

# BK PRECISION

## 5335B

Power Meter



Programming Manual

1	SCPI Commands	6
1.1	Example query command – Read the serial number	6
1.2	Example value set command	6
1.3	Example value query command	6
2	IEEE-488 Commands	7
2.1	* Query Event Register	7
2.2	*ESE	7
2.3	*ESR?	7
2.4	*IDN?	8
2.5	*OPC	8
2.6	*RST	8
2.7	*SRE	8
2.8	*STB?	9
2.9	*WAI	9
3	Current Measurement Commands	10
3.1	AC level <code>fetch:curr:ac?</code> , <code>meas:curr:ac?</code>	10
3.2	Crest Factor <code>fetch:curr:cfac?</code> , <code>meas:curr:cfac?</code>	10
3.3	DC Level <code>fetch:curr:dc?</code> , <code>meas:curr:dc?</code>	10
3.4	RMS <code>fetch:curr:rms?</code> , <code>meas:curr:rms?</code>	10
3.5	Rectified Mean <code>fetch:curr:rmn?</code> , <code>meas:curr:rmn?</code>	11
3.6	Mean <code>fetch:curr:mn?</code> , <code>meas:curr:mn?</code>	11
3.7	Inrush <code>fetch:curr:inr?</code> , <code>meas:curr:inr?</code>	11
3.8	Maximum Peak <code>fetch:curr:maxp?</code> , <code>meas:curr:maxp?</code>	11
3.9	Negative Peak (Minimum Peak) <code>fetch:curr:minp?</code> , <code>meas:curr:minp?</code>	11
3.10	Positive Peak <code>fetch:curr:ppe?</code> , <code>meas:curr:ppe?</code>	11
4	Voltage Measurement Commands	12
4.1	AC - <code>fetch:volt:ac?</code> , <code>meas:volt:ac?</code>	12
4.2	DC - <code>fetch:volt:dc?</code> , <code>meas:volt:dc?</code>	12
4.3	RMS - <code>fetch:volt:rms?</code> , <code>meas:volt:rms?</code>	12
4.4	Rectified Mean - <code>fetch:volt:rmn?</code> , <code>meas:volt:rmn?</code>	12
4.5	Mean - <code>fetch:volt:mn?</code> , <code>meas:volt:mn?</code>	12
4.6	Crest Factor - <code>fetch:volt:cfac?</code> , <code>meas:volt:cfac?</code>	13
4.7	Maximum Peak - <code>fetch:volt:maxp?</code> , <code>meas:volt:maxp?</code>	13
4.8	Negative peak (minimum peak) - <code>fetch:volt:minp?</code> , <code>meas:volt:minp?</code>	13
4.9	Positive Peak - <code>fetch:volt:ppe?</code> , <code>meas:volt:ppe?</code>	13
5	Power Measurement Commands	14
5.1	Active - <code>fetch:pow:act?</code> , <code>meas:pow:act?</code>	14
5.2	Apparent - <code>fetch:pow:app?</code> , <code>meas:pow:app?</code>	14
5.3	Reactive - <code>fetch:pow:reac?</code> , <code>meas:pow:reac?</code>	14
5.4	Pfactor - <code>fetch:pow:pfac?</code> , <code>meas:pow:pfac?</code>	14
5.5	Phase - <code>fetch:pow:phas?</code> , <code>meas:pow:phas?</code>	14
6	Frequency Measurement Commands	15
6.1	Voltage - <code>fetch:freq:volt?</code> , <code>meas:freq:volt?</code>	15
6.2	Current - <code>fetch:freq:curr?</code> , <code>meas:freq:curr?</code>	15
6.3	Synchronous Source - <code>fetch:freq:sso?</code> , <code>meas:freq:sso?</code>	15

7	Energy Commands	16
7.1	Total value - fetc:ener?, meas:ener?	16
7.2	Positive - fetc:ener:pos?, meas:ener:pos?	16
7.3	Negative - fetc:ener:neg?, meas:ener:neg?	16
7.4	Accumulated Charge - fetc:ener:char?, meas:ener:char?	16
7.5	Positive - fetc:ener:char:pos?, meas:ener:char:pos?	16
7.6	Negative - fetc:ener:char:neg?, meas:ener:char:neg?	17
7.7	Integration Time - fetc:ener:time?, meas:ener:time?	17
7.8	Average - fetc:ener:aver?, meas:ener:aver?	17
8	Current Harmonics	18
8.1	Current Harmonic Amplitude - fetc:harm:curr:ampl?, meas:harm:curr:ampl?	18
8.2	Fundamental Harmonic - fetc:harm:curr:fund?, meas:harm:curr:fund?	18
8.3	Total Harmonic Current - fetc:harm:curr:thar?, meas:harm:curr:thar?	18
8.4	Total Harmonic Distortion (THD) - fetc:harm:curr:thd?, meas:harm:curr:thd?	18
9	Voltage Harmonics	19
9.1	Total Harmonic Distortion Voltage Amplitude - fetc:harm:volt:ampl?, meas:harm:volt:ampl?	19
9.2	Fundamental Harmonic Amplitude - fetc:harm:volt:fund?, meas:harm:volt:fund?	19
9.3	Total Harmonic Voltage - fetc:harm:volt:thar?, meas:harm:volt:thar?	19
9.4	Total Harmonic Distortion (THD) Voltage - fetc:harm:volt:thd?, meas:harm:volt:thd?	19
10	Power Harmonics	20
10.1	Amplitude - fetc:harm:pow:ampl?, meas:harm:pow:ampl?	20
10.2	Fundamental - fetc:harm:pow:fund?, meas:harm:pow:fund?	20
10.3	Total - fetc:harm:pow:thar?, meas:harm:pow:thar?	20
10.4	Total Distortion - fetc:harm:pow:thd?, meas:harm:pow:thd?	20
10.5	Apparent - fetc:harm:pow:app?, meas:harm:pow:app?	20
10.6	Reactive - fetc:harm:pow:reac?, meas:harm:pow:reac?	20
10.7	Power Factor - fetc:harm:pow:pfac?, meas:harm:pow:pfac?	21
10.8	K-th order voltage harmonic - fetc:harm:pow:phase:uu?, meas:harm:pow:phase:uu?	21
10.9	K-th order harmonic current and voltage - fetc:harm:pow:phase:ui?, meas:harm:pow:phase:ui?	21
10.10	K-th order current harmonic - fetc:harm:pow:phase:ii?, meas:harm:pow:phase:ii?	21
11	Harmonics Etc...	22
11.1	Harmonic orders - harm:order	22
11.2	Measurement source for PLL - harm:pll	22
11.3	Harmonic calculation formula - harm:thd	22
11.4	Harmonic set - harm:seq	22
12	Measurement Setup	23
12.1	Average enable/disable	23
12.2	Averaging time	23
12.3	Average Type	23
12.4	Samples per average	23
12.5	Current Auto Range enable/disable	24
12.6	Voltage Auto Range enable/disable	24
12.7	Current Range	24
12.8	Voltage Range	24
12.9	External Sensor 1 enable/disable	24
12.10	External Sensor 2 enable/disable	25
12.11	Scaling factor: External Sensor 1	25
12.12	Scaling factor: External Sensor 2	25
12.13	External sensor current range	25

