BT4560
BT4560-50

HIOKI

Communication Command
Instruction Manual

BATTERY IMPEDANCE METER

The latest edition of the instruction manual

- ✓ This instruction manual explains the communication commands for Model BT4560, BT4560-50 Battery Impedance Meter.
- ✓ Before using BT4560 or BT4560-50, be sure to read the instruction manual of BT4560, BT4560-50.
- ✓ For details regarding the command settings, please refer to the instruction manual for Model BT4560, BT4560-50.
- ✓ 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)

EN

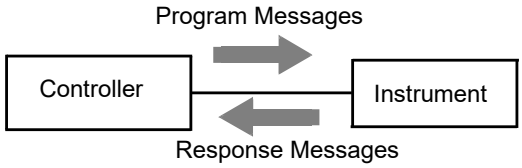
Mar. 2024 Revised edition 1
BT4560A987-01

Contents

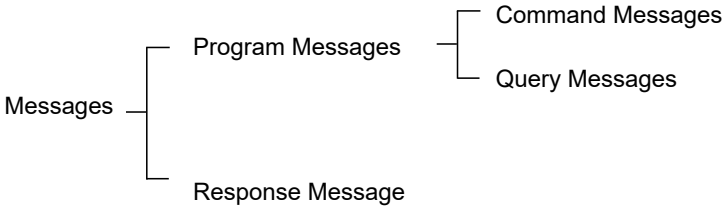
1 Introduction

If the [COMMAND MONITOR] function is used at the time of program creation, commands and responses will be conveniently displayed on the measurement screen. For information on the [COMMAND MONITOR] 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.2)”, “Separators (p.3)”, “Data Formats (p.4)”

Response Messages

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

Header ON	**:RANGE 3.0000E-3**
Header OFF	**3.0000E-3**
	(The current measurement range is 3mΩ)

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?**.

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.

CALIBRATION?	OK (long form)
CAL	OK (short form)
CALIB	Error
CA	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
  :ESE0
- **Headers for Compound Commands**
  These headers consist of multiple simple command type headers separated by colons “:”
  :SYSTem:RESet
- **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?
:CALCulate:LIMit:BEEPer?

Message Terminators

This instrument recognizes the following message terminators (delimiters):

- CR
- CR+LF

Also the terminator for response messages is as follows:

- CR+LF

Separators

(1) Message Unit Separator

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

:FREQUENCY 1000;*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).

:SYSTEM:HEADER OFF

(3) Data Separator

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

:SAMPle:RATE V,MED

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.

:SYSTEM:HEADER 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.

:ESE0 10

:FETCH?
+1.06571E-03

Compound Command Header Omission

When several commands having a common header are combined to form a compound command (for example, **:CALCulate:LIMit:RESistance** and **:CALCulate:LIMit:VOLTage**), if they are written together in sequence, the common portion (here, **:CALCulate:LIMit:**) may 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

:CALCulate:LIMit: RESistance 1.0E-2,5.0E-3 ;:CALCulate:LIMit: VOLTage 5.0,4.0

Compacted expression

:CALCulate:LIMit: RESistance 1.0E-2,5.0E-3; VOLTage 5.0,4.0

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:

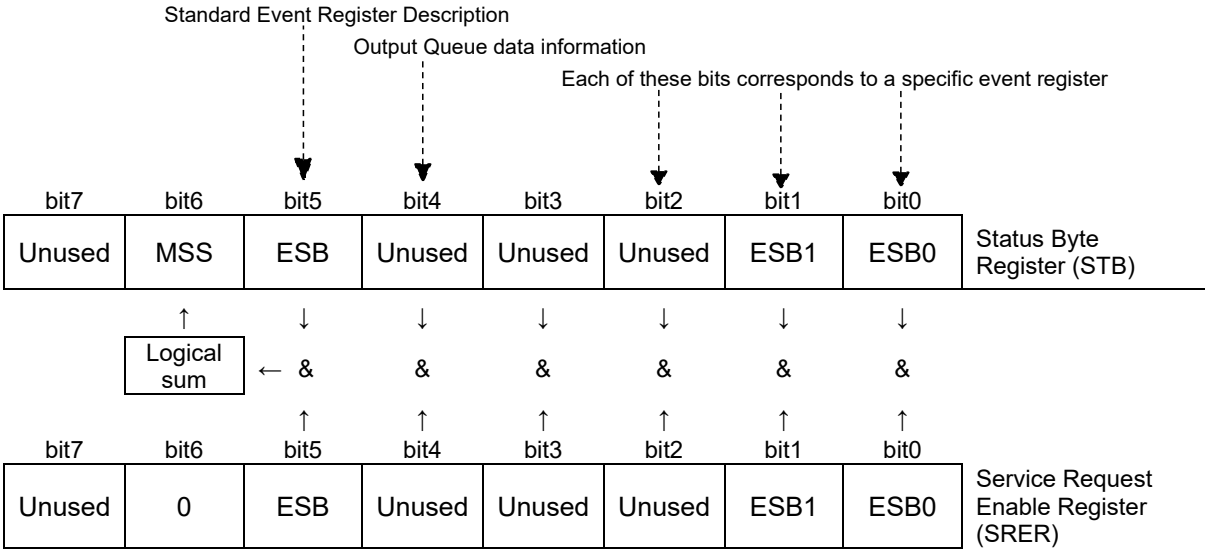
- Power on

Input Buffer

The input buffer capacity of the instrument is 256 bytes.
If 256 bytes are allowed to accumulate in this buffer so that it becomes full, the interface will not accept data beyond 256 bytes.

Note: Ensure that no command ever exceeds 256 bytes.

Status Byte Register



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.

Status Byte Register (STB)

When any Status Byte Register bit enabled by the Service Request Enable Register has switched from 0 to 1, the MSS bit becomes 1.

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	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	Unused
Bit 3		Unused
Bit 2		Unused
Bit 1	ESB1	Event Status (logical sum) bit 1 This is the logical sum of Event Status Register 1.
Bit 0	ESB0	Event Status (logical sum) bit 0 This is the logical sum of Event Status Register 0.

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.9)

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

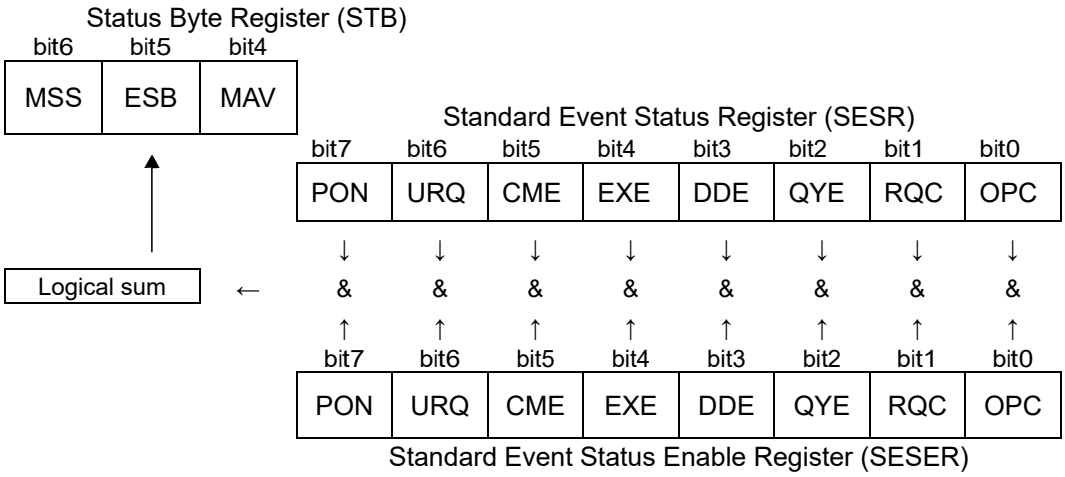
- When a *CLS command is executed
- When an event register query (*ESR?) is executed
- When the instrument is powered on

Bit 7	PON	**Power-On Flag** Set to 1 when the power is turned on, or upon recovery from an outage.
Bit 6	(Unused) URQ	**User Request**
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 • Invalid parameter format • 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. • 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	(Unused) 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	(Unused) 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 an attempt has been made to read an empty output queue (GP-IB only) • When the data overflows the output queue • When data in the output queue has been lost • When the next command is received while there is data in the output queue
Bit 1	(Unused) RQC	**Request Control**
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 (ESR0 and ESR1)

This instrument provides two Event Status Registers for controlling events. Each event register is an 8-bit register.

When any bit in one of these Event Status Registers enabled by its corresponding Event Status Enable Register is set to 1, the following happens:

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

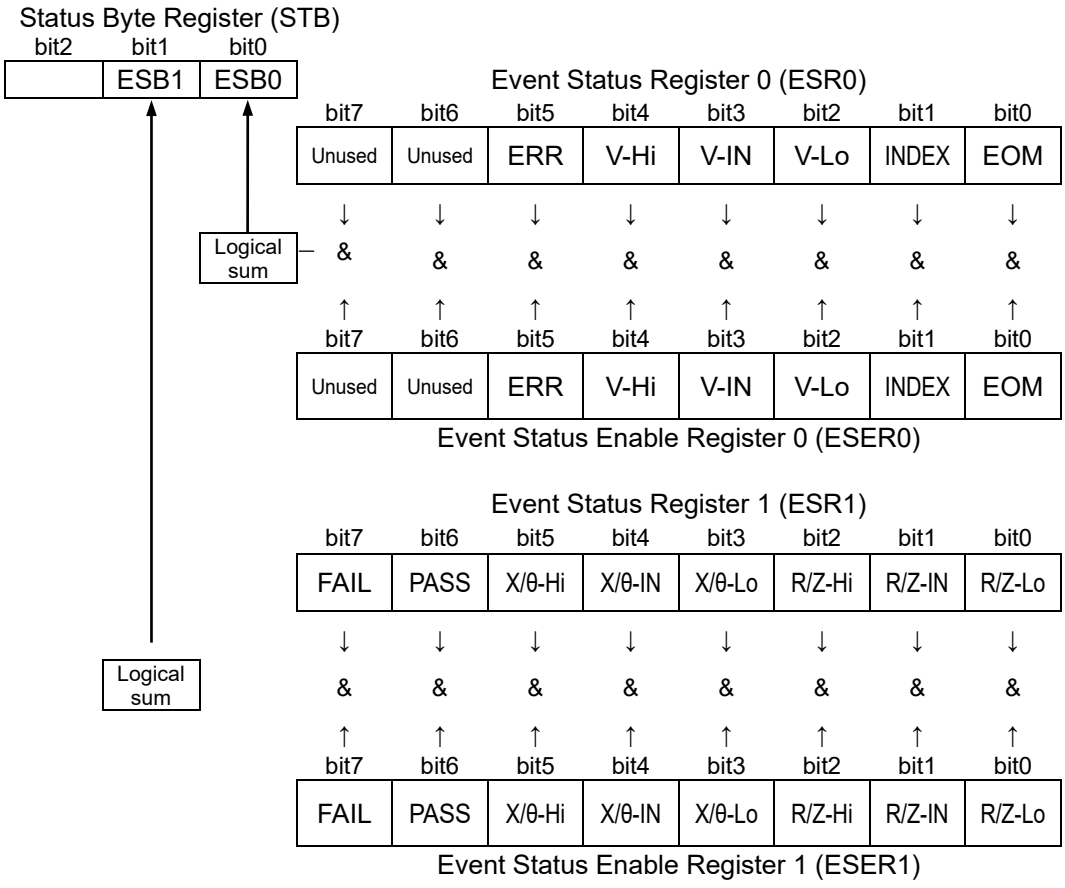
Event Status Registers 0 and 1 are cleared in the following situations:

- When a ***CLS** command is executed
- When an Event Status Register query (**:ESR0?** or **:ESR1?**) is executed
- When the instrument is powered on

Event Status Register 0 (ESR0)		
Bit 7		Unused
Bit 6		Unused
Bit 5	ERR	Measurement Fault
Bit 4	V-Hi	V Measurement High Comparator Result
Bit 3	V-IN	V Measurement IN Comparator Result
Bit 2	V-Lo	V Measurement Low Comparator Result
Bit 1	INDEX	End of Reading
Bit 0	EOM	End of Measurement

Event Status Register 1 (ESR1)		
Bit 7	FAIL	Total Judgment FAIL
Bit 6	PASS	Total Judgment PASS
Bit 5	X/θ-Hi	X/θ Measurement High Comparator Result
Bit 4	X/θ-IN	X/θ Measurement IN Comparator Result
Bit 3	X/θ-Lo	X/θ Measurement Low Comparator Result
Bit 2	R/Z-Hi	R/Z Measurement High Comparator Result
Bit 1	R/Z-IN	R/Z Measurement IN Comparator Result
Bit 0	R/Z-Lo	R/Z Measurement Low Comparator Result

Event Status Registers 0 (ESR0) and 1 (ESR1), and Event Status Enable Registers 0 (ESER0) and 1 (ESER1)



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 Status Register 0	:ESR0?	-
Event Status Enable Register 0	:ESE0?	:ESE0
Event Status Register 1	:ESR1?	-
Event Status Enable Register 1	:ESE1?	:ESE1

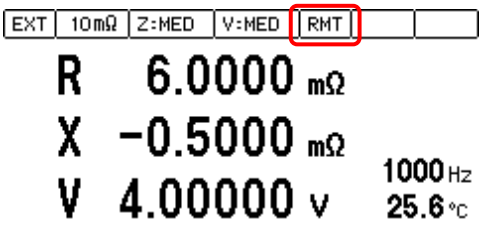
Initialization Items

Item \ Initialization Method	At Power-on	*RST Command	:SYSTem:RESET Command	*CLS Command	Factory Default
RS-232C setting (baud rate)	-	-	-	-	9600
Device-specific functions (range, etc.)	-	●*1	●	-	●
Output Queue	●	-	-	-	●
Input buffer	●	-	-	-	●
Status Byte Register	●	-	-	●	●
Event registers	●*2	-	-	●	●
Enable register	●	-	-	-	●
Current path	●	-	-	-	●
Headers on/off	OFF	-	-	-	OFF

*1. Except the zero adjustment value and the saved data.
*2. Except the PON bit (bit 7).

