BT6065
BT6075

HIOKI

Communication Command
Instruction Manual

PRECISION BATTERY TESTER

The latest edition of the instruction manual

EN

Apr. 2024 Edition 1
BT6065A981-00

- ✓ This manual explains the communication commands for Models BT6065 / BT6075 Precision Battery Tester.
- ✓ Please refer to the instruction manual for Models BT6065 / BT6075 for details regarding command settings.
- ✓ Although all reasonable care has been taken in the production of this instruction manual, should you find any points which are unclear or in error, please contact your local distributor or HIOKI's website.(https://www.hioki.com/contact)
- ✓ In the interest of product development, the contents of this manual may be subject to revision without notice.
- ✓

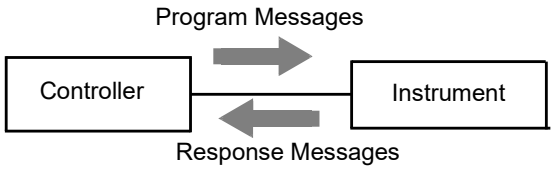
Contents

1 Introduction

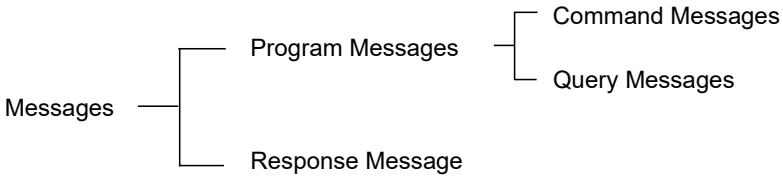
In this publication, the BT6065, BT6075 are indicated as “the instrument.”

If the communication monitoring function is used at the time of program creation, commands and responses will be conveniently displayed on the measurement screen. For information on the communication monitoring function, see the instruction manual of the instruments.

Various messages are supported for controlling the instrument through the interfaces.
Messages can be either program messages, sent from the controller such as PC to the instrument, or response messages, sent from the instrument to the controller.



Message types are further categorized as follows.



When issuing commands that contain data, make sure that the data is provided in the specified format.

Message Format

Program Messages

Program messages can be either Command Messages or Query Messages.

(1) Command Messages

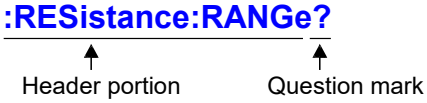
Instructions to control the instrument, such as to change settings or reset

Example: (instruction to set the measurement range)

(2) Query Messages

Requests for responses relating to results of operation or measurement, or the state of instrument settings

Example: (request for the current measurement range)



See: “Headers (p.3)”, “Separators (p.4)”, “Data Formats (p.5)”

Response Messages

When a query message is received, its syntax is checked and a response message is generated.
The **:SYSTem:COMMunicate:HEADer** command determines whether headers are prefixed to response messages.

Header ON	**:RESISTANCE:RANGE 300m**
Header OFF	**300m**
	(The current resistance measurement range is 300mΩ)

At power-on, Header OFF is selected.
If an error occurs when a query message is received, no response message is generated for that query.
Some query message has no header, such as
:FETCh? or **:COMParator:LIMit:RESistance:RESult?**.

Handshake Response

When a command message is received, its syntax is checked and response message is generated.
The **:SYSTem:COMMunicate:RESPonse** command determines whether the handshake message is generated.

Handshake ON	**:SYSTem:COMMunicate:RESPonse ON**
	OK (The handshake message)
Handshake OFF	**:SYSTem:COMMunicate:RESPonse OFF**
	(None)

When **:SYSTem:COMMunicate:RESPonse ON** is set,
sends a command message, and then the handshake message must be read by the controller before sending another command message.

Command Syntax

Command names are chosen to mnemonically represent their function, and can be abbreviated. The full command name is called the “long form”, and the abbreviated name is called the “short form”. The command references in this manual indicate the short form in upper-case letters, extended to the long form in lower case letters, although the commands are not case-sensitive in actual usage.

:FETCH?	OK (long form)
:FETC?	OK (short form)
:FET?	Error

Response messages generated by the instrument are in long form and in upper case letters.

Headers

Headers must always be prefixed to program messages.

(1) Command Program Headers

There are three types of commands: Simple, Compound and Standard.

- **Headers for Simple Commands**
  This header type is a sequence of letters and digits.
  :FUNCtion

- **Headers for Compound Commands**
  These headers consist of multiple simple command type headers separated by colons ":".
  :RESistance:RANGe

- **Headers for Standard Commands**
  This header type begins with an asterisk "*", indicating that it is a standard command defined by IEEE 488.2.
  ***RST**

(2) Query Program Header

These commands are used to interrogate the instrument about the results of operations, measured values and the current states of instrument settings.

As shown by the following examples, a query is formed by appending a question mark "?" after a program header.

:FETCh?
:RESistance:RANGe?

Characters within square brackets [ ] may be omitted.

:INITiate[:IMMediate] → Either form is valid
:INITiate:IMMediate
:INITiate

Message Terminators

This instrument recognizes the following message terminators (delimiters):

[RS-232C/USB/LAN]

- CR
- LF
- CR+LF

The following is selected as the terminator for response messages.

[RS-232C/USB/LAN]

- CR+LF

Separators

(1) Message Unit Separator

Multiple messages can be written in one line by separating them with semicolons ";".

:RESistance:RANGe 30m;*IDN?

- When messages are combined in this way and if one command contains an error, all subsequent messages up to the next terminator will be ignored.

(2) Header Separator

In a message consisting of both a header and data, the header is separated from the data by a space " " (ASCII code 20H).

:RESistance:RANGe 30m

(3) Data Separator

In a message containing multiple data items, commas are required to separate the data items from one another.

:SYSTem:COMMunicate:LAN:IPADdress 192,168,1,1

Data Formats

The instrument uses character data, decimal numeric data and character string data depending on the command.

(1) Character Data

Character data always begins with an alphabetic character, and subsequent characters may be either alphabetic or numeric. Character data is not case-sensitive, although response messages from the instrument are only upper case. When the command data portion contains <1/0/ON/OFF>, the operation will be similar to when 0 is OFF and 1 is ON.

:RESistance:RANGe:AUTO OFF

(2) Decimal Numeric Data

Three formats are used for numeric data, identified as NR1, NR2 and NR3. Numeric values may be signed or unsigned. Unsigned numeric values are handled as positive values. Values exceeding the precision handled by the instrument are rounded to the nearest valid digit.

- NR1 Integer data (e.g.: +12, -23, 34)
- NR2 Fixed-point data (e.g.: +1.23, -23.45, 3.456)
- NR3 Floating-point exponential representation data (e.g.: +1.0E-2, -2.3E+4)

The term “NRf format” includes all three of the above numeric decimal formats.
The instrument accepts NRf format data. The format of response data is specified for each command, and the data is sent in that format.

:STATus:OPERation:ENABle 33

:FETCh?
+1.00010E-03,+1.0000000E-06

Compound Command Header Omission

When several commands having a common header are combined to form a compound command (e.g., **:COMParator:LIMit:RESistance:UPPer** and **:COMParator:LIMit:RESistance:LOWer**) if they are written together in sequence, the common portion (here, **:COMParator:LIMit:RESistance:**) can be omitted after its initial occurrence.
This common portion is called the “current path” (analogous to the path concept in computer file storage), and until it is cleared, the interpretation of subsequent commands presumes that they share the same common portion.

This usage of the current path is shown in the following example:

Full expression

:COMParator:LIMit:RESistance:UPPer 0.28593;:COMParator:LIMit:RESistance:LOWer 0.28406

Compacted expression

:COMParator:LIMit:RESistance:UPPer 0.28593;LOWer 0.28406

This portion becomes the current path, and can be omitted from the messages immediately following.

The current path is cleared when the power is turned on, when reset by key input, by a colon “**:**” at the start of a command, and when a message terminator is detected.
Standard command messages can be executed regardless of the current path. They have no effect upon the current path.

A colon “**:**” is not required at the start of the header of a Simple or Compound command. However, to avoid confusion with abbreviated forms and operating mistakes, we recommend always placing a colon at the start of a header.

Output Queue and Input Buffer

Output Queue

Response messages are stored in the output queue until read by the controller. The output queue is also cleared in the following circumstances:

- Power on
- Query Error

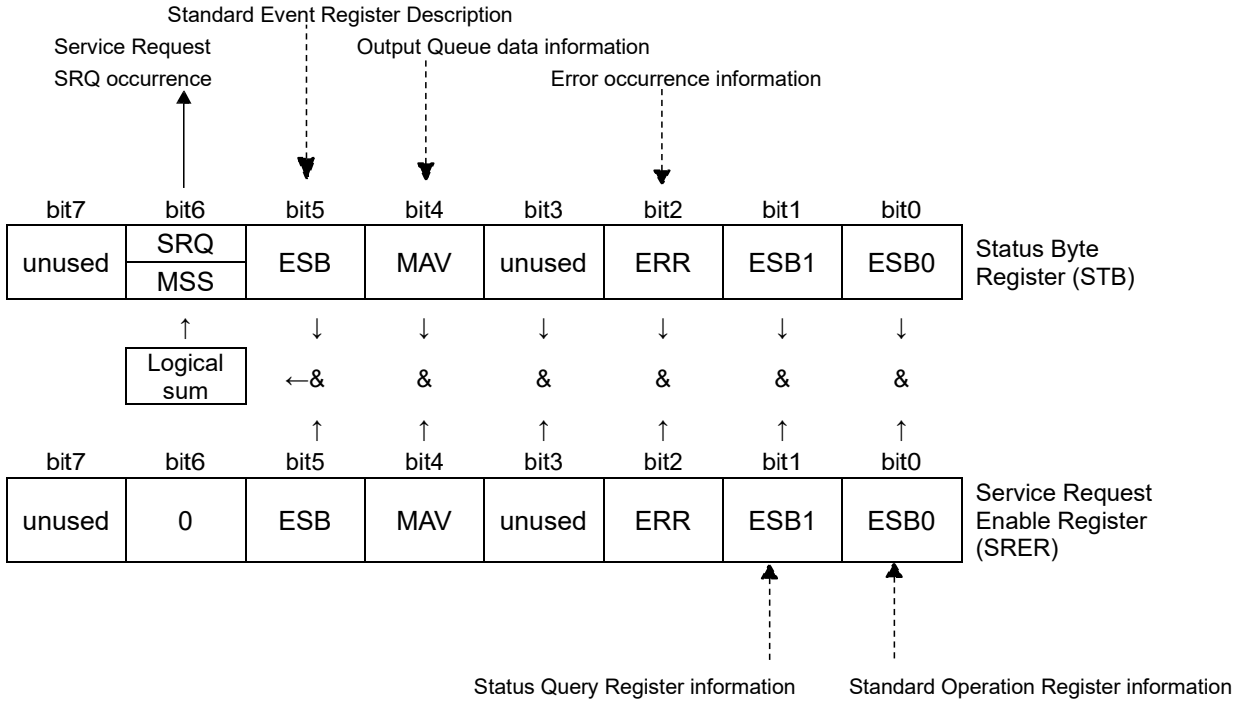
Input Buffer

The input buffer capacity of the instrument is 1460 bytes.
If 1460 bytes are allowed to accumulate in this buffer so that it becomes full, the interface bus enters the waiting state until space is cleared in the buffer.

Note: Ensure that the no command ever exceeds 1460 bytes.

Status Byte Register

This instrument implements the status model defined by IEEE 488.2 with regard to the serial poll function using the service request line. The term “event” refers to any occurrence that generates a service request.



Overview of Service Request Occurrence

The Status Byte Register contains information about the event registers and the output queue. Required items are selected from this information by masking with the Service Request Enable Register. When any bit selected by the mask is set, bit 6 (MSS; the Master Summary Status) of the Status Byte Register is also set, which generates an SRQ (Service Request) message and dispatches a service request.

Note: SRQ (Service Request) is a GP-IB function only. However, STB (Status Byte Register) information can be acquired with RS-232C/USB/LAN using the *STB? command.

[RS-232C/USB/LAN]

RS-232C/USB/LAN does not provide a function for issuing service requests.

Status Byte Register (STB)

During serial polling, the contents of the 8-bit Status Byte Register are sent from the instrument to the controller. When any Status Byte Register bit enabled by the Service Request Enable Register has switched from 0 to 1, the MSS bit becomes 1. Consequently, the SRQ bit is set to 1, and a service request is dispatched.

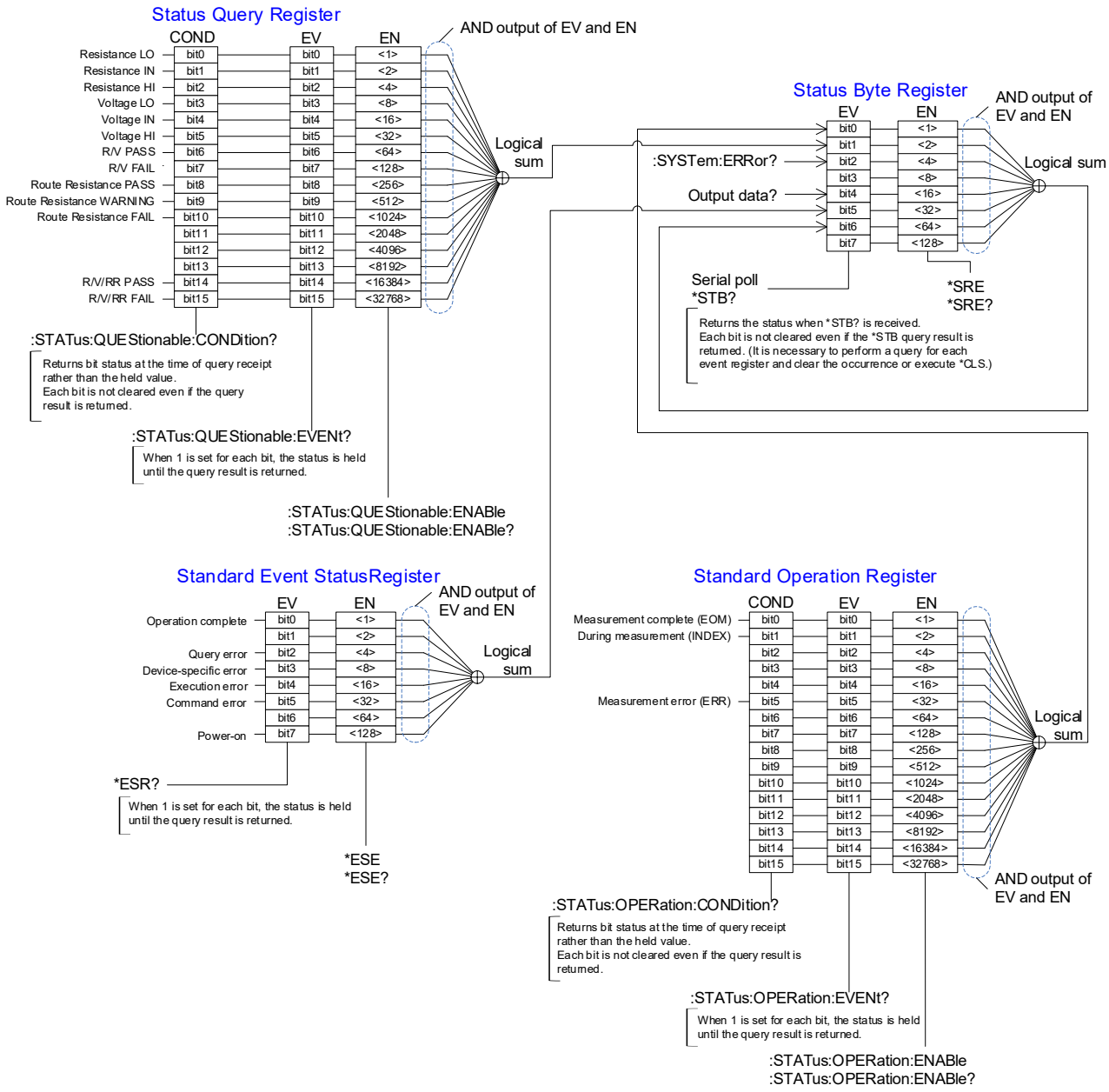
The SRQ bit is always synchronous with service requests, and is read and simultaneously cleared during serial polling. Although the MSS bit is only read by an ***STB?** query, it is not cleared until a clear event is initiated by the ***CLS** command.

Bit 7		unused
Bit 6	SRQ	Set to 1 when a service request is dispatched.
	MSS	This is the logical sum of the other bits of the Status Byte Register.
Bit 5	ESB	Standard Event Status (logical sum) bit This is logical sum of the Standard Event Status Register.
Bit 4	MAV	Message available Indicates that a message is present in the output queue.
Bit 3		unused
Bit 2	ERR	Error bit Set to 1 when an error information is present. Reset using **:SYSTem:ERRor?** to output error information.
Bit 1	ESB1	Event Status (logical sum) bit 1 This is the logical sum of the Status Query Register.
Bit 0	ESB0	Event Status (logical sum) bit 0 This is the logical sum of the Standard Operation Register.

Service Request Enable Register (SRER)

This register masks the Status Byte Register. Setting a bit of this register to 1 enables the corresponding bit of the Status Byte Register to be used.

Event Registers



Standard Event Status Register (SESR)

The Standard Event Status Register is an 8-bit register. If any bit in the Standard Event Status Register is set to 1 (after masking by the Standard Event Status Enable Register), bit 5 (ESB) of the Status Byte Register is set to 1.

See: “Standard Event Status Register (SESR) and Standard Event Status Enable Register (SESER)” (p.11)

The Standard Event Status Register is cleared in the following situations:

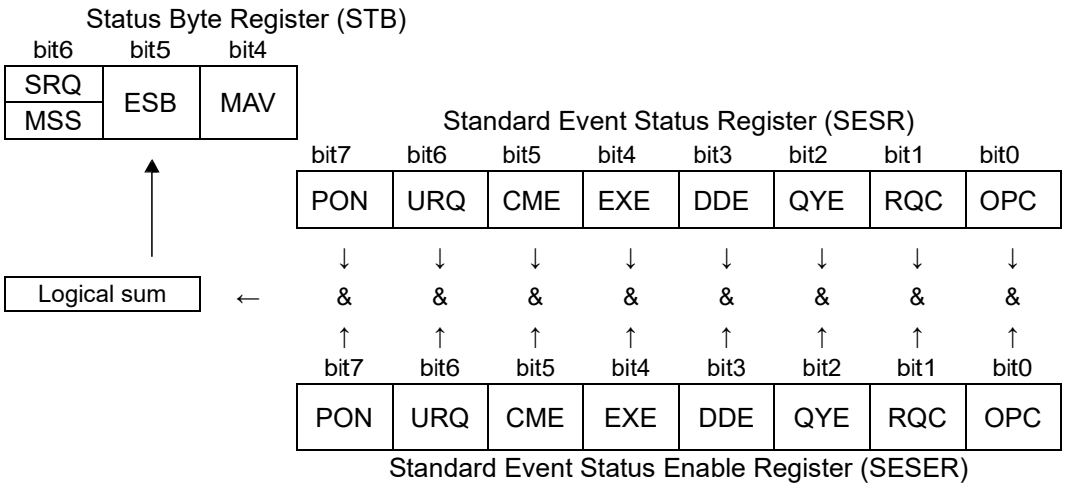
- When a ***CLS** command is executed
- When an event register query (***ESR?**) is executed
- When the instrument is powered on

Bit 7	PON	**Power-On Flag** Set to 1 when the power is turned on, or upon recovery from an outage.
Bit 6	URQ **(unused)**	**User Request** unused
Bit 5	CME	**Command error (The command to the message terminator is ignored.)** This bit is set to 1 when a received command contains a syntactic or semantic error: ・Program header error ・Incorrect number of data parameters ・Received a command not supported by the instrument
Bit 4	EXE	**Execution Error** This bit is set to 1 when a received command cannot be executed for some reason. ・Invalid parameter format ・The specified data value is outside of the set range ・The specified setting data cannot be set ・Execution is prevented by some other operation being performed
Bit 3	DDE	**Device-Dependent Error** This bit is set to 1 when a command cannot be executed due to some reason other than a command error, a query error or an execution error.
Bit 2	QYE	**Query Error (the output queue is cleared)** This bit is set to 1 when a query error is detected by the output queue control. ・When the data overflows the output queue ・When data in the output queue has been lost
Bit 1	RQC **(unused)**	**Request Control** unused
Bit 0	OPC	**Operation Complete** This bit is set to 1 in response to an “***OPC**” command. ・It indicates the completion of operations of all messages up to the “***OPC**” command

Standard Event Status Enable Register (SESER)

Setting any bit of the Standard Event Status Enable Register to 1 enables access to the corresponding bit of the Standard Event Status Register.

Standard Event Status Register (SESR) and Standard Event Status Enable Register (SESER)



Device-Specific Event Status Registers

This instrument provides two Event Status Registers for controlling events. Each event register is a 16-bit register. When any bit in one of these Event Status Registers enabled by its corresponding Enable Register is set to 1, the following happens:

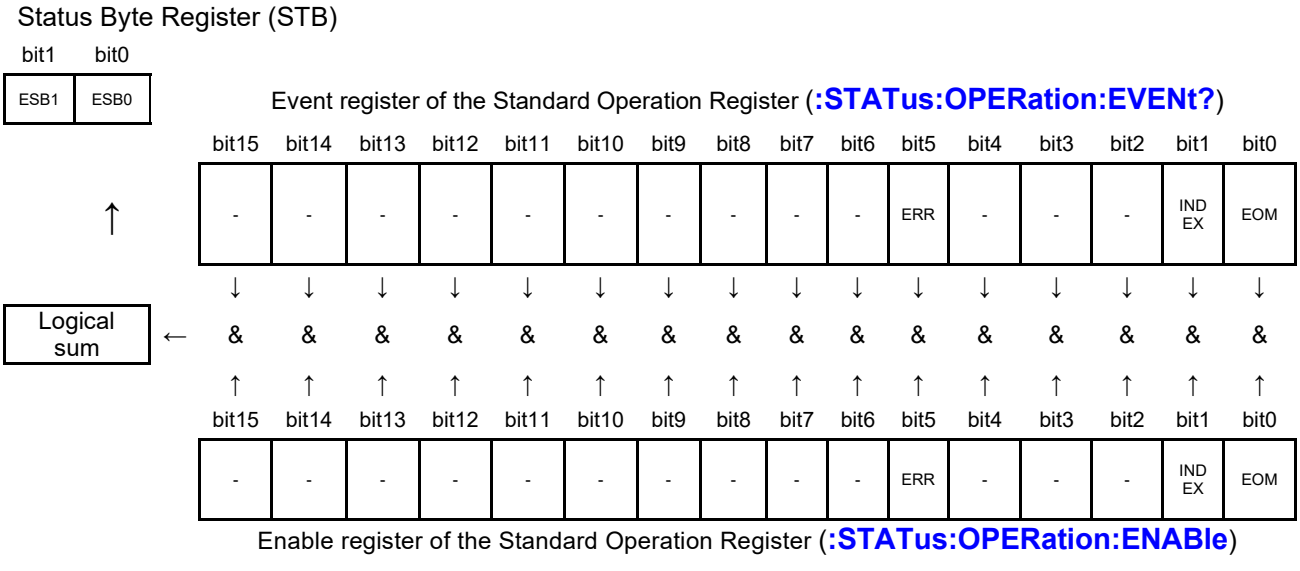
- For Standard Operation Register, bit 0 (ESB0) of the Status Byte Register (STB) is set to 1.
- For Status Query Register, bit 1 (ESB1) of the Status Byte Register (STB) is set to 1.

Device-Soecific Event Status Registers are cleared in the following situations:

- When a ***CLS** command is executed
- When an Event Status Register query is executed (**:STATus:OPERation:EVENt?** or **:STATus:QUEStionable:EVENt?**)
- When the instrument is powered on

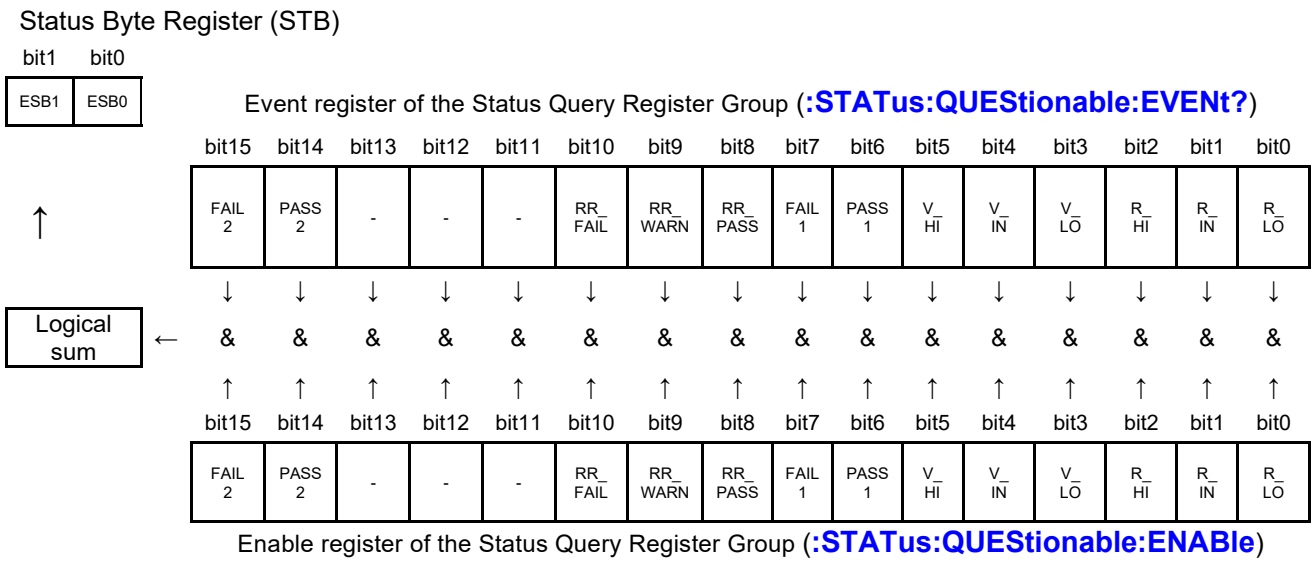
Standard Operation Register		
Bit 15	-	Unused
Bit 14	-	Unused
Bit 13	-	Unused
Bit 12	-	Unused
Bit 11	-	Unused
Bit 10	-	Unused
Bit 9	-	Unused
Bit 8	-	Unused
Bit 7	-	Unused
Bit 6	-	Unused
Bit 5	ERR	Set to 1 when an error occurs.
Bit 4	-	Unused
Bit 3	-	Unused
Bit 2	-	Unused
Bit 1	INDEX	Set to 1 when analog measurement is completed.
Bit 0	EOM	Set to 1 when measurement is completed.

