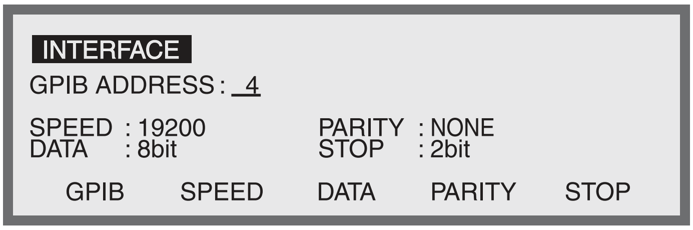
Part No. Z1-002-422, IB002894
Apr. 2004

OPERATION MANUAL

WITHSTANDING VOLTAGE/
INSULATION RESISTANCE TESTER
TOS9200 Series

TOS9200
TOS9201

GPIB/RS-232C INTERFACE



INTERFACE
GPIB ADDRESS : 4

```
SPEED  : 19200          PARITY  : NONE
DATA   : 8bit           STOP    : 2bit
```

⚠ DANGER

This Tester generates high voltage.

○ Any incorrect handling may cause death.
○ Read Chapter 2 “PRECAUTIONS ON HANDLING” in TOS9200/9201 manual to prevent accident.
○ Keep this manual together with the manual for TOS9200/9201, near the tester for easy accsse of the operator.

KIKUSUI

Use of Operation Manual

Please read through and understand this Operation Manual before operating the product. After reading, always keep the manual nearby so that you may refer to it as needed. When moving the product to another location, be sure to bring the manual as well.

If you find any incorrectly arranged or missing pages in this manual, they will be replaced. If the manual it gets lost or soiled, a new copy can be provided for a fee. In either case, please contact Kikusui distributor/ agent, and provide the “Kikusui Part No.” given on the cover.

This manual has been prepared with the utmost care; however, if you have any questions, or note any errors or omissions, please contact Kikusui distributor/agent.

Both unit specifications and manual contents are subject to change without notice.

Printed in Japan.

⚠ Safety Symbols

For the safe use and safe maintenance of this product, the following symbols are used throughout this manual and on the product. Understand the meanings of the symbols and observe the instructions they indicate (the choice of symbols used depends on the products).

ϟ OR ⚠	Indicates that a high voltage (over 1000 V) is used here. Touching the part causes a possibly fatal electric shock. If physical contact is required by your work, start work only after you make sure that no voltage is output here.
DANGER	Indicates an imminently hazardous situation which, if ignored, will result in death or serious injury.
⚠WARNING	Indicates a potentially hazardous situation which, if ignored, could result in death or serious injury.
⚠CAUTION	Indicates a potentially hazardous situation which, if ignored, may result in damage to the product and other property.
⊘	Shows that the act indicated is prohibited.
⚠	Is placed before the sign “DANGER,” “WARNING,” or “CAUTION” to emphasize these. When this symbol is marked on the product, see the relevant sections in this manual.
⏚	Indicates a protective conductor terminal.
⊥	Indicates a chassis(frame) terminal.

Description of Contents

This manual is composed of the following chapters.

Chapter 1 TOS9200 GPIB/RS-232CSetup

This chapter describes the preparations to be made for remote control using the GPIB/RS-232C interface.

Chapter 2 GPIB and RS-232C

This chapter describes the GPIB/RS-232C interface and device messages.

Chapter 3 Messages and Registers

This chapter describes the device messages and registers.

Appendix

This section provides some examples of programs to be created and used with the GPIB/RS-232C interface.

Contents

Chapter 1 TOS9200 GPIB/RS-232CSetup

Chapter 2 GPIB and RS-232C

Chapter 3 Messages and Registers

Appendix

Chapter 1 TOS9200 GPIB/RS-232CSetup

This is the operation manual designed to remotely control the TOS9200 Series withstanding voltage/insulation resistance tester using the GPIB or RS-232C interface. Attention is focused on the device message used in remote control.

Before starting remote control, prepare the Operation Manual for the TOS9200 Series and gain a full understanding of the instructions given in the manual.

⚠ WARNING

- In remote control, an external signal is used to turn a high voltage on/off, resulting in serious potential danger. To avoid the accidental generation of a high voltage and prevent workers from touching the device under testing (hereinafter referred to as a "DUT"), the high-voltage test leadwire, the high-voltage probe, and the output terminals when a high voltage is being output, provide full safety measures. Never conduct remote control without taking proper safety measures.

NOTE

- When an abnormal voltage occurs on the power line while a remote control operation using GPIB or RS-232C interface, the tester returns PROTECTION (line voltage monitoring) message. However, when the power of the tester is turned on while an abnormal voltage occurs already on the power line, the tester cannot return a message.

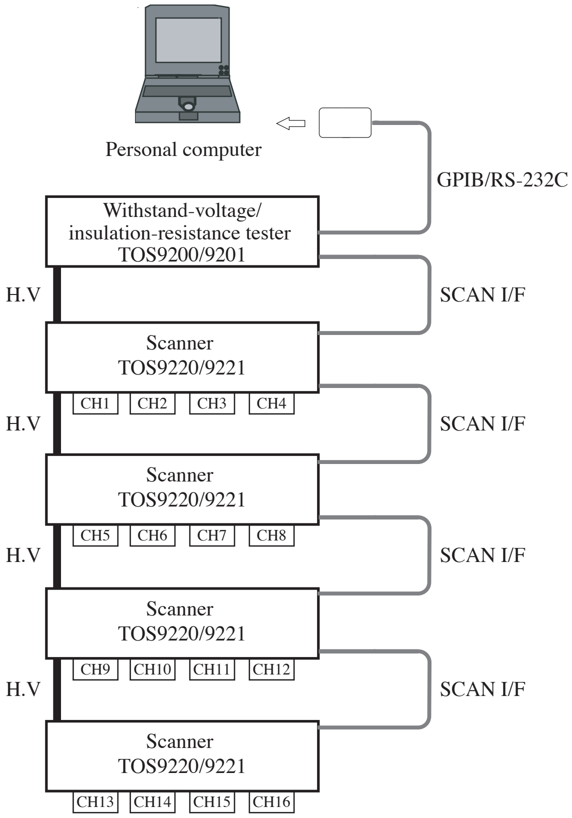


Fig.1-1 System configuration (example)

1.1 Preparing for GPIB Control

NOTE • If settings are made so that the tester controls the earth continuity tester TOS6200, the GPIB interface cannot be used for remote control. For details, see “Controlling the TOS6200” in the Operation Manual for the Tester.

1.1.1 Connecting the GPIB cable

Turn off all POWER switches on the TOS9200 and other devices that are part of the GPIB system.

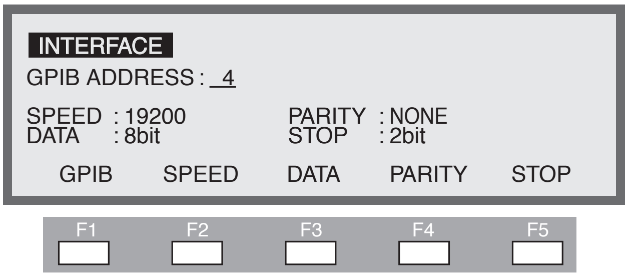
Connect the GPIB cable to the GPIB connector on the rear panel of the TOS9200/TOS9201.

1.1.2 Setting the GPIB address

To conduct remote control using the GPIB interface, a GPIB address must be set on the TOS9200/TOS9201. Do not use an address already set in the system.

To set a GPIB address, use the Interface Settings screen.

To display the Interface Settings screen (INTERFACE), press the SHIFT + SYSTEM(I/F) keys. The SYSTEM(I/F) key then lights up.



```
INTERFACE
GPIB ADDRESS : 4

SPEED  : 19200          PARITY  : NONE
DATA   : 8bit           STOP    : 2bit
```

Set the GPIB address on the TOS9200/TOS9201 to 0 through 30.

NOTE • The set GPIB address becomes valid after the TOS9200/9201 is restarted.

1. Using the F1 key (GPIB) or the ▲ ▼ ◀ ▶ keys, move the cursor to the GPIB ADDRESS.
2. Using the rotary knob, set a GPIB address.
3. On the TOS9200/TOS9201, turn off the POWER switch and then turn it on again.

1.2 Preparing for RS-232C Control

1.2.1 Connecting the RS-232C cable

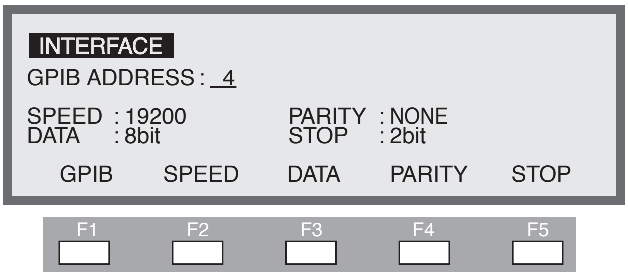
Turn off all POWER switches on the TOS9200 and other devices that are part of the RS-232C system.

Connect the RS-232C cable to the RS-232C connector on the rear panel of the TOS9200/TOS9201.

1.2.2 RS-232C settings

To make settings, use the Interface Settings screen.

To display the Interface Settings screen (INTERFACE), press the SHIFT + SYSTEM(I/F) keys. The SYSTEM(I/F) key will then light up.



```
INTERFACE
GPIB ADDRESS : 4

SPEED  : 19200          PARITY  : NONE
DATA   : 8bit           STOP    : 2bit

  GPIB     SPEED     DATA     PARITY     STOP
```

NOTE

- When the interface settings are changed, they become valid after the TOS9200/ 9201 is restarted.

Setting the communication speed of the RS-232C interface

Three communication speeds are available for the RS-232C interface:

38,400 bps
19,200 bps
9,600 bps

1. Using the F2 key (SPEED) or the ▲ ▼ ◀ ▶ keys, move the cursor to SPEED.
2. Using the rotary knob, select 38,400, 19,200, or 9,600 bps.

Setting the data length of the RS-232C interface

The following two data lengths are available for the RS-232C interface:

7 bits
8 bits

1. Using the F3 key (DATA) or the ▲ ▼ ◀ ▶ keys, move the cursor to DATA.
2. Using the rotary knob, select 7 or 8.

Setting the parity of the RS-232C interface

The following three parities are available for the RS-232C interface:

NONE
ODD
EVEN

1. Using the F4 key (PARITY) or the ▲ ▼ ◀ ▶ keys, move the cursor to PARITY.
2. Using the rotary knob, select NONE, ODD, or EVEN.

Setting the stop bit of the RS-232C interface

The following two stop bits are available for the RS-232C interface:

1 bit
2 bits

1. Using the F5 key (STOP) or the ▲ ▼ ◀ ▶ keys, move the cursor to STOP.
2. Using the rotary knob, select 1 or 2.

Chapter 2 GPIB and RS-232C

This chapter describes the GPIB/RS-232C interface and device messages.

2.1 Interface

The tester is provided with a GPIB and RS-232C interface. Either can be selected.

2.1.1 GPIB interface

List of GPIB functions

Function	Subset	Description
Source handshaking	SH1	All functions provided
Acceptor handshaking	AH1	All functions provided
Talker	T6	All functions provided except for the talk-only function
Expansion talker	TE0	No function
Listener	L4	All functions provided except for the listen-only function
Expansion listener	LE0	No function
Service request	SR1	All functions provided
Remote local	RL1	All functions provided
Parallel polling	PP0	No function
Device clear	DC1	All functions provided
Device trigger	DT0	No function
Controller	C0	No function
Electrical interface	E1	Open collector

GPIB cables are available from Kikusui. Contact the supplier or your Kikusui agent.

GPIB cable: 1 m (Product No. 92080)

GPIB cable: 2 m (Product No. 92070)

GPIB cable: 4 m (Product No. 92090)

2.1.2 RS-232C interface

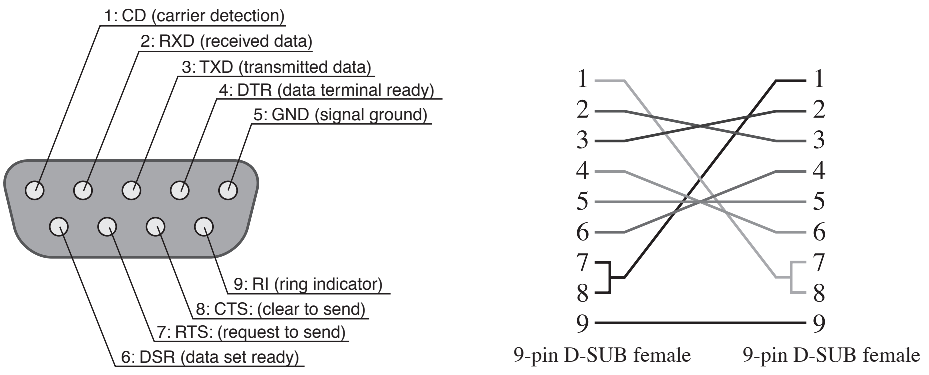
To use an RS-232C interface, the settings specified below must be made. For the setting procedure, see "1.2.2 RS-232C settings".

- RS-232C interface communication speed
- RS-232C interface data
- RS-232C interface parity
- RS-232C interface stop bit

Communication with the RS-232C must be regulated by flow control or using an acknowledge message. One-way transmission may make proper communication difficult. For details on acknowledge messages, see "2.2.1 Messages".

As the RS-232C cable, use an AT-type 9-pin D-SUB female-female cross cable.

RS-232C pin assignment



On the tester

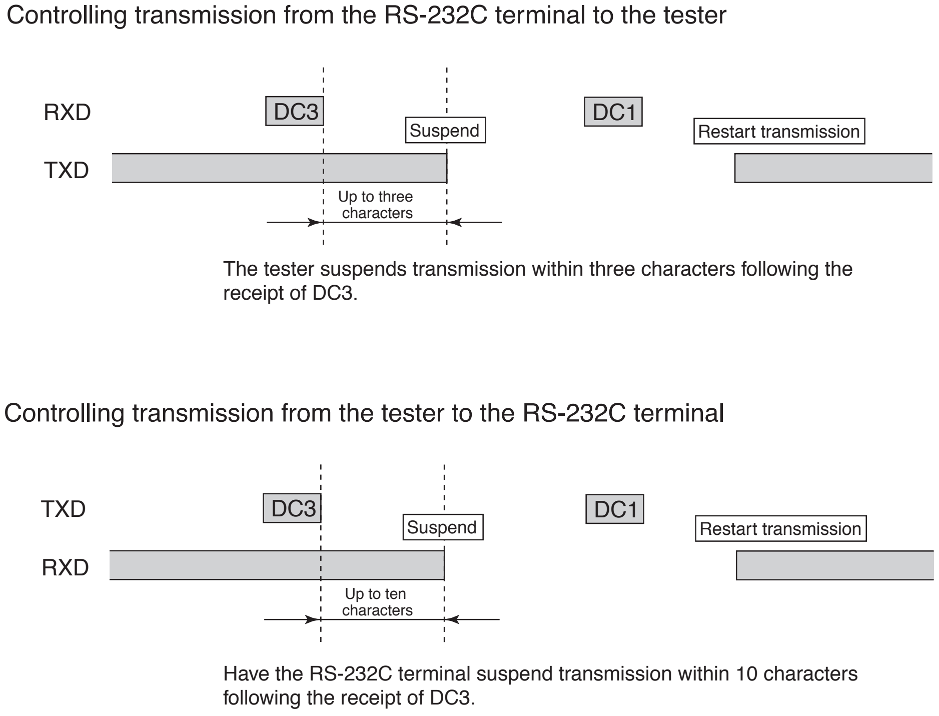
Example of a cross cable

Fig.2-1 AT-type 9-pin connector

RS-232C flow control

Communication among tester controllers can be controlled by executing Xon and Xoff. These control codes are controlled by DC (Device Control) codes.

	Function	ASCII code
DC1	Send request	11h
DC3	Send Stop request	13h


Controlling transmission from the RS-232C terminal to the tester

The tester suspends transmission within three characters following the receipt of DC3.

Controlling transmission from the tester to the RS-232C terminal

Have the RS-232C terminal suspend transmission within 10 characters following the receipt of DC3.

Fig.2-2 Controlling transmission between the RS-232C terminal and the tester

2.2 Messages and Terminators

This section explains the terms used for communication between the computer (controller) and the tester (device). See Fig. 2-3.

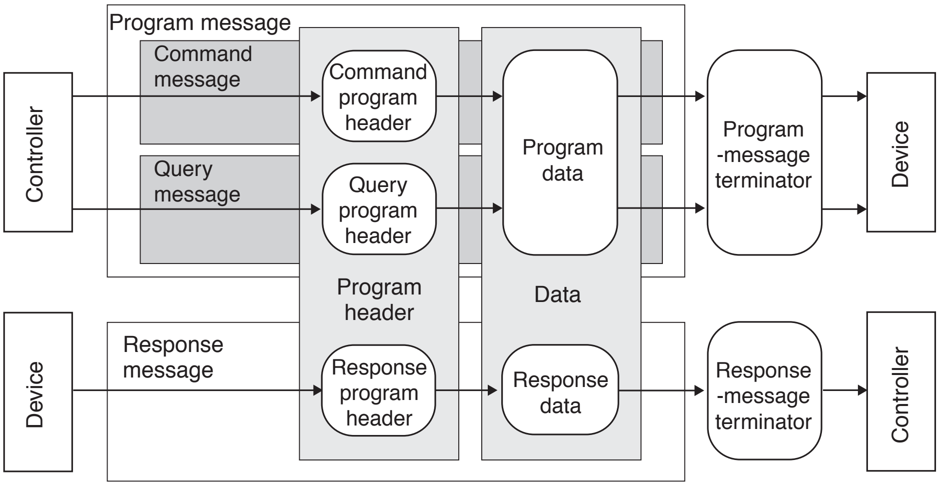


Fig.2-3 Messages and terminators

2.2.1 Messages

Commands sent from the computer to the tester are referred to as “program messages.” Responses sent from the tester to the computer are referred to as “response messages.”

Each message is composed of a program-header block and a data block.

Program messages

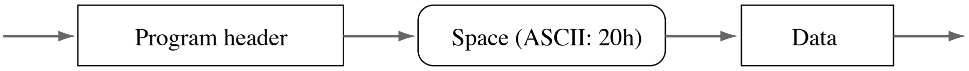
Program messages are either “command messages” or “query messages.”

Command messages are used to execute specific functions of the tester and to change settings.

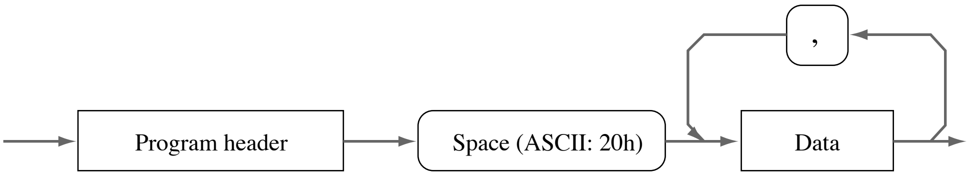
Query messages are used to inquire about the settings and status of the tester.

Describing a program message

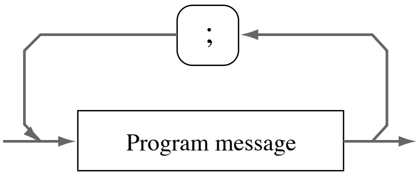
- A space (ASCII: 20h) must be inserted between the program header and the data.



- If there are two or more pieces of data, connect them using “ , ” (ASCII: 2Ch).



- To connect program messages, use “ ; ” (ASCII: 3Bh).



NOTE

- To describe data using hexadecimals, add “#H.”
  (Example) The decimal 10 is expressed as “#H0A” in hexadecimal.
- With the GPIB interface, add “@” to the end of a program message to maintain the HOLD OFF status until the message has been executed. However, in the event of a command message terminator with only EOI, use "@@".

Acknowledge message (RS-232C)

Acknowledge messages are used only with the RS-232C interface. They are sent from the tester to the controller, and are used to confirm that the handling of a program message has been completed.

Acknowledge header

Acknowledge messages contain only a header composed of character strings in ASCII codes. The following two acknowledge messages are used:

- OK : Normal termination
- ERROR : Abnormality such as a syntax error

Using the SILENT command message, settings can be made to specify whether to return an acknowledge message.

2.2.2 Terminators

A program-message terminator is used to end a program message. A response-message terminator is used to end a response message.

- Program-message terminators

  Any of the following can be used as a program-message terminator. No settings necessary:

  CR+LF+EOI, LF, LF+EOI, CR+EOI

- Response-message terminators

  By default, CR+LF+EOI is used. It can be changed to one of the following in accordance with the TRM command message. EOI is used only for GPIB.

