Boonton

PROGRAMMING REFERENCE

PMX40

RF Power Meter

98408800B| Rev 20230308

Revision 20230308

P/N 98408800B

SAFETY SUMMARY

The following general safety precautions must be observed during all phases of operation and maintenance of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Boonton Electronics assumes no liability for the customer's failure to comply with these requirements.

DO NOT OPERATE THE INSTRUMENT IN AN EXPLOSIVE ATMOSPHERE
Do not operate the instrument in the presence of flammable gases or fumes.

DO NOT OPERATE THE INSTRUMENT OUTSIDE
This instrument is designed for indoor use only.

KEEP AWAY FROM LIVE CIRCUITS
Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with the power cable connected. Under certain conditions dangerous voltages may exist even though the power cable was removed, therefore; always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE
Service and adjustments should be performed only by qualified service personnel. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT POSITION THE INSTRUMENT SO THAT IT IS DIFFICULT TO OPERATE THE DISCONNECTION DEVICE
The main power disconnection switch is located on the rear panel.

DO NOT SUBSTITUTE PARTS OR MODIFY INSTRUMENT
Do not install substitute parts or perform any unauthorized modifications on the instrument. Return the instrument to Boonton Electronics for repair to ensure that the safety features are maintained.

SAFETY SYMBOLS

This safety requirement symbol has been adopted by the International Electro-technical Commission, Document 66 (Central Office) 3, Paragraph 5.3, which directs that an instrument be so labeled if, for the correct use of the instrument, it is necessary to refer to the instruction manual. In this case it is recommended that reference be made to the instruction manual when connecting the instrument to the signal source and USB host.

The CAUTION symbol denotes a hazard. It calls attention to an operational procedure, practice or instruction that, if not followed, could result in damage to or destruction of part or all of the instrument and accessories. Do not proceed beyond a CAUTION symbol until its conditions are fully understood and met.

The NOTE symbol is used to mark information which should be read. This information can be very useful to the operator in dealing with the subjects covered in this section.

The HINT symbol is used to identify additional comments which are outside of the normal format of the manual and provide users additional information about the subject.

Contents

1 Remote Operation

1.1 LAN Configuration

The PMX40 LAN interface is configured using the *System > I/O Config > LAN* menu. The instrument may be set to automatically accept its IP address and associated information from a DHCP server, or the configuration data may be entered manually.

The PMX40 communicates via HiSLIP (High-Speed LAN Instrument Protocol), which is a TCP/IP based format that allows GPIB operation to be emulated via the LAN. Once the LAN parameters have been configured, the controller can use the instrument's IP address to assist the remote-control software in locating and connecting to the PMX40. Typically, the host will use VISA to provide the interface layer between the instrument and the remote-control software.

1.2 GPIB Configuration

GPIB remote control is enabled with option PMX40-GPIB.

The PMX40-GPIB interface is ready for use. Assign a GPIB address using the *System > I/O Config >* menu. The primary address can be set to any value from 1 to 30 inclusive. The value assigned must be unique to each GPIB device. Secondary address is not implemented.

1.3 SCPI Language

PMX40 instruments may be remotely controlled using commands that follow the industry standard SCPI programming conventions.

1.3.1 SCPI Structure

The SCPI instrument model defines a hierarchical command structure based on "command nodes". Each node may contain commands or names of a next-level command node. Each command is formed of a series of keywords joined together and delimited by a colon ":" character. The command begins with a colon at the "root node", and traverses downwards through the command tree to form a specific command. This structure is very similar to a DOS file system, where the file system begins at the root level (":"), and each directory (SCPI subsystem) may contain a list of files (SCPI commands) and lower-level directories. To execute an individual command, the entire command name ("path") must generally be specified, although there are several shortcuts available to reduce the command string length.

SCPI subsystems or command groups are usually aligned with instrument functions, and the standard provides a number of pre-defined subsystems that can be used for most instrument types. For example, the top-level SENSe subsystem groups commands that are related to sensing signals (detection, amplification, digitization, linearization), while the OUTPut subsystem contains commands that control output functions of the instrument such as voltage output or controlling an RF reference output.

1.3.2 Long and Short Form Keywords

Each command or subsystem may be represented by either its full keyword, or a short form of that keyword. The short form is typically the first several characters of the full name, although this is not necessarily the case. The short form of each keyword is identified in this manual by the keyword characters shown in UPPERCASE, while the long form will be shown in mixed case. For example, the short form of "CALCulate" is "calc", while the long form is "calculate". Long form and short form commands may be used interchangeably, but only the exact forms are permitted – intermediate length commands will not be recognized. Sending "CALCUL" will cause an error.

Note that not all keywords have long forms – in this case, the entire keyword will be shown in uppercase.

While uppercase and lowercase text is used to identify keywords, SCPI is generally case-insensitive, so it is acceptable to send uppercase, lowercase, or mixed case keywords to the instrument. The only exception is when a command accepts a literal string argument. In this case, quotes may be used to delimit a string of user-defined case.

1.3.3 Subsystem Numeric Suffixes

Certain subsystems, such as the SENSe or CALCulate subsystems in the PMX40, often exist as more than one instance (often called a "channel" in an instrument). In this case, an optional numeric suffix may be used to define the channel. If this suffix is not present, the default channel is assumed. For example, SENSe or SENSe1 defines operations affecting the instrument's "Channel 1" measurement path, while SENSe2 commands will apply to channel 2.

1.3.4 Colon Keyword Separators

The colon (":") character is used similar to the way a slash or backslash is used in a filesystem. Prefixing a command string with a colon is optional and is ignored by the PMX40.

1.3.5 Command Arguments and Queries

Many commands require arguments. In this case, the entire command string is sent, followed by the argument. A space is used to separate the command from the argument. For example, "SENSe:CORRection:DCYCle 25.0" sets duty cycle correction to a value of 25.0. Arguments may be numeric, or alphanumeric. If a command requires more than one numeric argument, the arguments must be sent as a comma delimited list.

To read the current value of a particular parameter, the Query Form of its command may be used. A command query is formed by appending a question-mark ("?") suffix to the command instead of an argument list. There should not be any whitespace between the command and the suffix. For example, "SENSe:CORRection:DCYCle?" queries the duty cycle correction parameter, and causes the instrument to return its current value.

1.3.6 Semicolon Command Separators

The semicolon (";") character is used to separate multiple commands on a single line. However, the full command path must be specified. For example, "SENSe:CORRection:DCYCle 25.0; SENSe:CORRection:CALFactor 2.12" will set the duty cycle correction parameter and change the correction calfactor in use.

1.3.7 Command Terminators

All SCPI command strings transmitted to the instrument via the GPIB bus must be terminated. Any character with the IEEE488 EOI (End-Or-Identify) control line asserted may be used as a terminator. This may be the last letter of the command, query or argument. Optionally, a CR (ASCII 13) and/or LF (ASCII 10) may be included. These are ignored by the parser, but if present, the EOI must be asserted on the last message character transmitted.

On LAN messages, the packetized protocols provide automatic termination of each message. Again, CR and/or LF may be present, but must be the last message character(s) of the packet.