13	Trigger Commands	26
13.1	Trigger Abort	26
13.2	Continuous trigger enable/disable	26
13.3	Initiate Trigger	26
13.4	Immediate Trigger	26
13.5	Source	27
13.6	Edge (slope)	27
13.7	Voltage Level	27
13.8	Current Level	27
14	Oscilloscope	28
14.1	Trigger source	28
14.2	Trigger edge (slope)	28
14.3	Trigger mode	28
14.4	Trigger delay	28
14.5	Time per division (Horizontal Scale)	29
14.6	Run (Start Capture)	29
14.7	Stop (Stop Capture)	29
14.8	Single capture	29
14.9	Retrieve waveform voltage data	29
14.10	Retrieve waveform current data	30
14.11	Real time data	30
14.12	Trigger status	30
15	Input Commands	31
15.1	Synchronous source	31
15.2	Crest factor setting	31
15.3	Filter enable/disable	31
15.4	Line filter enable/disable	31
15.5	Update rate	32
15.6	Automatic integral calculation enable/disable	32
15.7	Integration current mode	32
15.8	Integration power type	32
15.9	Inrush measurement enable/disable	32
15.10	Inrush current trigger level	33
15.11	Inrush measurement trigger delay	33
15.12	Inrush measurement time	33
16	Calculation setup	34
16.1	Integration state	34
16.2	Integration reset (clear)	34
16.3	Automatic integration clearing enable/disable	34
16.4	Integration start source	34
16.5	Start integration	35
16.6	Integration start date, time	35
16.7	Integration stop signal source	35
16.8	Stop integration	35
16.9	Integration stop date, time	35
16.10	Integration duration	36
16.11	Integration status/condition	36
16.12	Max value hold enable/disable	36
16.13	Clear measurment	36
16.14	Harmonic calculation enable/disable	36
16.15	Oscilloscope function enable/disable	37
16.16	Hold mode enable/disable	37

17	System Commands	38
17.1	Beep test	38
17.2	Beep enable/disable	38
17.3	System date	38
17.4	System time	38
17.5	Last key pressed	39
17.6	Retrieve error	39
17.7	Language	39
17.8	Set to local control	39
17.9	Set to remote mode	39
17.10	Error clear	40
17.11	Remote lock	40
17.12	Version	40
18	Calibration Commands	41
18.1	Zero	41
18.2	Unlock	41
18.3	Initialize	41
18.4	Factory Save	42
18.5	Save	42
18.6	Current calibration point	42
18.7	Current reading	42
18.8	External current point	43
18.9	External current reading	43
18.10	Voltage point	43
18.11	Voltage level reading	43

# SCPI Commands

General syntax for SCPI commands or query is a ``:` (colon) separated string with either a ``?` or an argument following the command string separated by a SPACE character. Commands are terminated by the linefeed character (0x10, ``

n``). Arguments are listed using `` <ARG | ARG | ...> `` in the following descriptions. The ``ARG`` will be from the table.

Symbol	Format
<NR1>	Number with an implicit decimal point at the end. Ex. 100
<NR2>	Number with an explicit decimal point. Ex. 100.5
<Boolean>	Boolean value. Ex. 0   OFF   1   ON
<STR>	String value. Ex. ON, MEASurement

Most commands have a long and short format, the capitalized portion of the name is the short form, and the complete name is the long form. For example, the FETch can be sent as either ``fet`` or fetch. Short and long forms may be mixed in a command string. An optional portion of a command is shown within ``[ ]``.

## 1.1 Example query command – Read the serial number

---

```
XX.XX SYStem
XX.XX.YY SERial?
```

This command is in the SYStem section of the SCPI command set. The command is a ``query`` command and will return a string containing the serial number. As a ``query`` it must end with a ``?``.

Command string: sys:ser?\n OR system:serial?\n OR sys:serial?\n etc...

## 1.2 Example value set command

---

```
XX.XX SYStem
XX.XX.YY SERial?
```

## 1.3 Example value query command

---

```
XX.XX SYStem
XX.XX.YY SERial?
```

# IEEE-488 Commands

2.1	* Query Event Register	7
2.2	*ESE	7
2.3	*ESR?	7
2.4	*IDN?	8
2.5	*OPC	8
2.6	*RST	8
2.7	*SRE	8
2.8	*STB?	9
2.9	*WAI	9

## 2.1 \* Query Event Register

<b>Description</b>	* Status Byte Register
<b>Command Syntax</b>	*CLS
<b>Parameters</b>	None

## 2.2 \*ESE

<b>Description</b>	Edit the value of standard event enable register.It defines the specified bits from standard event register that will cause the value of ESB bit in status byte register to be 1.
<b>Command Syntax</b>	*ESE <NR1>
<b>Parameters</b>	0?255
<b>Power-On Value</b>	Refer to *PSC command
<b>Example</b>	*ESE 128
<b>Query Syntax</b>	*ESE?
<b>Returned Value</b>	<NR1>
<b>Related Commands</b>	*ESR? *PSC *STB?

## 2.3 \*ESR?

<b>Description</b>	Read the value of standard event registers.And values will be cleared to zero after executing this command.The bit definition of standard event register and standard event enable register are the same.
<b>Query Syntax</b>	*ESR?
<b>Parameters</b>	None
<b>Returned Value</b>	<NR1>
<b>Related Commands</b>	*CLS *ESE *ESE? *OPC

## 2.4 \*IDN?

query related information of power meter.

**Query Syntax** \*IDN?

**Parameters** None

**Returned Value** <AARD>

**Example**

## 2.5 \*OPC

**Description** After all other commands executed before \*OPC command, the OPC bit in standard event register will be set to 1. Sending query standard event register command will return value 1 into the output buffer.

**Command Syntax** \*OPC

**Parameters** None

**Query Syntax** \*OPC?

**Returned Value** <NR1>

## 2.6 \*RST

**Description** This command reset the device to factory default setup.

**Command Syntax** \*RST

**Parameters** None

## 2.7 \*SRE

**Description** This command sets the condition of the Status Request Enable Register. After executing this command, power meter will return back a decimal value which is a binary weighted sum of all bits from the enable register.

**Command Syntax** \*SRE <NRf>

**Parameters** 0?255

**Power-on Value** Refer to \*PSC command

**Example** \*SRE 128

**Query Syntax** \*SRE?

**Returned Value** <NR1>

**Related Commands** \*ESE \*ESR? \*PSC \*STB?



## 2.8 \*STB?

<b>Description</b>	This command can query the Status Byte register. After executing this command, the bit6 of status byte register will be reset to 0.
<b>Query Syntax</b>	*STB?
<b>Parameters</b>	None
<b>Returned Value</b>	<NR1>
<b>Related Commands</b>	*CLS *ESE *ESR

## 2.9 \*WAI

<b>Description</b>	This command instructs the power meter not to process any further commands until all pending operations are completed. Pending operations are as defined under the *OPC command. *WAI can be aborted only by sending the power meter a device clear command.
<b>Command Syntax</b>	*WAI
<b>Parameters</b>	None
<b>Related Commands</b>	*OPC

[JL1]It's either POSitive, NEGative, or ANY [JL2]Returns VOLTAGE, CURRENT, oR EXTERNAL [JL3]Same as HOLD [JL4]Returns 1,<harmonic #> [JL5]Accepts 1 or 0, returns 1 or 0 [JL6]Returns in this format: hhhh:mm:ss. If nothing is written, each character will be replaced with a space. For example, /s/s/s/s:/s1:/s2 for 0000:1:2

# Current Measurement Commands

3.1	AC level <code>fetch:curr:ac?</code> , <code>meas:curr:ac?</code>	10
3.2	Crest Factor <code>fetch:curr:cfac?</code> , <code>meas:curr:cfac?</code>	10
3.3	DC Level <code>fetch:curr:dc?</code> , <code>meas:curr:dc?</code>	10
3.4	RMS <code>fetch:curr:rms?</code> , <code>meas:curr:rms?</code>	10
3.5	Rectified Mean <code>fetch:curr:rmn?</code> , <code>meas:curr:rmn?</code>	11
3.6	Mean <code>fetch:curr:mn?</code> , <code>meas:curr:mn?</code>	11
3.7	Inrush <code>fetch:curr:inr?</code> , <code>meas:curr:inr?</code>	11
3.8	Maximum Peak <code>fetch:curr:maxp?</code> , <code>meas:curr:maxp?</code>	11
3.9	Negative Peak (Minimum Peak) <code>fetch:curr:minp?</code> , <code>meas:curr:minp?</code>	11
3.10	Positive Peak <code>fetch:curr:ppe?</code> , <code>meas:curr:ppe?</code>	11

Each of the following commands is either a “Fetch” or a “Measure” command. A “Fetch” command retrieves the last measured value, and a “Measure” command initiates a new measurement and returns the resulting value.