  CR+LF+EOI, LF+EOI, EOI, CR+EOI

NOTE

- EOI is exclusively used for GPIB as a program-message terminator and a response-message terminator.

Chapter 3 Messages and Registers

Program messages and response messages supported by the tester are referred to as “device messages.”

The following describes each device message supported by the tester.

The symbol in parentheses following the device-message name is the abbreviation of the message name.

Special symbols and characters

The symbols and characters used to describe program messages and response messages are given below, along with their definitions.

Symbols and character	Description
< >	The parentheses indicate program data. In actual programming, do not use these parentheses.
{ }	Choose one from among the characters and numerals enclosed in parentheses and separated by “ : ”. In actual programming, do not use these parentheses.
_	Represents a space

3.1 Register-Related Messages and General Purpose Messages

Below, general purpose device messages and messages common to all modes are explained. general purpose device messages are used to set and reset each register, make inquiries, and specify a terminator.

*CLS

Resets the status byte register, event status register, device status register, protection register 1, protection register 2, fail register, invalid-setting register, and error register.

For each register, see "3.9 Registers".

Program message

- Syntax

  Command message: `*CLS`

*ESR

The event status register is reset when it is read by the *ESR? Message.

For details on the event status register, see "3.9 Registers".

Program message

- Syntax

  Query message: `*ESR?`

Response message

To reset, the contents of the event status register are returned to *ESR?.

(Example) When Bit 5 of the event status register is set, `32` is returned.

*IDN

Inquires about the model name and ROM version of the tester

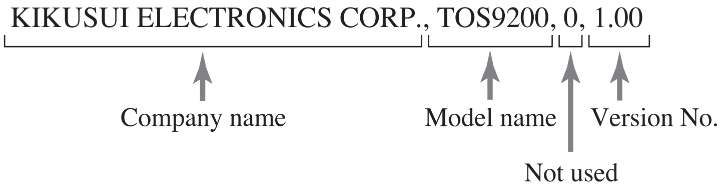
Program message

- Syntax

  Query message: `*IDN?`

Response message

The model name of the tester is returned to *IDN?, as follows:



```
KIKUSUI ELECTRONICS CORP., TOS9200, 0, 1.00
```

*RST

Initializes the tester (to original factory-set settings). Note the settings made in the INTERFACE screen will not be initialized.

For details on initialization, see the relevant section of the operation manual for the tester.

Program message

- Syntax

  Command message: `*RST`

*SRE

Sets or resets each bit of the service-request enable register, or inquires about the contents of the register.

For details on the service-request enable register, see "3.9 Registers".

Program message

- Syntax

  Command message: `*SRE_<value>`

  Query message: `*SRE?`

- Program data

Data format:	Hexadecimal or decimal
Set value:	00H to FFH (0 to 255)
Resolution:	1H (1)

(Example) To set the service-request enable register to #50H,

`*SRE #H50`

Response message

To *SRE?, the contents of the service-request enable register are returned.

(Example) When Bit 5 and Bit 6 of the service-request enable register are set, `96` is returned.

*STB

Used to inquires about the contents of the status byte register.

For details on the status byte register, see "3.9 Registers".

Program message

- Syntax

  Query message: `*STB?`

Response message

To *STB?, the contents of the status byte register are returned.

(Example) When Bit 5 and Bit 6 of the status byte register are set, `96` is returned.

CLR

Resets all registers exclusive of enable register and sets the STOP flag.

The same procedure is performed as when the GPIB bus line message DCL or SDC has been received.

Also used to perform the same processing as is performed by a DCL message from the RS-232C.

Program message

- Syntax

  Command message: `CLR`

DSE

Sets or resets each bit of the device-status enable register, or inquires about the contents of the register.

For details on the device-status enable register, see "3.9 Registers".

Program message

- Syntax

  Command message: `DSE_<`*`value`*`>`

  Query message: `DSE?`

- Program data

Data format:	Hexadecimal or decimal
Set value:	00H to FFH (0 to 255)
Resolution:	1H (1)

(Example) To set the device-status enable register to 01H,

```
DSE #H01
```

Response message

To DSE?, the contents of the device-status enable register are returned.

(Example) When Bit 5 of the device-status enable register is set, `32` is returned.

DSR

Used to inquires about the contents of the device status register.

For details on the device status register, see "3.9 Registers".

Program message

- Syntax

  Query message: `DSR?`

Response message

To DSR?, the contents of the device status register are returned.

(Example) When Bit 5 of the device-status enable register is set, `32` is returned.

ERR

The error register is reset when read by the ERR? Message.

For details on the error register, see "3.9 Registers".

Program message

- Syntax

  Query message: `ERR?`

Response message

To ERR?, the contents of the error register are returned.

(Example) When Bit 3 of the error register is set, `8` is returned.

FAIL

Inquires about the contents of the fail register.

For details on the fail register, see "3.9 Registers".

Program message

- Syntax

  Query message: `FAIL?`

Response message

To FAIL?, the contents of the fail register are returned.

(Example) When Bit 4 of the fail register is set, `16` is returned.

FUNCTION (FUN)

Shifts the LCD to each mode (setting screen) and Inquires about the current mode (displayed screen).

Program message

- Syntax

```
Command message:  FUNCTION_<value>
                  FUN_<value>
Query message:    FUNCTION?
                  FUN?
```

- Program data

  Data format: Integer

  Set value: 0: ACW; 1: DCW; 2: IR; 3: AUTO TEST; 4: AUTO EDIT; 5: SYSTEM; 6: OFFSET ADJ; 7: INTERFACE

(Example) To change the contents of the program (to shift to AUTO EDIT)

```
FUN 4
```

Response message

To FUN?, the current mode is returned.

(Example) If the current mode is the AC withstanding voltage test, 0 is returned.

INVALID (INV)

Used to inquires about the contents of the invalid-setting register.

For details on the invalid-setting register, see "3.9 Registers".

Program message

- Syntax

  Query message: `INVALID?`
  `INV?`

Response message

To INV?, the contents of the invalid-setting register are returned.

(Example) When Bit 3 of the invalid-setting register is set, 8 is returned.

LOCAL (LOC)

Return to the LOCAL from the REMOTE state.

Program message

- Syntax

  Query message: `LOCAL`
  `LOC`

PROTECTION (PROT)

Inquires about the contents of protection registers 1 and 2.

For details on the protection registers, see "3.9 Registers".

Program message

- Syntax

  Query message: `PROTECTION?`
  `PROT?`

Response message

To PROT?, the contents of protection registers 1 and 2 are returned.

(Example) When Bit 3 and Bit 5 are set in protection registers 1 and 2, 8 and 32 are returned, respectively.

SILENT (SIL)

Used in control via RS-232C to specify whether to return an acknowledgment message to a message divided by the response message terminator. The SILENT? message is used to inquire about the set value that specifies whether to return an acknowledgment message.

As an acknowledgment message, either “OK” or “ERROR.” is returned.

If an acknowledgment message is to be received, the RS-232C must be set at full duplex communication.

Full duplex communication: The transmission of data in two directions simultaneously. For full duplex settings, see the manual for your PC.

Program message

- Syntax

  Command message: `SILENT_<{0|1}>`
  `SIL_<{0|1}>`

  Query message: `SILENT?`
  `SIL?`

- Program data

  Data format: Integer

  Set value: 0: Return an acknowledge message.
  1: Do not return an acknowledge message.

(Example) To set “Do not return an acknowledge message,”

```
SIL 1
```

Response message

To SIL?, the set value for an acknowledge message is returned.

(Example) If the current settings are made to “Do not return an acknowledge message,” 1 is returned.

START

Starts testing.

While a test program is running, it shifts the program from the step suspended due to a HOLD setting in the interval to the next step.

This command message is valid only in the test conditions set up (ACW, DCW, IR), offset measurement (OFFSET), or program (AUTO) screens. Switch screens using the FUNCTION message.

Program message

- Syntax

  Command message: `START`

STOP

Stops a test. Also cancels FAIL, PASS (HOLD), and PROTECTION.

Program message

- Syntax

  Command message: `STOP`

TRM

Used to set the response-message terminator, then inquire about the set value it. Note that the GPIB uni-line message “EOI” is effective only in GPIB communication.

Program message

- Syntax

  Command message: `TRM_<{0|1|2|3}>`

  Query message: `TRM?`

- Program data

  Data format: Integer

  Set value: 0: CR/LF+EOI; 1: LF+EOI; 2: EOI; 3: CR+EOI

(Example) To set the response-message terminator to LF+EOI,

```
TRM 1
```

Response message

To TRM?, the preset response-message terminator is returned.

(Example) If the response-message terminator is set to LF+EOI, `1` is returned.

3.2 Messages Exclusively Used for AC Withstanding Voltage Testing

This section explains the messages used to check the test conditions and settings for AC withstanding voltage testing.

ACW:TESTV (A:TES)

Sets the test voltage for an AC withstanding voltage test, or inquires about the set or setting test voltage

NOTE

- This message is valid even during a test. However, make only fine voltage adjustment instead of making a large change to the voltage. A substantial change may activate the protective circuit to shift the tester to the protection status.

Program message

- Syntax

```
Command message:  ACW:TESTV_<voltage>
                  A:TES_<voltage>
Query message:    ACW:TESTV?
                  A:TES?
```

- Program data

Data format:	Real number
Set value:	0.00 to 5.20 E3 (or 0 to 5200)
Resolution:	0.01
Unit:	V

(Example) To set the test voltage to 5.00 kV,

```
A:TES 5.00E3
```

Response message

To A:TES?, the currently set test voltage is returned.

(Example) If the currently set test voltage is 2.50 kV, `2.50E3` is returned.

ACW:FREQUENCY (A:FREQ)

Sets the test frequency for an AC withstanding voltage test, or inquires about the set or setting test frequency.

Program message

- Syntax

  Command message: `ACW:FREQUENCY_<{50|60}>`
  `A:FREQ_<{50|60}>`

  Query message: `ACW:FREQUENCY?`
  `A:FREQ?`

- Program data

Data format:	Character
Set value:	50, 60
Unit:	Hz

(Example) To set the test frequency to 50 Hz,

```
A:FREQ 50
```

Response message

To A:FREQ?, the currently set test frequency is returned.

(Example) If the current test frequency is 60 Hz,
`60` is returned.

ACW:LOWER (A:LOW)

Sets the lower current (LOWER) and ON/OFF of the lower judgment function for an AC withstanding voltage test. Also inquires about the lower current and the ON/OFF status of the lower judgment function.

Program message

- Syntax

  Command message: `ACW:LOWER_<`*`lower current`*`, {ON|OFF}>`
  `A:LOW_<`*`lower current`*`, {ON|OFF}>`

  Query message: `ACW:LOWER?`
  `A:LOW?`

- Program data <*lower current*>

Data format:	Real number
Set value:	0.01 E-3 to 110 E-3
Resolution:	0.01 E-3 (0.01 E-3 to 9.99 E-3) 0.1 E-3 (10.0 E-3 to 99.9 E-3) 1 E-3 (100 E-3 to 110 E-3)
Unit:	A

- Program data <{ON|OFF}>

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the lower current (LOWER) to 10.0 mA,

```
A: LOW 10.0E-3,ON
```

Response message

To A:LOW?, the currently set lower current and the ON/OFF status of the lower judgment function are returned as “lower current, 1/0.”

(Example) If the present lower current is 5.00 mA and the lower judgment function is OFF, `5.00E-3,0` is returned.

ACW:UPPER (A:UPP)

Sets the upper current (UPPER) for an AC withstanding voltage test. Also inquires about the present upper current.

⚠CAUTION • If the upper current is set above 50 mA, the protective circuit may be activated. To prevent this, set the output time below 30 minutes and the pause to longer than the output time.

Program message

- Syntax

  Command message: `ACW:UPPER_<`*`upper current`*`>`
  `A:UPP_<`*`upper current`*`>`

  Query message: `ACW:UPPER?`
  `A:UPP?`

- Program data

  Data format: Real number

  Set value: 0.01 E-3 to 110 E-3

  Resolution: 0.01 E-3 (0.01 E-3 to 9.99 E-3)
  0.1 E-3 (10.0 E-3 to 99.9 E-3)
  1 E-3 (100 E-3 to 110 E-3)

  Unit: A

(Example) To set the upper current to 50.0 mA,

```
A:UPP 50.0E-3
```

Response message

To A:UPP?, the present upper current is returned.

(Example) If the present upper current is 80.0 mA, `80.0E-3` is returned.

ACW:TIMER (A:TIM)

Sets the test time (TEST TIME) and ON/OFF of the timer function for an AC withstanding voltage test. Also inquires about the current test time and the ON/OFF status of the timer function.

⚠CAUTION
- If the upper current is set above 50 mA, the protective circuit may be activated. To prevent this, set the output time below 30 minutes and the pause to longer than the output time.

Program message

- Syntax

  Command message: `ACW:TIMER_<test time, {ON|OFF}>`
  `A:TIM_<test time, {ON|OFF}>`

  Query message: `ACW:TIMER?`
  `A:TIM?`

- Program data <*test time*>

Data format:	Real number
Set value:	0.3 to 999
Resolution:	0.1 for 0.3 to 99.9, 1 for 100 to 999
Unit:	s

- Program data <`{ON|OFF}`>

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the test time to 5 s,

```
A:TIM 5,ON
```

Response message

To A:TIM?, the current test time and the ON/OFF status of the timer function are returned as “test time, ON/OFF.”

(Example) If the current test time is 2.0 s and the timer function is set to OFF, `2.0,0` is returned.

ACW:OFFSET (A:OFF)

Sets ON/OFF of the offset function for an AC withstanding voltage test. Also inquires about the ON/OFF status of the offset function.

Program message

- Syntax

Command message:	`ACW:OFFSET_<{ON|OFF}>`
	`A:OFF_<{ON|OFF}>`

Query message:	`ACW:OFFSET?` `A:OFF?`

- Program data

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the offset function to ON,

```
A:OFF 1
```

Response message

To A:OFF?, the ON/OFF status of the offset function is returned.

(Example) If the offset function is currently set to OFF,
`0` is returned.

ACW:STARTV (A:STAR)

Sets the start voltage for an AC withstanding voltage test. Also inquires about the set or setting start voltage.

Program message

- Syntax

Command message:	`ACW:STARTV_<`*`start voltage`*`>`
	`A:STAR_<`*`start voltage`*`>`

Query message:	`ACW:STARTV?` `A:STAR?`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1
Unit:	%

(Example) To set the start voltage to 10% of the test voltage,

```
A:STAR 10
```

Response message

To A:STAR?, the current start voltage is returned.

(Example) If the start voltage is currently set to 0%,
`0` is returned.

ACW:RISETIME (A:RTIM)

Sets the voltage rise time (RISE TIME) for an AC withstanding voltage test. Also inquires about the set or setting voltage rise time.

Program message

- Syntax

Command message:	`ACW:RISETIME_<voltage rise time>`
	`A:RTIM_<voltage rise time>`

Query message:	`ACW:RISETIME?` `A:RTIM?`

- Program data

Data format:	Real number
Set value:	0.1 to 200
Resolution:	0.1 for 0.1 to 99.9, 1 for 100 to 200
Unit:	s

(Example) To set the voltage rise time to 1 s,

```
A:RTIM 1
```

Response message

To A:RTIM?, the current voltage rise time is returned.

(Example) If the voltage rise time is currently set to 2.0 s, `2.0` is returned.

ACW:FALLTIME (A:FTIM)

Sets the voltage fall time (FALL TIME) for an AC withstanding voltage test. Also inquires about the set or setting voltage fall time.

Program message

- Syntax

Command message:	`ACW:FALLTIME_<voltage fall time>`
	`A:FTIM_<voltage fall time>`

Query message:	`ACW:FALLTIME?` `A:FTIM?`

- Program data

Data format:	Real number
Set value:	0.0 to 200
Resolution:	0.1 for 0.0 to 99.9, 1 for 100 to 200
Unit:	s

(Example) To set the voltage fall time to 1 s,

```
A:FTIM 1
```

Response message

To A:FTIM?, the current voltage fall time is returned.

(Example) If the voltage fall time is currently set to 2.0 s,
`2.0` is returned.

ACW:VRANGE (A:VRAN)

Sets the output-voltage range for an AC withstanding voltage test. Also inquires about the set or setting output range.

Program message

- Syntax

Command message: `ACW:VRANGE_<{0|1}>`
`A:VRAN_<{0|1}>`

Query message: `ACW:VRANGE?`
`A:VRAN?`

- Program data

Data format:	Integer
Set value:	0: AUTO; 1: 5 kV

(Example) To set the output range to AUTO,

```
A:VRAN 0
```

Response message

To A:VRAN?, the output-voltage range is returned.

(Example) If the current output-voltage range is 5 kV,
`1` is returned.

ACW:RESPONSE (A:RES)

Sets the current detection response speed (RESPONSE) for an AC withstanding voltage test.

Also inquires about the set or setting current detection response speed.

Program message

- Syntax

Command message: `ACW:RESPONSE_<{0|1|2}>`
`A:RES_<{0|1|2}>`

Query message: `ACW:RESPONSE?`
`A:RES?`

- Program data

Data format:	Integer
Set value:	0: SLOW; 1: MID; 2: FAST

(Example) To set the current detection response speed to FAST,

```
A:RES 2
```

Response message

To A:RES?, the preset current detection response speed is returned.

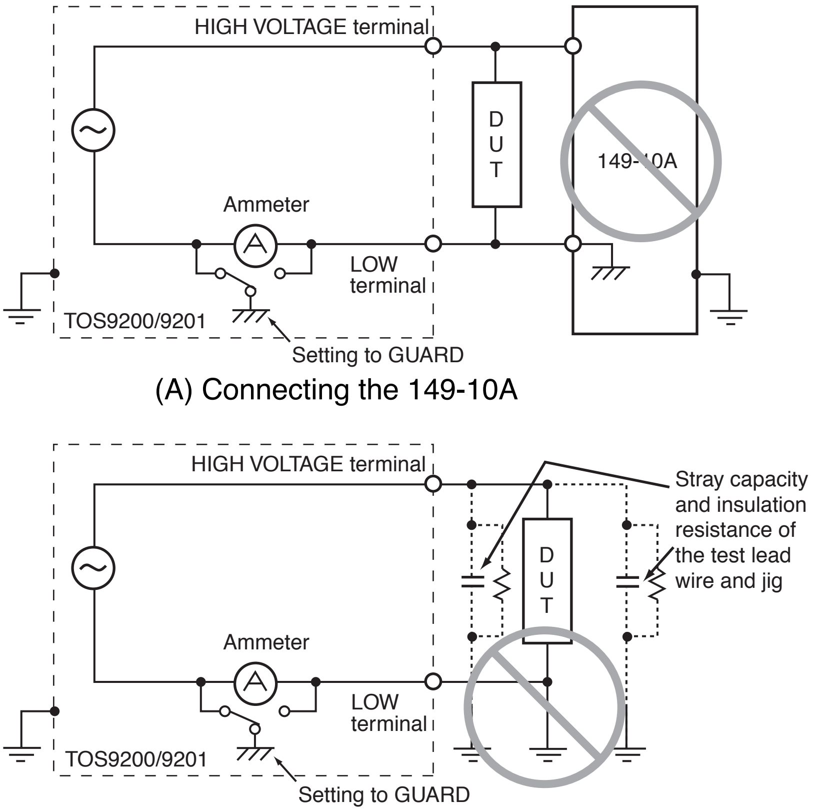
(Example) If the detection response speed is currently set to MID, 1 is returned.

ACW:GND (A:GND)

Sets LOW/GUARD of the GND for an AC withstanding voltage test. Also inquires about the set or setting value of the GND.

⚠WARNING

- If it is not known whether a DUT or jig is grounded, never select GUARD. If GUARD is selected with a DUT grounded, the ammeter will be short-circuited, disabling measurement and posing grave danger. See Fig. 3-1 (B).
- When GUARD is selected, do not connect the tester to devices grounded on a single side, such as Kikusui's high-voltage digital voltmeter 149-10A and the current calibrator TOS1200. Otherwise, the ammeter will be short-circuited. See Fig. 3-1 (A).



(A) Connecting the 149-10A

(B) Selecting GUARD when the DUT is grounded

Fig.3-1 Dangerous connections

⚠CAUTION • When the tester’s LOW terminal is connected to the HIGH or LOW terminal on the earth continuity tester TOS6200, the ammeter will make measurement errors if GUARD is selected. This is due to the fact that the resistor in the TOS6200 is connected in parallel with the tester’s ammeter. To use the TOS6200, do not connect one terminal to another, or set the LOW terminal to LOW.