Event register of the Standard Operation Register Group (**:STATus:OPERation:EVENt?**) and enable register of the Standard Operation Register Group (**:STATus:OPERation:ENABle**)



Status Query Register		
Bit 15	FAIL2	Set to 1 when PASS2 is not 1.
Bit 14	PASS2	Set to 1 when all resistance (IN), voltage (IN) and route-resistance (PASS or WARNING) comparator results are PASS.
Bit 13	-	Unused
Bit 12	-	Unused
Bit 11	-	Unused
Bit 10	RR_FAIL	Set to 1 when the route-resistance comparator result is upper threshold FAIL.
Bit 9	RR_WARN	Set to 1 when the route-resistance comparator result is not upper threshold FAIL but upper threshold WARNING.
Bit 8	RR_PASS	Set to 1 when the route-resistance comparator result is PASS.
Bit 7	FAIL1	Set to 1 when PASS1 is not 1.
Bit 6	PASS1	Set to 1 when both resistance (IN) and voltage (IN) comparator results are PASS.
Bit 5	V_HI	Set to 1 when the voltage comparator result is upper threshold FAIL.
Bit 4	V_IN	Set to 1 when the voltage comparator result is PASS.
Bit 3	V_LO	Set to 1 when the voltage comparator result is lower threshold FAIL.
Bit 2	R_HI	Set to 1 when the resistance comparator result is upper threshold FAIL.
Bit 1	R_IN	Set to 1 when the resistance comparator result is PASS.
Bit 0	R_LO	Set to 1 when the resistance comparator result is lower threshold FAIL.

Event register of the Status Query Register Group (**:STATus:QUEStionable:EVENt?**) and enable register of the Status Query Register Group (**:STATus:QUEStionable:ENABle**)



Register Reading and Writing

Register	Read	Write
Status Byte Register	*STB?	-
Service Request Enable Register	*SRE?	*SRE
Standard Event Status Register	*ESR?	-
Standard Event Status Enable Register	*ESE?	*ESE
Event register of the Standard Operation Register Group (Status data)	:STATus:OPERation :CONDition?	-
Event register of the Standard Operation Register Group (Event data)	:STATus:OPERation :EVENt?	-
Enable register of the Standard Operation Register Group	:STATus:OPERation :ENABle?	:STATus:OPERation :ENABle
Event register of the Status Query Register Group (Status data)	:STATus:QUEStionable:CONDition?	-
Event register of the Status Query Register Group (Event data)	:STATus:QUEStionable:EVENt?	-
Enable register of the Status Query Register Group	:STATus:QUEStionable:ENABle?	:STATus:QUEStionable:ENABle

Measurement Value Formats

In the measured value format settings, the measurement format that can be acquired from **:FETCh?**, **:READ?**, **:SYSTem:COMMunicate:DATAout ON, :MEMory:DATA?** can be changed.

When **:SYSTem:COMMunicate:FORMat FIX** is set

• Resistance: Unit Ω

Measurement Range	Measured Value	±OvrRng	Source RR Error	Sense RR Error	Sense Circuit OvrRng	Source Circuit Contact Error	Sense Circuit Contact Error	Measurement Fault (No data)
3mΩ(300mA)	±□.□□□□□E-03	±1.00000E+09	+1.00000E+10	+1.00000E+11	+1.00000E+12	+1.00000E+13	+1.00000E+14	+1.00000E+15
3mΩ(100mA)	±□.□□□□□E-03	±1.00000E+09	+1.00000E+10	+1.00000E+11	+1.00000E+12	+1.00000E+13	+1.00000E+14	+1.00000E+15
30mΩ	±□□.□□□□E-03	±10.0000E+08	+10.0000E+09	+10.0000E+10	+10.0000E+11	+10.0000E+12	+10.0000E+13	+10.0000E+14
300mΩ	±□□□.□□□E-03	±100.000E+07	+100.000E+08	+100.000E+09	+100.000E+10	+100.000E+11	+100.000E+12	+100.000E+13
3Ω	±□.□□□□□E+00	±1.00000E+09	+1.00000E+10	+1.00000E+11	+1.00000E+12	+1.00000E+13	+1.00000E+14	+1.00000E+15
30Ω	±□□.□□□□E+00	±10.0000E+08	+10.0000E+09	+10.0000E+10	+10.0000E+11	+10.0000E+12	+10.0000E+13	+10.0000E+14

• Voltage: Unit V

Measurement Range	Measured Value	±OvrRng	Source RR Error	Sense RR Error	Sense Circuit OvrRng	Source Circuit Contact Error	Sense Circuit Contact Error	Measurement Fault (No data)
10V (HIGH Z)	±□□.□□□□□□E+00	±10.000000E+08	+10.000000E+09	+10.000000E+10	+10.000000E+11	+10.000000E+12	+10.000000E+13	+10.000000E+14
10V (10MΩ)	±□□.□□□□□□E+00	±10.000000E+08	+10.000000E+09	+10.000000E+10	+10.000000E+11	+10.000000E+12	+10.000000E+13	+10.000000E+14
100V	±□□□.□□□□□E+00	±100.00000E+07	+100.00000E+08	+100.00000E+09	+100.00000E+10	+100.00000E+11	+100.00000E+12	+100.00000E+13

• Temperature:

Unit	Measured Value	±OvrRng	Measurement Fault (No data)
CELSIUS	±□□.□E+00	±10.0E+08	+10.0E+14
FAHRENHEIT	±□□□.□E+00	±100.0E+07	+100.0E+13

• Route-Resistance: Unit Ω

Measurement Range	Measured Value	±OvrRng	Sense Circuit OvrRng	Source Circuit Contact Error	Sense Circuit Contact Error	Measurement Fault (No data)
3mΩ(300mA)	±□□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
3mΩ(100mA)	±□□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
30mΩ	±□□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
300mΩ	±□□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
3Ω	±□□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
30Ω	±□□□.E+00	±100.E+07	+100.E+10	+100.E+11	+100.E+12	+100.E+13

Note：• When the number of digits for the integer is short, zeros are padded.
Example) When the measured value is 3mΩ in 300mΩ range,
the measured value is presented as +003.000E-03.

When **:SYSTem:COMMunicate:FORMat FLOAT** is set

• Resistance: Unit Ω

Measured Value	±OvrRng	Source RR Error	Sense RR Error	Sense Circuit OvrRng	Source Circuit Contact Error	Sense Circuit Contact Error	Measurement Fault (No data)
± □.□□□□□E±0□	±1.00000E+09	+1.00000E+10	+1.00000E+11	+1.00000E+12	+1.00000E+13	+1.00000E+14	+1.00000E+15

• Voltage: Unit V

Measured Value	±OvrRng	Source RR Error	Sense RR Error	Sense Circuit OvrRng	Source Circuit Contact Error	Sense Circuit Contact Error	Measurement Fault (No data)
± □.□□□□□□□E±0□	±1.0000000E+09	+1.0000000E+10	+1.0000000E+11	+1.0000000E+12	+1.0000000E+13	+1.0000000E+14	+1.0000000E+15

• Temperature:

Unit	Measured Value	±OvrRng	Measurement Fault (No data)
CELSIUS	± □□.□E+00	±10.0E+08	+10.0E+14
FAHRENHEIT	± □□□.□E+00	±100.0E+07	+100.0E+13

• Route-Resistance: Unit Ω

Measurement Range	Measured Value	±OvrRng	Sense Circuit OvrRng	Source Circuit Contact Error	Sense Circuit Contact Error	Measurement Fault (No data)
3mΩ(300mA)	± □□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
3mΩ(100mA)	± □□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
30mΩ	± □□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
300mΩ	± □□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
3Ω	± □□.□E+00	±10.0E+08	+10.0E+11	+10.0E+12	+10.0E+13	+10.0E+14
30Ω	± □□□E+00	±100E+07	+100E+10	+100E+11	+100E+12	+100E+13

Note：・ In case of temperature, when the number of digits for the integer is short, 0 is not padded.
Example) When the measured value is 5℃,
the measured value is presented as +5.0E+00.
・ In case of route-resistance, when the number of digits for the integer is short, 0 is not padded.
Example) When the measured value is 3Ω in 3Ω range,
the measured value is presented as +3.0E+00.

When **:SYSTem:COMMunicate:BT3562A ON** is set

:SYSTem:COMMunicate:FORMat <FIX/FLOAT> is not available.

• Resistance: Unit Ω

Measurement Range	Measured Value	±OvrRng	Measurement Fault
3mΩ(300mA)	± □□.□□□□□E-3	±10.00000E+8	+10.00000E+9
3mΩ(100mA)	± □□.□□□□□E-3	±10.00000E+8	+10.00000E+9
30mΩ	± □□□.□□□□E-3	±100.0000E+7	+100.0000E+8
300mΩ	± □□□□.□□□E-3	±1000.000E+6	+1000.000E+7
3Ω	± □□.□□□□□E+0	±10.00000E+8	+10.00000E+9
30Ω	± □□□.□□□□E+0	±100.0000E+7	+100.0000E+8

• Voltage: Unit V

Measurement Range	Measured Value	±OvrRng	Measurement Fault
10V (HIGH Z)	± □□.□□□□□□E+0	±10.000000E+8	+10.000000E+9
10V (10MΩ)	± □□.□□□□□□E+0	±10.000000E+8	+10.000000E+9
100V	± □□□.□□□□□E+0	±100.00000E+7	+100.00000E+8

• Temperature:

Unit	Measured Value	±OvrRng	Measurement Fault
CELSIUS	± □□.□E+0	±10.0E+8	+10.0E+9
FAHRENHEIT	± □□□.□E+0	±100.0E+7	+100.0E+8

• Route-Resistance: Unit Ω

Measurement Range	Measured Value	±OvrRng	Measurement Fault
3mΩ(300mA)	± □□.□E+0	±10.0E+8	+10.0E+9
3mΩ(100mA)	± □□.□E+0	±10.0E+8	+10.0E+9
30mΩ	± □□.□E+0	±10.0E+8	+10.0E+9
300mΩ	± □□.□E+0	±10.0E+8	+10.0E+9
3Ω	± □□.□E+0	±10.0E+8	+10.0E+9
30Ω	± □□□.E+0	±100.E+7	+100.E+8

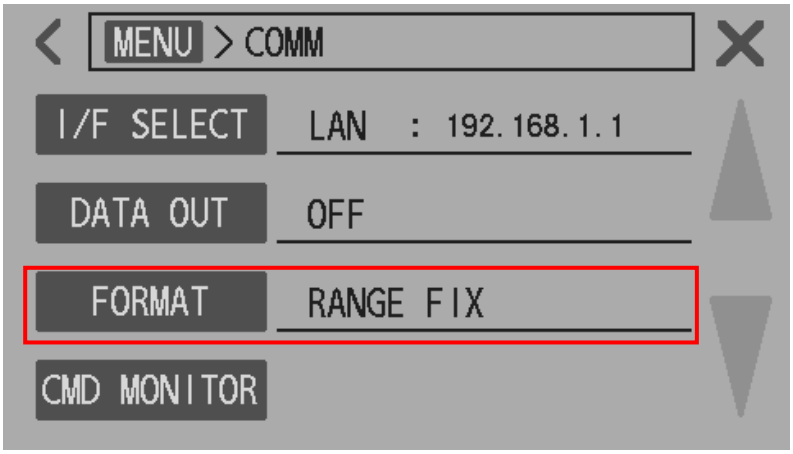
Note：・ For positive measured values, a space (ASCII 20H) represents the sign.
・ When the number of digits for the integer is short, a space (ASCII 20H) is padded.
Example) When the measured value is 3Ω in 3Ω range,
the measured value is presented as xx3.00000E+0.
When the measured value is -3V in 100V range,
the measured value is presented as -xx3.00000E+0.

(“x” is a space (ASCII 20H))

The format of measurement values can be changed in any of the following settings.

- Communication Commands
  :SYSTem:COMMunicate:FORMat <FIX/FLOAT> (See: Output Format p.38)
  :SYSTem:COMMunicate:BT3562A <1/0/ON/OFF> (See: BT3562A Compatible p.64)

- Instrument Screen



Initialization Items

Item \ Initialization Method	At Power-on	Instrument System Reset	:SYST:RES Command	*RST :SYST:PRES Command	*CLS Command	Factory Default
LAN IP address	-	192.168.1.1	-	-	-	192.168.1.1
LAN Subnet mask	-	255.255.255.0	-	-	-	255.255.255.0
LAN Gateway	-	0.0.0.0	-	-	-	0.0.0.0
LAN Communications command port	-	23	-	-	-	23
RS-232C Setting (baud rate)	-	9600	-	-	-	9600
Handshake response on/off	-	OFF	-	-	-	OFF
BT3562A Mode on/off	-	OFF	-	-	-	OFF
Data Output Settings	-	●	-	-	-	●
Headers on/off	OFF	OFF	OFF	OFF	-	OFF
Device-specific function (range etc)	-	●	●	●*1	-	●
Temperature Unit	-	-	-	-	-	●
Clock	-	-	-	-	-	-
Output Queue	●	●	-	-	-	●
Input buffer	●	●	-	-	-	●
Status Byte Register	●	●	-	-	●*2	●
Event registers	●*3	●	-	-	●	●
Enable registers	●	●	-	-	-	●

*1. Except for the Panel data and the adjustment values.
*2. All bits except the MAV bit(bit4) are cleared.
*3. All bits except the PON bit(bit7) are cleared.

Command Execution Time

Command execution time indicates the time for analyzing and processing long form commands.
However, the command execution time for commands with data is the time described according to the data format specified in the <data portion>.

- Display delays may occur depending on the frequency of communication processes and process contents.
- All commands are processed sequentially.
- In communications with the controller, time must be added for data transmission. USB or LAN transfer time depends on the controller.
  The RS-232C transfer time, with start bit 1, data length 8, no parity, and stop bit 1, has a total of 10-bit. When the transfer speed (baud rate) setting is N bps, the general result will be as follows:
  Transfer time T [1 character/sec] = Baud rate N [bps]/10 [bits]
  Since resistance and voltage measurement value is 29 characters, a 1 data transfer time will be 29/T.
  (Example) For 9600 bps, 29/(9600/10) = Approx. 30 ms
- Wait until measurements stabilize after a change when a setting command is used.

Command	Execution time (except communication time)
***RST** **:SYSTem:PRESet**	10 ms or less
:SYSTem:RESet	2 s or less
:FETCh?	5 ms or less
:READ?	Measurement time + 10 ms or less
***RCL** ***SAV**	1 s or less
***TST?**	30 s or less
:ADJust:ZERO:SINGle?	2 s or less
:ADJust:REFerential:BASE:DUT? **:ADJust:REFerential:POSItion?**	2 s or less
:SYSTem:CALibration?	60 s or less
Commands other than those above	10 ms or less

Errors During Communications

An error occurs when messages are executed in the following cases:

- Command Error
  When message syntax (spelling) is invalid
  When the data format in a command or query is invalid
- Query Error
  When the response message exceeds 512 bytes
- Parameter Error
  When invalid character or numeric data is received
- Execution Error
  When message cannot be executed under specific setting conditions or during measurement

2 Message List

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
Standard Commands		
*IDN?	(<Manufacture name>,<Model name>,<Serial number>,<Software version>)	Queries the Device ID (Identification code).
*OPT?	(0)	Identifies installed options.
*RST		Initializes the device.
*TST?	(0 to 7)	Executes the self-test and queries the result.
*SAV	<Panel No.>	Saves the setting conditions (panel save).
*RCL	<Panel No.>	Loads the setting conditions (panel load).
*TRG		Requests sampling.
*OPC		Sets OPC of SESR after all operations that are being executed are completed.
*OPC?		Responds with ASCII “1” after all operations that are being executed are completed.
*WAI		Executes subsequent commands after command processing is completed.
*CLS		Clears the Event Registers and the Status Byte Register except for Output Queue bit.
*ESE	0 to 255	Writes the Standard Event Status Enable Register (SESER).
*ESE?	(0 to 255)	Reads the Standard Event Status Enable Register (SESER).
*ESR?	0 to 255	Reads and clears the Standard Event Status Register (SESR).
*SRE	0 to 255	Writes the Service Request Enable Register (SRER).
*SRE?	(0 to 255)	Reads the Service Request Enable Register (SRER).
*STB?	(0 to 255)	Reads the Status Byte Register and MSS bit.
Event Registers		
:STATus:OPERation:CONDition?	(0 to 65535)	Reads the Condition Register of the Standard Operation Register Group.
:STATus:OPERation[:EVENt]?	(0 to 65535)	Reads the Event Register of the Standard Operation Register Group.
:STATus:OPERation:ENABle	0 to 65535	Writes the Enable Register of the Standard Operation Register Group.
:STATus:OPERation:ENABle?	(0 to 65535)	Reads the Enable Register of the Standard Operation Register Group.
:STATus:QUEStionable:CONDition?	(0 to 65535)	Reads the Condition Register of the Status Query Register Group.
:STATus:QUEStionable[:EVENt]?	(0 to 65535)	Reads the Event Register of the Status Query Register Group.
:STATus:QUEStionable:ENABle	0 to 65535	Writes the Enable Register of the Status Query Register Group.
:STATus:QUEStionable:ENABle?	(0 to 65535)	Reads the Enable Register of the Status Query Register Group.
Reading Measured Values		
:FETCh?	(<Measured value>)	Reads the most recent measured values.
:READ?	(<Measured value>)	Measurement (waits for trigger and reads the measured values)
:ABORt		Cancels the measurement.
:SYSTem:COMMunicate:FORMat	<FIX/FLOAT>	Sets the output format of measured values.
:SYSTem:COMMunicate:FORMat?	(<FIX/FLOAT>)	Queries the output format of measured values.
:TEMPerature:UNIT	<Celsius/Fahrenheit>	Sets the temperature unit.
:TEMPerature:UNIT?	(<CELSIUS/FAHRENHEIT>)	Queries the temperature unit.
Measurement Function		
:FUNCtion	<RV/R/V/RESistance/VOLTage>	Sets the measurement function.
:FUNCtion?	(<RV/R/V>)	Queries the measurement function.
Measurement Speed		
:SAMPle:RATE	<FAST1/FAST2/MEDIUM1/MEDIUM2/SLOW1/SLOW2/ EXFast/FAST/MEDium/SLOW>	Sets the measurement speed.
:SAMPle:RATE?	(<FAST1/FAST2/MEDIUM1/MEDIUM2/SLOW1/SLOW2>)	Queries the measurement speed.

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
Measurement Range		
:RESistance:RANGe	<3m/30m/300m/3/30/Resistance value of DUT>	Sets the resistance's measurement range.
:RESistance:RANGe?	(<3m/30m/300m/3/30>)	Queries the resistance's measurement range.
:RESistance:RANGe:AUTO	<1/0/ON/OFF>	Sets the measurement AUTO range.
:RESistance:RANGe:AUTO?	(<ON/OFF>)	Queries the measurement AUTO range.
:VOLTage:RANGe	<10V/100V/Voltage value of DUT>	Sets the voltage's measurement range.
:VOLTage:RANGe?	(<10V/100V>)	Queries the voltage's measurement range.
:VOLTage:RANGe:AUTO	<1/0/ON/OFF>	Sets the measurement AUTO range.
:VOLTage:RANGe:AUTO?	(<ON/OFF>)	Queries the measurement AUTO range.
Resistance's Measurement Current		
:RESistance:CURRent	<HIGH/LOW>	Sets the resistance's measurement current.
:RESistance:CURRent?	(<HIGH/LOW>)	Queries the resistance's measurement current.
Resistance's Mutual Interference Reductions		
:RESistance:MIR:STATe	<1/0/ON/OFF>	Sets the resistance's mutual interference reduction function.
:RESistance:MIR:STATe?	(<ON/OFF>)	Queries the resistance's mutual interference reduction function.
:RESistance:MIR:ROLE	<PRIMARY/SECONDARY>	Sets the role for resistance's mutual interference reduction.
:RESistance:MIR:ROLE?	(<PRIMARY/SECONDARY>)	Queries the role for resistance's mutual interference reduction.
Resistance's Number of Digits		
:RESistance:DIGits	<5/6>	Sets the number of digits for resistance measurement.
:RESistance:DIGits?	(<5/6>)	Queries the number of digits for resistance measurement.
Voltage's Input Impedance		
:VOLTage:IMPedance	<10M/HIGH_Z>	Sets the voltage's input impedance in 10V range.
:VOLTage:IMPedance?	(<10M/HIGH_Z>)	Queries the voltage's input impedance.
Voltage's Absolute Value		
:VOLTage:ABSolute	<1/0/ON/OFF>	Sets the voltage's absolute value function.
:VOLTage:ABSolute?	(<ON/OFF>)	Queries the voltage's absolute value function.
Trigger		
:INITiate:CONTinuous	<1/0/ON/OFF>	Sets the continuous measurement.
:INITiate:CONTinuous?	(<ON/OFF>)	Queries the continuous measurement.
:INITiate[:IMMediate]		Initiates the trigger wait state.
:TRIGger:SOURce	<INTernal/EXTernal/IMMediate>	Sets the trigger source.
:TRIGger:SOURce?	(<INTERNAL/EXTERNAL>)	Queries the trigger source.
:TRIGger:DELay:STATe	<1/0/ON/OFF>	Sets the trigger delay function.
:TRIGger:DELay:STATe?	(<ON/OFF>)	Queries the trigger delay function.
:TRIGger:DELay	<Delay time(sec)>	Sets the period of trigger delay.
:TRIGger:DELay?	(<Delay time(sec)>)	Queries the period of trigger delay.
Averaging Functions		
:CALCulate:AVERage:STATe	<1/0/ON/OFF>	Sets the averaging function.
:CALCulate:AVERage:STATe?	(<ON/OFF>)	Queries the averaging function.
:CALCulate:AVERage:COUNt	<1 to 256>	Sets the number of averaging calculation data.
:CALCulate:AVERage:COUNt?	(<1 to 256>)	Queries the number of averaging calculation data.
Zero Display Width		
:CALCulate:ZEROdisplay:WIDTH	<1/0/ON/OFF>	Sets the measured value's zero-display width.
:CALCulate:ZEROdisplay:WIDTH?	(<ON/OFF>)	Queries the measured value's zero-display width.

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
Comparator		
:COMParator:LIMit:STATe	<1/0/ON/OFF>	Sets the comparator function.
:COMParator:LIMit:STATe?	(<ON/OFF>)	Queries the comparator function.
:COMParator:LIMit:BEEPer	<OFF/HL/IN/BOTH1/BOTH2>	Sets the judgement buzzer.
:COMParator:LIMit:BEEPer?	(<OFF/HL/IN/BOTH1/BOTH2>)	Queries the judgement buzzer.
:COMParator:LIMit:ABSolute	<1/0/ON/OFF>	Sets the voltage's absolute value judgement function.
:COMParator:LIMit:ABSolute?	(<ON/OFF>)	Queries the voltage's absolute value judgement function.
:COMParator:LIMit:RESistance:UPPer	<Upper threshold>	Sets the resistance's upper threshold.
:COMParator:LIMit:RESistance:UPPer?	(<Upper threshold>)	Queries the resistance's upper threshold.
:COMParator:LIMit:VOLTage:UPPer	<Upper threshold>	Sets the voltage's upper threshold.
:COMParator:LIMit:VOLTage:UPPer?	(<Upper threshold>)	Queries the voltage's upper threshold.
:COMParator:LIMit:RESistance:LOWer	<Lower threshold>	Sets the resistance's lower threshold.
:COMParator:LIMit:RESistance:LOWer?	(<Lower threshold>)	Queries the resistance's lower threshold.
:COMParator:LIMit:VOLTage:LOWer	<Lower threshold>	Sets the voltage's lower threshold.
:COMParator:LIMit:VOLTage:LOWer?	(<Lower threshold>)	Queries the voltage's lower threshold.
:COMParator:LIMit:RR:STATe	<1/0/ON/OFF>	Sets the route-resistance's comparator function.
:COMParator:LIMit:RR:STATe?	(<ON/OFF>)	Queries the route-resistance's comparator function.
:COMParator:LIMit:RR:FAIL	<Fail threshold>	Sets the route-resistance's fail threshold.
:COMParator:LIMit:RR:FAIL?	(<Fail threshold>)	Queries the route-resistance's fail threshold.
:COMParator:LIMit:RR:WARNing	<Warning threshold>	Sets the route-resistance's warning threshold.
:COMParator:LIMit:RR:WARNing?	(<Warning threshold>)	Queries the route-resistance's warning threshold.
:COMParator:LIMit:CLEar		Clears the Comparator Event Status Register.
:COMParator:LIMit:RESistance:RESult?	(<HI/IN/LO/ERR/OFF>)	Queries the comparator result for resistance measurement.
:COMParator:LIMit:VOLTage:RESult?	(<HI/IN/LO/ERR/OFF>)	Queries the comparator result for voltage measurement.
:COMParator:LIMit:RR:RESult?	(<PASS/WARNING/FAIL/ERR/OFF>)	Queries the comparator result for route-resistance measurement.
Adjustment		
:ADJust:TYPE	<ZERO/REFerential/OFF>	Sets the adjustment type.
:ADJust:TYPE?	(<ZERO/REFERENTIAL/OFF>)	Queries the adjustment type.
:ADJust:ZERO:MODE	<SINGle/MULTiple>	Sets the zero-adjustment mode.
:ADJust:ZERO:MODE?	(<SINGLE/MULTIPLE>)	Queries the zero-adjustment mode.
:ADJust:ZERO:SINGle?	(<PASS/FAIL>)	Executes the zero-adjustment for single mode and queries the result.
:ADJust:ZERO:MULTiple?	<Channel No.> (<PASS/FAIL>)	Executes the zero-adjustment for multiple mode and queries the result.
:ADJust:ZERO:MULTiple:DONE?	(<Completed MIN Channel No.>,<Completed MAX Channel No.>)	Queries the completed channel number for zero-adjustment multiple mode.
:ADJust:ZERO:MULTiple:CLEar		Deletes the zero-adjustment values for multiple mode.
:ADJust:ZERO:MULTiple:REFLect		Reflects setting conditions when the zero-adjustment multiple mode was executed.
:ADJust:ZERO:MULTiple:CH	<Channel No.>	Sets the channel to use for zero-adjustment multiple mode.
:ADJust:ZERO:MULTiple:CH?	(<Channel No.>)	Queries the channel to use for zero-adjustment multiple mode.
:ADJust:ZERO:MULTiple:LIST	<Start Channel No.>,<End Channel No.>	Sets the start and end channels when executing zero-adjustment multiple mode.
:ADJust:ZERO:MULTiple:LIST?	(<Start Channel No.>,<End Channel No.>)	Queries the start and end channels when executing zero-adjustment multiple mode.