## 3.1 AC level `fetch:curr:ac?`, `meas:curr:ac?`

---

<b>Description</b>	Read the current
<b>Command Syntax</b>	<code>FETCh[:SCALar]:CURRent:AC?</code> <code>MEASure[:SCALar]:CURRent:AC?</code>
<b>Returned Value</b>	<NRF>
<b>Example</b>	Command: <code>fetch:curr:ac?</code> Response: <code>something amazing</code>

## 3.2 Crest Factor `fetch:curr:cfac?`, `meas:curr:cfac?`

---

<b>Description</b>	Read the current crest factor
<b>Command Syntax</b>	<code>FETCh[:SCALar]:CURRent:CFACTOR?</code> <code>MEASure[:SCALar]:CURRent:CFACTOR?</code>
<b>Returned Value</b>	<NRF>

## 3.3 DC Level `fetch:curr:dc?`, `meas:curr:dc?`

---

<b>Description</b>	Read the average current
<b>Command Syntax</b>	<code>FETCh:[SCALar]:CURRent:DC?</code> <code>MEASure:[SCALar]:CURRent:DC?</code>
<b>Returned Value</b>	<NRF>

## 3.4 RMS `fetch:curr:rms?`, `meas:curr:rms?`

---

<b>Description</b>	Read the current effective value
--------------------	----------------------------------

**Command Syntax** FETCh:[SCALar]:CURRent:RMS?  
MEASure:[SCALar]:CURRent:RMS?

**Returned Value** <NRF>

### 3.5 Rectified Mean `fetch:curr:rmn?, meas:curr:rmn?`

---

**Description** Read the current rectified mean value(RMN)

**Command Syntax** FETCh:[SCALar]:CURRent:RMN?  
MEASure:[SCALar]:CURRent:RMN?

**Returned Value** <NRF>

### 3.6 Mean `fetch:curr:mn?, meas:curr:mn?`

---

**Description** Read the MN(Calibration to the average rectified value of the effective value) of current.

**Command Syntax** FETCh:[SCALar]:CURRent:MN?  
MEASure:[SCALar]:CURRent:MN?

**Returned Value** <NRF>

### 3.7 Inrush `fetch:curr:inr?, meas:curr:inr?`

---

**Description** Read the inrush current.

**Command Syntax** FETCh[:SCALar]:CURRent:INRush?  
MEASure[:SCALar]:CURRent:INRush?

**Returned Value** <NRF>

### 3.8 Maximum Peak `fetch:curr:maxp?, meas:curr:maxp?`

---

**Description** Read the positive peak current

**Command Syntax** FETCh[:SCALar]:CURRent:MAXPk?  
MEASure[:SCALar]:CURRent:MAXPk?

**Returned Value** <NRF>

### 3.9 Negative Peak (Minimum Peak) `fetch:curr:minp?, meas:curr:minp?`

---

**Description** Read the negative peak current

**Command Syntax** FETCh[:SCALar]:CURRent:MINPk?  
MEASure[:SCALar]:CURRent:MINPk?

**Returned Value** <NRF>

### 3.10 Positive Peak `fetch:curr:ppe?, meas:curr:ppe?`

---

**Description** Read the peak-peak current

**Command Syntax** FETCh[:SCALar]:CURRent:PPEak?  
MEASure[:SCALar]:CURRent:PPEak?

**Returned Value** <NRF>

# Voltage Measurement Commands

4.1	AC - <code>fetch:volt:ac?</code> , <code>meas:volt:ac?</code>	12
4.2	DC - <code>fetch:volt:dc?</code> , <code>meas:volt:dc?</code>	12
4.3	RMS - <code>fetc:volt:rms?</code> , <code>meas:volt:rms?</code>	12
4.4	Rectified Mean - <code>fetc:volt:rmn?</code> , <code>meas:volt:rmn?</code>	12
4.5	Mean - <code>fetc:volt:mn?</code> , <code>meas:volt:mn?</code>	12
4.6	Crest Factor - <code>fetc:volt:cfac?</code> , <code>meas:volt:cfac?</code>	13
4.7	Maximum Peak - <code>fetc:volt:maxp?</code> , <code>meas:volt:maxp?</code>	13
4.8	Negative peak (minimum peak) - <code>fetch:volt:minp?</code> , <code>meas:volt:minp?</code>	13
4.9	Positive Peak - <code>fetch:volt:ppe?</code> , <code>meas:volt:ppe?</code>	13

## 4.1 AC - `fetch:volt:ac?`, `meas:volt:ac?`

**Description** Read the AC voltage

**Command Syntax** `FETCh[:SCALar]:VOLTage:AC?`  
`MEASure[:SCALar]:VOLTage:AC?`

**Returned Value** <NR2>

## 4.2 DC - `fetch:volt:dc?`, `meas:volt:dc?`

**Description** Read the DC average voltage

**Command Syntax** `FETCh[:SCALar]:VOLTage:DC?`  
`MEASure[:SCALar]:VOLTage:DC?`

**Returned Value** <NR2>

## 4.3 RMS - `fetc:volt:rms?`, `meas:volt:rms?`

**Description** Read the voltage root-mean-square value.

**Command Syntax** `FETCh[:SCALar]:VOLTage:RMS?`  
`MEASure[:SCALar]:VOLTage:RMS?`

**Returned Value** <NR2>

## 4.4 Rectified Mean - `fetc:volt:rmn?`, `meas:volt:rmn?`

**Description** Read the voltage rectified mean value

**Command Syntax** `FETCh[:SCALar]:VOLTage:RMN?`  
`MEASure[:SCALar]:VOLTage:RMN?`

**Returned Value** <NR2>

## 4.5 Mean - `fetc:volt:mn?`, `meas:volt:mn?`

**Description** Read the MN(Calibration to the average rectified value of the effective value) of voltage.

**Command Syntax** FETCh[:SCALar]:VOLTage:MN?  
MEASure[:SCALar]:VOLTage:MN?

**Returned Value** <NR2>

#### 4.6 Crest Factor - fetc:volt:cfac?, meas:volt:cfac?

---

**Description** Read the voltage crest factor

**Command Syntax** FETCh[:SCALar]:VOLTage:CFACTOR?  
MEASure[:SCALar]:VOLTage:CFACTOR?

**Returned Value** <NR2>

#### 4.7 Maximum Peak - fetc:volt:maxp?, meas:volt:maxp?

---

**Description** Read the positive peak voltage

**Command syntax** FETCh[:SCALar]:VOLTage:MAXPk?  
MEASure[:SCALar]:VOLTage:MAXPk?

**Returned Value** <NR2>

#### 4.8 Negative peak (minimum peak) - fetch:volt:minp?, meas:volt:minp?

---

**Description** Read the negative peak voltage

**Command Syntax** FETCh[:SCALar]:VOLTage:MINPk?  
MEASure[:SCALar]:VOLTage:MINPk?

**Returned Value** <NR2>

#### 4.9 Positive Peak - fetch:volt:ppe?, meas:volt:ppe?

---

**Description** Read the peak-peak voltage

**Command Syntax** FETCh[:SCALar]:VOLTage:PPEak?  
MEASure[:SCALar]:VOLTage:PPEak?

**Returned Value** <NR2>

# Power Measurement Commands

5.1	Active - fetc:pow:act?, meas:pow:act?	14
5.2	Apparent - fetc:pow:app?, meas:pow:app?	14
5.3	Reactive - fetc:pow:reac?, meas:pow:reac?	14
5.4	Pfactor - fetc:pow:pfac?, meas:pow:pfac?	14
5.5	Phase - fetc:pow:phas?, meas:pow:phas?	14

## 5.1 Active - fetc:pow:act?, meas:pow:act?

---

<b>Description</b>	Read the active power
<b>Command Syntax</b>	FETCh[:SCALar]:POWer:ACTive? MEASure[:SCALar]:POWer:ACTive?
<b>Returned Value</b>	<NR2>