1.3.8 PMX40 SCPI Implementation

The SCPI implementation for the PMX40 provides a single or dual SENSe sub-system to handle sensor input and a matching single or dual CALCulate sub-system to process the data obtained from the sensors into useful results. A TRIGger sub-system provides for measurement and signal synchronization. DISPlay commands are used for graph and text display control, formatting and timebase selection. The CALibration sub-system is used to calibrate power sensors. Channel dependent commands end with a number to indicate the desired channel as follows:

Examples:

:CALCulate:STATe ON	Turn on measurement channel 1 (default channel number)
:CALCulate1:STATe ON	Turn on measurement channel 1 (specified channel number)
:CALC:STAT ON	Turn on measurement channel 1 (short form, default chan #)
:CALC1:STAT ON	Turn on measurement channel 1 (short form, specified chan #)
:CALCulate:STATe?	Query the state of measurement channel 1 (default chan #)
:CALC:STAT?	Query the state of channel 1 (short form, default chan #)
:CALCulate1:STATe?	Query the state of measurement channel 1 (specified chan #)
:CALC1:STAT?	Query the state of channel 1 (short form, specified chan #)
:DISPlay:PULSe:TIMEBASE 0.0001	Set timebase range to 100 us/Div (time in seconds)
:DISP:PULS:TIMEBASE 100 us	Set timebase range to 100 us/Div (short form with time units)
:SENSe:CORRection:OFFSet 0.42	Set channel 1 offset correction to 0.42 dB (chan units dBm)
:TRIGger:LEVel –3.12	Set trigger level to –3.12 dBm (trig units dBm)
:SENS:CORR:OFF 0.42; :TRIG:LEV –3.12	Concatenated commands using semi-colon (short form)

In the discussion and tables below, the following notation will be used:

Command name long and short form:	SYSTem
Optional command name in brackets:	SYSTem:ERRor[:NEXT]?
Command with channel dependence:	CALCulate[1|2]:STATe OFF
Default channel 1:	CALCulate:STATe OFF
Explicit channel 1:	CALCulate1:STATe OFF
Select channel 2:	CALCulate2:STATe OFF
Short Form:	CALC2:STAT OFF
Command which takes numeric argument:	SENSe1:AVERage <numeric_value>
Same command; query:	SENSe1:AVERage?
Command with literal text argument:	TRIGger:SOURce <character data>
Command with no query form:	*CLS
Command with query form only:	SENSe:SENSOR:TYPE?

SYNTAX NOTES

Square brackets [ ] are used to enclose the list of valid channels for a command, or a list of command options separated by the vertical separator bar | character. This character is for syntax only, and is not to be entered as part of the command. By default, if no channel number is specified, Channel 1 is selected.

A literal argument denoted by <character data> indicates a word or series of characters, which must exactly match one of the choices for the command. An argument denoted by <numeric_value> requires a string which, when converted to a number, is within the range of valid arguments. Numerical values can generally be in any common form including decimal and scientific notation. <Boolean> indicates an argument which must be either true or false. Boolean arguments are represented by the values 0 or OFF for false, and 1 or ON for true. Queries of Boolean parameters always return 0 or 1.

Curly braces { } are used to enclose two or more possible choices for a mandatory entry, separated by the comma character. One of the enclosed options MUST be inserted into the command, and the braces are not to be entered as part of the command.

1.4 Basic Measurement Information

The easiest way to obtain a reading is by use of the MEASure command. This command initiates one complete measurement sequence which includes a default configuration. Examples are:

MEAS1:POWER?	To return the average power of channel 1, or
MEAS1:VOLTAGE?	To return the average voltage of channel 1.

For finer control over the measurement, individual configuration and function commands should be used. Readings are obtained using the FETCh[ ]? command for current data or the READ[ ]? command for fresh data. These commands may return multiple results if an array is read.

Readings are in fundamental units as set by the CALCulate[1|2|3|4]:UNIT command. Each reading is preceded by a condition code, which has the following meaning:

-1 Measurement is STOPPED. Value returned is not updated.

0 Error return. Measurement is not valid.

1 Normal return. No error.

2 An Under-range condition exists.

3 An Over-range condition exists.

With the INITiate:CONTinuous OFF condition, a single measurement cycle is started by use of the INITiate[:IMMEDIATE] command, where bracketed commands are optional. Multiple triggered measurement cycles are enabled by INITiate:CONTinuous ON and a TRIGger:SOURce selection. If TRIGger:MODE is set to FREERUN, a free running measurement process is started. Otherwise, a measurement cycle begins with each valid trigger condition.

1.5 SCPI Command Reference

This section contains a list of all SCPI remote commands accepted by the PMX40. The list is grouped by SCPI subsystem or IEEE488.2 function, and includes a detailed description of each command.

1.5.1 SCPI Command Summary

Table 1-1 SCPI COMMAND SUMMARY	
*IDN	Identification Query
*OPC?	Operation Complete Query
*OPC	Operation Complete Command
*TST?	Self-Test Query
ABORt	Immediately set measurement trigger system to idle
CALCulate:MODE	Set/return instrument mode. <numeric_value> = PULSe, MODulated, STATistical
CALCulate[1|2|3|4]:PKHLD	Set/return state of the peak hold function. <Boolean> = OFF, AVG, INST
CALCulate[1|2|3|4]:STATe	Enable currently selected channel allowing measurements to be made. <Boolean> = 0, 1, OFF, ON
CALCulate[1|2|3|4]:UNIT s	Change channel units. <numeric_value> = DBM, Watts, Volts, DBV, DBMV, DBUV
DISPlay:CLEar	Clear measurement data and display
DISPlay:ENVELOPE	Enable the Envelope display mode. <Boolean> = 0, 1, ON, OFF
DISPlay:MODUlated:TIMEBASE	Set/return modulated timebase.
DISPlay: MODUlated:TSPAN	Set/return horizontal time span.Ex: <numeric_value>

Table 1-1 SCPI COMMAND SUMMARY	
DISPlay:PULSe:TIMEBASE	Set/return horizontal timebase. Ex: <numeric_value> = 0.05 sets timebase to 50 ms
DISPlay:PULSe:TSPAN	Set/return horizontal time span. Ex:<numeric_value> = 0.05 (50 ms)
DISPlay:[TEXt:]LIN:RESolution	Set number of significant digits for linear displays andremote return values. <numeric_value> = 3 to 5
DISPlay:[TEXt:]LOG:RESolution	Set number of decimal places for log displays and remote return values. <numeric_value> = 0 to 3
DISPlay:TRACe[1|2|3|4]:HOFFSet	Set/return the horizontal offset for statistical mode in dBr. <numeric_value> = -50.00 to +50.00 dBr
DISPlay:TRACe[1|2|3|4]:HSCALE	Set/return the horizontal scale for statistical mode in dB/Div. <numeric_value> = 0.1 to 5.0 dB/Div in 1-2-5 sequence
DISPlay:TRACe[1|2|3|4]:VCENTer	Change vertical center based on current units. Ex:<numeric_value> = 1.23
DISPlay:TRACe[1|2|3|4]:VSCALe	Change vertical scale based on current units. Ex:<numeric_value> = 20 set to 20dB/div (current units)
FETCh[1|2|3|4]:ARRay:AMEAsure:POWer?	Return current Ppeak, Pavgcyc, Pavgpulse, Ptop, Pbot, Overshoot
FETCh[1|2|3|4]:ARRay:AMEAsure:STATistical?	Return current Pavg, Ppeak, Pmin, PkToAvgRatio, CursorPwr, CursorPct, Sample-Count
FETCh[1|2|3|4]:ARRay:AMEAsure:TIMe?	Return current Freq, Period, Width, Offtime, Dcyc, Risetime, Falltime
FETCh[1|2|3|4]:ARRay:CW:POWer?	Return current Pavg, Pmax, Pmin, Ppulse or Pk/Avg
FETCh[1|2|3|4]:ARRay:MARKer:POWer?	Return current Pavg, Pmax, Pmin, P/Avg, Pmrk1, Pmrk2, Pmrk1/Pmrk2
FETCh[1|2|3|4]:CW:POWer?	Return current average reading in power units
FETCh[1|2|3|4]:INTERval:AVERage?	Return average power between MK1 and MK2.
FETCh[1|2|3|4]:INTERval:MAXFilt?	Return maximum filtered power between MK1 and MK2.
FETCh[1|2|3|4]:INTERval:MINFilt?	Return minimum filtered power between MK1 and MK2.
FETCh[1|2|3|4]:INTERval:MAXimum?	Return maximum instantaneous power between MK1 and MK2.
FETCh[1|2|3|4]:INTERval:MINimum?	Return minimum instantaneous power between MK1 and MK2.
FETCh[1|2|3|4]:INTERval:PKAVG?	Return peak to average power between MK1 and MK2.
FETCh[1|2|3|4]:MARKer[1|2]:AVERage?	Return current reading at the specified marker.
FETCh[1|2|3|4]:MARKer:DELTa?	Return difference between MK2 and MK1.
FETCh[1|2|3|4]:MARKer[1|2]:MAXimum?	Return maximum reading at the specified marker
FETCh[1|2|3|4]:MARKer[1|2]:MINimum?	Return minimum reading at the specified marker
FETCh[1|2|3|4]:MARKer:CURsor:PERcent?	Return percent at cursor.
FETCh[1|2|3|4]:MARKer:CURsor:POWer?	Return relative power at cursor.
FETCh[1|2|3|4]:MARKer:RATio?	Return ratio between MK2 and MK1 as a percentage.
FETCh[1|2|3|4]:MARKer:RDELTa?	Return difference between MK1 and MK2.
FETCh[1|2|3|4]:MARKer:RRATio?	Return ratio between MK1 and MK2 as a percentage.
FETCh[1|2|3|4]:MBUF:DATa:AVErage?	Return a partial list of Avg measurements from the last MBUF session
FETCh[1|2|3|4]:MBUF:DATa:AVErage:ALL?	Return a list of Avg measuremenst from the last MBUF session
FETCh[1|2|3|4]:MBUF:DATa:DURation?	Return a partial list of durations from the last MBUF session.
FETCh[1|2|3|4]:MBUF:DATa:DURation:ALL?	Return a list of durations from the last MBUF session.
FETCh[1|2|3|4]:MBUF:DATa:MAXimum?	Return a partial list of Max measurements from the last MBUF session
FETCh[1|2|3|4]:MBUF:DATa:MAXimum:ALL?	Return a list of Max measurements from the last MBUF session.
FETCh[1|2|3|4]:MBUF:DATa:MINimum?	Return a partial list of Min measurements from the last MBUF session
FETCh[1|2|3|4]:MBUF:DATa:MINimum:ALL?	Return a list of Min measurements from the last MBUF session.
FETCh[1|2|3|4]:MBUF:DATa:STARt?	Return a partial list of Start Times from the last MBUF session
FETCh[1|2|3|4]:MBUF:DATa:STARt:ALL?	Return a list of Start Times from the last MBUF session.
FETCh:TEMPerature:AVERage?	Return average internal instrument temperature in degrees C
FETch:TEMPerature:CURRent?	Return current internal instrument temperature in degrees C