Local Function

The Remote state is entered during communication. [RMT] is displayed in the measurement display and operation keys are disabled.



Canceling the Remote state

• Pressing the [LOCAL] key on the front panel cancels the Remote state and enables key operations.

LOCAL

• Sending **:SYSTem:LOCal** command can also cancel the Remote state.

Command Execution Time

Command execution time indicates the time for analyzing and processing long form commands.

- Display delays may occur depending on the frequency of communication processes and process contents.
- All commands except ***TRG** and **:INIT** are processed sequentially.
- In communications with the controller, time must be added for data transmission. Transfer time depends on the controller.
  The 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]
  If a measurement value is 11 characters, a 1 data transfer time will be 11/T.
  (Example) For 9600 bps, 11/(9600/10) = Approx. 11 ms

Command	Execution time (except communication time) *1
:ADJust? ALL	15 s or less
:FETCh?	4 ms or less
:READ?	Measurement time + 4 ms or less
:LOAD	90 ms or less
:CALibration?	Calibration time + 6 ms or less
***RST**	75 ms or less

*1 The values indicate an execution time when the instrument is not under a measurement.
The execution time may increase during a measuremt.

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
- Execution Error
  When invalid character or numeric data is present

2 Message List

Message [ ]: Omissible	Data Formats [ ]: Omissible ( ): Response data	Description
Standard Commands		
*IDN?	(<Manufacturer name>,<Model name>, <Serial number>, <Software version>)	Queries the Instrument ID.
*RST		Initializes the Instrument.(Normal Reset)
*TST?	(0 to 1)	Initiates a self-test and queries the result.
*OPC		Sets an OPC after execution completion.
*OPC?	(1)	Queries execution completion.
*WAI		Wait for operations to finish.
*CLS		Clears the Status Byte Register and the related ques.
*ESE	0 to 255	Sets the Standard Event Status Enable Register.
*ESE?	(0 to 255)	Queries the Standard Event Status Enable Register.
*ESR?	(0 to 255)	Queries the Standard Event Status Register.
*SRE	0 to 255	Sets the Service Request Enable Register.
*SRE?	(0 to 255)	Queries the Service Request Enable Register.
*STB?	(0 to 255)	Queries the Status Byte Register.
*TRG		Executes one sampling.
Instrument Model Name		
:QPID	<Model name>	Queries the instrument model name.
Event Registers		
:ESE0	0 to 255	Sets the Event Status Enable Register 0.
:ESE0?	(0 to 255)	Queries the Event Status Enable Register 0.
:ESR0?	(0 to 255)	Queries the Event Status Register 0.
:ESE1	0 to 255	Sets the Event Status Enable Register 1.
:ESE1?	(0 to 255)	Queries the Event Status Enable Register 1.
:ESR1?	(0 to 255)	Queries the Event Status Register 1.
I/O		
:IO:MODE?	(NPN/PNP)	Queries the NPN/PNP switch status.
Measurement Functions		
:FUNCtion	RV/ZV/R/Z/V	Sets the measurement function.
:FUNCtion?	(RV/ZV/R/Z/V)	Queries the measurement function.
Measurement Frequency		
:FREQuency	<Frequency>	Sets the measurement frequency.
:FREQuency?	(<Frequency>)	Queries the measurement frequency.
Measurement Range		
:RANGe	<Measurement range>	Sets the measurement range.
:RANGe?	(<Measurement range>)	Queries the measurement range.
Sampling Speed		
:SAMPle:RATE	<V/Z>, <FAST/MEDium/SLOW >	Sets the sampling speed.
:SAMPle:RATE?	<V/Z> (<FAST/MEDIUM/SLOW >)	Queries the sampling speed.
Sample Delay		
:SAMPle:DELay:MODE	WAVE/VOLTage	Sets the sample delay mode.
:SAMPle:DELay:MODE?	(WAVE/VOLTAGE)	Queries the sample delay mode.
:SAMPle:DELay:WAVE	<Wavenumber> = 0.0 to 9.0	Sets the sample delay with the frequency of the alternating current signal.
:SAMPle:DELay:WAVE?	(<Wavenumber>)	Queries the sample delay frequency of the alternating current signal.
:SAMPle:DELay:VOLTage	<Deviation of voltage fluctuation > =0.001 to 10.000	Sets the sample delay with the deviation of voltage fluctuation.
:SAMPle:DELay:VOLTage?	(<Deviation of voltage fluctuation >)	Queries the sample delay deviation of voltage fluctuation.

Message [ ]: Omissible	Data Formats [ ]: Omissible ( ): Response data	Description
Voltage Limit		
:LIMiter	1/0/ON/OFF	Sets the voltage limit function.
:LIMiter?	(ON/OFF)	Queries the voltage limit function.
:LIMiter:VOLTage	<Voltage limit value> =0.01 to 5.00	Sets the voltage limit value.
:LIMiter:VOLTage?	(<Voltage limit value>)	Queries the voltage limit value.
Measurement Signal Zero Cross Stop		
:ZEOR:CROSs	1/0/ON/OFF	Sets the measurement signal zero cross stop function.
:ZERO:CROSs?	(ON/OFF)	Queries the measurement signal zero cross stop function.
Averaging		
:CALCulate:AVERage	<Averaging count>	Sets the measurement averaging.
:CALCulate:AVERage?	(<Averaging count>)	Queries the measurement averaging.
Zero Adjustment		
:ADJust?	<SPOT/ALL> (0/1)	Executes zero adjustment and queries the result.
:ADJust:CLEar		Clears zero adjustment.
:ADJust:DATA:ALL?	V/R/RV (<Adjusted value V>,<Adjusted value R at 10Hz>,<Adjusted value X at 10Hz>,<Adjusted value R at 100Hz>,<Adjusted value X at 100Hz>,<Adjusted value R at 330Hz>,<Adjusted value X at 330Hz>,<Adjusted value R at 660Hz>,<Adjusted value X at 660Hz>,<Adjusted value R at 1kHz>,<Adjusted value X at 1kHz>)	Queries the zero adjustment value (ALL).
:ADJust:DATA:SPOT?	V/R/RV (<Adjusted value V>,<Adjusted value R>,<Adjusted value X>)	Queries the zero adjustment value (SPOT).
:ADJust:STATe?	(ON/OFF)	Queries the state of zero adjustment execution.
Potential Slope Correction		
Adjust:SLOPe	1/0/ON/OFF	Sets the slope correction for AC signal response.
Adjust:SLOPe?	(ON/OFF)	Queries the slope correction for AC signal response.
Self-Calibration		
:CALibration		Executes self-calibration.
:CALibration:AUTO	1/0/ON/OFF	Sets the automatic self-calibration.
:CALibration:AUTO?	(ON/OFF)	Queries the automatic self-calibration.
Comparator		
:CALCulate:LIMit:STATe	1/0/ON/OFF	Sets the comparator.
:CALCulate:LIMit:STATe?	(ON/OFF)	Queries the comparator.
:CALCulate:LIMit:BEEPer	OFF/HL/IN/ALL	Sets the beep sound.
:CALCulate:LIMit:BEEPer?	(OFF/HL/IN/ALL)	Queries the beep sound.
:CALCulate:LIMit:ABS	1/0/ON/OFF	Sets the judgement of the voltage component comparator with the absolute value
:CALCulate:LIMit:ABS?	(ON/OFF)	Queries the judgement of the voltage component comparator with the absolute value
:CALCulate:LIMit:RESistance	<Upper limit>,<Lower limit>	Sets the upper/lower limits for the resistance component.
:CALCulate:LIMit:RESistance?	(<Upper limit>,<Lower limit>)	Queries the upper/lower limits for the resistance component.
:CALCulate:LIMit:REACtance	<Upper limit>,<Lower limit>	Sets the upper/lower limits for the reactance component.
:CALCulate:LIMit:REACtance?	(<Upper limit>,<Lower limit>)	Queries the upper/lower limits for the reactance component.
:CALCulate:LIMit:IMPedance	<Upper limit>,<Lower limit>	Sets the upper/lower limits for the impedance component.
:CALCulate:LIMit:IMPedance?	(<Upper limit>,<Lower limit>)	Queries the upper/lower limits for the impedance component.
:CALCulate:LIMit:PHASe	<Upper limit>,<Lower limit>	Sets the upper/lower limits for the phase component.

Message [ ]: Omissible	Data Formats [ ]: Omissible ( ): Response data	Description
:CALCulate:LIMit:PHASe?	(<Upper limit>,<Lower limit>)	Queries the upper/lower limits for the phase component.
:CALCulate:LIMit:VOLTage	<Upper limit>,<Lower limit>	Sets the upper/lower limits for the voltage component.
:CALCulate:LIMit:VOLTage?	(<Upper limit>,<Lower limit>)	Queries the upper/lower limits for the voltage component.
Saving and Reading Measurement Conditions		
:SAVE	Save No.	Saves the measurement conditions.
:SAVE:CLEar	Save No.	Clears the saved measurement conditions.
:LOAD	Save No.	Reads the measurement conditions.
System Reset		
:SYSTem:RESet		Initializes the instrument.(System Reset)
Output measured value after measurement completes		
:SYSTem:DATAout	1/0/ON/OFF	Sets the measured value output.
:SYSTem:DATAout?	(ON/OFF)	Queries the measured value output.
Key-Lock		
:SYSTem:KLOCk	1/0/ON/OFF	Sets the key-lock.
:SYSTem:KLOCk?	(ON/OFF)	Queries the key-lock.
Key Beeper		
:SYSTem:BEEPer	1/0/ON/OFF	Sets the key beeper.
:SYSTem:BEEPer?	(ON/OFF)	Queries the key beeper.
Communications Settings		
:SYSTem:LOCal		Cancels the communication (remote) state.
Header Presence		
:SYSTem:HEADer	1/0/ON/OFF	Sets the header presence.
:SYSTem:HEADer?	(ON/OFF)	Queries the header presence.
Serial Number		
:SYSTem:SERial?	Serial number	Queries the serial number.
LCD Settings		
:SYSTem:DISPlay:CONTrast	<Contrast> =1 to 100	Sets the contrast.
:SYSTem:DISPlay:CONTrast?	(<Contrast>)	Queries the contrast.
:SYSTem:DISPlay:BACKlight	<Brightness> =10 to 100	Sets the backlight brightness.
:SYSTem:DISPlay:BACKlight?	(<Brightness>)	Queries the backlight brightness.
Triggering		
:TRIGger:SOURce	IMMediate/EXTernal	Sets the trigger source.
:TRIGger:SOURce?	(IMMediate/EXTernal)	Queries the trigger source.
:INITiate:CONTinuous	1/0/ON/OFF	Sets continuous measurement (permits/prohibits transition to the idle state).
:INITiate:CONTinuous?	(ON/OFF)	Queries the continuous measurement.
:INITiate		Transits to the trigger waiting state.
Reading Measured Values		
:ABORt		Measurement is aborted (forcibly terminated).
:MEASure:Valid	<MR0> =1 to 7	Sets the response data to be returned from a measurement value reading query.
:MEASure:Valid?	(<MR0>)	Queries the response data for a measurement value reading query.
:FETCh?	(<Total judgment result>,<Measurement value>,<Judgment result>,<Measurement value>,<Judgment result>,…)	Queries the last measurement value.
:FETCh:TEMPerature?	(<Temperature measurement value>)	Queries the temperature measurement value.
:READ?	(<Total judgment result>,<Measurement value>,<Judgment result>,<Measurement value>,<Judgment result>,…)	Cancels the idle state and queries the measurement value after the measurement is completed.

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

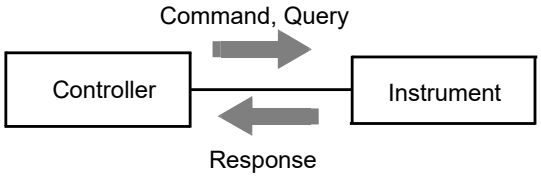
Command: The SESER mask is set to the numerical value 0 to 255. The initial value (at power-on) is 0.

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 Instrument ID (Identification Code)

Syntax

Query ***IDN?**

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

Example

```
*IDN?
HIOKI, BT4560,123456,V1.00
```

The Instrument ID is HIOKI BT4560, serial number 123456, software version 1.00.

(2) Internal Operation Command

Initialize Instrument (Normal Reset)

Syntax

Command ***RST**

Description

Command Resets the instrument to factory settings excluding the communication state, , zero adjustment value and saved data.(Normal Reset)
The initial display is displayed after initialization.

