COMP:AVER ON"
③ ":COMP:RANG 5"
④ ":COMP:FLIM 100.0,110.0"
⑤ ":COMP:PAR LD"
⑥ ":COMP:SLIM 0.0012,0.0015"
⑦ ":COMP:PAR LQ"
⑧ ":COMP:SLIM 650.0,850.0"
⑨ ":COMP:TYPE 3"
```

<Points for attention:>

- Step ①:
  Because the ":COMP:TRIG" command exerts no influence on the other setting conditions, it can be located anywhere in this sequence.
- Steps ② and ⑥:
  When the upper and lower limit values for D (dissipation) are to be set using 4 1/2 digit accuracy, be sure to execute the ":COMP:SLIM" command to set the second parameter upper and lower limit values only after turning averaging on with the ":COMP:AVER" command. If the second parameter upper and lower limit values are set with averaging off, then the fifth decimal places of the values set will be rounded up or down from the values input. For example, if Step ⑥ were executed with averaging still turned off, then the second parameter upper limit value would be set to 0.002 and its lower limit value would be set to 0.001.
- Steps ③ and ④:
  The range within which the upper and lower limit values for inductance (L) can be set differs depending upon the test range. Accordingly, be sure to execute the ":COMP:FLIM" command to set the first parameter upper and lower limit values only after setting the test range.
- Steps ⑤ through ⑧:
  The upper and lower limit values for dissipation (D) and quality factor (Q) can be set individually, as shown in the above example. Of course, it is also possible to perform only the settings for dissipation (D) by performing only Steps ⑤ and ⑥ above, or to perform only the settings for quality factor (Q) by performing only Steps ⑦ and ⑧ above. Further, if you wish to switch over the second parameter after setting all these comparison conditions, then, after having selected dissipation (D) or quality factor (Q) as the second parameter for comparison by using the ":COMP:PAR" command, set the parameters for comparison using the ":COMP:TYPE" command.
- Step ⑦:
  Use the ":COMP:TYPE" command to designate the test parameters for comparison only after setting all the comparison conditions.

7-5 Commands Which Cannot be Used During Comparison

Commands Specific to the 3502 and/or 3510
AUTO
BIAS
COMParator:AVERaging
COMParator:First LIMit
COMParator:FREQuency
COMParator:PARameter
COMParator:RANGe
COMParator:Secondary LIMit
COMParator:TRIG
COMParator:TYPE
CORRection:OPEN
CORRection:SHORt
DISPlay
DISPlay:MONitor
DISPlay:MONitor?
DISPlay:SIGNal
DISPlay:SIGNal?
FREQuency
RANGe

* All standard commands can be used.
* If these commands are used while the comparator function is enabled, an execution error occurs.

7-6 Test Range Settings

3502 Test range settings (120 Hz)

Serial equivalent circuit mode		
Range number	Display value range (for a fixed range)	Output resistance
12	0.000μF ~ 4.000μF	10 kΩ
13	2.000μF ~ 9.900μF	10 kΩ
14	0.00μF ~ 40.00μF	1 kΩ
15	20.00μF ~ 99.00μF	1 kΩ
16	0.0μF ~ 400.0μF	100 Ω
17	200.0μF ~ 990.0μF	100 Ω
18	0.000mF ~ 4.000mF	10 Ω
19	2.000mF ~ 9.900mF	10 Ω
20	1.00mF ~ 40.00mF	10 Ω
21	20.00mF ~ 99.00mF	10 Ω
22	10.0mF ~ 400.0 mF	10 Ω

Parallel equivalent circuit mode		
Range number	Display value range (for a fixed range)	Output resistance
1	0.0pF ~ 400.0pF	100 kΩ
2	200.0pF ~ 990.0pF	100 kΩ
3	0.000nF ~ 4.000nF	100 kΩ
4	2.000nF ~ 9.900nF	10 kΩ
5	0.00nF ~ 40.00nF	10 kΩ
6	20.00nF ~ 99.00nF	1 kΩ
7	0.0nF ~ 400.0nF	1 kΩ
8	200.0nF ~ 990.0nF	100 Ω
9	0.000μF ~ 4.000μF	100 Ω
10	2.000μF ~ 9.900μF	10 Ω
11	0.00μF ~ 40.00μF	10 Ω

3502 Test Range Settings (1 kHz)

Parallel equivalent circuit mode		
Range number	Display value range (for a fixed range)	Output resistance
1	0.00pF ~ 400.0pF	100 kΩ
2	20.00pF ~ 99.00pF	100 kΩ
3	0.0pF ~ 400.0pF	100 kΩ
4	200.0pF ~ 990.0pF	10 kΩ
5	0.000nF ~ 4.000nF	10 kΩ
6	2.000nF ~ 9.900nF	1 kΩ
7	0.00nF ~ 40.00nF	1 kΩ
8	20.00nF ~ 99.00nF	100 Ω
9	0.0nF ~ 400.0nF	100 Ω
10	200.0nF ~ 990.0F	10 Ω
11	0.000μF ~ 4.000μF	10 Ω

Serial equivalent circuit mode		
Range number	Display value range (for a fixed range)	Output resistance
12	0.0nF ~ 400.0nF	10 kΩ
13	200.0nF ~ 990.0nF	10 kΩ
14	0.000μF ~ 4.000μF	1 kΩ
15	2.000μF ~ 9.900μF	1 kΩ
16	0.00μF ~ 40.00μF	100 Ω
17	20.00μF ~ 99.00μF	100 Ω
18	0.0μF ~ 400.0μF	10 Ω
19	200.0μF ~ 990.0μF	10 Ω
20	0.100mF ~ 4.000mF	10 Ω
21	2.000mF ~ 9.900mF	10 Ω
22	1.00mF ~ 40.00 mF	10 Ω

3510 Test Range Settings

Serial equivalent circuit mode		
Range number	Display value range (for a fixed range)	Output resistance
1	0.00μH ~ 20.00μH	10 Ω
2	10.0μH ~ 200.0μH	10 Ω
3	0.100mH ~ 2.000mH	100 Ω
4	1.00mH ~ 20.00mH	1k Ω
5	10.0mH ~ 200.0mH	10k Ω
6	0.100 H ~ 2.000 H	100 Ω

Parallel equivalent circuit mode		
Range number	Display value range (for a fixed range)	Output resistance
7	10.0mH ~ 200.0mH	10 Ω
8	0.100mH ~ 2.000mH	100 Ω
9	1.00 H ~ 20.00 H	1k Ω
10	10.0 H ~ 200.0 H	10k Ω

MEMO

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 2-3, "Specifications".