INITiate:CONTinuous	Set/return state of mode which triggers meas cycles continuously. <Boolean> = 0, 1, OFF, ON
INITiate[:IMMediate[:ALL]]	Set mode which starts a measurement cycle when trigger event occurs
MARKer[1|2]:POSItion:TIMe	Set/return marker time relative to the trigger.
MARKer:POSItion:PERcent	Set or return the percent probability (y-axis-position) of the CCDF cursors. <numeric_value> = 0.000 to 100.0 %

Table 1-1 SCPI COMMAND SUMMARY	
MARKer:POSItion:POWer	Set or return the cumulative relative power (x-axis-position) of the CCDF cursors. <numeric_value> = 0.000 to 200.000
MEASure[1|2|3|4]:POWer?	Return Modulated Mode Pavg in dBm
MEASure[1|2|3|4]:VOLTage?	Return Modulated Mode Pavg in equivalent volts.
MEMory[1|2|3|4]:SNSR:CFFAST?	Return the sensor high bandwidth (FAST) frequency cal-factor table.
MEMory[1|2|3|4]:SNSR:CFSLOW?	Return the sensor low bandwidth (SLOW) frequency cal-factor table.
READ[1|2|3|4]:ARRay:AMEAsure:POWer?	Return new Ppeak, Pavgcyc, Pavgpulse, Ptop, Pbot, Overshoot
READ[1|2|3|4]:ARRay:AMEAsure:STATistical?	Return new Pavg, Ppeak, Pmin, PkToAvgRatio, CursorPwr, CursorPct, Sample-Count
READ[1|2|3|4]:ARRay:AMEAsure:TIMe?	Return new Freq, Period, Width, Offtime, Dcyc, Risetime, Falltime
READ[1|2|3|4]:ARRay:CW:POWer?	Return new Pavg, Pmax, Pmin, Ppulse or Pk/Avg
READ[1|2|3|4]:ARRay:MARKer:POWer?	Return new Pavg, Pmax, Pmin, P/Avg, Pmrk1, Pmrk2, Pmrk1/Pmrk2
READ[1|2|3|4]:CW:POWer?	Return new average reading in power units
READ[1|2|3|4]:INTERval:AVERage?	Return average power between MK1 and MK2.
READ[1|2|3|4]:INTERval:MAXFilt?	Return maximum filtered power between MK1 and MK2.
READ[1|2|3|4]:INTERval:MINFilt?	Return minimum filtered power between MK1 and MK2.
READ[1|2|3|4]:INTERval:MAXimum?	Return maximum instantaneous power between MK1 and MK2.
READ[1|2|3|4]:INTERval:MINimum?	Return minimum instantaneous power between MK1 and MK2.
READ[1|2|3|4]:INTERval:PKAVG?	Return peak to average power between MK1 and MK2.
READ[1|2|3|4]:MARKer[1|2]:AVERage?	Return new reading at the specified marker.
READ[1|2|3|4]:MARKer:DELTa?	Return difference between MK2 and MK1.
READ[1|2|3|4]:MARKer[1|2]:MAXimum?	Return maximum reading at the specified marker
READ[1|2|3|4]:MARKer[1|2]:MINimum?	Return minimum reading at the specified marker
READ[1|2|3|4]:MARKer:CURsor:PERcent?	Return percent at cursor.
READ[1|2|3|4]:MARKer:CURsor:POWer?	Return relative power at cursor.
READ[1|2|3|4]:MARKer:RATio?	Return ratio between MK2 and MK1 as a percentage.
READ[1|2|3|4]:MARKer:RDELTa?	Return difference between MK1 and MK2.
READ[1|2|3|4]:MARKer:RRATio?	Return ratio between MK1 and MK2 as a percentage.
SENSe[1|2|3|4]:AVERage	Set/return trace averaging count. <numeric_value> = 1 to 16384
SENSe[1|2|3|4]:BANDwidth	Set/return sensor video bandwidth. <character data> = LOW, HIGH
SENSe[1|2|3|4]:CORRection:CALFactor	Set correction factor in dB. <numeric_value> = -3.00 to 3.00
SENSe[1|2|3|4]:CORRection:DCYCle	Set/return duty cycle correction factor in percent. <numeric_value> = 0.01 to 100.0%
SENSe[1|2|3|4]:CORRection:FREQuency	Set channel frequency. <numeric_value> = 0.001e9 to 110.00e9 Hz
SENSe[1|2|3|4]:CORRection:OFFSet	Set/return sensor offset value in dB. <numeric_value> = -300 to 300
SENSe[1|2|3|4]:FILTer:STATe	Set/return filter state. <character data> = OFF, AUTO, ON
SENSe[1|2|3|4]:FILTer:TIMe	Set or return the current length of the integration filter on the selected channel. <numeric_value> = 0.002 to 16.000 seconds
SENSe:MBUF:ACQuire?	Start a new MBUF session
SENSe:MBUF:DELay:END	Set/return time added to end of burst for MBUF analysis.
SENSe:MBUF:DELay:STARt	Set/return time added to beginning of burst for MBUF analysis
SENSe:MBUF:ENABled:FLAG	Set/return bit mask indicating which arrays are filled during MBUF analysis.
SENSe:MBUF:GATe:MODe	Set/return gate mode used during MBUF analysis.
SENSe[1|2|2|3|4]:MBUF:MEASurements:AVAilable?	Return the number of measurements available for the specified channel.
SENSe:MBUF:MEASurements:CLEar	Clear cached avg, min, max, duration, start time, and sequence numbers taken during last MBUF session.
SENSe:MBUF:PERiod	Set/return period I seconds for timed mode MBUF measurements.