Program message

- Syntax

  Command message: `ACW:GND_<{GUARD|LOW}>`
  `A:GND_<{GUARD|LOW}>`

  Query message: `ACW:GND?`
  `A:GND?`

- Program data

  Data format: Character

  Set value: LOW (0), GUARD (1)

(Example) To set the GND to GUARD,

`A:GND 1`

Response message

To A:GND?, the current GND setting is returned.

(Example) If the GND is currently set to LOW,
`0` is returned.

ACW:SCAN (A:SCAN)

Sets HIGH/LOW/OPEN for the scanner channel for an AC withstanding voltage test.

Also inquires about the set or setting value of the high-voltage scanner channel.

Program message

- Syntax

  Command message: `ACW:SCAN_<`*`channel`*`, {H|L|O}>`
  `A:SCAN_<`*`channel`*`, {H|L|O}>`

  Query message: `ACW:SCAN?_<`*`channel`*`>`
  `A:SCAN?_<`*`channel`*`>`

- Program data <*`channel`*>

  Data format: Integer

  Set value: 1 to 16

  Resolution: 1

- Program data `<{H|L|O}>`

  Data format: Character

Set value: O (0): OPEN; L (1): LOW; H (2): HIGH

(Example) To set scanner channel 1 to HIGH,

```
A:SCAN 1,H
```

Response message

To A:SCAN?2, the preset value for scanner channel 2 is returned.

(Example) If scanner channel 2 is currently set to OPEN,
`0` is returned.

ACW:SCANW (A:SCANW)

Sets HIGH/LOW/OPEN for 16 scanner channels at a time using 16 bits for an AC withstanding voltage test. Also inquires about the current settings for all 16 channels.

Program message

- Syntax

  Command message: `ACW:SCANW_<`*`setting 1, setting 2`*`>`
  `A:SCANW_<`*`setting 1, setting 2`*`>`

  Query message: `ACW:SCANW?`
  `A:SCANW?`

- Program data <*setting 1*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  LOW is represented by 0 and HIGH by 1.

- Program data <*setting 2*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  OPEN is represented by 1 and NOT OPEN by 0.

(Example) To set scanner channels 1 through 4 to LOW, scanner channels 5 through 8 to HIGH, and the other channels to OPEN,

```
A:SCANW #H00F0,#HFF00
```

OPEN in setting 2 has priority over setting 1.

Response message

To A:SCANW?, the settings for all 16 channels are returned as “setting 1” and “setting 2” using 16 bits.

(Example) If scanner channels 1 through 5 are currently set to HIGH, channels 6 through 10 to OPEN, and the other channels to LOW,
`31,992` is returned.

ACW:CONTACTCHECK (A:CCH)

Sets ON/OFF of the contact check function for an AC withstanding voltage test. Also inquires about the status of the function.

Program message

- Syntax

Command message:	`ACW:CONTACTCHECK_<{ON|OFF}>`
	`A:CCH_<{ON|OFF}>`

Query message:	`ACW:CONTACTCHECK?` `A:CCH?`

- Program data

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the contact check to ON,

```
A:CCH ON
```

Response message

To A:CCH?, the current setting for the contact check is returned.

(Example) If the contact check is currently set to ON,
`1` is returned.

3.3 Messages Used Exclusively for DC Withstanding Voltage Testing

This section explains the messages used to check the test conditions and settings for DC withstanding voltage testing.

These messages are exclusively designed for the TOS9201 equipped with a DC withstanding voltage testing function.

With the TOS9200, these messages are invalid. However, they do not cause an error.

DCW:TESTV (D:TES)

Sets the test voltage for a DC withstanding voltage test. Also inquires about the set or setting test voltage

NOTE

- This message is valid even during a test. However, make only fine voltage adjustments rather than large changes to the voltage. A substantial change may activate the protective circuit to shift the tester to the protection status.

Program message

- Syntax

Command message: `DCW:TESTV_<voltage>`
`D:TES_<voltage>`

Query message: `DCW:TESTV?`
`D:TES?`

- Program data

Data format:	Real number
Set value:	0.00 E3to 6.10 E3 (or 0 to 6100)
Resolution:	0.01 E3
Unit:	V

(Example) To set the test voltage to 5.00 kV,

```
D:TES 5.00E3
```

Response message

To D:TES?, the currently set test voltage is returned.

(Example) If the test voltage is currently set to 2.50 kV,
`2.50E3` is returned.

DCW:LOWER (D:LOW)

Sets the lower current (LOWER) and ON/OFF of the lower judgment function for a DC withstanding voltage test. Also inquires about the lower current and the ON/OFF status of the lower judgment function.

Program message

- Syntax

  Command message: `DCW:LOWER_<`*`lower current`*`, {ON|OFF}>`
  `D:LOW_<`*`lower current`*`, {ON|OFF}>`

  Query message: `DCW:LOWER?`
  `D:LOW?`

- Program data <*`lower current`*>

Data format:	Real number
Set value:	0.01 E-3 to 11.0 E-3
Resolution:	0.01 for 0.01 E-3 to 9.99 E-3, 0.1 for 10.0 E-3 to 11.0 E-3
Unit:	A

- Program data <`{ON|OFF}`>

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the lower current to 1 mA,

```
D:LOW 1E-3,ON
```

Response message

To D:LOW?, the currently set lower current and the ON/OFF status of the lower judgment function are returned as “lower current, ON/OFF.”

(Example) If the lower current is currently set to 2.0 mA and the lower judgment function to OFF,
“`2.00E-3,0`” is returned.

DCW:UPPER (D:UPP)

Sets the upper current (UPPER) for a DC withstanding voltage test. Also inquires about the set or setting upper current.

⚠CAUTION • If the upper current is set above 5 mA, the protective circuit may be activated. To prevent this, set the output time below 1 minute and the pause to longer than the output time.

Program message

- Syntax

  Command message: `DCW:UPPER_<`*`upper current`*`>`
  `D:UPP_<`*`upper current`*`>`

  Query message: `DCW:UPPER?`
  `D:UPP?`

- Program data

Data format:	Real number
Set value:	0.01 E-3 to 11.0 E-3
Resolution:	0.01 for 0.01 E-3 to 9.99 E-3, 0.1 for 10.0 E-3 to 11.0 E-3
Unit:	A

(Example) To set the upper current to 5 mA,

```
A:UPP 5E-3
```

Response message

To D:UPP?, the preset upper current is returned.

(Example) If the upper current is currently set to 2.00 mA,
`2.00E-3` is returned.

DCW:TIMER (D:TIM)

Sets the test time (TEST TIME) and ON/OFF of the timer function for an DC withstanding voltage test. Also inquires about the current test time and the ON/OFF status of the timer function.

Program message

- Syntax

  Command message: `DCW:TIMER_<`*`test time`*`, {ON|OFF}>`
  `D:TIM_<`*`test time`*`, {ON|OFF}>`

  Query message: `DCW:TIMER?`
  `D:TIM?`

- Program data <*test time*>

  Data format: Real number

Set value:	0.3 to 999
Resolution:	0.1 for 0.3 to 99.9, 1 for 100 to 999
Unit:	s

- Program data `<{ON|OFF}>`

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the test time to 5 s,

```
D:TIM 5,ON
```

Response message

To D:TIM?, the current test time and the ON/OFF status of the timer function are returned as “test time, 1/0.”

(Example) If the test time is currently set to 2.0 s and the timer function to OFF,
`2.0,0` is returned.

DCW:STARTV (D:STAR)

Sets the start voltage for a DC withstanding voltage test. Also inquires about the set or setting start voltage.

Program message

- Syntax

Command message:	`DCW:STARTV_<`*`start voltage`*`>`
	`D:STAR_<`*`start voltage`*`>`

Query message:	`DCW:STARTV?` `D:STAR?`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1
Unit:	%

(Example) To set the start voltage to 10% of the test voltage,

```
D:STAR 10
```

Response message

To D:STAR?, the current start voltage is returned.

(Example) If the start voltage is currently set to 0%,
`0` is returned.

DCW:RISETIME (D:RTIM)

Sets the voltage rise time (RISE TIME) for a DC withstanding voltage test. Also inquires about the set or setting voltage rise time.

Program message

- Syntax

Command message:	`DCW:RISETIME_<`*`voltage rise time`*`>`
	`D:RTIM_<`*`voltage rise time`*`>`

Query message:	`DCW:RISETIME?` `D:RTIM?`

- Program data

Data format:	Real number
Set value:	0.1 to 200
Resolution:	0.1 for 0.1 to 99.9, 1 for 100 to 200
Unit:	s

(Example) To set the voltage rise time to 1 s,

```
D:RTIM 1
```

Response message

To D:RTIM?, the current voltage rise time is returned.

(Example) If the voltage rise time is currently set to 2.0 s,
`2.0` is returned.

DCW:WAITTIME (D:WTIM)

Sets the WAIT TIME for a DC withstanding voltage test. Also inquires about the set or setting WAIT TIME.

Program message

- Syntax

Command message:	`DCW:WAITTIME_<`*`WAIT TIME`*`>`
	`D:WTIM_<`*`WAIT TIME`*`>`

Query message:	`DCW:WAITTIME?` `D:WTIM?`

- Program data

Data format:	Real number
Set value:	0.3 to 10.0
Resolution:	0.1
Unit:	s

(Example) To set the WAIT TIME to 1 s,

```
D:WTIM 1
```

Response message

To D:WTIM?, the currently set WAIT TIME is returned.

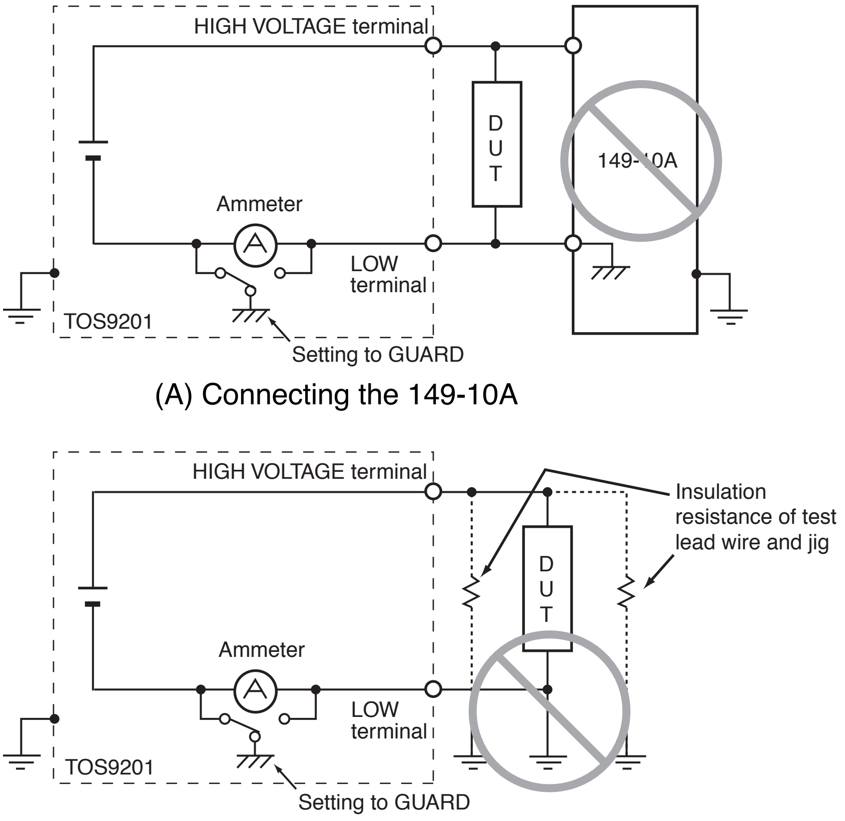
(Example) If the WAIT TIME is currently set to 2.0 s,
`2.0` is returned.

DCW:GND (D:GND)

Sets LOW/GUARD of the GND for a DC withstanding voltage test. Also inquires about the set or setting value of the GND.

⚠ WARNING

- If it is not known whether a DUT or jig is grounded, never select GUARD.
  If GUARD is selected with a DUT grounded, the ammeter will be short-circuited, disabling measurement and posing a grave danger. See Fig. 3-2 (B).
- When GUARD is selected, do not connect the tester to devices grounded on a single side, such as Kikusui's high-voltage digital voltmeter 149-10A or current calibrator TOS1200. Otherwise, the ammeter will be short-circuited. See Fig. 3-2 (A).



(A) Connecting the 149-10A

(B) Selecting GUARD when the DUT is grounded

Fig.3-2 Dangerous connections

⚠CAUTION • When the tester's LOW terminal is connected to the HIGH or LOW terminal of the earth continuity tester TOS6200, the ammeter will suffer measurement errors if GUARD is selected, as the resistor in the TOS6200 is connected in parallel with the tester's ammeter. To use the TOS6200, either do not connect one terminal to another, or set the LOW terminal to LOW.

Program message

- Syntax

Command message: `DCW:GND_<{GUARD|LOW}>`
`D:GND_<{GUARD|LOW}>`

Query message: `DCW:GND?`
`D:GND?`

- Program data

Data format:	Character
Set value:	LOW (0), GUARD (1)

(Example) To set the GND to GUARD,

```
D:GND 1
```

Response message

To D:GND?, the current GND setting is returned.

(Example) If the GND is currently set to LOW,
0 is returned.

DCW:SCAN (D:SCAN)

Sets HIGH/LOW/OPEN for the scanner channel in a DC withstanding voltage test. Also inquires about the current setting for the scanner channel.

Program message

- Syntax

Command message: `DCW:SCAN_<`*`channel`*`, {H|L|O}>`
`D:SCAN_<`*`channel`*`, {H|L|O}>`

Query message: `DCW:SCAN?_<`*`channel`*`>`
`D:SCAN?_<`*`channel`*`>`

- Program data <*`channel`*>

Data format:	Integer
Set value:	1 to 16
Resolution:	1

- Program data `<{H|L|O}>`

Data format:	Character
Set value:	O (0): OPEN; L (1): LOW; H (2): HIGH

(Example) To set scanner channel 1 to HIGH,

```
D:SCAN 1,H
```

Response message

To D:SCAN?2, the preset value for scanner channel 2 is returned.

(Example) If scanner channel 2 is currently set to OPEN,
`0` is returned.

DCW:SCANW (D:SCANW)

Sets HIGH/LOW/OPEN for 16 channels at a time using 16 bits in a DC withstanding voltage test. Also inquires about the current settings for all 16 channels.

Program message

- Syntax

  Command message: `DCW:SCANW_<`*`setting 1, setting 2`*`>`
  `D:SCANW_<`*`setting 1, setting 2`*`>`

  Query message: `DCW:SCANW?`
  `D: SCANW?`

- Program data <*`setting 1`*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  LOW is represented by 0 and HIGH by 1.

- Program data <*`setting 2`*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  OPEN is represented by 1 and NOT OPEN by 0.

(Example) To set scanner channels 1 through 4 to LOW, scanner channels 5 through 8 to HIGH, and the other channels to OPEN,

```
D:SCANW #H00F0,#HFF00
```

OPEN in setting 2 has priority over setting 1.

Response message

To D:SCANW?, the settings for all 16 channels are returned as “setting 1” and “setting 2” using 16 bits.

(Example) If scanner channels 1 through 5 are currently set to HIGH, channels 6 through 10 to OPEN, and the other channels to LOW,
“`31,992`” is returned.

DCW:CONTACTCHECK (D:CCH)

Sets ON/OFF of the contact check function for a DC withstanding voltage test. Also inquires about the status of the function.

Program message

- Syntax

Command message:	`DCW:CONTACTCHECK_<{ON|OFF}>`
	`D:CCH_<{ON|OFF}>`

Query message:	`DCW:CONTACTCHECK?` `D:CCH?`

- Program data

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the contact check to ON,

```
D:CCH ON
```

Response message

To D:CCH?, the current setting for the contact check is returned.

(Example) If the contact check is currently set to ON,
`1` is returned.

3.4 Messages Used Exclusively for Insulation Resistance Testing

This section explains the messages used to check the test conditions and settings for insulation resistance testing.

IR:TESTV (I:TES)

Sets the test voltage for an insulation resistance test. Also inquires about the preset test voltage.

NOTE

- This message is valid even during a test. However, make only fine voltage adjustments rather than a large change to the voltage. A substantial change may activate the protective circuit to shift the tester to the protection status.

Program message

- Syntax

Command message: `IR:TESTV_<voltage>`
`I:TES_<voltage>`

Query message: `IR:TESTV?`
`I:TES?`

- Program data

Data format:	Integer
Set value:	10 to 1020
Resolution:	1
Unit:	V

(Example) To set the test voltage to 500 V,

```
I:TES 500
```

Response message

To I:TES?, the currently set test voltage is returned.

(Example) If the test voltage is currently set to 250 V,
`250` is returned.

IR:LOWER (I:LOW)

Sets the lower resistance (LOWER) and ON/OFF of the lower judgment function for an insulation resistance test. Also inquires about the lower resistance and the ON/OFF status of the lower judgment function.

⚠CAUTION

- If the lower or upper judgment function is OFF, no FAIL judgment is made for that function. With this setting, note that, a PASS judgment is made when the timer is turned on.

Program message

- Syntax

Command message: `IR:LOWER_<`*`lower resistance`*`, {ON|OFF}>`
`I:LOW_<`*`lower resistance`*`, {ON|OFF}>`

Query message: `IR:LOWER?`
`I:LOW?`

- Program data

Data format:	Real number
Set value:	0.01 E6 to 9.99 E9
Resolution:	0.01 E6 (0.01 E6 to 9.99 E6)
	0.1 E6 (10.0 E6 to 99.9 E6) 1 E6 (100 E6 to 999 E6)
	0.01 E9 (1.00 E9 to 9.99 E9)
Unit:	Ω

(Example) To set the lower resistance to 99.9 MΩ,

```
I:LOW 99.9E6,1
```

Response message

To I:LOW?, the currently set lower resistance and the ON/OFF status of the lower judgment function are returned as “lower resistance, 1/0.”

(Example) If the lower resistance is currently set to 1.00 GΩ and the lower judgment function to OFF,
“`1.00E9,0`” is returned.

IR:UPPER (I:UPP)

Sets the upper resistance (UPPER) and ON/OFF of the upper judgment function for an insulation resistance test. Also inquires about the set or setting upper resistance and the ON/OFF status of the upper judgment function.

Program message

- Syntax

Command message: `IR:UPPER_<upper resistance, {ON|OFF}>`
`I:UPP_<upper resistance, {ON|OFF}>`

Query message: `IR:UPPER?`
`I:UPP?`

- Program data

Data format:	Real number
Set value:	0.01 E6 to 9.99 E9
Resolution:	0.01 E6 (0.01 E6 to 9.99 E6)
	0.1 E6 (10.0 E6 to 99.9 E6) 1 E6 (100 E6 to 999 E6)
	0.01 E9 (1.00 E9 to 9.99 E9)
Unit:	Ω

- Program data `<{ON|OFF}>`

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the upper resistance to 1.00 GΩ,

```
I: UPP 1.00E9,1
```

Response message

To I:UPP?, the preset upper resistance and the ON/OFF status of the upper judgment function are returned as “upper resistance, 1/0.”

(Example) If the upper resistance is currently set to 5.00 GΩ and the upper judgment function to OFF,
“`5.00E9,0`” is returned.

IR:TIMER (I:TIM)

Sets the test time (TEST TIME) and ON/OFF of the timer function for an insulation resistance test. Also inquires about the current test time and the ON/OFF status of the timer function.

Program message

- Syntax

Command message: `IR:TIMER_<test time, {ON|OFF}>`
`I:TIM_<test time, {ON|OFF}>`

Query message: `IR:TIMER?`
`I:TIM?`

- Program data <*`test time`*>

Data format:	Real number
Set value:	0.5 to 999
Resolution:	0.1 for 0.5 to 99.9, 1 for 100 to 999
Unit:	s

- Program data <`{ON|OFF}`>

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the test time to 5 s,

```
I:TIM 5,ON
```

Response message

To I:TIM?, the current test time and the ON/OFF status of the timer function are returned.

(Example) If the test time is currently set to 2 s and the timer function to OFF, “`2.0,0`” is returned.

IR:RISETIME (I:RTIM)

Sets the voltage rise time (RISE TIME) for an insulation resistance test. Also inquires about the set or setting voltage rise time.

