



## **IMPORTANT SAFETY INSTRUCTIONS**

This equipment is designed to comply with BSEN 61010-1 (Safety requirements for electrical equipment for measurement, control, and laboratory use) – observe the following precautions:

- Ensure that the supply voltage agrees with the rating of the instrument printed on the back panel **before** connecting the mains cord to the supply.
- This appliance **must** be earthed. Ensure that the instrument is powered from a properly grounded supply.
- The inputs must not be connected to signals greater than is indicated on the front panel.
- Keep the ventilation holes on the underneath and sides free from obstruction.
- Do not operate or store under conditions where condensation may occur or where conducting debris may enter the case.
- There are no user serviceable parts inside the instrument – do not attempt to open the instrument, refer service to the manufacturer or his appointed agent.

**Note: Newtons4th Ltd. shall not be liable for any consequential damages, losses, costs or expenses arising from the use or misuse of this product however caused.**

## **ABOUT THIS MANUAL**

This manual gives details of the communication commands recognized by the PSM17xx series of instruments over RS232, LAN or GPIB. For more general operating instructions for the instrument refer to the specific user manual.

Each command is listed alphabetically with details of any arguments and reply. A one line summary of each command is given in the appendix. Although most of the commands apply to all instruments in the range there are some commands that are specific to one instrument or another.

The information in this manual is believed to be accurate and complete but Newtons4th Ltd cannot accept any liability whatsoever for any consequential damage or losses arising from any errors, inaccuracies, or omissions.

Revision 1.49 (PsimetriQ) 1.33 (NumetriQ)

This manual is copyright © 2004 - 2007 Newtons4th Ltd. and all rights are reserved. No part may be copied or reproduced in any form without prior written consent.

1 Sep 2007

CONTENTS

1	Using remote control.....	1-1
1.1	Standard event status register.....	1-3
1.2	Serial Poll status byte.....	1-4
1.3	RS232 connections.....	1-5
1.4	Data streaming.....	1-6
2	Communication commands .....	2-1
	*CLS.....	2-1
	*ESE.....	2-2
	*ESR?.....	2-3
	*IDN?.....	2-4
	*OPC?.....	2-5
	*RST.....	2-6
	*SRE.....	2-7
	*STB?.....	2-9
	*TRG.....	2-10
	*TST?.....	2-11
	*WAI.....	2-12
	ABORT.....	2-13
	ACTRIM.....	2-14
	ALARM.....	2-15
	ALARM?.....	2-16
	ALARM1.....	2-17
	ALARM2.....	2-18
	ALARME.....	2-19
	ALARME?.....	2-20
	AMPLIT.....	2-21
	ANALOG.....	2-22
	BANDWI.....	2-23
	BEEP.....	2-24
	BLANKI.....	2-25
	CONFIG.....	2-26
	COUPLI.....	2-28
	DATALO.....	2-29
	DAV?.....	2-31
	DAVER.....	2-32
	FILTER.....	2-34
	FRA.....	2-35
	FREQUE.....	2-37
	FSWEEP.....	2-38

# PSM17xx communications manual

GAINPH .....	2-39
HARMON .....	2-41
INPUT .....	2-43
INTYPE .....	2-44
KEYBOA .....	2-45
LCR .....	2-46
LOWFRE .....	2-48
MARKER .....	2-49
MODE .....	2-50
OFFSET .....	2-51
OUTPUT .....	2-52
PAV .....	2-53
PHASE .....	2-55
PHCONV .....	2-57
PHREF .....	2-58
POWER .....	2-59
PPORT .....	2-61
PROGRA .....	2-63
RANGE .....	2-65
RESOLU .....	2-66
REZERO .....	2-67
SCALE .....	2-68
SHUNT .....	2-69
SPEED .....	2-70
SSWEEP .....	2-71
START .....	2-72
STATUS? .....	2-73
STOP .....	2-74
STREAM .....	2-75
TFA .....	2-77
TXA .....	2-79
TXTEST .....	2-81
USER? .....	2-82
VECTOR .....	2-83
VERSIO? .....	2-85
VRMS .....	2-86
WAVEFO .....	2-88
ZERO .....	2-89
ZOOM .....	2-90

Appendix – command summary

1

1 Using remote control

The instrument is fitted with an RS232 serial communications port as standard, and may have an IEEE488 (GPIB) interface or LAN interface fitted as an option. All the interfaces use the same ASCII protocol with the exception of the end of line terminators:

	Rx expects	Tx sends
RS232	carriage return (line feed ignored)	carriage return and line feed
LAN	carriage return (line feed ignored)	carriage return and line feed
IEEE488	carriage return or line feed or EOI	carriage return with EOI

All the functions of the instrument can be programmed via either interface, and results read back. When the IEEE488 interface is set to 'remote' the RS232 port is ignored.

The commands are not case sensitive and white space characters are ignored (e.g. tabs and spaces). Replies from the instrument are always upper case, delimited by commas, without spaces.

Only the first six characters of any command are important – any further characters will be ignored. For example, the command to set the generator frequency is FREQUE but the full word FREQUENCY may be sent as the redundant NCY at the end will be ignored.

Fields within a command are delimited by comma, multiple commands can be sent on one line delimited with a semi-colon. Eg.

AMPLIT,1.5;OUTPUT,ON

Mandatory commands specified in the IEEE488.2 protocol have been implemented, (e.g. \*IDN?, \*RST) and all

commands that expect a reply are terminated with a question mark.

The instrument maintains an error status byte consistent with the requirements of the IEEE488.2 protocol (called the standard event status register) that can be read by the mandatory command \*ESR? (see section 5.1).

The instrument also maintains a status byte consistent with the requirements of the IEEE488.2 protocol, that can be read either with the IEEE488 serial poll function or by the mandatory command \*STB? over RS232 or IEEE or LAN (see section 5.2).

The IEEE address defaults to 23 and can be changed via the COMMS menu.

The keyboard is disabled when the instrument is set to "remote" using the IEEE. Press HOME to return to "local" operation.

RS232 data format is: start bit, 8 data bits (no parity), 1 stop bit. Flow control is RTS/CTS (see section 5.2), baud rate is selectable via the MONITOR menu.

A summary of the available commands is given in the Appendix. Details of each command are given in the communication command section of the manual.

Commands are executed in sequence except for two special characters that are immediately obeyed:

- Control T (20) – reset interface (device clear)
- Control U (21) – warm restart

1.1 Standard event status register

PON		CME	EXE	DDE	QYE		OPC
-----	--	-----	-----	-----	-----	--	-----

- bit 0 OPC (operation complete)  
cleared by most commands  
set when data available or sweep complete
- bit 2 QYE (unterminated query error)  
set if no message ready when data read
- bit 3 DDE (device dependent error)  
set when the instrument has an error
- bit 4 EXE (execution error)  
set when the command cannot be executed
- bit 5 CME (command interpretation error)  
set when a command has not been recognised
- bit 7 PON (power on event)  
set when power first applied or unit has reset

The bits in the standard event status register except for OPC are set by the relevant event and cleared by specific command (\*ESR?, \*CLS, \*RST). OPC is also cleared by most commands that change any part of the configuration of the instrument (such as MODE or START).