Note

The communication state is not initialized.
Use the **:SYSTem:RESet** command(p.34) to initialize the zero adjustment value and saved data.

Execute Self-Test and Query Result

Syntax

Query ***TST?**

Response <0 to 1 (NR1)>

<0> = No error
<1> = ROM error

Description

Perform the instrument self-test and return the result as NR1 value 0 or 1.
Returns zero when no error occurs.

Example

```
*TST?
1
```

A ROM error occurred.

(3) Synchronization Commands

Set OPC bit of SESR when Finished with All Pending Operations

Syntax Command ***OPC**

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

Example A;B;*OPC;C
The OPC bit of SESR is set after A and B command processing has been completed.

Respond with ASCII "1" when Finished with All Pending Operations

Syntax Query ***OPC?**
Response **1**

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

Wait for Pending Commands to Finish

Syntax Command ***WAI**

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

Example
```
:TRIG:SOUR EXT
:INIT:CONT ON
*TRG;*WAI;FETC?
```
Reads the measurement value after waiting for the measurement triggered by the *TRG command to finish.

Note The *WAI command is accepted, as it is a mandatory command under IEEE Standard 488.2-1987. However, since all the device-specific commands implemented in this instrument, except the *TRG and the :INITiate, are sequential, the *WAI command has no effect even if used.

(4) Status and Event Control Commands

Clear Event Register, Status Byte Register (Except Output Queue)

Syntax Command ***CLS**

Description Clears the event status registers. The Status Byte Register bits corresponding to the event status registers are also cleared. (SESR, ESR0, ESR1)

Note The output queue is unaffected.

Set and Query 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	Unused	CME	EXE	Unused	Unused	Unused	OPC

Example

*ESE 32
Sets bit 5 of SESER.

Query and Clear Standard Event Status Register (SESR)

Syntax

Query ***ESR?**
Response <0 to 255 (NR1)>

Description

Returns the contents of the SESR as an NR1 value from 0 to 255, then clears register contents.
The response message has no header.

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	Unused	Unused	Unused	OPC

Example

*ESR?
32
Bit 5 of the SESR has been set to 1.

Set and Query Service Request Enable Register (SRER)

Syntax

Command ***SRE** <0 to 255 (NR1)>
Query ***SRE?**
Response <0 to 255 (NR1)>

Description

Command: The SRER mask is set to the numerical value 0 to 255.
Although NRf numerical values are accepted, values to the right of the decimal are rounded to the nearest integer.
Bit 6 and unused bits 2, 3 and 7 are ignored. The data is initialized to 0 at power-on.

Query: The contents of the SRER, as set by the *SRE command, are returned as an NR1 value (0 to 255). Bit 6 and unused bits 2, 3 and 7 always return as zero.

128 bit 7	64 bit 6	32 bit 5	16 bit 4	8 bit 3	4 bit 2	2 bit 1	1 bit 0
Unused	0	ESB	Unused	Unused	Unused	ESE1	ESE0

Example

*SRE 33
Set SRER bits 0 and 5 to 1.

*SRE?
33
SRER bits 0 and 5 have been set to 1.

Query Status Byte and MSS Bit

Syntax

Query ***STB?**
Response <0 to 255 (NR1)>

Description

The contents of the STB are returned as an NR1 value (0 to 255).
The response message has no header.

128 bit 7	64 bit 6	32 bit 5	16 bit 4	8 bit 3	4 bit 2	2 bit 1	1 bit 0
Unused	MSS	ESB	Unused	Unused	Unused	ESE1	ESE0

Example

*STB?
1
STB bit 0 has been set to 1.

Request a Sample

Syntax Command ***TRG**

Description Performs one measurement when external triggering (trigger source <EXTERNAL>) is enabled.

Example
```
:TRIG:SOUR EXT
:INIT:CONT ON
*TRG;*WAI;:FETC?
```
Reads the measurement value after waiting for the measurement triggered by *TRG command to finish.

Note
- An execution error occurs when the trigger source is :TRIGger:SOURce IMMediate.
- If the continuous measurement setting is :INITiate:CONTinuous OFF, measurement is not triggered by *TRG command. Instead, use :INITiate or :READ? and input the external trigger signal to perform the measurement.
- To interrupt the measurement in process while awaiting a response to the *TRG;*WAI;:FETC?(**WAI* is replaceable with **OPC* or **OPC*?) or :READ? query, press the LOCAL key. The instrument may not clear [RMT] to return measured values after the LOCAL key is pressed. In such case, hold the LOCAL key until [RMT] disappears.

Device-Specific Commands

(1) Instrument Model Name

Query Instrument Model Name (Identification Code)

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

Example
```
:QPID
BT4560
```
The instrument model name is BT4560.

(2) Event Status Register

Set and Query Device-Specific Event Status Enable Register ESER0

Syntax Command **:ESE0** <0 to 255 (NR1)>
Query **:ESE0?**
Response <0 to 255 (NR1)>

Description Command Sets the mask pattern in Event Status Enable Register 0 (ESER0) for the Event Status Register.

Query Queries the mask pattern in Event Status Enable Register 0 (ESER0) for the Event Status Register.

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	ERR	V-Hi	V-IN	V-Lo	INDEX	EOM

Note Data initializes to zero at power-on.

Query Device-Specific Event Status Register ESR0

Syntax

Query **:ESR0?**
Response <0 to 255 (NR1)>

Note For the description of each ESR0 register, see the :ESE0 command table.
Executing ESR0? clears the contents of ESR0.

Set and Query Device-Specific Event Status Enable Register ESE1

Syntax

Command **:ESE1** <0 to 255 (NR1)>
Query **:ESE1?**
Response <0 to 255 (NR1)>

Description

Command Sets the mask pattern in Event Status Enable Register 1 (ESER1) for the Event Status Register.

Query Queries the mask pattern in Event Status Enable Register 1 (ESER1) for the Event Status Register.

128 bit 7	64 bit 6	32 bit 5	16 bit 4	8 bit 3	4 bit 2	2 bit 1	1 bit 0
FAIL	PASS	X/θ-Hi	X/θ-IN	X/θ-Lo	R/Z-Hi	R/Z-IN	R/Z-Lo

Note Data initializes to zero at power-on.

Query Device-Specific Event Status Register ESR1

Syntax

Query **:ESR1?**
Response <0 to 255 (NR1)>

Note For the description of each ESR1 register, see the :ESE1 command table.
Executing ESR1? clears the contents of ESR1.

(3) I/O

Query NPN/PNP Switch Status

Syntax

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

Example

```
:IO:MODE?
NPN
```

(4) Measurement Functions

Set and Query Measurement Functions

Syntax

Command **:FUNCtion** <RV / ZV / R / Z / V>
Query **:FUNCtion?**
Response <RV / ZV / R / Z / V>

RV…(R,X,V,T) function (resistance, reactance, voltage, temperature)
ZV…(Z,θ,V,T) function (impedance, phase angle, voltage, temperature)
R…..(R,X,T) function (resistance, reactance, temperature)
Z…..(Z,θ,T) function (impedance, phase angle, temperature)
V…..(V,T) function (voltage, temperature)

Example

:FUNC RV
Set the function to (R,X,V,T).

:FUNC?
RV
The function has been set to (R,X,V,T).

(5) Measurement Frequency

Set and Query Measurement Frequency

Syntax

Command **:FREQuency** <Frequency>
Query **:FREQuency?**
Response **<Frequency>**

Description

BT4560:<Frequency(Hz)> = **0.10 to 1050 (NR3)**
BT4560-50:<Frequency(Hz)> = **0.01 to 1050 (NR3)**

Example

:FREQ 1000
Set the measurement frequency to 1000 (Hz).

:FREQ?
1000
The measurement frequency has been set to 1000 (Hz).

Note

- Even when the settings are changed during a measurement with the internal trigger and low-frequency settings, the setting change is not applied immediately due to the long measurement time.
  To apply the change immediately, it is necessary to abort the measurement to enter the instrument into the idle state.
  To enter the instrument into the idle state, follow the procedure below:

  1. Deactivate the continuous measurement. (command: :INITiate:CONTinuous OFF)
  2. Abort the measurement. (command: :ABORt)
  3. Require a measurement-interruption response. (query: *OPC?)

(6) Measurement range

Set and Query Measurement Range

Syntax

Command **:RANGe** <Measurement range>
Query **:RANGe?**
Response <Measurement range (Ω)>

Description

Command <Measurement range (Ω)> = 0.0～120.0E-03(NR3)

- 0.0 ≤ <Measurement range (Ω)> ≤ 3.0E-03
  Measurement range is set to 3mΩ.
- 3.0E-03 < <Measurement range (Ω)> ≤ 10.0E-03
  Measurement range is set to 10mΩ.
- 10.0E-03 < <Measurement range (Ω)> ≤ 120.0E-03
  Measurement range is set to 100mΩ.

Query Returns the currently set measurement range as the following format.
<Measurement range (Ω)> = 3.0000E-3/10.0000E-3/100.000E-3

Example

:RANG 100.00E-03
Set the measurement range to 100mΩ.

:RANG?
100.000E-3
The measurement range has been set to 100mΩ.

(7) Sampling Speed

Set and Query Sampling Speed

Syntax

Command **:SAMPle:RATE <V / Z>,**<Measurement speed>

Query **:SAMPle:RATE?** <V / Z>

Response <Measurement speed>

<V> = V measurement sampling
<Z> = Z measurement sampling
<Measurement speed> = FAST / MEDium / SLOW

Example

:SAMP:RATE V, MED
Set the V measurement sampling speed to MEDIUM.

:SAMP:RATE? V
MEDIUM
The V measurement sampling speed has been set to MEDIUM.

(8) Sample Delay

Set and Query Sample Delay Mode

Syntax

Command **:SAMPle:DELay:MODE** <WAVE / VOLTage>
Query **:SAMPle:DELay:MODE?**
Response <WAVE / VOLTAGE>

<WAVE> = Sets the sample delay with the frequency of the alternating current signal.
<VOLTage> = Sets the sample delay with the deviation of voltage fluctuation.

Example

:SAMP:DEL:MODE WAVE
Sets the mode to execute the sample delay with the frequency of the alternating current signal.
:SAMP:DEL:MODE?
WAVE
The mode has been set to execute the sample delay with the frequency of the alternating current signal.

Set and Query Sample Delay with the frequency of the alternating current signal

Syntax

Command **:SAMPle:DELay:WAVE** <Wavenumber>
Query **:SAMPle:DELay:WAVE?**
Response <Wavenumber>

<Wavenumber (wave)> = 0.0 to 9.0(NR2)

Example

:SAMP:DEL:WAVE 6.0
Sets the delay wavenumber to 6.0.
:SAMP:DEL:WAVE?
6.0
The delay wavenumber has been set to 6.0.

Set and Query Sample Delay with the deviation of voltage fluctuation

Syntax

Command **:SAMPle:DELay:VOLTage** <Deviation of voltage fluctuation>
Query **:SAMPle:DELay:VOLTage?**
Response <Deviation of voltage fluctuation>

<Deviation of voltage fluctuation (mV)> = 0.001 to 10.000 (NR2)

Example

:SAMP:DEL:VOLT 0.1
Sets the delay deviation of voltage fluctuation to 0.1mV.
:SAMP:DEL:VOLT?
0.100
The delay deviation of voltage fluctuation has been set to 0.1mV.

(9) Potential Slope Correction

Set and Query Potential Slope Correction

Syntax

Command	**:ADJust:SLOPe** <1 / 0 / ON / OFF>
Query	**:ADJust:SLOPe?**
Response	<ON / OFF>

Example

```
:ADJ:SLOP ON
:ADJ:SLOP?
ON
```

(10) Voltage Limit

Set and Query Voltage Limit Function

Syntax

Command	**:LIMiter** <1 / 0 / ON / OFF>
Query	**:LIMiter?**
Response	<ON / OFF>

Example

```
:LIM ON
:LIM?
OFF
```

Set and Query Voltage Limit Function

Syntax

Command	**:LIMiter:VOLTage** <Voltage limit value>
Query	**:LIMiter:VOLTage?**
Response	<Voltage limit value>

<Voltage limit value (V)> = 0.01 to 5.00 (NR2)

Example

```
:LIM:VOLT 5.00
:LIM:VOLT?
5.00
```

(11) Measurement Signal Zero Cross Stop

Set and Query Measurement Signal Zero Cross Stop Function

Syntax

Command	**:ZERO:CROSs** <1 / 0 / ON / OFF>
Query	**:ZERO:CROSs?**
Response	<ON / OFF>

Example

```
:ZERO:CROSs ON
:LIM:VOLT?
OFF
```

(12) Averaging

Set and Query Measurement Averaging

Syntax

Command **:CALCulate:AVERage** <Count>
Query **:CALCulate:AVERage?**
Response <Count>

<Count> = 1 to 99 (NR1)
Only the number of impedance measurement values specified in <Count> are averaged and output.

Example

:CALC:AVER 10
:CALC:AVER?
10

(13) Zero Adjustment

Execute Zero Adjustment and Query Result

Syntax

Query **:ADJust?** <SPOT / ALL>
Response <0 / 1>

Description

<SPOT>= Executes zero adjustment for the frequency and voltage measurement in the currently set range.
<ALL> = Executes zero adjustment for the currently set all frequency and voltage measurement.
<0> = Indicates zero adjustment succeeded.
<1> = Indicates that zero adjustment has failed.