## 5.2 Apparent - fetc:pow:app?, meas:pow:app?

---

<b>Description</b>	Read the apparent power
<b>Command Syntax</b>	FETCh[:SCALar]:POWer:APParent? MEASure[:SCALar]:POWer:APParent?
<b>Returned Value</b>	<NR2>

## 5.3 Reactive - fetc:pow:reac?, meas:pow:reac?

---

<b>Description</b>	Read the reactive power
<b>Command Syntax</b>	FETCh[:SCALar]:POWer:REAActive? MEASure[:SCALar]:POWer:REAActive?
<b>Returned Value</b>	<NR2>

## 5.4 Pfactor - fetc:pow:pfac?, meas:pow:pfac?

---

<b>Description</b>	Read the power factor
<b>Command Syntax</b>	FETCh[:SCALar]:POWer:PFACTOR? MEASure[:SCALar]:POWer:PFACTOR?
<b>Returned Value</b>	<NR2>

## 5.5 Phase - fetc:pow:phas?, meas:pow:phas?

---

<b>Description</b>	Read the phase difference between voltage and current
<b>Command Syntax</b>	FETCh[:SCALar]:POWer:PHASe? MEASure[:SCALar]:POWer:PHASe?
<b>Returned Value</b>	<NR2>

# Frequency Measurement Commands

6.1	Voltage - <code>fetc:freq:volt?</code> , <code>meas:freq:volt?</code>	15
6.2	Current - <code>fetc:freq:curr?</code> , <code>meas:freq:curr?</code>	15
6.3	Synchronous Source - <code>fetc:freq:sso?</code> , <code>meas:freq:sso?</code>	15

## 6.1 Voltage - `fetc:freq:volt?`, `meas:freq:volt?`

---

<b>Description</b>	Read the voltage frequency
<b>Command Syntax</b>	<code>FETCh[:SCALar]:FREQuency:VOLTage?</code> <code>MEASure[:SCALar]:FREQuency:VOLTage?</code>
<b>Returned Value</b>	<NR2>

## 6.2 Current - `fetc:freq:curr?`, `meas:freq:curr?`

---

<b>Description</b>	Read the current frequency
<b>Command Syntax</b>	<code>FETCh[:SCALar]:FREQuency:CURRent?</code> <code>MEASure[:SCALar]:FREQuency:CURRent?</code>
<b>Returned Value</b>	<NR2>

## 6.3 Synchronous Source - `fetc:freq:sso?`, `meas:freq:sso?`

---

<b>Description</b>	Read the synchronous source frequency
<b>Command Syntax</b>	<code>FETCh[:SCALar]:FREQuency:SSource?</code> <code>MEASure[:SCALar]:FREQuency:SSource?</code>
<b>Returned Value</b>	<NR2>

# Energy Commands

7.1	Total value - fetc:ener?, meas:ener?	16
7.2	Positive - fetc:ener:pos?, meas:ener:pos?	16
7.3	Negative - fetc:ener:neg?, meas:ener:neg?	16
7.4	Accumulated Charge - fetc:ener:char?, meas:ener:char?	16
7.5	Postive - fetc:ener:char:pos?, meas:ener:char:pos?	16
7.6	Negative - fetc:ener:char:neg?, meas:ener:char:neg?	17
7.7	Integration Time - fetc:ener:time?, meas:ener:time?	17
7.8	Average - fetc:ener:aver?, meas:ener:aver?	17

## 7.1 Total value - fetc:ener?, meas:ener?

---

<b>Description</b>	Read the the sum of positive watt hour and negative watt hour
<b>Command Syntax</b>	FETCh[:SCALar]:ENERgy[ACTive][:SUM]? MEASure[:SCALar]:ENERgy[ACTive][:SUM]?
<b>Returned Value</b>	<NR2>

## 7.2 Positive - fetc:ener:pos?, meas:ener:pos?

---

<b>Description</b>	Read the positive watt hour.
<b>Command Syntax</b>	FETCh[:SCALar]:ENERgy[ACTive]:POSitive? MEASure[:SCALar]:ENERgy[ACTive]:POSitive?
<b>Returned Value</b>	<NR2>

## 7.3 Negative - fetc:ener:neg?, meas:ener:neg?

---

<b>Description</b>	Read the negative watt hour.
<b>Command Syntax</b>	FETCh[:SCALar]:ENERgy[ACTive]:NEGative? MEASure[:SCALar]:ENERgy[ACTive]:NEGative?
<b>Returned Value</b>	<NR2>

## 7.4 Accumulated Charge - fetc:ener:char?, meas:ener:char?

---

<b>Description</b>	Read the sum of positive ampere hour and negative ampere hour
<b>Command Syntax</b>	FETCh[:SCALar]:ENERgy:CHARge[:SUM]? MEASure[:SCALar]:ENERgy:CHARge[:SUM]?
<b>Returned Value</b>	<NR2>

## 7.5 Postive - fetc:ener:char:pos?, meas:ener:char:pos?

---

<b>Description</b>	Read the positive ampere hour
--------------------	-------------------------------



**Command Syntax** FETCh[:SCALar]:ENERgy:CHARge:POStive?  
MEASure[:SCALar]:ENERgy:CHARge:POStive?

**Returned Value** <NR2>

## 7.6 Negative - fetc:ener:char:neg?, meas:ener:char:neg?

---

**Description** Read the negative ampere hour

**Command Syntax** FETCh[:SCALar]:ENERgy:CHARge:NEGative?  
MEASure[:SCALar]:ENERgy:CHARge:NEGative?

**Returned Value** <NR2>

## 7.7 Integration Time - fetc:ener:time?, meas:ener:time?

---

**Description** Read the intergration time

**Command Syntax** FETCh[:SCALar]:ENERgy:TIME?  
MEASure[:SCALar]:ENERgy:TIME?

**Returned Value** <NR2>

## 7.8 Average - fetc:ener:aver?, meas:ener:aver?

---

**Description** Read the active power integral value

**Command Syntax** FETCh[:SCALar]:ENERgy[ACTive]:AVERage?  
MEASure[:SCALar]:ENERgy[ACTive]:AVERage?

**Returned Value** <NR2>

# Current Harmonics

8.1	Current Harmonic Amplitude - <code>fetc:harm:curr:ampl?</code> , <code>meas:harm:curr:ampl?</code>	18
8.2	Fundamental Harmonic - <code>fetc:harm:curr:fund?</code> , <code>meas:harm:curr:fund?</code>	18
8.3	Total Harmonic Current - <code>fetc:harm:curr:thar?</code> , <code>meas:harm:curr:thar?</code>	18
8.4	Total Harmonic Distortion (THD) - <code>fetc:harm:curr:thd?</code> , <code>meas:harm:curr:thd?</code>	18

## 8.1 Current Harmonic Amplitude - `fetc:harm:curr:ampl?`, `meas:harm:curr:ampl?`

---

<b>Description</b>	Read the current harmonics
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:CURRent:AMPLitude?</code> <code>MEASure[:SCALar]:HARMonic:CURRent:AMPLitude?</code>
<b>Returned Value</b>	<NR2>

## 8.2 Fundamental Harmonic - `fetc:harm:curr:fund?`, `meas:harm:curr:fund?`

---

<b>Description</b>	Read the current fundamental harmonic
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:CURRent:FUNDamental?</code> <code>MEASure[:SCALar]:HARMonic:CURRent:FUNDamental?</code>
<b>Returned Value</b>	<NR2>

## 8.3 Total Harmonic Current - `fetc:harm:curr:thar?`, `meas:harm:curr:thar?`

---

<b>Description</b>	Read the total harmonic current
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:CURRent:THARmonic?</code> <code>MEASure[:SCALar]:HARMonic:CURRent:THARmonic?</code>
<b>Returned Value</b>	<NR2>

## 8.4 Total Harmonic Distortion (THD) - `fetc:harm:curr:thd?`, `meas:harm:curr:thd?`

---

<b>Description</b>	Read the total harmonic distortion factor of current
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:CURRent:THDistort?</code> <code>MEASure[:SCALar]:HARMonic:CURRent:THDistort?</code>
<b>Returned Value</b>	<NR2>