SENSe:MBUF:PERiod:DURation	Set/return duration in seconds for timed mode MBUF measurements.
SENSe:QUALifier:END	Set/return time power stays below trigger to count as end of burst in MBUF mode..
SENSe:QUALifier:STARt	Set/return time power stays above trigger to count as beginning of burst in MBUF mode.
SENSe:MBUF:SESsion:COUNt	Set/return the count of elements for arrays for MBUF analysis.
SENSe:MBUF:SESsion:TIMEout	Set/return the time in seconds for MBUF analysis.
SENSe:MBUF:START:MODe	Set/return the mode used to start acquiring MBUF entries.
SENSe[1|2|3|4]:PULSe:DISTal	Set/return distal parameter of the risetime. <numeric_value> = 50.00 to 100.00
SENSe[1|2|3|4]:PULSe:ENDGT	Set/return pulse gate end position in percent of pulse time duration. <numeric_value> = 60.00 to 100.00
SENSe[1|2|3|4]:PULSe:MESIal	Set/return mesial parameter of the risetime. <numeric_value> = 10.00 to 90.00
SENSe[1|2|3|4]:PULSe:PROXimal	Set/return proximal parameter of the risetime. <numeric_value> = 0.00 to 50.00 percent
SENSe[1|2|3|4]:PULSe:STARTGT	Set/return pulse gate start position in percent of pulse timeduration. <numeric_value> = 0.00 to 40.00

Table 1-1 SCPI COMMAND SUMMARY	
SENSe[1|2|3|4]:PULSe:UNIT	Set/return pulse definitions units. <numeric_value> = WATTS, VOLTS
SENSe[1|2|3|4]:SENSOR:TYPE?	Returns the sensor type for the selected channel. <character data> = NONE, CW, VOLT, PEAK
SYSTem:BEEP[:ENABle]	Set/return keypad audible beeper status. <Boolean> = 0, 1, OFF, ON
SYSTem:COMMunicate:GPIB:ADDRess	Set or return the GPIB bus address.<numeric_value> = 1 to 30
SYSTem:COMMunicate:LAN:ADDRess	Set or return the IP address for the Ethernet port. <character data> = instrument IP address in nnn.nnn.nnn.nnn ("dot decimal") format
SYSTem:COMMunicate:LAN:DHCP[:STATe]	Set or return the state of DHCP/AutoIP system for the Ethernet port.<Boolean> = 0, 1, OFF, ON
SYSTem:COMMunicate:LAN:CURRent:ADDRess?	Returns the current IP address for the Ethernet port.
SYSTem:ERRor	Return system error code and description
SYSTem:ERRor[:NEXT]	Return system error code and description
SYSTem:ERRor:CODE	Return system error code
SYSTem:PRESet	Set instrument to default conditions.
SYSTem:VERSion?	Return SCPI version compliance. <numeric_value> = yyyy.v
TRACe[1|2|3|4]:[AVERage]:DATA[:NEXT]?	Return count min, max or average power samples starting at index
TRACe[1|2|3|4]:COUNt	Set/return total number of trace pwr points to return.<numeric_value> = 1 to 501
TRACe[1|2|3|4]:INDEX	Set/return index of the next trace power point. <numeric_value> = 0 to 500
TRIGger:CDF:COUNt	Set/return Statistical Mode terminal count. <numeric_value> = 1 to 4000 megasamples
TRIGger:CDF:DECImate	Set/return termination action when terminal count or time is reached. <character data> = { DECIMATE, RESTART, STOP}
TRIGger:CDF:TIMe	Set/return Statistical Mode terminal time. <numeric_value> = 1 to 3600 sec
TRIGger:DELay	Set/return trigger delay with respect to the trigger. <numeric_value> = see Specifications Section 1-6.
TRIGger:HOLDoff	Set/return trigger holdoff time. <numeric_value> = 0 to 1 sec in 10 ns increments
TRIGger:LEVel	Set/return trigger level. Internal Trigger: <numeric_value> = -40 to +20 dBm plus offset. External Trigger: <numeric_value> = ±5 volts
TRIGger:MODE	Set/return trigger mode on display. < character data > = AUTOPKPK, AUTO, NORMAL, FREERUN
TRIGger:POSition	Set/return trigger position on display. < character data > = LEFT, MIDDLE, RIGHT
TRIGger:SLOPe	Set/return trigger slope on display. <character data> = POS, NEG
TRIGger:SOURce	Select trigger source. < character data > = CH1, CH2, EXT
TRIGger:VERNier	Set or return the fine position of the trigger event on the displayed sweep. <numeric> -30 to +30

1.5.2 IEEE 488.2 Commands

The purpose of IEEE488.2 commands is to provide management and data communication instructions for the system by defining a set of "*" commands (an asterisk followed by a three-character code). These commands allow device control and monitoring.

***IDN?**

Description: Return the instrument identification string. This string contains the manufacturer, model number, serial number and firmware version number.

Syntax: *IDN?

Returns: < Mfgr, Model#, Serial#, Version# >

Valid Modes: Any

***OPC**

Description: Clears the OPC (Operation Complete) status flag. This command is issued before the command to be checked for completion. After this, the flag may be queried by *OPC? until a value of one is returned, indicating the command has completed. Note that the query is not a true query - a value of zero will never be returned.

Syntax: *OPC

Argument: None

Valid Modes: Any

***OPC?**

Description: This command examines the OPC (Operation Complete) status flag and returns a ”1” if all pending operations are complete. If pending operations are not yet complete, it does not return.

Syntax: *OPC?

Returns: Always returns 1 to indicate operations complete. Otherwise, does not return.

Valid Modes: Any

***TST?**

Description: Self-test query. This command initiates a self-test of the instrument and returns a result code when complete. The result is zero for no errors, or a signed, 16-bit number if any errors are detected.

Syntax: *TST

Returns: Error Code

Valid Modes: Any

***WAI**

Description: Wait command. This command insures sequential, non-overlapped execution. The PMX40 always operates in non-overlapped, sequential mode, therefore this command is accepted as valid, but takes no action.

Syntax: *WAI

Argument: None

Valid Modes: Any

1.5.3 CALCulate Subsystem

Functions in the CALCulate subsystem are used to configure the measurement mode and control which portions of the acquired measurement data is used and how it is processed to yield a finished

measurement. In addition to measurement mode, CALCulate is used to define mathematical operations, and measurement units. The numeric suffix of the CALCulate program mnemonic in the CALCulate commands refers to a processing and display "channel", that is CALCulate1 and CALCulate2 represent the power meter's Channel 1 and Channel 2 functions. The CALCulate commands generally DO NOT affect the data acquisition portion of the measurement (see the SENSe subsystem, Section 1.5.12). In a signal-flow block diagram, the CALCulate block operations will follow those of the SENSe block. Note that CALCulate commands will not generate an error if used with a disconnected or non-existing channel on the PMX40.

CALCulate:DCYCle

Description: Set or return the duty cycle correction factor currently in use on the selected channel.

Syntax: CALCulate[1|2|3|4]:CORRection:DCYCle <numeric value>

Argument: <numeric_value> = 0.01 to 100.00 percent

Valid Modes: CW Sensor (Modulated Mode)

CALCulate:MODE

Description: Set or return the system measurement mode. MODULATED mode is a continuous measurement mode primarily for continuously modulated or CW signals. PULSE mode is a signal triggered, oscilloscope-like mode that acquires and analyzes a pulsed signal as a series of one or more triggered sweeps. STATISTICAL mode performs long-term power distribution analysis on modulated signals, and may be operated in a start-stop continuous mode, or a decimated continuous mode.

Note that the measurement mode is global and affects all channels. If any channel has a CW sensor installed, peak or statistical measurements will be unavailable, but the "primary " average power measurement will still be performed.

Syntax: CALCulate:MODE <character data>

Argument: <character data> = {MODulated, PULSe, STATistical}

Valid Modes: Any

CALCulate:PKHLD

Description: Set or return the operating mode of the selected channel's peak hold function. When set to OFF, instantaneous peaks are only held for a short time, and then decayed towards the average power at a rate proportional to the filter time. This is the best setting for most signals, because the peak will always represent the peak power of the current signal, and the resulting peak-to-average ratio will be correct shortly after any signal level changes. When set to ON, instantaneous peaks are held until reset by a new INITiate command or cleared manually. This setting is used when it is desirable to hold the highest peak over a long measurement interval without any decay.

Syntax: CALCulate[1|2|3|4]:PKHLD <Boolean>

Argument: <Boolean> = { 0, 1, OFF, AVG, INST }

Valid Modes: Modulated and Pulse Modes

CALCulate:STATe

Description: Set or return the measurement state of the selected channel. When ON, the channel performs measurements; when OFF, the channel is disabled and no measurements are performed.