For information on zero adjustment, see the instrument instruction manual.

Example

ADJ? SPOT
0
Zero adjustment is executed in the SPOT setting and completed normally.

Clear Zero Adjustment

Syntax

Command **:ADJust:CLEar**

Description

Clears zero adjustment.

Example

ADJ:CLE

Query Zero Adjustment Value (ALL)

Syntax

Query **:ADJust:DATA:ALL?** <V / R / RV>

Response <Voltage adjusted value>, <Resistance adjusted value at 100Hz>, <Reactance adjusted value at 100Hz>, <Resistance adjusted value at 1kHz>, <Reactance adjusted value at 1kHz>

Description

Query

<V> = Queries the voltage adjusted value.

<R> = Queries the resistance/reactance adjusted value at 10Hz, 100Hz, 330Hz, 660Hz and 1kHz.

<RV> = Queries the voltage adjusted value, and the resistance/reactance adjusted value at 10Hz, 100Hz, 330Hz, 660Hz and 1kHz.

Response

- Response for :ADJust:DATA:ALL? V

<Voltage adjusted value>

- Response for :ADJust:DATA:ALL? R

<Resistance adjusted value at 10Hz>, <Reactance adjusted value at 10Hz>, <Resistance adjusted value at 100Hz>, <Reactance adjusted value at 100Hz>, <Resistance adjusted value at 330Hz>, <Reactance adjusted value at 330Hz>, <Resistance adjusted value at 660Hz>, <Reactance adjusted value at 660Hz>, <Resistance adjusted value at 1kHz>, <Reactance adjusted value at 1kHz>

- Response for :ADJust:DATA:ALL? RV

<Voltage adjusted value>, <Resistance adjusted value at 10Hz>, <Reactance adjusted value at 10Hz>, <Resistance adjusted value at 100Hz>, <Reactance adjusted value at 100Hz>, <Resistance adjusted value at 330Hz>, <Reactance adjusted value at 330Hz>, <Resistance adjusted value at 660Hz>, <Reactance adjusted value at 660Hz>, <Resistance adjusted value at 1kHz>, <Reactance adjusted value at 1kHz>

Example

ADJ:DATA:ALL? V
+1.10000E-03
The voltage adjusted value has been set to 1.1mV.
ADJ:DATA:ALL? R
+1.10000E-04,+5.00000E-06,+2.50000E-04,+2.0000E-05,+4.50000E-04,+4.0000E-05,+4.00000E-04,+5.0000E-05,+1.50000E-04,+1.00000E-05
The resistance/reactance adjusted value at 10Hz has been set to 0.11mΩ/0.005mΩ.
The resistance/reactance adjusted value at 100Hz has been set to 0.25mΩ/0.02mΩ.
The resistance/reactance adjusted value at 330Hz has been set to 0.45mΩ/0.04mΩ.
The resistance/reactance adjusted value at 660Hz has been set to 0.40mΩ/0.05mΩ.
The resistance/reactance adjusted value at 1kHz has been set to 0.15mΩ/0.01mΩ.
ADJ:DATA:ALL? RV
+1.10000E-03,+1.10000E-04,+5.00000E-06,+2.50000E-04,+2.0000E-05,+4.50000E-04,+4.0000E-05,+4.00000E-04,+5.0000E-05,+1.50000E-04,+1.00000E-05,+4.00000E-04,+5.0000E-05,+1.50000E-04,+1.00000E-05
The voltage adjusted value has been set to 1.1mV.
The resistance/reactance adjusted value at 10Hz has been set to 0.11mΩ/0.005mΩ.
The resistance/reactance adjusted value at 100Hz has been set to 0.25mΩ/0.02mΩ.
The resistance/reactance adjusted value at 330Hz has been set to 0.45mΩ/0.04mΩ.
The resistance/reactance adjusted value at 660Hz has been set to 0.40mΩ/0.05mΩ.
The resistance/reactance adjusted value at 1kHz has been set to 0.15mΩ/0.01mΩ.

Query Zero Adjustment Value (SPOT)

Syntax

Query **:ADJust:DATA:SPOT?** <V / R / RV>

Response <Voltage adjusted value>, <Resistance adjusted value>, <Reactance adjusted value>

Description

Query

<V> = Queries the voltage adjusted value.

<R> = Queries the resistance/reactance adjusted value at the currently set frequency.

<RV> = Queries the voltage adjusted value, and the resistance/reactance adjusted value at the currently set frequency.

Response

- Response for :ADJust:DATA:ALL? V

<Voltage adjusted value>

- Response for :ADJust:DATA:ALL? R

<Resistance adjusted value>, <Reactance adjusted value>

- Response for :ADJust:DATA:ALL? RV

<Voltage adjusted value>, <Resistance adjusted value>, <Reactance adjusted value>

Example

ADJ:DATA:SPOT? V
+1.10000E-03
The voltage adjusted value has been set to 1.1mV.
ADJ:DATA: SPOT? R
+2.50000E-04,+2.0000E-04
The resistance/reactance adjusted value has been set to 0.25mΩ/0.20mΩ.
ADJ:DATA: SPOT? RV
+1.10000E-03,+2.50000E-04,+2.0000E-04
The voltage adjusted value has been set to 1.1mV.
The resistance/reactance adjusted value has been set to 0.25mΩ/0.20mΩ.

Query Zero Adjustment State

Syntax

Query **:ADJust:STATe?**

Response <ON / OFF>

Description

<ON> = The state of zero adjustment execution has been set to effective.

<OFF> = The state of zero adjustment execution has been set to disabled.

Example

ADJ:STAT?
OFF
The state of zero adjustment execution has been set to disabled.

(14) Self-Calibration

Execute Self-Calibration

Syntax

Command **:CALibration**

Note If this command is received while measuring, self-calibration executes after the measurement is finished.

Execute and Set Self-Calibration

Syntax

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

<ON> = AUTO Self-Calibration selected
Self-calibration is set to be executed at the time of voltage measurement.

<OFF> = MANUAL Self-Calibration selected
Self-calibration is set to be executed manually.
Self-calibration is executed with the :CALibration command or external CAL signal.

Example

```
:CAL:AUTO OFF
:CAL:AUTO?
OFF
```

(15) Comparator

Execute and Query Comparator

Syntax

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

Example

```
:CALC:LIM:STAT ON
:CALC:LIM:STAT?
ON
```

Set and Query Beeper

Syntax

Command **:CALCulate:LIMit:BEEPer** <Condition>
Query **:CALCulate:LIMit:BEEPer?**
Response <Condition>

<Condition> = OFF / HL / IN / ALL

OFF	No buzzer
HL	Beeps when the value is outside of the upper or lower limit range.
IN	Beeps when the value is within upper and lower limit range.
ALL	Always beeps.

Example

```
:CALC:LIM:BEEP IN
:CALC:LIM:BEEP?
IN
```

Set and Query Judgement of Voltage Component Comparator with Absolute Value

Syntax

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

Example

:CALC:LIM:ABS ON
Sets the judgement of the voltage component comparator to be performed with the absolute value.
:CALC:LIM:ABS?
ON
The judgement of the voltage component comparator has been set to be performed with the absolute value.

Set and Query Resistance Component Comparator Upper/Lower Limit

Syntax

Command **:CALCulate:LIMit:RESistance** <Upper limit>,<Lower limit>
Query **:CALCulate:LIMit: RESistance?**
Response <Upper limit (Ω)>,<Lower limit (Ω)>

Command
<Upper limit (Ω)> = -3.00000E-03 to +1.20000E-01(NR3), or OFF
<Lower limit (Ω)> = -3.00000E-03 to +1.20000E-01(NR3), or OFF
If the upper limit is set to a value out of above range, the upper limit becomes OFF.
If the lower limit is set to a value out of above range, the lower limit becomes OFF.

Query
<Upper limit (Ω)> = -3.00000E-03 to +1.20000E-01(NR3), or OFF
<Lower limit (Ω)> = -3.00000E-03 to +1.20000E-01(NR3), or OFF

Example

:CALC:LIM:RES 0.1, 0.05
The upper and lower limits (regardless of the range) are 100mΩ and 50mΩ, respectively.
:CALC:LIM:RES 5.0,0.05
:CALC:LIM:RES?
OFF,+5.00000E-02
The upper and lower limits (regardless of the range) are OFF and 50mΩ, respectively.
:CALC:LIM:RES 0.1,OFF
:CALC:LIM:RES?
+1.00000E-01,OFF
The upper and lower limits (regardless of the range) are 100mΩ, and OFF, respectively.

Note

If the upper limit is set to a value smaller than the lower limit, the upper limit becomes the same value as the lower.

Set and Query Reactance Component Comparator Upper/Lower Limit

Syntax

Command **:CALCulate:LIMit:REACtance** <Upper limit>,<Lower limit>
Query **:CALCulate:LIMit: REACtance?**
Response <Upper limit (Ω)>,<Lower limit (Ω)>

Command
<Upper limit (Ω)> = -1.20000E-01 to +1.20000E-01(NR3), or OFF
<Lower limit (Ω)> = -1.20000E-01 to +1.20000E-01(NR3), or OFF
If the upper limit is set to a value out of above range, the upper limit becomes OFF.
If the lower limit is set to a value out of above range, the lower limit becomes OFF.

Query
<Upper limit (Ω)> = -1.20000E-01 to +1.20000E-01(NR3), or OFF
<Lower limit (Ω)> = -1.20000E-01 to +1.20000E-01(NR3), or OFF

Example

:CALC:LIM:REAC 0.1, 0.05
The upper and lower limits (regardless of the range) are 100mΩ and 50mΩ, respectively.
:CALC:LIM:REAC 5.0,0.05
:CALC:LIM:REAC?
OFF,+5.00000E-02
The upper and lower limits (regardless of the range) are OFF and 50mΩ, respectively.
:CALC:LIM:REAC 0.1,OFF
:CALC:LIM:REAC?
+1.00000E-01,OFF
The upper and lower limits (regardless of the range) are 100mΩ, and OFF, respectively.

Note

If the upper limit is set to a value smaller than the lower limit, the upper limit becomes the same value as the lower.

Set and Query Impedance Component Comparator Upper/Lower limit

Syntax

Command **:CALCulate:LIMit:IMPedance** <Upper limit>,<Lower limit>
Query **:CALCulate:LIMit: IMPedance?**
Response <Upper limit (Ω)>,<Lower limit (Ω)>

Command
<Upper limit (Ω)> = 0 to +1.20000E-01(NR3), or OFF
<Lower limit (Ω)> = 0 to +1.20000E-01(NR3), or OFF
If the upper limit is set to a value out of above range, the upper limit becomes OFF.
If the lower limit is set to a value out of above range, the lower limit becomes OFF.

Query
<Upper limit (Ω)> = 0 to +1.20000E-01(NR3), or OFF
<Lower limit (Ω)> = 0 to +1.20000E-01(NR3), or OFF

Example

:CALC:LIM:IMP 0.10, 0.05
The upper and lower limits (regardless of the range) are 100mΩ and 50mΩ, respectively.
:CALC:LIM:IMP 5.0,0.05
:CALC:LIM:IMP?
OFF,+5.00000E-02
The upper and lower limits (regardless of the range) are OFF and 50mΩ, respectively.
:CALC:LIM:IMP 0.1,OFF
:CALC:LIM:IMP?
+1.00000E-01,OFF
The upper and lower limits (regardless of the range) are 100mΩ, and OFF, respectively.

Note

If the upper limit is set to a value smaller than the lower limit, the upper limit becomes the same value as the lower.

Set and Query Phase Component Comparator Upper/Lower limit

Syntax

Command **:CALCulate:LIMit:PHASe** <Upper limit>,<Lower limit>
Query **:CALCulate:LIMit: PHASe?**
Response <Upper limit (°)>,<Lower limit (°)>

Command
<Upper limit (°)> = -1.80000E+02 to +1.80000E+02(NR3), or OFF
<Lower limit (°)> = -1.80000E+02 to +1.80000E+02(NR3), or OFF
If the upper limit is set to a value out of above range, the upper limit becomes OFF.
If the lower limit is set to a value out of above range, the lower limit becomes OFF.

Query
<Upper limit (°)> = -1.80000E+02 to +1.80000E+02(NR3), or OFF
<Lower limit (°)> = -1.80000E+02 to +1.80000E+02(NR3), or OFF

Example

:CALC:LIM:PHAS 90.0, -90.0
The upper and lower limits are 90.0° and –90.0°, respectively.
:CALC:LIM:PHAS 300,-90.0
:CALC:LIM:PHAS?
OFF,-9.00000E+01
The upper and lower limits (regardless of the range) are OFF and -90.0°, respectively.
:CALC:LIM:PHAS 90.0,OFF
:CALC:LIM:PHAS?
+9.00000E+01,OFF
The upper and lower limits (regardless of the range) are 90.0° and OFF, respectively.

Note

If the upper limit is set to a value smaller than the lower limit, the upper limit becomes the same value as the lower.