Program message

- Syntax

Command message: `IR:RISETIME_<`*`voltage rise time`*`>`
`I:RTIM_<`*`voltage rise time`*`>`

Query message: `IR:RISETIME?`
`I:RTIM?`

- Program data

Data format:	Real number
Set value:	0.1 to 200
Resolution:	0.1 for 0.1 to 99.9, 1 for 100 to 200
Unit:	s

(Example) To set the voltage rise time to 1 s,

```
I:RTIM 1
```

Response message

To I:RTIM?, the current voltage rise time is returned.

(Example) If the voltage rise time is currently set to 2.0 s, `2.0` is returned.

IR:WAITTIME (I:WTIM)

Sets the WAIT TIME for an insulation resistance test. Also inquires about the set or setting WAIT TIME.

Program message

- Syntax

  Command message: `IR:WAITTIME_<WAIT TIME>`
  `I:WTIM_<WAIT TIME>`

  Query message: `IR:WAITTIME?`
  `I:WTIM?`

- Program data

Data format:	Real number
Set value:	0.3 to 10.0
Resolution:	0.1
Unit:	s

(Example) To set the WAIT TIME to 1 s,

```
I:WTIM 1
```

Response message

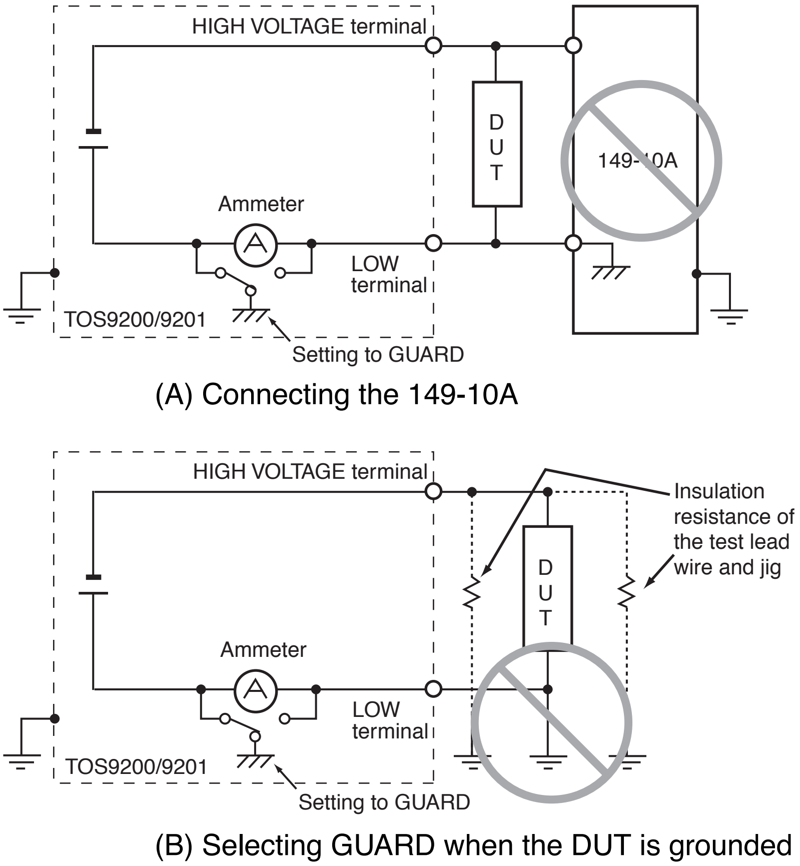
To I:WTIM?, the currently set WAIT TIME is returned.

(Example) If the WAIT TIME is currently set to 2.0 s,
`2.0` is returned.

IR:GND (I:GND)

Sets LOW/GUARD of the GND for an insulation resistance test. Also inquires about the set or setting value of the GND.

⚠WARNING
- If it is not known whether a DUT or jig is grounded, never select GUARD. If GUARD is selected with a DUT grounded, the ammeter is short-circuited, disabling measurement and posing grave danger. See Fig. 3-3 (B).
- When GUARD is selected, do not connect the tester to devices grounded on a single side, such as Kikusui's high-voltage digital voltmeter 149-10A or current calibrator TOS1200. Otherwise, the ammeter will be short-circuited. See Fig. 3-3 (A).



(A) Connecting the 149-10A

(B) Selecting GUARD when the DUT is grounded

Fig.3-3 Dangerous connections

⚠CAUTION
- When the tester's LOW terminal is connected to the HIGH or LOW terminal on the earth continuity tester TOS6200, the ammeter will suffer measurement errors if GUARD is selected, as the resistor in the TOS6200 is connected in parallel with the tester's ammeter. To use the TOS6200, either do not connect one terminal to another, or set the LOW terminal to LOW.

Program message

- Syntax

  Command message: `IR:GND_<{GUARD|LOW}>`
  `I:GND_<{GUARD|LOW}>`

  Query message: `IR:GND?`
  `I:GND?`

- Program data

Data format:	Character
Set value:	LOW (0), GUARD (1)

(Example) To set the GND to GUARD,

```
I:GND 1
```

Response message

To I:GND?, the current GND setting is returned.

(Example) If the GND is currently set to LOW,
0 is returned.

IR:SCAN (I:SCAN)

Sets HIGH/LOW/OPEN for the scanner channel in an insulation resistance test. Also inquires about the current setting of the high-voltage scanner channel.

Program message

- Syntax

Command message:	`IR:SCAN_<channel, {H|L|O}>`
	`I:SCAN_<channel, {H|L|O}>`

Query message:	`IR:SCAN?_<channel>`
	`I:SCAN?_<channel>`

- Program data *<channel>*

Data format:	Integer
Set value:	1 to 16
Resolution:	1

- Program data <{H|L|O}>

Data format:	Character
Set value:	O (0): OPEN; L (1): LOW; H (2): HIGH

(Example) To set scanner channel 1 to HIGH,

```
I:SCAN 1,H
```

Response message

To I:SCAN? 2, the scanner channel 2 setting is returned.

(Example) If the scanner channel is currently set to OPEN,
0 is returned.

IR:SCANW (I:SCANW)

Sets HIGH/LOW/OPEN for 16 scanner channels at a time using 16 bits for an insulation resistance test. Also inquires about the settings for all 16 channels.

Program message

- Syntax

  Command message: `IR:SCANW_<`*`setting 1, setting 2`*`>`
  `I:SCANW_<`*`setting 1, setting 2`*`>`

  Query message: `IR:SCANW?`
  `I:SCANW?`

- Program data `<`*`setting 1`*`>`

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  LOW is represented by 0 and HIGH by 1.

- Program data `<`*`setting 2`*`>`

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  OPEN is represented by 1 and NOT OPEN by 0.

(Example) To set scanner channels 1 through 4 to LOW, scanner channels 5 through 8 to HIGH, and the other channels to OPEN,

```
I:SCANW #H00F0,#HFF00
```

OPEN in setting 2 has priority over setting 1.

Response message

To I:SCANW?, the settings for all 16 channels of the high-voltage scanner are returned as “setting 1” and “setting 2” using 16 bits.

(Example) If scanner channels 1 through 5 are currently set to HIGH, channels 6 through 10 to OPEN, and the other channels to LOW,
`31,992` is returned.

IR:CONTACTCHECK (I:CCH)

Sets ON/OFF of the contact check function for an insulation resistance test. Also inquires about the status of the function.

Program message

- Syntax

  Command message: `IR:CONTACTCHECK_<{ON|OFF}>`
  `I:CCH_<{ON|OFF}>`

  Query message: `IR:CONTACTCHECK?`
  `I:CCH?`

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the contact check to ON,

```
I:CCH ON
```

Response message

To I:CCH?, the current setting for the contact check is returned.

(Example) If the contact check is currently set to ON,
`1` is returned.

3.5 Messages Common to All Tests

This section explains the messages commonly used in all tests.

VDATA (VDAT)

Inquires about the monitor voltage for each test. During the test, the real-time voltage is returned. After the test, the previous test voltage is returned.

Program message

- Syntax

Query message:
```
VDATA?
VDAT?
```

Response message

To VDAT?, the monitor voltage is returned for ACW in an AC withstanding voltage test, for DCW in a DC withstanding voltage test, and for IR in an insulation resistance test.

(Example) If the current monitor voltage in an AC withstanding voltage test is 1.5 kV,
`1.50E3` is returned.

NOTE

- In insulation resistance testing with the lower judgment function off, even if settings are made so that the test voltage divided by the lower resistance exceeds 1.1 mA, testing can be performed. However, when the voltage value which is out of the range of ±(10 % of setting +50 V) is detected during the test, the measurement voltage value flashes on the LCD.

  In this case, -1 is returned.

IDATA (IDAT)

Inquires about the monitor current.

During the test, the measured normal value or the maximum value is returned, depending on the MEAS MODE setting in the system settings. After the test, the previous test current is returned.

Program message

- Syntax

Query message:
```
IDATA?
IDAT?
```

Response message

To IDAT?, the monitor current is returned for ACW in an AC withstanding voltage test and for DCW in a DC withstanding voltage test.

(Example) If the present monitor current in an AC withstanding voltage test is 10.0 mA,
`10.0E-3` is returned.

RDATA (RDAT)

Inquires about the monitor resistance

During the test, the measured normal value or the maximum value is returned, depending on the MEAS MODE setting in the system settings. After the test, the previous monitor resistance is returned.

Program message

- Syntax

  Query message: `RDATA?`
  `RDAT?`

Response message

To RDAT?, the monitor resistance is returned for IR in an insulation resistance test.

(Example) If the current monitor resistance is 10.0 MΩ,
`10.0E6` is returned.

NOTE

- In insulation resistance testing, when the resistance value to exceed 99.9 GΩ is detected, “99.9” flashes on the LCD.

  In this case, -1 is returned.

REALDATA (REAL)

Inquires about the real current.

During the test, the real-time real current is returned. After the test, the previous real current is returned.

Program message

- Syntax

  Query message: `REALDATA?`
  `REAL?`

Response message

To REAL?, the present real current is returned.

(Example) If the present real current is 1 mA,
`1E-3` is returned.

IMAGDATA (IMAG)

Inquires about the imaginary current. During the test, the real-time imaginary current is returned. After the test, the previous imaginary current is returned.

Program message

- Syntax

  Query message: `IMAGDATA?`
`IMAG?`

Response message

To IMAG?, the present imaginary current is returned.

(Example) If the present imaginary current is 20 μA,
`20E-6` is returned.

TIME

Inquires about the test time elapsed (the remaining time when the timer function is on) and the test status

Program message

- Syntax

  Query message: `TIME?`

Response message

To TIME?, the time elapsed (the remaining time when the timer function is on) and the test status are returned as "time, status."

Rising (Rise): 0; Testing (Test): 1; Falling (Fall): 2; Ending (End): 3

(Example) If the time elapsed (or the remaining time) is 7.0 s and the test is under way,
`7.0,1` is returned.

MON

Inquires about each monitor value

Program message

- Syntax

  Query message: `MON?`

Response message

To MON?, the current monitor value is returned.

When no test is under way, the previous test results are returned.

Returning in AC withstanding voltage testing

Test type (0: ACW), monitor voltage, monitor current NORM, monitor current MAX, real current, imaginary current, elapsed (remaining) time, test status (0: Rise/ 1: Test/2: Fall/3: End)

Returning in DC withstanding voltage testing

Test type (1: DCW), monitor voltage, monitor current NORM, monitor current MAX, monitor resistance MIN, elapsed (remaining) time, test status (0: Rise/1: Test/3: End)

Returning in insulation resistance testing

Test type (2: IR), monitor voltage, monitor resistance NORM, monitor resistance MIN, monitor current , 0, elapsed (remaining) time, test status (0: Rise/1: Test/3: End)

(Example) If an AC withstanding voltage test is under way at a monitor voltage of 2.50 kV, monitor current NORM of 5.2 mA, monitor current MAX of 12 mA, real current of 5 mA, imaginary current of 10 μA, and remaining time of 2.6 s,
"`0,2.5E3,5.2E-3,12E-3,5E-3,10E-6,2.6,1`" is returned.

NOTE

- In insulation resistance testing with the lower judgment function off, even if settings are made so that the test voltage divided by the lower resistance exceeds 1.1 mA, testing can be performed. However, when the voltage value which is out of the range of ±(10 % of setting +50 V) is detected during the test, the measurement voltage value flashes on the LCD.

  In this case, -1 is returned.
- In insulation resistance testing, when the resistance value to exceed 99.9 GΩ is detected, "99.9" flashes on the LCD.

  In this case, -1 is returned.

SCANTYPE (STYP)

Inquires about the number of valid connected channels and the model name of the scanner

Program message

- Syntax

  Query message: `SCANTYPE?`
  `STYP?`

Response message

To STYP?, the number of valid channels and the model name of the scanner are returned as follows:

Number of valid channels, scanner 1, scanner 2, scanner 3, scanner 4

0: None; 1: TOS9220; 2: TOS9221

(Example) If there are 12 valid channels and scanners TOS9220, TOS9220, and TOS9221 (in that order, from the nearest to the tester) are connected without a fourth scanner connected,
"`12,1,1,2,0`" is returned.

CFAILCH (CFCH)

Inquires about the channel in the C FAIL status

Program message

- Syntax

  Query message: `CFAILCH?`
  `CFCH?`

Response message

To CFCH?, the channel in the CONTACT FAIL status is returned.

(Example) If channel 12 is currently in the CONTACT FAIL status, `12` is returned.

3.6 System-Related Messages

This section describes the messages to be set on the system screen.

MEASMODE (MMOD)

Sets the display mode (MEAS MODE) of the measured current (or measured resistance). Also inquires about the display mode of the measured current (or measured resistance).

Program message

- Syntax

  Command message: `MEASMODE_<{NORM|MAX}>`
  `MMOD_<{NORM|MAX}>`

  Query message: `MEASMODE?`
  `MMOD?`

- Program data

  Data format: Character

  Set value: NORM: Real Time; MAX: Max Hold (Min Hold)

(Example) To set the display mode of the measured current (or resistance) to Max Hold (Min Hold for resistance),

```
MMOD MAX
```

Response message

To MMOD?, the present display mode of the measured current (or resistance) is returned as 0 for NORM and 1 for MAX.

(Example) If the display mode of the present measured current is set to Real Time, `0` is returned.

PASSHOLD (PHOL)

Sets the PASS hold time in a PASS judgment. Also inquires about the current PASS hold time.

Program message

- Syntax

  Command message: `PASSHOLD_<`*`set value`*`>`
  `PHOL_<`*`set value`*`>`

  Query message: `PASSHOLD?`
  `PHOL?`

- Program data

Data format:	Real number, character
Set value:	0.2 to 10.0, HOLD
Unit:	s

(Example) To set the PASS hold time to HOLD,

```
PHOL HOLD
```

Response message

To PHOL?, the current PASS hold time is returned.

(Example) If the current PASS hold time is 0.5 s,
`0.5` is returned.

BUZZERVOL (BVOL)

Set the buzzer volume. Also inquires about the current buzzer volume.

Program message

- Syntax

  Command message: `BUZZERVOL_<`*`set value`*`>`
  `BVOL_<`*`set value`*`>`

  Query message: `BUZZERVOL?`
  `BVOL?`

- Program data

Data format:	Integer
Set value:	0 to 10
Resolution:	1

(Example) To set the buzzer volume to 5,

```
BVOL 5
```

Response message

To BVOL?, the current volume is returned.

(Example) If the current buzzer volume is 3,
3 is returned.

CONTRAST (CON)

Sets the LCD contrast. Also inquires about the current LCD contrast.

Program message

- Syntax

  Command message: `CONTRAST_<`*`set value`*`>`
  `CON_<`*`set value`*`>`

  Query message: `CONTRAST?`
  `CON?`

- Program data

Data format:	Integer
Set value:	0 to 10
Resolution:	1

(Example) To set the LCD contrast to 5,

```
CON 5
```

Response message

To CON?, the current LCD contrast is returned.

(Example) If the current LCD contrast is 6,
`6` is returned.

MOMENTARY (MOM)

Sets the start momentary function (MOMENTARY). Also inquires about the current start momentary setting.

Program message

- Syntax

  Command message: `MOMENTARY_<{ON|OFF}>`
  `MOM_<{ON|OFF}>`

  Query message: `MOMENTARY?`
  `MOM?`

- Program data

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the start momentary function to OFF,

`MOM OFF`

Response message

To MOM?, the current start momentary setting is returned.

(Example) If the start momentary function is currently ON,
`1` is returned.

FAILMODE (FMOD)

Sets the FAIL mode (FAIL MODE). Also inquires about the current FAIL mode setting.

Program message

- Syntax

Command message:	`FAILMODE_<{ON|OFF}>`
	`FMOD_<{ON|OFF}>`

Query message:	`FAILMODE?` `FMOD?`

- Program data

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the FAIL mode to OFF,

`FMOD OFF`

Response message

To FMOD?, the current FAIL mode setting is returned.

(Example) If the FAIL mode function is currently ON,
`1` is returned.

DOUBLE ACTION (DAC)

Sets the double-action mode (DOUBLE ACTION). Also inquires about the current double-action mode setting.

Program message

- Syntax

Command message:	`DOUBLEACTION_<{ON|OFF}>`
	`DAC_<{ON|OFF}>`

Query message:	`DOUBLEACTION?` `DAC?`

- Program data

Data format:	Character

Set value: OFF (0), ON (1)

(Example) To set the double-action mode to OFF,

```
DAC OFF
```

Response message

To DAC?, the current double-action mode setting is returned.

(Example) If the double-action mode is currently ON,
`1` is returned.

SIGHVON (SHV)

Sets H.V ON of Status Signal Output. Also inquires about the H.V ON setting.

Program message

- Syntax

Command message: `SIGHVON_<{ON|OFF}>`
`SHV_<{ON|OFF}>`

Query message: `SIGHVON?`
`SHV?`

- Program data

Data format: Character

Set value: OFF (0), ON (1)

(Example) To set H.V ON of Status Signal Output to OFF,

```
SHV OFF
```

Response message

To SHV?, the current H.V ON setting for Status Signal Output is returned.

(Example) If H.V ON for Status Signal Output is currently set to ON,
`1` is returned.

SIGTEST (STES)

Sets TEST for Status Signal Output. Also inquires about the current TEST setting.

Program message

- Syntax

Command message: `SIGTEST_<{ON|OFF}>`
`STES_<{ON|OFF}>`

Query message: `SIGTEST?`
`STES?`

- Program data

Data format: Character

Set value: OFF (0), ON (1)

(Example) To set TEST for Status Signal Output to OFF,

```
STES OFF
```

Response message

To STES?, the current TEST setting for Status Signal Output is returned.

(Example) If the TEST status of Status Signal Output is currently set to ON, 1 is returned.

SIGPASS (SPAS)

Sets PASS for Status Signal Output. Also inquires about the current PASS setting.

Program message

- Syntax

Command message:	`SIGPASS_<{ON|OFF}>`
	`SPAS_<{ON|OFF}>`

Query message:	`SIGPASS?` `SPAS?`

- Program data

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set the PASS status of Status Signal Output to OFF,

```
SPAS OFF
```

Response message

To SPAS?, the current PASS setting for Status Signal Output is returned.

(Example) If the PASS status of Status Signal Output is currently set to ON, 1 is returned.

SIGUFAIL (SUF)

Sets U FAIL for Status Signal Output. Also inquires about the current U FAIL setting.

Program message

- Syntax

Command message:	`SIGUFAIL_<{ON|OFF}>`
	`SUF_<{ON|OFF}>`

Query message:	`SIGUFAIL?` `SUF?`

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the U FAIL status of Status Signal Output to OFF,

```
SUF OFF
```

Response message

To SUF?, the current U FAIL setting for Status Signal Output is returned.

(Example) If the U FAIL status of Status Signal Output is currently set to ON, `1` is returned.

SIGLFAIL (SLF)

Sets L FAIL for Status Signal Output. Also inquires about the current LOW FAIL setting.

Program message

- Syntax

```
Command message:  SIGLFAIL_<{ON|OFF}>
                  SLF_<{ON|OFF}>
Query message:    SIGLFAIL?
                  SLF?
```

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the L FAIL status of Status Signal Output to OFF,

```
SLF OFF
```

Response message

To SLF?, the current L FAIL setting for Status Signal Output is returned.

(Example) If the L FAIL status of Status Signal Output is currently set to ON, `1` is returned.

SIGREADY (SREA)

Sets READY for Status Signal Output. Also inquires about the current READY setting.

Program message

- Syntax

```
Command message:  SIGREADY_<{ON|OFF}>
                  SREA_<{ON|OFF}>
```

Query message: `SIGREADY?`
`SREA?`

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the READY status of Status Signal Output to OFF,

```
SREA OFF
```

Response message

To SREA?, the current READY setting for Status Signal Output is returned.

