PSM2200

QuanteQ

communications manual

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 outlet ***before*** connecting the inputs to hazardous voltages.
- The input connectors are High Voltage safety types for use up to 500V peak input from earth, overvoltage category II. Do not exceed 500V peak on any input connection. Only use test leads that are fitted with approved High Voltage safety connectors when working with hazardous voltages.
- Keep the ventilation holes on the underneath, rear, 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 QuanteQ over RS232 or GPIB. For more general operating instructions for the instrument refer to the “QuanteQ 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.

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.

3 August 2004

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
*SRE? 2-8
*STB? 2-9
*TRG 2-10
*TST? 2-11
*WAI 2-12
ABORT 2-13
ACTRIM 2-14
AMPLIT 2-15
AMPLI2 2-16
BLANKI 2-17
CONFIG 2-18
CONFIG? 2-19
COUPLI 2-20
DAV? 2-21
DAVER 2-22
DAVER? 2-23
DCTRIM 2-24
DUALFR 2-25
FALLTI 2-26
FILTER 2-27
FRA 2-28
FRA? 2-29
FREQUE 2-30
FSWEEP 2-31
GAINPH 2-32

GAINPH? 2-33
HARMON 2-34
HARMON? 2-35
INPUT 2-36
LCR 2-37
LCR? 2-38
LOWFRE 2-39
MODE 2-40
OFFSET 2-41
OUTPUT 2-42
PAV 2-43
PAV? 2-44
PHASE 2-45
PHASE? 2-46
PHCONV 2-47
PHREF 2-48
POWER 2-49
POWER? 2-50
PPORT 2-51
PPORT? 2-52
PRETRI 2-53
PROGRA 2-54
PSD 2-55
PSD? 2-56
RANGE 2-57
RESOLU 2-58
REZERO 2-59
RISETI 2-60
SCALE 2-61
SHUNT 2-62
SLM 2-63
SLM? 2-64
SPEED 2-65
START 2-66
STATUS? 2-67
STOP 2-68
STREAM 2-69
STREAM? 2-70
TFA 2-71
TFA? 2-72
TIMEBA 2-73
TRIGGE 2-74
TXA 2-75
TXA? 2-76

TXTEST....2-77
USER? 2-78
VECTOR....2-79
VECTOR? 2-80
VERSIO?....2-81
VRMS 2-82
VRMS?....2-83
WAVEFO 2-84
WIDTH 2-85
ZERO 2-86
ZOOM....2-87
ZOOM? 2-88

Appendix – command summary 1

1 Using remote control

QuanteQ is fitted with an RS232 serial communications port as standard, and may have an IEEE488 (GPIB) interface fitted as an option. The two 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
IEEE488	carriage return or EOI	carriage return with EOI

All the functions of QuanteQ 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 QuanteQ 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 timebase for the oscilloscope function is TIMEBA but the full word TIMEBASE may be sent as the redundant SE 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.

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.

QuanteQ 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 next section).

QuanteQ 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 (see section 5.2).

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

A summary of the available commands is given in the Appendix.

Commands are executed in sequence except for two special characters that override the usual control procedures and 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		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 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 QuanteQ 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

QuanteQ will only transmit when CTS (pin 8) is asserted, and can only receive if DCD (pin 1) is asserted. QuanteQ constantly asserts (+12V) DTR (pin 4) so this pin can be connected to any unwanted modem control inputs to force operation without handshaking. QuanteQ 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 QuanteQ 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 meter, 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.

QuanteQ 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, 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,QUANTEQ,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.

***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 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
tolerance:
required accuracy in percent

Reply: none

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

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

AMPLIT **AMPLIT**

Function:	Set output amplitude
Description:	Sets the output amplitude in Volts for the selected generator.
Format:	AMPLIT,*amplitude*
Arguments:	peak amplitude in Volts
Reply:	none
Example:	AMPLIT,0.5 (set peak amplitude to 0.5V)
Notes:	The same command is used for all the generators. The data is applied to whichever generator has been selected.

AMPLI2 **AMPLI2**

Function:	Set output amplitude 2
Description:	Sets the output amplitude in Volts for the second component of the dual frequency generator.
Format:	AMPLI2,*amplitude*
Arguments:	peak amplitude in Volts
Reply:	none
Example:	AMPLI2,0.5 (set peak amplitude to 0.5V)
Notes:	This command has no effect on the other generators, only the dual frequency generator.