Set and Query Voltage Component Comparator Upper/Lower limit

Syntax

Command **:CALCulate:LIMit:VOLTage** <Upper limit>,<Lower limit>
Query **:CALCulate:LIMit: VOLTage?**
Response <Upper limit (V)>,<Lower limit (V)>

Command
<Upper limit (V)> = -5.10000E+00 to +5.10000E+00(NR3), or OFF
<Lower limit (V)> = -5.10000E+00 to +5.10000E+00(NR3), or OFF
If the upper limit is set to a value out of above range, the upper limit becomes OFF.
If the lower limit is set to a value out of above range, the lower limit becomes OFF.

Query
<Upper limit (V)> = -5.10000E+00 to +5.10000E+00(NR3), or OFF
<Lower limit (V)> = -5.10000E+00 to +5.10000E+00(NR3), or OFF

Example

:CALC:LIM:VOLT 5.0, 4.0
The upper and lower limits are 5.0V and 4.0V, respectively.
:CALC:LIM:VOLT 30,4.0
:CALC:LIM:VOLT?
OFF,+4.00000E+00
The upper and lower limits are OFF and 4.0V, respectively.
:CALC:LIM:VOLT 5.0,OFF
:CALC:LIM:VOLT?
+5.00000E+00,OFF
The upper and lower limits are 5.0V and OFF, respectively.

Note

If the upper limit is set to a value smaller than the lower limit, the upper limit becomes the same value as the lower.

(16) Saving and Reading Measurement Conditions

Save and Read Measurement Conditions

Syntax

Command **:SAVE** <Save No.>
:LOAD <Save No.>
<Save No.> = 1 to 126 (NR1)

Example

```
:SAVE 10
:SAVE:CLE 10
:LOAD 5
```

Note

If :SAVE is executed when measurement conditions for the <Save No.> have previously been saved, the saved measurement conditions will be overwritten.
If :LOAD is executed specifying a <Save No.> that does not have previously saved measurement conditions, an execution error will occur.

Clear Measurement Conditions

Syntax

Command **:SAVE:CLEar** <Save No.>
<Save No.> = 1 to 126 (NR1)

Example

```
:SAVE:CLE 10
```

Note

If :SAVE:CLEar is executed specifying a <Save No.> that does not have previously saved measurement conditions, an execution error will occur.

(17) System Reset

Initialize Instrument (System Reset)

Syntax

Command **:SYSTem:RESet**

Description

Initializes all data except communication settings.(System Reset)
Panel save data and zero adjust value are also initialized.

Note

Use the *RST command (p.17) to leave the zero adjustment value and the panel save data.

(18) Measured value output

Setting and querying whether to output measured values after measurement is complete

Syntax

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

Description

Command <ON> = Outputs the measured value to the selected interface after measurement is complete.
<OFF> = Outputs the measured value only when there is a measured value load request (:FETCh? or :READ? query).
Query Responds with the setting for outputting measured values after measurement is complete.

Examples

```
:SYST:DATA ON
:SYST:DATA?
ON
```

(19) Key Beeper

Set and Query Key Beeper

Syntax

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

Example

```
:SYST:BEEP ON
:SYST:BEEP?
ON
```

(20) Key-Lock

Set and Query Key-Lock State

Syntax

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

Example

```
:SYST:KLOC ON
:SYST:KLOCK?
ON
```

(21) Communications Settings

Return to Local Control

Syntax

Command **:SYSTem:LOCal**

Description

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

Example

```
:SYST:LOC
```

(22) Header

Set and Query Header Presence

Syntax

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

Description

Sets whether the header is to be included in response messages.
The response to a message header setting query is returned as ON or OFF.

Example

```
:SYST:HEAD ON
:SYST:HEAD?
:SYSTEM:HEADER ON

:SYST:HEAD OFF
:SYST:HEAD?
:OFF
```

Note

When turning the power on, this is initialized to OFF (no header).

(23) Serial Number

Query Serial Number

Syntax

Query **:SYSTem:SERial?**

Response <Serial number>

Example

:SYST:SER?
123456789

(24) LCD Settings

Set and Query Contrast

Syntax

Command **:SYSTem:DISPlay:CONTrast** <Contrast>

Query **:SYSTem:DISPlay:CONTrast?**

Response <Contrast>

<Contrast> = <0 to 100>

Example

:DISP:CONT 50
Set the display contrast to 50%.

:DISP:CONT?
50
The display contrast has been set to 50%.

Set and Query Backlight

Syntax

Command **:SYSTem:DISPlay:BACKlight** <Brightness>

Query **:SYSTem:DISPlay:BACKlight?**

Response <Brightness>

<Brightness> = <10 to 100>

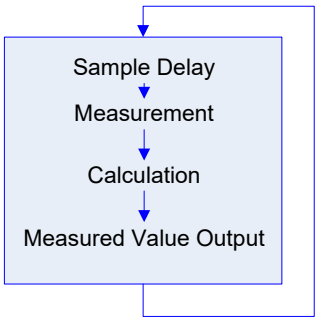
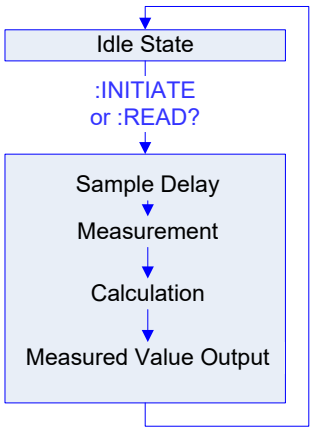
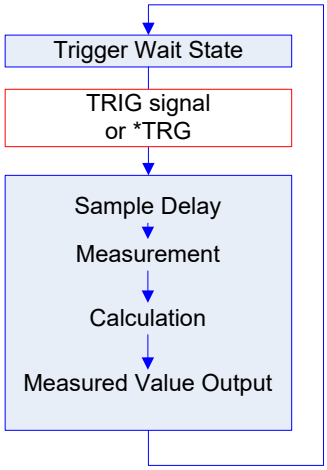
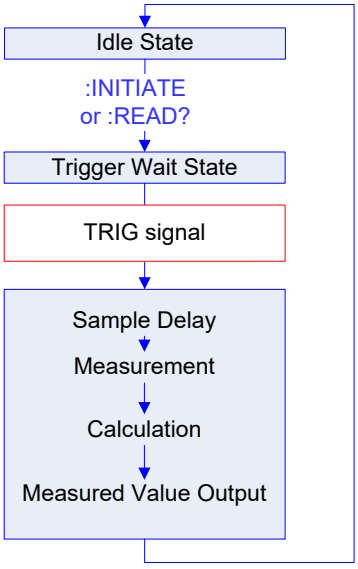
Example

:SYST:DISP:BACK 50
Set the backlight brightness to 50%.

:SYST:DISP:BACK?
50
The backlight brightness has been set to 50%.

(25) Triggering

Relationship between Trigger Source and Continuous Measurement Operation
Operation depends on the continuous measurement setting (:INITIATE:CONTINUOUS) (p.38) and the trigger source setting (:TRIGGER:SOURCE) (p.38) as follows.
See: “5 Data Exporting Methods” (p.44)

<table>
<tr><th colspan="2" rowspan="2">Measurement Flow</th><th colspan="2">Continuous Measurement 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 IMM</td><td>Free-Run state. Measurement continues automatically.
 
</td><td>Start measurement by :INITIATE (or :READ?) command.
 
</td></tr>
<tr><td>:TRIGGER:SOURCE EXT</td><td>Triggered by TRIG signal or *TRG command.
After measurement, enters the trigger wait state.
 
</td><td>Issue :INITIATE (or :READ?) command to wait for trigger.
Triggered by TRIG signal.
After measurement, enters the idle state.
 
</td></tr>
</table>

The :INITIATE:CONTINUOUS OFF can only be set by Remote command.
If this has been set to OFF, when operation is returned to the Local state or power is turned off, the :INITIATE:CONTINUOUS ON state occurs when power is turned back on.
See: “Return to Local Control” (p.35, p.11)
or Exporting measured values: “Data Exporting Methods” (p.44)

Set and Query Continuous Measurement

Syntax

Command **:INITiate:CONTinuous** <1/0/ON/OFF>
Query **:INITiate:CONTinuous?**
Response <ON/OFF>
<ON> = Continuous Measurement Enabled
<OFF> = Continuous Measurement Disabled

Description

- Continuous Measurement Enabled:
  After measurement, enters the Trigger Wait State. When there is an internal trigger (trigger source <IMMEDIATE>), the next trigger is promptly generated and enters a free run state.
- Continuous Measurement Disabled:
  After measurement, enters the Idle State instead of the Trigger Wait State. Triggering is ignored in the Idle State. Executing :INITiate enables the Trigger Wait State.
  Continuous measurement is enabled upon exiting from the Remote State.

Example

:INIT:CONT OFF

Set continuous measurement to disabled.

:INIT:CONT?
OFF

Continuous measurement has been set to disabled.

Set and Query Trigger Source

Syntax

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

Example

:TRIG:SOUR IMM

Set the trigger source to internal triggering.

:TRIG:SOUR?
IMMEDIATE

The trigger source has been set to internal triggering.

Note

- When setting the trigger source <IMMEDIATE>,:LIMITer:VOLTage will be set to ON.
- Even when the settings are changed during a measurement with the internal trigger and low-frequency settings, the setting change is not applied immediately due to the long measurement time.
  To apply the change immediately, it is necessary to abort the measurement to enter the instrument into the idle state.
  To enter the instrument into the idle state, follow the procedure below:

  1. Deactivate the continuous measurement. (command: :INITiate:CONTinuous OFF)
  2. Abort the measurement. (command: :ABORt)
  3. Require a measurement-interruption response. (query: *OPC?)

Transit to Trigger Waiting State

Syntax Command **:INITiate**

Description Switches triggering from the Idle State to the Trigger Wait State.

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

Example

:TRIG:SOUR IMM..... Set the trigger source to internal triggering.
:INIT..... Switch triggering to Trigger Waiting State.
Trigger a single measurement immediately upon internal triggering.

Example

:TRIG:SOUR EXT..... Set the trigger source to external triggering.
:INIT..... Switch triggering to Trigger Waiting State.
Trigger a single measurement when an external triggering signal is received.

Note

- When this command is received, automatically switches to :INITIATE:CONTINUOUS OFF.
- When there is an internal trigger (trigger source <IMMEDIATE>), triggering promptly occurs and enters the idle state.
- When there is an external trigger (trigger source <EXTERNAL>), the external trigger wait state is entered. When a trigger is received, a single measurement is performed and enters the idle state.

(26) Reading Measured Values

Measurement Value Formats

• Impedance (absolute value display: unit Ω)

Measured Value	Measurement Fault
+ □ . □□□□□ E-0□	See: Measurement Value Formats(Measurement Fault) (p.40)

• Phase angle (absolute value display: unit °)

Measured Value	Measurement Fault
± □ . □□□□□ E±0□	See: Measurement Value Formats(Measurement Fault) (p. 40)

• Resistance / Reactance (absolute value display: unit Ω)

Measured Value	Measurement Fault
± □ . □□□□□ E-0□	See: Measurement Value Formats(Measurement Fault) (p. 40)

• Voltage (absolute value display: unit V)

Measured Value	Measurement Fault
± □ . □□□□□ E±0□	See: Measurement Value Formats(Measurement Fault) (p. 40)

• Temperature (absolute value display: unit °C)

Measured Value	Measurement Fault
± □ . □□□□□ E±0□	See: Measurement Value Formats(Measurement Fault) (p. 40)

Measurement Value Formats(Measurement Fault)

• Impedance / Phase angle / Resistance / Reactance / Voltage

Measurement Fault	Display	Measured Value
Out of Z range	OverRange	+ 1. 00000 E+08
Impedance measurement error by drift voltage	DRIFT VOLTAGE	+ 2. 00000 E+08
Contact error between SOURCE-L and SENSE-L	CONTACT ERROR L	+ 3. 00000 E+08
Contact error between SOURCE-H and SENSE-H	CONTACT ERROR H	+ 4. 00000 E+08
Return cabole not connected	RETURN CABLE ERROR	+ 5. 00000 E+08
Out of limit voltage	OVER V LIMIT	+ 6. 00000 E+08
Out of V range	OVER VOLTAGE	+ 7. 00000 E+08
Constant current error between SOURCE-H and SOURCE-L	------	+ 8. 00000 E+08
A/D converter communication error	A/D ERROR	+ 9. 00000 E+08
Internal battery error	VREF B ERROR	+ 1. 00000 E+09
No measurement after power on		+ 2. 00000 E+09

・温度

Measurement Fault	Display	Measured Value
Over the upper limit of T range	+Over ℃	+ 1. 00000 E+08
Under the lower limit of T range	-Under ℃	+ 2. 00000 E+08
Temperature sensor not connected	--.- ℃	+ 3. 00000 E+08
No measurement after power on		+ 4. 00000 E+08

Time to receive measured values is different for the :FETCh?/:FETCh:TEMPerature? and :READ? commands.
See: Data Exporting Methods (p.44), Triggering (p.37)

Abort Measurement

Syntax

Query **:ABORt**

Description

Measurement is abort (forced termination).
:READ? cannot be abort.

Example

```
:TRIG:SOUR EXT
:INIT:CONT ON
*TRG
:ABOR
```

Aborts a measurement.

```
:TRIG:SOUR EXT
:INIT:CONT ON
*TRG;*WAI
:ABOR
```

In this case, a measurement cannot be aborted because the instrument waits the measurement to finish.

