

# M550 Impedance Calibrator

remote control manual



# Content

<b>1. Connection Setup</b>	<b>3</b>
1.1. IEEE488 bus properties	3
1.2. RS232 serial line setting	3
<b>2. Executing Commands</b>	<b>4</b>
2.1. Command Syntax	4
<b>3. Status Reporting</b>	<b>5</b>
3.1. Status Data Structure-Register Model	5
3.2. Status Data Structure Registers	6
3.3. Error messages	7
<b>4. M143 SCPI Command Reference</b>	<b>8</b>
OUTPut subsystem	8
OUTP[STAT](?) <CPD> { ON   OFF   1   0 }	8
OUTP:CORR(?) <CPD> { ON   OFF   1   0 }	8
SOURce subsystem	8
[SOUR]:MODE?	9
[SOUR]:R4W:POS(?) <DNPD>	10
[SOUR]:R4W:TYPE(?) <CPD> { RSLs   RSCS   RPLP   RPCP   ZTD   ZTR   YTD   YTR   RX   GB }	10
[SOUR]:R4P:VAL(?) <DNPD> [<DNPD>]	10
FREQ(?) <DNPD>	10
FREQ:SYNC	10
SYSTem subsystem	11
SYST:DATE(?) <DNPD>,<DNPD>,<DNPD>	11
SYST:TIME(?) <DNPD>,<DNPD>,<DNPD>	11
SYST:REM	11
SYST:RWL	11
SYST:LOC	11
IEEE488.2 Common Commands	12
/D (instrument identification)	12
Operation complete	12
Operation complete?	12
Wait-to-Continue command	12
Reset	12
Test operation	12
Status byte reading	12
Service Request Enable	12
Service Request Enable reading	12
Event Status Register reading	12
Event Status Enable setting	12
Event Status Enable reading	13
Clear status	13

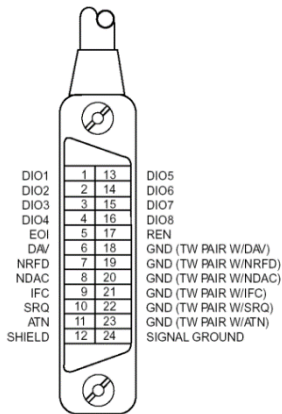
# 1. Connection Setup

The calibrator can be integrated into automated calibration systems (ATS) and generally controlled from remote computer via RS232 and GPIB (IEEE488) interface. When controlled remotely, maximum ratings of calibrator’s output signals as well as all other specifications are the same as in manual mode. Only one interface can be used for communication at any given time. To establish connection between the calibrator and computer, select preferred interface in MENU (RS232 is active by default) and follow instructions from relevant subchapter below.

## 1.1. IEEE488 bus properties

The instrument performs the following functions based on IEEE488 bus commands:

*SHI, AH1, T5, L3, RL1, DC1, SRI*



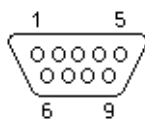
The instrument also recognizes the following general commands:

DCL	Device Clear	reset the calibrator
SDC	Selected Device Clear	reset the calibrator
EOI	End or Identify Message Terminator	close the message
GTL	Go To Local	close remote control mode
LLO	Local Lock Out	local control locked
SPD	Serial Poll Disable	close serial message status
SPE	Serial Poll Enable	release serial message status

## 1.2. RS232 serial line setting

To transfer the data using RS232 bus, 8N1 data format is used, i.e. each data word includes 8 bits, no parity and one stop bit. The communication speed can be set using the system menu. Available values: 150, 300, 600, 1200, 2400, 4800, 9600 and 19200 Bd. Software handshake (communication control) XON/XOFF can be set to control the transfer of the data through the bus.

RS-232 connector layout:



Pin	Name	Direction	Description
2	TXD	Output	Transmitter
3	RXD	Input	Receiver
5	GND	-	Ground

9-pin connector D-SUB FEMALE

Cable between the calibrator and PC (configuration 1:1)

PC	D-Sub 1	D-Sub 2	Calibrator
Receiver	2	2	Transmitter
Transmitter	3	3	Receiver
Ground	5	5	Ground

Note:

It is not possible to send or receive data over RS232 or Ethernet when the calibrator is not in remote mode.