Syntax: CALCulate[1|2|3|4]:STATe <Boolean>

Argument: <Boolean> = { 0, 1, OFF, ON }

Valid Modes: Any

CALCulate:UNITs

Description: Set or return units for the selected channel. For power sensors, voltage is calculated with reference to the sensor input impedance. Note that for ratiometric results, logarithmic units will always return dBr (dB relative) while linear units return percent.

Syntax: CALCulate[1|2|3|4]:UNITs <character data>

Argument: <character data> = {DBMw, Watts, Volts, DBV, DBMV, DBUV}

Valid Modes: Any

1.5.4 DISPlay Subsystem

The DISPlay group of commands is used to control the selection and presentation of textual, graphical and TRACe measurements.

DISPlay:CLEar

Description: Clear display traces and all data buffers. Clears averaging filters to empty. Does NOT clear errors.

Syntax: DISPlay:CLEar

Argument: None

Valid Modes: All

DISPlay:ENVELOPE[:STATe]

Description: Enable or disable the Envelope mode.

Syntax: DISPlay:ENVELOPE <Boolean>

Argument: <Boolean> = { 0, 1, ON, OFF }

Valid Modes: Pulse and Modulated

DISPlay:MODUlated:TIMEBASE

Description: Set or return the Modulated Mode timebase in seconds/division. The PMX40 has fixed timebase settings in a 1-2-5 sequence. If the argument does not match one of these settings, it will be forced to the next highest entry.

Optional units: minutes.

Syntax: DISPlay:MODUlated:TIMEBASE <numeric_value>

Argument: <numeric_value> = 10e-9 to 10 s, (1-2-5 sequence), 30 s, 1, 2, 5, 10, 30, 60 min

Valid Modes: Modulated Mode

Note: There are separate time bases for the modulated mode and the pulse mode. The arguments selected are saved and restored independently by mode.

DISPlay:MODUlated:TSPAN

Description: Set or return the horizontal time span of the display in modulated mode. Time span = 10* Time/Division.

Optional units: minutes.

Syntax: DISPlay:MODUlated:TSPAN <numeric_value>

Argument: <numeric_value> = 100e-9 to 100 s in a 1-2-5 sequence, 300 s, 10, 20, 50, 100, 300, 600 min

Valid Modes: Modulated mode

Note There are separate time bases for the modulated mode and the pulse mode. The arguments selected are saved and restored independently by mode.

DISPlay:PULSe:TIMEBASE

Description: Set or return the Pulse Mode timebase in seconds/division. The PMX40 has fixed timebase settings in a 1-2-5 sequence. If the argument does not match one of these settings, it will be forced to the next highest entry.

Optional units: minutes.

Syntax: DISPlay:PULSe:TIMEBASE <numeric_value>

Argument: <numeric_value> = 5e-9 to 50e-3, (1-2-5 sequence) 60 min Valid Modes:
Pulse Mode

Note There are separate time bases for the modulated mode and the pulse mode. The arguments selected are saved and restored independently by mode.

DISPlay:PULSe:TSPAN

Description: Set or return the horizontal time span of the display in pulse mode. Time span = 10* Time/Division.

Optional units: minutes.

Syntax: DISPlay:PULSe:TSPAN <numeric_value>

Argument: <numeric_value> = 50e-9 to 500e-3 in a 1-2-5 sequence,
600 min

Valid Modes: Pulse mode

Note There are separate time bases for the modulated mode and the pulse mode. The arguments selected are saved and restored independently by mode.

DISPlay:[TEXt:]LIN:RESolution

Description: Set or return the display resolution for linear power and voltage readings. The number of significant digits displayed is equal to the argument. This command also sets the resolution of measurements returned in remote mode.

Syntax: DISPlay:[TEXt:]LIN:RESolution <numeric_value>

Argument: <numeric_value> = 3 to 5

Valid Modes: All

DISPlay:[TEXt:]LOG:RESolution

Description: Set or return the display resolution for logarithmic power and voltage readings. The number of decimal places displayed is equal to the argument. This command also sets the resolution of measurements returned in remote mode.

Syntax: DISPlay:[TEXt:]LOG:RESolution <numeric_value>

Argument: <numeric_value> = 0 to 3

Valid Modes: All

DISPlay:TRACe:HOFFSet

Description: Set or return the statistical mode horizontal scale offset in dB. The offset value will appear at the leftmost edge of the scale with units dBr (decibels relative).

Syntax: DISPlay:TRACe[1|2|3|4]:HOFFSet <numeric_value>

Argument: <numeric_value> = -50.00 to +50.00

Valid Modes: Statistical

DISPlay:TRACe:HSCALe

Description: Set or return the statistical mode horizontal scale in dB/Div.

Syntax: DISPlay:TRACe[1|2|3|4]:HSCALe <numeric_value>

Argument: <numeric_value> = 0.1 to 5.0 in a 1-2-5 sequence Valid Modes:
Statistical

DISPlay:TRACe:VCENTer

Description: Set or return the power or voltage level of the horizontal centerline of the graph for the specified channel in channel units. If a change in the vertical scale causes the center maximum value to be exceeded, the center will be forced to the maximum value for the new range.

Syntax: DISPlay:TRACe[1|2|3|4]:VCENTer <numeric_value>

Argument: <numeric_value> = -200.00 to +200.00 for dBm units
+/- 10,000 times the vertical scale power/div for watts units
+/- 10,000 times the vertical scale volts/div for volts units

Valid Modes: All

DISPlay:TRACe:VSCALe

Description: Set or return the power or voltage vertical sensitivity of the trace display in channel units.

Syntax: DISPlay:TRACe[1|2|3|4]:VSCALe <numeric_value>

Argument: Units = dBm, dBV, dBmV, dBuV , <numeric_value> = range in dB/division
Units = watts <numeric_value> = range in watts/division
Units = volts <numeric_value> = range in volts/division

Valid Modes: All

1.5.5 FETCh Queries

The FETCh? group of queries is used to return specific measurement data from a measurement cycle that has been INITiated and is complete or free-running. FETCh? performs the data output portion of the measurement. FETCh? does not start a new measurement, so a series of FETCh? queries may be used to return more than one set of processed measurements from a complete set of acquired data. FETCh? usually returns the current value of measurements, and should be used anytime free running data acquisition is taking place (INITiate:CONTinuous ON).

FETCh:ARRay:AMEAsure:POWer?

Description: Returns an array of the current automatic amplitude measurements performed on a periodic pulse waveform. Measurements performed are: peak amplitude during the pulse, average amplitude over a full cycle of the pulse waveform, average amplitude during the pulse, IEEE top amplitude, IEEE bottom amplitude, overshoot, and droop. Units are the same as the channel's units. Note the pulse overshoot is returned in dB for logarithmic channel units, and percent for all other units. Also, the pulse "ON" interval used for peak and average calculations is defined by the SENSe:PULSe:STARTGT and :ENDGT time gating settings. A full pulse (rise and fall) must be visible on the display to make average and peak pulse power measurements, and a full cycle of the waveform must be visible to calculate average cycle amplitude.

Syntax: FETCh[1|2|3|4]:ARRay:AMEAsure:POWer?

Returns: CC1, PulsePeak, CC2, PulseCycleAvg, CC3, PulseOnAvg, CC4, IEEE Top, CC5, IEEE Bot, CC6, Overshoot, CC7, Droop.

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse Mode only.

Restrictions: Timebase must be set appropriately to allow measurements (see above)

FETCh:ARRay:AMEAsure:STATistical?

Description: Returns an array of the current automatic statistical measurements performed on a sample population. Measurements performed are: long term average, peak and minimum amplitude, peak-to-average ratio, amplitude at the CCDF percent cursor, statistical percent at the CCDF power cursor, and the sample population size in samples. Note the peak-to-average ratio is returned in dB for logarithmic channel units, and percent for all other channel units.

Syntax: FETCh[1|2|3|4]:ARRay:AMEAsure:STATistical?

Returns: CC1, Pavg, CC2, Ppeak, CC3, Pmin, CC4, PkToAvgRatio, CC5, CursorPwr, CC6, CursorPct, CC7, Sample-Count

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Statistical Mode only

FETCh:ARRay:AMEAsure:TIMe?