1.2 Serial Poll status byte

		ESB	MAV	ALM	FDV	SDV	RDV
--	--	-----	-----	-----	-----	-----	-----

- bit 0 RDV (result data available)  
set when results are available to be read as enabled by DAVER
- bit 1 SDV (sweep data available)  
set when sweep results are available to be read as enabled by DAVER
- bit 2 FDV (fast data available (streaming))  
set when data streaming results are available to be read as enabled by DAVER
- bit 3 ALA (alarm active)  
set when an alarm is active and enabled by ALARMER
- bit 4 MAV (message available)  
set when a message reply is waiting to be read
- bit 5 ESB (standard event summary bit)  
set if any bit in the standard event status register is set as well as the corresponding bit in the standard event status enable register (set by \*ESE).

### 1.3 RS232 connections

The RS232 port on the instrument uses the same pinout as a standard 9 pin serial port on a PC or laptop (9-pin male 'D' type).

Pin	Function	Direction
1	DCD	in (+ weak pull up)
2	RX data	in
3	TX data	out
4	DTR	out
5	GND	
6	DSR	not used
7	RTS	out
8	CTS	in
9	RI	not used

The instrument will only transmit when CTS (pin 8) is asserted, and can only receive if DCD (pin 1) is asserted. The instrument constantly asserts (+12V) DTR (pin 4) so this pin can be connected to any unwanted modem control inputs to force operation without handshaking. The instrument has a weak pull up on pin 1 as many null modem cables leave it open circuit. In electrically noisy environments, this pin should be driven or connected to pin 4.

To connect the instrument to a PC, use a 9 pin female to 9 pin female null modem cable:

1 & 6	-	4
2	-	3
3	-	2
4	-	1 & 6
5	-	5
7	-	8
8	-	7

## 1.4 Data streaming

The phase angle voltmeter and power meter modes have the option of high speed data streaming. In this operation, the window width for the measurement may be specified from 660us to 100ms and the data for each measurement window is transmitted over the communications in a continuous stream. The window is adjusted to synchronise to the measured frequency.

The instrument buffers the data and transmits at the fastest rate that is possible. The buffer depth is over 8000 data values so more than 5 seconds of data can be captured at the fastest rate of 1500 readings per second even if the data is not read at all. If the window size is such that the data can be read out in real time then data streaming can continue indefinitely.

Once the data streaming window has been setup but before the streaming has been started, the display periodically shows the measured value. Once streaming has been started, the display is blanked to minimise processing overheads. Streaming can be stopped either immediately (ABORT) or may be stopped but remaining data continues to be transmitted until the buffer is empty (STOP).

```
STREAM,ENABLE,0.01
START
read data
STOP
continue to read stored data
```

2 Communication commands

**\*CLS**

**\*CLS**

Function: Clear status

Description: Clears the *standard event status register*.

Format: \*CLS

Arguments: none

Reply: none

Example: \*CLS  
\*ESR?  
0

Notes:

**\*ESE**

**\*ESE**

Function: Set standard event status enable register.

Description: Enable which bits of the *standard event status register* set the ESB bit in the serial poll status byte..

Format: \*ESE, value

Arguments: decimal equivalent of bits in standard event status enable register

Reply: can be read by \*ESE?

Example: \*ESE, 60

Notes: The following bits in the standard event status enable register have been implemented:

- bit 0 OPC (operation complete)
- bit 2 QYE (unterminated query error)
- bit 3 DDE (device dependent error)
- bit 4 EXE (execution error)
- bit 5 CME (command interpretation error)
- bit 7 PON (power on event)

For example, \*ESE, 60 enables all the error bits so that the ESB bit in the serial poll status byte is set in the event of any error.

**\*ESR?**

**\*ESR?**

Function: Standard event status register query

Description: Returns the contents of the *standard event status register* and clears it.

Format: \*ESR?

Arguments: none

Reply: decimal equivalent of bits in standard event status register

Example: \*ESR?  
33

Notes: The following bits in the standard event status register have been implemented:

- bit 0 OPC (operation complete)
- bit 2 QYE (unterminated query error)
- bit 3 DDE (device dependent error)
- bit 4 EXE (execution error)
- bit 5 CME (command interpretation error)
- bit 7 PON (power on event)

For example, if a command is sent incorrectly and is not recognised, the CME bit will be set and the value of 33 will be returned.

**\*IDN?**

**\*IDN?**

Function: Identify query

Description: Returns a standard format identification string.

Format: \*IDN?

Arguments: none

Reply: An ASCII string in the IEEE488.2 format:  
manufacturer,model,serial no,version

Example: \*IDN?  
NEWTONS4TH,PSIMETRIQ,01234,1.00

Notes:

**\*OPC?**

**\*OPC?**

Function: Test for operation complete

Description: Returns 1 if previous operation is completed, 0 if not.

Format: \*OPC?

Arguments: none

Reply: 0 or 1

Example: START  
\*OPC?  
0  
\*OPC?  
0  
\*OPC?  
1

Notes: \*OPC? can be used to indicate when data is available or when a frequency sweep has completed.

**\*RST**

**\*RST**

Function: Reset

Description: Resets the instrument to the default state and clears the *standard event status register*.

Format: \*RST

Arguments: none

Reply: none

Example: \*RST

Notes: The \*RST command loads the default configuration. This is the same as loading the default configuration via the PROGRAM menu.

Any preceding setup commands will be overwritten.

\*RST should be followed by an end of line not a message separator. It may be helpful to follow it with a short pause to allow the new configuration to become active before sending further commands.

**\*SRE**

**\*SRE**

Function: Set service request enable register.

Description: Enable which bits of the *status byte register* initiate a service request.

Format: \*SRE, value

Arguments: decimal equivalent of bits in status byte register

Reply: can be read by \*SRE?

Example: \*SRE, 1  
generate a service request when data available.

Notes:

**\*SRE?**

**\*SRE?**

Function: Read service request enable register.

Description: Read back the present setting of the service request enable register.

Format: \*SRE?

Arguments:

Reply: decimal equivalent of bits in status byte register that would generate a service request.

Example: \*SRE?  
1

Notes:

**\*STB?**

**\*STB?**

Function: Read serial poll status byte

Description: Returns the decimal value of the serial poll status byte.

Format: \*STB?

Arguments: none

Reply: decimal value of the serial poll status byte

Example: \*STB?  
1

Notes: The following bits in the serial poll status register have been implemented:

bit 0 RDV (results data available)  
bit 1 SDV (sweep data available)  
bit 3 ALA (alarm active)  
bit 4 MAV (message available)  
bit 5 ESB (standard event summary bit)

**\*TRG**

**\*TRG**

Function: Trigger

Description: Initiates a new measurement, resets the ranging and filtering.

Format: \*TRG

Arguments: none

Reply: none

Example: MODE,VRMS  
\*TRG  
VRMS?

Notes:

**\*TST?**

**\*TST?**

Function: Self test query

Description: Returns the results of self test

Format: \*TST?

Arguments: none

Reply: single integer  
bit 0 – set if uncalibrated  
bit 1 – set if error with analogue zero  
> 15 – major system error

Example: \*TST?  
0

Notes:

**\*WAI**

**\*WAI**

Function: Wait for operation complete

Description: Suspends communication until the previous operation has completed

Format: \*WAI

Arguments: none

Reply: none

Example: GAINPH  
START  
\*WAI  
GAINPH,SWEEP?

Notes: In the example, the query command GAINPH,SWEEP? can be sent immediately after the \*WAI command and the sweep data will be returned as soon as the sweep has completed.

**ABORT**

**ABORT**

Function: Abort sweep

Description: Abort an active sweep, or data streaming.