(Example) If the READY status of Status Signal Output is currently set to ON, `1` is returned.

SIGPROTECTION (SPR)

Sets PROTECT for Status Signal Output. Also inquires about the current PROTECT setting.

Program message

- Syntax

  Command message: `SIGPROTECTION_<{ON|OFF}>`
  `SPR_<{ON|OFF}>`

  Query message: `SIGPROTECTION?`
  `SPR?`

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the PROTECT status of Status Signal Output to OFF,

```
SPR OFF
```

Response message

To SPR?, the current PROTECT setting for Status Signal Output is returned.

(Example) If the PROTECT status of Status Signal Output is currently set to ON, `1` is returned.

SIGPOWERON (SPOW)

Sets POWER ON for Status Signal Output. Also inquires about the current POWER ON setting.

Program message

- Syntax

  Command message: `SIGPOWERON_<{ON|OFF}>`
  `SPOW_<{ON|OFF}>`

  Query message: `SIGPOWERON?`
  `SPOW?`

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the POWER ON status of Status Signal Output to OFF,

```
SPOW OFF
```

Response message

To SPOW?, the current POWER ON setting for Status Signal Output is returned.

(Example) If the POWER ON status of Status Signal Output is currently set to ON, `1` is returned.

SIGCFAIL (SCF)

Sets C FAIL for Status Signal Output. Also inquires about the current C FAIL setting.

Program message

- Syntax

  Command message: `SIGCFAIL_<{ON|OFF}>`
  `SCF_<{ON|OFF}>`

  Query message: `SIGCFAIL?`
  `SCF?`

- Program data

  Data format: Character

  Set value: OFF (0), ON (1)

(Example) To set the C FAIL status of Status Signal Output to OFF,

```
SCF OFF
```

Response message

To SCF?, the current C FAIL setting for Status Signal Output is returned.

(Example) If the CONTACT FAIL status of Status Signal Output is currently set to ON,
`1` is returned.

COMMENT (COM)

Inputs a comment (memo). Also inquires about the current comment. Up to three lines can be entered, with a maximum of 12 characters per line.

Program message

- Syntax

  Command message: `COMMENT_<`*`"1st line","2nd line","3rd line"`*`>`

  `COM_<`*`"1st line","2nd line","3rd line"`*`>`

  Query message: `COMMENT?`

  `COM?`

- Program data

  Data format: Character

  Set value: ASCII Code 20H to 7EH

  (Except 22H (“), 27H (‘), 2CH (,), and 40H (@))

(Example) To enter “KIKUSUI” in the 1st line, a blank in the 2nd line, and “TOS9201” in the 3rd line,

```
COM “KIKUSUI”,“␣”,“TOS9201”
```

Response message

To COM?, the current comment is returned.

(Example) If “KIKUSUI” is entered in the 1st line and “123456789ABCDEFGHIJK” in the 3rd line, with no entry in the 2nd line, the following comment is returned (␣ = space):

```
KIKUSUI␣␣␣␣␣␣␣␣␣␣␣␣␣,␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣,
123456789ABCDEFGHIJK
```

3.7 Memory-Related Messages

This section explains the messages used in connection with memory.

ACW:RECALL (A:REC)

Recalls the contents of memory stored in an AC withstanding voltage test

Program message

- Syntax

  Command message: `ACW:RECALL_<memory number>`
  `A:REC_<memory number>`

- Program data

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

(Example) To recall memory 10 for AC withstanding voltage testing,

```
A:REC 10
```

DCW:RECALL (D:REC)

Recalls the contents of memory stored in a DC withstanding voltage test

Program message

- Syntax

  Command message: `DCW:RECALL_<memory number>`
  `D:REC_<memory number>`

- Program data

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

(Example) To recall memory 10 for DC withstanding voltage testing,

```
D:REC 10
```

IR:RECALL (I:REC)

Recalls the contents of memory stored in an insulation resistance test

Program message

- Syntax

  Command message: `IR:RECALL_<`*`memory number`*`>`
  `I:REC_<`*`memory number`*`>`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1

(Example) To recall memory 10 for insulation resistance testing,

```
I:REC 10
```

ACW:STORE (A:STOR)

Stores the current settings for an AC withstanding voltage test. Note that the memory name is not stored (the existing memory name is not replaced).

Program message

- Syntax

  Command message: `ACW:STORE_<`*`memory number`*`>`
  `A:STOR_<`*`memory number`*`>`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1

(Example) To store the current settings for an AC withstanding voltage test to memory 10,

```
A:STOR 10
```

DCW:STORE (D:STOR)

Stores the current settings for a DC withstanding voltage test. Note that the memory name is not stored (the existing memory name is not replaced).

Program message

- Syntax

  Command message: `DCW:STORE_<memory number>`
  `D:STOR_<memory number>`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1

(Example) To store the current settings for a DC withstanding voltage test to memory 10,

```
D:STOR 10
```

IR:STORE (I:STOR)

Stores the current settings for an insulation resistance test. Note that the memory name is not stored (the existing memory name is not replaced).

Program message

- Syntax

  Command message: `IR:STORE_<memory number>`
  `I:STOR_<memory number>`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1

(Example) To store the current settings for an insulation resistance test to memory 10,

```
I:STOR 10
```

ACW:MEMORY (A:MEM)

Stores the test conditions for an AC withstanding voltage test to a specified memory number. Also inquires about the contents of the memory with the specified number.

Program message

- Syntax

  Command message: `ACW:MEMORY_<`*`memory number, memory name, test voltage, lower current, upper current, test time, frequency`*`{50|60},` *`lower judgment`* `{ON|OFF},` *`offset`*`{ON|OFF},` *`timer`*`{ON|OFF},` *`start voltage, voltage rise time, voltage fall time, voltage range`*`{0|1},` *`current-detection response speed`*`{0|1|2},` *`GND`*`{GUARD|LOW},` *`contact check`*`{ON|OFF},` *`scanner Hi/Low, scanner Open`*`>`

  Query message: `ACW:MEMORY?`
  `A:MEM?`

- Program data <*memory number*>

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

- Program data <*memory name*>

  Data format: Character
  Set value: ASCII Code 20H to 7EH (up to 12 characters)
  (Except 22H (“), 27H (‘), 2CH (,), and 40H (@))

- Program data <test voltage>

  Data format: Real number
  Set value: 0.00 to 5.20 E3
  Resolution: 0.01 E3
  Unit: V

- Program data <*lower current*>

  Data format: Real number
  Set value: 0.01 E-3 to 110 E-3
  Resolution: 0.01 E-3 (0.01 E3 to 9.99 E-3)
  0.1 E-3 (10.0 E-3 to 999 E-3)
  1 E-3 (100 E-3 to 110 E-3)
  Unit: A

- Program data <*upper current*>

  Data format: Real number
  Set value: 0.01 E-3 to 110 E-3
  Resolution: 0.01 E-3 (0.01 E3 to 9.99 E-3)
  0.1 E-3 (10.0 E-3 to 999 E-3)
  1 E-3 (100 E-3 to 110 E-3)

Unit:	A

- Program data <*test time*>

Data format:	Real number
Set value:	0.3 to 999
Resolution:	0.1 for 0.3 to 99.9, 1 for 100 to 999
Unit:	s

- Program data <*frequency*>

Data format:	Character
Set value:	50, 60
Unit:	Hz

- Program data <*lower judgment*{ON|OFF}>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <*offset*{ON|OFF}>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <*timer*{ON|OFF}>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <*start voltage*>

Data format:	Integer
Set value:	1 to 99
Resolution:	1
Unit:	%

- Program data <*voltage rise time*>

Data format:	Real number
Set value:	0.1 to 200
Resolution:	0.1 for 0.1 to 99.9, 1 for 100 to 200
Unit:	s

- Program data <*voltage fall time*>

Data format:	Real number
Set value:	0.0 to 200
Resolution:	0.1 for 0.1 to 99.9, 1 for 100 to 200
Unit:	s

- Program data <*test-voltage range*{0|1}>

Data format:	Integer
Set value:	0: AUTO; 1: 5 kV

- Program data <*current-detection response speed*{0|1|2}>

Data format:	Integer
Set value:	0: SLOW; 1: MID; 2: FAST

- Program data <*GND*{GUARD|LOW}>

  Data format: Character

  Set value: LOW (0), GUARD (1)

- Program data <*contact check*{ON|OFF}>

  Data format: Character

  Set value: OFF, ON (0, 1)

- Program data <*scanner Hi/Low*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  LOW is represented by 0 and HIGH by 1.

- Program data <*scanner Open*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  OPEN is represented by 1 and NOT OPEN by 0.

  OPEN has priority over HIGH/LOW.

Response message

To A:MEM? _<*memory number*>, the contents of memory specified by memory number are returned.

The memory contents are returned in the same order as with program messages.

(Example) If memory 10, at inquiry, contains the memory name ACW10, test voltage of 5.00 kV, lower current of 5.00 mA, upper current of 80.0 mA, test time of 2.0 s, frequency of 50 Hz, lower judgment set to ON, offset set to OFF, timer set to ON, start voltage of 15%, voltage rise time of 0.2 s, voltage fall time of 0.4 s, voltage range of AUTO, current-detection response speed of SLOW, GND set to LOW, contact check set to ON, scanners 1 through 5 set to Low, scanners 6 through 10 set to High, and the other scanners set to Open, then,

"ACW10,5.00E3,5.00E-3,80.0E-3,2.0,50,1,0,1,15, 0.2,0.4,0,0,0,1,31,992" is returned.

DCW:MEMORY (D:MEM)

Stores the test conditions for a DC withstanding voltage test to a specified memory number. Also inquires about the contents of the memory with the specified number.

Program message

- Syntax

  Command message: `DCW:MEMORY (D: MEM)_<`*`memory number`*`,`
  `memory name``,` *`test voltage`*`,` *`lower current`*`,` *`upper`*
  `current``,` *`test time`*`,` *`lower judgment`*`{ON|OFF},`
  `timer``{ON|OFF},` *`start voltage`*`,` *`voltage rise time`*`,`
  `WAIT TIME``,` *`GND`*`{GUARD|LOW},` *`contact check`*`{ON|OFF},`
  `scanner Hi/Low``,` *`scanner Open`*`>`

  Query message: `DCW:MEMORY?`
  `D:MEM?`

- Program data <*memory number*>

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

- Program data <*memory name*>

  Data format: Character
  Set value: ASCII Code 20H to 7EH (up to 12 characters)
  (Except 22H ("), 27H ('), 2CH (,), and 40H (@))

- Program data <*test voltage*>

  Data format: Real number
  Set value: 0.00 to 6.00 E3
  Resolution: 0.01 E3
  Unit: V

- Program data <*lower current*>

  Data format: Real number
  Set value: 0.01 E-3 to 11.0 E-3
  Resolution: 0.01 for 0.01 E3 to 9.99 E-3, 0.1 for 10.0 E-3 to 11.0 E-3
  Unit: A

- Program data <*upper current*>

  Data format: Real number
  Set value: 0.01 E-3 to 11.0 E-3
  Resolution: 0.01 for 0.01 E3 to 9.99 E-3, 0.1 for 10.0 E-3 to 11.0 E-3
  Unit: A

- Program data <*test time*>

  Data format: Real number
  Set value: 0.3 to 999
  Resolution: 0.1 for 0.3 to 99.9, 1 for 100 to 999
  Unit: s

- Program data <*lower judgment*{ON|OFF}>

  Data format: Character

  Set value: OFF, ON (0, 1)

- Program data <*timer*{ON|OFF}>

  Data format: Character

  Set value: OFF, ON (0, 1)

- Program data <*start voltage*>

  Data format: Integer

  Set value: 0 to 99

  Resolution: 1

  Unit: %

- Program data <*voltage rise time*>

  Data format: Real number

  Set value: 0.1 to 200

  Resolution: 0.1 for 0.1 to 99.9, 1 for 100 to 200

  Unit: s

- Program data <*WAIT TIME*>

  Data format: Real number

  Set value: 0.3 to 10.0

  Resolution: 0.1

  Unit: s

- Program data <*GND*{GUARD|LOW}>

  Data format: Character

  Set value: LOW (0), GUARD (1)

- Program data <*contact check*{ON|OFF}>

  Data format: Character

  Set value: OFF, ON (0, 1)

- Program data <*scanner Hi/Low*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  LOW is represented by 0 and HIGH by 1.

- Program data <*scanner Open*>

  Data format: Integer

  Set value: 0 to 65535 (#HFFFF)

  Resolution: 1

  OPEN is represented by 1 and NOT OPEN by 0.

  OPEN has priority over HIGH/LOW.

Response message

To D:MEM? _<*memory number*>, the contents of the memory specified by memory number are returned. The memory contents are returned in the same order as with program messages.

(Example) If memory 12, at inquiry, contains the memory name DCW12, test voltage of 5.50 kV, lower current of 5.00 mA, upper current of 10.0 mA, test time of 2.0 s., lower judgment set to ON, timer set to ON, start voltage of 20%, voltage rise time of 0.2 s, WAIT TIME of 0.5 s, GND set to LOW, contact check set to ON, scanners 1 through 5 set to Low, scanners 6 through 10 set to High, and the other scanners set to Open, then, "`DCW12,5.50E3,5.00E-3,10.0E-3,2.0,1,1,20,0.2, 0.5,0,1,31,992`" is returned.

IR:MEMORY (I:MEM)

Stores the test conditions for an insulation resistance test to a specified memory number. Also inquires about the contents of the memory with the specified number.

Program message

- Syntax

  Command message: `IR:MEMORY (I: MEM)_<`*memory number, memory name, test voltage, lower resistance, upper resistance, test time, lower judgment*`{ON|OFF},` *upper judgment*`{ON|OFF},` *timer*`{ON|OFF},` *voltage rise time, WAIT TIME, GND*`{GUARD|LOW},` *contact check*`{ON|OFF},` *scanner Hi/Low, scanner Open*`>`

  Query message: `IR:MEMORY?`
  `I:MEM?`

- Program data <*memory number*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

- Program data <*memory name*>

Data format:	Character
Set value:	ASCII Code 20H to 7EH (up to 12 characters)

  (Except 22H ("), 27H ('), 2CH (,), and 40H (@))

- Program data <*test voltage*>

Data format:	Real number
Set value:	10 to 1020
Resolution:	1
Unit:	V

- Program data <`lower resistance`>

Data format:	Real number
Set value:	0.01 E6 to 9.99 E9
Resolution:	0.01 E6 (0.01 E6 to 9.99 E6)
	0.1 E6 (10.0 E6 to 99.9 E6)
	1 E6 (100 E6 to 999 E6)
	0.01 E9 (1.00 E9 to 9.99 E9)
Unit:	Ω

- Program data <`upper resistance`>

Data format:	Real number
Set value:	0.01 E6 to 9.99 E9
Resolution:	0.01 E6 (0.01 E6 to 9.99 E6)
	0.1 E6 (10.0 E6 to 99.9 E6)
	1 E6 (100 E6 to 999 E6)
	0.01 E9 (1.00 E9 to 9.99 E9)
Unit:	Ω

- Program data <test time>

Data format:	Real number
Set value:	0.5 to 999
Resolution:	0.1 for 0.5 to 99.9, 1 for 100 to 999
Unit:	s

- Program data <`lower judgment{ON|OFF}`>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <`upper judgment{ON|OFF}`>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <`timer{ON|OFF}`>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <`voltage rise time`>

Data format:	Real number
Set value:	0.1 to 200
Resolution:	0.1 for 0.1 to 99.9, 1 for 100 to 200
Unit:	s

- Program data <`WAIT TIME`>

Data format:	Real number
Set value:	0.3 to 10.0
Resolution:	0.1
Unit:	s

- Program data <*GND*{GUARD|LOW}>

Data format:	Character
Set value:	LOW (0), GUARD (1)

- Program data <*contact check*{ON|OFF}>

Data format:	Character
Set value:	OFF, ON (0, 1)

- Program data <*scanner Hi/Low*>

Data format:	Integer
Set value:	0 to 65535 (#HFFFF)
Resolution:	1

  LOW is represented by 0 and HIGH by 1.

- Program data <scanner Open>

Data format:	Integer
Set value:	0 to 65535 (#HFFFF)
Resolution:	1

  OPEN is represented by 1 and NOT OPEN by 0.

  OPEN has priority over HIGH/LOW.

Response message

To I:MEM? _<*memory number*>, the contents of the memory specified by memory number are returned.

The memory contents are returned in the same order as with program messages.

(Example) If memory 15, at inquiry, contains the memory number IR15, test voltage of 50 V, lower resistance of 0.05 MΩ, upper resistance of 10.0 MΩ, test time of 2.0 s., lower judgment set to ON, upper judgment set to OFF, timer set to ON, voltage rise time of 0.2 s, WAIT TIME of 0.5 s, LOW terminal set to GND, contact check set to ON, scanners 1 through 5 set to Low, scanners 6 through 10 set to High, and the other scanners set to Open, then "`IR15,50,0.05E6,10.0E6,2.1,0,1,0.2,0.5,0,1,31,992`" is returned.

3.8 Program-Related Messages

This section describes the messages used in connection with the program (AUTO).

PRGNAME (PNAM)

Specifies a program number and names the program. Also inquires about the program name.

Program message

- Syntax

Command message:	`PRGNAME_<program number,program name>`
	`PNAM_<program number,program name>`

Query message:	`PRGNAME?_<program number>`
	`PNAM?_<program number>`

- Program data <*program number*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

- Program data <*program name*>

Data format:	Character
Set value:	ASCII Code 20H to 7EH (up to 12 characters) (Except 22H (“), 27H (‘), 2CH (,), and 40H (@))

(Example) To name program 10 “TEST 10,”

```
PNAM 10,“TEST 10”
```

Response message

To PNAM?_<*program number*>, the name of the program with the specified number is returned.

(Example) If the name of program 10 is TEST10, “`TEST10`” is returned.

PRGTEST (PTES)

Sets the specified program.

Program message

- Syntax

  Command message: `PRGTEST_<`*`program number`*`>`
  `PTES_<`*`program number`*`>`

- Program data

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

(Example) To set program 10,

```
PTES 10
```

PRGEDIT (PED)

Creates a program using stored memory. Also inquires about the contents of the program corresponding to a program number and step number.

Following execution of this message, the program edit screen automatically appears.

Program message

- Syntax

  Command message: `PRGEDIT_<`*`program number, step number, mode selection`*` {0|1|2}>, `*`memory number, interval time`*`>`
  `PED_<`*`program number, step number, mode selection`*` {0|1|2}, `*`memory number, interval time`*`>`

  Query message: `PRGEDIT? (PED?)_<`*`program number, step number`*`>`
  `PED?_<`*`program number, step number`*`>`

- Program data <*`program number`*>

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

- Program data <*`step number`*>

  Data format: Integer
  Set value: 0 to 99
  Resolution: 1

- Program data <*mode selection* {0|1|2}>

Data format:	Integer
Set value:	0: ACW; 1: DCW; 2: IR

- Program data <*memory number*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

- Program data <*interval time*>

Data format:	Real number, character
Set value:	0.2 to 9.9, HOLD
Resolution:	0.1
Unit:	s

NOTE

- All parameters must be entered, even if the value isn't changed.
- Do not skip a step number. Overwrite the existing step, or add a new step after the last step.
- Up to 500 steps can be used in a set of programs.
  For example, if 100 steps are set for each of programs 0 to 4, no step is available for program 5.

(Example) At step 15 of program 10, to program memory 3 for AC withstanding voltage testing at an interval time of 2 s,

```
PED 10,15,0,3,2.0
```

Response message

To PED?_<*program number, step number*>, the contents of the specified step of a specified program are returned.

(Example) If, at step 15 of program 10, memory 3 for AC withstanding voltage testing is programmed at an interval time of 2 s,
"0,3,2" is returned.

PRGRETURN (PRET)

Makes the RETURN/END settings for a specified program.

Also inquires about the RETURN/END setting for a specified program.

Program message

- Syntax

  Command message: `PRGRETURN_<program number,{ON|OFF}>`
  `PRET_<program number,{ON|OFF}>`

  Query message: `PRGRETURN?_<program number>`
  `PRET?_<program number>`

- Program data <*program number*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

- Program data <*program name*>

Data format:	Character
Set value:	OFF (0), ON (1)

(Example) To set program 10 to RETURN,

```
PRET 10,ON
```

Response message

To PRET?_<*program number*>, the RETURN setting for the specified program is returned.

(Example) If program 10 is set to the END mode,
`0` is returned.

PRGNEW (PNEW)

Deletes a specified program.