Description: Returns an array of the current automatic timing measurements performed on a periodic pulse waveform. Measurements performed are: the frequency, period, width, offtime and duty cycle of the pulse waveform, and the risetime and falltime of the edge transitions. For each of the measurements to be performed, the appropriate items to be measured must be visible on the display in GRAPH mode. Pulse frequency, period, offtime and duty cycle measurements require that an entire cycle of the pulse waveform (minimum of three edge transitions) be present. Pulse width measurement requires that at least one full pulse is visible and is most accurate if the pulse width is at least 0.4 divisions (20 pixels). Risetime and falltime measurements require that the edge being measured is visible and will be most accurate if the transition takes at least 0.1 divisions (5 pixels). It is always best to have the power meter set on the fastest timebase possible that meets the edge visibility restrictions. Set the trace averaging as high as practical to reduce fluctuations and noise in the pulse timing measurements. Note that the timing of the edge transitions is defined by the settings of the SENSe:PULSe:DISTal, :MESIal and :PROXimal settings; see the descriptions for those commands. Units are the same as the channel's units.

Syntax: FETCh[1|2|3|4]:ARRay:AMEAsure:TIMe?

Returns: CC1, PulseFreq, CC2, PulsePeriod, CC3, PulseWidth, CC4, Offtime, CC5, DutyCycle, CC6, Risetime, CC7, Falltime, CC8, EdgeDly, CC9, Skew in seconds

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse Mode only

Restrictions: Timebase must be set appropriately to allow measurements (see above)

FETCh:ARRay:CW:POWer?

Description: Returns the current average, maximum, minimum powers or voltages and the peak-to-average ratio of the specified channel. Units are the same as the channel's units. Note the peak-to-average ratio is returned in dB for logarithmic channel units, and percent for all other channel units.

Note that the values for maximum and minimum power will depend on the peak hold mode; see the description of the CALCulate:PKHLD command for details.

Syntax: FETCh[1|2|3|4]:ARRay:CW:POWer?

Returns: CC1, Pavg, CC2, Pmax, CC3, Pmin, CC4, PkToAvgRatio

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse and Modulated

FETCh:ARRay:MARKer:POWer?

Description: Returns an array of the current marker measurements for the specified channel. The array consists of the average, maximum, and minimum power and peak-to-average ratio

between the two markers, powers at both markers, and the ratio of the two markers. Note the peak-to-average ratio and marker ratio are returned in dB for log units, and percent for linear units.

Syntax: FETCh[1|2|3|4]:ARRay:MARKer:POWer?

Returns: CC1, Pavg, CC2, Pmax, CC3, Pmin, CC4, PkToAvgRatio, CC5, Pwr@Marker1, CC6, Pwr@Marker2, CC7, Mrk1/Mrk2 ratio

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse and Modulated

FETCh:CW:POWer?

Description: Return current average amplitude reading in channel units.

Syntax: FETCh[1|2|3|4]:CW:POWer?

Returns: CC, average power (watts, dBm) or average voltage (volts, dBv)

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:INTERval:AVERage?

Description: For the specified channel, return the average power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: FETCh[1|2|3|4]:INTERval:AVERage?

Returns: CC, average power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:INTERval:MAXFilt?

Description: For the specified channel, return the maximum filtered power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: FETCh[1|2|3|4]:INTERval:MAXFilt?

Returns: CC, maximum filtered power or voltage between the markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:INTERval:MINFilt?

Description: For the specified channel, return the minimum filtered power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: FETCh[1|2|3|4]:INTERval:MINFilt?

Returns: CC, minimum filtered power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:INTERval:MAXimum?

Description: For the specified channel, return the maximum instantaneous power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: FETCh[1|2|3|4]:INTERval:MAXimum?

Returns: CC, maximum instantaneous power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:INTERval:MINimum?

Description: For the specified channel, return the minimum instantaneous power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: FETCh[1|2|3|4]:INTERval:MINimum?

Returns: CC, minimum instantaneous power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:INTERval:PKAVG?

Description: For the specified channel, return the peak-to-average ratio of the power or voltage between marker 1 and marker 2. The units are dB for logarithmic channel units or percent for linear channel units.

Syntax: FETCh[1|2|3|4]:INTERval:PKAVG?

Returns: CC, peak-to-average ratio of power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

FETCh:MARKer:AVERage?

Description: For the specified channel and marker, return the average power or voltage at the marker. The units are the same as the specified channel.

Syntax: FETCh[1|2|3|4]:MARKer[1|2]:AVERage?

Returns: CC, average power or voltage at marker

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated.

FETCh:MARKer:CURsor:PERcent?

Description: Return the CCDF cursor y-axis position in percent with respect to the value set by MARKer:POSItion:POWer (CCDF cursor mode is Power Ref). If CCDF cursor mode is Percent, returns user setting. See MARKer:POSItion:POWer and MARKer:POSItion:PERcent. Refer to the PMX40 Instruction Manual Statistical Mode Automatic Measurements section for more information.

Syntax: FETCh[1|2|3|4]:MARKer:CURsor:PERcent?

Returns: CC, percent CCDF

Where CC is the measurement condition code.

Valid Modes: Statistical.

FETCh:MARKer:CURsor:POWer?

Description: Return the CCDF cursor x-axis position in relative power with respect to the value set by MARKer:POSItion:PERcent (CCDF cursor mode is Percent). If CCDF cursor mode is Power Ref, returns user setting. See MARKer:POSItion:POWer and MARKer:POSItion:PERcent. Refer to the PMX40 Instruction Manual Statistical Mode Automatic Measurements section for more information.

Syntax: FETCh[1|2|3|4]:MARKer:CURsor:POWer?

Returns: CC, relative power (dBr) CCDF

Where CC is the measurement condition code.

Valid Modes: Statistical

FETCh:MARKer:DELTa?

Description: For the specified channel return the difference between MK1 and MK2. The units will be the same as marker units.

Syntax: FETCh[1|2|3|4]:MARKer:DELTa?

Returns: CC, (MK1 – MK2)

Where CC is the measurement condition code.

Valid Modes: All

FETCh:MARKer:MAXimum?

Description: For the specified channel and marker, return the maximum power or voltage at the marker. The units are the same as the specified channel.

Syntax: FETCh[1|2|3|4]:MARKer[1|2]:MAXimum?

Returns: CC, maximum power or voltage at marker

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated.

FETCh:MARKer:MINimum?

Description: For the specified channel and marker, return the minimum power or voltage at the marker. The units will be the same as the specified channel.

Syntax: FETCh[1|2|3|4]:MARKer[1|2]:MINimum?

Returns: CC, minimum power or voltage at marker

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated.

FETCh:MARKer:RATio?

Description: For the specified channel return the ratio of MK1 to MK2. The units will be dB for logarithmic units or percent for linear units.

Syntax: FETCh[1|2|3|4]:Marker:RATio?

Returns: CC, MK1/MK2

Where CC is the measurement condition code.

Valid Modes: All

FETCh:MARKer:RDELTa?

Description: For the specified channel return the difference between MK2 and MK1. The units will be the same as marker units.

Syntax: FETCh[1|2|3|4]:MARKer:RDELTa?

Returns: CC, MK2-MK1

Where CC is the measurement condition code.

Valid Modes: All

FETCh:MARKer:RRATio?

Description: For the specified channel return the ratio of MK2 to MK1. The units will be dB for logarithmic units or percent for linear units.

Syntax: FETCh[1|2|3|4]:MARKer:RRATio?

Returns: CC, MK2/MK1

Where CC is the measurement condition code.

Valid Modes: All

FETCh#:MBUF:DATa:AVErage?

Description: Query the power meter for a list of average power measurements that were captured during the last SENSe:MBUF:ACQuire? query. The array will consist of <count> measurement values beginning at measurement number <index>.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:AVErage? ? <index>,<count>

Returns: Avg(index), Avg(index+1), Avg(index+2)… Avg(index+count-1).

Valid Modes: Pulse

FETCh#:MBUF:DATa:AVErage:ALL?