Format: ABORT

Arguments: none

Reply: none

Example: MODE,PHASE,STREAM,0.01  
START  
*read data values as required*  
ABORT

Notes: Any remaining values held in the buffer will be discarded.

**ACTRIM**

**ACTRIM**

Function: Set ac control parameters

Description: Sets the specified signal level, tolerance and input channel. for the ac control (amplitude compression).

Format: *ACTRIM,channel,level,tolerance*

Arguments: channel:  
          DISABL  
          CH1  
          CH2  
          level:  
              required ac level in V or A or dBm  
          tolerance:  
              required accuracy in percent

Reply: none

Example: ACTRIM,CH1,1.0,5       (1.0V, 5%)

Notes: The level should be set in dBm if dBm mode is selected (OUTPUT,DBM)

It is not necessary to send all the arguments but those that are sent must be in the correct sequence.

**ALARM**

**ALARM**

Function: Set common controls for alarm1 and alarm2.

Description: Set the alarm latch and sounder control.

Format: *ALARM,latch,sounder*

Arguments: latch:  
              ON  
              OFF  
              sounder:  
              ENABLED  
              DISABLED

Reply: none

Example: ALARM,ON,DISABLED

Notes:

**ALARM?**

**ALARM?**

Function: Read alarm status.

Description: Reads the status of the measurements and 2 alarms.

Format: ALARM?

Arguments: none

Reply: single integer  
bit 0 data available  
bit 1 data error  
bit 2 alarm 1  
bit 3 alarm 2

Example: ALARM?  
1

Notes: An alarm is present if bit 0 is high (data is available) and either alarm 1 or alarm 2 bits are high.

**ALARM1**

**ALARM1**

Function: Set parameters for alarm1.

Description: Set alarm1 type and thresholds.

Format: *ALARM1 , type, data, high, low*

Arguments: type:  
            DISABLED  
            HIGH  
            LOW  
            INSIDE  
            OUTSIDE  
            LINEAR  
data  
            1-4  
high:  
            high threshold  
low:  
            low threshold

Reply: none

Example: *ALARM1,HIGH,1,2,0*

Notes: Both thresholds must be sent even if only one is used.

**ALARM2**

**ALARM2**

Function: Set parameters for alarm2.

Description: Set alarm2 type and thresholds.

Format: *ALARM2, type, data, high, low*

Arguments: type:  
          DISABLED  
          HIGH  
          LOW  
          INSIDE  
          OUTSIDE  
data  
          1-4 for zoom data  
high:  
          high threshold  
low:  
          low threshold

Reply: None

Example: *ALARM2,LOW,3,0,0.5*

Notes: Both thresholds must be sent even if only one is used.  
There is no LINEAR option for alarm 2.

**ALARME**

**ALARME**

Function: Set alarm status enable register

Description: Sets bits in the alarm status enable register to control which alarm bit if any set the alarm active bits in the status byte.

Format: *ALARME, value*

Arguments: decimal equivalent of alarm bits  
bit2 set bit 3 of status byte when alarm 1 is active  
bit3 set bit 3 of status byte when alarm 2 is active

Reply: none

Example: *ALARME, 12*  
*\*SRE,8*  
set bit 3 in status byte when either alarm 1 or alarm 2 is active and generate a service request

Notes: default value is 0

**ALARME?**

**ALARME?**

Function: Read alarm status enable register

Description: Read back present bits in the alarm status enable register which controls the alarm active bit in the status byte.

Format: ALARME?

Arguments: none

Reply: decimal equivalent of alarm bits

Example: ALARME?  
12

Notes:

**AMPLIT**

**AMPLIT**

Function: Set output amplitude

Description: Sets the output amplitude in Volts or dBm for the generator.

Format: *AMPLIT,amplitude*

Arguments: peak amplitude in Volts or amplitude in dBm

Reply: none

Example: *AMPLIT,0.5* (set peak amplitude to 0.5V)

Notes: dBm mode is selected by *OUTPUT,DBM*

**ANALOG**

**ANALOG**

Function: Set up analogue output

Description: Sets the scaling of the analogue output or a constant value.

Format: *ANALOG,MONITOR,zero,scale*  
*ANALOG,MANUAL,value*

Arguments: zero level for monitored value  
full scale for monitored value  
value between 0 and 1 for manual

Reply: none

Example: *ANALOG,MONITOR,*

Notes: Analog output given by:  
 $\text{fraction} = (\text{value} - \text{zero}) / \text{full scale}$

**BANDWI**

**BANDWI**

**PSM1735 NumetriQ only**

Function: Select bandwidth or selective (heterodyning) measurements.

Description: Selective measurement automatically starts at around 10kHz for those functions that support it. It can be disabled by forcing the bandwidth to "wide". For low noise measurements at low frequency the bandwidth can be restricted to "low".

Format: BANDWI, *type*

Arguments: type:  
          AUTO  
          WIDE  
          LOW

Reply: none

Example: BANDWI,WIDE

Notes: In wide bandwidth mode the frequency range is limited to 1MHz.  
In low bandwidth mode, the frequency is restricted to 30kHz

**BEEP**

**BEEP**

Function: Sound the buzzer  
Description: Makes a "beep" from the instrument.  
Format: BEEP  
Arguments: none  
Reply: none  
Example: BEEP  
Notes:

**BLANKI**

**BLANKI**

Function: Select blanking

Description: Enable or disable low value blanking.

Format: *BLANKI, value, threshold*

Arguments: value:  
          ON  
          OFF  
          threshold:  
          threshold in appropriate units

Reply: none

Example: BLANKI,OFF  
          BLANKI,ON,-35

Notes: It is not necessary to send the threshold.  
Not all functions have a threshold option,  
it will only be used if in an appropriate  
mode.

**CONFIG**

**CONFIG**

Function: Direct access of configuration parameters

Description: Sets configuration parameter for which there may not be a direct command.

Format: *CONFIG,index,data*

Arguments: index is the number of the parameter  
data is the data for that parameter

Reply: none

Example: CONFIG,6,1 (set phase convention)

Notes: The list of configurable parameters is given in the appendix.  
CONFIG goes through the same limit checking as when entering data from the menus.

**CONFIG?**

**CONFIG?**

Function: Configurable parameter query

Description: Reads the present value of a single parameter.

Format: *CONFIG,index?*  
or: *CONFIG?index*

Arguments: index is the parameter number

Reply: Value of parameter, real or integer as appropriate.

Example: *CONFIG,6?* (read phase convention)  
0  
*CONFIG,6,1*  
*CONFIG,6?*  
1

Notes: The list of configurable parameters is given in the appendix.

**COUPLI**

**COUPLI**

Function: Set ac or dc coupling.

Description: Selects the input coupling for a given input channel.

Format: COUPLI,*channel,coupling*

Arguments: channel:  
            CH1  
            CH2  
            coupling:  
            AC+DC  
            ACONLY

Reply: none

Example: COUPLI,CH2,AC+DC

Notes:

**DATALO**

**DATALO**

Function: Set up datalog

Description: Sets datalog parameters or accesses datalog non-volatile store.