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: or:	CONFIG,*index?* 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
- ACHIGH

Reply: none

Example: COUPLI,CH2,AC+DC

Notes:

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

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:	

DCTRIM **DCTRIM**

Function: Set dc control parameters

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

Format: DCTRIM,*channel,level,tolerance*

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

Reply: none

Example: DCTRIM,CH1,0.15,10 (150mV, 10%)

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

DUALFR **DUALFR**

Function: Set parameters for dual frequency generator

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

Format: DUALFR,*type,cycles1,cycles2*

Arguments: type:
HARMON
FSK
cycles1:
number of cycles for component 1
cycles2:
number of cycles for component 2

Reply: none

Example: DUALFR,HARMON,1,5

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

FALLTI **FALLTI**

Function: Set the pulse generator fall time

Description: Sets the fall time in seconds.

Format: FALLTI,*falltime*

Arguments: Fall time in seconds

Reply: None

Example:
```
OUTPUT,OFF
OUTPUT,PULSE
AMPLIT,3
FREQUE,0.1
WIDTH,10e-3
RISETI,1e-3
FALLTI,3e-3          (set fall time to 3ms)
OUTPUT,ON
```

Notes: Set the fall time to 0 to select the fastest possible ramp limited only by output filtering.
The pulse waveform is built up in a 32768 byte buffer so the step resolution within the ramp is 1 part in 32768 of the cycle time

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: or: or:	FRA? FRA?SWEEP FRA,SWEEP?
Arguments:	none, or SWEEP
Reply:	5 data values separated by commas freq,mag1,mag2,db,phase 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 output frequency in Hz for the selected generator.
Format:	FREQUE,*frequency*
Arguments:	frequency in Hz
Reply:	none
Example:	FREQUE,5e4 (set frequency to 50kHz)
Notes:	The same command is used for all the generators. The data is applied to whichever generator has been selected.

FSWEEP **FSWEEP**

Function: Set the frequency sweep parameters

Description: Sets the start frequency in Hz, the end frequency and the number of steps for the selected function.

Format: FSWEEP,*steps,start,end*

Arguments: steps:
number of steps
start:
start frequency in Hz
end
end frequency in Hz

Reply: none

Example: MODE,GAINPH
FSWEEP,50,1000,1e6
(set 50 steps between 1kHz and 1MHz)

Notes: 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: or: or:	GAINPH? GAINPH?SWEEP GAINPH,SWEEP?
Arguments:	none, or SWEEP
Reply:	5 data values separated by commas freq,mag1,mag2,db,phase 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: or: or:	HARMON? HARMON?SWEEP HARMON,SWEEP?
Arguments:	none, or SWEEP
Reply: single: thd:	7 data values separated by commas: freq,mag1,mag2,hmag1,hmag2,harm1,harm2 freq,mag1,mag2,thd1,thd2,harm1,harm2 1 line per result for sweep data
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:

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
- LOW
- 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: or: or:	LCR? LCR?SWEEP LCR,SWEEP?
Arguments:	none, or SWEEP
Reply:	13 data values separated by commas: freq, mag1, mag2, impedance, phase, series R, series C, series L, //R, //C, //L, tanδ, Q
sweep reply:	8 data values per line per sweep result: freq,Q,tanδ,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 only relevant when not using the QuanteQ generator for the frequency reference.

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)
SCOPE	(oscilloscope)
GAINPH	(gain/phase analyser)
VECTOR	(vector voltmeter)
PHASE	(phase meter)
PSD	(phase sensitive detector)
POWER	(power meter)
SLM	(selective level meter)
LCR	(LCR meter)
HARMON	(harmonic analyser)
TXA	(transformer analyser)

Reply: none

Example: MODE,GAINPH

Notes: MODE sets the measurement mode of the instrument, use the OUTPUT command to select the output mode

OFFSET **OFFSET**

Function:	Set the output offset
Description:	Sets the output offset in Volts for the selected generator.
Format:	OFFSET,*offset*
Arguments:	offset in Volts
Reply:	none
Example:	OFFSET,5e-3 (set offset to 5mV)
Notes:	The same command is used for all the generators. The data is applied to whichever generator has been selected.

OUTPUT **OUTPUT**

Function: Set output

Description: Selects the output generator of the instrument or turns the output on or off.

Format: OUTPUT,*command*
OUTPUT,*type*

Arguments: command:
- OFF
- ON

type:

DISABLE	(none)
SIGNAL	(sin/squ/tri)
DCONLY	(dc)
PULSE	(pulse)
NOISE	(white noise)
DUALFR	(dual frequency)

Reply: none

Example: OUTPUT,PULSE

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: or: or:	PAV? PAV?SWEEP 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,*conventionl*

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:
- 8 data values separated by commas
  W,W.f,VA,VA.f,pf,pf.f,Wdc,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:
- 25 data values separated by commas
  WATTS, RMS, INTEGR

Example: POWER?WATTS

Notes:

PPORT **PPORT**

Function:	Set the printer port
Description:	Force the logic level on the printer 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 printer port may be used as an 8 bit logic level output port

PPORT? **PPORT?**

Function: Read the printer port

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

Format: PPORT?