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
:ADJust:REFerential:BASE:ZERO?	(<PASS/FAIL>)	Executes the zero-adjustment at base position for referential adjustment and queries the result.
:ADJust:REFerential:BASE:DUT?	(<PASS/FAIL>)	Executes the measurement at base position for referential adjustment and queries the result.
:ADJust:REFerential:BASE:EXISt?	(<EXIST/NONE>)	Queries the existence of measured value at base position for referential adjustment.
:ADJust:REFerential:POSItion?	<Position No.> (<PASS/FAIL>)	Executes the referential adjustment and queries the result.
:ADJust:REFerential:POSItion:DONE?	(<Completed MIN Position No.>,<Completed MAX Position No.>)	Queries the completed position number for referential adjustment.
:ADJust:REFerential:POSItion:CLEar		Deletes the referential adjustment values.
:ADJust:REFerential:POSItion:REFLect		Reflects setting conditions when the referential adjustment was executed.
:ADJust:REFerential:POSItion:NO	<Position No.>	Sets the position to use for referential adjustment.
:ADJust:REFerential:POSItion:NO?	(<Position No.>)	Queries the position to use for referential adjustment.
:ADJust:REFerential:POSItion:LIST	<Start Position No.>,<End Position No.>	Sets the start and end positions when executing referential adjustment.
:ADJust:REFerential:POSItion:LIST?	(<Start Position No.>,<End Position No.>)	Queries the start and end positions when executing referential adjustment.
Self-Calibration		
:SYSTem:CALibration?	<RV/R/V> (<PASS/FAIL>)	Executes the self-calibration and queries the result.
:SYSTem:CALibration:AUTO	<1/0/ON/OFF>	Sets the voltage's AUTO self-calibration.
:SYSTem:CALibration:AUTO?	(<ON/OFF>)	Queries the voltage's AUTO self-calibration.
Saving and Loading Measurement Conditions		
*SAV	<Panel No.>	Saves the measurement conditions (panel save).
*RCL	<Panel No.>	Loads the measurement conditions (panel load).
:SYSTem:PANel:CLEar	<Panel No.>	Deletes the panel.
Clock		
:SYSTem:DATE	<Year>,<Month>,<Day>	Sets the system date.
:SYSTem:DATE?	(<Year>,<Month>,<Day>)	Queries the system date.
:SYSTem:TIME	<Hour>,<Minute>,<Second>	Sets the clock.
:SYSTem:TIME?	(<Hour>,<Minute>,<Second>)	Queries the clock.
Key-Lock		
:SYSTem:KLOCk	<1/0/ON/OFF>	Sets the key-lock state.
:SYSTem:KLOCk?	(<ON/OFF>)	Queries the key-lock state.
Key-Beeper		
:SYSTem:BEEPer:KEY	<1/0/ON/OFF>	Sets the key-beeper state.
:SYSTem:BEEPer:KEY?	(<ON/OFF>)	Queries the key-beeper state.
Line Frequency		
:SYSTem:LFRequency	<AUTO/50/60>	Sets the line frequency.
:SYSTem:LFRequency?	(<AUTO/50/60>)	Queries the line frequency.
Advaced Mode		
:SYSTem:ADVanced:STATe	<1/0/ON/OFF>	Sets the advanced mode state.
:SYSTem:ADVanced:STATe?	(<ON/OFF>)	Queries the advanced mode state.
Reset		
:SYSTem:PRESet		Executes normal reset.
:SYSTem:RESet		Executes system reset.

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
Display Settings		
:DISPlay:BACKlight	<Brightness>	Sets the screen brightness.
:DISPlay:BACKlight?	(<Brightness>)	Queries the screen brightness.
:DISPlay:VIEW	<NUMeric/RR/ADJust/COMParator/CONFiguration/ADVanced>	Sets the display type.
:DISPlay:VIEW?	(<NUMERIC/RR/ADJUST/COMPARATOR/CONFIGURATION/ ADVANCED>)	Queries the display type.
:DISPlay:SAVEr:STATe	<1/0/ON/OFF>	Sets the screen saver state.
:DISPlay:SAVEr:STATe?	(<ON/OFF>)	Queries the screen saver state.
:DISPlay:SAVEr:WAIT	<Wating time(min)>	Sets the waiting time for screen saver.
:DISPlay:SAVEr:WAIT?	(<Wating time(min)>)	Queries the waiting time for screen saver.
:DISPlay:SAVEr:COMMunicate:WAKE	<1/0/ON/OFF>	Sets the method to return from screen saver mode with communication.
:DISPlay:SAVEr:COMMunicate:WAKE?	(<ON/OFF>)	Queries the method to return from screen saver mode with communication.
:DISPlay:SAVEr:CLEar		Opens the screen.
Communication Settings		
:SYSTem:LOCal		Returns to the local control state.
:SYSTem:COMMunicate	<USB/LAN/RS232c/MMEMory>	Sets the communication interface.
:SYSTem:COMMunicate?	(<USB/LAN/RS232C>)	Queries the communication interface.
:SYSTem:COMMunicate:MONitor	<1/0/ON/OFF>	Sets the command monitoring state.
:SYSTem:COMMunicate:MONitor?	(<ON/OFF>)	Queries the command monitoring state.
:SYSTem:COMMunicate:LAN:IPADdress	<IP address>	Sets the IP address.
:SYSTem:COMMunicate:LAN:IPADdress?	(<IP address>)	Queries the IP address.
:SYSTem:COMMunicate:LAN:CONTrol	<Port No.>	Sets the LAN port.
:SYSTem:COMMunicate:LAN:CONTrol?	(<Port No.>)	Queries the LAN port.
:SYSTem:COMMunicate:LAN:SMASk	<Sub-net mask>	Sets the sub-net mask.
:SYSTem:COMMunicate:LAN:SMASk?	(<Sub-net mask>)	Queries the sub-net mask.
:SYSTem:COMMunicate:LAN:GATeway	<Address>	Sets the default gateway.
:SYSTem:COMMunicate:LAN:GATeway?	(<Address>)	Queries the default gateway.
:SYSTem:COMMunicate:LAN:UPDate		Reflects the LAN settings.
:SYSTem:COMMunicate:LAN:MAC?	(<MAC address>)	Queries the MAC address.
:SYSTem:COMMunicate:RS232C:SPEed	<Baud rate>	Sets the baud rate.
:SYSTem:COMMunicate:RS232C:SPEed?	(<Baud rate>)	Queries the baud rate.
Data Output Settings		
:SYSTem:COMMunicate:DATAout	<1/0/ON/OFF>	Sets the measurement-synchronized data output function.
:SYSTem:COMMunicate:DATAout?	(<ON/OFF>)	Queries the measurement-synchronized data output function.
:SYSTem:COMMunicate:DATAout:FORMat	<Type of items>	Sets the type of items for the measurement-synchronized data output function.
:SYSTem:COMMunicate:DATAout:FORMat?	(<Type of items>)	Queries the type of items for the measurement-synchronized data output function.
Command Response Settings		
:SYSTem:COMMunicate:HEADer	<1/0/ON/OFF>	Sets the header presence.
:SYSTem:COMMunicate:HEADer?	(<ON/OFF>)	Queries the header presence.
:SYSTem:COMMunicate:RESPonse	<1/0/ON/OFF>	Sets the handshake response state.
:SYSTem:COMMunicate:RESPonse?	(<ON/OFF>)	Queries the handshake response state.
:SYSTem:COMMunicate:BT3562A	<1/0/ON/OFF>	Sets the BT3562A compatible mode.
:SYSTem:COMMunicate:BT3562A?	(<ON/OFF>)	Queries the BT3562A compatible mode.

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
Memory Functions		
:MEMory:DATA?	(<Memory No.>,<Resistance value>,<Voltage value>)	Reads the measured values stored in the memory.
N	(<Memory No.>,<Resistance value>,<Voltage value>)	Reads the measured values stored in the memory individually.
:MEMory:STATe	<1/0/ON/OFF>	Sets the memory function.
:MEMory:STATe?	(<ON/OFF>)	Queries the memory function.
:MEMory:COUNt?	(<Number of Memory data>)	Queries the number of measured values stored in the memory.
:MEMory:CLEar		Deletes the memory data.
EXT. I/O		
:IO:MODE?	(<NPN/PNP>)	Queries the NPN/PNP hardware switch state.
:IO:FILTer:STATe	<1/0/ON/OFF>	Sets the TRIG signal filter function.
:IO:FILTer:STATe?	(<ON/OFF>)	Queries the TRIG signal filter function.
:IO:FILTer:TIME	<Filter time(sec)>	Sets the period of TRIG signal filter function.
:IO:FILTer:TIME?	(<Filter time(sec)>)	Queries the period of TRIG signal filter function.
:IO:EOM:MODE	<HOLD/PULSe>	Sets the EOM signal output mode.
:IO:EOM:MODE?	(<HOLD/PULSE>)	Queries the EOM signal output mode.
:IO:EOM:PULSe	<Pulse width(sec)>	Sets the width of EOM pulse signal.
:IO:EOM:PULSe?	(<Pulse width(sec)>)	Queries the width of EOM pulse signal.
:IO:ERRor	<SYNChronous/ASYNchronous>	Sets the ERR signal output mode.
:IO:ERRor?	(<SYNCHRONOUS/ASYNCHRONOUS>)	Queries the ERR signal output mode.
:IO:INPut?	(0 to 65535)	Reads the External I/O Input Register.
Error Information		
:SYSTem:ERRor?	(<Error No.>,<Error information>)	Reads and clears the error information.
Device ID Informations		
*IDN?	(<Manufacture name>,<Model name>,<Serial number>,<Software version>)	Queries the Device ID (Identify code).
:QPID	(<Model name>)	Queries the model name.
:QSERIAL	(<Serial number>)	Queries the serial number.
:FPGA:VERsion?	<DIGITAL/SENSE> (<Version No.>)	Queries the FPGA version number.
Command Compatible with BT3562A		
:ESR0?	(0 to 65535)	Reads the Event Register of the Standard Operation Register Group.
:ESE0	0 to 65535	Writes the Enable Register of the Standard Operation Register Group.
:ESE0?	(0 to 65535)	Reads the Enable Register of the Standard Operation Register Group.
:ESR1?	(0 to 65535)	Reads the Event Register of the Status Query Register Group.
:ESE1	0 to 65535	Writes the Enable Register of the Status Query Register Group.
:ESE1?	(0 to 65535)	Reads the Enable Register of the Status Query Register Group.
:AUTorange	<1/0/ON/OFF>	Sets the measurement AUTO range.
:AUTorange?	(<ON/OFF>)	Queries the measurement AUTO range.
:CALCulate:AVERage	<Average count>	Sets the number of averaging calculation data.
:CALCulate:AVERage?	(<Average count>)	Queries the number of averaging calculation data.
:CALCulate:SUPPress	<1/0/ON/OFF>	Sets the zero-suppress function.
:CALCulate:SUPPress?	(<ON/OFF>)	Queries the zero-suppress function.
:CALCulate:ABSolute	<1/0/ON/OFF>	Sets the voltage's absolute value function.
:CALCulate:ABSolute?	(<ON/OFF>)	Queries the voltage's absolute value function.
:CALCulate:LIMit:STATe	<1/0/ON/OFF>	Sets the comparator function.
:CALCulate:LIMit:STATe?	(<ON/OFF>)	Queries the comparator function.
:CALCulate:LIMit:BEEPer	<OFF/HL/IN/BOTH1/BOTH2>	Sets the judgement buzzer.
:CALCulate:LIMit:BEEPer?	(<OFF/HL/IN/BOTH1/BOTH2>)	Queries the judgement buzzer.
:CALCulate:LIMit:ABS	<1/0/ON/OFF>	Sets the voltage's absolute value judgement function.
:CALCulate:LIMit:ABS?	(<ON/OFF>)	Queries the voltage's absolute value judgement function.

Message [ ]: Omissible	Data [ ]: Omissible, (): Response data	Description
:CALCulate:LIMit:RESistance:MODE	<HL/REF>	Sets the resistance's comparator mode.
:CALCulate:LIMit:RESistance:MODE?	(<HL/REF>)	Queries the resistance's comparator mode.
:CALCulate:LIMit:VOLTage:MODE	<HL/REF>	Sets the voltage's comparator mode.
:CALCulate:LIMit:VOLTage:MODE?	(<HL/REF>)	Queries the voltage's comparator mode.
:CALCulate:LIMit:RESistance:UPPer	<Upper threshold>	Sets the resistance's upper threshold.
:CALCulate:LIMit:RESistance:UPPer?	(<Upper threshold>)	Queries the resistance's upper threshold.
:CALCulate:LIMit:VOLTage:UPPer	<Upper threshold>	Sets the voltage's upper threshold.
:CALCulate:LIMit:VOLTage:UPPer?	(<Upper threshold>)	Queries the voltage's upper threshold.
:CALCulate:LIMit:RESistance:LOWer	<Lower threshold>	Sets the resistance's lower threshold.
:CALCulate:LIMit:RESistance:LOWer?	(<Lower threshold>)	Queries the resistance's lower threshold.
:CALCulate:LIMit:VOLTage:LOWer	<Lower threshold>	Sets the voltage's lower threshold.
:CALCulate:LIMit:VOLTage:LOWer?	(<Lower threshold>)	Queries the voltage's lower threshold.
:CALCulate:LIMit:RESistance:REFerence	<Base value>	Sets the resistance's base value.
:CALCulate:LIMit:RESistance:REFerence?	(<Base value>)	Queries the resistance's base value.
:CALCulate:LIMit:VOLTage:REFerence	<Base value>	Sets the voltage's base value.
:CALCulate:LIMit:VOLTage:REFerence?	(<Base value>)	Queries the voltage's base value.
:CALCulate:LIMit:RESistance:PERCent	<Percentage value>	Sets the resistance's percentage value.
:CALCulate:LIMit:RESistance:PERCent?	(<Percentage value>)	Queries the resistance's percentage value.
:CALCulate:LIMit:VOLTage:PERCent	<Percentage value>	Sets the voltage's percentage value.
:CALCulate:LIMit:VOLTage:PERCent?	(<Percentage value>)	Queries the voltage's percentage value.
:CALCulate:LIMit:RESistance:RESult?	(<HI/IN/LO/ERR/OFF>)	Queries the comparator result for resistance measurement.
:CALCulate:LIMit:VOLTage:RESult?	(<HI/IN/LO/ERR/OFF>)	Queries the comparator result for voltage measurement.
:ADJust?	(<PASS/FAIL>)	Executes the zero-adjustment for single mode and queries the result.
:ADJust:CLEAr		Sets the adjustment type OFF.
:MEMory:CLEAr		Deletes the memory data.
:SYSTem:CALibration		Executes the voltage's self-calibration.
:SYSTem:SAVE	<Panel No.>	Saves the setting conditions (panel save).
:SYSTem:LOAD	<Panel No.>	Loads the setting conditions (panel load).
:SYSTem:BEEPer:STATe	<1/0/ON/OFF>	Sets the key-beeper state.
:SYSTem:BEEPer:STATe?	(<ON/OFF>)	Queries the key-beeper state.
:SYSTem:DATAout	<1/0/ON/OFF>	Sets the measurement-synchronized data output function.
:SYSTem:DATAout?	(<ON/OFF>)	Queries the measurement-synchronized data output function.
:SYSTem:HEADer	<1/0/ON/OFF>	Sets the header presence.
:SYSTem:HEADer?	(<ON/OFF>)	Queries the header presence.
:SYSTem:ELOCk	<1/0/ON/OFF>	Sets the External I/O input lock.
:SYSTem:ELOCk?	(<ON/OFF>)	Queries the External I/O input lock.
:SYSTem:EOM:MODE	<HOLD/PULSe>	Sets the EOM signal output mode.
:SYSTem:EOM:MODE?	(<HOLD/PULSE>)	Queries the EOM signal output mode.
:SYSTem:EOM:PULSe	<Pulse width(sec)>	Sets the width of EOM pulse signal.
:SYSTem:EOM:PULSe?	<Pulse width(sec)>	Queries the width of EOM pulse signal.
:SYSTem:ERRor	<SYNChronous/ASYNchronous>	Sets the ERR signal output mode.
:IO:IN?	(<Input data>)	Reads the External I/O Input Register.

3 Message Reference

Message Reference Interpretation

< >: Indicates the contents (character or numeric parameters) of the data portion of a message. Character parameters are returned as all capital letters.

Numeric Parameters:

- NRf Number format may be any of NR1, NR2 and NR3
- NR1 Integer data (e.g.: +12, -23, 34)
- NR2 Fixed-point data (e.g.: +1.23, -23.45, 3.456)
- NR3 Floating-point exponential representation data (e.g.: +1.0E-2, -2.3E+4)

Shows the command description.

Shows the message syntax. Explains the command data or response message. Describes the message.

Shows an example of an actual command application. (Normally described with HEADER OFF [except the HEADER command itself].)

Read/Write the Standard Event Status Enable Register (SESER)

Syntax

Command: ***ESE** <0 to 255 (NR1)>

Query: ***ESE?**

Response: <0 to 255 (NR1)>

Description

Command: The SESER mask is set to the numerical value 0 to 255. The initial value (at power-on) is 0.

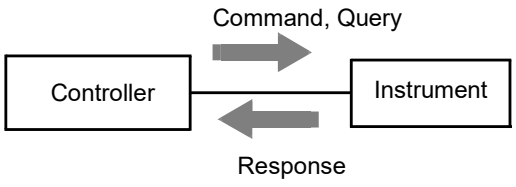
Query: The contents of the SESER, as set by the *ESE command, are returned as an NR1 value (0 to 255).

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

Example

***ESE 36**

(Sets bits 5 and 2 of SESER)



Standard Commands

(1) System Data Command

Query the Device ID (Identification Code)

Syntax

Query ***IDN?**

Response <Manufacturer name>,<Model name>,<Serial number>,
<Software version>

Example

*IDN?
HIOKI,BT6075,1234567890,V1.00
The Device ID is HIOKI BT6075, 1234567890, software version 1.00.
The <Model name> will be the following.
For BT6065 :BT6065
For BT6065-01 :BT6065-01
For BT6075 :BT6075
For BT6075-01 :BT6075-01

Identify Installed Options

Syntax

Query ***OPT?**

Response <0 (NR1) >

Example

*OPT?
0
When an option board is not installed.

(2) Internal Operation Command

Initialize the Device

Syntax

Command ***RST**

Description

Resets the instrument to its initial state.

Note

- The communication conditions, clock, temperature unit, panel data and adjustment values are not initialized.
- The Event Status Register is not cleared.
- When :SYSTem:RESet is executed, the panel data and adjustment values are also initialized.

Execute the Self-Test and Query the Result

Syntax Query ***TST?**
Response <0 to 7 (NR1)>

Description Performs the instrument self-test and return the result as NR1 value 0 to 7.
Returns zero when no error occurs.

128	64	32	16	8	4	2	1
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
unused	unused	unused	unused	unused	SDRAM Error	Internal-RAM Error	ROM Error

Example *TST?
1
A ROM error occurred.
Correct measurement may not be possible. Please repair before further use.

Note Approximately 30 seconds is spent to complete the self-test.

Save the Setting Conditions (Panel Save)

Syntax Command ***SAV** <Panel No.>
<Panel No.> = 1 to 6 (NR1)

Description Saves the current measurement conditions to the specified panel number.

Example *SAVE 1
The current measurement conditions are saved to panel No.1.

Note When a panel number to which measurement conditions are already saved is specified, the saved data is overwritten with the current measurement conditions.

Load the Setting Conditions (Panel Load)

Syntax Command ***RCL** <Panel No.>
<Panel No.> = 1 to 6 (NR1)

Description Loads the measurement conditions saved for the specified panel number.

Example *RCL 1
The measurement conditions saved for panel No.1 are loaded.

Note When a panel number to which measurement conditions are not saved is specified, an execution error occurs.

Request Sampling

Syntax Command ***TRG**

Description
- Performs one measurement in the Trigger Wait State.
  It may be necessary to insert wait processing after panel load or range selection.
  The wait time depends on DUT(Device Under Test).
- When :SYSTem:COMMunicate:DATAout ON is set,
  the first measured value is responded after sending *TRG.
- When :MEMory:STATe ON is set,
  the first measured value is saved to the internal memory after sending *TRG.

Example :TRIG:SOUR EXT
*TRG
External triggering is set and performs one measurement.

(3) Synchronization Commands

Set OPC bit of SESR after All Operations that are being Executed are Completed

Syntax Command **∗OPC**

Description Sets OPC (bit 0) of the Standard Event Status Register (SESR) when all commands prior to the ∗OPC have finished processing.

Respond with ASCII "1" after All Operations that are being Executed are Completed

Syntax Query **∗OPC?**
Response <1 (NR1)>

Description Responds with ASCII "1" when all commands prior to the ∗OPC have finished processing.

Execute Subsequent Commands after Command Processing is Completed

Syntax Command **∗WAI**

Description The instrument waits until all prior commands finish before executing any subsequent commands.

(4) Status and Event Control Commands

Clear the Event Registers and Status Byte Register (Except for Output Queue bit)

Syntax Command **∗CLS**

Description Clears the event status registers. The Status Byte Register bits corresponding to the event status registers are also cleared. (Standard Event Status Register, Standard Operation Register, Status Query Register)

Note [USB/LAN/RS-232C] The output queue bit is unaffected.

Read/Write the Standard Event Status Enable Register (SESER)

Syntax Command **∗ESE** <0 to 255 (NR1)>
Query **∗ESE?**
Response <0 to 255 (NR1)>

Description Command The SESER mask pattern is set to a numerical value (0 to 255). The default value (at power-on) is 0.

Query The status of the SESER, as set by the ∗ESE command, is returned as an NR1 value (0 to 255).

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

Example
∗ESE 36
(Sets bits 5 and 2 of SESER.)

∗ESE?
36
(Bits 5 and 2 of SESER has been set to 1.)

Read and Clear the Standard Event Status Register (SESR)

Syntax

Query ***ESR?**

Response <0 to 255 (NR1)>

Description

The status of the SESR is returned as an NR1 value (0 to 255), then the status is cleared.

[USB/LAN/ RS-232C]

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

Example

*ESR?

32

(Bit 5 of the SESR has been set to 1.)

Read/Write the Service Request Enable Register (SRER)

Syntax

Command ***SRE** <0 to 255 (NR1)>

Query ***SRE?**

Response <0 to 255 (NR1)>

Description

Command The SRER mask pattern is set to the numerical value (0 to 255). The default value (at power-on) is 0.

Query The status of the SRER, as set by the *SRE command, is returned as an NR1 value (0 to 255). Bit 6 and unused bits (bit3 and 7) are always 0.

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

Example

*SRE 33

(Set SRER bits 0 and 5 to 1.)

*SRE?

33

(SRER bits 0 and 5 have been set to 1.)

Read the Status Byte Register and MSS bit

Syntax

Query ***STB?**

Response <0 to 255 (NR1)>

Description

The status of the STB is returned as an NR1 value (0 to 255).

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

Example

*STB?

16

(STB bit 4 has been set to 1.)

Note

When :SYSTem:COMMunicate:BT3562A ON is set, bit 2 is always 0.

Device-Specific Commands

(1) Event Registers

See the following for a relationship with the Status Byte Register.
• Status Byte Register (p.7)
• Event Register (p.9)

Read the Condition Register of Standard Operation Register Group

Syntax Query **:STATus:OPERation:CONDition?**
Response <0 to 65535 (NR1)>

Description The status of the condition register of Standard Operation Register group is returned as an NR1 value (0 to 65535).

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	-	-	-	-	-	-	-

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	ERR	-	-	-	INDEX	EOM

Example
:STAT:OPER:COND?
2
The instrument is in the analog measurement state.

Note
• The latest status is returned when this command is received.
• Unlike :STATus:OPERation[:EVENt]?, bit information is not held.

Read the Event Register of Standard Operation Register Group

Syntax Query **:STATus:OPERation[:EVENt]?**
Response <0 to 65535 (NR1)>

Description The status of the event register of Standard Operation Register group is returned as an NR1 value (0 to 65535).

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	-	-	-	-	-	-	-

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	ERR	-	-	-	INDEX	EOM

Example
:STAT:OPER?
2
This instrument has been in the analog measurement state.

Note
• When an event occurs, 1 is set for the bit corresponding to each event.
• Until *CLS, this query is executed, or the power is turned on again, the bit is not cleared.

Write and Read the Enable Register of Standard Operation Register Group

Syntax

Command **:STATus:OPERation:ENABle** <0 to 65535 (NR1)>
Query **:STATus:OPERation:ENABle?**
Response <0 to 65535 (NR1)>

Description

Command: The mask pattern of enable register of standard operation register group is set to the numerical value (0 to 65535). The default value (at power-on) is 0.

Query: The status of the enable register of standard operation register group, as set by the :STATus:OPERation:ENABle command, is returned as an NR1 value (0 to 65535). Unused bits (bit15 to bit6 and bit4 to bit2) are always 0.

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	-	-	-	-	-	-	-

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	ERR	-	-	-	INDEX	EOM

Read the Condition Register of Status Query Register Group

Syntax

Query **:STATus:QUEStionable:CONDition?**
Response <0 to 65535 (NR1)>

Description

The status of the condition register of Status Query Register group is returned as an NR1 value (0 to 65535).

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
FAIL 2	PASS 2	-	-	-	RR_ FAIL	RR_ WARN	RR_ PASS

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FAIL 1	PASS 1	V_HI	V_IN	V_LO	R_HI	R_IN	R_LO

Example

```
:STAT:QUES:COND?
2
```

The comparator result of the resistance value is IN state.

Note

- The latest status is returned when this command is received.
- Unlike :STATus:QUEStionable[:EVENt]?, bit information is not held.

Read the Event Register of Status Query Register Group

Syntax Query **:STATus:QUEStionable[:EVENt]?**
Response <0 to 65535 (NR1)>

Description The status of the event register of Status Query Register group is returned as an NR1 value (0 to 65535).

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
FAIL 2	PASS 2	-	-	-	RR_ FAIL	RR_ WARN	RR_ PASS

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FAIL 1	PASS 1	V_HI	V_IN	V_LO	R_HI	R_IN	R_LO

Example
```
:STAT:QUES?
1
```
A comparator result of the resistance value has been LO state.

Note
- When an event occurs, 1 is set for the bit corresponding to each event.
- Until ∗CLS, this query is executed, or the power is turned on again, the bit is not cleared.

Write and Read the Enable Register of Status Query Register Group

Syntax Command **:STATus:QUEStionable:ENABle** <0 to 65535 (NR1)>
Query **:STATus:QUEStionable:ENABle?**
Response <0 to 65535 (NR1)>

Description Command The mask pattern of enable register of status query register group is set to the numerical value (0 to 65535). The default value (at power-on) is 0.

Query The status of the enable register of status query register group, as set by the :STATus:QUEStionable:ENABle command, is returned as an NR1 value (0 to 65535). Unused bits (bit13 to bit11) are always 0.