Format: *DATALO,function,interval*

Arguments: function:  
              DISABLE  
              RAM  
              NONVOL  
              RECALL  
              DELETE  
              interval:  
              datalog interval in seconds

Reply: none

Example: DATALOG, NONVOL, 10

Notes:

**DATALO?**

**DATALO?**

Function: Read back datalog results

Description: Return datalog values, one record per line

Format: *DATALO,start,records*

Arguments: start:  
                   first record to return  
 records:  
                   number of records to return

Reply: 3 to 6 data values depending on settings:  
           index 1-n  
           elapsed time in hours  
           data1  
           data2 (if stored)  
           data3 (if stored)  
           data4 (if stored)  
           one record per line

Example: DATALOG, NONVOL, 36  
 START  
 wait for datalog  
 STOP  
 DATALOG, 20, 4?  
 20, 1.9000E-1, 1.2345E0  
 21, 2.0000E-1, 1.2345E0  
 22, 2.1000E-1, 5.6789E3  
 23, 2.2000E-1, 1.2345E0

Notes: if no arguments are sent then DATALOG?  
 returns all data in the same format

**DAV?**

**DAV?**

Function: Data available query  
Description: Returns data availability status.  
Format: DAV?  
Arguments: none  
Reply: Decimal equivalent of data available bits:  
bit0 new data available  
bit1 data available  
bit2 new full sweep data available  
bit3 sweep data available  
bit4 streaming data available  
bit5 more streaming data to come  
bit6 integration data available  
bit7 datalog data available

Example: START (trigger sweep)  
DAV?  
0  
DAV?  
11 (first data available)  
DAV?  
11  
DAV?  
11  
DAV?  
15 (full sweep data available)

Notes: DAV? does not modify the status bits.

**DAVER**

**DAVER**

Function: Set data available enable register

Description: Sets bits in the data available enable register to control which status bits set the data available bits in the status byte.

Format: DAVER,value

Arguments: decimal equivalent of data available bits  
 bit0 set bit 0 of status byte when new data available  
 bit1 set bit 0 of status byte when data available  
 bit2 set bit 1 of status byte when new full sweep data available  
 bit3 set bit 1 of status byte when sweep data available  
 bit4 set bit 2 of status byte when streaming data available  
 bit5 set bit 2 of status byte if more streaming data is to come

Reply: none

Example: DAVER, 4  
 set bit 1 in status byte only when full sweep data is ready

Notes: default value is 6:  
 bit 0 of status byte is set whenever data is available  
 bit 1 of status byte is set when full sweep data is available.

**DAVER?**

**DAVER?**

Function: Read data available enable register

Description: Read back present setting of the data available enable register, which controls the status bits that set the data available bits in the status byte.

Format: DAVER?

Arguments: none

Reply: decimal equivalent of bits

Example: DAVER?  
4

Notes:

**FILTER**

**FILTER**

Function: Select the filtering

Description: Sets the filter time constant and dynamic response.

Format: *FILTER, type, dynamics*

Arguments: type:  
              NONE  
              NORMAL  
              SLOW  
              dynamics:  
              AUTO  
              FIXED

Reply: none

Example: FILTER,NORMAL,FIXED  
          FILTER,NONE

Notes: It is not necessary to send both parameters if it is only required to set the type. Both arguments must be sent to set the dynamics.

**FRA**

**FRA**

Function: Set frequency response analyser mode.

Description: Set frequency response analyser mode.

Format: FRA

Arguments:

Reply: none

Example: FRA

Notes: This command has the same effect as MODE,GAINPH.  
FRA, GAINPH, TFA are aliases for the same command.

**FRA?**

**FRA?**

Function: frequency response analyser query

Description: Read frequency response analyser results.  
Sets frequency response analyser mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: FRA?  
or: FRA?SWEEP  
or: FRA,SWEEP?

Arguments: none, or SWEEP

Reply: 6 data values separated by commas  
freq,mag1,mag2,db,phase,delay  
one line per result for sweep data

Example: OUTPUT,ON  
FRA  
FSWEEP,20,10,20E3  
START  
DAV?  
3  
DAV?  
15  
FRA?SWEEP  
data returned

Notes: FRA? waits for next unread data.  
FRA?SWEEP does not wait for new data – data can be read multiple times.  
FRA, GAINPH, TFA are aliases for the same command

**FREQUE**

**FREQUE**

Function: Set the output frequency

Description: Sets the generator output frequency in Hz.

Format: `FREQUE, frequency`

Arguments: frequency in Hz

Reply: none

Example: `FREQUE,5e4` (set frequency to 50kHz)

Notes:

**FSWEEP****FSWEEP**

Function:	Set the frequency sweep parameters
Description:	Sets the start frequency in Hz, the end frequency, the number of steps and log/linear for the selected function.
Format:	<i>FSWEEP, steps, start, end, type</i>
Arguments:	steps: number of steps start: start frequency in Hz end: end frequency in Hz type: LOGARI LINEAR
Reply:	none
Example:	MODE,GAINPH FSWEEP,50,1000,1e6 (set 50 steps between 1kHz and 1MHz)
Notes:	It is not necessary to send all the arguments, but if they must be in the specified order. The same command is used for all the functions – the data is applied to whichever function has been selected if valid (such as gain/phase analyser). If the selected mode is not valid (such as rms), then the command is ignored and an execution error is flagged in the standard event status register, sesr.

**GAINPH**

**GAINPH**

Function: Set gain/phase analyser mode.

Description: Set gain/phase analyser mode.

Format: GAINPH

Arguments:

Reply: none

Example: GAINPH

Notes: This command has the same effect as MODE,GAINPH.  
FRA, GAINPH, TFA are aliases for the same command.

**GAINPH?**

**GAINPH?**

Function: Gain/phase query

Description: Read gain/phase analyser results.  
Sets gain/phase analyser mode if not already set.  
Waits for next unread data if necessary .  
Clears new data available bit read by DAV?

Format: GAINPH?  
or: GAINPH?SWEEP  
or: GAINPH,SWEEP?

Arguments: none, or SWEEP

Reply: 6 data values separated by commas  
freq,mag1,mag2,db,phase,delay  
one line per result for sweep data

Example: OUTPUT,ON  
GAINPH  
FSWEEP,20,10,20E3  
START  
DAV?  
3  
DAV?  
15  
GAINPH?SWEEP  
data returned

Notes: GAINPH? waits for next unread data.  
GAINPH?SWEEP does not wait for new data – data can be read multiple times.

**HARMON**

**HARMON**

Function: Set harmonic analyser mode.

Description: Set harmonic analyser mode and parameters.

Format: *HARMON,scan,parameter,harmonic,max*

Arguments: scan:  
          SINGLE  
          THDD  
          THDS  
          parameter:  
          PERCEN  
          DB  
          harmonic:  
          single harmonic 2-50 for display  
          max:  
          harmonic series 2-50 for series thd

Reply: none

Example: *HARMON,SINGLE,PERCEN,3*

Notes: It is not necessary to send any arguments, but if any are sent they must be in the specified order.

## HARMON?

## HARMON?

Function: Harmonic analyser query

Description: Read harmonic results.  
Sets harmonic analyser mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: HARMON?  
or: HARMON,SWEEP?  
or: HARMON,SERIES?

Arguments: none, or SWEEP, or SERIES

Reply: 7 data values separated by commas:  
single: freq,mag1,mag2,h1,h2,harm1,harm2  
thd: freq,mag1,mag2,thd1,thd2,harm1,harm2  
1 line per result for sweep data  
series: 6 data values separated by commas:  
mag1,%1,phase1,mag2,%2,phase2

Example: HARMON?  
data returned

Notes: HARMON? waits for next unread data.  
HARMON?SWEEP does not wait for new data – data can be read multiple times.