You should use the command SYST:REM or SYST:RWL

## 2. Executing Commands

Most commands can be executed in Remote mode only. In Remote mode, device blocks manual entry interfaces (keyboard, touchscreen) to prevent unintended manipulation during remote control session. GPIB interface switches device into and out of Remote mode automatically. Other interfaces (RS232) need to use following commands:

- SYSTem:REMOte      Puts device into Remote mode and locks keyboard except Local button
- SYSTem:LOCAL      Returns device from Remote mode and unlocks keyboard

### 2.1. Command Syntax

All commands listed in this chapter are made of KEYWORDS and PARAMETERS.

One or more KEYWORDS divided by colons (:) make up name of a command. When typing a command, keywords in square brackets ( [ ] ) can be omitted as well as lowercase letters and leading colons. For example “:OUTPut:STATe ON”, “:OUTPut ON” and “OUTP 1” are considered the same.

Different PARAMETER types describe what kind of data is expected. For more details on parameter types, see table below. In command reference parameters are indicated by sharp brackets ( < > ) and separated by commas (,). Square brackets ( [ ] ) indicate non-mandatory parameters, vertical lines ( | ) indicate parameter alternatives.

Multiple commands can be executed at once if separated by semicolon (;). In any case, command (or set of commands) has to end with line feed (LF) terminator in order to be executed. With exception of GPIB interface, the device will also accept carriage return (CR) and combination of both (CRLF) as terminators. Inversely, device will always terminate its reply with CRLF or just LF in case of GPIB. The communication may look like this:

1. Controller> „RES 100;OUTP ON“+LF (ASCII code 10)
2. Device> „1.000000E+02 OHM“+LF
3. Device> „1“+LF

#### Used parameter types

Type	Description
<DNPD>	Decimal Numeric Program Data, this format is used to express decimal number with or without the exponent.
<CPD>	Character Program Data. Usually, it represents a group of alternative character parameters. E.g. { GPIB}.
<SPD>	String Program Data (quoted string). This type of parameter is similar to CPD, but allows transmission of more ISO characters.
?	A flag indicating a request for the value of the parameter specified by the command. No other parameter than the question mark can be used.
(?)	A flag indicating a request for the parameter specified by the command. This command permits a value to be set as well as requested.

### 3. Status Reporting

#### 3.1. Status Data Structure-Register Model

Device meets standard protocol according to the standard IEEE488.2. The protocol can be used for checking of error and status behavior of the device. It enables single-wire transmitting of SRQ command. The conditions on which SRQ signal (local control request) is sent can be set with commands \*STB?, \*SRE?, \*SRE, \*ESR?, \*ESE?, \*ESE a \*CLS.