Program message

- Syntax

  Command message: `PRGNEW_<program number>`
  `PNEW_<program number>`

- Program data

Data format:	Integer
Set value:	0 to 99
Resolution:	1

(Example) To delete program 10,

```
PNEW 10
```

PRGTOTAL (PTOT)

Inquires about the total number of steps in a specified program

Program message

- Syntax

  Query message: `PRGTORAL?_<`*`program number`*`>`
  `PTOT?_<`*`program number`*`>`

- Program data <*`program number`*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

Response message

To PRET?_<*`program number`*>, the total number of steps in a specified program is returned.

(Example) If the total number of step in program 10 is 55,
`55` is returned.

PRGINS (PIN)

Inserts a memory number into the specified step of a specified program. Following the execution of this message, the tester automatically displays the program edit screen.

Program message

- Syntax

  Command message: `PRGINS_<`*`program number`*`, `*`step number`*`,`
  `mode selection``{0|1|2}, `*`memory number`*`>`
  `PIN_<`*`program number`*`, `*`step number`*`,`
  `mode selection``{0|1|2}, `*`memory number`*`>`

- Program data <*`program number`*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

- Program data <*`step number`*>

Data format:	Integer
Set value:	0 to 99
Resolution:	1

- Program data <*mode selection* {0|1|2}>

  Data format: Integer

  Set value: 0: ACW; 1: DCW; 2: IR

- Program data <*memory number*>

  Data format: Integer

  Set value: 0 to 99

  Resolution: 1

NOTE

- Do not skip a step number. Overwrite the existing step or add a new step after the last step.
- Up to 500 steps can be used in a set of programs. For example, if 100 steps are set for each of programs 0 to 4, no step is available for program 5.

(Example) To insert memory 3 for AC withstanding voltage testing into step 15 of program 10,

```
PIN 10,15,0,3
```

PRGDEL (PDEL)

Deletes the specified step of a specified program

Program message

- Syntax

  Command message: PRGDEL_<*program number*, *step number*>
  PDEL_<*program number*, *step number*>

- Program data <*program number*>

  Data format: Integer

  Set value: 0 to 99

  Resolution: 1

- Program data <*step number*>

  Data format: Integer

  Set value: 0 to 99

  Resolution: 1

(Example) To delete step 15 from program 10,

```
PDEL 10,15
```

AUTORUNNING (ARUN)

Inquires which step the program is currently executing

Program message

- Syntax

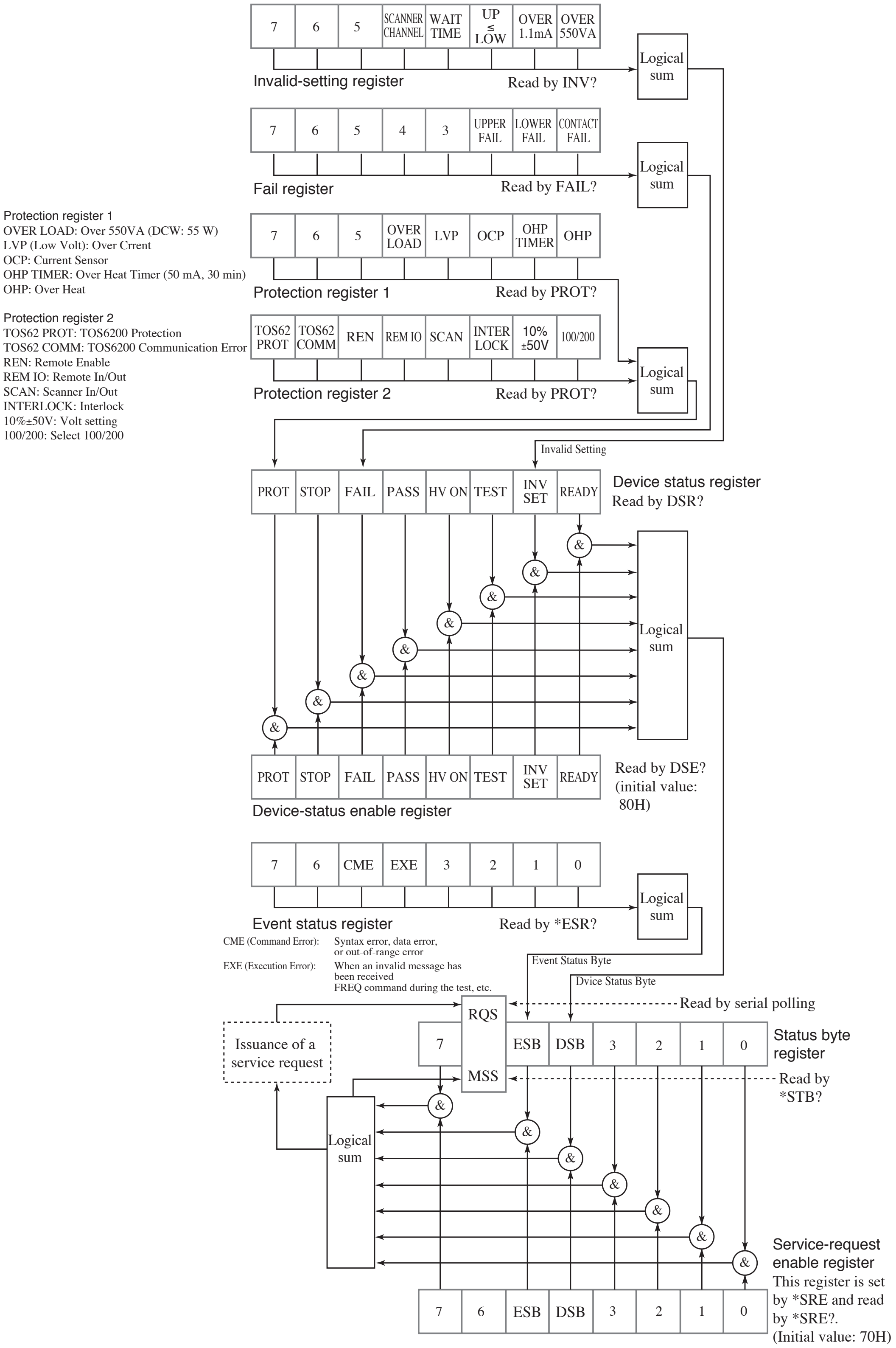
  Query message: `AUTORUNNING?`
  `ARUN?`

Response message

To ARUN?, the step currently being executed by the program is returned.

(Example) If step 55 of the program is being executed,
`55` is returned.

3.9 Registers



Protection register 1
OVER LOAD: Over 550VA (DCW: 55 W)
LVP (Low Volt): Over Crrent
OCP: Current Sensor
OHP TIMER: Over Heat Timer (50 mA, 30 min)
OHP: Over Heat

Protection register 2
TOS62 PROT: TOS6200 Protection
TOS62 COMM: TOS6200 Communication Error
REN: Remote Enable
REM IO: Remote In/Out
SCAN: Scanner In/Out
INTERLOCK: Interlock
10%±50V: Volt setting
100/200: Select 100/200

CME (Command Error): Syntax error, data error, or out-of-range error

EXE (Execution Error): When an invalid message has been received FREQ command during the test, etc.

Fig.3-4 Structure of status data

NOTE

- In a command message in each register, hexadecimal data can be used by adding “#H”.
- In a query message in each register, all response data is composed of decimals.
- For each bit of each register, 1 represents “set” and 0 represents “reset.”
- The contents of the enable register are not backed up.

Status Byte register, Service-Request Enable register

Bit		Description
7		Not used with this tester
6	"RQS (Request)	Confirms that a service request has been issued. Reset by being read by series polling.
	"MSS (Master Summary Status)"	Logical sum of the status byte register and the service-request enable register. Read by *STB?
5	"ESB (Standard Event Status Bit)"	Indicates that a bit of the event status register has been set
4	"DSB (Device Status Bit)"	Indicates that a bit of the device status register has been set
3		Not used with this tester
2		Not used with this tester
1		Not used with this tester
0		Not used with this tester

Table 3-1 Status Byte register and Service-Request Enable register

Event Status register

Bit		Description
7		Not used with this tester
6		Not used with this tester
5	CME (Command Error)	Indicates a syntax error, data error, or our-of-range error
4	EXE (Execution Error)	Indicates that an invalid message has been received during the test or in the protection status
3		Not used with this tester
2		Not used with this tester
1		Not used with this tester
0		Not used with this tester

Table 3-2 Event Status register

Device Status register, Device-Status Enable register

Bit		Description
7	PROT (Protection)	Protection status
6	STOP	Suspension of the test
5	FAIL	FAIL judgment
4	PASS	PASS judgment
3	HV ON	Waiting for a contact check and outputting a voltage
2	TEST	Outputting a test voltage
1	INV SET (Invalid setting)	LCD displaying an invalid message such as “UP≤LOW” or “OVER 550 VA” (“OVER 55 W” in DC)
0	READY	READY status

Table 3-3 Device Status register and Device-Status Enable register

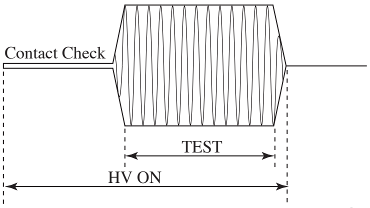


Fig.3-5 Relation of TEST (Bit 2) to HV ON (Bit 3).

Protection register 1

Bit		Description
7		Not used with this tester
6		Not used with this tester
5		Not used with this tester
4	OVER LOAD	When 550 VA is exceeded in ACW (55 W in DCW)
3	LVP (Low Voltage Protection)	When 1.1 mA is exceeded in IR
2	OCP (24 V)	When a 24 V overcurrent has been detected in the SIGNAL I/O connector or INTERLOCK connector
1	OHP TIMER (Over Heat Timer)	When a current exceeding 50 mA has been detected for over 30 minutes during the test
0	OHP (Over Heat Protection)	When the internal temperature has increased abnormally

Table 3-4 Protection register

Protection register 2

Bit		Description
7	TOS6200 PROT (Protection)	When the TOS6200 is in the protection status
6	TOS6200 COMM (Communication Error)	When an error has occurred in communication with the TOS6200
5	REN (Remote Enable)	Change in the Enable signal for SIGNAL I/O
4	REM IO (Remote In/Out)	When the remote-control connector has been connected/disconnected
3	SCAN (Scanner In/Out)	When the scanner connector has been connected/disconnected, or when a channel specified at a step during the AUTO test is disconnected.
2	INTERLOCK	In the INTERLOCK status
1	10 %±50 V (Setting 10 %±50 V)	When the test voltage has deviated from 10% ±50 V
0	100/200 (Select 100/200)	When a source voltage outside the preset range has been applied

Table 3-5 Protection register

Fail register

Bit		Description
7		Not used with this tester
6		Not used with this tester
5		Not used with this tester
4		Not used with this tester
3		Not used with this tester
2	UPPER FAIL	Indicates FAIL for the upper cutoff value
1	LOWER FAIL	Indicates FAIL for the lower cutoff value
0	CONTACT FAIL	Indicates FAIL in a contact check

Table 3-6 Fail register

Invalid-Setting register

Bit		Description
7		Not used with this tester
6		Not used with this tester
5		Not used with this tester
4	SCANNER CHANNEL	When an unconnected channel has been specified
3	WAIT TIME	At rise time + test time ≤ wait time
2	UP≤LOW	When the lower cutoff value exceeds the upper cutoff value
1	OVER 1.1mA	In IR, when the test voltage divided by the lower resistance has exceeded 1.1 mA
0	OVER 550VA	In ACW, when the test voltage multiplied by the upper current has exceeded 550 VA (55 W in DCW)

Table 3-7 Invalid-Setting register

Error register

Bit		Description
7		Not used with this tester
6		Not used with this tester
5		Not used with this tester
4		Not used with this tester
3	Invalid message	Indicates an invalid message
2	Out-of-range error	Indicates an out-of-range error
1	Data Error	Indicates a data error
0	Syntax Error	Indicates a header error

Table 3-8 Error register

3.10 Message List

The symbols in parentheses in the Header column represent abbreviated messages. R/W represents a query message (R) or command message (W).

Note 1: The commands that are valid during a rise time, test time, fall time, pass judgment, fail judgment, or interval time are marked with “①”, and invalid commands are indicated by “②”. The commands valid only in the ready status or interval hold status are marked with “③”.

Note 2: The commands valid in the protection status are marked with “①”, and the invalid commands are marked with “②”.

3.10.1 Register-related messages and general messages

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
1	*RST	W				--	Initializes devices. (Resets to factory settings, except for the INTERFACE screen settings)	①	①
2	*IDN?	R				--	Returns “KIKUSUI ELECTRONICS CORP., TOS9200, 0, x.xx” (TOS9200).	①	①
3	ERR?	R				--	Returns an error value and clears it.	①	①
4	TRM	W	0	3		--	Sets the terminator. 0: CR LF+EOI; 1: LF+EOI; 2: EOI; 3: CR+EOI	②	②
5	TRM?	R				--	Returns the preset terminator value.	①	①
6	*CLS	W				--	Clears the status byte register, event status register, device status register, protection register 1, protection register 2, fail register, invalid-setting register, and error register.	①	①
7	*STB?	R				--	Returns the status-byte-register value.	①	①
8	*SRE	W	0	255		--	Sets the service-request enable register.	②	②
9	*SRE?	R				--	Returns the service-request-enable-register value.	①	①
10	*ESR?	R				--	Returns the event-status-register value and clears it.	①	①
11	DSE	W	0	255		--	Sets the device-status enable register.	②	②
12	DSE?	R				--	Returns the device-status-enable-register value.	①	①
13	DSR?	R				--	Returns the device-status-register value.	①	①
14	PROTECTION? (PROT?)	R				--	Returns the values of protection registers 1 and 2.	①	①
15	FAIL?	R				--	Returns the fail-register value.	①	①
16	INVALID? (INV?)	R				--	Returns the invalid-setting-register value.	①	①
17	CLR	W				--	Clears all registers and sets a Stop flag (same as DCL and SDC).	①	①
18	SILENT (SIL)	W	0	1	0/1	--	Acknowledge message (RS-232C)	②	②
19	SILENT? (SIL?)	R				--	Returns the acknowledge-message (RS-232C) value.	①	①
20	START (STAR)	W				--	Starts a test. (valid in the ready status and the interval hold status)	③	②
21	STOP	W				--	Suspends the test and cancels the protection status.	①	①
22	FUNCTION (FUN)	W	0	4	1	--	Shifts between modes. 0: ACW; 1: DCW; 2: IR; 3: AUTO TEST; 4: AUTO EDIT; 5: SYSTEM; 6: OFFSET ADJ	②	②

23	FUNCTION?
	(FUN?)
		R				--	Returns the current mode.
							0: ACW; 1: DCW; 2: IR; 3: AUTO TEST;
							4: AUTO EDIT; 5: SYSTEM; 6: OFFSET ADJ;
							7: INTERFACE
								①	①

24	LOCAL (LOC)	W					Return to the LOCAL from the REMOTE state.	②	①

3.10.2 Messages for AC withstanding voltage testing

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
25	ACW:TESTV (A:TES)	W	0.00	5.20 E3	0.01 E3	V	Sets the test voltage.	①	②
26	ACW:TESTV? (A:TES?)	R	0.00 E3	5.20 E3		V	Returns the test-voltage setting.	①	①
27	ACW:FREQUENCY (A:FREQ)	W	50	60	50/60	Hz	Sets the output frequency.	②	②
28	ACW:FREQUENCY? (A:FREQ?)	R				Hz	Returns the output-frequency setting.	①	①
29	ACW:LOWER (A:LOW)	W	0.01 E-3	110 E-3	0.01, 0.1, 1	A	Lower current (0.01 to 9.99/10.0 to 99.9/100 to 110)	②	②
			OFF	ON	OFF/ON	--	Turns ON/OFF the low judgment function.		
30	ACW:LOWER? (A:LOW?)	R				A, --	Returns the lower current and the ON/OFF status of the judgment function. Returns ON/OFF as 1 or 0.	①	①
31	ACW:UPPER (A:UPP)	W	0.01 E-3	110 E-3	0.01, 0.1, 1	A	Upper current (0.01 to 9.99/10.0 to 99.9/100 to 110)	②	②
32	ACW:UPPER? (A:UPP?)	R				A	Returns the upper current.	①	①
33	ACW:TIMER (A:TIM)	W	0.3	999	0.1, 1	s	Timer set value (0.3 to 99.9/100 to 999)	②	②
			OFF	ON	0/1	--	Timer function		
34	ACW:TIMER? (A:TIM?)	R				s, --	Returns the timer ON/OFF setting. Returns ON/OFF as 1 or 0.	①	①
35	ACW:OFFSET (A:OFF)	W	OFF	ON	0/1	--	Offset function.	②	②
36	ACW:OFFSET? (A:OFF?)	R				--	Returns the ON/OFF status of the offset function Returns ON/OFF as 1 or 0.	①	①
37	ACW:STARTV (A:STAR)	W	0	99	1	%	Sets the start voltage.	②	②
38	ACW:STARTV? (A:STAR?)	R				%	Returns the start-voltage setting.	①	①
39	ACW:RISETIME (A:RTIM)	W	0.1	200	0.1, 1	s	Sets the voltage rise time (0.1 to 99.9/100 to 200).	②	②
40	ACW:RISETIME? (A:RTIM?)	R				s	Returns the voltage-rise-time setting.	①	①
41	ACW:FALLTIME (A:FTIM)	W	0	200	0.1, 1	s	Sets the voltage fall time (0.1 to 99.9/100 to 200).	②	②
42	ACW:FALLTIME? (A:FTIM?)	R				s	Returns the voltage-fall-time setting.	①	①
43	ACW:VRANGE (A:VRAN)	W	AUTO	5kV	0/1	--	Sets VRANGE (0: AUTO; 1: 5 kV).	②	②
44	ACW:VRANGE? (A:VRAN?)	R	0	1	0/1	--	Returns the VRANGE setting (0: AUTO; 1: 5 kV).	①	①
45	ACW:RESPONSE (A:RES)	W	SLOW	FAST	0/1/2	--	Sets the response filter (0: SLOW; 1: MID; 2: FAST).	②	②
46	ACW:RESPONSE? (A:RES?)	R				--	Returns the response-filter setting. (0: SLOW; 1: MID; 2: FAST)	①	①
47	ACW:GND (A:GND)	W	LOW	GUARD	0/1	--	1 to set GND to GUARD, and 0 to set to LOW.	②	②
48	ACW:GND? (A:GND?)	R				--	Returns the GUARD/LOW setting for GND as 1 or 0.	①	①
49	ACW:SCAN (A:SCAN)	W	1	16	1	--	Sets channels 1 through 16 to Hi/Low/Open (0: Open; 1: Low; 2: High).	②	②
			H	L	H/L/O	--			
50	ACW:SCAN? (A:SCAN?)	R	1	16	1	--	Returns the Hi/Low/Open setting for channels 1 through 16 (0: Open; 1: Low; 2: High).	①	①
51	ACW:SCANW (A:SCANW)	W	0	65535 #HFFFF	1	--	The 1st data represents channels 1 through 16 with 0: Low and 1: Hi, using 16 bits. The 2nd data represents Open and Not Open, with 1 for Open. The Open setting for the 2nd data has priority over the 1st data.	②	②
			0	65535 #HFFFF	1	--			

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
52	ACW:SCANW? (A:SCANW?)	R				--	Returns a WORD-type value for Hi/Low and Open/Not Open.	①	①
53	ACW:CONTACT CHECK (A:CCH)	W	OFF	ON	0/1	--	Sets ON/OFF for the scanner contact check function.	②	②
54	ACW:CONTACT CHECK?(A:CCH?)	R	0	1	0/1	--	Returns the ON/OFF status of the scanner contact check function.	①	①