## INPUT

## INPUT

Function: Set input mode

Description: Selects the input type of the instrument

Format: `INPUT,channel,type`

Arguments: channel:  
            CH1  
            CH2  
          type:  
            DISABLE  
            VOLTAGE  
            SHUNT

Reply: none

Example: `INPUT,CH1,SHUNT`

Notes:

**INTYPE**

**INTYPE**

**PSM1735 NumetriQ only**

Function: Set input connection

Description: Selects the input type for each channel to be non-inverting, inverting, or differential.

Format: *INPUT,channel,type*

Arguments: channel:  
          CH1  
          CH2  
          type:  
          MAIN  
          SECOND  
          DIFFER

Reply: none

Example: INPUT,CH1,DIFFERENTIAL

Notes:

**KEYBOA**

**KEYBOA**

Function: Disable front panel keyboard.

Description: The front panel keyboard can be disabled to prevent accidental operation.

Format: `KEYBOARD, value`

Arguments: value:  
          ENABLE  
          DISABLE

Reply: none

Example: `KEYBOARD,DISABLE`

Notes: The keyboard can be re-enabled from the front panel only by pressing the HOME key.

**LCR**

**LCR**

Function: Set LCR meter mode.

Description: Set LCR mode and conditions.

Format: *LCR,conditions,parameter,head*

Arguments: conditions:  
          AUTO  
          MANUAL  
          parameter:  
          AUTO  
          CAPACITANCE  
          INDUCTANCE  
          IMPEDANCE  
          head:  
          NONE  
          NORMAL  
          HIGH  
          VHIGH

Reply: none

Example: LCR,AUTO,IMPEDA,NORMAL

Notes: It is not necessary to send any arguments, but if any are sent they must be in the specified order.

**LCR?**

**LCR?**

Function: LCR meter query

Description: Read LCR meter results.  
Sets LCR meter mode if not already set.  
Waits for next unread data if necessary .  
Clears new data available bit read by DAV?

Format: LCR?  
or: LCR?SWEEP  
or: LCR,SWEEP?

Arguments: none, or SWEEP

Reply: 14 data values separated by commas:  
freq, mag1, mag2, impedance,  
phase, series R, series C, series L,  
//R, //C, //L, tan $\delta$ , Q, reactance

sweep reply: 8 data values per line per sweep result:  
freq,Q,tan $\delta$ ,impedance,phase,L,C,R

Example: OUTPUT,ON  
LCR?  
data returned

Notes: LCR? waits for next unread data.  
LCR?SWEEP does not wait for new data –  
data can be read multiple times.

**LOWFRE**

**LOWFRE**

Function: Set low frequency mode

Description: Sets the low frequency option for external frequency measurement.

Format: LOWFRE, *value*

Arguments: value:  
            ON  
            OFF

Reply: none

Example: LOWFRE,ON

Notes: LOWFRE is mainly used for measuring low frequencies when not using the instrument generator for the frequency reference. However, as it applies digital filtering, it may also be useful when analysing any signals below a few hundred Hertz.

**MARKER**

**MARKER**

Function: Set frequency marker

Description: Enable or disable frequency marker.

Format: *MARKER, value, frequency*

Arguments: value:  
          ON  
          OFF  
          frequency:  
          marker frequency in Hz

Reply: none

Example: *MARKER,OFF*  
*MARKER,ON,25e3*

Notes: It is not necessary to send the frequency when enabling the marker if it has already been set.

**MODE**

**MODE**

Function: Set mode

Description: Sets the fundamental operating mode of the instrument.

Format: *MODE, type*

Arguments: type:  
           SIGGEN (signal generator only)  
           VRMS (rms voltmeter)  
           GAINPH (gain/phase analyser)  
           VECTOR (vector voltmeter)  
           POWER (power meter)  
           LCR (LCR meter)  
           HARMON (harmonic analyser)  
           TXA (transformer analyser)

Reply: none

Example: *MODE,GAINPH*

Notes: MODE sets the measurement mode of the instrument

**OFFSET**

**OFFSET**

Function: Set the output offset  
Description: Sets the output generator offset in Volts.  
Format: `OFFSET,offset`  
Arguments: offset in Volts  
Reply: none  
Example: `OFFSET,5e-3` (set offset to 5mV)  
Notes:

**OUTPUT**

**OUTPUT**

Function: Set output

Description: Turns the output on or off, or sets the level mode to dBm or voltage

Format: *OUTPUT,command*

Arguments: command:  
OFF  
ON  
VOLT  
DBM

Reply: none

Example: OUTPUT,ON

Notes: For safety, the output defaults to off and must be turned on explicitly.

**PAV**

**PAV**

Function: Set phase angle voltmeter mode.

Description: Set phase angle voltmeter mode and parameter.

Format: *PAV,parameter,lvdt scale*

Arguments: parameter:  
                   INPHAS  
                   QUADR  
                   TANPHI  
                   MAGNIT  
                   POLAR  
                   A2/1  
                   RMS2  
                   RMS2/1  
                   LVDT-D  
                   LVDT-R  
           lvdt scale:  
                   scale factor in m for lvdt applications

Reply: none

Example: PAV,LVDT-D,0.1

Notes: It is not necessary to send any arguments, but those that are sent must be in the specified order.  
       PAV and VECTOR are aliases for the same command.

**PAV?**

**PAV?**

Function: Phase angle voltmeter query

Description: Read phase angle voltmeter results.  
Sets phase angle voltmeter mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: PAV?  
or: PAV?SWEEP  
or: PAV,SWEEP?

Arguments: none, or SWEEP

Reply: 7 data values separated by commas:  
freq,mag1,mag2,parameter,phase,a,b  
1 line per result for sweep data

Example: `FREQ,3300`  
`OUTPUT,ON`  
`PAV?LVDT_D,0.1`  
data returned

Notes: PAV? waits for next unread data.  
PAV?SWEEP does not wait for new data – data can be read multiple times.  
PAV and VECTOR are aliases for the same command.

**PHASE**

**PHASE**

Function: Set phase meter mode.

Description: Select phase meter mode.

Format: PHASE, *mode*, *window size*

Arguments: mode:  
              NORMAL  
              STREAM  
window size  
              streaming window size in seconds

Reply: none

Example: PHASE

Notes: It is not necessary to send any arguments but if they are sent they must be in the specified order.

**PHASE?**

**PHASE?**

Function: Phase meter query

Description: Reads phase meter results.  
Sets phase meter mode if not already set.  
Waits for next unread data if available.  
Clears new data available bit read by DAV?

Format: PHASE?

Arguments: none

Reply: 2 data values separated by commas  
freq,phase

Example: PHASE?  
1.8396E2,5.0342E-2,

Notes: The phase convention can be set to 0° to -360°, 0° to +360°, or +180° to -180° in the SYSTEM menu or using CONFIG,6,0-2 (see appendix) .

**PHCONV**

**PHCONV**

Function: Set phase convention

Description: Set phase convention

Format: PHCONV, *convention*

Arguments: convention:  
180: -180 to +180  
-360: 0 to -360  
+360: 0 to +360

Reply: none

Example: PHCONV, -360

Notes:

**PHREF**

**PHREF**

Function: Set phase reference

Description: Select measurement of phase as CH2 relative to CH1 or as CH1 relative to CH2

Format: PHREF, *channel*

Arguments: channel:  
CH1: phase = ch2 wrt ch1  
CH2: phase = ch1 wrt ch2

Reply: none

Example: PHREF, CH2

Notes: This parameter influences the phase meter mode and the phase angle voltmeter mode

**POWER**

**POWER**

Function: Set up power meter mode.