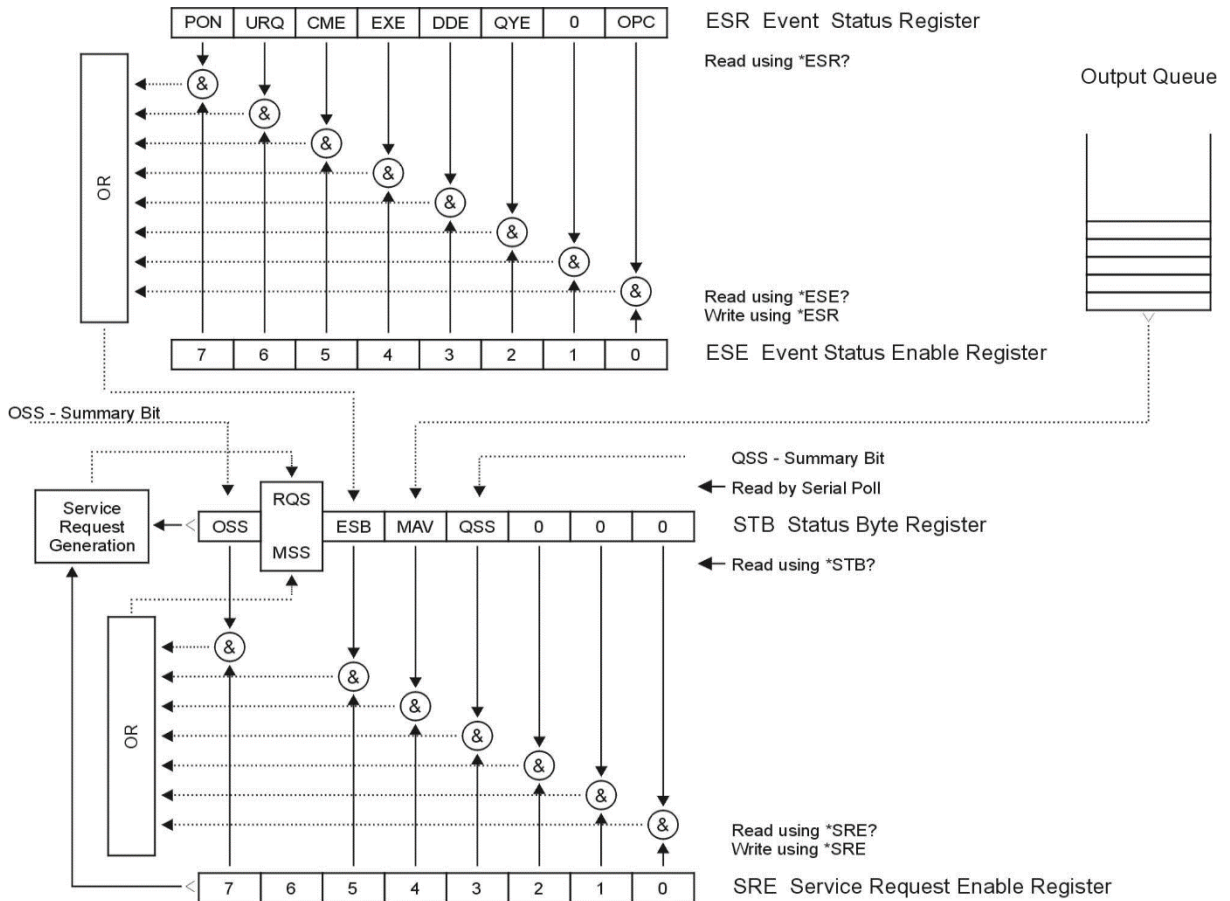


Figure 1 Status register overview

Status data structure contains following registers:

- STB – Status Byte Register
- SRE – Service Request Enable Register
- ESR – Event Status Register
- ESE – Event Status Enable Register
- Output Queue

## 3.2. Status Data Structure Registers

### STB Status Byte Register

STB is main register where information from other status registers and from output queue is collected. Value of STB register is reset after switching on the device or after sending command \*CLS. This command reset the STB register except bit MAV, which remains set if the output queue is not empty. STB register value can be read via serial message or through general query \*STB?. Bit configuration of Status Byte Register is as follows:

- RQS Request Service, bit 6. The bit is read as a part of status byte only when serial message is sent.
- MSS Master Summary Status, bit 6. The MSS bit is set to 1 whenever bits ESB or MAV are 1 and enabled (1) in the SRE. This bit can be read using the \*STB? command. His value is derived from STB and SRE status.
- ESB Event Summary Bit, bit 5. His value is derived from STB and SRE status. The ESB bit is set to 1 when one or more enabled ESR bits are set to 1.
- MAV Message Available, bit 4. The MAV bit is set to 1 whenever data is available in the M143/143i IEEE488 Output Queue (the response on query is ready).
- QSS Questionable Summary Status, bit 3. SCPI-defined. The QSS bit is set to 1 when the data in the QSR (Questionable Status Register) contains one or more enabled bits which are true.

### SRE Service Request Enable Register

The Service Request Enable Register suppresses or allows the STB bits. “0” value of a SRE bit means, that the bit does not influence value of MSS bit. Value of any unmasked STB bit results in setting of the MSS bit to the level “1”. SRE bit 6 is not influenced and its value is “0”. The SRE register value can be set via the command \*SRE followed by mask register value (0 – 191). The register can be read with the command \*SRE?. The register is automatically resets after switching the device on. The register is not reset by the command \*CLS.