# Voltage Harmonics

## 9.1 Total Harmonic Distortion Voltage Amplitude - `fetc:harm:volt:ampl?`, `meas:harm:volt:ampl?`

---

<b>Description</b>	Read the total harmonic distortion factor of voltage
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:VOLTage:AMPLitude?</code> <code>MEASure[:SCALar]:HARMonic:VOLTage:AMPLitude?</code>
<b>Returned Value</b>	<NR2>

## 9.2 Fundamental Harmonic Amplitude - `fetc:harm:volt:fund?`, `meas:harm:volt:fund?`

---

<b>Description</b>	Read the voltage fundamental harmonic
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:VOLTage:FUNDamental?</code> <code>MEASure[:SCALar]:HARMonic:VOLTage:FUNDamental?</code>
<b>Returned Value</b>	<NR2>

## 9.3 Total Harmonic Voltage - `fetc:harm:volt:fund?`, `meas:harm:volt:thar?`

---

<b>Description</b>	Read the total harmonic voltage
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:VOLTage:THARmonic?</code> <code>MEASure[:SCALar]:HARMonic:VOLTage:THARmonic?</code>
<b>Returned Value</b>	<NR2>

## 9.4 Total Harmonic Distortion (THD) Voltage - `fetc:harm:volt:thd?`, `meas:harm:volt:thd?`

---

<b>Description</b>	Read the total harmonic distortion factor of voltage.
<b>Command Syntax</b>	<code>FETCh[:SCALar]:HARMonic:VOLTage:THDistort?</code> <code>MEASure[:SCALar]:HARMonic:VOLTage:THDistort?</code>
<b>Returned Value</b>	<NR2>

# Power Harmonics

## 10.1 Amplitude - `fetc:harm:pow:ampl?`, `meas:harm:pow:ampl?`

---

**Description** Read the power harmonics.

**Command Syntax** `FETCh[:SCALar]:HARMonic:POWer[ACTive]:AMPLitude?`  
`MEASure[:SCALar]:HARMonic:POWer[ACTive]:AMPLitude?`

**Returned Value** <NR2>

## 10.2 Fundamental - `fetc:harm:pow:fund?`, `meas:harm:pow:fund?`

---

**Description** Read the power fundamental harmonic.

**Command Syntax** `FETCh[:SCALar]:HARMonic:POWer[ACTive]:FUNDamental?`  
`MEASure[:SCALar]:HARMonic:POWer[ACTive]:FUNDamental?`

**Returned Value** <NR2>

## 10.3 Total - `fetc:harm:pow:thar?`, `meas:harm:pow:thar?`

---

**Description** Read the total harmonic power.

**Command Syntax** `FETCh[:SCALar]:HARMonic:POWer[ACTive]:THARmonic?`  
`MEASure[:SCALar]:HARMonic:POWer[ACTive]:THARmonic?`

**Returned Value** <NR2>

## 10.4 Total Distortion - `fetc:harm:pow:thd?`, `meas:harm:pow:thd?`

---

**Description** Read the total harmonic distortion factor of power.

**Command Syntax** `FETCh[:SCALar]:HARMonic:POWer[ACTive]:THDistort?`  
`MEASure[:SCALar]:HARMonic:POWer[ACTive]:THDistort?`

**Returned Value** <NR2>

## 10.5 Apparent - `fetc:harm:pow:app?`, `meas:harm:pow:app?`

---

**Description** Read the apparent power harmonics

**Command Syntax** `FETCh[:SCALar]:HARMonic:POWer:APParent?`  
`MEASure[:SCALar]:HARMonic:POWer:APParent?`

**Returned Value** <NR2>

## 10.6 Reactive - `fetc:harm:pow:reac?`, `meas:harm:pow:reac?`

---

**Description** Read the reactive power harmonics

**Command Syntax** FETCh[:SCALar]:HARMonic:POWer:REACtive?  
MEASure[:SCALar]:HARMonic:POWer:REACtive?

**Returned Value** <NR2>

## 10.7 Power Factor - fetc:harm:pow:pfac?, meas:harm:pow:pfac?

---

**Description** Read the power factor harmonics

**Command Syntax** FETCh[:SCALar]:HARMonic:POWer:PFACTOR?  
MEASure[:SCALar]:HARMonic:POWer:PFACTOR?

**Returned Value** <NR2>

## 10.8 K-th order voltage harmonic - fetc:harm:pow:phase:uu?, meas:harm:pow:phase:uu?

---

**Description** Read the phase difference of K-order harmonic voltage(Uk) and fundamental wave(U1)

**Command Syntax** FETCh[:SCALar]:HARMonic:POWer:PHASe:UU?  
MEASure[:SCALar]:HARMonic:POWer:PHASe:UU?

**Returned Value** <NR2>

## 10.9 K-th order harmonic current and voltage - fetc:harm:pow:phase:ui?, meas:harm:pow:phase:ui?

---

**Description** Read the phase difference of K-order harmonic voltage(Uk) and harmonic current

**Command Syntax** FETCh[:SCALar]:HARMonic:POWer:PHASe:UI?  
MEASure[:SCALar]:HARMonic:POWer:PHASe:UI?

**Returned Value** <NR2>

## 10.10 K-th order current harmonic - fetc:harm:pow:phase:ii?, meas:harm:pow:phase:ii?

---

**Description** Read the phase difference of K-order harmonic current(Ik) and fundamental wave(I1)

**Command Syntax** FETCh[:SCALar]:HARMonic:POWer:PHASe:II?  
MEASure[:SCALar]:HARMonic:POWer:PHASe:II?

**Returned Value** <NR2>

# Harmonics Etc...

## 11.1 Harmonic orders - `harm:order`

---

**Description** set the harmonic analysis orders.

**Command Syntax** `HARMonic:ORDer <NRI>`

**Parameters** 2-50

**Query Syntax** `HARMonic:ORDer?`

## 11.2 Measurement source for PLL - `harm:pll`

---

**Description** This command set PLL (Phase Locked Loop)source which is used for determining the fundamental wave cycle as the reference for analysis of harmonic orders.

**Command Syntax** `HARMonic:PLLSource <OFF,U,I>`

**Parameters** OFF,U,I

**Query Syntax** `HARMonic:PLLSource?`

## 11.3 Harmonic calculation formula - `harm:thd`

---

**Description** set the calculation formula of THD(Total Harmonic Distrotion).

**Command Syntax** `HARMonic:THD <THDR, THDF>`

**Parameters** %r,%f

**Query Syntax** `HARMonic:THD?`

## 11.4 Harmonic set - `harm:seq`

---

**Description** select the harmonic sequence of ALL/ODD/EVEN.

**Command Syntax** `HARMonic:SEQuence <ALL,ODD,EVEN >`

**Parameters** ALL,ODD,EVEN

**Query Syntax** `HARMonic:SEQuence?`

# Measurement Setup

12.1	Average enable/disable	23
12.2	Averaging time	23
12.3	Average Type	23
12.4	Samples per average	23
12.5	Current Auto Range enable/disable	24
12.6	Voltage Auto Range enable/disable	24
12.7	Current Range	24
12.8	Voltage Range	24
12.9	External Sensor 1 enable/disable	24
12.10	External Sensor 2 enable/disable	25
12.11	Scaling factor: External Sensor 1	25
12.12	Scaling factor: External Sensor 2	25
12.13	External sensor current range	25