Note

An abort cannot be executed as the instrument waits until all prior commands finish if the query is sent after a *WAI command.
An abort cannot be executed as the instrument waits until all prior commands finish if the query is sent after a *OPC and *OPC? command.

Read Measurement Value and Set and Query Response Data

Syntax

Command **:MEASure:VALid** <MR0>
Query **:MEASure:VALid?**
Response <MR0>
<MR0> = 0 to 7(NR1)
MR: Measurement register (see below)

Description

Command: Uses bits in <MR0> to specify the measurement parameters to be returned in response to a query from a FETCh?/:READ? command.
1 is displayed at power-on.

Query: Returns the measurement parameters specified using bits in <MR0> in response to a query from a FETCh?/:READ? command.

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	Total judgment result	Judgment result	Measured value

Measurement register 0 (MR0)

Example

```
:MEAS:VAL 7
```

Sets to return the Measured value, Judgment result and Total judgment result.

```
:MEAS:VAL?
7
```

The Measured value, Judgment result and Total judgment result have been set to be returned.

Read Most Recent Measurement Value

Syntax

Query :FETCh?

Response <Total judgment result>,<Measurement value>,<Judgment result>
,<Measurement value>,<Judgment result>, …
See: “Measurement Value Formats” (p.40)

Description

Reads the most recent impedance and voltalge measurement. No trigger occurs.
See: Data Exporting Methods (p.44), Triggering (p.37)
For (R,X,V,T) measurement, the response is as follows.
<Total judgment result> = PASS/FAIL/OFF
<Measurement value> = R measurement value
<Judgment result> = R measurement comparator judgment result <HI/IN/LO/OFF>
<Measurement value> = X measurement value
<Judgment result> = X measurement comparator judgment result < HI/IN/LO/OFF >
<Measurement value> = V measurement value
<Judgment result> = V measurement comparator judgment result < HI/IN/LO/OFF >

Example

:MEAS:VAL 1···· Sets to return the Measured value.
:FETC?
+1.02500E-01,+1.02800E-01,+3.00000E+00

:MEAS:VAL 3···· Sets to return the Measured value and Judgment result.
:FETC?
+1.02500E-01,IN,+1.02800E-01,IN,+3.00000E+00,IN

:MEAS:VAL 7···· Sets to return the Measured value, Judgment result and Total judgment result.
:FETC?
PASS,+1.02500E-01,IN,+1.02800E-01,IN,+3.00000E+00

Note

- To interrupt the measurement in process while awaiting a response to the *TRG;*WAI;:FETC?（**WAI* is replaceable with **OPC* or **OPC*?） or :READ? query, press the LOCAL key. The instrument may not clear [RMT] to return measured values after the LOCAL key is pressed. In such case, hold the LOCAL key until [RMT] disappears.

Read Temperature Measurement Value

Syntax

Query :FETCh:TEMPerature?

Response <Measurement value> See: “Measurement Value Formats” (p.40)

Description

Reads the last (most recent) temperature measurement value.

Example

:FETC:TEMP?
+2.51000E+01

Measure (Await Triggers and Read Measurements)

Syntax

Query **:READ?**

Response <Total judgment result>,<Measurement value>,<Judgment result>
,<Measurement value>,<Judgment result>, …

See: “Measured value Formats” (p.40)

Description

Switches from the Idle State to the Trigger Wait State, then reads the next measured value.

Trigger Source	Operation
IMMediate	Reads the measured value.
EXTernal	Triggers by the external TRIG signal input, and continuously reads the measured values.

For (R,X,V,T) measurement, the response is as follows.
<Total judgment result> = PASS/FAIL/OFF
<Measurement value> = R measurement value
<Judgment result> = R measurement comparator judgment result <HI/IN/LO/OFF>
<Measurement value> = X measurement value
<Judgment result> = X measurement comparator judgment result <HI/IN/LO/OFF>
<Measurement value> = V measurement value
<Judgment result> = V measurement comparator judgment result <HI/IN/LO/OFF>

Example

:TRIG:SOUR IMM	Set the trigger source to internal triggering.
:MEAS:VAL 1	Set to return the Measured value.
:READ?	Trigger a single measurement immediately upon
+1.02500E-01,+1.02800E-01,+3.00000E+00	internal triggering.

Example

:TRIG:SOUR EXT	Set the trigger source to external triggering.
:MEAS:VAL 1	Set to return the Measured value.
:READ?	Trigger a single measurement when an external
+1.02500E-01,+1.02800E-01,+3.00000E+00	triggering signal is received.

Note

- When this command is received, automatically switches to :INITIATE:CONTINUOUS OFF.
- When there is an internal trigger (trigger source <IMMEDIATE>), triggering promptly occurs and enters the idle state.
- When there is an external trigger (trigger source <EXTERNAL>), the external trigger wait state is entered. When a trigger is received, a single measurement is performed and enters the idle state.
- After sending the :Read? query, the measurement cannot be interrupted by sending communications commands.
  Pressing the key can interrupt the measurement, resulting in the instrument responding with measured values at the time.
- The next command does not execute until measurement is finished.
- To interrupt the measurement in process while awaiting a response to the *TRG;*WAI;:FETC?（**WAI* is replaceable with **OPC* or **OPC*?） or :READ? query, press the LOCAL key. The instrument may not clear [RMT] to return measured values after the LOCAL key is pressed. In such case, hold the LOCAL key until [RMT] disappears.

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 IMMediate** (internal triggering)
Exporting	**:FETCh?** Imports the most recent measurement.

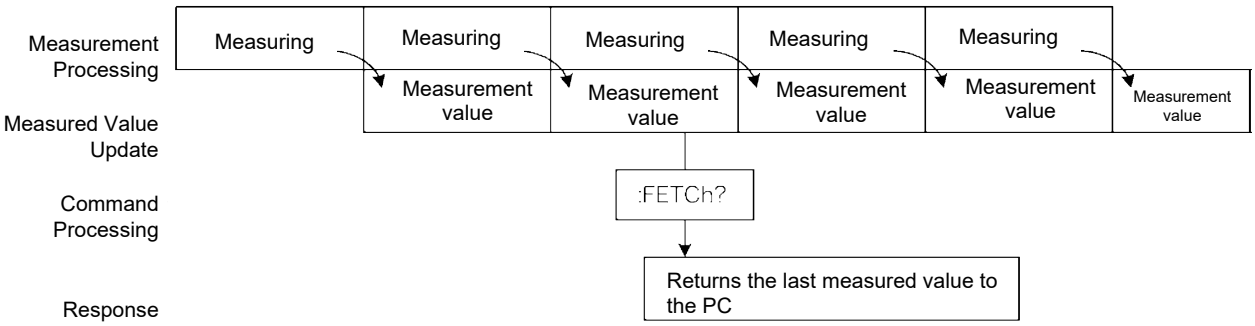
Export by Controller (PC, PLC) Triggering

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

Export by Applying TRIG Signal

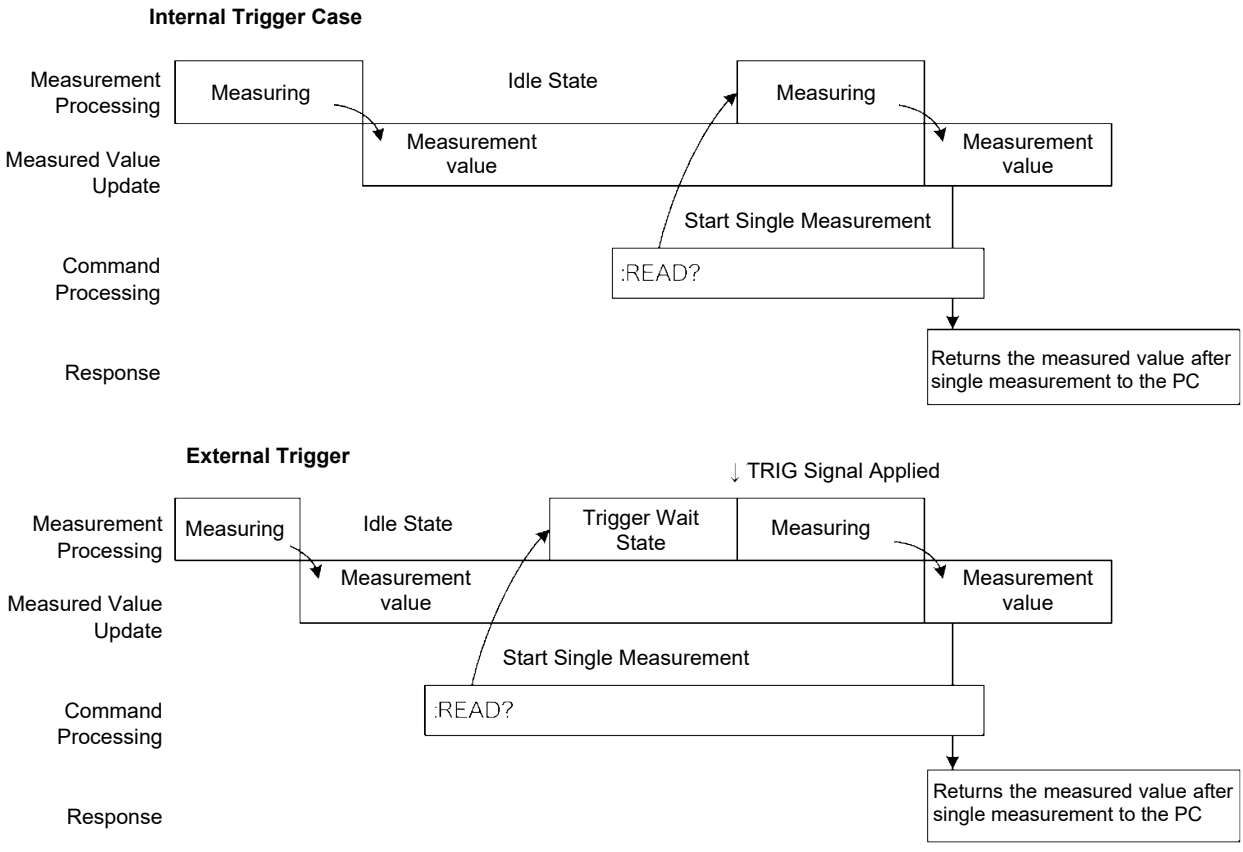
Default Setting	**:INITiate:CONTinuous OFF** (continuous measurement disable) **:TRIGger:SOURce EXT** (external triggering)
Exporting	**:READ?** When triggered by TRIG signal, a measurement is performed and the result is transferred.

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



This is the simplest 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 measurement target, wait for twice the measurement time before exporting the measured value.

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



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

5 Sample Programs

These programs can be created using Visual Basic 5.0, 6.0 or Visual Basic2013.
Visual Basic is a registered trademark of Microsoft Corporation.

Using Visual Basic 5.0 or 6.0

These sample programs are created with Microsoft Visual Basic 5.0 and 6.0.

The following are used for communication:
For RS-232C/USB communication: MSComm from Visual Basic Professional

During communications, the terminator setting is supposed to be as follows:
RS-232C/USB: CR+LF

RS-232C/USB Communications (Using Microsoft Visual Basic Professional MSComm)

Simple Measurement

Imports measured values 10 times, and saves measurements in a text file.

```
Private Sub MeasureSubRS()
   Dim recvstr As String                          'Receiving character string
   Dim i As Integer

   MSComm1.CommPort = 1                           'COM1 (Check a communication port)
   MSComm1.Settings = "9600,n,8,1"                'Set a communication port (not required with USB)
   MSComm1.PortOpen = True                        'Open a port
   Open App.Path & "¥data.csv" For Output As #1   'Open a text file for saving

   MSComm1.Output = ":FUNC RV" & vbCrLf           'Select (R,X,V,T) measurement function
   MSComm1.Output = ":MEAS:VAL 1" & vbCrLf        'Output only the measurement value
   MSComm1.Output = ":TRIG:SOUR IMM" & vbCrLf     'Select an internal triggering
   MSComm1.Output = ":INIT:CONT ON" & vbCrLf      'Continuous measurement ON
   For i = 1 To 10
      MSComm1.Output = ":FETCH?" & vbCrLf         'Send ":FETCH?" to import the most recent measurement
      recvstr = ""                                'From here on, continue receiving until an LF code occurs
      While Right(recvstr, 1) <> Chr(10)
         recvstr = recvstr + MSComm1.Input
         DoEvents
      Wend
      recvstr = Left(recvstr, Len(recvstr) - 2)   'Delete the terminator (CR+LF)
      Print #1, Str(i) & "," & recvstr            'Write to the file
   Next

   Close #1
   MSComm1.PortOpen = False
End Sub
```