3.10.3 Messages for DC withstanding voltage testing

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
55	DCW:TESTV (D:TES)	W	0.00	6.00 E3	0.01 E3	V	Sets the test voltage.	①	②
56	DCW:TESTV? (D:TES?)	R	0.00 E3	6.00 E3		V	Returns the test-voltage setting.	①	①
57	DCW:LOWER (D:LOW)	W	0.01 E-3	11.0 E-3	0.01,0.1	A	Lower current (0.01 to 9.99/10.0 to 11.0)	②	②
			OFF	ON	OFF/ON	--	Turns the low judgment function ON/OFF.		
58	DCW:LOWER? (D:LOW?)	R				A, --	Returns the lower current and the ON/OFF status of the judgment function. Returns ON/OFF as 1 or 0.	①	①
59	DCW:UPPER (D:UPP)	W	0.01 E-3	11.0 E-3	0.01,0.1	A	Upper current (0.01 to 9.99/10.0 to 11.0)	②	②
60	DCW:UPPER? (D:UPP?)	R				A	Returns the upper current.	①	①
61	DCW:TIMER (D:TIM)	W	0.1	999	0.1, 1	s	Timer set value (0.3 to 99.9/100 to 999)	②	②
			OFF	ON	0/1	--	Timer function		
62	DCW:TIMER? (D:TIM?)	R				s, --	Returns the timer ON/OFF setting. Returns ON/OFF as 1 or 0.	①	①
63	DCW:STARTV (D:STAR)	W	0	99	1	%	Sets the start voltage.	②	②
64	DCW:STARTV? (D:STAR?)	R				%	Returns the start-voltage setting.	①	①
65	DCW:RISETIME (D:RTIM)	W	0.1	200	0.1, 1	s	Sets the voltage rise time (0.1 to 99.9/100 to 200)	②	②
66	DCW:RISETIME? (D:RTIM?)	R				s	Returns the voltage-rise-time setting.	①	①
67	DCW:WAITTIME (D:WTIM)	W	0.3	10	0.1	s	Sets the WAIT TIME (0.3 to 10.0)	②	②
68	DCW:WAITTIME? (D:WTIM?)	R				s	Returns the judgment-wait-time setting.	①	①
69	DCW:GND(D:GND)	W	LOW	GUARD	0/1	--	1 to set GND to GUARD, and 0 to set to LOW	②	②
70	DCW:GND? (D:GND?)	R				--	Returns the GUARD/LOW setting for GND as 1 or 0.	①	①
71	DCW:SCAN (D:SCAN)	W	1	16	1	--	Sets channels 1 through 16 to Hi/Low/Open (0: Open; 1: Low; 2: High).	②	②
			H	L	H/L/O	--			
72	DCW:SCAN? (D:SCAN?)	R	1	16	1	--	Returns the Hi/Low/Open setting for channels 1 through 16 (0: Open; 1: Low; 2: High).	①	①
73	DCW:SCANW (D:SCANW)	W	0	65535 #HFFFF	1	--	The 1st data represents channels 1 through 16 with 0: Low and 1: Hi, using 16 bits. The 2nd data represents Open and Not Open, with 1 for Open. The Open setting for the 2nd data has priority over the 1st data.	②	②
			0	65535 #HFFFF	1	--			
74	DCW:SCANW? (D:SCANW?)	R				--	Returns a WORD-type value for Hi/Low and Open/Not Open.	①	①

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
75	DCW:CONTACT CHECK (D:CCH)	W	OFF	ON	0/1	--	Sets ON/OFF of the scanner contact check function.	②	②
76	DCW:CONTACT CHECK? (D:CCH?)	R	0	1	0/1	--	Returns the ON/OFF status of the scanner contact check function.	①	①

3.10.4 Messages for insulation resistance testing

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
77	IR:TESTV (I:TES)	W	10	1020	1	V	Sets the test voltage.	①	②
78	IR:TESTV? (I:TES?)	R				V	Returns the test-voltage setting.	①	①
79	IR:LOWER (I:LOW)	W	0.01 E6	9.99 E9	0.01	Ω	Lower resistance (0.01 M to 9.99 M, 10.0 M to 99.9 M, 100 M to 999 M, 1.00 G to 9.99 G)	②	②
			OFF	ON	OFF/ON				
80	IR:LOWER? (I:LOW?)	R				Ω	Returns the lower resistance. Uses MΩ as the unit. Returns the ON/OFF status of the judgment function. Returns ON/OFF as 1 or 0.	①	①
81	IR:UPPER (I:UPP)	W	0.01 E6	9.99 E9	0.01	Ω	Upper resistance (0.01 M to 9.99 M, 10.0 M to 99.9 M, 100 M to 999 M, 1.00 G to 9.99 G)	②	②
			OFF	ON	OFF/ON	--	Turns the upper judgment function ON/OFF.		
82	IR:UPPER? (I:UPP?)	R				Ω --	Returns the upper resistance (in MΩ) and the ON/OFF status of the upper judgment function. Returns ON/OFF as 1 or 0.	①	①
83	IR:TIMER (I:TIM)	W	0.5	999	0.1, 1	s	Timer set value (0.5 to 99.9/100 to 999)	②	②
			OFF	ON	0/1	--	Timer function		
84	IR:TIMER? (I:TIM?)	R				s, --	Returns the timer value and the ON/OFF status of the timer. Returns ON/OFF as 1 or 0.	①	①
85	IR:RISETIME (I:RTIM)	W	0.1	200	0.1, 1	s	Sets the voltage rise time (0.1 to 99.9/100 to 200).	②	②
86	IR:RISETIME? (I:RTIM?)	R				s	Returns the voltage-rise-time setting.	①	①
87	IR:WAITTIME (I:WTIM)	W	0.3	10	0.1	s	Sets the WAIT TIME (0.3 to 10.0).	②	②
88	IR:WAITTIME? (I:WTIM?)	R				s	Returns the judgment-wait-time setting.	①	①
89	IR:GND (I:GND)	W	LOW	GUARD	0/1	--	1 to set GND to GUARD, and 0 to set it to LOW	②	②
90	IR:GND? (I:GND?)	R				--	Returns the GUARD/LOW setting for GND as 1 or 0.	①	①
91	IR:SCAN (I:SCAN)	W	1	16	1	--	Sets channels 1 through 16 to Hi/Low/Open.	②	②
			H	L	H/L/O	--			
92	IR:SCAN? (I:SCAN?)	R	1	16	1	--	Returns the Hi/Low/Open settings for channels 1 through 16.	①	①
93	IR:SCANW (I:SCANW)	W	0	65535 #HFFFF	1	--	The 1st data represents channels 1 through 16 with 0: Low and 1: Hi, using 16 bits. The 2nd data represents Open and Not Open, with 1 for Open. The Open setting for the 2nd data has priority over the 1st data.	②	②
			0	65535 #HFFFF	1	--			
94	IR:SCANW? (I:SCANW?)	R				--	Returns a WORD-type value for Hi/Low and Open/Not Open.	①	①
95	IR:CONTACT CHECK (I:CCH)	W	OFF	ON	0/1	--	Sets ON/OFF for the scanner contact check function.	②	②
96	IR:CONTACT CHECK?(I:CCH?)	R	0	1	0/1	--	Returns the ON/OFF status of the scanner contact check function.	①	①

3.10.5 Messages common to all tests

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
97	VDATA? (VDAT?)	R				V	Monitor voltage (ACW: 0 E3 to 9.99 E3) (DCW: 0 E3 to 99.9 E3) (IR: 0 to 9999)	①	①
98	IDATA? (IDAT?)	R				A	Monitor current (0 to 999 E–6, 1.00 E–3 to 9.99 E–3, 10.0 E–3 to 99.9 E–3, 100 E–3 to 999 E–3)	①	①
99	RDATA? (RDAT?)	R				Ω	Monitor resistance (0.01 E6 to 9.99 E6, 10.0 E6 to 99.9 E6, 100 E6 to 999 E6, 1.00 E9 to 9.99 E9)	①	①
100	REALDATA? (REAL?)	R				A	Real current (0 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3)	①	①
101	IMAGDATA? (IMAG?)	R				A	Imaginary current (0 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3)	①	①
102	TIME?	R				s, --	Returns the elapsed (remaining) time (0 s to 99.9 s, 100 s to 999 s) and Rise/Test/Fall/End (0/1/2/3). Returns the elapsed time when the timer is off. Returns the remaining time when the timer is on.	①	①
103	MON?	R					Test type (0: ACW; 1: DCW; 2: IR) Monitor voltage (W: 0 E3 to 99.9 E3; IR: 0 to 9999) Monitor current Norm (0 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3) Monitor current MAX (0 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3) Monitor resistance Norm (0.01 E6 to 9.99 E6, 10.0 E6 to 99.9 E6, 100 E6 to 999 E6, 1.00 E9 to 9.99 E9) Monitor resistance Min (0.01 E6 to 9.99 E6, 10.0 E6 to 99.9 E6, 100 E6 to 999 E6, 1.00 E9 to 9.99 E9) Real current (0 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3) Imaginary current (0 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3) Monitor current IR (0 to 999 E-9, 1.00 E–6 to 9.99 E–6, 100 E–6 to 999 E–6, 1.00 E-3 to 9.99 E-3, 10.0 E-3 to 99.9 E-3, 100 E-3 to 999 E-3) Monitor resistance DCW (0.01 E6 to 9.99 E6, 10.0 E6 to 99.9 E6, 100 E6 to 999 E6, 1.00 E9 to 9.99 E9, 10.0 E9 to 99.9 E9, 100 E9 to 999 E9) Elapsed (remaining) time (0 s to 99.9 s, 100 s to 999 s) Separates Rise/Test/Fall/End (0/1/2/3) using "," in this order and then returns it. The results of the previous test are returned except during the test. Returns the monitor current in ACW and DCW, and the monitor resistance in IR.	①	①
104	SCANTYPE? (STYP?)	R				--	Returns the number of valid channels and the scanner type (Units 1 through 4). (0: none; 1: TOS9220; 2: TOS9221)	①	①
105	CFAILCH? (CFCH?)	R				--	Returns the channel number (1 through 16) in the CONTACT FAIL status.	①	①

3.10.6 System-related messages

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
106	MEASMODE (MMOD)	W	NORM	MAX		--	Sets the measured-current (resistance) display mode. NORM: Real Time; MAX: Max Hold	②	②
107	MEASMODE? (MMOD?)	R	0	1	0/1	--	Sets the measured-current (resistance) display mode. NORM: Real Time; MAX: Max Hold	①	①
108	PASSHOLD (PHOL)	W	0.2	10 HOLD	0.1	s	Pass hold time	②	②
109	PASSHOLD? (PHOL?)	R				s	Returns the pass hold time. 0.2 to 10.0 or HOLD	①	①
110	BUZZERVOL (BVOL)	W	0	10	1	--	Sets the buzzer volume.	①	①
111	BUZZERVOL? (BVOL?)	R				--	Returns the buzzer-volume setting.	①	①
112	CONTRAST (CON)	W	0	10	1	--	Sets the contrast level.	①	①
113	CONTRAST? (CON?)	R				--	Returns the contrast setting.	①	①
114	MOMENTARY (MOM)	W	OFF	ON	OFF/ON	--	Start momentary	②	②
115	MOMENTARY? (MOM?)	R				--	Returns the start-momentary setting. ON/OFF is returned as 1 or 0.	①	①
116	FAILMODE (FMOD)	W	OFF	ON	OFF/ON	--	Fail mode (The STOP key used to cancel the Fail and Protection statuses.)	②	②
117	FAILMODE? (FMOD?)	R				--	Returns the Fail-mode value. ON/OFF is returned as 1 or 0.	①	①
118	DOUBLEACTION (DAC)	W	OFF	ON	OFF/ON	--	Start double action	②	②
119	DOUBLEACTION? (DAC?)	R				--	Returns the start-double-action value. ON/OFF is returned as 1 or 0.	①	①
120	SIGHVON (SHV)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: HVON.	①	①
121	SIGHVON? (SHV?)	R				--	ON/OFF is returned as 1 or 0.	①	①
122	SIGTEST (STES)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: TEST.	①	①
123	SIGTEST? (STES?)	R				--	ON/OFF is returned as 1 or 0.	①	①
124	SIGPASS (SPAS)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: PASS.	①	①
125	SIGPASS? (SPAS)	R				--	ON/OFF is returned as 1 or 0.	①	①
126	SIGUFAIL (SUF)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: U FAIL.	①	①
127	SIGUFAIL? (SUF?)	R				--	ON/OFF is returned as 1 or 0.	①	①
128	SIGLFAIL (SLF)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: L FAIL.	①	①
129	SIGLFAIL? (SLF)	R				--	ON/OFF is returned as 1 or 0.	①	①
130	SIGREADY (SREA)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: READY.	①	①
131	SIGREADY? (SREA?)	R				--	ON/OFF is returned as 1 or 0.	①	①
132	SIGPROTECTION (SPR)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: PROTECTION.	①	①
133	SIGPROTECTION? (SPR?)	R				--	ON/OFF is returned as 1 or 0.	①	①
134	SIGCFAIL (SCF)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: CONTACT FAIL.	①	①

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
135	SIGCFAIL? (SCF?)	R				--	ON/OFF is returned as 1 or 0.	①	①
136	SIGPOWERON (SPOW)	W	OFF	ON	OFF/ON	--	Sets the value of Status Signal Output: POWER ON.	①	①
137	SIGPOWERON? (SPOW?)	R				--	ON/OFF is returned as 1 or 0.	①	①
138	COMMENT (COM)	W	20H	7EH		--	Comment (memo) 20 characters in the 1st line *1	②	②
			20H	7EH		--	Comment (memo) 20 characters in the 2nd line *1		
			20H	7EH		--	Comment (memo) 20 characters in the 3rd line *1		
			(“*1: 20H to 7EH are ASCII code. Except (“), (‘), (,), and (@))						
139	COMMENT? (COM?)	R				--,--, --	Returns a comment (memo). 1st line, 2nd line, 3rd line	①	①

3.10.7 Memory-related messages

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
140	ACW:RECALL (A:REC)	W	0	99	1	--	Recalls ACW memory.	②	②
141	DCW:RECALL (D:REC)	W	0	99	1	--	Recalls DCW memory.	②	②
142	IR:RECALL (I:REC)	W	0	99	1	--	Recalls IR memory.	②	②
143	ACW:STORE (A:STOR)	W	0	99	1	--	Stores the current settings in ACW memory (except for the memory name).	②	②
144	DCW:STORE (D:STOR)	W	0	99	1	--	Stores the current settings in DCW memory (except for the memory name).	②	②
145	IR:STORE (I:STOR)	W	0	99	1	--	Stores the current settings in IR memory (except for the memory name).	②	②
146	ACW:MEMORY (A:MEM)	W	0	99	1	--	Stores the next contents in the memory corresponding to an ACW memory number.	②	②
			20H	7EH		--	Memory name: 12 characters (20H to 7EH are ASCII code. Except (“), (‘), (,), and (@))		
			0	5.2 E3	0.01	V	Test voltage		
			0.01 E-3	110 E-3	0.01, 0.1, 1	A	Lower current		
			0.01 E-3	110 E-3		A	Upper current		
			0.3	999	0.1, 1	s	Test time		
			50	60	50/60	Hz	Frequency		
			OFF	ON	OFF/ON	--	Lower ON/OFF		
			OFF	ON	OFF/ON	--	Offset ON/OFF		
			OFF	ON	OFF/ON	--	Timer ON/OFF		
			0	99	1	%	Start-voltage/test-voltage ratio		
			0.1	200	0.1, 1	s	Voltage rise time		
			0	200	0.1, 1	s	Voltage fall time		
			0	1	0/1	--	V RANGE (0: AUTO; 1: 5 kV)		
			0	2	0/1/2	--	Response settings (0: SLOW; 1: MID; 2: FAST)		
			0	1	0/1	--	GND (0: LOW; 1: GUARD)		
			0	1	0/1	--	Contact Check		
			0	65535	1	--	Scanner Hi/Low (0 to 0xFFFF:16Bit, 0:Low, 1:Hi)		
			0	65535	1	--	Scanner Open (0 to 0xFFFF:16Bit, 0:Hi or Low, 1:Open)		

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
147	ACW:MEMORY? (A:MEM?)	R	0	99	1	--	Returns the contents of memory corresponding to the ACW memory number (memory name, test voltage, lower current, upper current, test time, frequency, lower ON/OFF, offset ON/OFF, timer ON/OFF, start-voltage/test-voltage ratio, voltage rise time, voltage fall time, V RANGE (0/1), response filter setting (0/1/2), GND, contact check, scanner Hi/Low, scanner Open). ON/OFF is returned as 1 or 0. Scanner Hi/Low and scanner Open are returned in decimal form using 16 bits.	①	①
148	DCW:MEMORY (D:MEM)	W	0	99	1	--	Stores the next contents in the memory corresponding to a DCW memory number.	②	②
			20H	7EH		--	Memory name: 12 characters (20H to 7EH are ASCII code. Except (“), (‘), (,), and (@))		
			0	6 E3	0.01	V	Test voltage		
			0.01 E-3	11 E-3	0.01,0.1	A	Lower current		
			0.01 E-3	11 E-3	0.01,0.1	A	Upper current		
			0.3	999	0.1, 1	s	Test time		
			OFF	ON	OFF/ON	--	Lower ON/OFF		
			OFF	ON	OFF/ON	--	Timer ON/OFF		
			0	99	1	%	Start-voltage/test-voltage ratio		
			0.1	200	0.1, 1	s	Voltage rise time		
			0.3	10	0.1	s	WAIT TIME		
			0	1	0/1	--	GND (0: LOW; 1: GUARD)		
			0	1	0/1	--	Contact Check		
			0	65535	1	--	Scanner Hi/Low (0 to 0xFFFF:16Bit, 0:Low, 1:Hi)		
			0	65535	1	--	"Scanner Open (0 to 0xFFFF:16Bit, 0:Hi or Low, 1:Open)"		
149	DCW:MEMORY? (D:MEM?)	R	0	99	1	--	Returns the contents of memory corresponding to the DCW memory number (memory name, test voltage, lower current, upper current, test time, lower ON/OFF, timer ON/OFF, start-voltage/test-voltage ratio, voltage rise time, WAIT TIME, response filter setting (0/1/2), GND, contact check, scanner Hi/Low, scanner Open). ON/OFF is returned as 1 or 0. Scanner Hi/Low and scanner Open are returned in decimal form using 16 bits.	①	①
150	IR:MEMORY (I:MEM)	W	0	99	1	--	Stores the next contents in the memory corresponding to an IR memory number.	②	②
			20H	7EH		--	Memory name: 12 characters (20H to 7EH are ASCII code. Except (“), (‘), (,), and (@))		
			10	1020	1	V	Test voltage		
			0.01 E6	9.99G E9	0.01, 0.1, 1	Ω, Ω	Lower current		
			0.01 E6	9.99 E9			Upper current		
			0.5	999	0.1, 1	s	Test time		
			OFF	ON	OFF/ON	--	Lower ON/OFF		
			OFF	ON	OFF/ON	--	Timer ON/OFF		
			OFF	ON	OFF/ON	--	Start-voltage/test-voltage ratio		
			0.1	200	0.1, 1	s	Voltage rise time		
			0.3	10	0.1	s	WAIT TIME		
			0	1	0/1	--	GND (0: LOW; 1: GUARD)		
			0	1	0/1	--	Contact Check		
			0	65535	1	--	Scanner Hi/Low (0 to 0xFFFF:16Bit, 0:Low, 1:Hi)		
			0	65535	1	--	"Scanner Open (0 to 0xFFFF:16Bit, 0:Hi or Low, 1:Open)"		

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
151	IR:MEMORY? (I:MEM?)	R	0	99	1	--	Returns the contents of the memory corresponding to an IR memory number (memory name, test voltage, lower resistance, upper resistance, test time, upper ON/OFF, timer ON/OFF, voltage rise time, WAIT TIME, GND, contact check, scanner Hi/Low, scanner Open). ON/OFF is returned as 1 or 0. Scanner Hi/Low and scanner Open are returned in decimal form using 16 bits.	①	①

3.10.8 Program-related messages

	Header	Data					Function and response data	A	B
		R/W	Min.	Max.	Resolution	Unit		Note 1	Note 2
152	PRGNAME (PNAM)	W	0	99	1	--	Program number	②	②
			20H	7EH		--	Program name: 12 characters (20H to 7EH are ASCII code. Except ("), ('), (,), and (@))		
153	PRGNAME? (PNAM?)	R	0	99	1		Returns the program name corresponding to a program number	①	①
154	PRGTEST (PTES)	W	0	99	1	--	Sets the program corresponding to a program number.	②	②
155	PRGEDIT (PED) Note 3	W	0	99	1	--	Program number	②	②
			0	99	1	--	Program step number		
			0	2	0/1/2	--	0: ACW, 1: DCW, 2: IR		
			0	99	1	--	Memory number		
			0.2	9.9, HOLD	0.1	s	Interval time Proceeds to the next step by pressing the START key when in the HOLD status.		
156	PRGEDIT? (PED?)	R	0	99	1	--	Returns the contents of a program corresponding to a program number and step number.	①	①
			0	99	1	--			
157	PRGRETURN (PRET)	W	0	99	1	--	Program number	②	②
			OFF	ON	OFF/ON	--	OFF: END (Ends the program) ON: RET (Returns to the first step)		
158	PRGRETURN? (PRET?)	R	0	99	1	--	Returns ON/OFF of the RETURN setting for the program corresponding to a program number.	①	①
159	PRGNEW (PNEW)	W	0	99	1	--	Clears the program corresponding to a program number.	②	②
160	PRGTOTAL? (PTOT?)	R	0	99	1	--	Returns the total number of steps in the program corresponding to a program number.	①	①
161	PRGINS (PIN) Note3	W	0	99	1	--	Program number	②	②
			0	99	1	--	Program step number		
			0	2	0/1/2	--	0: ACW, 1: DCW, 2: IR		
			0	99	1	--	Memory number		
162	PRGDEL (PDEL)	W	0	99	1	--	Program number	②	②
			0	99	1	--	Program step number		
163	AUTORUNNING? (ARUN?)	R					Returns the step currently being executed by a program.	①	①

Note 3: When PRGEDIT (PED) or PRGINS (PIN) is executed, the program EDIT screen automatically appears.