## 12.1 Average enable/disable

**Description** Enable or disable the average function.

**Command Syntax** AVERAge <b>

**Parameters** 0,1,OFF,ON

**Query Syntax** [:SENSe] AVERAge:STATe?

## 12.2 Averaging time

**Description** set the linear averaging type.

**Command Syntax** AVERAge:TCONtrol <name>

**Parameters** REPeat, MOVing

**Query Syntax** AVERAge:TCONtrol?

## 12.3 Average Type

**Description** set the mode of the averaging function.

**Command Syntax** AVERAge:TYPE <boolean>

**Parameters** EXP,LINE

**Query Syntax** AVERAge:TYPE?

## 12.4 Samples per average

**Description** set the times of the averaging function.

**Command Syntax** AVERAge:COUNT <n>

**Parameters** 1764

**Query Syntax** AVERAge:COUNT?

## 12.5 Current Auto Range enable/disable

---

**Description** Enable or disable the current auto range function.

**Command Syntax** CURRent:RANGe:AUTO <Boolean>

**Parameters** OFF,ON

**Query Syntax** CURRent:RANGe:AUTO?

## 12.6 Voltage Auto Range enable/disable

---

**Description** Enable or disable the voltage auto range function.

**Command Syntax** VOLTage:RANGe:AUTO <Boolean>

**Parameters** OFF,ON

**Query Syntax** VOLTage:RANGe:AUTO?

## 12.7 Current Range

---

**Description** set the current range.

**Command Syntax** CURRent:RANGe <NR.f>

**Parameters** 5mA/10mA/20mA/50mA/100mA/200mA/0.5A/1A/2A/5A/10A/20A

**Query Syntax** CURRent:RANGe?

## 12.8 Voltage Range

---

**Description** set the voltage range.

**Command Syntax** VOLTage:RANGe <NR.f>

**Parameters** 15V,30V,60V,150V,300V,600V

**Query Syntax** VOLTage:RANGe?

## 12.9 External Sensor 1 enable/disable

---

**Description** Enable or disable the external current sensor 1.

**Command Syntax** CURRent:EXS1[:STATe] <b>

**Parameters** OFF,ON

**Query Syntax** CURRent:EXS1:STATe?



## 12.10 External Sensor 2 enable/disable

---

**Description** Enable or disable the external current sensor 2.

**Command Syntax** CURRent:EXS2[:STATe] <b>

**Parameters** OFF,ON

**Query Syntax** CURRent:EXS2:STATe?

## 12.11 Scaling factor: External Sensor 1

---

**Description** The command is used to set the conversion ratio of the external current sensor 1.

**Command Syntax** CURRent:SRATio:EXS1 <NRf>

**Parameters** 0.001?9999.999

**Query Syntax** CURRent:SRATio:EXS1?

## 12.12 Scaling factor: External Sensor 2

---

**Description** The command is used to set the conversion ratio of the external current sensor 2.

**Command Syntax** CURRent:SRATio:EXS2 <NRf>

**Parameters** 0.001?9999.999

**Query Syntax** CURRent:SRATio:EXS2?

## 12.13 External sensor current range

---

**Description** set the current range of external sensor. Default unit is V.

**Command Syntax** CURRent:EXS:RANGe <NRf>

**Parameters** <NRf>

**Query Syntax** CURRent:EXS:RANGe?

# Trigger Commands

13.1	Trigger Abort	26
13.2	Continuous trigger enable/disable	26
13.3	Initiate Trigger	26
13.4	Immediate Trigger	26
13.5	Source	27
13.6	Edge (slope)	27
13.7	Voltage Level	27
13.8	Current Level	27

## 13.1 Trigger Abort

**Description** This command set related sensors in idle condition. When measuring process is terminated, it will start a new self trigger system of measurement without affected.

**Command Syntax** ABORt

**Parameters** None

**Query Syntax** None

## 13.2 Continuous trigger enable/disable

**Description** Enable or disable the state of continuous measurement period. In off mode, that means to enable a single measurement.

**Command Syntax** INITiate:CONTinuous <state>

**Parameters** ON/OFF

**Query Syntax** INITiate:CONTinuous?

## 13.3 Initiate Trigger

**Description** This command will create a single measurement operation and relevant sensors are in request status.

**Command Syntax** INITiate:IMMediate

**Parameters** None

**Query Syntax** None

## 13.4 Immediate Trigger

**Description** This command is available for all trigger mode. It will generate a trigger signal.

**Command Syntax** TRIGger:IMMediate

**Parameters** None

**Relevant Commands** \*TRG TRIG:SOUR

## 13.5 Source

**Description** select trigger source. Trigger source modes are as below. When execute \*RST command, trigger source should be set in MANUAL mode.

**Parameters** TRIGger:SOURce <state>

**Parameters** IMMEDIATE,BUS,EXTERNAL,VOLTage,CURRENT

**Query Syntax** TRIGger:SOURce?

## 13.6 Edge (slope)

**Description** set the trigger slope, parameters could be rising edge/descending dege/any edge.

**Command Syntax** TRIGger:SLOPe <state>

**Parameters** Up/Down/Any[JL1]

**Query Syntax** TRIGger:SLOPe?

## 13.7 Voltage Level

**Description** set the voltage trigger level when Voltage selected as trigger source mode.

**Command Syntax** TRIGger:VOLTage:LEVel <level>

**Parameters** <level>

**Query Syntax** TRIGger:VOLTage:LEVel?

## 13.8 Current Level

**Description** set the current trigger level when Current selected as trigger source mode.

**Command Syntax** TRIGger:CURRENT:LEVel <level>

**Parameters** level

**Query Syntax** TRIGger:CURRENT:LEVel?

# Oscilloscope

14.1	Trigger source	28
14.2	Trigger edge (slope)	28
14.3	Trigger mode	28
14.4	Trigger delay	28
14.5	Time per division (Horizontal Scale)	29
14.6	Run (Start Capture)	29
14.7	Stop (Stop Capture)	29
14.8	Single capture	29
14.9	Retrieve waveform voltage data	29
14.10	Retrieve waveform current data	30
14.11	Real time data	30
14.12	Trigger status	30

## 14.1 Trigger source

**Description** This command can set or read the device taking which kind waveform as trigger source.

**Command Syntax** WAVE:TRIG:SOUR < VOLTage,CURRent,EXTernal >

**Parameters** VOLTage,CURRent,EXTernal

**Query Syntax** WAVE:TRIG:SOUR?

## 14.2 Trigger edge (slope)

**Description** set trigger slope when takes waveform as trigger source.

**Command Syntax** WAVE:TRIG:SLOP < POSitive,NEGative,ANY >

**Parameters** POSitive,NEGative,ANY

**Query Syntax** WAVE:TRIG:SLOP?

## 14.3 Trigger mode

**Description** set the trigger mode when take waveform as trigger source.

**Command Syntax** WAVE:TRIG:MODE < AUTO,NORMal >

**Parameters** AUTO,NORMal

**Query Syntax** WAVE:TRIG:MODE?

## 14.4 Trigger delay

**Description** This time is used to set the trigger delay time when take a waveform as trigger source.(Note:the delay time setting should be less than the time width of the whole screen.)

**Command Syntax** WAVE:TRIG:DElay:TIME <n>

**Parameters** <n> Unit S

**Query Syntax** WAVE:TRIG:DElay:TIME?

## 14.5 Time per division (Horizontal Scale)

---

**Description** set the time/grid value of the waveform display.  
?Avaible setting:0.0005,0.001,0.002,0.005,,0.01,0.02.,0.05,0.1,0.2,0.5?

**Command Syntax** WAVE:TRIG:DIVTime?UNIT:S?<n>

**Parameters** <n> Unit S

**Query Syntax** WAVE:TRIG:DIVTime?

## 14.6 Run (Start Capture)

---

**Description** start waveform capture.

**Command Syntax** WAVE:RUN

**Parameters** None

**Query Syntax** None

## 14.7 Stop (Stop Capture)

---

**Description** This command can stop the waveform capture.

**Command Syntax** WAVE:STOP

**Parameters** None

**Query Syntax** None

## 14.8 Single capture

---

**Description** trigger a single waveform capture.

**Command Syntax** WAVE:SINGLE

**Parameters** None

**Query Syntax** None

## 14.9 Retrieve waveform voltage data

---

**Description** obtain the voltage datas after normalization.

**Command Syntax** WAVE:VOLTage:DATA?

**Parameters** <n>

**Query Syntax** WAVE:VOLTage:DATA?

## 14.10 Retrieve waveform current data

---

**Description** obtain the current datas after normalization.

**Command Syntax** WAVE:CURRent:DATA[:NORMAlization]?

**Parameters** <n>

**Query Syntax** WAVE:CURRent:DATA[:NORMAlization]?

## 14.11 Real time data

---

**Description** obtain real-time datas of voltage and current.(Total data' capacity is 640.The first half is voltage datas.)