Measure by PC Key

Measures and imports by key input on the PC, and saves measurements in a text file.

```
Private Sub MeasureReadSubRS()
  Dim recvstr As String                              'Receiving character string
  Dim i As Integer

  MSComm1.CommPort = 1                               'COM1 (Check a communication port)
  MSComm1.Settings = "9600,n,8,1"                    'Set a communication port (not required with USB)
  MSComm1.PortOpen = True                            'Open a port
  Open App.Path & "¥data.csv" For Output As #1       'Open a text file for saving

  MSComm1.Output = ":FUNC RV" & vbCrLf               'Select (R,X,V,T) measurement function
  MSComm1.Output = ":MEAS:VAL 1" & vbCrLf            'Output only the measurement value
  MSComm1.Output = ":TRIG:SOUR IMM" & vbCrLf         'Select internal triggering
  MSComm1.Output = ":INIT:CONT OFF" & vbCrLf         'Continuous measurement OFF
  For i = 1 To 10
    'Wait for PC key input
    'Create a key input check routine to set InputKey() = True when a key is pressed
    Do While 1
      If InputKey() = True Then Exit Do
      DoEvents
    Loop
    'After confirming key input, measure once, and read the measured value
    MSComm1.Output = ":READ?" & vbCrLf               'Send ":READ?" to measure and import the measurement
    recvstr = ""                                     'From here on, continue receiving until an LF code occurs
    While Right(recvstr, 1) <> Chr(10)
      recvstr = recvstr + MSComm1.Input
      DoEvents
    Wend
    recvstr = Left(recvstr, Len(recvstr) - 2)        'Delete the terminator (CR+LF)
    Print #1, Str(i) & "," & recvstr                 'Write to the file
  Next

  Close #1
  MSComm1.PortOpen = False
End Sub
```

External Trigger Measurement

Measures and imports based on external triggering (TRIG signal input), and saves measurements in a text file.

```
Private Sub MeasureTrigSubRS()
   Dim recvstr As String                              'Receiving character string
   Dim i As Integer

   MSComm1.CommPort = 1                               'COM1 (Check a communication port)
   MSComm1.Settings = "9600,n,8,1"                    'Set a communication port (not required with USB)
   MSComm1.PortOpen = True                            'Open a port
   Open App.Path & "¥data.csv" For Output As #1       'Open a text file for saving

   MSComm1.Output = ":FUNC RV" & vbCrLf               'Select (R,X,V,T) measurement function
   MSComm1.Output = ":MEAS:VAL 1" & vbCrLf            'Output only the measurement value
   MSComm1.Output = ":TRIG:SOUR EXT" & vbCrLf         'Select external triggering
   MSComm1.Output = ":INIT:CONT OFF" & vbCrLf         'Continuous measurement OFF
   For i = 1 To 10
      MSComm1.Output = ":READ?" & vbCrLf              'Send ":READ?" to measure and import the measurement
      recvstr = ""                                    'From here on, continue receiving until an LF code occurs
      While Right(recvstr, 1) <> Chr(10)
         recvstr = recvstr + MSComm1.Input
         DoEvents
         'After comfirming external trigger signal, measure once, and read the measurement value.
      Wend
      recvstr = Left(recvstr, Len(recvstr) - 2)       'Delete the terminator (CR+LF)
      Print #1, Str(i) & "," & recvstr                'Write to the file
   Next

   Close #1
   MSComm1.PortOpen = False
End Sub
```

■ Set Measurement Conditions

Sets up the measurement setting state.

```
'Measurement Setting Configuration
'Configures instrument settings for measurement
'Function: [R, X, V, T]
'Measurement frequency: 1000Hz
'Range: 100mΩ
'Z sampling: FAST, V sampling: FAST
'Trigger: Internal triggering
'Comparator enabled, beep upon Hi or Lo
'Resistance: Upper limit 100mΩ, lower limit 50mΩ
'Reactance: Upper limit 100mΩ, lower limit 50mΩ
'Impedance: Upper limit 100mΩ, lower limit 50mΩ
'Voltage; Upper limit 5V, lower limit 2.5V

Private Sub SettingsSubRS()
   MSComm1.CommPort = 1                                  'COM1 (Check a communication port)
   MSComm1.Settings = "9600,n,8,1"                       'Set a communication port (not required with USB)
   MSComm1.PortOpen = True                               'Open a port

   MSComm1.Output = ":FUNC RV" & vbCrLf                  'Select (R,X,V,T) function
   MSComm1.Output = ":FREQ 1000"                         'Select 1000Hz measurement frequency
   MSComm1.Output = ":RANG 1E-1" & vbCrLf                'Select 100mΩ range
   MSComm1.Output = ":SAMP:RATE Z FAST" & vbCrLf         'Select FAST sampling
   MSComm1.Output = ":SAMP:RATE V FAST" & vbCrLf
   MSComm1.Output = ":TRIG:SOUR :IMM" & vbCrLf           'Select internal triggering
   MSComm1.Output = ":INIT:CONT ON" & vbCrLf             'Continuous measurement ON
   MSComm1.Output = ":CALC:LIM:STAT ON" & vbCrLf         'From here on, comparator setting
   MSComm1.Output = ":CALC:LIM:BEEP HL" & vbCrLf
   MSComm1.Output = ":CALC:LIM:RES 1E-1,0.5E-1" & vbCrLf
   MSComm1.Output = ":CALC:LIM:REAC 1E-1,0.5E-1" & vbCrLf
   MSComm1.Output = ":CALC:LIM:IMP 1E-1,0.5E-1" & vbCrLf
   MSComm1.Output = ":CALC:LIM:VOLT 5E+0,2.5E+0" & vbCrLf

   MSComm1.PortOpen = False
End Sub
```

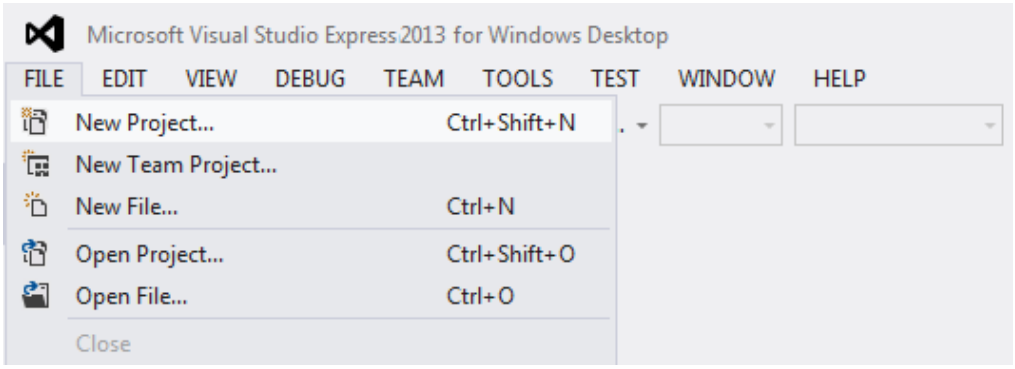
Using Visual Basic2013

This section describes an example of how to use the Windows development language, Visual Basic2013 Express Edition, to operate the BT4560 unit from a PC via an RS232C/USB interface, incorporate measurement values, and save measurement values to a file.
Visual Basic2013 is referred to as VB2013 hereafter.

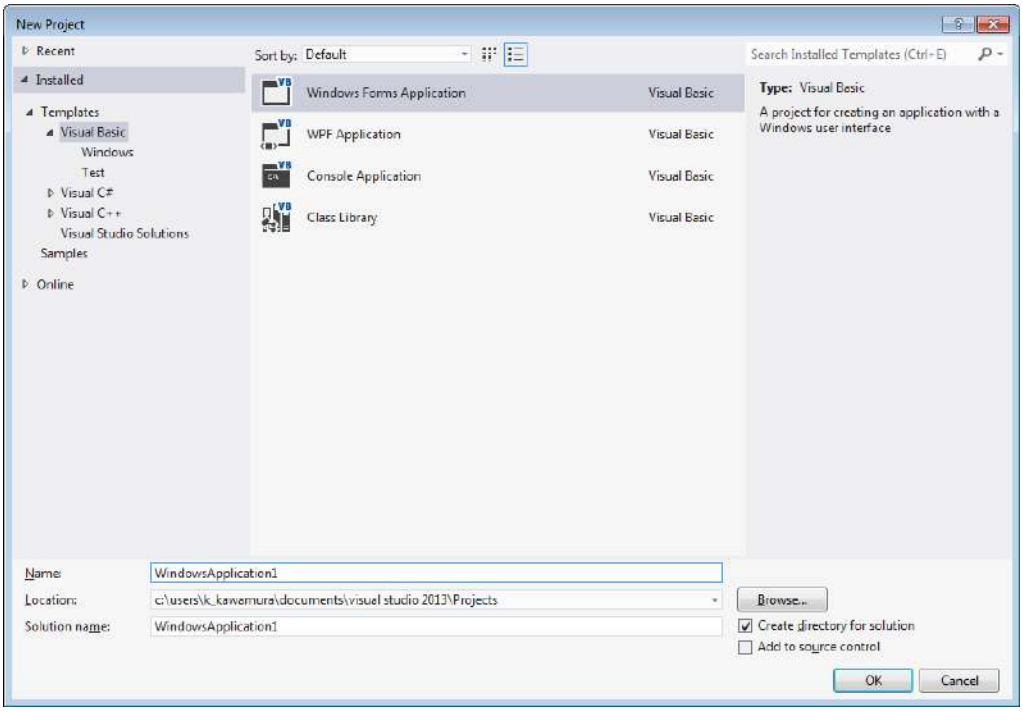
Note: The procedure may differ slightly from the one described here depending on the environment of the PC and VB2013. For a detailed explanation on how to use VB2013, refer to the instruction manual or the Help feature of VB2013.

1. Create a new project.

1. Startup VB2013.



2. Select [File] - [New Project].



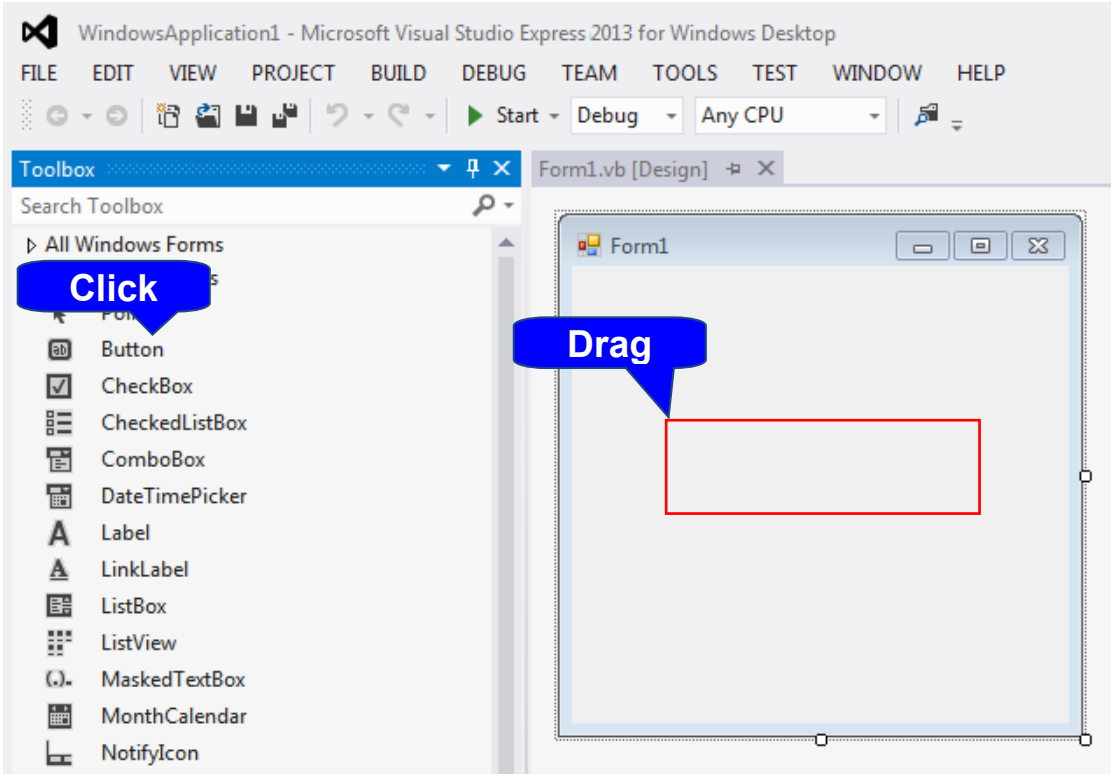
3. Select [Windows Forms Application] from the templates.

4. Click [OK].

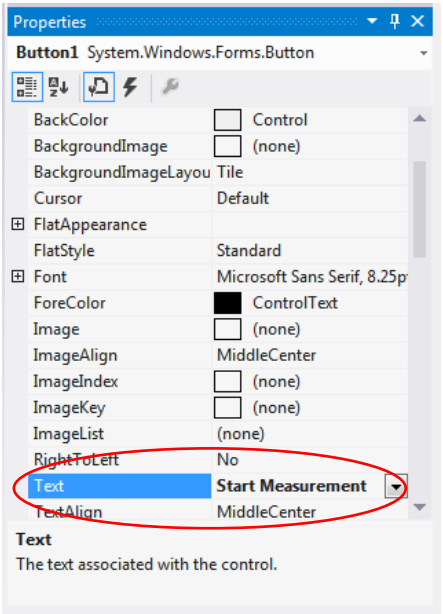
2. Place a button.