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
FAIL 2	PASS 2	-	-	-	RR_ FAIL	RR_ WARN	RR_ PASS

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
FAIL 1	PASS 1	V_HI	V_IN	V_LO	R_HI	R_IN	R_LO

(2) Reading Measured Values

Read the Most Recent Measured Value

Syntax Query **:FETCh?** [<TEMPerature/RR/TEMPerature,RR>]

Response (1) When TEMPerature/RR/TEMPerature,RR as the parameter is not specified

<Resistance (NR3)>,<Voltage (NR3)> (ΩV Function)
<Resistance (NR3)> (Ω Function)
<Voltage (NR3)> (V Function)

(2) When TEMPerature as the parameter is specified

<Resistance>,<Voltage>,<Temperature (NR3)> (ΩV Function)
<Resistance>,<Temperature (NR3)> (Ω Function)
<Voltage>,<Temperature (NR3)> (V Function)

(3) When RR as the parameter is specified

<Resistance>,<Voltage>,<Route-Resistance (NR3)> (ΩV Function)
<Resistance>,<Route-Resistance (NR3)> (Ω Function)
<Voltage>,<Route-Resistance (NR3)> (V Function)
Note: Route-Resistance value is as follows:
the Source Hi, the Source Lo, the Sense Hi and the Sense Lo.

(4) When TEMPerature,RR as the parameters are specified

<Resistance>,<Voltage>,<Temperature>,<Route-Resistance> (ΩV Function)
<Resistance>,<Temperature>,<Route-Resistance> (Ω Function)
<Voltage>,<Temperature>,<Route-Resistance> (V Function)
Note: Route-Resistance value is as follows:
the Source Hi, the Source Lo, the Sense Hi and the Sense Lo.

See: “Measurement Value Formats” (p.15)

Description Reads the most recent measured value. No trigger occurs.

See: Data Exporting Methods (p.78), Triggering (p.44)

Example

```
:TRIG:SOUR EXT
:INIT:CONT OFF
:FUNC RV
:SYST:COMM:FORM FIX
:INIT
*TRG
:FETC?
+1.00010E-03,+00.000001E+00
:FETC? TEMP
+1.00010E-03,+00.000001E+00,+23.8E+00
```

The most recent resistance measured value is 1.0001mΩ, voltage is 0.000001V, temperature is 23.8℃(74.84℉)

```
:SYST:COMM:FORM FLOAT
:INIT
*TRG
:FETC? RR
+1.00010E-03,+00.000001E+00,+0.1E+00,+0.2E+00,+0.3E+00,+0.4E+00
```

The most recent resistance measured value is 1.0001mΩ, voltage is 0.000001V, route-resistance source Hi is 0.1Ω, route-resistance source Lo is 0.2Ω, route-resistance sense Hi is 0.3Ω, route-resistance sense Lo is 0.4Ω.

```
:FETC? TEMP,RR
+1.00010E-03,+00.000001E+00,+23.8E+00,+0.1E+00,+0.2E+00,+0.3E+00,+0.4E+00
```

The most recent resistance measured value is 1.0001mΩ, voltage is 0.000001V, temperature is 23.8℃(74.84℉),route-resistance source Hi is 0.1Ω, route-resistance source Lo is 0.2Ω, route-resistance sense Hi is 0.3Ω, route-resistance sense Lo is 0.4Ω.

Measurement (waits for trigger and reads the measured values)

Syntax Query **:READ?** [<TEMPerature/RR/TEMPerature,RR>]

Response (1) When TEMPerature/RR/TEMPerature,RR as the parameter is not specified

<Resistance (NR3)>,<Voltage (NR3)> (ΩV Function)
<Resistance (NR3)> (Ω Function)
<Voltage (NR3)> (V Function)

(2) When TEMPerature as the parameter is specified

<Resistance>,<Voltage>,<Temperature (NR3)> (ΩV Function)
<Resistance>,<Temperature (NR3)> (Ω Function)
<Voltage>,<Temperature (NR3)> (V Function)

(3) When RR as the parameter is specified

<Resistance>,<Voltage>,<Route-Resistance (NR3)> (ΩV Function)
<Resistance>,<Route-Resistance (NR3)> (Ω Function)
<Voltage>,<Route-Resistance (NR3)> (V Function)

Note: Route-Resistance value is as follows:
the Source Hi, the Source Lo, the Sense Hi and the Sense Lo.

(4) When TEMPerature,RR as the parameters are specified

<Resistance>,<Voltage>,<Temperature>,<Route-Resistance> (ΩV Function)
<Resistance>,<Temperature>,<Route-Resistance> (Ω Function)
<Voltage>,<Temperature>,<Route-Resistance> (V Function)

Note: Route-Resistance value is as follows:
the Source Hi, the Source Lo, the Sense Hi and the Sense Lo.

See: “Measurement Value Formats” (p.15)

Description Switches from the Idle State to the Trigger Wait State, and then reads the measured values after the measurement is completed.

With the auto range enabled, the most suitable range is selected before measurement.
See: Data Exporting Methods (p.78), Triggering (p.44)

Trigger Source	Operation
INTernal	Triggers and continuously reads the measured values.
EXTernal	Triggers by TRIG signal input, and continuously reads the measured values.

Example

```
:TRIG:SOUR INT
:INIT:CONT OFF
:FUNC RV
:SYST:COMM:FORM FIX
:READ?
+1.00010E-03,+00.000001E+00
:READ? TEMP
+1.00010E-03,+00.000001E+00,+23.8E+00
```

The resistance measured value is 1.0001mΩ, voltage is 0.000001V, temperature is 23.8℃.

Note

- When :READ? is received, :INITiate:CONTinuous OFF is set automatically.
- The next command does not execute until measurement is finished. But *TRG or :ABORt is available to execute.
- When there is an internal trigger (trigger source <INTernal>), triggering immediately occurs, a single measurement is performed and enters the Idle State.
- When there is an external trigger (trigger source <EXTernal>), the Trigger Wait State is entered. After a trigger is received, a single measurement is performed and enters the Idle State.
- When there is an external trigger (trigger source <EXTernal>), the measured value’s response will be twice when the data output function is ON. Please switch the data output function OFF.
- It may be necessary to insert wait processing after panel load or range selection is executed. Wait time depends on Measurement device.
- When measuring sequence is operating by :INITiate[:IMMediate] or :READ?, an execution error occurs.

Abort Measurement

Syntax Query **:ABORt**

Description

When :TRIGger:SOURce EXTernal is set,
- Switches from the Trigger Wait State by :INITiate[:IMMediate] or :READ? to the Idle State.
- Switches from the Trigger Delay State or Analog Measurement State to the Trigger Wait State (:INITiate:CONTinuous ON) or the Idle State (:INITiate:CONTinuous OFF).

When :TRIGger:SOURce INTernal and :INITiate:CONTinuous OFF are set,
- Switches from the Trigger Delay State or Analog Measurement State to the Idle State.

See: Triggering (p.44)

Example

```
:READ?
:ABOR
```
Executes an abort.

Note When :TRIGger:SOURce INTernal and :INITiate:CONTinuous ON are set, an execution error occurs.

Set and Query the Output Format of Measured Values

Syntax
Command **:SYSTem:COMMunicate:FORMat** <FIX/FLOAT>
Query **:SYSTem:COMMunicate:FORMat?**
Response <FIX/FLOAT>

Description

Receives the values with the specific format when FETCh?, READ? or SYSTem:COMMunicate:DATAout ON is executed.

See: “Measurement Value Formats” (p.15)

Example

```
:SYST:COMM:FORM FIX
:SYST:COMM:FORM?
FIX
```

Set and Query the Temperature Unit

Syntax
Command **:TEMPerature:UNIT** <Celsius/Fahrenheit>
Query **:TEMPerature:UNIT?**
Response <CELSIUS/FAHRENHEIT>

Example

```
:TEMP:UNIT F
:TEMP:UNIT?
FAHRENHEIT
```

(3) Measurement Function

Set and Query the Measurement Function

Syntax

Command **:FUNCtion** <RV/R/V/RESistance/VOLTage>
Query **:FUNCtion?**
Response <RV/R/V>
<RV> = Displays the resistance, voltage, and temperature measured value.
<R/RESistance> = Displays the resistance and temperature measured value.
<V/VOLTage> = Displays the voltage and temperature measured value.

Example

```
:FUNC RV
:FUNC?
RV
```

Note

When :SYSTem:COMMunicate:BT3562A ON is set,
the response of query is <RV/RESISTANCE/VOLTAGE>.

(4) Measurement Speed

Set and Query the Measurement Speed

Syntax

Command **:SAMPle:RATE** <FAST1/FAST2/MEDIUM1/MEDIUM2/SLOW1/SLOW2/
EXFast/FAST/MEDium/SLOW>
FAST1 is set when the parameter is EXFast.
FAST2 is set when the parameter is FAST.
MEDIUM2 is set when the parameter is MEDium.
SLOW2 is set when the parameter is SLOW.
Query **:SAMPle:RATE?**
Response <FAST1/FAST2/MEDIUM1/MEDIUM2/SLOW1/SLOW2>

Example

```
:SAMP:RATE MEDIUM1
:SAMP:RATE?
MEDIUM1
```

Note

When :SYSTem:COMMunicate:BT3562A ON is set,
the response of query is <EXFAST/FAST/MEDIUM/SLOW>.
EXFAST is received when FAST1 is set.
FAST is received when FAST2 is set.
MEDIUM is received when MEDIUM1/MEDIUM2 is set.
SLOW is received when SLOW1/SLOW2 is set.

(5) Measurement Range

Set and Query the Resistance's Measurement Range

Syntax

Command **:RESistance:RANGe** <Measurement range>
Query **:RESistance:RANGe?**
Response <Measurement range>
Command
<Measurement range> = 3m/30m/300m/3/30/Resistance value of DUT(Device Under Test)

The parameter from -1.0 to 51.0 (NRf) [Ω] is available when Resistance value of DUT is used.

Query
<Measurement range> = +3.00000E-03/ +3.00000E-02/ +3.00000E-01/
+3.00000E+00/ +3.00000E+01

Example

```
:RES:RANG 300m
:RES:RANG?
300m
```

Note

- When the measurement range is set using the command, the instrument is set to the most suitable range for measuring the DUT if Resistance value of DUT is used as the parameter.
  Example)
  :RES:RANG 0.1
  :RES:RANG?
  +3.00000E-01
  300mΩ range is set for measuring the 0.1Ω DUT.

- When :RESistance:RANGe:AUTO ON is set,
  the response of query is the resistance's range selected by the measurement system.
- When :SYSTem:COMMunicate:BT3562A ON is set,
  the response of query is <3.0000E-3/30.000E-3/300.00E-3/3.0000E+0/30.000E+0>.

Set and Query the Measurement Auto Range

Syntax

Command **:RESistance:RANGe:AUTO** <1/0/ON/OFF>
Query **:RESistance:RANGe:AUTO?**
Response <ON/OFF>

Example

```
:RES:RANG:AUTO OFF
:RES:RANG:AUTO?
OFF
```

Note

The auto range setting is common setting between the resistance measurement and the voltage measurement.

Set and Query the Voltage's Measurement Range

Syntax

Command **:VOLTage:RANGe** <Measurement range>
Query **:VOLTage:RANGe?**
Response <Measurement range>
Command
<Measurement range> = 10V/100V/Voltage value of DUT(Device Under Test)

The parameter from -120.0 to 120.0 (NRf) [V] is available when Voltage value of DUT is used.

Query
<Measurement range> = +1.0000000E+01/ +1.0000000E+02

Example

```
:VOLT:RANG 100V
:VOLT:RANG?
+1.0000000E+02
```

Note

- When the measurement range is set using the command, the instrument is set to the most suitable range for measuring the DUT if Voltage value of DUT is used as the parameter.
  Example)
  :VOLT:RANG 6.0
  :VOLT:RANG?
  +1.0000000E+01
  10V range is set for measuring the 6V DUT.

- When :VOLTage:RANGe:AUTO ON is set,
  the response of query is the voltage's range selected by the measurement system.
- When :SYSTem:COMMunicate:BT3562A ON is set,
  the response of query is <10.00000E+0/100.0000E+0>.

Set and Query the Measurement Auto Range

Syntax

Command **:VOLTage:RANGe:AUTO** <1/0/ON/OFF>
Query **:VOLTage:RANGe:AUTO?**
Response <ON/OFF>

Example

```
:VOLT:RANG:AUTO OFF
:VOLT:RANG:AUTO?
OFF
```

Note

The auto range setting is common setting between the resistance measurement and the voltage measurement.

(6) Resistance's Measurement Current

Set and Query the Resistance's Measurement Current

Syntax
Command **:RESistance:CURRent** <HIGH/LOW>
Query **:RESistance:CURRent?**
Response <HIGH/LOW>

Description The resistance's measurement current is following.

Measurement range	Resistance's measurement current	
	HIGH	LOW
3mΩ	300mA	100mA
30mΩ	100mA	
300mΩ	10mA	
3Ω	1mA	
30Ω	100μA	

Example
:RES:CURR HIGH
:RES:CURR?
HIGH
The resistance's measurement current value is 300mA in 3mΩ range.

Note Only resistance's measurement current value in 3mΩ range can be set using the command.
100mA is fixed in 30mΩ range.
10mA is fixed in 300mΩ range.
1mA is fixed in 3Ω range.
100uA is fixed in 30Ω range.

(7) Resistance's Mutual Interference Reductions

Set and Query the Resistance's Mutual Interference Reduction Function

Syntax
Command **:RESistance:MIR:STATe** <1/0/ON/OFF>
Query **:RESistance:MIR:STATe?**
Response <ON/OFF>

Example
:RES:MIR:STAT ON
:RES:MIR:STAT?
ON

Set and Query the Role for the Resistance's Mutual Interference Reduction

Syntax
Command **:RESistance:MIR:ROLE** <PRIMARY/SECONDARY>
Query **:RESistance:MIR:ROLE?**
Response <PRIMARY/SECONDARY>

Example
:RES:MIR:ROLE PRIMARY
:RES:MIR:ROLE?
PRIMARY

(8) Resistance's Number of Digits

Set and Query the Number of Digits for the Resistance's Measurement

Syntax
Command **:RESistance:DIGits** <5/6 (NR1)>
Query **:RESistance:DIGits?**
Response <5/6 (NR1)>
<5> = HIGH RESOLUTION mode is OFF
<6> = HIGH RESOLUTION mode is ON

Example
:RES:DIG 6
:RES:DIG?
6

(9) Voltage's Input Impedance

Set and Query the Voltage's Input Impedance

Syntax
Command **:VOLTage:IMPedance** <10M/HIGH_Z>
Query **:VOLTage:IMPedance?** [10V]
Response <10M/HIGH_Z>

(1) When 10V as the parameter is not specified
The response of query is the voltage's input impedance in current voltage's range

(2) When 10V as the parameter is specified
The response of query is the voltage's input impedance in 10V range.

Example
:VOLT:IMP HIGH_Z
:VOLT:IMP? 10V
HIGH_Z
The voltage's input impedance is HIGH_Z in 10V range.

Note
Only voltage's input impedance in 10V range can be set using the command.
10M is fixed in 100V range.

(10) Voltage's Absolute Value

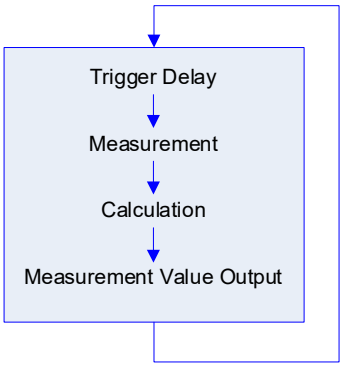
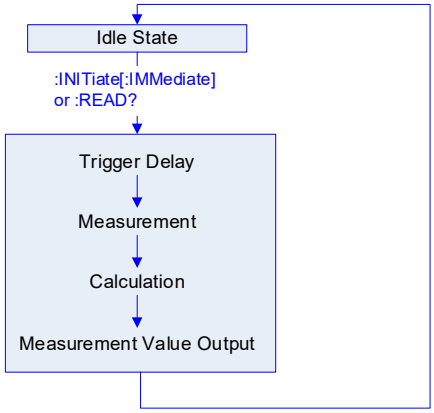
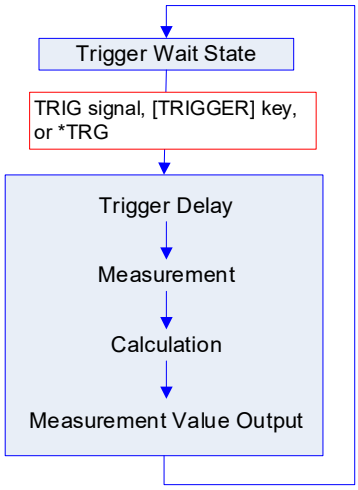
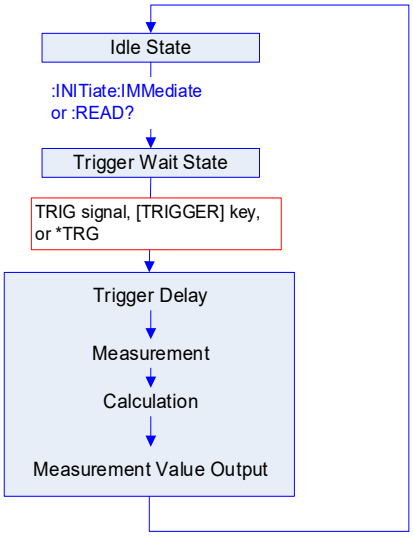
Set and Query the Voltage's Absolute Value Function

Syntax
Command **:VOLTage:ABSolute** <1/0/ON/OFF>
Query **:VOLTage:ABSolute?**
Response <ON/OFF>

Example
:VOLT:ABS ON
:VOLT:ABS?
ON

(11) Trigger

Relationship between Trigger Source and Continuous Measurement Operation
Operation depends on continuous measurement setting (:INITiate:CONTinuous) (p.45) and the trigger source setting (:TRIGger:SOURce) (p.45) as follows.

<table>
<tr><th colspan="2" rowspan="2">Measurement Flow</th><th colspan="2">Continuous Measurement Operation (Command-Specific Settings)</th></tr>
<tr><th>:INITiate:CONTinuous ON</th><th>:INITiate:CONTinuous OFF</th></tr>
<tr><td rowspan="2">Trigger Source</td><td>:TRIGger:SOURce INTernal</td><td>Free-Run state. Measurement continues automatically.
 
</td><td>Trigger by :INITIATE[:IMMediate] (or :READ?) command.
After measurement, enters the idle state.
 
</td></tr>
<tr><td>:TRIGger:SOURce EXTernal</td><td>Trigger by TRIG signal, [TRIGGER] key, or ∗TRG command.
After measurement, enters the trigger wait state.
 
</td><td>Enters the trigger wait state by :INITiate[:IMMediate] (or :READ?) command.
Trigger by TRIG signal, [TRIGGER] key or ∗TRG command.
After measurement, enters the idle state.
 
</td></tr>
</table>

If the :INITiate:CONTinuous OFF has been set, when operation is returned to the Local state or the power is turned off and then back on, the :INITiate:CONTinuous ON state occurs.

See: “Return to the Local Control State” (p.60)
Exporting measured values: “Data Exporting Methods” (p.78)

Set and Query the Continuous Measurement

Syntax
Command **:INITiate:CONTinuous** <1/0/ON/OFF>
Query **:INITiate:CONTinuous?**
Response <ON/OFF>

Description
- When :INITiate:CONTinuous ON:
  After measurement, enters the Trigger Wait State. When there is an internal trigger (trigger source <INTernal>), the next trigger is promptly generated (Free-Run State).
- When :INITiate:CONTinuous OFF:
  After measurement, enters the Idle State instead of the Trigger Wait State and triggering is ignored in the Idle State. When there is an internal trigger (trigger source <INTernal>), after :INITiate[:IMMediate] or :READ? is executed, a single measurement is performed. When there is an external trigger (trigger source <EXTernal>), after :INITiate[:IMMediate] or :READ? is executed, enters the Trigger Wait State.

See: Triggering (p.44)

Example
```
:INIT:CONT OFF
:INIT:CONT?
OFF
```

Note
:INITiate:CONTinuous ON is set when exiting from the Remote State.

Initiate the Trigger Wait State

Syntax
Command **:INITiate[:IMMediate]**

Description
Enters the Trigger Wait State from the Idle State.

Example
Disable continuous measurement, and read one value for each trigger event.

:TRIG:SOUR INT	Triggers immediately when entering the Trigger Wait State.
:INIT:CONT OFF	Disables continuous measurement.
:INIT	Enters the Trigger Wait State then triggers immediately.

Note
- When :INITiate[:IMMediate] is received, :INITiate:CONTinuous OFF is set automatically.
- When there is an internal trigger (trigger source <INTernal>), triggering immediately occurs, a single measurement is performed and enters the Idle State.
- When there is an external trigger (trigger source <EXTernal>), the Trigger Wait State is entered. After a trigger is received, a single measurement is performed and enters the Idle State.
- When measuring sequence is operating by :INITiate[:IMMediate] or :READ?, an execution error occurs.

Set and Query the Trigger Source

Syntax
Command **:TRIGger:SOURce** <INTernal/EXTernal/IMMediate>
Query **:TRIGger:SOURce?**
Response <INTERNAL/EXTERNAL>
<INTernal/IMMediate> = Internal triggering
<EXTernal> = External triggering

Example
```
:TRIG:SOUR INT
:TRIG:SOUR?
INT
```

Note
When :SYSTem:COMMunicate:BT3562A ON is set, the response of query is <IMMEDIATE/EXTERNAL>.

Set and Query the Trigger Delay Function

Syntax
Command **:TRIGger:DELay:STATe** <1/0/ON/OFF>
Query **:TRIGger:DELay:STATe?**
Response <ON/OFF>

Example
```
:TRIG:DEL:STAT ON
:TRIG:DEL:STAT?
ON
```

Set and Query the Period of Trigger Delay

Syntax
Command **:TRIGger:DELay** <Delay time>
Query **:TRIGger:DELay?**
Response <Delay time>
<Delay time> = 0.0 to 10.0 (NRf) [sec]

Example
```
:TRIG:DEL 0.1
:TRIG:DEL?
1.00000000E-01
```

Note
When :SYSTem:COMMunicate:BT3562A ON is set,
the response of query is <Delay time> = 0.000 to 10.000 (NR2) [sec].

(12) Averaging Functions

Set and Query the Averaging Function

Syntax
Command **:CALCulate:AVERage:STATe** <1/0/ON/OFF>
Query **:CALCulate:AVERage:STATe?**
Response <ON/OFF>

Example
```
:CALC:AVER:STAT ON
:CALC:AVER:STAT?
ON
```

Set and Query the Number of Averaging Calculation Data

Syntax
Command **:CALCulate:AVERage:COUNt** <1 to 256 (NR1)>
Query **:CALCulate:AVERage:COUNt?**
Response <1 to 256 (NR1)>

Example
```
:CALC:AVER:COUN 50
:CALC:AVER:COUN?
50
```

(13) Zero Display Width

Set and Query the Measured Values' Zero Display Width

Syntax

Command **:CALCulate:ZERODisplay:WIDTH** <1/0/ON/OFF>
Query **:CALCulate:ZERODisplay:WIDTH?**
Response <ON/OFF>
<ON> = The measured values within the zero display range is treated as zero.
<OFF> = The measured value is not treated as zero to display small values.

Description

The zero display range is the following.

Hi-Res ON	The zero display range is within	±0.08 μ Ω	in 3mΩ (300mA)	range.
	The zero display range is within	±0.50 μ Ω	in 3mΩ (100mA)	range.
	The zero display range is within	±0.5 μ Ω	in 30mΩ	range.
	The zero display range is within	±5 μ Ω	in 300mΩ	range.
	The zero display range is within	±50 μ Ω	in 3Ω	range.
	The zero display range is within	±0.5mΩ	in 30Ω	range.
Hi-Res OFF	The zero display range is within	±0.1 μ Ω	in 3mΩ (300mA)	range.
	The zero display range is within	±0.5 μ Ω	in 3mΩ (100mA)	range.
	The zero display range is within	±1 μ Ω	in 30mΩ	range.
	The zero display range is within	±10 μ Ω	in 300mΩ	range.
	The zero display range is within	±100 μ Ω	in 3Ω	range.
	The zero display range is within	±1mΩ	in 30Ω	range.
BT6065	The zero display range is within	±20 μ V	in 10V	range.
	The zero display range is within	±0.6mV	in 100V	range.
BT6075	The zero display range is within	±11 μ V	in 10V	range.
	The zero display range is within	±0.60mV	in 100V	range.

(The zero display range is within the values of digit error in SLOW2 sampling rate and each range.)

Example

```
:CALC:ZERO:WIDTH ON
:CALC:ZERO:WIDTH?
ON
```

(14) Comparator

Set and Query the Comparator Function

Syntax
Command **:COMParator:LIMit:STATe** <1/0/ON/OFF>
Query **:COMParator:LIMit:STATe?**
Response <ON/OFF>

Example
```
:COMP:LIM:STAT ON
:COMP:LIM:STAT?
ON
```

Set and Query the Judgement Buzzer

Syntax
Command **:COMParator:LIMit:BEEPer** <OFF/HL/IN/BOTH1/BOTH2>
Query **:COMParator:LIMit:BEEPer?**
Response <OFF/HL/IN/BOTH1/BOTH2>

<OFF> = No beep sounds.
<HL> = The beeper sounds repeatedly upon Hi or Lo judgements.
<IN> = The beeper sounds continuously upon In judgements.
<BOTH1> = The beeper sounds repeatedly upon Hi or Lo judgements, and continuously upon In judgements.
<BOTH2> = The beeper sounds repeatedly upon Hi or Lo judgements, and once briefly upon In judgements.

Example
```
:COMP:LIM:BEEP IN
:COMP:LIM:BEEP?
IN
```

Set and Query the Voltage's Absolute Value Judgment Function

Syntax
Command **:COMParator:LIMit:ABSolute** <1/0/ON/OFF>
Query **:COMParator:LIMit:ABSolute?**
Response <ON/OFF>

Example
```
:COMP:LIM:ABS ON
:COMP:LIM:ABS?
ON
```

Note When :COMParator:LIMit:ABSolute ON is set, the absolute value is only taken for voltage measured values.

Set and Query the Resistance's Upper Threshold

Syntax Command **:COMParator:LIMit:RESistance:UPPer** <Upper threshold>
Query **:COMParator:LIMit:RESistance:UPPer?**
Response <Upper threshold>
<Upper threshold> = -1.0 to +51.0 (NRf) [Ω]

Example
:COMP:LIM:RES:UPP 0.28593
:COMP:LIM:RES:UPP?
+2.85930000E-01
The resistance's upper threshold is 285.93 mΩ.

Note <Upper threshold> smaller than :COMParator:LIMit:RESistance:LOWer <Lower threshold> cannot be set.

Set and Query the Voltage's Upper Threshold

Syntax Command **:COMParator:LIMit:VOLTage:UPPer** <Upper threshold>
Query **:COMParator:LIMit:VOLTage:UPPer?**
Response <Upper threshold>
<Upper threshold> = -120.0 to +120.0 (NRf) [V]

Example
:COMP:LIM:VOLT:UPP 38.0
:COMP:LIM:VOLT:UPP?
+3.80000000E+01
The voltage's upper threshold is 38.0 V.

Note <Upper threshold> smaller than :COMParator:LIMit:VOLTage:LOWer <Lower threshold> cannot be set.