### ESR Event Status Register

Every bit of the EventStatusRegister corresponds to one event. Bit is set when the event is changed and it remains set also when the event passed. The ESR is cleared when the power is turned on (except bit PON which is set), and every time it is read via command \*ESR? Or cleared with \*CLS. Bit configuration of Event Status Register is as follows:

- PON Power On, bit 7. This event bit indicates that an off-to-on transition has occurred in the device’s power supply.
- URQ User Request, bit 6. Bit is not used and it is always “0”.
- CME Command Error, bit 5. This event bit indicates that an incorrectly formed command or query has been detected by the instrument.
- EXE Execution Error, bit 4. This event bit indicates that the received command cannot be executed, owing to the device state or the command parameter being out of limits.
- DDE Device Dependent Error, bit 3. This event bit indicates that an error has occurred which is neither a Command Error, a Query Error, nor an Execution Error. A Device-specific Error is any executed device operation that did not properly complete due to some condition, such as overload.
- QYE Query Error, bit 2. The bit is set if the device is addressed as talker and output queue is empty or if control unit did not pick up response before sending next query.
- OPC Operation Complete, bit 0. This event bit is generated in response to the \*OPC command. It indicates that the device has completed all selected pending operations.

### ESE Event Status Enable Register

The Event Status Enable Register allows one or more events in the Event Status Register to be reflected in the ESB summary-message bit. This register is defined for 8 bits, each corresponding to the bits in the Event Status Register. The Event Status Enable Register is read with the common query \*ESE?. Data is returned as a binary-weighted value. The Event Status Enable Register is written to by the common command, \*ESE. Sending the \*ESE common command followed by a zero clears the ESE. The Event Status Enable Register is cleared upon power-on.

It suppresses or allows bits in ESR register. Value „0“ of a bit of ESE register suppresses influence of appropriate bit of ESR register on value of sum bit of ESB status register. Setting of any unmask bit of ESR register results in setting of ESB status register. ESE register value can be modified by command \*ESE followed by value of mask register (integer in range 0 –255). Reading of the register can be performed with command \*ESE?. The register is automatically reset after switching on. The register is not reset with \*CLS command.

### Output Queue

The Output Queue stores response messages until they are read from control unit. If there is at minimum one sign in the output queue, MAV register (message available) is set. The Output Queue is cleared upon power-on and after reading all signs from output queue.

### 3.3. Error messages

Error message	Description
Function is not available	The selected function is not available in the calibrator.
Value too low	Set value within specified limits.
Value too high	Set value within specified limits.
Turn output terminals ON	Output terminals are OFF while test level meter has been activated. Turn the output terminals ON.
Wrong password	Enter correct calibration password.
Function METER is OFF	The message will occur when test signal level meter is set to OFF in SETUP menu and F/U button is pushed to activate measuring. Select position ON in SETUP menu before activating the level meter.

## 4. M143 SCPI Command Reference

### OUTPut subsystem

This subsystem enables to configure and activate / deactivate the calibrator output.

Keyword	Parameters
OUTPut	
[:STATe](?)	<CPD> { ON   OFF   1   0 }
:CORRection (?)	<CPD> { ON   OFF   1   0 }

#### **OUTP[:STAT](?) <CPD> { ON | OFF | 1 | 0 }**

This command activates or deactivates output terminals.

- ON or 1 - activates the output
- OFF or 0 - deactivates the output

If query is sent, calibrator returns 1 if the output is active or 0 if it is inactive

Example:       OUTP ON - activates the output  
              OUTP? - the calibrator returns 1 or 0

#### **OUTP:CORR(?) <CPD> { ON | OFF | 1 | 0 }**

This command activates or deactivates correction of residual parameters.

- ON or 1 - activates correction
- OFF or 0 - deactivates correction

If query is sent, calibrator returns 1 if correction is active or 0 if it is inactive

Example:       OUTP:CORR ON - activates correction  
              OUTP:CORR? - the calibrator returns 1 or 0

### SOURce subsystem

This subsystem allows controlling the individual functions of the calibrator.