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

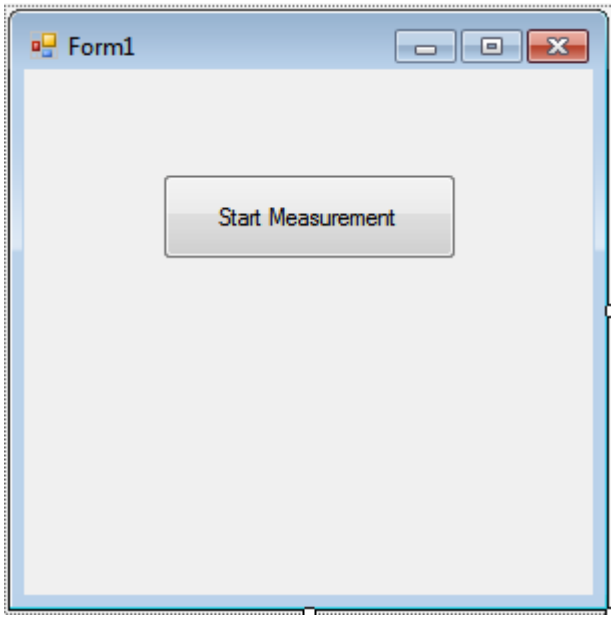
2. Drag and drop the button onto the form layout screen.



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



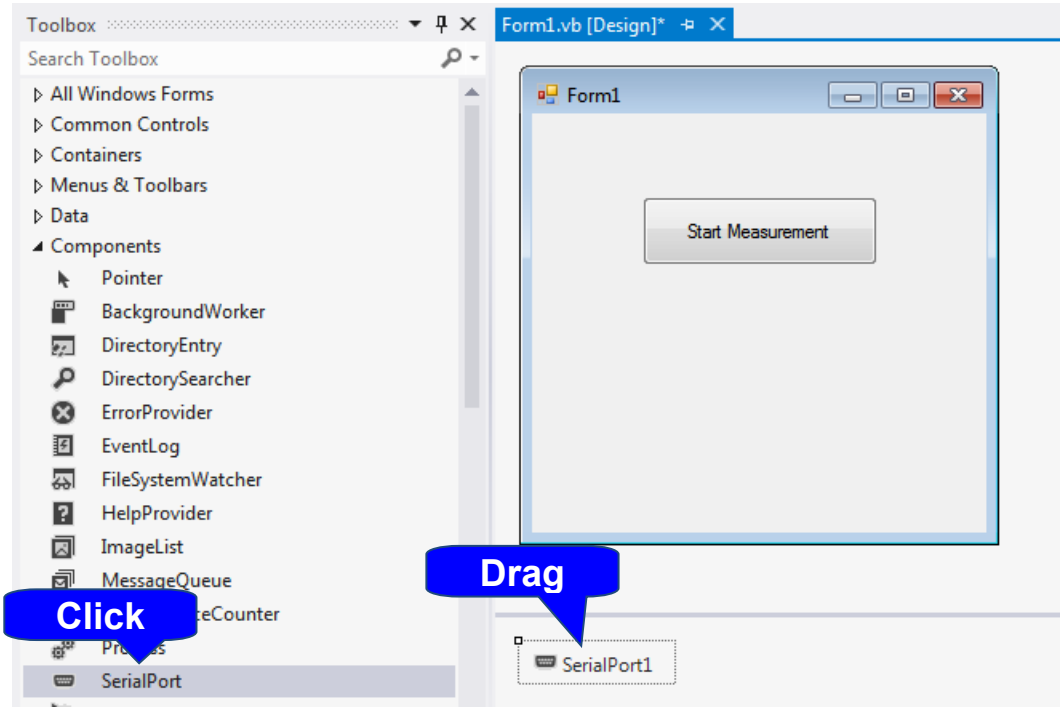
4. The [Start Measurement] is placed on the form.



3. Place a serial communication component.

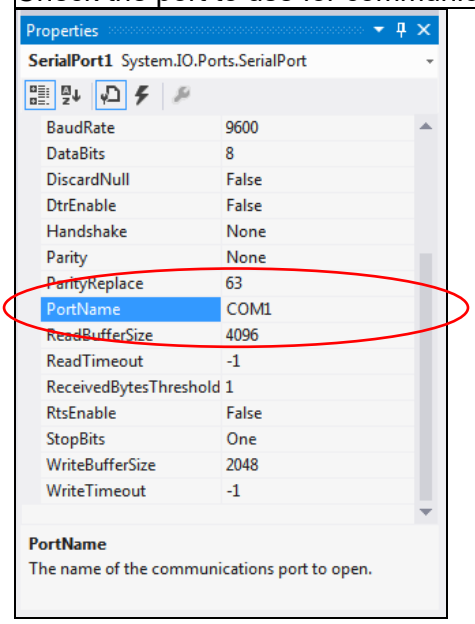
1. Click [SerialPort] from [Components] of [Toolbox].

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



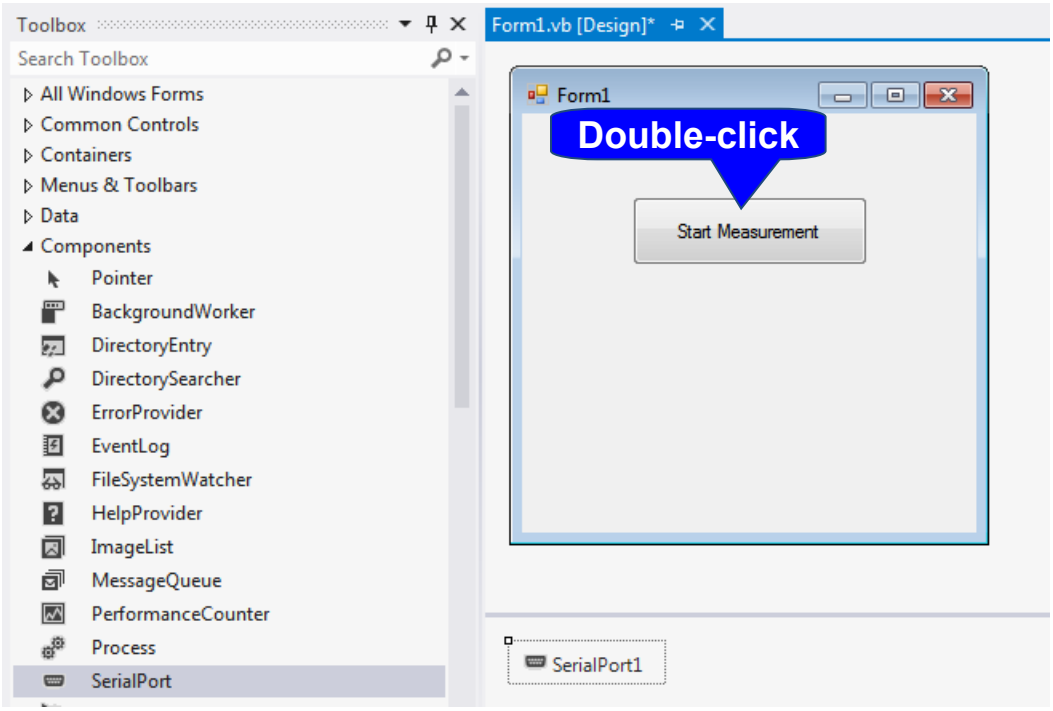
3. Change [PortName] to the port name to use for communication from the Properties window.

Check the port to use for communication beforehand.

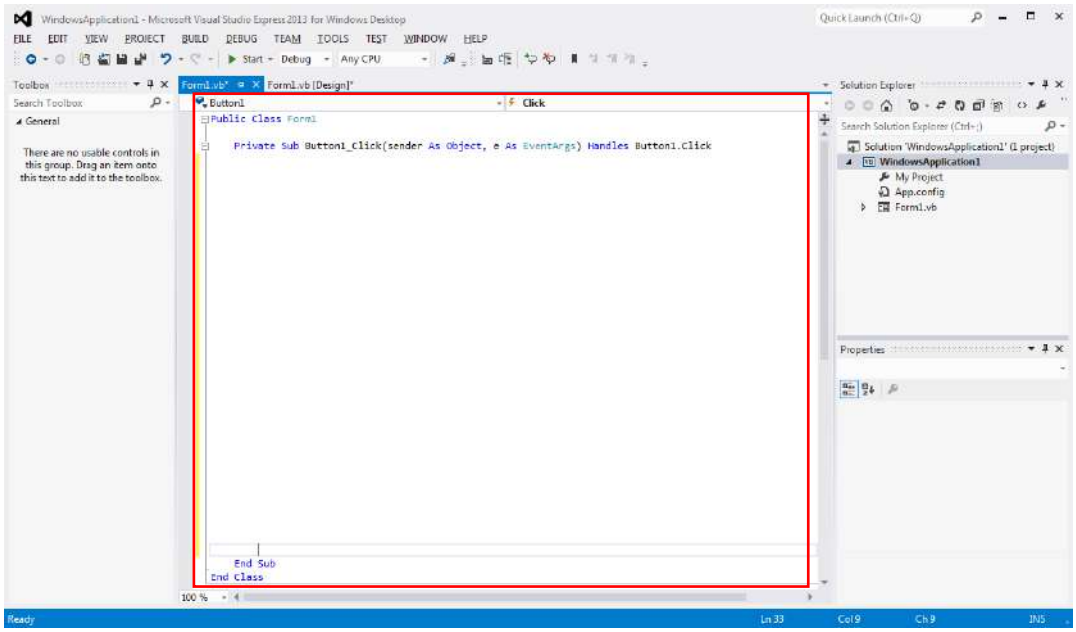


4. Describe the code.

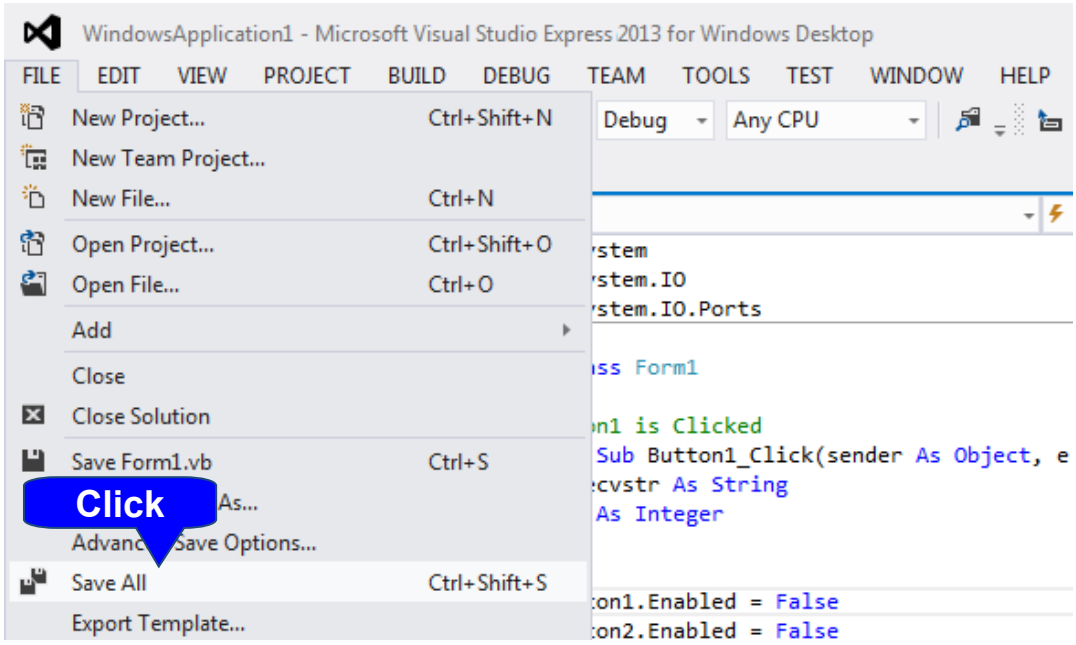
1. Double-click the placed button to display the code editor.



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 VB2013 to initiate RS-232C/USB communications, 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
Button created to close application	Button2

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

```
Imports System
Imports System.IO
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..... (a)
      Button2.Enabled = False
      SerialPort1.NewLine = vbCrLf               'Terminator setting ..... (b)
      SerialPort1.ReadTimeout = 2000             '2 seconds time out .....(c)
      SerialPort1.Open()                         'Open a port
      SendSetting(SerialPort1)                   'Instrument settings
      FileOpen(1, "data.csv", OpenMode.Output)   'Create text file to be saved..... (d)
      For i = 1 To 10
        SerialPort1.WriteLine(":FETCH?")         'Get measurement results ..... (e)
        recvstr = SerialPort1.ReadLine()         'Read measurement results
        PrintLine(1, recvstr)                    'Write to file
      Next i
    Catch ex As Exception
      MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
    FileClose(1)                                 'Close file
    SerialPort1.Close()                          'Close port
    Button1.Enabled = True
    Button2.Enabled = True

  End Sub

  'Set measurement conditions
  Private Sub SendSetting(ByVal sp As SerialPort)
    Try
      sp.WriteLine(":FUNC RV")                   '[R,X,V,T] function
      sp.WriteLine(":FREQ 1000")                 'Set frequency to 1000Hz
      sp.WriteLine(":TRIG:SOUR IMM")             '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

  'Close program when Button2 is pressed
  Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    Me.Dispose()
  End Sub

End Class
```

(a) During communication the [Begin Measurement] and [Close] buttons cannot be pressed.
(b) Sets CR + LF as the terminator indicating the end of the sending and receiving character string.
(c) Sets the reading operation time to 2 seconds.
(d) Opens the “data.csv” file. However, if a file with this name already exists, the previous “data.csv” will be deleted and a new file created.
(e) Sends the command to the instrument to return the measurement result to the computer.

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