Set and Query the Resistance's Lower Threshold

Syntax Command **:COMParator:LIMit:RESistance:LOWer** <Lower threshold>
Query **:COMParator:LIMit:RESistance:LOWer?**
Response <Lower threshold>
<Lower threshold> = -1.0 to +51.0 (NRf) [Ω]

Example
:COMP:LIM:RES:LOW 0.28406
:COMP:LIM:RES:LOW?
+2.84060000E-01
The resistance's lower threshold is 284.06 mΩ.

Note <Lower threshold> larger than :COMParator:LIMit:RESistance:UPPer <Upper threshold> cannot be set.

Set and Query the Voltage's Lower Threshold

Syntax Command **:COMParator:LIMit:VOLTage:LOWer** <Lower threshold>
Query **:COMParator:LIMit:VOLTage:LOWer?**
Response <Lower threshold>
<Lower threshold> = -120.0 to +120.0 (NRf) [V]

Example
:COMP:LIM:VOLT:LOW 36.0
:COMP:LIM:VOLT:LOW?
+3.60000000E+01
The voltage's lower threshold is 36.0 V.

Note <Lower threshold> larger than :COMParator:LIMit:VOLTage:UPPer <Upper threshold> cannot be set.

Set and Query the Route-Resistance's Comparator Function

Syntax
Command **:COMParator:LIMit:RR:STATe** <1/0/ON/OFF>
Query **:COMParator:LIMit:RR:STATe?**
Response <ON/OFF>

Example
:COMP:LIM:RR:STAT ON
:COMP:LIM:RR:STAT?
ON

Set and Query the Route-Resistance's Fail Threshold

Syntax
Command **:COMParator:LIMit:RR:FAIL** <Fail threshold>
Query **:COMParator:LIMit:RR:FAIL?**
Response <Fail threshold>
<Fail threshold> = -10.0 to +50.0 (NRf) [Ω]

Example
:COMP:LIM:RR:FAIL 6.0
:COMP:LIM:RR:FAIL?
+6.00000000E+00
The route-resistance's fail threshold is 6.0 Ω.

Note
<Fail threshold> smaller than :COMParator:LIMit:RR:WARNing <Warning threshold> cannot be set.

Set and Query the Route-Resistance's Warning Threshold

Syntax
Command **:COMParator:LIMit:RR:WARNing** <Warning threshold>
Query **:COMParator:LIMit:RR:WARNing?**
Response <Warning threshold>
<Warning threshold> = -10.0 to +50.0 (NRf) [Ω]

Example
:COMP:LIM:RR:WARN 5.0
:COMP:LIM:RR:WARN?
+5.00000000E+00
The route-resistance's warning threshold is 5.0 Ω.

Note
<Warning threshold> larger than :COMParator:LIMit:RR:FAIL <Fail threshold> cannot be set.

Clear the Comparator Event Status Register

Syntax Command **:COMParator:LIMit:CLEar**

Description Clears the comparator-related Condition Register bits (bit 0 to bit 10, bit 14, and bit 15) of the Status Query Register Group (:STATus:QUEStionable:CONDition?).

Query the Comparator Result

Syntax Query **:COMParator:LIMit:RESistance:RESult?**
Response <HI/IN/LO/ERR/OFF>

Description Reads the most recent judgement result.

See: "Update Timing of Comparator Result" (p.80)

Example
```
:COMP:LIM:RES:RES?
IN
```
The resistance's comparator result is IN.

Note When there are no measured values, the response is ERR.

Query the Comparator Result

Syntax Query **:COMParator:LIMit:VOLTage:RESult?**
Response <HI/IN/LO/ERR/OFF>

Description Reads the most recent judgement result.

See: "Update Timing of Comparator Result" (p.80)

Example
```
:COMP:LIM:VOLT:RES?
IN
```
The voltage's comparator result is IN.

Note When there are no measured values, the response is ERR.

Query the Comparator Result

Syntax Query **:COMParator:LIMit:RR:RESult?**
Response <PASS/WARNING/FAIL/ERR/OFF>

Description Reads the most recent judgement result.

See: "Update Timing of Comparator Result" (p.80)

Example
```
:COMP:LIM:RR:RES?
PASS
```
The route-resistance's comparator result is IN.

Note When there are no measured values, the response is ERR.

(15) Adjustment

Set and Query the Adjustment Type

Syntax Command **:ADJust:TYPE** <ZERO/REFerential/OFF>
Query **:ADJust:TYPE?**
Response <ZERO/REFERENTIAL/OFF>
<ZERO> = Uses the zero-adjustment values
<REFERENTIAL> = Uses the referential adjustment values
<OFF> = NOT use adjustment values

Example
```
:ADJ:TYPE ZERO
:ADJ:TYPE?
ZERO
```

Set and Query the Zero-Adjustment Mode

Syntax Command **:ADJust:ZERO:MODE** <SINGle/MULTiple>
Query **:ADJust:ZERO:MODE?**
Response <SINGLE/MULTIPLE>

Example
```
:ADJ:ZERO:MODE SING
:ADJ:ZERO:MODE?
SINGLE
```

Note When other than :ADJust:TYPE ZERO is set, an execution error occurs.

Execute the Zero-Adjustment for Single Mode and Query the Result

Syntax Query **:ADJust:ZERO:SINGle?** [ALL]
Response <PASS/FAIL>

(1) When ALL as the parameter is not specified
The zero-adjustment is executed for the current range.
When the auto-range is set, the zero-adjustment is executed for all ranges.

(2) When ALL as the parameter is specified
The zero-adjustment is executed for all ranges.

Example
```
:ADJ:ZERO:SING? ALL
PASS
```

Note When :SYSTem:COMMunicate:BT3562A ON is set, the response of query is <0/1>.
<0> = PASS
<1> = FAIL

Execute the Zero-Adjustment for Multiple Mode and Query the Result

Syntax Query **:ADJust:ZERO:MULTiple?** <Channel No.>
Response <PASS/FAIL>
<Channel No.> = 1 to 528 (NR1)

Example
```
:ADJ:ZERO:MULT? 1
PASS
```

Note When :SYSTem:COMMunicate:BT3562A ON is set, the response of query is <0/1>.
<0> = PASS
<1> = FAIL

Query the Completed Channel Number for Zero-Adjustment Multiple Mode

Syntax Query **:ADJust:ZERO:MULTiple:DONE?**
Response <Completed MIN Channel No.>,<Completed MAX Channel No.>
<Completed Channel No.> = 1 to 528 (NR1)

Description The response of query is the minimum and maximum channel number among completed channels with :ADJust:ZERO:MULTiple?.
There is at least one zero-adjustment value for the multiple mode between <Completed MIN Channel No.> and <Completed MAX Channel No.>.

Example
```
:ADJ:ZERO:MULT:DONE?
1,7
```

Note When there is no completed channel for zero-adjustment multiple mode, the response of query is -1,-1.

Delete the Zero-Adjustment Values for Multiple Mode

Syntax Command **:ADJust:ZERO:MULTiple:CLEar**

Reflect Setting Conditions when Zero-Adjustment Multiple Mode was Executed

Syntax Query **:ADJust:ZERO:MULTiple:REFLect**

Description Updates the setting conditions of measurement function, measurement range, resistance's measurement current and voltage's input impedance when zero-adjustment multiple mode was executed.

Note
- When other than :ADJust:TYPE ZERO is set, an execution error occurs.
- When there is no completed channel for zero-adjustment multiple mode, an execution error occurs.

Set and Query the Channel to Use for Zero-Adjustment Multiple Mode

Syntax Command **:ADJust:ZERO:MULTiple:CH** <Channel No.>
Query **:ADJust:ZERO:MULTiple:CH?**
Response <Channel No.>
<Channel No.> = 1 to 528 (NR1)

Example
```
:ADJ:ZERO:MULT:CH 1
:ADJ:ZERO:MULT:CH?
1
```

Note
- When other than :ADJust:TYPE ZERO is set, an execution error occurs.
- When there is no completed channel for zero-adjustment multiple mode, an execution error occurs.

Set and Query the Start and End Channels when Executing Zero-Adjustment Multiple Mode

Syntax Command **:ADJust:ZERO:MULTiple:LIST** <Start Channel No.>,<End Channel No.>
Query **:ADJust:ZERO:MULTiple:LIST?**
Response <Start Channel No.>,<End Channel No.>
<Channel No.> = 1 to 528 (NR1)

Description The range when executing zero-adjustment multiple mode with the [ADJUST] key is <Start Channel No.> to <End Channel No.>.

Example
```
:ADJ:ZERO:MULT:LIST 1,8
:ADJ:ZERO:MULT:LIST?
1,8
```

Execute the Zero-Adjustment at Base Position for Referential Adjustment and Query the Result

Syntax Query **:ADJust:REFerential:BASE:ZERO?**
Response <PASS/FAIL>

Example
:ADJ:REF:BASE:ZERO?
PASS

Note
- When :FUNCtion V is set, an execution error occurs.
- When :SYSTem:COMMunicate:BT3562A ON is set, the response of query is <0/1>.
  <0> = PASS
  <1> = FAIL

Execute the Measurement at Base Position for Referential Adjustment and Query the Result

Syntax Query **:ADJust:REFerential:BASE:DUT?**
Response <PASS/FAIL>

Example
:ADJ:REF:BASE:DUT?
PASS

Note
- When the zero-adjustment at base position is not completed, an execution error occurs.
- When current setting conditions is different from the setting conditions when the zero-adjustment at base position was executed, an execution error occurs.
  Setting conditions : Measurement function, Measurement range, Resistance's measurement current and Voltage's input impedance
- When :FUNCtion V is set, an execution error occurs.
- When :SYSTem:COMMunicate:BT3562A ON is set, the response of query is <0/1>,<Measured value>.
  <0> = PASS
  <1> = FAIL

Query the Existence of a Measured Value at Base Position for Referential Adjustment

Syntax Query **:ADJust:REFerential:BASE:EXISt?**
Response <EXIST/NONE>

Example
:ADJ:REF:BASE:EXIS?
EXIST
Both adjustment and measurement at base position for referential adjustment are completed.

Execute the Referential Adjustment and Query the Result

Syntax Query **:ADJust:REFerential:POSItion?** <Position No.>
Response <PASS/FAIL>
<Position No.> = 1 to 528 (NR1)

Example
:ADJ:REF:POSI? 1
PASS

Note
- When the measurement at base position is not completed, an execution error occurs.
- When current setting conditions is different from the setting conditions when the measurement at base position was executed, an execution error occurs.
  Setting conditions : Measurement function, Measurement range, Resistance's measurement current and Voltage's input impedance
- When :FUNCtion V is set, an execution error occurs.
- When :SYSTem:COMMunicate:BT3562A ON is set, the response of query is <0/1>,<Measured value>.
  <0> = PASS
  <1> = FAIL

Query the Completed Position Number for Referential Adjustment

Syntax Query **:ADJust:REFerential:POSItion:DONE?**
Response <Completed MIN Position No.>,<Completed MAX Position No.>
<Completed Position No.> = 1 to 528 (NR1)

Description The response of query is the minimum and maximum position number among completed positions with :ADJust:REFerential:POSItion?.
There is at least one referential adjustment value between <Completed MIN Position No.> and <Completed MAX Position No.>.

Example
```
:ADJ:REF:POSI:DONE?
1,7
```

Note When there is no completed position for referential adjustment, the response of query is -1,-1.

Delete the Referential Adjustment Values

Syntax Command **:ADJust:REFerential:POSItion:CLEar**

Reflect Setting Conditions when Referential Adjustment was Executed

Syntax Query **:ADJust:REFerential:POSItion:REFLect**

Description Updates the setting conditions of measurement function, measurement range, resistance's measurement current and voltage's input impedance when referential adjustment was executed.

Note
- When other than :ADJust:TYPE REFerential is set, an execution error occurs.
- When there is no completed position for referential adjustment, an execution error occurs.

Set and Query the Position to Use for Referential Adjustment

Syntax Command **:ADJust:REFerential:POSItion:NO** <Position No.>
Query **:ADJust:REFerential:POSItion:NO?**
Response <Position No.>
<Position No.> = 1 to 528 (NR1)

Example
```
:ADJ:REF:POSI:NO 1
:ADJ:REF:POSI:NO?
1
```

Note
- When other than :ADJust:TYPE REFerential is set, an execution error occurs.
- When there is no completed position for referential adjustment, an execution error occurs.

Set and Query the Start and End Positions when Executing Referential Adjustment

Syntax Command **:ADJust:REFerential:POSItion:LIST** <Start Position No.>,<End Position No.>
Query **:ADJust:REFerential:POSItion:LIST?**
Response <Start Position No.>,<End Position No.>
<Position No.> = 1 to 528 (NR1)

Description The range when executing referential adjustment with the [ADJUST] key is <Start Position No.> to <End Position No.>.

Example
```
:ADJ:REF:POSI:LIST 1,8
:ADJ:REF:POSI:LIST?
1,8
```

(16) Self-Calibration

Execute the Self-Calibration and Query the Result

Syntax Query **:SYSTem:CALibration?** <RV/R/V>
Response <PASS/FAIL>

Example
```
:SYST:CAL? RV
PASS
```

Note
- When :INITiate:CONTinuous OFF is set and the Idle State, an execution error occurs.
- When :SYSTem:CALibration? <RV/R/V> is received during measurement, the measurement is aborted immediately and then the self-calibration is executed.

Set and Query the Voltage's AUTO Self-Calibration

Syntax Command **:SYSTem:CALibration:AUTO** <1/0/ON/OFF>
Query **:SYSTem:CALibration:AUTO?**
Response <ON/OFF>
<ON> = the Voltage's Self-Calibration mode is AUTO
<OFF> = the Voltage's Self-Calibration mode is MANUAL

Example
```
:SYST:CAL:AUTO ON
:SYST:CAL:AUTO?
ON
```

Note Even when :SYSTem:CALibration:AUTO ON is set, the voltage's self-calibration can be executed manually with :SYSTem:CALibration? <RV/R/V>.

(17) Saving and Loading Measurement Conditions

Save and Load the Measurement Conditions

Syntax Command ***SAV** <Panel No.>
***RCL** <Panel No.>

Note For details, check the *SAV and *RCL standard commands.

Delete the Panel

Syntax Command **:SYSTem:PANel:CLEar** <Panel No.>
<Panel No.> = 1 to 6 (NR1)

(18) Clock

Set and Query the System Date

Syntax
Command **:SYSTem:DATE** <Year>,<Month>,<Day>
Query **:SYSTem:DATE?**
Response <Year>,<Month>,<Day>
<Year> = 22 to 99
<Month> = 01 to 12
<Day> = 01 to 31

Example
:SYST:DATE 24,9,2
:SYST:DATE?
24,9,2
Set the date to September 2, 2024.

Note When a non-existent date (such as 24,6,31) is set, a parameter error occurs.

Set and Query the System Time

Syntax
Command **:SYSTem:TIME** <Hour>,<Minute>,<Second>
Query **:SYSTem:TIME?**
Response <Hour>,<Minute>,<Second>
<Hour> = 00 to 23
<Minute> = 00 to 59
<Second> = 00 to 59

Example
:SYST:TIME 23,9,0
:SYST:TIME?
23,9,53
The time is 23:9 and 53 seconds.

(19) Key-Lock

Set and Query the Key-Lock State

Syntax
Command **:SYSTem:KLOCk** <1/0/ON/OFF>
Query **:SYSTem:KLOCk?**
Response <ON/OFF>

Example
:SYST:KLOC ON
:SYST:KLOC?
ON

Note When KEY_LOCK signal is ON,
the key-lock state is continued even if :SYSTem:KLOCk OFF is set.

(20) Key-Beeper

Set and Query the Key-Beeper State

Syntax
Command **:SYSTem:BEEPer:KEY** <1/0/ON/OFF >
Query **:SYSTem:BEEPer:KEY?**
Response <ON/OFF>

Example
:SYST:BEEP:KEY ON
:SYST:BEEP:KEY?
ON

(21) Line Frequency

Set and Query the Line Frequency

Syntax
Command **:SYSTem:LFRequency** <AUTO/50/60>
Query **:SYSTem:LFRequency?**
Response <AUTO/50/60>

Example
```
:SYST:LFR 50
:SYST:LFR?
50
```

(22) Advanced Mode

Set and Query the Advanced Mode State

Syntax
Command **:SYSTem:ADVanced:STATe** <1/0/ON/OFF>
Query **:SYSTem:ADVanced:STATe?**
Response <ON/OFF>

Description
When :SYSTem:ADVanced:STATe ON is set,
:DISPlay:VIEW ADVanced command is available.

Example
```
:SYST:ADV:STAT ON
:SYST:ADV:STAT?
ON
```

(23) Reset

Execute Normal Reset

Syntax
Command **:SYSTem:PRESet**

Description
Resets the instrument to its initial state.
Same function as ∗RST.

Note
- The communication conditions, temperature unit, clock, the panel data and the adjustment values are not initialized.
- The Event Status Register is not cleared.
- When :SYSTem:RESet is executed, the panel data and adjust values are also initialized.

Execute System Reset

Syntax
Command **:SYSTem:RESet**

Description
Resets the instrument to its initial state except for the communication conditions, temperature unit and clock.

Note
- The Event Status Register is not cleared.
- When :SYSTem:PRESet is executed, the panel data and adjust values are not initialized.

(24) Display Settings

Set and Query the Screen Brightness

Syntax
Command **:DISPlay:BACKlight** <Brightness>
Query **:DISPlay:BACKlight?**
Response <Brightness>
<Brightness> = 0 to 100 (NR1) [%]

Example
:DISP:BACK 50
:DISP:BACK?
50

Set and Query the Display Type

Syntax
Command **:DISPlay:VIEW** <NUMeric/RR/ADJust/COMParator/CONFiguration/ADVanced>
Query **:DISPlay:VIEW?**
Response <NUMERIC/RR/ADJUST/COMPARATOR/CONFIGURATION/ADVANCED>
<NUMeric> = Displays measured values only.
<RR> = Displays route-resistance monitor.
<ADJust> = Displays adjustment status monitor.
<COMParator> = Displays comparator threshold monitor.
<CONFiguration> = Displays configuration list monitor.
<ADVanced> = Displays advanced monitor.

Example
:DISP:VIEW NUM
:DISP:VIEW?
NUMERIC

Note
When :SYSTem:ADVanced:STATe OFF is set,
:DISPlay:VIEW ADVanced command cannot be used. An execution error occurs.

Set and Query the Screen Saver State

Syntax
Command **:DISPlay:SAVEr:STATe** <1/0/ON/OFF>
Query **:DISPlay:SAVEr:STATe?**
Response <ON/OFF>

Example
:DISP:SAVE:STAT ON
:DISP:SAVE:STAT?
ON

Set and Query the Waiting Time for Screen Saver

Syntax
Command **:DISPlay:SAVEr:WAIT** <Waiting time>
Query **:DISPlay:SAVEr:WAIT?**
Response <Waiting time>
<Waiting time> = 1 to 60 (NR1) [min]

Example
:DISP:SAVEr:WAIT 10
:DISP:SAVEr:WAIT?
10

Set and Query the Method to Return from Screen Saver Mode with Communication

Syntax
Command **:DISPlay:SAVEr:COMMunicate:WAKE** <1/0/ON/OFF>
Query **:DISPlay:SAVEr:COMMunicate:WAKE?**
Response <ON/OFF>

Example
:DISP:SAVE:COMM:WAKE ON
:DISP:SAVE:COMM:WAKE?
ON

Open the Screen

Syntax Command **:DISPlay:SAVEr:CLEar**

(25) Communication Settings

Return to the Local Control State

Syntax Command **:SYSTem:LOCal**

Description Disables communications remote control and re-enables local control.
The panel keys are re-enabled.

Example
```
:SYST:LOC
```

Set and Query the Communications Interface

Syntax Command **:SYSTem:COMMunicate** < USB/LAN/RS232c/MMEMory>
Query **:SYSTem:COMMunicate?**
Response <USB/LAN/RS232C>
<USB> = USB (COM mode)
<LAN> = LAN
<RS232c> = RS-232C
<MMEMory> = USB memory (MEM mode)

Note The interface is changed immediately after the command is received.

Execute and Query the Command Monitoring State

Syntax Command **:SYSTem:COMMunicate:MONitor** <1/0/ON/OFF>
Query **:SYSTem:COMMunicate:MONitor?**
Response <ON/OFF>

Example
```
:SYST:COMM:MON ON
:SYST:COMM:MON?
ON
```

Set and Query the IP Address [LAN]

Syntax
Command **:SYSTem:COMMunicate:LAN:IPADdress** <IP address>
Query **:SYSTem:COMMunicate:LAN:IPADdress?**
Response <IP address>
<IP address> = nnn,nnn,nnn,nnn

Example
```
:SYST:COMM:LAN:IPAD 192,168,1,2
:SYST:COMM:LAN:UPD
:SYST:COMM:LAN:IPAD?
192,168,1,2
```

Note After :SYSTem:COMMunicate:LAN:IPADdress <IP address> is sent, the <IP address> is not reflected on the instrument until :SYSTem:COMMunicate:LAN:UPDate is executed.

Set and Query the LAN Port [LAN]

Syntax
Command **:SYSTem:COMMunicate:LAN:CONTrol** <Port No.>
Query **:SYSTem:COMMunicate:LAN:CONTrol?**
Response <Port No.>
<Port No.> = 1 to 65535

Example
```
:SYST:COMM:LAN:CONT 6065
:SYST:COMM:LAN:UPD
:SYST:COMM:LAN:CONT?
6065
```

Note
- After :SYSTem:COMMunicate:LAN:CONTrol <Port No.> is sent, the <Port No.> is not reflected on the instrument until :SYSTem:COMMunicate:LAN:UPDate is executed.
- <Port No.> = 80 are not available. An execution error occurs.

Set and Query the Sub-Net Mask [LAN]

Syntax
Command **:SYSTem:COMMunicate:LAN:SMASk** <Sub-net mask>
Query **:SYSTem:COMMunicate:LAN:SMASk?**
Response <Sub-net mask>
<Sub-net mask> = nnn,nnn,nnn,nnn

Example
```
:SYST:COMM:LAN:SMAS 255,255,255,0
:SYST:COMM:LAN:UPD
:SYST:COMM:LAN:SMAS?
255,255,255,0
```

Note After :SYSTem:COMMunicate:LAN:SMASk <Sub-net mask> is sent, the <Sub-net mask> is not reflected on the instrument until :SYSTem:COMMunicate:LAN:UPDate is executed.

Set and Query the Default Gateway [LAN]

Syntax
Command **:SYSTem:COMMunicate:LAN:GATeway** <Address>
Query **:SYSTem:COMMunicate:LAN:GATeway?**
Response <Address>
<Address> = nnn,nnn,nnn,nnn

Example
```
:SYST:COMM:LAN:GAT 192,168,1,100
:SYST:COMM:LAN:UPD
:SYST:COMM:LAN:GAT?
192,168,1,100
```

Note After :SYSTem:COMMunicate:LAN:GATeway <Address> is sent, the <Address> is not reflected on the instrument until :SYSTem:COMMunicate:LAN:UPDate is executed.

Reflect the LAN Settings [LAN]

Syntax Command **:SYSTem:COMMunicate:LAN:UPDate**

Example
```
:SYST:COMM:LAN:IPAD 192,168,1,2
:SYST:COMM:LAN:UPD

:SYST:COMM:LAN:IPAD?
192,168,1,2
```

Note After :SYSTem:COMMunicate:LAN:IPADdress, :SYSTem:COMMunicate:LAN:CONTrol, :SYSTem:COMMunicate:LAN:GATeway or :SYSTem:COMMunicate:LAN:SMASk are sent, the <LAN settings> are not reflected on the instrument until :SYSTem:COMMunicate:LAN:UPDate is executed.

Query the MAC Address [LAN]

Syntax Query **:SYSTem:COMMunicate:LAN:MAC?**
Response <MAC address>

Example
```
:SYST:COMM:LAN:MAC?
00-01-67-21-03-85
```

Set and Query the Baud Rate [RS-232C]

Syntax Command **:SYSTem:COMMunicate:RS232C:SPEed** <Baud rate>
Query **:SYSTem:COMMunicate:RS232C:SPEed?**
Response <Baud rate>
<Baud rate> = 9600/19200/38400

Note The <Baud rate> is reflected immediately after the command is received.

(26) Data Output Settings

Set and Query the Measurement-Synchronized Data Output Function

Syntax

Command **:SYSTem:COMMunicate:DATAout** <1/0/ON/OFF>
Query **:SYSTem:COMMunicate:DATAout?**
Response <ON/OFF>

Description

<ON> = When the external trigger (trigger source <EXTernal>) is set, measured values are automatically sent when trigger measurements are completed. When the internal trigger (trigger source <INTernal>) is set, measured values are automatically sent when ∗TRIG command is received, the TRIG signal is input from the external I/O or the [TRIGGER] key is pressed.
<OFF> = Measured values are not automatically sent.

Note

USB(COM mode), LAN, and RS-232C interface is available for the data output function.

Set and Query the Type of Items for the Measurement-Synchronized Data Output Function

Syntax

Command **:SYSTem:COMMunicate:DATAout:FORMat** <Type of items>
Query **:SYSTem:COMMunicate:DATAout:FORMat?**
Response <Type of items>

<Type of items> = 0/1/2/3
<0> = Resistance value + Voltage value
<1> = Resistance value + Voltage value + Temperature value
<2> = Resistance value + Voltage value + Route-Resistance value
<3> = Resistance value + Voltage value + Temperature value + Route-Resistance value

Note: Route-Resistance value is as follows:
the Source Hi, the Source Lo, the Sense Hi and the Sense Lo.

Description

Sets the type of items for measurement-synchronized data output function.

```
:INIT:CONT OFF
:TRIG:SOUR EXT
:SYST:COMM:DATA ON

:SYST:COMM:DATA:FORM 0
:INIT;∗TRG
+0.00890E-03,+00.000001E+00

:SYST:COMM:DATA:FORM 1
:INIT;∗TRG
+0.00890E-03,+00.000001E+00,+25.3E+00

:SYST:COMM:DATA:FORM 2
:INIT;∗TRG
+0.00890E-03,+00.000001E+00,+00.1E+00,+00.1E+00,+01.9E+00,+02.3E+00

:SYST:COMM:DATA:FORM 3
:INIT;∗TRG
+0.00890E-03,+00.000001E+00,+25.3E+00,+00.1E+00,+00.1E+00,+01.9E+00,+02.3E+00
```

(27) Command Response Settings

Set and Query the Header Presence

Syntax
Command **:SYSTem:COMMunicate:HEADer** <1/0/ON/OFF>
Query **:SYSTem:COMMunicate:HEADer?**
Response <ON/OFF>

Example

```
:SYST:COMM:HEAD ON
:SYST:COMM:HEAD?
:SYSTEM:COMMUNICATE:HEADER ON
:SYST:COMM:HEAD OFF
:SYST:COMM:HEAD?
OFF
```

Note When power-on, *RST, :SYSTem:PRESet or :SYSTem:RESet is executed, Header OFF is selected.