Keyword	Parameters
[SOURce]	
:MODE?	{ R4W   R2W   R4P   C4W   C2W   C4P   L4W   L2W   L4P   SHOR   OPEN   EXT   SIM }
:R4W	
:POSition(?)	<DNPd>
:TYPe(?)	<CPD> { RSLs   RSCS   RPLP   RPCP   ZTD   ZTR   YTD   YTR   RX   GB }
[:VALUe] (?)	<DNPd> [,<DNPd>]
:R2W	
:POSition(?)	<DNPd>
:TYPe(?)	<CPD> { RSLs   RSCS   RPLP   RPCP   ZTD   ZTR   YTD   YTR   RX   GB }
[:VALUe] (?)	<DNPd> [,<DNPd>]
:R4P	
:POSition(?)	<DNPd>
:TYPe(?)	<CPD> { RSLs   RSCS   RPLP   RPCP   ZTD   ZTR   YTD   YTR   RX   GB }
[:VALUe] (?)	<DNPd> [,<DNPd>]
:C4W	
:POSition(?)	<DNPd>
:TYPe(?)	<CPD> { CSD   CSRS   CPD   CPGP   CPRP }



	ZTD   ZTR   YTD   YTR }
[ :VALUe ] (?)	<DNPd> [, <DNPd>]
:C2W	
:POSition(?)	<DNPd>
:TYPE(?)	<CPD> { CSD   CSRS   CPD   CPGP   CPRP   ZTD   ZTR   YTD   YTR }
[ :VALUe ] (?)	<DNPd> [, <DNPd>]
:C4P	
:POSition(?)	<DNPd>
:TYPE(?)	<CPD> { CSD   CSRS   CPD   CPGP   CPRP   ZTD   ZTR   YTD   YTR }
[ :VALUe ] (?)	<DNPd> [, <DNPd>]
:L4P	
:POSition(?)	<DNPd>
:TYPE(?)	<CPD> { LSQ   LSRS   ZTD   ZTR   YTD   YTR }
[ :VALUe ] (?)	<DNPd> [, <DNPd>]
:SH4W	
:SH2W	
:SH4P	
:OP4W	
:OP2W	
:OP4P	
:EXTernal	
:FREQ(?)	<DNPd>
:SYNChronize	

### [SOUR]:MODE?

This command returns selected mode of the calibrator { R4W | R2W | R4P | C4W | C2W | C4P | | L4P | SH4W | SH2W | SH4P | OP4W | OP2W | OP4P | EXT }.

- R4W       –     Resistance 4W mode
- R2W       –     Resistance 2W mode
- R4P       –     Resistance 4P mode
- C4W       –     Capacitance 4W mode
- C2W       –     Capacitance 2W mode
- C4P       –     Capacitance 4P mode
- L4P       –     Inductance 4P mode
- SH4W      –     Short 4W mode
- SH2W      –     Short 2W mode
- SH4P      –     Short 4P mode
- OP4W      –     Open 4W mode
- OP2W      –     Open 2W mode
- OP4P      –     Open 4P mode
- EXT       –     External mode

Example:            MODE?

**[SOUR]:R4W:POS(?) <DNPD>**

This command selects one of the 10 available resistances in 4W configuration. Index of resistance is from 1 (100mΩ) to 10 (100MΩ).

<DNPD>

Parameter represents index of required resistance. Acceptable range is 1 ... 10.

If query is sent, calibrator returns the index of selected resistance.

Example:

R4W:POS? Calibrator return "4"

Example:

Resistance index 4 (100Ω) in 4W configuration:

R4W:POS 4

Note:

Command set 100Ω in Resistance 4W mode.

**[SOUR]:R4W:TYPE(?) <CPD> { RSLs | RSCS | RPLP | RPCP | ZTD | ZTR | YTD | YTR | RX | GB }**

This command selects type of displayed parameters.

If query is sent, calibrator returns actual Resistance 4W parameter type.

Example:

Select Rs-Ls resistance 4W parameter type:

R4W:TYPE RSLs

Note:

Command switches the calibrator in Resistance 4W mode.

**[SOUR]:R4P:VAL(?) <DNPD> [,<DNPD>]**

This command sets the resistance value in 4W configuration.

<DNPD>

Parameter represents the value of the resistance expressed in units of selected parameter type.

Calibrator selects the nearest resistance value.

If query is sent, calibrator returns the set value of resistance using standard exponential format.