Appendix

Sample Program

The following provides some examples of remote programming using the GPIB or RS-232C interface.

These sample programs are created with Microsoft Visual Basic running on Windows 95/98/NT/2000, and use as a GPIB board a National Instruments NI-488.2M-compatible board or an Agilent Technologies HP-IB board. To run the program, the VISA (Virtual Instrument Software Architecture) library is required.

To use the VISA library on Visual Basic, follow the procedure specified below.

1. Obtain the VISA library.

   The VISA library may be included as a standard or optional feature with a GPIB/HP-IB board. It is also available on certain CD-ROMs such as LabVIEW. In addition, it can be downloaded at http://www.ni.com.

2. Install the VISA library.

3. Add VISA32.BAS and VPPTYPE.BAS to the Visual Basic project.

   These files can be found in c:\vxipnp\winnt\include folder ("winnt" may be changed in accordance with the OS).

The sample programs specified in this Appendix use GPIB. To create a program using the COM1 or COM2 port, convert the character strings that are passed to viOpen functions into "ASRL1" or "ASRL2."

Sample Program 1

This sample program runs a test after setting TOS9200 test conditions using GPIB and displays the test results in the message box. When the RS-232C is used, comments are used for commands.

```
Private Sub Command1_Click()
   'Writing style is based on VISA 1.20 or 2.01
   '------------------------------------------
   Dim vi As Long, tos As Long
   Dim vs As Long

   'Initialize VISA library                               'Initialize the VISA library.
   vs = viOpenDefaultRM(vi)
   'Open GPIB or COM1 port for tos                        'Open the port.
   vs = viOpen(vi, "GPIB::4", vbNull, 10, tos)           'Open address 4 of the GPIB port.
   'vs = viOpen(vi, "ASRL1", vbNull, 10, tos)             'To use the open RS-232C  of the
                                                         'COM1 port, delete the existing
                                                         'comment and use the GPIB lines as a
                                                         ' comment instead.
   'Sets attributes if RS232                              'For the RS-232C, make communication settings.
   Dim lIntfType As Long
   vs = viGetAttribute(tos, VI_ATTR_INTF_TYPE, lIntfType)
   If lIntfType = VI_INTF_ASRL Then
      vs = viSetAttribute(tos, VI_ATTR_ASRL_BAUD, 9600)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_PARITY, VI_ASRL_PAR_NONE)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_DATA_BITS, 8)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_STOP_BITS, VI_ASRL_STOP_TWO)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_FLOW_CNTRL, VI_ASRL_FLOW_XON_XOFF)
   End If

   Const DSR_READY = 1: Const DSR_INVSET = 2: Const DSR_TEST = 4: Const DSR_TESTON =8
                                                         'Define the device status register.
   Const DSR_PASS = 16: Const DSR_FAIL = 32: Const DSR_STOP = 64: Const DSR_PROTECTION = 128
   Const LOWER_FAIL = 2: Const UPPER_FAIL = 4            'Define the fail register.
   Const OHP = 1: Const OHPT = 2: Const LVP = 16: Const OVLD = 32
                                                         'Define protection register 1.
   Const LCK = 4: Const REN = 32                         'Define protection register 2.

   Dim r As Long, c As Long
   Dim strCommand As String, strRdBack As String, DSR As Integer, Result As String
   Dim PROT1 As Integer, PROT2 As Integer
   strRdBack = Space(255)

   'Device Clear
   vs = viClear&(tos)                                    'Clear the device used with GPIB.
   'strCommand = "SILENT 1" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'RS-232C command

   'FREQUENCY 50Hz,VOLT 0.05kV,UPPER 0.35mA,LOWER 0.01 ON,OFFSET OFF,TIMER 10sec ON
   '-----------------------------------------------------------------------------
   strCommand = "FUNCTION 0" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Make settings on the ACW screen.
   strCommand = "A:FREQ 50" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Test frequency: 50Hz
   strCommand = "A:TES 0.05E+3" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Test voltage: 0.05kV
   strCommand = "A:UPP 0.35E-3" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Upper current: 0.35mA
```

```
strCommand = "A:LOW 0.01E-3,OFF" + vbCrLf: vs = viWrite(tos, strCommand,
Len(strCommand), r)                                   'Lower current: 0.01mA, ON
strCommand = "A:OFF OFF" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                      'Offset function OFF
strCommand = "A:TIM 10,ON" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                      'Timer function: 10s, ON
strCommand = "PASSHOLD HOLD" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                      'Pass hold function: HOLD
strCommand = "DSE #HFF" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                      'Set the device-status enable
                                                      'register to FFH.

Do            'Check the device status register, wait until the READY status is ON.
  strCommand = "DSR?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
  vs = viRead(tos, strRdBack, 255, c)
  If Val(strRdBack) = DSR_READY Then Exit Do          'Exit the loop when the test is
Loop                                                  'ready.

strCommand = "START" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                      'Start the program because the READY
                                                      'status is ON.

Do          'Read the DSR value using the DSR? command. Read the measured value and other
            'data using the MON? command.
  strRdBack = Space(255)
  strCommand = "DSR?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
  vs = viRead(tos, strRdBack, 255, c)
  DSR = Val(strRdBack)                                'Retrieve the DSR value.
  strRdBack = Space(255)
  strCommand = "MON?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
  vs = viRead(tos, strRdBack, 255, c)
  Result = strRdBack                                  'Retrieve the measured value.
  If DSR = DSR_READY Or DSR = DSR_PASS Or DSR = DSR_FAIL Or DSR = DSR_STOP
  Or DSR = DSR_PROTECTION Then Exit Do                'Upon completion of the test, exit
Loop                                                  'the loop.

Select Case DSR                                       'Display the test results.
  Case DSR_STOP
    MsgBox ("USER CANCEL! " + Result)
                                                      'The STOP switch has been pressed
  Case DSR_READY                                      'during the test.
    MsgBox ("USER CANCEL! " + Result)
                                                      'The STOP switch has been pressed
  Case DSR_PASS                                       'during the test.
    MsgBox ("PASS! " + Result)
                                                      'Display the PASS test results.
  Case DSR_FAIL
    strRdBack = Space(255)
    strCommand = "FAIL?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                      'Read the contents of the fail
    vs = viRead(tos, strRdBack, 255, c)               'register, and determine whether the
                                                      'LOWER FAIL or UPPER FAIL test has
                                                      'been failed.
    If Val(strRdBack) = LOWER_FAIL Then MsgBox ("LOWER FAIL! " + Result)
                                                      'Display the LOWER FAIL test results.
    If Val(strRdBack) = UPPER_FAIL Then MsgBox ("UPPER FAIL! " + Result)
                                                      'Display the UPPER FAIL test results.
  Case DSR_PROTECTION
    strRdBack = Space(255)
```

```
         strCommand = "PROT?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Read the protection register, check
         vs = viRead(tos, strRdBack, 255, c)             'the cause of the protection status.
         PROT1 = Val(Left$(strRdBack, InStr(1, strRdBack, ",") - 1))
         PROT2 = Val(Mid$(strRdBack, InStr(1, strRdBack, ",") + 1))
         If PROT1 = OHP Then MsgBox ("OVER HEAT PROTECTION! " + Result)
                                                         'The overheat protection function has
                                                         'activated.
         If PROT1 = OHPT Then MsgBox ("OVER HEAT TIMER PROTECTION! " + Result)
                                                         'A current of 50 mA or more has been
                                                         'detected for 30 minutes.
         If PROT1 = LVP Then MsgBox ("OVER CURRENT PROTECTION! " + Result)
                                                         'The voltage of the main power supply
                                                         'has dropped.
         If PROT1 = OVLD Then MsgBox ("OVER LOAD PROTECTION! " + Result)
                                                         'The overload (500 VA) protection
                                                         'function has activated.
         If PROT2 = LCK Then MsgBox ("INTER LOCK PROTECTION! " + Result)
                                                         'There has been a change in the
                                                         'signal for SIGNAL I/O.
         If PROT2 = REN Then MsgBox ("SIGNAL I/O PROTECTION! " + Result)
                                                         'There has been a change in ENABLE
                                                         'for SIGNAL I/O.
      Case Else
         MsgBox ("ERROR!!")
   End Select
   strCommand = "STOP" & vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Check the test results, and press
                                                         'the STOP switch.
   'Closes the port
   vs = viClose(tos)                                     'Close the port.
   'Finalize VISA library
   vs = viClose(vi)                                      'Close the VISA library.

End Sub
```

Sample Program 2

This program sets test conditions in three memories of the TOS9200 using GPIB, then sets a memory number for the program memory and prepares for execution of an AUTO test.

```
Private Sub Command1_Click()
   'Writing style is based on VISA 1.20 or 2.01
   '-------------------------------------------
   Dim vi As Long, tos As Long
   Dim vs As Long

   'Initialize VISA library                              'Initialize the VISA library.
   vs = viOpenDefaultRM(vi)

   'Open GPIB or COM1 port for tos                       'Open the port.
   vs = viOpen(vi, "GPIB::4", vbNull, 10, tos)           'Open address 4 of the GPIB port.

   'vs = viOpen(vi, "ASRL1", vbNull, 10, tos)            'To use the open RS-232C of the COM1
                                                         'port, delete the existing comment
                                                         'and instead use the GPIB lines as a
                                                         'comment.

   'Sets attributes if RS232                             'For RS-232C, make communication
   Dim lIntfType As Long                                 'settings.
   vs = viGetAttribute(tos, VI_ATTR_INTF_TYPE, lIntfType)
   If lIntfType = VI_INTF_ASRL Then
     vs = viSetAttribute(tos, VI_ATTR_ASRL_BAUD, 9600)
     vs = viSetAttribute(tos, VI_ATTR_ASRL_PARITY, VI_ASRL_PAR_NONE)
     vs = viSetAttribute(tos, VI_ATTR_ASRL_DATA_BITS, 8)
     vs = viSetAttribute(tos, VI_ATTR_ASRL_STOP_BITS, VI_ASRL_STOP_TWO)
     vs = viSetAttribute(tos, VI_ATTR_ASRL_FLOW_CNTRL, VI_ASRL_FLOW_XON_XOFF)
   End If

   Dim r As Long, c As Long
   Dim strCommand As String

   'Device Clear
   vs = viClear&(tos)                                    'Clear the device used with the GPIB.
   'strCommand = "SILENT 1" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'RS-232C command

   strCommand = "FUNCTION 0" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Make settings on the ACW screen.

   strCommand = "A:MEM 20," + Chr$(34) + "TEST1" + Chr$(34) + ",1.05E3,0.1E-3,0.2E-3,10,50,
   0,0,1,0,0.1,0.0,0,0,0,0,0,65535" + vbCrLf             'A: Set test conditions in MEM20.
   vs = viWrite(tos, strCommand, Len(strCommand), r)
   strCommand = "A:MEM 21," + Chr$(34) + "TEST2" + Chr$(34) + ",1.15E3,0.1E-3,0.2E-3,10,50,
   0,0,1,0,0.1,0.0,0,0,0,0,0,65535" + vbCrLf             'A: Set test conditions in MEM21.
   vs = viWrite(tos, strCommand, Len(strCommand), r)
   strCommand = "A:MEM 22," + Chr$(34) + "TEST3" + Chr$(34) + ",1.25E3,0.1E-3,0.2E-3,10,50,
   0,0,1,0,0.1,0.0,0,0,0,0,0,65535" + vbCrLf             'A: Set test conditions in MEM22.
   vs = viWrite(tos, strCommand, Len(strCommand), r)

   strCommand = "FUNCTION 4" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Make settings on the AUTO EDIT
                                                         'screen.
```

```
  strCommand = "PRGNEW 10" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                        'Clear program 10.

  strCommand = "PRGNAME 10," + Chr$(34) + "TEST_SAMPLE" + Chr$(34) + vbCrLf
                                                        'Name program 10.
  vs = viWrite(tos, strCommand, Len(strCommand), r)

  strCommand = "PRGEDIT 10,0,0,20,0.5" + vbCrLf: vs = viWrite(tos, strCommand,
  Len(strCommand), r)                                   'Set MEM20, MEM21, and MEM22 in
                                                        'program 10.
  strCommand = "PRGEDIT 10,1,0,21,1.5" + vbCrLf: vs = viWrite(tos, strCommand,
  Len(strCommand), r)
  strCommand = "PRGEDIT 10,2,0,22,2.5" + vbCrLf: vs = viWrite(tos, strCommand,
  Len(strCommand), r)

  strCommand = "FUNCTION 3" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                        'Make settings on the AUTO TEST
                                                        'screen.
  'Closes the port
  vs = viClose(tos)                                     'Close the port.
  'Finalize VISA library
  vs = viClose(vi)                                      'Close the VISA library.

  MsgBox ("END")

End Sub
```

Sample Program 3

This sample program automatically runs Program 10 that is programmed on Sample Program 2 and displays the test results in the message box.

```
Private Sub Command1_Click()
   'Writing style is based on VISA 1.20 or 2.01
   '-------------------------------------------
   Dim vi As Long, tos As Long
   Dim vs As Long

   'Initialize VISA library                              'Initialize the VISA library.
   vs = viOpenDefaultRM(vi)

   'Open GPIB or COM1 port for tos                       'Open the port.
   vs = viOpen(vi, "GPIB::4", vbNull, 10, tos)           'Open address 4 of the GPIB port

   'vs = viOpen(vi, "ASRL1", vbNull, 10, tos)            'To use the open RS-232C of the COM1
                                                         'port, delete the existing comment
                                                         'and instead use GPIB lines as a
                                                         'comment.

   'Sets attributes if RS232                             'For RS-232C, make communication
                                                         'settings.
   Dim lIntfType As Long
   vs = viGetAttribute(tos, VI_ATTR_INTF_TYPE, lIntfType)
   If lIntfType = VI_INTF_ASRL Then
      vs = viSetAttribute(tos, VI_ATTR_ASRL_BAUD, 9600)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_PARITY, VI_ASRL_PAR_NONE)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_DATA_BITS, 8)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_STOP_BITS, VI_ASRL_STOP_TWO)
      vs = viSetAttribute(tos, VI_ATTR_ASRL_FLOW_CNTRL, VI_ASRL_FLOW_XON_XOFF)
   End If

   Const DSR_READY = 1: Const DSR_INVSET = 2: Const DSR_TEST = 4: Const DSR_TESTON = 8
                                                         'Define the device status register.
   Const DSR_PASS = 16: Const DSR_FAIL = 32: Const DSR_STOP = 64: Const DSR_PROTECTION = 128
                                                         'Define the fail register.
   Const OHP = 1: Const OHPT = 2: Const LVP = 16: Const OVLD = 32
                                                         'Define protection register 1.
   Const LCK = 4: Const REN = 32                         'Define protection register 2.

   Dim r As Long, c As Long
   Dim strCommand As String, strRdBack As String, DSR As Integer, Result As String
   Dim PROT1 As Integer, PROT2 As Integer
   strRdBack = Space(255)

   'Device Clear
   vs = viClear&(tos)                                    'Clear the device used with GPIB.
   'strCommand = "SILENT 1" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'RS-232C command

   strCommand = "FUNCTION 3" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Make settings on the AUTO TEST
                                                         'screen.
    strCommand = "PRGTEST 10" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                         'Recall No. 10.
```

```
 strCommand = "PASSHOLD HOLD" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                     'Set the pass hold function to HOLD.
 strCommand = "DSE #HFF" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                     'Set the device-status enable
                                                     'register to FFH.
strCommand = "PRGTOTAL? 10" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                     'Check the total number of steps in No.10.
vs = viRead(tos, strRdBack, 255, c)
If Val(strRdBack) = 0 Then                           'If the total number of steps is 0,
   MsgBox ("Not program! ")                          'end the program.
   GoTo Err
End If

Do            'Check the device status register and wait until the READY status is on.
   strCommand = "DSR?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
   vs = viRead(tos, strRdBack, 255, c)
   If Val(strRdBack) = DSR_READY Then Exit Do        'Exit the loop when the READY status
Loop                                                 'is on.

   strCommand = "START" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                     'Start the program because the READY
                                                     'status is on.

Do          'Read the DSR value using the DSR? command. Read the measured value and other
            'data using the MON? command.
   strRdBack = Space(255)
   strCommand = "DSR?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
   vs = viRead(tos, strRdBack, 255, c)
   DSR = Val(strRdBack)                              'Retrieve the DSR value.
   strRdBack = Space(255)
   strCommand = "MON?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
   vs = viRead(tos, strRdBack, 255, c)
   Result = strRdBack                                'Retrieve the measured value.
   If DSR = DSR_READY Or DSR = DSR_PASS Or DSR = DSR_FAIL Or DSR = DSR_STOP
   Or DSR = DSR_PROTECTION Then Exit Do              'Upon completion of the test, exit
Loop                                                 'the loop.

Select Case DSR                                      'Display the test results.
   Case DSR_STOP
      MsgBox ("USER CANCEL! " + Result)              'The STOP switch has been pressed
                                                     'during the test.
   Case DSR_READY
      MsgBox ("USER CANCEL! " + Result)              'The STOP switch has been pressed
                                                     'during the test.
   Case DSR_PASS
      MsgBox ("PASS! " + Result)                     'Display the PASS test results.
   Case DSR_FAIL
      strRdBack = Space(255)
      strCommand = "FAIL?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                     'Read the contents of the fail
      vs = viRead(tos, strRdBack, 255, c)            'register and determine whether the
                                                     'LOWER FAIL or UPPER FAIL test has
                                                     'been failed.
      If Val(strRdBack) = LOWER_FAIL Then MsgBox ("LOWER FAIL! " + Result)
                                                     'Display the LOWER FAIL test results.
      If Val(strRdBack) = UPPER_FAIL Then MsgBox ("UPPER FAIL! " + Result)
                                                     'Display the UPPER FAIL test results.
   Case DSR_PROTECTION
      strRdBack = Space(255)
```

```
        strCommand = "PROT?" + vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                          'Read the protection register and
        vs = viRead(tos, strRdBack, 255, c)               'check the cause of the protection status.
        PROT1 = Val(Left$(strRdBack, InStr(1, strRdBack, ",") - 1))
        PROT2 = Val(Mid$(strRdBack, InStr(1, strRdBack, ",") + 1))
        If PROT1 = OHP Then MsgBox ("OVER HEAT PROTECTION! " + Result)
                                                          'The overheat protection function has
                                                          'activated.
        If PROT1 = OHPT Then MsgBox ("OVER HEAT TIMER PROTECTION! " + Result)
                                                          'A current of 50 mA or more has been
                                                          'detected for 30 minutes.
        If PROT1 = LVP Then MsgBox ("OVER CURRENT PROTECTION! " + Result)
                                                          'The voltage of the main power supply
                                                          'has dropped.
        If PROT1 = OVLD Then MsgBox ("OVER LOAD PROTECTION! " + Result)
                                                          'The overload (500 VA) protection
                                                          'function has activated.
        If PROT2 = LCK Then MsgBox ("INTER LOCK PROTECTION! " + Result)
                                                          'There has been a change in the
                                                          'signal for SIGNAL I/O.
        If PROT2 = REN Then MsgBox ("SIGNAL I/O PROTECTION! " + Result)
                                                          'There has been a change in ENABLE
                                                          'for SIGNAL I/O.
     Case Else
        MsgBox ("ERROR!!")
     End Select
     strCommand = "STOP" & vbCrLf: vs = viWrite(tos, strCommand, Len(strCommand), r)
                                                          'Check the test results, and press
                                                          'the STOP switch.
Err:
     'Closes the port
     vs = viClose(tos)                                    'Close the port.
     'Finalize VISA library
     vs = viClose(vi)                                     'Close the VISA library.

End Sub
```

Index

TOS9200 GPIB/RS232-C INTERFACE   Operation Manual