Arguments: none

Reply: single integer data value

Example:
PPORT?
40

Notes: The printer port may be used as a 5 bit logic level input port:

bit 0	ERR
bit 1	SLI
bit 2	END
bit 3	BUSY
bit 4	ACK

PRETRI **PRETRI**

Function: Set the pretrigger

Description: Select the pretrigger none, 25%, 50% or 75%.

Format: PRETRI,*data*

Arguments: data:

0	(none)
1	(25%)
2	(50%)
3	(75%)

Reply: none

Example: PRETRI,0 (no pretrigger)

Notes:

PROGRA **PROGRA**

Function: Access non volatile program stores.

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

Format: PROGRA,*functio,numbere*

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.

PSD **PSD**

Function:	Set phase sensitive detector mode.
Description:	Set phase sensitive detector mode and integration type.
Format:	PSD,*integration type*
Arguments:	integration type: COHERENT MAGNITUDE
Reply:	none
Example:	PSD,COHERENT PSD
Notes:	It is not necessary to send the integration type argument.

PSD? **PSD?**

Function:	Phase sensitive detector query
Description:	Reads phase sensitive detector results. Sets phase sensitive detector mode if not already set. Waits for next unread data if necessary. Clears new data available bit read by DAV?
Format:	PSD?
Arguments:	none
Reply:	3 data values separated by commas: freq,magnitude,integrated result
Example:	PSD? 1.8396E2,1.4033E-3,4.0235E-4
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 resolution
Description:	Sets the number of digits in the data sent over the communications to 5 (NORMAL) or 6 (HIGH).
Format:	RESOLU,*resolution*
Arguments:	Resolution: NORMAL HIGH
Reply:	None
Example:	RESOLU,HIGH (set resolution to 6 digits)
Notes:	The resolution only affects the data sent by communications, not the display.

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:	

RISETI **RISETI**

Function: Set the pulse generator rise time

Description: Sets the rise time in seconds.

Format: RISETI,*risetime*

Arguments: Rise time in seconds

Reply: None

Example:
```
OUTPUT,OFF
OUTPUT,PULSE
AMPLIT,3
FREQUE,0.1
WIDTH,10e-3
RISETI,1e-3          (set rise time to 1ms)
FALLTI,3e-3
OUTPUT,ON
```

Notes: Set the rise time to 0 to select the fastest possible ramp limited only by output filtering.
The pulse waveform is built up in a 32768 byte buffer so the step resolution within the ramp is 1 part in 32768 of the cycle time

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.

SLM **SLM**

Function: Set gain/phase analyser mode.

Description: Set gain/phase analyser mode with selectivity, scan type and frequency.

Format: SLM*,select,scan,frequency1,frequency2*

Arguments:
select:
- COARSE
- MEDIUM
- FINE
- VFINE

scan:
- SINGLE
- DUAL
- SWEEP

frequency1:
- search frequency 1 in Hz

frequency2:
- search frequency 2 in Hz

Reply: none

Example:
SLM,COARSE
SLM,MEDIUM,DUAL,1800,2400

Notes: It is not necessary to send all the arguments.
Set sweep parameters with FSWEEP
Set blanking with BLANKI

SLM? **SLM?**

Function:	Selective level meter query
Description:	Read selective level meter results. Sets selective level meter mode if not already set. Waits for next unread data if necessary. Clears new data available bit read by DAV?
Format: or: or:	SLM? SLM?SWEEP SLM,SWEEP?
Arguments:	none, or SWEEP
Reply:	5 data values separated by commas: freq,mag1,mag2,dbm1,dbm2 10 data values if dual scan, frequency 1 first then frequency 2 one 5 value line per result for sweep data
Example:	SLM,COARSE,DUAL,1800,2400 SLM? data returned
Notes:	SLM? waits for next unread data. SLM?SWEEP does not wait for new data – data can be read multiple times.

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:

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: or: or:	TFA? TFA?SWEEP TFA,SWEEP?
Arguments:	none, or SWEEP
Reply:	5 data values separated by commas freq,mag1,mag2,db,phase 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

TIMEBA **TIMEBA**

Function:	Set the timebase.
Description:	Sets the timebase in seconds per division.
Format:	TIMEBA,*timebase*
Arguments:	timebase in s/div
Reply:	none
Example:	TIMEBA,5E-4 (set timebase to 500μs/div)
Notes:	

TRIGGE **TRIGGE**

Function: Set the trigger conditions

Description: Set the trigger level, channel, rising or falling edge, and auto normal or single shot.