Description: Query the power meter for all buffered average power measurements on the channel specified that were captured during the last SENSe:MBUF:ACQuire? query.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:AVErage:ALL?

Returns: Avg(index), Avg(index+1), Avg(index+2)…

Valid Modes: Pulse

FETCh#:MBUF:DATa:DURation?

Description: Query the power meter for a list of duration measurements that were captured during the last SENSe:MBUF:ACQuire? query. The array will consist of <count> measurement values beginning at measurement number <index>.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:DURation? <index>,<count>

Returns: Duration(index), Duration(index+1), Duration(index+2)… Duration(index+count-1).

Valid Modes: Pulse

FETCh#:MBUF:DATa:DURation:ALL?

Description: Query the power meter for all buffered durations on the channel specified that were captured during the last SENSe:MBUF:ACQuire? query.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:DURation:ALL?

Returns: Duration(index), Duration(index+1), Duration(index+2)….

Valid Modes: Pulse

FETCh#:MBUF:DATa:MAXimum?

Description: Query the power meter for a list of maximum power measurements that were captured during the last SENSe:MBUF:ACQuire? query. The array will consist of <count> measurement values beginning at measurement number <index>.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:MAXimum? <index>,<count>

Returns: Max(index), Max(index+1), Max(index+2)…Max(index+count-1).

Valid Modes: Pulse

FETCh#:MBUF:DATa:MAXimum:ALL?

Description: Query the power meter for all buffered maximum power measurements on the channel specified that were captured during the last SENSe:MBUF:ACQuire? query.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:MAXimum:ALL?

Returns: Max(index), Max(index+1), Max(index+2)…

Valid Modes: Pulse

FETCh#:MBUF:DATa:MINimum?

Description: Query the power meter for a list of minimum power measurements that were captured during the last SENSe:MBUF:ACQuire? query. The array will consist of <count> measurement values beginning at measurement number <index>.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:MINimum? <index>,<count>

Returns: Min(index), Min(index+1), Min(index+2)… Min(index+count-1).

Valid Modes: Pulse

FETCh#:MBUF:DATa:MINimum:ALL?

Description: Query the power meter for all buffered minimum power measurements on the channel specified that were captured during the last SENSe:MBUF:ACQuire? query.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:MINimum:ALL?

Returns: Min(index), Min(index+1), Min(index+2)…

Valid Modes: Pulse

FETCh#:MBUF:DATa:STARt?

Description: Query the power meter for a list of start times in seconds that were captured during the last SENSe:MBUF:ACQuire? query. The array will consist of <count> measurement values beginning at measurement number <index>.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:STARt? <index>,<count>?

Returns: Start_Time(index), Start_Time(index+1), Start_Time(index+2)…Start_Time(index+count-1)

Valid Modes: Pulse

FETCh#:MBUF:DATa:STARt:ALL?

Description: Query the power meter for all buffered start times in seconds on the channel specified that were captured during the last SENSe:MBUF:ACQuire? query.

Syntax: FETCh[1|2|3|4]:MBUF:DATa:STARt:ALL?

Returns: Start_Time(index), Start_Time(index+1), Start_Time(index+2)…

Valid Modes: Pulse

FETCh:TEMPerature:AVERage?

Description: Return the average internal temperature of the specified power sensor.

Syntax: FETCh[1|2|3|4]:TEMPerature:AVERage?

Returns: CC, Sensor[1|2|3|4] Average Temp in degrees C Where CC is the measurement condition code.

Valid Modes: RTP Sensors only

FETCh:TEMPerature:CURRent?

Description: Return the current internal temperature of the specified power sensor.

Syntax: FETCh[1|2|3|4]:TEMPerature:CURRent?

Returns: CC, Sensor[1|2|3|4] Temp in degrees C

Where CCn is the measurement condition code.

Valid Modes: RTP Sensors only

1.5.6 INITiate and ABORt Commands

The purpose of the INITiate group of commands is to start and control the process of data acquisition once a measurement has been configured. Depending on settings, the PMX40 RF Power Meter may be commanded to begin either a single measurement (INITiate:CONTinuous OFF) which stops when complete, or enter a “free-run” mode where data acquisition occurs continuously (INITiate:CONTinuous ON). The ABORt command terminates any operation in progress and prepares the instrument for an

INITiate command. In some operating modes, the INITiate commands do not actually start measurements, but rather arms a hardware trigger, which is then used to gate the actual measurements cycle.

ABORt

Description: Terminates any measurement in progress and resets the state of the trigger system. Note that ABORt will leave the measurement in a stopped condition with all current measurements cleared, and forces INITiate:CONTinuous to OFF.

Syntax: ABORt

Argument: None

Valid Modes: Any

INITiate:CONTinuous

Description: Set or return the data acquisition mode for single or free-run measurements. If INITiate:CONTinuous is set to ON, the PMX40 immediately begins taking measurements (Modulated, CW and Statistical Modes), or arms its trigger and takes a measurement each time a trigger occurs (Pulse Mode). If set to OFF, the measurement will begin (or be armed) as soon as the INITiate command is issued, and will stop once the measurement criteria (averaging, filtering or sample count) has been satisfied. Note that INITiate:IMMediate and READ commands are invalid when INITiate:CONTinuous is set to ON; however, by convention this situation does not result in a SCPI error.

Syntax: INITiate:CONTinuous <Boolean>

Argument: <Boolean> = { 0, 1, OFF, ON }

Valid Modes: Any

INITiate[:IMMediate[:ALL]]

Description: Starts a single measurement cycle when INITiate:CONTinuous is set to OFF. In Modulated Mode, the measurement will complete once the power has been integrated for the full FILTer time. In Pulse Mode, enough trace sweeps must be triggered to satisfy the AVERaging setting. In Statistical Mode, acquisition stops once the terminal condition(s) are met. In each case, no reading will be returned until the measurement is complete. This command is not valid when INITiate:CONTinuous is ON, however, by convention this situation does not result in a SCPI error.

Syntax: INITiate[:IMMediate[:ALL]]

Argument: None

Valid Modes: Any

Restrictions: INITiate:CONTinuous must be OFF

1.5.7 MARKer Subsystem

The MARKer group of commands is used to configure and locate measurement markers (cursors) at specific points on the processed measurement waveform. FETCH? or READ? queries may then be used to retrieve measurements at the two markers and in the interval between them. Markers are used in Pulse Mode to perform measurements at or between two time offsets relative to the trigger, and in

Statistical Mode to measure the power at a particular statistical percent, or the percent at a specified power level. In Pulse Mode, the markers can only be placed on the visible portion of the trace (as defined by the timebase and trigger delay settings), while Statistical Mode markers may be placed at any power or percent value and will still return readings.

MARKer:POSItion:PERcent

Description: Set or return the percent probability (y-axis-position) of the CCDF cursor. Changing this setting will force the CCDF cursor mode to Percent for display purposes. READ:MARKer:CURsor:POWer? and FETCh:MARKer:CURsor:POWer? commands are referenced to this value.

Syntax: MARKer:POSItion:PERcent? <numeric_value>

Argument: <numeric_value> = 0.000 to 100.000

Valid Modes: Statistical Modes only

MARKer:POSItion:POWer

Description: Set or return the cumulative relative power (x-axis-position) of the CCDF cursor in dBr. Changing this setting will force the CCDF cursor mode to Power Ref for display purposes. READ:MARKer:CURsor:PERcent? and FETCh:MARKer:CURsor:PERcent? commands are referenced to this value.

Syntax: MARKer:POSItion:POWer? <numeric_value>

Argument: <numeric_value> = -100.000 to 100.000Valid

Modes: Statistical Modes only

MARKer:POSItion:TIMe

Description: Set or return the time (x-axis-position) of the selected marker relative to the trigger. Note that time markers must be positioned within the time limits of the trace window in the graph display. If a time outside of the display limits is entered, the marker will be placed at the first or last time position as appropriate.