Set and Query the Handshake Response State

Syntax
Command **:SYSTem:COMMunicate:RESPonse** <1/0/ON/OFF>
Query **:SYSTem:COMMunicate:RESPonse?**
Response <ON/OFF>
<ON> = The response message is replied when a command message is received.
<OFF> = The response message is not replied when a command message is received.

Example

```
:SYST:COMM:RESP ON
OK
:SYST:COMM:RESP OFF
:SYST:COMM:RESP?
OFF
```

Note
- When :SYSTem:COMMunicate:BT3562A ON is set, the response message is not replied even when a command message is received in handshake response ON.
- When :SYSTem:COMMunicate:RESPonse ON is set, sends a command, and then the handshake message must be read by the controller before sending another command.

Set and Query the BT3562A Compatible Mode

Syntax
Command **:SYSTem:COMMunicate:BT3562A** <1/0/ON/OFF>
Query **:SYSTem:COMMunicate:BT3562A?**
Response <ON/OFF>

Description When :SYSTem:COMMunicate:BT3562A ON is set, the query response and the measured value format are the same as BT3562A's

```
:TRIG:SOUR INT
:SYST:COMM:FORM FIX

:SYST:COMM:BT3562A OFF
:TRIG:SOUR?
INTERNAL
:FETC?
+0.00890E-03,-00.000001E+00

:SYST:COMM:BT3562A ON
:TRIG:SOUR?
IMMEDIATE
:FETC?
xx0.00890E-3,-x0.000001E+0
```

("x" is a space (ASCII 20H))

(28) Memory Functions

Read the Measured Values Stored in the Memory

Syntax

Query **:MEMory:DATA?** [STEP]

Response <Memory No. (NR1)>,<Resistance value (NR3)>,<Voltage value (NR3)>

Note: The measured values are output in the order that they are stored.

See: Measurement Value Format (p.15)

(1) When STEP as the parameter is not specified
Reads all measured values stored in the memory continuously.

(2) When STEP as the parameter is specified
Reads a measured value stored in the memory.
And reads another measured value stored in the memory by N command.

Example

```
:MEM:COUN?
3
:MEM:DATA?
001,+0.00890E-03,+00.000001E+00
002,+0.00889E-03,+00.000000E+00
003,+0.00900E-03,-00.000001E+00
END

:MEM:DATA? STEP
001,+0.00890E-03,+00.000001E+00
N
002,+0.00889E-03,+00.000000E+00
N
003,+0.00900E-03,-00.000001E+00
N
END
```

Note

- When :MEMory:STATe ON is set,
  measured values are stored in the memory when ∗TRIG command is received,
  the TRIG signal is input from the external I/O or the [TRIGGER] key is pressed.
- Up to 528 data can be stored in the memory. When the memory is full, measured values are not stored any more.
- The “END” character is sent after all measured values stored in the memory is read.

Set and Query the Memory Function

Syntax
Command **:MEMory:STATe** <1/0/ON/OFF>
Query **:MEMory:STATe?**
Response <ON/OFF>

Example
```
:MEM:STAT ON
:MEM:STAT?
ON
```

Query the Number of Measured Values Stored in the Memory

Syntax
Query **:MEMory:COUNt?**
Response <Memory data count> = 0 to 528 (NR1)

Description A count has a resistance value and a voltage value.

Example
```
:MEM:COUN?
5
```

Delete the Memory Data

Syntax
Command **:MEMory:CLEar**

Description Deletes the measured values stored in the memory.

(29) EXT. I/O

Query the NPN/PNP Hardware Switch State

Syntax Query **:IO:MODE?**
Response <NPN/ PNP >

Example
:IO:MODE?
NPN
The I/O mode switch on the back of the instrument is on the NPN side.

Set and Query the TRIG Signal Filter Function

Syntax Command **:IO:FILTer:STATe** <1/0/ON/OFF>
Query **:IO:FILTer:STATe?**
Response <ON/OFF>

Example
:IO:FILT:STAT ON
:IO:FILT:STAT?
ON

Set and Query the Period of TRIG Signal Filter Function

Syntax Command **:IO:FILTer:TIME** <Filter time>
Query **:IO:FILTer:TIME?**
Response <Filter time>
<Filter time> = 0.050 to 0.500 (NRf) [sec]

Example
:IO:FILT:TIME 0.1
:IO:FILT:TIME?
1.00000000E-01

Note When :SYSTem:COMMunicate:BT3562A ON is set,
the response of query is <Filter time> = 0.050 to 0.500 (NR2) [sec].

Set and Query the EOM Signal Output Mode

Syntax Command **:IO:EOM:MODE** <HOLD/PULSe>
Query **:IO:EOM:MODE?**
Response <HOLD/PULSE>
<HOLD> = Holds the EOM signal's output until the next trigger is received.
<PULSe> = Outputs the set pulse.

Example
:IO:EOM:MODE HOLD
:IO:EOM:MODE?
HOLD

Set and Query the Width of EOM Pulse Signal

Syntax Command **:IO:EOM:PULSe** <Pulse width>
Query **:IO:EOM:PULSe?**
Response <Pulse width>
<Pulse width> = 0.001 to 0.100 (NRf) [sec]

Example
:IO:EOM:PULS 0.005
:IO:EOM:PULS?
5.00000000E-03

Note When :SYSTem:COMMunicate:BT3562A ON is set,
the response of query is <Pulse width> = 0.001 to 0.100 (NR2) [sec].

Set and Query the ERR Signal Output Mode

Syntax Command **:IO:ERRor** <SYNChronous/ASYNchronous>
Query **:IO:ERRor?**
Response <SYNCHRONOUS/ASYNCHRONOUS>
<SYNChronous> = Measurement faults detected while measuring (not include trigger wait state, trigger delay state, or calculation state) are indicated by the ERR output synchronous with the EOM output.
<ASYNchronous> = Measurement faults (wire break detection) are indicated by the ERR output asynchronous with the EOM output.

Example
```
:IO:ERR SYNC
:IO:ERR?
SYNCHRONOUS
```

Note When :IO:ERRor SYNChronous is set,
only the sense circuit's wire break detection is operated but not the source circuit's in :TRIGger:SOURce:EXTernal setting.

Read the External I/O Input Register

Syntax Query **:IO:INPut?**
Response <0 to 65535 (NR1)>

Description Reads the ON edge of the TRIG, ADJ, CAL, CAL2 of the EXT. I/O, and then clears them.
When the ON edge of the EXT. I/O or front-key is detected, the related bits are set.
After the External I/O Input Register is read by this query, the External I/O Input Register is cleared as 0.

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	-	-	CAL2	-	-	CAL	ADJ	TRIG
Pin No.	-	-	2	-	-	21	20	1

Please see the instrument instruction manual regarding external control (EXT. I/O).

Note
- When :IO:FILTer:STATe ON is set,
  the ON edge is recognized after the response time has passed.
- When :SYSTem:COMMunicate:BT3562A ON is set,
  the response of bit5 is always 0.

(30) Error Information

Read and Clear the Error Information

Syntax Query **:SYSTem:ERRor?**
Response <Error No.>,<Error name>

Example
```
:SYST:ERR?
220,"Parameter error"
```

Note When :SYSTem:COMMunicate:BT3562A ON is set,
the response of query is the same as :IO:ERRor?.

(31) Device ID Informations

Query the Device ID (Identify Code)

Syntax Query ***IDN?**
Response <Manufacture name>,<Model name>,<Serial number>,<Software version>

Note For details, check the *IDN? common commands.

Query the Model Name

Syntax Query **:QPID**
Response <Model name>

Example
```
:QPID
BT6065
```

Query the Serial Number

Syntax Query **:QSERIAL**
Response <Serial number>

Example
```
:QSERIAL
1234567890
```

Query the FPGA Version Number

Syntax Query **:FPGA:VERsion?** <DIGITAL/SENSE>
Response <Version number>

Example
```
:FPGA:VER? DIGITAL
A1234567
:FPGA:VER? SENSE
B1234567
```

BT3562A Compatible Commands

Read the Event Register of the Standard Operation Register Group

Syntax Query **:ESR0?**
Response <0 to 65535 (NR1)>
Note For details, check the :STATus:OPERation[:EVENt]? device-specific command.

Write and Read the Enable Register of the Standard Operation Register Group

Syntax Command **:ESE0** <0 to 65535 (NR1)>
Query **:ESE0?**
Response <0 to 65535 (NR1)>
Note For details, check the :STATus:OPERation:ENABle device-specific command.

Read the Event Register of the Status Query Register Group

Syntax Query **:ESR1?**
Response <0 to 65535 (NR1)>
Note For details, check the :STATus:QUEStionable[:EVENt]? device-specific command.

Write and Read the Enable Register of the Status Query Register Group

Syntax Command **:ESE1** <0 to 65535 (NR1)>
Query **:ESE1?**
Response <0 to 65535 (NR1)>
Note For details, check the :STATus:QUEStionable:ENABle device-specific command.

Set and Query the Measurement AUTO Range

Syntax Command **:AUTorange** <1/0/ON/OFF>
Query **:AUTorange?**
Response <ON/OFF>
Note For details, check the :RESistance:RANGe:AUTO device-specific command.

Set and Query the Number of Averaging Calculation Data

Syntax Command **:CALCulate:AVERage** <Average count>
Query **:CALCulate:AVERage?**
Response <Average count>
Note For details, check the :CALCulate:AVERage:COUNt device-specific command.

Set and Query the Zero-Suppress Function

Syntax Command **:CALCulate:SUPPress** <1/0/ON/OFF>
Query **:CALCulate:SUPPress?**
Response <ON/OFF>
Note For details, check the :CALCulate:ZERODISPlay:WIDTH device-specific command.

Set and Query the Voltage's Absolute Value Function

Syntax Command **:CALCulate:ABSolute** <1/0/ON/OFF>
Query **:CALCulate:ABSolute?**
Response <ON/OFF>
Note For details, check the :VOLTage:ABSolute device-specific command.

Set and Query the Comparator Function

Syntax Command **:CALCulate:LIMit:STATe** <1/0/ON/OFF>
Query **:CALCulate:LIMit:STATe?**
Response <ON/OFF>

Note For details, check the :COMParator:LIMit:STATe device-specific command.

Set and Query the Judgement Buzzer

Syntax Command **:CALCulate:LIMit:BEEPer** <OFF/HL/IN/BOTH1/BOTH2>
Query **:CALCulate:LIMit:BEEPer?**
Response <OFF/HL/IN/BOTH1/BOTH2>

Note For details, check the :COMParator:LIMit:BEEPer device-specific command.

Set and Query the Voltage's Absolute Value Judgement Function

Syntax Command **:CALCulate:LIMit:ABS** <1/0/ON/OFF>
Query **:CALCulate:LIMit:ABS?**
Response <ON/OFF>

Note For details, check the :COMParator:LIMit:ABSolute device-specific command.

Set and Query the Resistance's Comparator Mode

Syntax Command **:CALCulate:LIMit:RESistance:MODE** <HL/REF>
Query **:CALCulate:LIMit:RESistance:MODE?**
Response <HL/REF>
<HL> = Upper threshold/Lower threshold comparison
<REF> = Tolerance comparison with base value/percentage value.

Example
```
:CALC:LIM:RES:MODE HL
:CALC:LIM:RES:MODE?
HL
```

Note When :CALCulate:LIMit:RESistance:MODE REF is set,
the upper threshold and the lower threshold also are updated after base value or percentage value is changed.

Set and Query the Voltage's Comparator Mode

Syntax Command **:CALCulate:LIMit:VOLTage:MODE** <HL/REF>
Query **:CALCulate:LIMit:VOLTage:MODE?**
Response <HL/REF>
<HL> = Upper threshold/Lower threshold comparison
<REF> = Tolerance comparison with base value/percentage value.

Example
```
:CALC:LIM:VOLT:MODE HL
:CALC:LIM:VOLT:MODE?
HL
```

Note When :CALCulate:LIMit:VOLTage:MODE REF is set,
the upper threshold and the lower threshold also are updated after base value or percentage value is changed.

Set and Query the Resistance's Upper Threshold

Syntax Command **:CALCulate:LIMit:RESistance:UPPer** <Upper threshold>
Query **:CALCulate:LIMit:RESistance:UPPer?**
Response <Upper threshold>
When :RESistance:DIGits 5 is set,
<Upper threshold> = -1000 to 51000 (NR1)

When :RESistance:DIGits 6 is set,
<Upper threshold> = -10000 to 510000 (NR1)

Example
```
:CALC:LIM:RES:UPP 28593
:CALC:LIM:RES:UPP?
28593
```

When :RESistance:DIGits 5 is set,
the upper threshold is 285.93 mΩ in 300 mΩ range.
(If 3 Ω range is selected, the upper threshold is 2.8593 Ω.)

When :RESistance:DIGits 6 is set,
the upper threshold is 28.593 mΩ in 300 mΩ range.
(If 3 Ω range is selected, the upper threshold is 0.28593 Ω.)

Note
- The upper threshold parameter is a whole integer (count).
- When :SYSTem:COMMunicate:BT3562A ON is set, the upper threshold is calculated in 5 digits mode.
- <Upper threshold> smaller than :CALCulate:LIMit:RESistance:LOWer <Lower threshold> cannot be set.

Set and Query the Voltage's Upper Threshold

Syntax Command **:CALCulate:LIMit:VOLTage:UPPer** <Upper threshold>
Query **:CALCulate:LIMit:VOLTage:UPPer?**
Response <Upper threshold>
BT6065 <Upper threshold> = -1200000 to 1200000 (NR1)
BT6075 <Upper threshold> = -12000000 to 12000000 (NR1)

Example
```
:CALC:LIM:VOLT:UPP 380000
:CALC:LIM:VOLT:UPP?
380000
```

BT6065 The upper threshold is 3.8 V in 10 V range.
(If 100 V range is selected, the upper threshold is 38 V.)

BT6075 The upper threshold is 0.38 V in 10 V range.
(If 100 V range is selected, the upper threshold is 3.8 V.)

Note
- The upper threshold parameter is a whole integer (count).
- When :SYSTem:COMMunicate:BT3562A ON is set, the upper threshold is calculated as BT6065 model.
- <Upper threshold> smaller than :CALCulate:LIMit:VOLTage:LOWer <Lower threshold> cannot be set.

Set and Query the Resistance's Lower Threshold

Syntax Command **:CALCulate:LIMit:RESistance:LOWer** <Lower threshold>
Query **:CALCulate:LIMit:RESistance:LOWer?**
Response <Lower threshold>

When :RESistance:DIGits 5 is set,
<Lower threshold> = -1000 to 51000 (NR1)

When :RESistance:DIGits 6 is set,
<Lower threshold> = -10000 to 510000 (NR1)

Example
```
:CALC:LIM:RES:LOW 28406
:CALC:LIM:RES:LOW?
28406
```

When :RESistance:DIGits 5 is set,
the lower threshold is 284.06 mΩ in 300 mΩ range.
(If 3 Ω range is selected, the lower threshold is 2.8406 Ω.)

When :RESistance:DIGits 6 is set,
the lower threshold is 28.406 mΩ in 300 mΩ range.
(If 3 Ω range is selected, the lower threshold is 0.28406 Ω.)

Note
- The lower threshold parameter is a whole integer (count).
- When :SYSTem:COMMunicate:BT3562A ON is set, the lower threshold is calculated in 5 digits mode.
- <Lower threshold> larger than :COMParator:LIMit:RESistance:UPPer <Upper threshold> cannot be set.

Set and Query the Voltage's Lower Threshold

Syntax Command **:CALCulate:LIMit:VOLTage:LOWer** <Lower threshold>
Query **:CALCulate:LIMit:VOLTage:LOWer?**
Response <Lower threshold>

BT6065 <Lower threshold> = -1200000 to 1200000 (NR1)
BT6075 <Lower threshold> = -12000000 to 12000000 (NR1)

Example
```
:CALC:LIM:VOLT:LOW 360000
:CALC:LIM:VOLT:LOW?
360000
```

BT6065 The lower threshold is 3.6 V in 10 V range.
(If 100 V range is selected, the lower threshold is 36 V.)

BT6075 The lower threshold is 0.36 V in 10 V range.
(If 100 V range is selected, the lower threshold is 3.6 V.)

Note
- The lower threshold parameter is a whole integer (count).
- When :SYSTem:COMMunicate:BT3562A ON is set, the lower threshold is calculated as BT6065 model.
- <Lower threshold> larger than :COMParator:LIMit:VOLTage:UPPer <Upper threshold> cannot be set.

Set and Query the Resistance's Base Value

Syntax Command **:CALCulate:LIMit:RESistance:REFerence** <Base value>
Query **:CALCulate:LIMit:RESistance:REFerence?**
Response <Base value>
When :RESistance:DIGits 5 is set,
<Base value> = -1000 to 51000 (NR1)

When :RESistance:DIGits 6 is set,
<Base value> = -10000 to 510000 (NR1)

Example
```
:CALC:LIM:RES:REF 5076
:CALC:LIM:RES:REF?
5076
```

When :RESistance:DIGits 5 is set,
the base value is 50.76 mΩ in 300 mΩ range.
(If 3 Ω range is selected, the lower threshold is 0.5076 Ω.)

When :RESistance:DIGits 6 is set,
the lower threshold is 5.076 mΩ in 300 mΩ range.
(If 3 Ω range is selected, the lower threshold is 0.05076 Ω.)

Note
- The base value parameter is a whole integer (count).
- When :SYSTem:COMMunicate:BT3562A ON is set, the base value is calculated in 5 digits mode.
- When :CALCulate:LIMit:RESistance:MODE REF is set, the upper threshold and the lower threshold also are updated after base value or percentage value is changed.

Set and Query the Voltage's Base Value

Syntax Command **:CALCulate:LIMit:VOLTage:REFerence** <Base value>
Query **:CALCulate:LIMit:VOLTage:REFerence?**
Response <Base value>
BT6065 <Base value> = -1200000 to 1200000 (NR1)
BT6075 <Base value> = -12000000 to 12000000 (NR1)

Example
```
:CALC:LIM:VOLT:REF 370000
:CALC:LIM:VOLT:REF?
370000
```

BT6065 The base value is 3.7 V in 10 V range.
(If 100 V range is selected, the base value is 37 V.)

BT6075 The base value is 0.37 V in 10 V range.
(If 100 V range is selected, the base value is 3.7 V.)

Note
- The base value parameter is a whole integer (count).
- When :SYSTem:COMMunicate:BT3562A ON is set, the base value is calculated as BT6065 model.
- When :CALCulate:LIMit:VOLTage:MODE REF is set, the upper threshold and the lower threshold also are updated after base value or percentage value is changed.

Set and Query the Resistance's Percentage Value

Syntax Command **:CALCulate:LIMit:RESistance:PERCent** <Percentage value>
Query **:CALCulate:LIMit:RESistance:PERCent?**
Response <Percentage value>
<Percentage value> = 0.000 to 99.999 (NR2) [%]

Example
```
:CALC:LIM:RES:PERC 0.3
:CALC:LIM:RES:PERC?
0.300
```

Note When :CALCulate:LIMit:RESistance:MODE REF is set, the upper threshold and the lower threshold also are updated after base value or percentage value is changed.

Set and Query the Voltage's Percentage Value

Syntax Command **:CALCulate:LIMit:VOLTage:PERCent** <Percentage value>
Query **:CALCulate:LIMit:VOLTage:PERCent?**
Response <Percentage value>
<Percentage value> = 0.000 to 99.999 (NR2) [%]

Example
```
:CALC:LIM:VOLT:PERC 1.538
:CALC:LIM:VOLT:PERC?
1.538
```

Note When :CALCulate:LIMit:VOLTage:MODE REF is set, the upper threshold and the lower threshold also are updated after base value or percentage value is changed.

Query the Comparator Result for Resistance Measurement

Syntax Query **:CALCulate:LIMit:RESistance:RESult?**
Response <HI/IN/LO/ERR/OFF>

Note For details, check the :COMParator:LIMit:RESistance:RESult? device-specific command.

Query the Comparator Result for Voltage Measurement

Syntax Query **:CALCulate:LIMit:VOLTage:RESult?**
Response <HI/IN/LO/ERR/OFF>

Note For details, check the :COMParator:LIMit:VOLTage:RESult? device-specific command.

Execute the Zero-Adjustment for Single Mode and Query the Result

Syntax Query **:ADJust?**
Response <PASS/FAIL>

Description Executes :ADJust:ZERO:SINGle? (without ALL).

Note For details, check the :ADJust:ZERO:SINGle? device-specific command.

Set the Adjustment Type OFF

Syntax Command **:ADJust:CLEAr**

Description Executes :ADJust:TYPE OFF.

Note For details, check the :ADJust:TYPE device-specific command.

Delete the Memory Data

Syntax Command **:MEMory:CLEAr**

Note For details, check the :MEMory:CLEar device-specific command.

Execute the Voltage's Self-Calibration

Syntax Command **:SYSTem:CALibration**

Note
- When the calibration result is FAIL, an execution error occurs.
- When :SYSTem:CALibration is received during measurement, the measurement is aborted immediately and then the self-calibration is executed.

Save and Load the Measurement Conditions

Syntax Command **:SYSTem:SAVE** <Panel No.>
:SYSTem:LOAD <Panel No.>

Note For details, check the ∗SAV and ∗RCL standard commands.

Set and Query the Key-Beeper State

Syntax Command **:SYSTem:BEEPer:STATe** <1/0/ON/OFF>
Query **:SYSTem:BEEPer:STATe?**
Response <ON/OFF>

Note For details, check the :SYSTem:BEEPer:KEY device-specific command.

Set and Query the Measurement-Synchronized Data Output Function

Syntax Command **:SYSTem:DATAout** <1/0/ON/OFF>
Query **:SYSTem:DATAout?**
Response <ON/OFF>

Note For details, check the :SYSTem:COMMunicate:DATAout device-specific command.

Set and Query the Header Presence

Syntax Command **:SYSTem:HEADer** <1/0/ON/OFF>
Query **:SYSTem:HEADer?**
Response <ON/OFF>

Note For details, check the :SYSTem:COMMunicate:HEADer device-specific command.

Set and Query the External I/O Input Lock

Syntax Command **:SYSTem:ELOCk** <1/0/ON/OFF>
Query **:SYSTem:ELOCk?**
Response <ON/OFF>

Note This function is only used with communication commands.

Set and Query the EOM Signal Output Mode

Syntax Command **:SYSTem:EOM:MODE** <HOLD/PULSe>
Query **:SYSTem:EOM:MODE?**
Response <HOLD/PULSE>

Note For details, check the :IO:EOM:MODE device-specific command.

Set and Query the Width of EOM Pulse Signal

Syntax Command **:SYSTem:EOM:PULSe** <Pulse width (sec)>
Query **:SYSTem:EOM:PULSe?**
Response <Pulse width (sec)>
<Pulse width> = 0.001 to 0.100 (NRf) [sec]

Note For details, check the :IO:EOM:PULSe device-specific command.

Set the ERR Signal Output Mode

Syntax Command **:SYSTem:ERRor** <SYNChronous/ASYNchronous>

Note For details, check the :IO:ERRor device-specific command.

Read the External I/O Input Register

Syntax Query **:IO:IN?**
Response <0 to 65535 (NR1)>

Note For details, check the :IO:INPut? device-specific command.

4 Data Exporting Methods

Basic Data Exporting Methods

Flexible data exporting is available depending on the application.

Export Free-Run Data

Default Setting	**:INITiate:CONTinuous ON** (continuous measurement enable) **:TRIGger:SOURce INTernal** (internal triggering)
Exporting	**:FETCh?** Imports the most recent measurement.

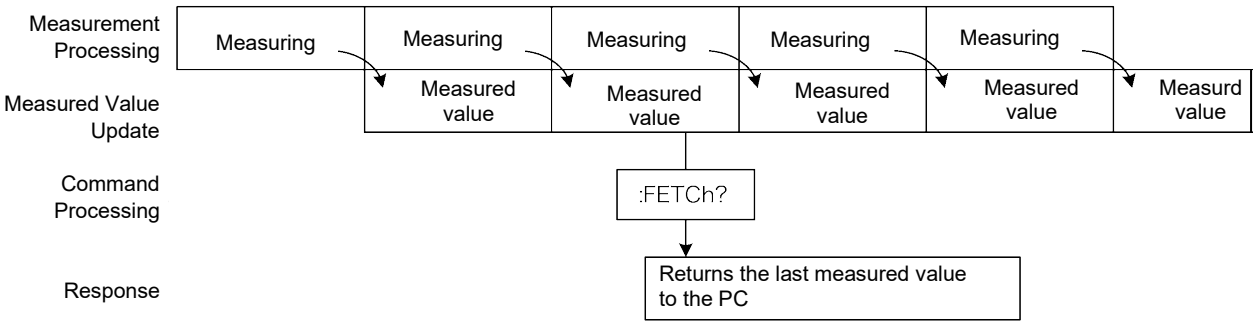
Export by Controller (PC, PLC) Triggering

Default Setting	**:INITiate:CONTinuous OFF** (continuous measurement disable) **:TRIGger:SOURce INTernal** (internal triggering)
Exporting	**:READ?** A trigger occurs, a measurement is performed and the result is transferred.

Export by Applying TRIG Signal, Pressing [TRIGGER] Key or Sending ∗TRG command

Default Setting	**:INITiate:CONTinuous OFF** (continuous measurement disable) **:TRIGger:SOURce EXTernal** (external triggering)
Exporting	**:READ?** When triggered by TRIG signal, the [TRIGGER] Key or ∗TRG command, a measurement is performed and the result is transferred.