**Command Syntax** WAVE:DATA[:REAL]?

**Parameters** <n>,<n>

**Query Syntax** WAVE:DATA[:REAL]?

## 14.12 Trigger status

---

**Description** query the trigger status.

**Command Syntax** WAVE:TRIGger[:STATe]?

**Returned Value** Auto, Auto?, Trig, Trig?, Stop

**Query Syntax** WAVE:TRIGger[:STATe]?

# Input Commands

15.1	Synchronous source	31
15.2	Crest factor setting	31
15.3	Filter enable/disable	31
15.4	Line filter enable/disable	31
15.5	Update rate	32
15.6	Automatic integral calculation enable/disable	32
15.7	Integration current mode	32
15.8	Integration power type	32
15.9	Inrush measurement enable/disable	32
15.10	Inrush current trigger level	33
15.11	Inrush measurement trigger delay	33
15.12	Inrush measurement time	33

## 15.1 Synchronous source

**Description** set the synchronous source type.

**Command Syntax** SSource <OFF,U,I>

**Parameters** OFF,U,I

**Query Syntax** SSource?

## 15.2 Crest factor setting

**Description** set the crest factor.

**Command Syntax** CFACtor <NRI>

**Parameters** 3,6

**Query Syntax** CFACtor?

## 15.3 Filter enable/disable

**Description** set the state of frequency filter.

**Command Syntax** FILTer:FREQuency <boolean>

**Parameters** OFF,ON[JL5]

**Query Syntax** FILTer:FREQuency?

## 15.4 Line filter enable/disable

**Description** set the state of line filter.

**Command Syntax** FILTer:LINE <boolean>

**Parameters** OFF,ON  
**Query Syntax** FILTER:LINE?

## 15.5 Update rate

---

**Description** set the data updating rate.  
**Command Syntax** RATE <NRI>  
**Parameters** 0.1s/0.25s/0.5s/1s/2s/5s  
**Query Syntax** RATE?

## 15.6 Automatic integral calculation enable/disable

---

**Description** Enable or disable the auto calculation function of integral.  
**Command Syntax** INTEGRAL:ACAL <boolean>  
**Parameters** OFF,ON  
**Query Syntax** INTEGRAL:ACAL?

## 15.7 Integration current mode

---

**Description** set the current integral mode.  
**Command Syntax** INTEGRATE:QMODE RMS,MN,DC,RMN,AC  
**Parameters** RMS,MN,DC,RMN,AC  
**Query Syntax** INTEGRATE:QMODE?

## 15.8 Integration power type

---

**Description** set integral mode of watt hour.  
**Command Syntax** INTEGRATE:WPTYPE CHARGE,SOLD,DISCHARGE,BOUGHT  
**Parameters** CHARGE,SOLD,DISCHARGE,BOUGHT  
**Query Syntax** INTEGRATE:WPTYPE?

## 15.9 Inrush measurement enable/disable

---

**Description** Enable or disable the inrush current measurement function.  
**Command Syntax** [INPUT:]INRUSH[:STATE] < OFF,ON >  
**Parameters** OFF,ON  
**Query Syntax** INRUSH?



## 15.10 Inrush current trigger level

---

**Description** set the trigger level for inrush current measurement.

**Command Syntax** [INPut:]INRush:TRIGger:CURRent[:LEVel] <n>

**Parameters** <n> Unit A

**Query Syntax** INRush:TRIGger:CURRent?

## 15.11 Inrush measurement trigger delay

---

**Description** set the delay time of inrush current measurement.(Max time is 0.0002s).

**Command Syntax** [INPut:]INRush:DELay:TIME <n>

**Parameters** <n> Unit S

**Query Syntax** INRush:DELay:TIME?

## 15.12 Inrush measurement time

---

**Description** set the inrush current measuring time.(max time is 300s).

**Command Syntax** [INPut:]INRush:MEASure:TIME <n>

**Parameters** <n> Unit S

**Query Syntax** INRush:MEASure:TIME?

# Calculation setup

16.1	Integration state	34
16.2	Integration reset (clear)	34
16.3	Automatic integration clearing enable/disable	34
16.4	Integration start source	34
16.5	Start integration	35
16.6	Integration start date, time	35
16.7	Integration stop signal source	35
16.8	Stop integration	35
16.9	Integration stop date, time	35
16.10	Integration duration	36
16.11	Integration status/condition	36
16.12	Max value hold enable/disable	36
16.13	Clear measurment	36
16.14	Harmonic calculation enable/disable	36
16.15	Oscilloscope function enable/disable	37
16.16	Hold mode enable/disable	37

## 16.1 Integration state

<b>Description</b>	state of integral function.
<b>Command Syntax</b>	CALCulate:INTEgral[:STATe]
<b>Parameters</b>	ON,OFF
<b>Query Syntax</b>	CALCulate:INTEgral?

## 16.2 Integration reset (clear)

<b>Description</b>	clear all integral value.
<b>Command Syntax</b>	CALCulate:INTEgral:CLEar
<b>Parameters</b>	None
<b>Query Syntax</b>	None

## 16.3 Automatic integration clearing enable/disable

<b>Description</b>	Enable or disable the automatic zero clearing function.
<b>Command Syntax</b>	INTEgral:CLEar:AUTO <ON,OFF>
<b>Parameters</b>	ON,OFF
<b>Query Syntax</b>	INTEgral:CLEar:AUTO?

## 16.4 Integration start source

<b>Description</b>	define the mode how to start the integral function. Time:start to integral on the set date
--------------------	--

MAN:press the “START” soft key in the integral measurement interface to trigger the manual start of the integral function.

**Command Syntax** INTegral:STARt:SOURce <TIME,MAN>

**Parameters** TIME,MAN

**Query Syntax** INTegral:STARt:SOURce?

## 16.5 Start integration

**Description** When integral start mode is set to CMD,sending this command will trigger integral operation.

**Command Syntax** INTegral:STARt

**Parameters** None

**Query Syntax** None

## 16.6 Integration start date, time

**Description** set the start date,when time reached,device start integral operation.

**Command Syntax** INTegral:STARt:DATE <yy,MM,dd> or INTegral:STARt:TIME <hh,mm,ss>

**Parameters** <yy,MM,dd> <hh,mm,ss>

**Query Syntax** INTegral:STARt:DATE?  
INTegral:STARt:TIME?

## 16.7 Integration stop signal source

**Description** set the integral stop mode.

**Command Syntax** INTegral:STOP:SOURce < TIME,MAN,TINTerval>

**Parameters** TIME,MAN,TINTerval

**Query Syntax** INTegral:STOP:SOURce?

## 16.8 Stop integration

**Description** When integral stop mode is set to CMD,sending this command will stop the integral operation.When stopped,integral value will not be cleared to zero.

**Command Syntax** INTegral:STOP

**Parameters** None

**Query Syntax** None

## 16.9 Integration stop date, time

**Description** set the stop date,when time reached,device stops integral operation.

**Command Syntax** INTegral:STOP:DATE:<yy,MM,dd> or INTegral:STOP:TIME: <hh,mm,ss>

**Parameters** <yy,MM,dd> <hh,mm,ss>

**Query Syntax** INTegral:STOP:DATE?  
INTegral:STOP:TIME?

## 16.10 Integration duration

**Description** When integral stop mode is set to TINTerval,user can set the integral measuring period via this command.When time length reached,integral operation will be terminated.