Description: Configure power meter with integration type

Format: *POWER, integration type*

Arguments: integration type:  
MAGNITUDE  
SIGNED

Reply: none

Examples: POWER,SIGNED  
POWER

Notes: It is not necessary to send the integration type argument.

**POWER?**

**POWER?**

Function: Read power meter results

Description: Reads back latest power meter results.  
Sets power meter mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: POWER?  
or: POWER?*results*  
or: POWER,*results?*

Arguments: results:  
          WATTS  
          RMS  
          INTEGR

Reply: WATTS:  
          9 data values separated by commas  
          W,W.f,VA,VA.f,pf,pf.f,Wdc,W.h,freq  
RMS:  
          8 data values separated by commas  
          rms1,2,dc1,2,f1,2,phase1,2  
INTEGR:  
          9 data values separated by commas  
          Wh,Wh.f,VAh,VAh.f,avpf,avpf.f,  
          Ah,Ah.f,time  
no argument:  
          26 data values separated by commas  
          WATTS, RMS, INTEGR

Example: POWER?WATTS

Notes:

**PPORT**

**PPORT**

Function: Set the parallel port

Description: Force the logic level on the parallel port data lines

Format: PPORT, *value*

Arguments: decimal value to be written to the port

Reply: None

Example: PPORT,64 {set data bit 6, clear others}

Notes: The parallel port may be used as an 8 bit logic level output port and a 4 bit logic level input port.

**PPORT?**

**PPORT?**

Function: Read the parallel port

Description: Read the logic level on the parallel port control input lines

Format: PPORT?

Arguments: None

Reply: Single integer data value

Example: PPORT?  
12

Notes: The parallel port may be used as a 4 bit logic level input port and an 8 bit logic level output port.

**PROGRA**

**PROGRA**

Function: Access non volatile program stores.

Description: Recall, store or delete non-volatile program store.

Format: *PROGRA,function,number*

Arguments: function:  
              RECALL  
              STORE  
              DELETE  
              number  
              0-100

Reply: none

Example: *PROGRA,RECALL,13*

Notes: Number 0 represents factory default, which can only be recalled.

**PROGRA?**

**PROGRA?**

Function: Identify current program.

Description: Reads the name of the last program to be loaded or recalled.

Format: PROGRA?

Arguments: none

Reply: text string

Example: PROGRA?  
factory default

Notes:

**RANGE**

**RANGE**

Function: Set channel ranging.

Description: Select minimum range and range control for a given input channel.

Format: *RANGE,channel,ranging,range*

Arguments: channel:  
          CH1  
          CH2  
          ranging:  
          AUTO  
          UPAUTO  
          MANUAL  
          range:  
          nominal range value

Reply: none

Example: RANGE,CH2,MANUAL,3V

Notes:

**RESOLU**

**RESOLU**

Function: Set the data resolution

Description: Data is returned in scientific format with exponent and mantissa. The resolution of the mantissa may be selected to be 5 digit (NORMAL) or 6 digit (HIGH).

Format: RESOLU,*format*

Arguments: format:  
                   NORMAL          (5 digit mantissa)  
                   HIGH            (6 digit mantissa)

Reply: none

Example: RESOLU,HIGH

Notes: Data format for NORMAL is:  
           [-]1.2345E[-]00  
       Data format for HIGH is:  
           [-]1.23456E[-]00  
       The sign of the mantissa and exponent are only sent if negative shown as [-] in the above examples

**REZERO**

**REZERO**

Function: Rezero front end

Description: Request the DSP to re-compensate for dc offset and compute a new autozero

Format: REZERO

Arguments: none

Reply: none

Example: REZERO

Notes:

**SCALE**

**SCALE**

Function: Set channel scale factor.

Description: Set a multiplying scale factor for a given input channel.

Format: *SCALE,channel,factor*

Arguments: channel:  
            CH1  
            CH2  
            factor:  
            multiplying scale factor

Reply: none

Example: SCALE,CH2,10

Notes:

## SHUNT

## SHUNT

Function: Set channel shunt value

Description: Set the resistance factor of a current shunt to be divided into the measured voltage for a given input channel.

Format: SHUNT,*channel,resistance*

Arguments: channel:  
          CH1  
          CH2  
          resistance:  
          shunt resistance in Ohms

Reply: none

Example: SHUNT,CH1,10

Notes: The SHUNT command is still accepted if the channel has not been configured for current. The value stored will be used when the channel is configured for current.

**SPEED**

**SPEED**

Function: Sets the measurement speed

Description: Sets the minimum window size for the measurement.

Format: *SPEED, value*

Arguments: value:  
FAST  
MEDIUM  
SLOW

Reply: none

Example: *SPEED, SLOW*

Notes:

**SSWEEP**

**SSWEEP**

Function: Access non volatile sweep results stores.

Description: Recall, store or delete non-volatile sweep results store.

Format: *SSWEEP,function,number*

Arguments: function:  
                  RECALL  
                  STORE  
                  DELETE  
          number  
          1-30

Reply: none

Example: *SSWEEP,RECALL,13*

Notes: The sweep data can be read back using the sweep query command for each mode eg. *FRA,SWEEP?* for an FRA sweep.

**START**

**START**

Function: Start sweep

Description: Initiate sweep in those functions that have a sweep or resets filtering in others.

Format: START

Arguments: none

Reply: none

Example: MODE,GAINPH (set gain phase analyser)  
START

Notes:

**STATUS?**

**STATUS?**

Function: Read back channel ranging status.

Description: Read back condition of selected channel:  
range number (1-16)  
range text  
overflow/underflow status

Format: *STATUS,channel?*  
or: *STATUS?channel*

Arguments: channel:  
CH1  
CH2

Reply: range number,range text,over/under/ok  
1-16  
range as per RANGE command  
OVER if overflow  
LOW if underflow  
OK if in range

Example: STATUS,CH1?  
6,3V,OK

Notes:

**STOP**

**STOP**

Function: Stop sweep  
Description: Stop an active sweep, or data streaming.  
Format: STOP  
Arguments: none  
Reply: none  
Example: MODE,PHASE,STREAM,0.01  
START  
*read data values as required*  
STOP  
*read remaining data values*

Notes:

**STREAM**

**STREAM**

Function: Set data streaming mode

Description: Set instrument ready for data streaming with specified window

Format: *STREAM,control,window*

Arguments: control  
          ENABLE  
          DISABL  
          window size  
          streaming window size in seconds

Reply: none

Example: PAV,TANPHI  
          STREAM,ENABLE,0.01  
          START  
          *read data values as required*  
          STOP  
          *read remaining data values*

Notes: Data streaming is valid for phase meter and phase angle voltmeter (vector voltmeter) modes.

**STREAM?**

**STREAM?**

Function: Start to read streaming data

Description: Start to read streaming data

Format: STREAM?  
STREAM,max?  
STREAM?max

Arguments: max  
                  maximum number of values  
          none  
                  return all data

Reply: data stream separated by commas

Example: PAV,TANPHI  
          STREAM,ENABLE,0.01  
          START  
          STREAM?  
          data,data,data,data, .....