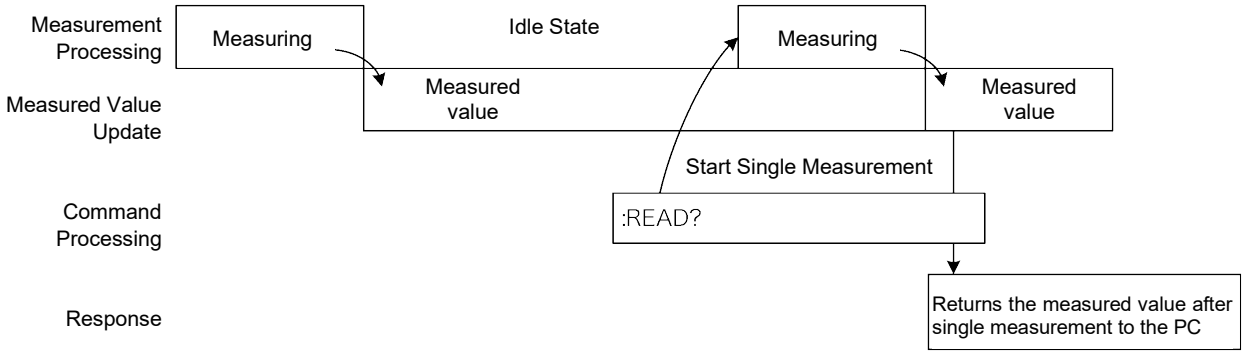
Using the **:FETCh?** Command during Continuous Measurement with Internal Triggering



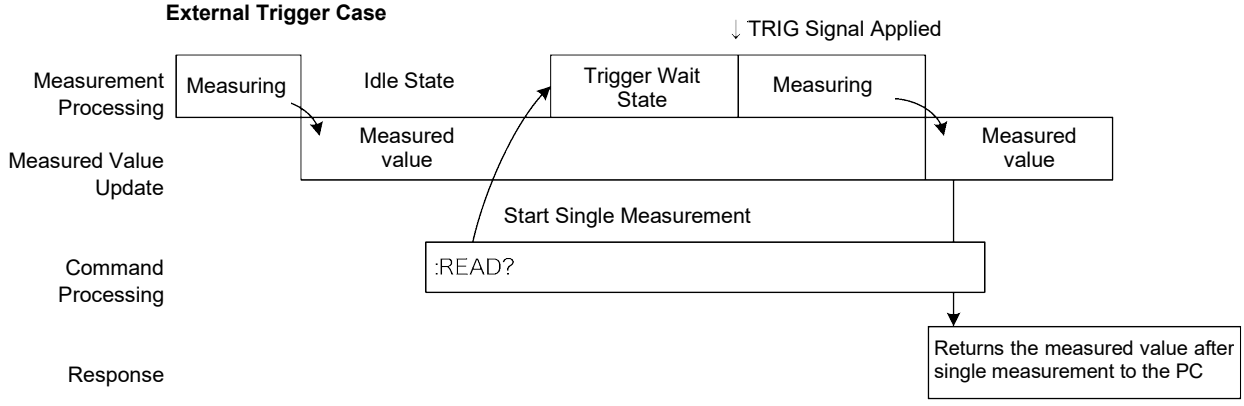
This is the basic method for exporting measured values.
It is ideal when measurement (tact) time is not limited, and when external synchronization is not needed.
After connecting to the DUT (Device Under Test), wait for twice the measurement time before exporting the measured value.

Using the **:READ?** Command while Continuous Measurement is Disabled

Internal Trigger Case



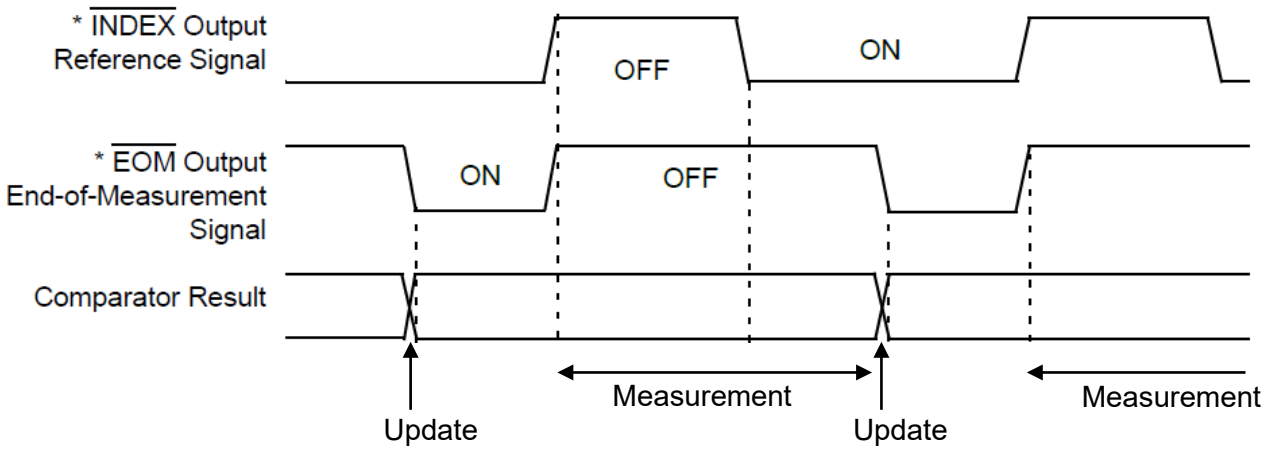
External Trigger Case



Use this method to measure (and export) synchronously with the controller (PC, PLC) or external trigger signal.
Measurement time can be minimized.

5 Update Timing of Comparator Result

Timing Chart



Holds the previous comparator result during measurement.
The Comparator result is updated when the measurement is completed.

6 Sample Programs

These programs can be created using Visual Basic 2022 and Visual C#.
Visual Basic or Visual C# is a registered trademark of Microsoft Cprporation.

Using Visual Basic

Describes an example of how to use the Visual Basic 2022 to operate from a Computer via RS-232C/USB, incorporate measurement values, and save measurement values to a file.

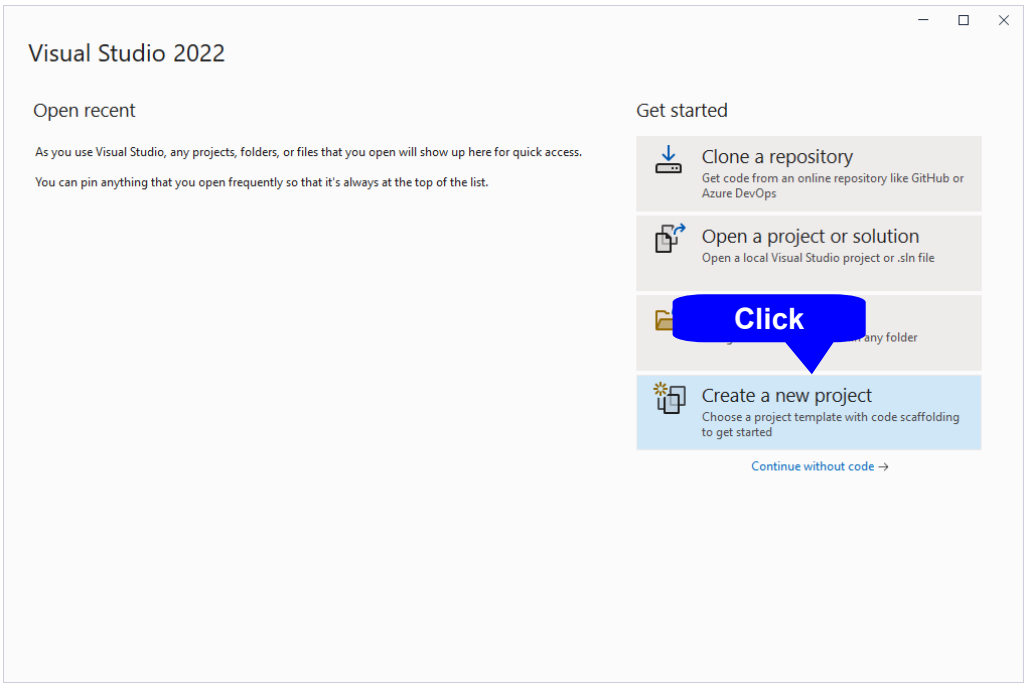
Visual Basic 2022 is referred to as VB2022 hereafter.

Note: Depending on the environment of the PC and VB2022, the procedure may differ slightly from the one described here. For a detailed explanation on how to use VB2022, refer to the instruction manual or Help of VB2022.

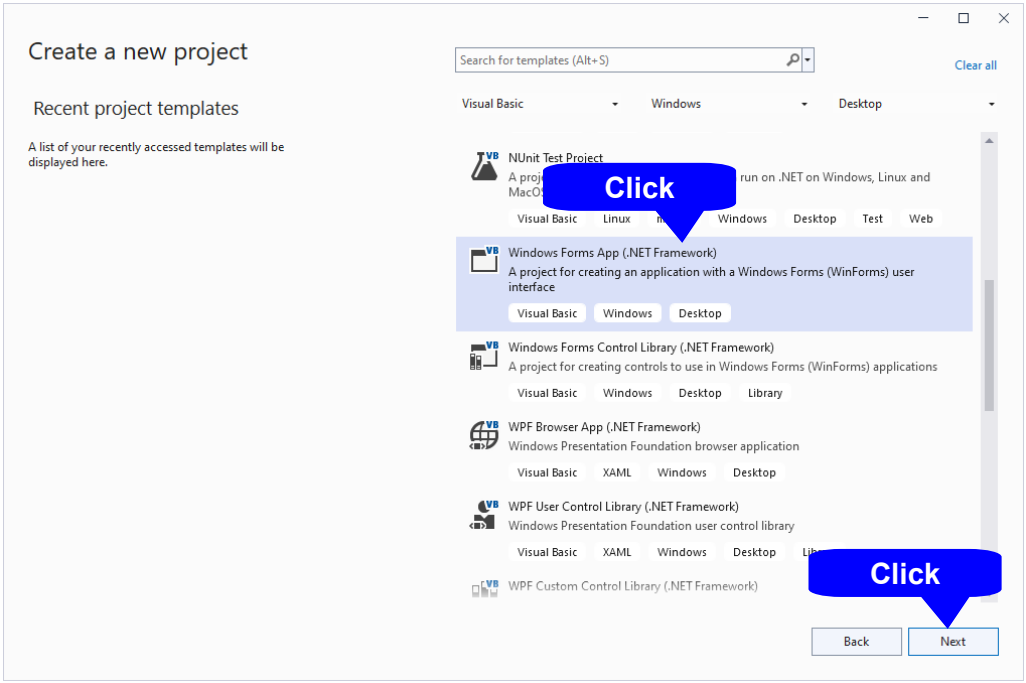
1. Create a new project.

1. Startup Visual Studio.

2. Select [Create a new project].

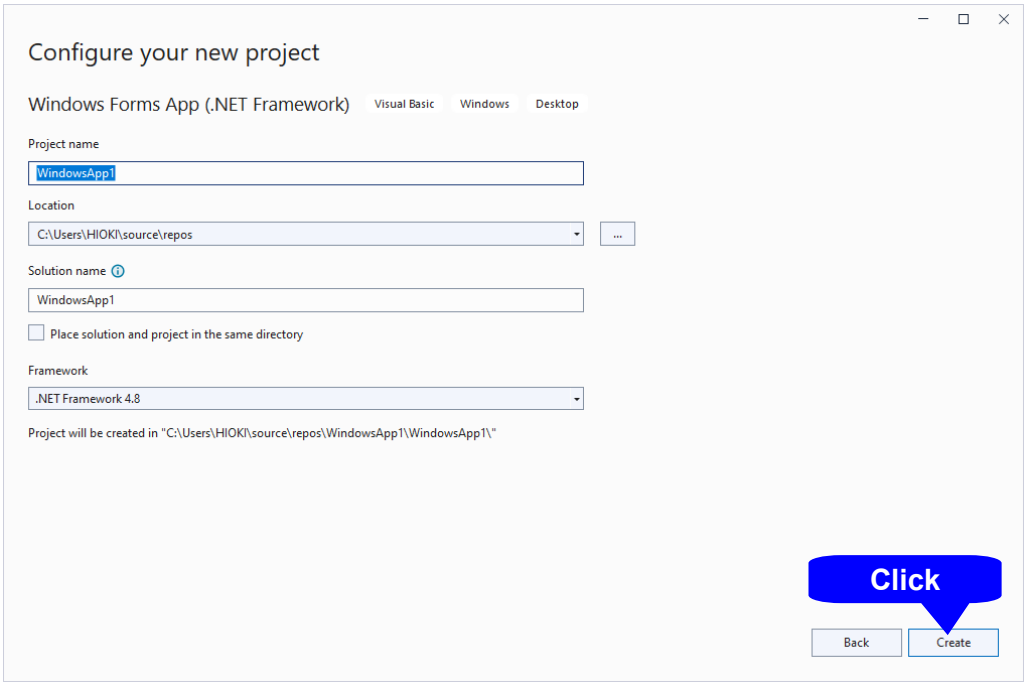


3. Select [Visual Basic]-[Windows]-[Desktop]-[Windows Forms App **(.NET Framework)**] from the templates.



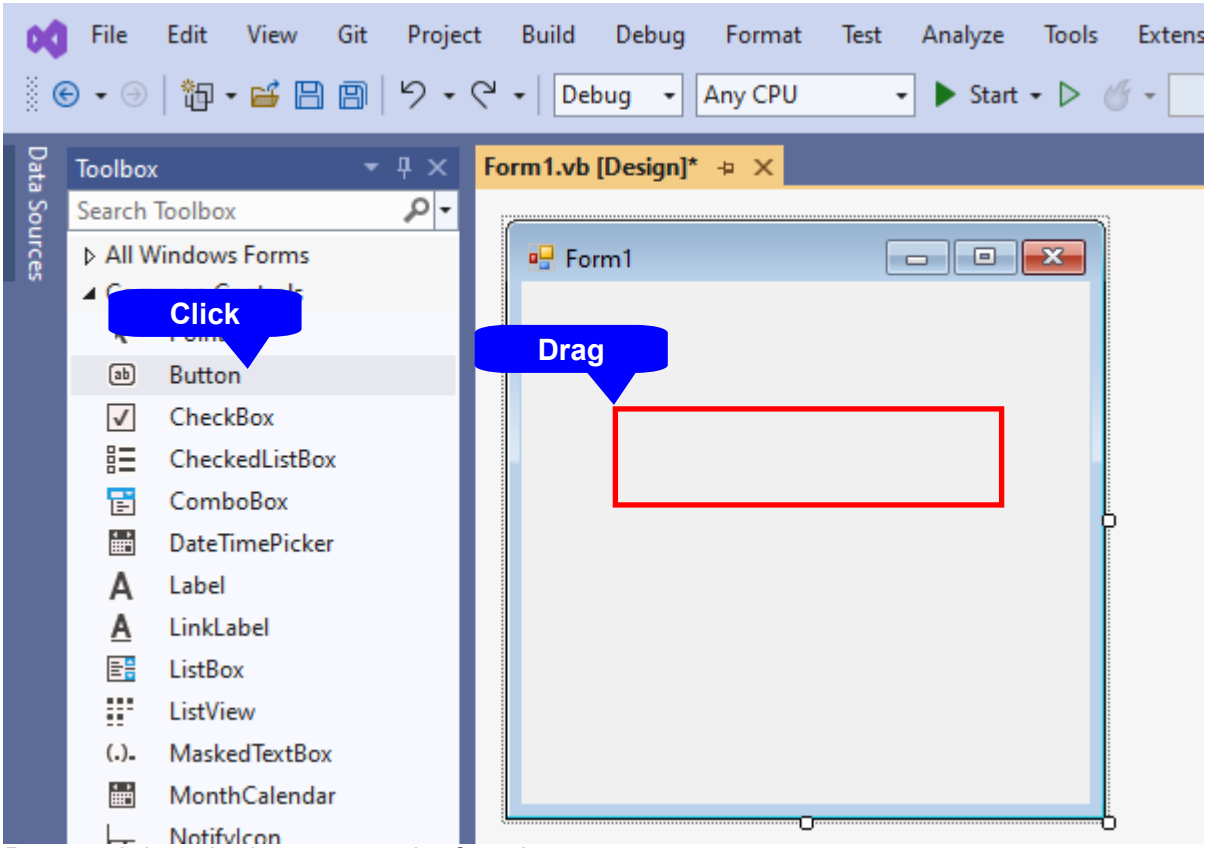
4. Click [Next].

5. Click [Create].



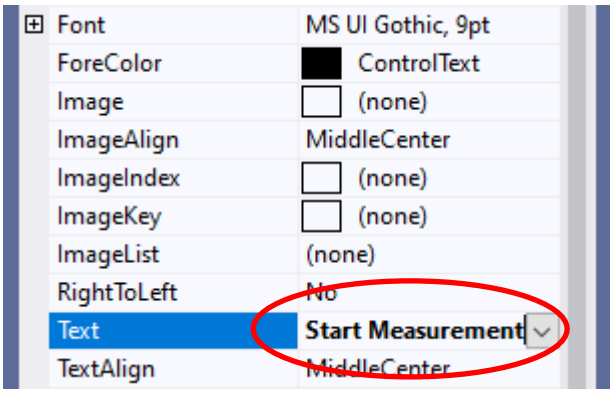
2. Place a button.

1. Click [Button] from [Common Controls] of [Toolbox].

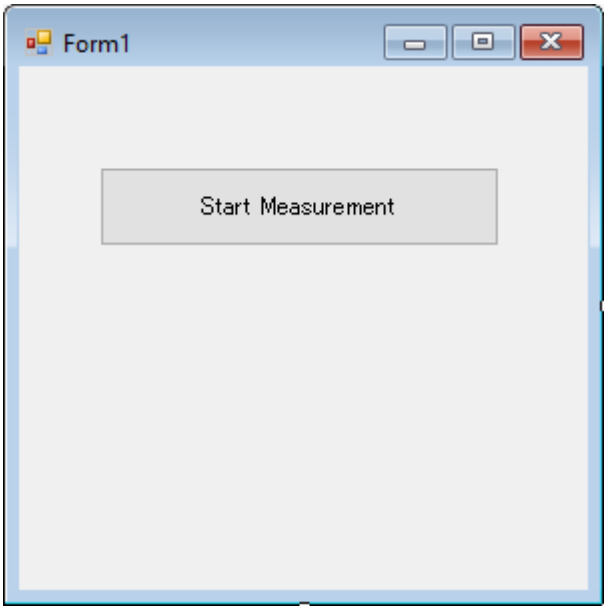


2. Drag and drop the button onto the form layout screen.

3. Change [Text] to [Start Measurement] from the Properties window.

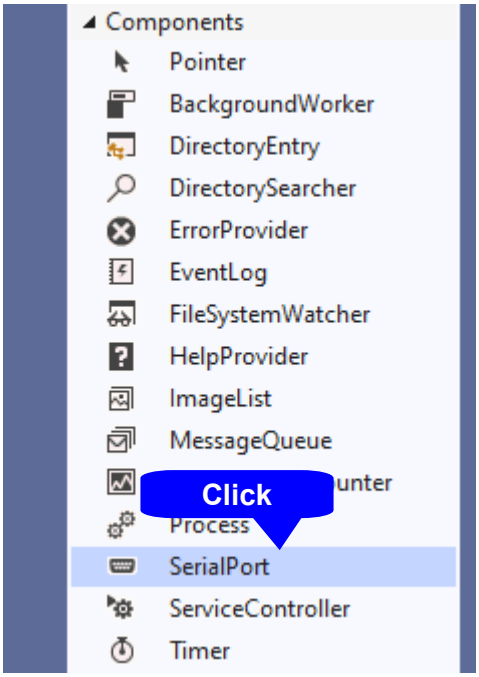


4. The [Start Measurement] is placed on the form.

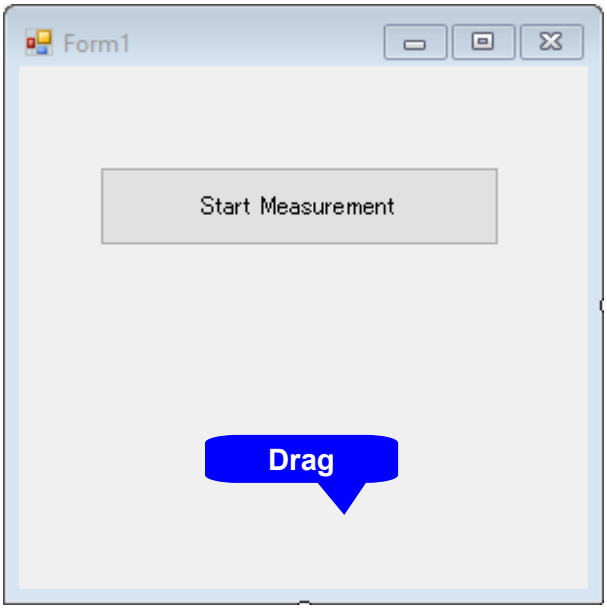


3. Place a serial communication component.

1. Click [SerialPort] from [Components] of [Toolbox].

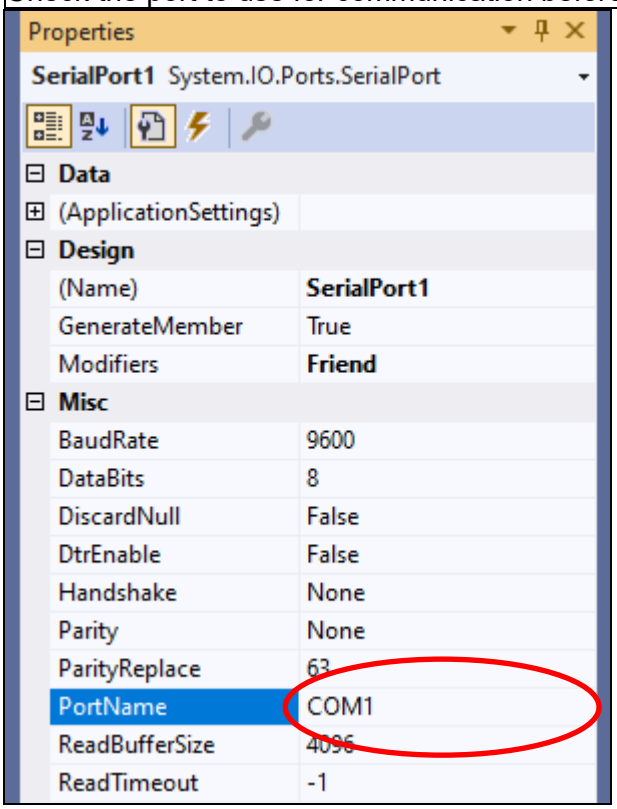


2. Drag and drop the [SerialPort] component onto the form layout screen.



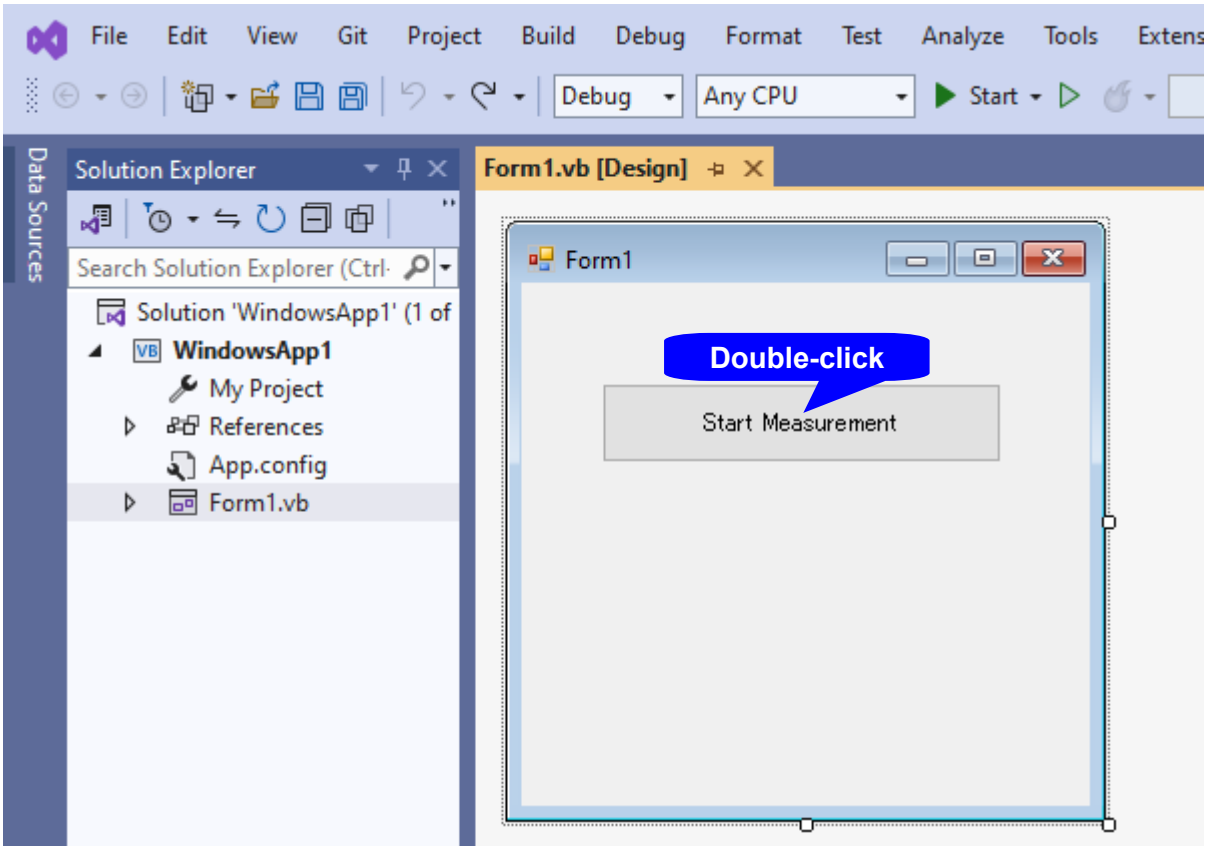
3. Change [PortName] to the port name to use for communication from the properties window.

Check the port to use for communication beforehand.

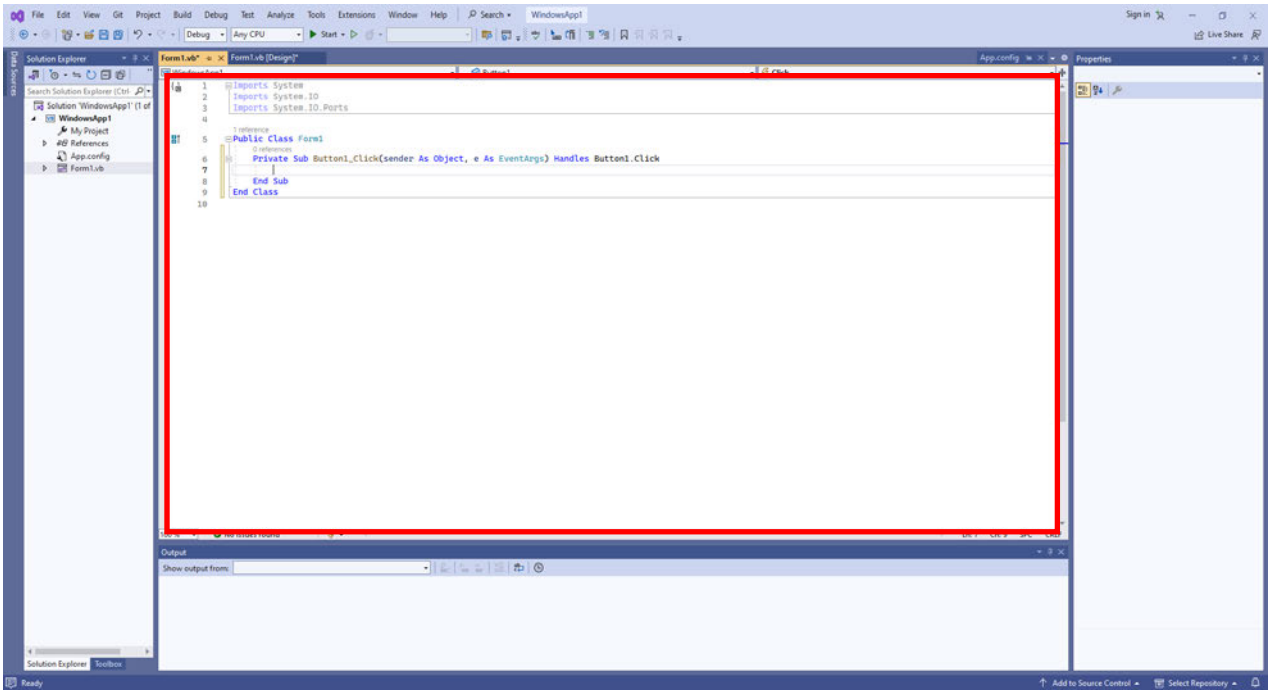


4. Describe the code.

1. Double-click the placed button to display the code editor.



2. Enter the sample program into the code editor.



3. Select [Save All] from the [File] menu.

Shown below is a sample program which uses VB2022 to enact RS-232C/USB communication, set the instrument measurement conditions, read measurement results and then save them to file.
The sample program will be written in the following manner.