**Command Syntax** INTegral:STOP:TINTerval < hhhh,mm,ss >

**Parameters** hhhh,mm,ss

**Query Syntax** INTegral:STOP:TINTerval?

## 16.11 Integration status/condition

**Description** obtain the integral running state.

**Command Syntax** INTegral:CONDition?

**Returned Value** Ready,Start,Stop,Time up,Error

**Query Syntax** INTegral:CONDition?

## 16.12 Max value hold enable/disable

**Description** set the hold mode of maximum value.

**Command Syntax** CALCulate:METer:MAXHold <boolean>

**Parameters** OFF,ON

**Query Syntax** CALCulate:METer:MAXHold?

## 16.13 Clear measurement

**Description** clear measuring value to zero.

**Command Syntax** CALCulate:METer:CLEar:IMMediate

**Parameters** None

**Query Syntax** None

## 16.14 Harmonic calculation enable/disable

**Description** set the harmonic state.

**Command Syntax** CALCulate:HARMonic <boolean>

**Parameters** OFF,ON

**Query Syntax** CALCulate:HARMonic?

## **16.15 Oscilloscope function enable/disable**

---

**Description** Enable or disable the oscilloscope function.

**Command Syntax** CALCulate:SCOPE <boolean>

**Parameters** OFF,ON

**Query Syntax** CALCulate:SCOPE?

## **16.16 Hold mode enable/disable**

---

**Description** Enable or disable the Hold mode.

**Command Syntax** [CALCulate:]HOLD[:STATe]

**Parameters** OFF,ON

**Query Syntax** HOLD?

# System Commands

17.1	Beep test	38
17.2	Beep enable/disable	38
17.3	System date	38
17.4	System time	38
17.5	Last key pressed	39
17.6	Retrieve error	39
17.7	Language	39
17.8	Set to local control	39
17.9	Set to remote mode	39
17.10	Error clear	40
17.11	Remote lock	40
17.12	Version	40

## 17.1 Beep test

**Description** test the beeper. The power meter will beep for once after executing this command.

**Command Syntax** SYSTem:BEEPer:IMMEDIATE

**Parameters** None

**Query Syntax** None

**Note** A real office/lab/factory pleaser.

## 17.2 Beep enable/disable

**Description** turn on or turn off the beeper.

**Command Syntax** SYSTem:BEEPer <boolean>

**Parameters** OFF,ON,0,1

**Query Syntax** SYSTem:BEEPer:STATE?

## 17.3 System date

**Description** set the system date.

**Command Syntax** SYSTem:DATE <NRf>,<NRf>,<NRf>

**Parameters** yyyy,mm,dd

**Query Syntax** SYSTem:DATE?

## 17.4 System time

**Description** set the system time.

**Command Syntax** SYSTem:TIME <NRf>,<NRf>,<NRf>

**Parameters** hh,mm,ss

**Query Syntax** SYSTem:TIME?

## 17.5 Last key pressed

**Description** query the final one of the key pressed.

**Command Syntax** SYSTem:KEY?

**Parameters** None

## 17.6 Retrieve error

**Description** query the error information.

**Command Syntax** SYSTem:ERRor?

**Parameters** None

## 17.7 Language

**Description** This command is use to set the system language.

**Command Syntax**

**Parameters**

**Query Syntax** ?

## 17.8 Set to local control

**Description** switch the power meter to local operation mode.

**Command Syntax** SYST:LOC

**Parameters** None

**Query Syntax** None

## 17.9 Set to remote mode

**Description** switch the power meter to remote control mode.In this mode,except Esc button(pressing this button for 5s will switch the unit to local mode),other keys are locked.

**Command Syntax** SYST:REM

**Parameters** None

**Query Syntax** None

## 17.10 Error clear

<b>Description</b>	clear the error information.
<b>Command Syntax</b>	SYSTem:CLEar
<b>Parameters</b>	None
<b>Returned Value</b>	None

## 17.11 Remote lock

<b>Description</b>	set power meter to remote control mode via RS232 communication interface.And Esc button is not available.Other buttons are locked too.
<b>Command Syntax</b>	SYST:RWL
<b>Parameters</b>	None
<b>Returned Value</b>	None

## 17.12 Version

<b>Description</b>	query the device version.Return value is a character string as like YYYY.V.YYYY represents the year and V means the version of that year.
<b>Command Syntax</b>	SYST:VERS?
<b>Parameters</b>	None
<b>Returned Value</b>	<NR2>
<b>Example</b>	



# Calibration Commands

18.1	Zero	41
18.2	Unlock	41
18.3	Initialize	41
18.4	Factory Save	42
18.5	Save	42
18.6	Current calibration point	42
18.7	Current reading	42
18.8	External current point	43
18.9	External current reading	43
18.10	Voltage point	43
18.11	Voltage level reading	43

## 18.1 Zero

**Description** Zero calibration command.

**Command Syntax** CALibration:ZERO

**Parameters** None

## 18.2 Unlock

**Description** The calibration mode must be enabled before the load will accept any other calibration commands. The first parameter specifies the enabled or disabled state. The second parameter is the password. It is required if the calibration mode is being enabled and the existing password is not 0. If the password is not entered or is incorrect, an error is generated and the calibration mode remains disabled. The query statement returns only the state, not the password. Whenever the calibration state is changed from enabled to disabled, any new calibration constants are lost unless they have been stored with CALibrate:SAVE.  
enable or disable calibration mode.

**Command Syntax** CALibrate:SECure[:STATe] <bool> [,<SRD>]

**Parameters** 0 , 1 , OFF , ON [,<password>]

**Reset value** ON

**Example** CAL:SEC 0, N3301A CAL:SEC ON

**Query Syntax** CALibrate:SECure[:STATe]?

**Parameters** <NR1>

**Relevant Commands** CAL:SAVE CAL:INIT

## 18.3 Initialize

**Description** This command can only be used in calibration mode. It restore factory calibration constants from nonvolatile memory.

**Command Syntax** CALibrate:INITial

**Parameters** None

**Example** CAL:INIT

Relevant command

## 18.4 Factory Save

**Description** CHAN:FACTory:SAVe

**Command Syntax** CHAN:FACTory:SAVe

**Parameters** None

## 18.5 Save

**Description** This command can only be used in calibration mode. It saves any new calibration constants (after a current or voltage calibration procedure has been completed) in nonvolatile memory.

**Command Syntax** CALibrate:SAVE

**Parameters** None

**Example** Relevant command CAL:STAT CAL:INIT

## 18.6 Current calibration point

**Description** specify the current calibration points.

**Command Syntax** CALibrate:CURRent:POINt <point>

**Parameters** P1 , P2 , P3 , P4

**Example**

**Related commands** CAL:STAT, CAL:SAV

## 18.7 Current reading

**Description** It enters a calibration voltage value that you obtain by reading an external meter. You must first select a calibration level (with CALibrate:VOLTage:POINt) for the value being entered. These constants are not stored in nonvolatile memory until they are saved with CALibrate:SAVE. This command only used for calibration mode.

**Command Syntax** CALibrate:CURRent[:LEVel] <NRf>

**Parameters** external reading

Unit:A (amps)

**Example** Relevant command CAL:STAT CAL:SAV

## 18.8 External current point

---

**Description** calibrate external current points.

**Command Syntax** CALibrate:CURRent:EXTernal:POINt<point>

**Parameters** <point>

## 18.9 External current reading

---

**Description** input external calibration current.

**Command Syntax** CALibrate:CURRent:EXTernal:[LEVel] <NRf>

**Parameters** <NRf>

## 18.10 Voltage point

---

**Description** This command can only be used in calibration mode. It is used to set the calibration points of constant voltage mode. P1, P2 is used in low voltage meter range, P3, P4 is used in high voltage meter range. It can use calibrate voltage source and voltage meter.

**Command Syntax** CALibrate:VOLTage:POINt <point>

**Parameters** P1 , P2 , P3 , P4

**Example** Relevant command CAL:STAT CAL:SAV

## 18.11 Voltage level reading

---

**Description** This command is only used in calibration mode. It enters a calibration voltage value that you obtain by reading an external meter. You must first select a calibration level (with CALibrate:VOLTage:POINt) for the value being entered. These constants are not stored in nonvolatile memory until they are saved with CALibrate:SAVE.

**Command Syntax** CALibrate:VOLTage[:LEVel] <NRf>

**Parameters** external reading Unit V (volts)

**Example**

**Related Commands** CAL:STAT CAL:SAV