(2) Operation with a device address other than 0 through 30

- The bus is disabled.

(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, and all other items are preserved. However, the header on/off setting, and response message separator and terminator are all reinitialized.

(5) List of message exchange options

(a) Input buffer capacity and operation:

- These are detailed in Section 4-10, "Input Buffer".

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

- ":COMParator:First LIMit?" query ... two
- ":COMParator:Secondary LIMit?" query ... two
- ":MEASure?" query ... from two to four

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

- On the 3502 or 3510, 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, and decimal program data.
- Compound commands and program headers can be used.

(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 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 7, "3502/3510 GP-IB Command 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 Section 5-2, "Standard Commands".

(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 3502 or 3510 to its initial state. (Refer to Section 5-2, "Standard Commands", and Section 7-3, "Initialization").

(20) Scope of the self-testing executed as a result of the "*TST?" query

- This is detailed in Section 5-2, "Standard Commands".

(21) Additional organization of the status data used in a device status report

- This is detailed in Section 4-13, "Event Registers".

(22) Whether commands are overlap or sequential type

- All the commands are sequential commands.

(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. The ":MEASure?" query, when and only when there is no effective data, terminates when effective data has been produced. The "*TRG" command terminates the moment that measured data has been obtained.
- The ":CORRection:OPEN" command and ":CORRection:SHORt" command terminate when open circuit or short circuit compensation has completed on the 3502 or 3510.

Chapter 9

Troubleshooting

If the GP-IB appears to be malfunctioning, refer to the information below before calling for servicing. The following troubleshooting information should be consulted with particular care if problems are being experienced while using an NEC PC-9801 series personal computer as the controller, because there are a number of important points to which attention should be paid.

Symptom	Cause / Treatment
The GP-IB has stopped working completely.	• Are the cables properly connected? • Is the device address for the 3502 or 3510 set correctly? • Does some other device have the same device address? • Are all the devices powered on?
Transmission on the GP-IB is not taking place properly (PC9801).	• Is the controller message terminator set correctly? (Refer to Section 4-5, "Message Terminators")
After transmission on the GP-IB bus, the keys on the 3502 or 3510 freeze up and have no effect.	• Press the LOCK (LOCAL) button on the front panel of the 3502 or 3510 to release the remote state. • Has a LLO command (Local Lock-Out) been transmitted? Transmit a GTL command to put the 3502 or 3510 into the local state.
When attempting to read data using a Basic INPUT@ statement, the GP-IB bus hangs.	• Be sure to transmit one query before each INPUT@ statement. • Have any of these transmitted queries resulted in an error?
Although a command has been transmitted, nothing has happened.	• Using the "*ESR?" command, inspect the standard event status register, and check what type of error has occurred.
The amount of data read in is insufficient (PC9801).	• If the data includes one or more commas, then try using a LINE INPUT statement.
Sending several queries, produces only one response.	• Has an error occurred? • Send the queries one at a time, and read the responses individually. When you want to read them in all at once, try doing so by putting them all on one line separated by the message separator character. • Have you used the "*IDN?" query?
Although a response has been read in, the data does not appear (PC9801).	• Have the response messages from the 3502 or 3510 exceeded the 255-byte buffer capacity of the PC-9801? Try dividing up their reading in, by increasing the number of variables of the INPUT statement.
Sometimes service requests are not generated.	• Have the service request enable register and the various event status enable registers been correctly set? • Clear all the event registers at the end of SRQ processing subroutines by using the "*CLS" command. If an event bit is not cleared, no service request will be generated for that event.

Symptom	Cause / Treatment
Service requests function peculiarly (PC9801).	• If you are using N88 BASIC, add to the SRQ processing subroutine an instruction setting the PC9801 SRQ flag off. • If you are using N88 BASIC under MS-DOS, change the portion that sets the SRQ flag as follows: DEF␣SEG=&H60 ⇓ DEF␣SEG=SEGPTR(7)
The response message to a query differs from the display on the front panel of the 3502 or 3510.	• Due to the response message being produced at the instant that the 3502 or 3510 receives the query, there is a possibility that it may not agree with the display at the instant that the controller reads it in.

INDEX

N

O

P

Q

R

S

Sep.	1992	edition 1
Aug.	1993	Revised edition 1
Oct.	1995	Revised edition 2

HIOKI E.E. CORPORATION Printed in Japan

HIOKI

HIOKI E.E. CORPORATION

81 Koizumi, Ueda, Nagano 386-11, Japan
TEL: 0268-28-0562 FAX: 0268-28-0568

9518A981-02 95-10-0001H