Description of creation procedure	Description in sample program
Button created to begin measurement	Button1

When the [Begin Measurement] button is pressed, the instrument performs 10 measurements and writes the measurement values to a “data.csv” file.
When the [X] button is pressed, the program closes.
The following program is written entirely in [Form1] code.

```
Imports System.IO.Ports

Public Class Form1
    'Perform process when Button1 is pressed
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim recvstr As String
        Dim i As Integer
        Try
            Button1.Enabled = False 'Disable buttons during communication
            SerialPort1.NewLine = vbCrLf    'Terminator setting
            SerialPort1.ReadTimeout = 2000  '2 seconds time out
            SerialPort1.Open()  'Open a port
            SendSetting(SerialPort1)    'Instrument settings
            FileOpen(1, "data.csv", OpenMode.Output)    'Create text file to be saved
            For i = 1 To 10
                SerialPort1.WriteLine(":FETCH?")
                                    'Begin measurement and read measurement results command
                recvstr = SerialPort1.ReadLine()    'Read measurement results
                WriteLine(1, recvstr)   'Write to file
            Next i
            FileClose(1)    'Close file
            SerialPort1.Close()  'Close port
            Button1.Enabled = True
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    'Set measurement conditions
    Private Sub SendSetting(ByVal sp As SerialPort)
        Try
            sp.WriteLine(":FUNC RV")        'Select RV function
            sp.WriteLine(":TRIG:SOUR INT")  'Select internal triggering
            sp.WriteLine(":INIT:CONT ON")   'Continuous measurement ON
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub
End Class
```

Using Visual C#

Describes an example of how to use the Visual C# 2022 to operate from a Computer via LAN, incorporate measurement values, and save measurement values to a file.

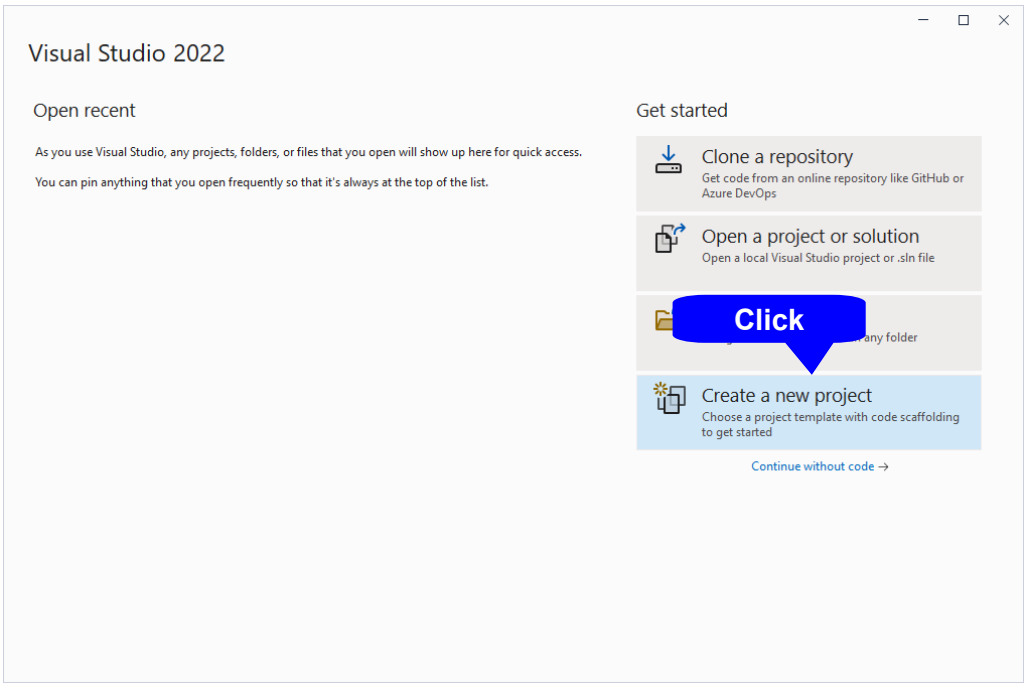
Visual Basic C# 2022 is referred to as CS2022 hereafter.

Note: Depending on the environment of the PC and CS2022, the procedure may differ slightly from the one described here. For a detailed explanation on how to use CS2022, refer to the instruction manual or Help of CS2022.

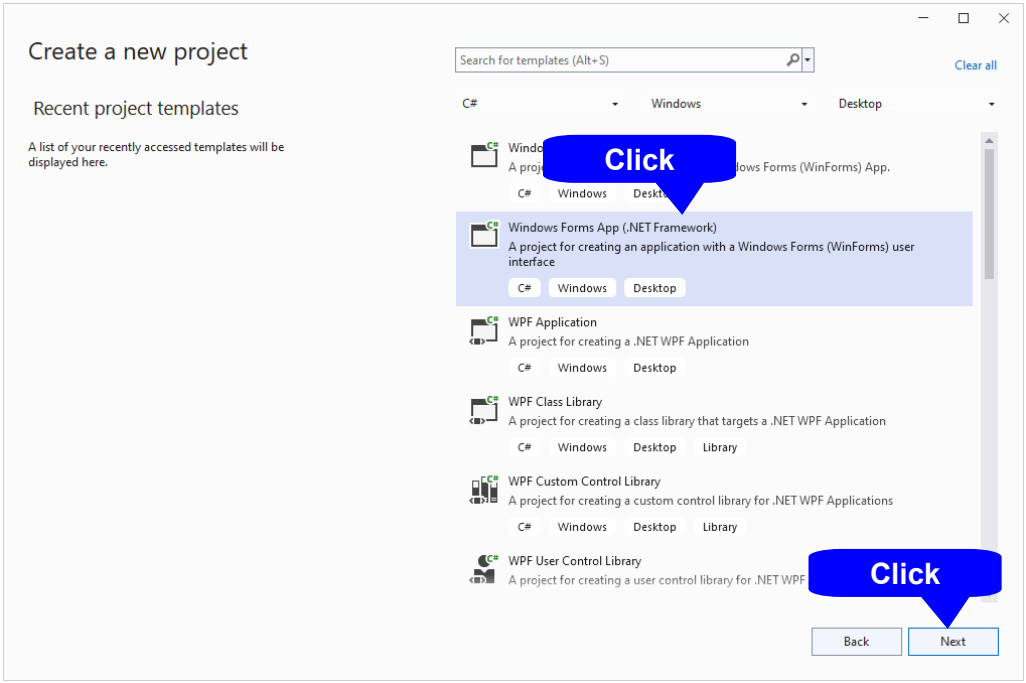
1. Create a new project.

1. Startup Visual Studio.

2. Select [Create a new project].

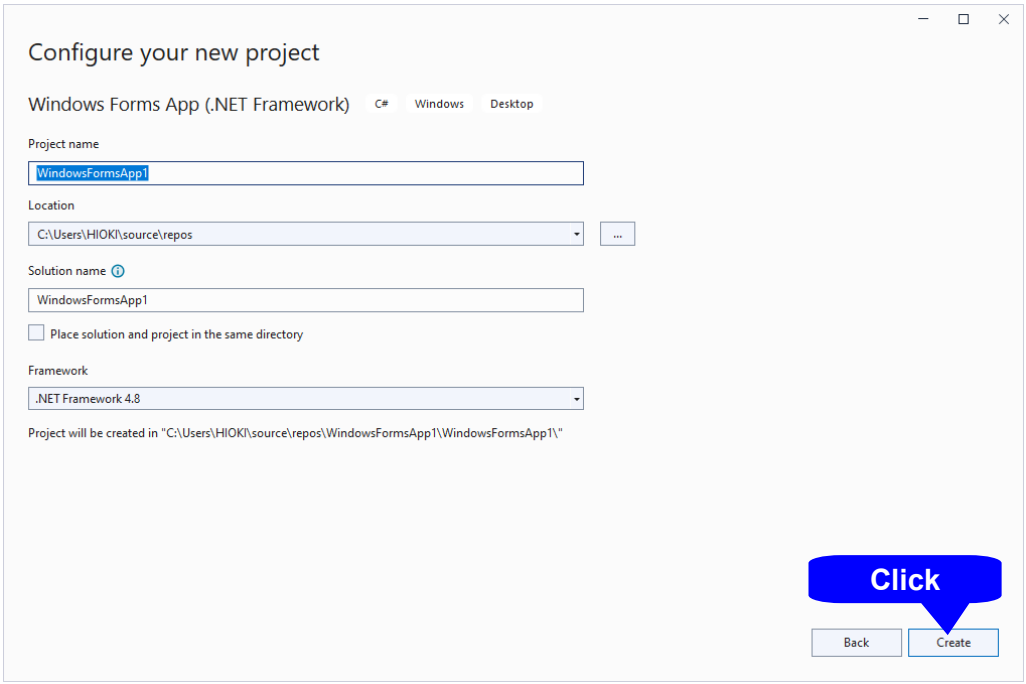


3. Select [C#]-[Windows]-[Desktop]-[Windows Forms App **(.NET Framework)**] from the templates.



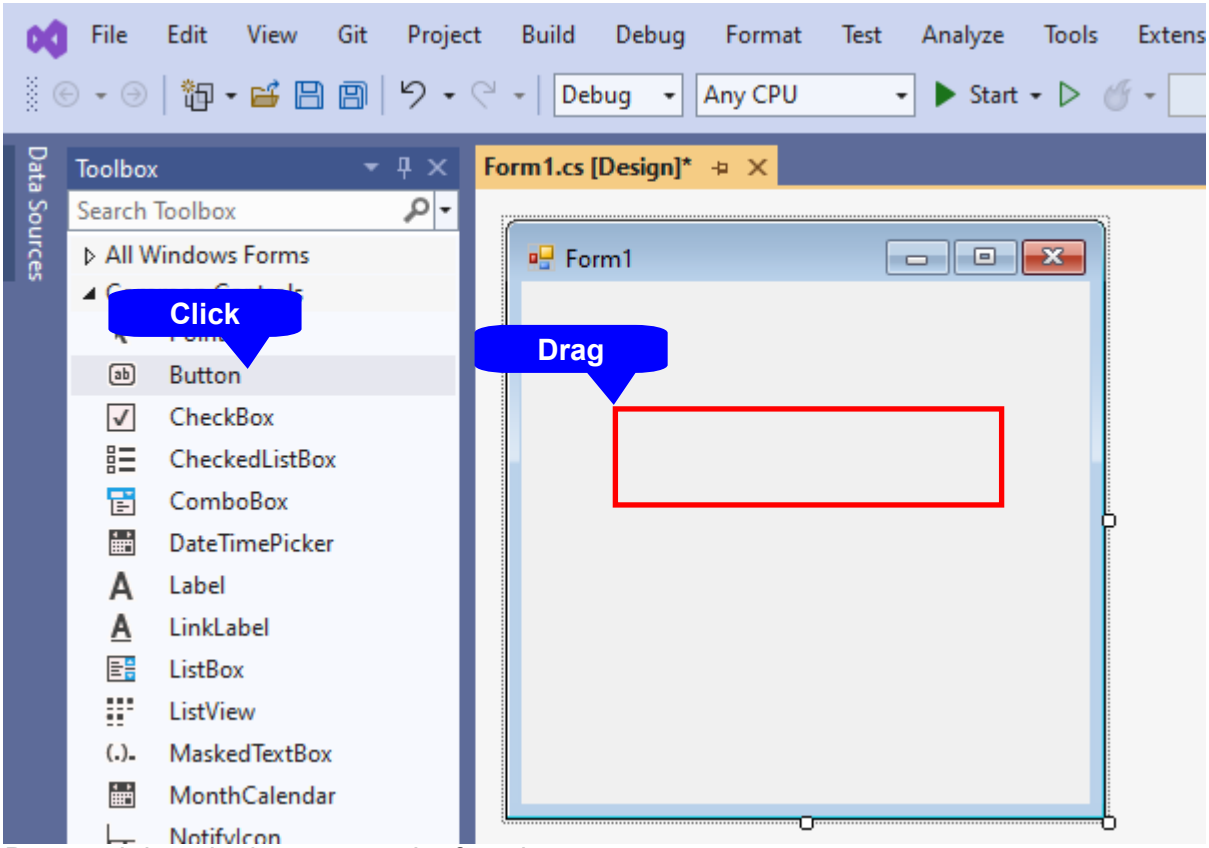
4. Click [Next].

5. Click [Create].



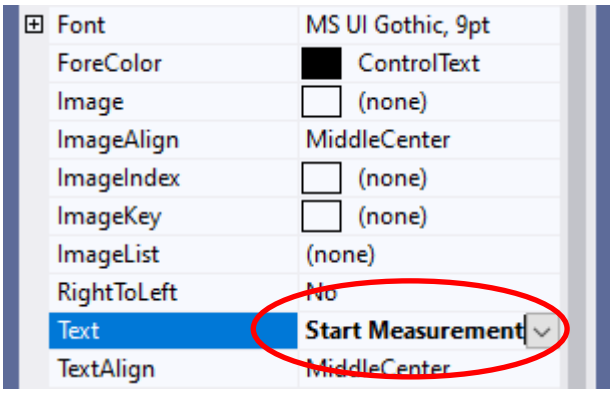
2. Place a button.

1. Click [Button] from [Common Controls] of [Toolbox].

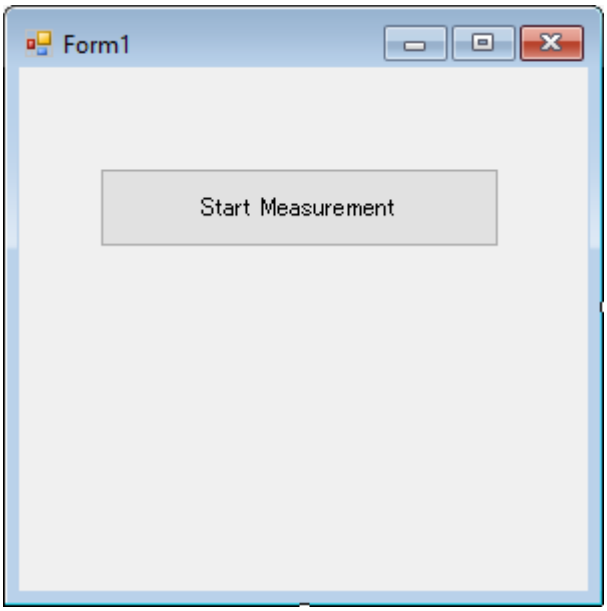


2. Drag and drop the button onto the form layout screen.

3. Change [Text] to [Start Measurement] from the Properties window.

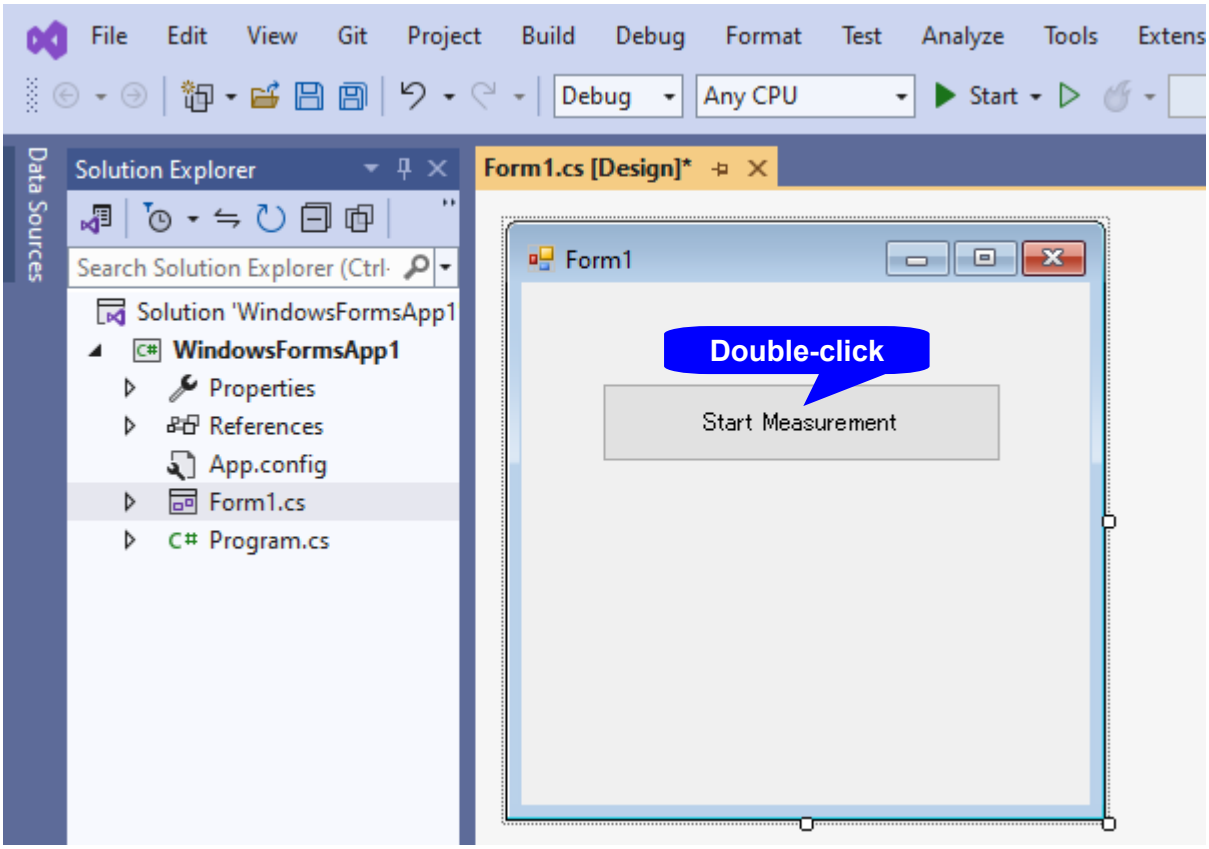


4. The [Start Measurement] is placed on the form.

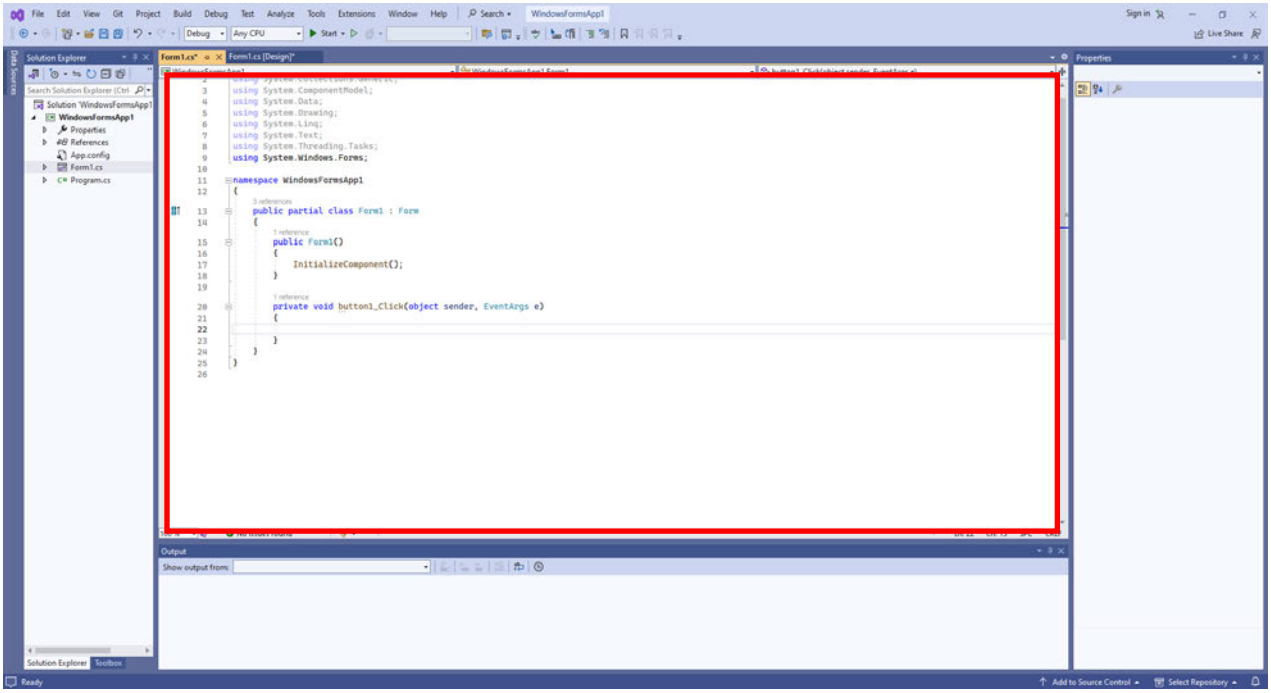


4. Describe the code.

1. Double-click the placed button to display the code editor.



2. Enter the sample program into the code editor.



3. Select [Save All] from the [File] menu.

Shown below is a sample program which uses CS2022 to enact LAN communication, set the instrument measurement conditions, read measurement results and then save them to file.
The sample program will be written in the following manner.

Description of creation procedure	Description in sample program
Button created to begin measurement	button1

When the [Begin Measurement] button is pressed, the instrument performs 10 measurements and writes the measurement values to a “data.csv” file.
When the [X] button is pressed, the program closes.
The following program is written entirely in [Form1] code.

```
using System;
using System.Diagnostics;
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Reflection;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        private TcpClient LanSocket;                // LAN socket
        private String MsgBuf = "";                 // Received data
        private const long Timeout_default = 2000;  // Receive timeout default time (ms)

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            String ip = "192.168.1.1";              // IP address
            String port = "23";                     // Port number
            int i;

            button1.Enabled = false;                // Disable buttons during communication

            // Connect
            if (OpenInterface(ip, port))
            {
                // Open text file to output test result values
                Assembly myAssembly = Assembly.GetEntryAssembly();
                string path = Path.GetDirectoryName(myAssembly.Location);
                                        // Output path (where executable file is located)
                StreamWriter fp = new StreamWriter(path + "\\data.csv", false, Encoding.UTF8);

                // Set measurement conditions
                SendMsg(":FUNC RV");                // Select RV function
                SendMsg(":TRIG:SOUR INT");          // Select internal triggering
                SendMsg(":INIT:CONT ON");           // Continuous measurement ON

                for (i = 1; i <= 10; i++)
                {
                    SendQueryMsg(":FETCH?");        // Get the latest measurement results
                    fp.Write(MsgBuf + "\r\n");      // Write to file
                }

                // Close the file
                fp.Close();

                // Disconnection
                CloseInterface();
```

```
            // Enaable buttons
            button1.Enabled = true;
        }
    }

    // Connect
    private Boolean OpenInterface(String ipaddress, String port)
    {
        Boolean ret = false;
        IPAddress ip = new IPAddress(0);          // IP address

        try
        {
            if (System.Net.IPAddress.TryParse(ipaddress, out ip))
            {
                LanSocket = new TcpClient();      // Create LAN socket object
                LanSocket.NoDelay = true;     // Disable transmission delay (Nagle algorithm)
                LanSocket.Connect(ip, Convert.ToInt32(port));   // LAN socket open
                ret = true;
            }
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message);
        }
        return ret;
    }

    // Disconnection
    private Boolean CloseInterface()
    {
        Boolean ret = false;

        try
        {
            LanSocket.Close();                    // LAN socket close
            ret = true;
        }
        catch (Exception e)
        {
            MessageBox.Show(e.Message);
        }
        return ret;
    }

    // Send command
    private Boolean SendMsg(String strMsg)
    {
        Boolean ret = false;
        Byte[] sendBuffer;

        try
        {
            strMsg += "\r\n";                             // Add terminator "CR+LF"
            sendBuffer = Encoding.Default.GetBytes(strMsg); // Convert to byte type
            LanSocket.GetStream().Write(sendBuffer, 0, sendBuffer.Length);
                                                          // Write to send buffer
            ret = true;
```

```
            }
            catch (Exception e)
            {
                MessageBox.Show(e.Message);
            }
            return ret;
        }

        // Receive command response
        private Boolean ReceiveMsg(long timeout = Timeout_default)
        {
            Boolean ret = false;
            Byte[] rcv = new Byte[1024];
            System.Diagnostics.Stopwatch sw = new System.Diagnostics.Stopwatch();

            try
            {
                MsgBuf = "";                                    // Clear received data
                sw.Start();                                     // Start stopwatch for timeout

                // Loop until terminator "LF" is received
                while (true)
                {
                    // Received one character at a time
                    if (LanSocket.GetStream().DataAvailable)
                                                // Read if data is in the receive buffer
                    {
                        LanSocket.GetStream().Read(rcv, 0, 1);
                                                // Read one character from the receive buffer
                        if (Convert.ToChar(rcv[0]) == '\n')
                                                // Terminate when terminator "LF" is received
                        {
                            break;
                        }
                        else if (Convert.ToChar(rcv[0]) == '\r')
                                                // Ignore terminator 'CR'
                        {
                            ;
                        }
                        else
                        {
                            MsgBuf += Convert.ToChar(rcv[0]);  // Save received data
                        }
                    }
                    // Timeout processing
                    if (sw.ElapsedMilliseconds > timeout)
                    {
                        MsgBuf = "Timeout";
                        MessageBox.Show(MsgBuf);
                        return ret;
                    }
                }
                sw.Stop();                                      // Stop the stopwatch
                ret = true;
            }
            catch (Exception e)
            {
                MsgBuf = "Error";
                MessageBox.Show(e.Message);
```

```
            }
            return ret;
        }

        // Send and receive commands
        private Boolean SendQueryMsg(String strMsg, long timeout = Timeout_default)
        {
            Boolean ret = false;

            ret = SendMsg(strMsg);            // Send command
            if (ret)
            {
                ret = ReceiveMsg(timeout);    // Receive response after successful transmission
            }
            return ret;
        }
    }
}
```

HIOKI

www.hioki.com/

All regional contact information

HIOKI E.E. CORPORATION
81 Koizumi, Ueda, Nagano 386-1192 Japan

2402 EN

Edited and published by HIOKI E.E. CORPORATION Printed in Japan

Europe only

・EU declaration of conformity can be downloaded from our website.
・Contact in Europe: HIOKI EUROPE GmbH
Helfmann-Park 2, 65760 Eschborn, Germany hioki@hioki.eu