Example: 105.547mΩ is returned as 1.055470e-001.

Example:

Resistance 100mΩ:

R4P:VAL?

calibrator return "+1.00000e-001,+3.40000e-009:

**FREQ(?) <DNPD>**

This command sets the frequency. Displayed parameters are automatically recalculated according to this frequency.

<DNPD>

Parameter represents the value of frequency expressed in Hz.

If query is sent, calibrator returns the value of frequency using standard exponential format. Example:

50Hz is returned as 5.00000e+001.

Example:

Frequency 60Hz:

FREQ 60

**FREQ:SYNC**

This command sets the frequency according to the measuring frequency of connected RLC meter.

Displayed parameters are automatically recalculated according to this frequency.

Example:

FREQ:SYNC synchronized the frequency with connected RLC meter

## **SYSTem subsystem**

The subsystem enables to control various functions from the MENU.

SYSTem

:DATE(?) <DNPD>,<DNPD>,<DNPD>  
:TIME(?) <DNPD>,<DNPD>,<DNPD>  
:ERRor?  
:REMote  
:RWLock  
:LOCal

### **SYST:DATE(?) <DNPD>,<DNPD>,<DNPD>**

This command sets system date of the calibrator.

<DNPD>,<DNPD>,<DNPD>

Represents date in format YYYY, MM, DD.

If query is sent, calibrator returns current value of system date in format YYYY,MM,DD.

where YYYY = year (2000..2099)

MM = month (01..12)

DD = day (01..31)

### **SYST:TIME(?) <DNPD>,<DNPD>,<DNPD>**

This command sets time of the calibrator.

<DNPD>,<DNPD>,<DNPD>

Represents time in format HH,MM,SS.

If query is sent, calibrator returns current value of system time in format HH,MM,SS.

where HH hour (00..23)

MM minute (00..59)

SS second (00..59)

### **SYST:REM**

This command places the calibrator in the remote mode for RS232 or Ethernet operation. All keys on the front panel, except the LOCAL key, are disabled.

*Note:*

*It is not possible to send or receive data over RS232 or Ethernet when the calibrator is not in remote mode.*

### **SYST:RWL**

This command places the calibrator in the remote mode for RS232 or Ethernet operation. All keys on the front panel, including the LOCAL key, are disabled.

### **SYST:LOC**

This command returns the calibrator in the local mode. This command is for RS232 and Ethernet interfaces.

## ***IEEE488.2 Common Commands***

### ***I/D (instrument identification)***

#### **\*IDN?**

This command returns the identification of the manufacturer, model, serial number and firmware revision.

The reply is formatted as follows:

MEATEST,M550,100002,1.22

### ***Operation complete***

#### **\*OPC**

This command sets the OPC bit in the ESR (Event Status Register) when all pending operations are complete.

### ***Operation complete?***

#### **\*OPC?**

This command returns “1” to the output queue after all pending operations are complete.

### ***Wait-to-Continue command***

#### **\*WAI**

Prevents the instrument from executing any further commands or queries until all previous remote commands have been executed.

### ***Reset***

#### **\*RST**

This command resets the calibrator to its initial status.

### ***Test operation***

#### **\*TST?**

This command launches an internal self-test. Return the self-test result (“0” for pass or “1” for fail).

### ***Status byte reading***

#### **\*STB?**

This query returns number in range 0 to 255 with information about content of register STB, which carries the MSS bit status.

### ***Service Request Enable***

#### **\*SRE <value>**

This command sets condition of the Service Request Enable register. Since bit 6 is not used, the maximum value is 191.

### ***Service Request Enable reading***

#### **\*SRE?**

This query returns the Service Request Enable Register number.

### ***Event Status Register reading***

#### **\*ESR?**

This query returns the contents of the Event Status Register and clears the register.

### ***Event Status Enable setting***

#### **\*ESE <value>**

This command programs the Event Status Enable register bits. Parameter “value” is number in range 0 – 255.

***Event Status Enable reading***

**\*ESE?**

This query returns the Event Status Enable register.

***Clear status***

**\*CLS**

This command clears the Event Status Register and the Status Byte Register except the MAV bit and output queue. Output line is not reset.