Format: TRIGGE,*level*,*channel*,*edge*,*type*

Arguments:
level:
absolute trigger level in Volts
channel:
CH1
CH2
edge:
RISING
FALLIN
type:
AUTO
NORMAL
SINGLE

Reply: none

Example:
TRIGGE,0.2,CH1,RISING,SINGLE
TRIGGE,0.5

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

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: or: or:	TXA? TXA?SWEEP TXA,SWEEP?
Arguments:	none, or SWEEP
Reply: txdcr txmagi	3 or 4 data values separated by commas: freq,mag1,mag2,parameter 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
QuanteQ #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: or: or:	VECTOR? VECTOR?SWEEP 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: version query

Description: Returns an identification string to give details about the version.

Format: *IDN?

Arguments: none

Reply: An ASCII string with the following fields separated by comms:

- model
- mark
- code version
- fpga version
- dsp version
- boot code version

Example: *VERSION?
QUANTEQ,3,3.10,3.10.3.10,1.05

Notes:

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:

```
OUTPUT,SIGNAL
FREQUE,500
WAVEFO,TRIANG    (triange wave)
OUTPUT,ON
```

Notes: This only applies to the signal generator

WIDTH **WIDTH**

Function:	Set the pulse width
Description:	Sets the output pulse width in seconds.
Format:	WIDTH,*width*
Arguments:	width in seconds
Reply:	none
Example:	OUTPUT,OFF OUTPUT,PULSE AMPLIT,3 FREQUE,0.5 WIDTH,1e-3 (set pulse width to 1ms) OFFSET,0 OUTPUT,ON
Notes:	This only applies to the pulse generator

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

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	
AMPLIT,amplitude	
AMPLI2,amplitude	
BLANKI,on/off,threshold	
CONFIG,parameter,data	
CONFIG,parameter?	single integer or real data value
COUPLI,channel,coupling	
COUPLI,channel?	single integer data value
DAV?	single integer data value
DAVER,value	
DAVER?	single integer data value
DCTRIM,channel,level,tol	
DUALFR,type,cyc1,cyc2	
FALLTI,falltime	
FILTER,type,dynamics	
FRA	
FRA?	freq,mag1,mag2,dB,phase
FRA,SWEEP?	n lines of GAINPH? data
FREQUE,frequency	
FSWEEP,steps,start,end	
GAINPH	
GAINPH?	freq,mag1,mag2,dB,phase
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,SWEEP?	n lines of HARMON? data
INPUT,channel,type	
INPUT,channel?	single integer data value
LCR,conditions,param,head	
LCR?	freq,mag1,mag2,impedance,phase,R, L,C (series),R,L,C (parallel),tanδ,Q
LCR,SWEEP?	n lines of data: freq,QF,tanδ,impedance,phase,L,C,R
LOWFRE,on/off	
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,freq
POWER,RMS?	rms1,rms2,dc1,dc2,fnd1,fnd2,ϕ1,ϕ2
POWER,INTEGR?	Wh,Wh.f,VAh,VAh.f,pf,pf.f,Ah,Ah.f,t
PPORT,value	
PPORT?	single integer data value
PRETRI,data	
PRINT	
PROGRA,function,number	
PSD,integration type	
PSD?	freq,magnitude,integration
RANGE,ch,ranging,range	
RESOLU,resolution	
REZERO	
RISETI,risetime	
SCALE,channel,factor	
SCALE,channel?	single real data value
SHUNT,channel,resistance	
SHUNT,channel?	single real data value
SLM,select,scan,freq1,freq2	
SLM?	freq,mag1,mag2,dbm1,dbm2
SLM,SWEEP?	n lines of SLM? data
SPEED,speed	
START	

STATUS,channel?	range number,range text,over/low/ok
STOP	
STREAM,enable,window	
STREAM,disable	
STREAM?	data, data, data, data, data, …..
TIMEBA,timebase	
TFA	
TFA?	freq,mag1,mag2,dB,phase
TFA,SWEEP?	n lines of GAINPH? data
TRIGGE,level,ch,edge,type	
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 strings
VECTOR,parameter,scaling	
VECTOR?	freq,mag1,mag2,parameter,phase,a,b
VECTOR,SWEEP?	n lines of VECTOR? data
VERSIO?	model,mark,cpu,fpga,dsp,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	
WIDTH,width	
ZERO	
ZERO,DELETE	
ZOOM,level,d1,d2,d3,d4	
ZOOM?	level,d1,d2,d3,d4

calibration commands

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

Newtons4th Ltd contact details

Please direct all queries or comments regarding the QuanteQ instrument or manual to:

Newtons4th Ltd.
Ratcliffe House
30 Rothley Road
Mountsorrel
LOUGHBOROUGH
LE12 7JU
United Kingdom

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

E-mail address: sales@newtons4th.com
office@newtons4th.com

web site: www.newtons4th.com

At Newtons4th 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.