Notes: This command is only used with IEEE488  
(GPIB) - streaming starts immediately  
with RS232.

**TFA**

**TFA**

Function: Set transfer function analyser mode.  
Description: Set transfer function analyser mode.  
Format: TFA  
Arguments:  
Reply: none  
Example: TFA  
Notes: This command has the same effect as  
MODE,GAINPH.  
FRA, GAINPH, TFA are aliases for the  
same command.

**TFA?**

**TFA?**

Function: transfer function analyser query

Description: Read transfer function analyser results.  
Sets transfer function analyser mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: TFA?  
or: TFA?SWEEP  
or: TFA,SWEEP?

Arguments: none, or SWEEP

Reply: 6 data values separated by commas  
freq,mag1,mag2,db,phase,delay  
one line per result for sweep data

Example: OUTPUT,ON  
TFA  
FSWEEP,20,10,20E3  
START  
DAV?  
3  
DAV?  
15  
TFA?SWEEP  
data returned

Notes: TFA? waits for next unread data.  
TFA?SWEEP does not wait for new data – data can be read multiple times.  
FRA, GAINPH, TFA are aliases for the same command

**TXA**

**TXA**

Function: Set transformer analyser mode.

Description: Set transformer analyser mode, test, fixture, and resistances

Format: *TXA, test, fixture, source, load*

Arguments: test:  
           TXTR  
           TXL  
           TXLL  
           TXACR  
           TXDCR  
           TXIWC  
           TXMAGI  
           TXRLOS  
           TXILOS  
           TXHARM  
           TXTHD  
           TXLBAL  
 fixture:  
           NONE  
           TAF01  
           TAF02  
 source resistance  
 load resistance

Reply: none

Example: TXA, TXILOS, TAF01, 135, 25.4

Notes: It is not necessary to send any arguments, but they must be in the specified order.

**TXA?**

**TXA?**

Function: Transformer analyser query

Description: Read transformer analyser results.  
 Sets transformer analyser mode if not already set.  
 Waits for next unread data if necessary .  
 Clears new data available bit read by DAV?

Format: TXA?  
 or: TXA?SWEEP  
 or: TXA,SWEEP?

Arguments: none, or SWEEP

Reply: 3 or 4 data values separated by commas:  
           txdcr                  freq,mag1 ,mag2,parameter  
           txmagi                dc1,dc2,resistance  
                                   freq,rms1 ,rms2,watts

sweep reply: Data as above, one line per sweep result.

Example: TXA?  
 data returned

Notes: TXA? waits for next unread data.  
 TXA?SWEEP does not wait for new data – data can be read multiple times.

**TXTEST**

**TXTEST**

Function: Set transformer analyser test.

Description: Set transformer analyser test and winding(s).

Format: TXTEST, *test*, *winding1*, *winding2*

Arguments: test:  
                   *as TXA command*  
 winding1:  
                   W1  
                   W2  
                   W3  
                   W4                   (TAF02 only)  
                   W2+3               (TAF01 only)  
 winding2:                   (turns ratio test only)  
                   W1:W2  
                   W2:W1  
                   W1:W3  
                   W3:W1  
                   W2:W3  
                   W3:W2  
                   W1:W4               (TAF02 only)  
                   W4:W1               (TAF02 only)  
                   W1:2+3             (TAF01 only)  
                   W2+3:1             (TAF01 only)

Reply: none

Example: TXTEST, TXTR, W1, W2:W3

Notes: It is not necessary to send all the arguments, but they must be in the specified order.

**USER?**

**USER?**

Function: Read the user data  
Description: Returns up to 3 lines of user data  
Format: USER?  
Arguments: none  
Reply: 3 lines of ASCII terminated by CR  
Example: USER?  
          Newtons4th Ltd  
          R&D department  
          PsimetriQ #4

Notes:

**VECTOR**

**VECTOR**

Function: Set vector voltmeter mode.

Description: Set vector voltmeter mode and parameter.

Format: *VECTOR,parameter,lvdt scale*

Arguments: parameter:  
 NONE  
 INPHAS  
 QUADR  
 TANPHI  
 POLAR  
 A2/A1  
 RMS2/1  
 LVDT-D  
 LVDT-R  
 lvdt scale:  
 scale factor in m for lvdt applications

Reply: none

Example: VECTOR,LVDT-D,0.1

Notes: It is not necessary to send any arguments, but those that are sent must be in the specified order.  
 PAV and VECTOR are aliases for the same command.

**VECTOR?**

**VECTOR?**

Function: Vector voltmeter query

Description: Read vector voltmeter results.  
Sets vector voltmeter mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: VECTOR?  
or: VECTOR?SWEEP  
or: VECTOR,SWEEP?

Arguments: none, or SWEEP

Reply: 7 data values separated by commas:  
freq,mag1,mag2,parameter,phase,a,b  
1 line per result for sweep data

Example: `FREQ,3300`  
`OUTPUT,ON`  
`VECTOR?LVDT_D,0.1`  
data returned

Notes: VECTOR? waits for next unread data.  
VECTOR?SWEEP does not wait for new data – data can be read multiple times.  
PAV and VECTOR are aliases for the same command.

**VERSIO?**

**VERSIO?**

Function: Read the instrument code versions.

Description: Returns an ASCII string with the details of the various parts of the instrument firmware.

Format: VERSIO?

Arguments: none

Reply: date code, type, cpu, dsp, fpga, boot

Examples: VERSION?  
PQ3504,1,1.21,1.21,1.21,1.02

Notes: This data can be displayed on the screen by pressing SYSTEM then BACK

**VRMS**

**VRMS**

Function: Set up rms voltmeter.  
Description: Set mode to rms voltmeter.  
Format: VRMS  
Arguments: none  
Reply: none  
Examples: VRMS  
Notes: This has the same effect as MODE,VRMS

**VRMS?**

**VRMS?**

Function: Read true rms voltmeter results

Description: Reads back latest voltmeter results.  
Sets voltmeter mode if not already set.  
Waits for next unread data if necessary.  
Clears new data available bit read by DAV?

Format: VRMS?  
or: VRMS,*results?*  
or: VRMS?*results*

Arguments: results:  
RMS  
SURGE

Reply: RMS:  
8 data values separated by commas  
rms1,2,dc1,2,ac1,2,dbm1,2  
SURGE:  
6 data values separated by commas  
pk1,2,cf1,2,surge1,2  
no argument:  
14 data values separated by commas  
RMS results then SURGE

Example: VRMS?RMS

Notes: As VRMS? does not send the same data twice but waits instead for the next result, it is not necessary to check the data available bits before sending the VRMS? command.

**WAVEFO**

**WAVEFO**

Function: Set the output waveform

Description: Selects the output waveform for the signal generator.

Format: WAVEFO, *type*

Arguments: type:  
                   SINEWA           (sine wave)  
                   TRIANG          (triangle wave)  
                   SQUARE          (square wave)  
                   LEADIN          (leading sawtooth)  
                   TRAILI          (trailing sawtooth)

Reply: None

Example: FREQUE,500  
           WAVEFO,TRIANG   (triangle wave)  
           OUTPUT,ON

Notes: PSM1735 NumetriQ only has SINEWA or SQUARE option

**ZERO**

**ZERO**

Function: Apply or remove the zero

Description: Applies or removes a zero function depending on the measurement mode (same as pressing ZERO key).

Format: ZERO  
ZERO,DELETE

Arguments: none

Reply: none

Example: ZERO

Notes:

**ZOOM**

**ZOOM**

Function: Sets the display zoom parameters.

Description: Sets the zoom level and data.

Format: *ZOOM,level,data1,data2,data3,data4*

Arguments: level:  
           0 – normal  
           1 – 2 line display (zoom level 1)  
           2 – single line display (zoom level 2)  
 data1:  
           first data (zoom level 1)  
           or data for single line (zoom level 2)  
 data2-4:  
           other data (zoom level 1)

          data consists of line number for channel 1  
           or line number + 128 for channel 2

Reply: None

Example: VRMS  
           ZOOM,1,1,12 (level 1, ch1 rms, ch2 rms)

Notes: It is not necessary to send all the parameters, but whatever parameters are sent must be in the correct order.

**ZOOM?**

**ZOOM?**

Function: Read the display zoom parameters.

Description: Reads the zoom level and data.

Format: ZOOM?

Arguments:

Reply: 5 integers separated by commas:

level:

0 – normal

1 – 2-4 value display (zoom level 1)

2 – single line display (zoom level 2)

data1-4:

zoom data

data consists of line number for channel 1  
or line number + 128 for channel 2

Example:

ZOOM?

1,1,129,0,0 (level 1, ch1 rms, ch2 rms)

Notes:



Appendix – command summary

COMMAND SUMMARY

## PSM17xx communications manual

command format	reply format
*CLS	
*ESE,value	
*ESE?	single integer data value
*ESR?	single integer data value
*IDN?	company,product,serial no,version
*OPC?	0 or 1
*RST	
*SRE,value	single integer data value
*SRE?	
*STB?	single integer data value
*TRG	
*TST?	single integer data value
*WAI	
ABORT	
ACTRIM,channel,level,tol	
ALARM,latch,sounder	
ALARM?	single integer data value
ALARME,value	
ALARME?	single integer data value
ALARM1,type,data,high,low	
ALARM2,type,data,high,low	
AMPLIT,amplitude	
ANALOG,type,value1,value2	
BANDWI,type	
BEEP	
BLANKI,on/off,threshold	
CONFIG,parameter,data	
CONFIG,parameter?	single integer or real data value
COUPLI,channel,coupling	
COUPLI,channel?	single integer data value
DATALO,function,interval	
DATALO,start,records?	index,time,data... one record per line
DAV?	single integer data value
DAVER,value	
DAVER?	single integer data value
FILTER,type,dynamics	
FRA	
FRA?	freq,mag1,mag2,dB,phase,delay
FRA,SWEEP?	n lines of FRA? data
FREQUE,frequency	
FSWEEP,steps,start,end,log	

## PSM17xx communications manual

GAINPH	
GAINPH?	freq,mag1,mag2,dB,phase,delay
GAINPH,SWEEP?	n lines of GAINPH? data
HARMON,scan,para,h,hmax	
HARMON?	freq,mag1,mag2,hmag1,hmag2,h1,h2
or	freq,mag1,mag2,thd1,thd2,h1,h2
HARMON,SERIES?	mag1,%1, $\phi$ 1,mag2,%2, $\phi$ 2
HARMON,SWEEP?	n lines of HARMON? data
INPUT,channel,type	
INPUT,channel?	single integer data value
INTYPE,channel,type	
KEYBOA,value	
LCR,conditions,param,head	
LCR?	freq,mag1,mag2,impedance,phase,R, L,C (series),R,L,C (parallel), $\tan\delta$ ,Q, reactance
LCR,SWEEP?	n lines of data: freq,QF, $\tan\delta$ ,impedance,phase,L,C,R
LOWFRE,on/off	
MARKER,on/off,frequency	
MODE,type	
OFFSET,offset	
OUTPUT,type	
PAV,parameter,scaling	
PAV?	freq,mag1,mag2,parameter,phase,a,b
PAV,SWEEP?	n lines of VECTOR? data
PHASE	
PHASE?	freq,phase
PHASE,STREAM>window	phase,phase,phase,phase,phase,.....
PHCONV,convention	
PHREF,channel	
POWER,integration type	
POWER,WATTS?	W,W.f,VA,VA.f,pf,pf.f,Wdc,W.h,freq
POWER,RMS?	rms1,rms2,dc1,dc2,fnd1,fnd2, $\phi$ 1, $\phi$ 2
POWER,INTEGR?	Wh,Wh.f,VAh,VAh.f,pf,pf.f,Ah,Ah.f,t
PPOINT,value	
PPOINT?	single integer data value
PROGRAM,function,number	
PROGRAM?	CR terminated text string
RANGE,ch,ranging,range	
RESOLU.format	
REZERO	
SCALE,channel,factor	
SCALE,channel?	single real data value

## PSM17xx communications manual

SHUNT,channel,resistance	
SHUNT,channel?	single real data value
SPEED,speed	
SSWEEP,function,number	
START	
STATUS,channel?	range number,range text,over/low/ok
STOP	
STREAM,enable>window	
STREAM,disable	
STREAM?	data, data, data, data, data, .....
TFA	
TFA?	freq,mag1,mag2,dB,phase,delay
TFA,SWEEP?	n lines of TFA? data
TXA,test,fixture,load,source	
TXA?	freq,mag1,mag2,parameter
TXA,SWEEP?	n lines of TXA? data
TXTEST,test,wind1,wind2	
USER?	3 CR terminated text strings
VECTOR,parameter,scaling	
VECTOR?	freq,mag1,mag2,parameter,phase,a,b
VECTOR,SWEEP?	n lines of VECTOR? data
VERSION?	datecode,type,cpu,dsp,fpga,boot
VRMS	
VRMS?	RMS? data followed by SURGE?
VRMS,RMS?	rms1,rms2,dc1,dc2,ac1,ac2,db1,db2
VRMS,SURGE?	pk1,pk2,cf1,cf2,surge1,surge2
WAVEFO,type	
ZERO	
ZERO,DELETE	
ZOOM,level,d1,d2,d3,d4	
ZOOM?	level,d1,d2,d3,d4

# PSM17xx communications manual

## calibration commands

CALAPP	
CALCOM,freq	
CALDCO,value	
CALFIL,index,value	
CALFIL?	six real data values
CALFRQ,index,freq	
CALFRQ?	seven real data values
CALHF,index,value	
CALIBR,index,value	
CALIBR?	single integer data value
CALIDS,string	
CALIDS?	string
CALOUT,index,value	
CALPHA,index	
CALRES	
CALSAV,password	
CALSNO,serial number	
CALSTR,string	
CALSTR?	string



## Newtonson4th Ltd. contact details

Please direct all queries or comments regarding the PSM17xx instruments or this manual to:

Newtonson4th Ltd.  
30 Loughborough Rd.  
Mountsorrel  
LOUGHBOROUGH  
LE12 7AT  
United Kingdom

Tel: (0116) 230 1066    international    +44 116 230 1066  
Fax: (0116) 230 1061    international    +44 116 230 1061

E-mail address:    [sales@newtonson4th.com](mailto:sales@newtonson4th.com)  
   [office@newtonson4th.com](mailto:office@newtonson4th.com)

web site:                            [www.newtonson4th.com](http://www.newtonson4th.com)

At Newtonson4th Ltd. we have a policy of continuous product improvement and are always keen to hear comments, whether favourable or unfavourable, from users of our products. Please telephone, fax, write or e-mail with your comments.