Syntax: MARKer[1|2]:POSItion:TIMe <numeric_value>

Argument: <numeric_value> = display±time in seconds (see restrictions) Valid

Modes: Pulse and Modulated Modes

Restrictions: For zero delay trigger position in the center of the display , the following relationship must be satisfied: TrigDly - (5*time/div) < MarkerTime < TrigDly + (5*time/div) where the timebase setting is time/div

1.5.8 MEASure Queries

The MEASure group of commands is used to acquire data using a set of high level instructions. They are structured to allow the user to trade off fine control of the measurement process for easy operability. MEASure? provides a complete capability where the power meter is configured, a measurement taken, and results returned in one operation. The instrument is set to a basic, predefined measurement state with little user intervention necessary or possible. Sometimes, more precise control of measurement is required. In these cases, MEASure? should not be used. Rather, a sequence of configuration commands, generally from the CALCulate and SENSe groups should be used to set up the instrument for the

measurement, then READ? or FETCH? commands are used to return the desired measurement data in a specific format.

MEASure:POWer?

Description: Return average power using a default instrument configuration in Modulated Mode and dBm units. Instrument remains stopped after a measurement.

Syntax: Measure[1|2|3|4]:POWer?

Returns: CC, Average power in dBm

Where CC is the measurement condition code.

Valid Modes: Automatically set to Modulated Mode before measurement

MEASure:VOLTage?

Description: Return average voltage using a default instrument configuration in Modulated Mode and volts units. Instrument remains stopped after a measurement.

Syntax: MEASure[1|2|3|4]:VOLTage?

Returns: CC, Average voltage in linear volts

Where CC is the measurement condition code.

Valid Modes: Automatically set to Modulated Mode before measurement

1.5.9 MEMory Subsystem

The MEMory group of commands is used to review frequency dependent offset (FDOF) tables for the sensors.

MEMory:SNSR:CFFAST?

Description: Return the sensor high bandwidth (FAST) frequency cal-factor table.

Syntax: MEMory:SNSR[1|2|3|4]:CFFAST?

Returns: None, query only.

Valid Modes: Any

MEMory:SNSR:CFSLOW?

Description: Return the sensor low bandwidth (SLOW) frequency cal-factor table.

Syntax: MEMory:SNSR[1|2|3|4]:CFSLOW?

Argument: None, query only.

Valid Modes: Any

1.5.10 READ Queries

The purpose of the READ? group of queries is to initiate a measurement cycle, acquire data, and return specific measurement data. READ? performs the initiation, data acquisition, postprocessing, and data output portions of the measurement. READ? is equivalent to ABORting any operation in progress,

INITiating a new measurement, then FETChing the data when it is ready. READ? generally does not return data unless acquisition is complete. Since READ? INITiates a new measurement every time it is issued, READ? queries should not be used for free running data acquisition (INITiate:CONTinuous ON) - in this case, use FETCh queries instead. For CW and Modulated Modes, the measurement is generally considered complete when the integration filter (see SENSe:FILTer) is filled. In Pulse Mode, the measurement is considered complete when all the number of complete traces specified by the SENSe:AVERage command have been acquired and averaged together. In Statistical Mode, the measurement is considered complete when the number of samples specified by TRIGger:CDF:COUNt has been gathered.

READ:ARRay:AMEAsure:POWer?

Description: Returns an array of the current automatic amplitude measurements performed on a periodic pulse waveform. Measurements performed are: peak amplitude during the pulse, average amplitude over a full cycle of the pulse waveform, average amplitude during the pulse, IEEE top amplitude, IEEE bottom amplitude, overshoot, and droop. Units are the same as the channel's units. Note the pulse overshoot is returned in dB for logarithmic channel units, and percent for all other units. Also, the pulse "ON" interval used for peak and average calculations is defined by the SENSe:PULSe:STARTGT and :ENDGT time gating settings. A full pulse (rise and fall) must be visible on the display to make average and peak pulse power measurements, and a full cycle of the waveform must be visible to calculate average cycle amplitude.

Syntax: READ[1|2|3|4]:ARRay:AMEAsure:POWer?

Returns: CC1, PulsePeak, CC2, PulseCycleAvg, CC3, PulseOnAvg, CC4, IEEE Top, CC5, IEEE Bot, CC6, Overshoot, CC7, Droop

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse Mode only

Restrictions: Timebase must be set appropriately to allow measurements (see above)

READ:ARRay:AMEAsure:STATistical?

Description: Returns an array of the current automatic statistical measurements performed on a sample population. Measurements performed are: long term average, peak and minimum amplitude, peak-to-average ratio, amplitude at the CCDF cursor, statistical percent at the CCDF cursor, and the sample population size in samples. Note the peak-to-average ratio is returned in dB for logarithmic channel units, and percent for all other channel units.

Syntax: READ[1|2|3|4]:ARRay:AMEAsure:STATistical?

Returns: CC1, Pavg, CC2, Ppeak, CC3, Pmin, CC4, PkToAvgRatio, CC5, CursorPwr, CC6, CursorPct, CC7, Sample-Count

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Statistical Mode only

READ:ARRay:AMEAsure:TIMe?

Description: Returns an array of the current automatic timing measurements performed on a periodic pulse waveform. Measurements performed are: the frequency, period, width, offtime and duty cycle of the pulse waveform, and the risetime and falltime of the edge

transitions. For each of the measurements to be performed, the appropriate items to be measured must be visible on the display in GRAPH mode. Pulse frequency, period, offtime and duty cycle measurements require that an entire cycle of the pulse waveform (minimum of three edge transitions) be present. Pulse width measurement requires that at least one full pulse is visible, and is most accurate if the pulse width is at least 0.4 divisions (20 pixels). Risetime and falltime measurements require that the edge being measured is visible, and will be most accurate if the transition takes at least 0.1 divisions (5 pixels). It is always best to have the power meter set on the fastest timebase possible that meets the edge visibility restrictions. Set the trace averaging as high as practical to reduce fluctuations and noise in the pulse timing measurements. Note that the timing of the edge transitions is defined by the settings of the SENSe:PULSe:DISTal, :MESIal and :PROXimal settings; see the descriptions for those commands. Units are the same as the channel's units.

Syntax: READ[1|2|3|4]:ARRay:AMEAsure:TIMe?

Returns: CC1, PulseFreq, CC2, PulsePeriod, CC3, PulseWidth, CC4, Offtime, CC5, DutyCycle, CC6, Risetime, CC7, Falltime, CC8, EdgeDly, CC9, Skew in Hz

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse Mode only.

Restrictions: Timebase must be set appropriately to allow measurements (see above)

READ:ARRay:CW:POWer?

Description: Returns the current average, maximum, minimum powers or voltages and the peak-to-average ratio of the specified channel. Units are the same as the channel's units. Note the peak-to-average ratio and marker ratio are returned in dB for logarithmic channel units, and percent for all other channel units.

Note that the values for maximum and minimum power will depend on the peak hold mode; see the description of the CALCulate:PKHLD command for details.

Syntax: READ[1|2|3|4]:ARRay:CW:POWer?

Returns: CC1, Pavg, CC2, Pmax, CC3, Pmin, CC4, PkToAvgRatio

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse and Modulated

READ:ARRay:MARKer:POWer?

Description: Returns an array of the current marker measurements for the specified channel. The array consists of the average, maximum, and minimum power and peak-to-average ratio *between* the two markers, powers *at both* markers, and the ratio of the two markers. Note the peak-to-average ratio and marker ratio are returned in dB for log units, and percent for linear units.

Syntax: READ[1|2|3|4]:ARRay:MARKer:POWer?

Returns: CC1, Pavg, CC2, Pmax, CC3, Pmin, CC4, PkToAvgRatio, CC5, Pwr@Marker1, CC6, Pwr@Marker2, CC7, Mrk1/Mrk2 ratio

Where the CCn's are the measurement condition codes for each measurement.

Valid Modes: Pulse and Modulated

READ:CW:POWer?

Description: Return current average amplitude reading in channel units.

Syntax: READ[1|2|3|4]:CW:POWer?

Returns: CC, Average power (watts, dBm) or average voltage (volts, dBv)
Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:INTERval:AVERage?

Description: For the specified channel, return the average power or voltage between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: READ[1|2|3|4]:INTERval:AVERage?

Returns: CC, average power or voltage between markers
Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:INTERval:MAXFilt?

Description: For the specified channel, return the maximum filtered power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: READ[1|2|3|4]:INTERval:MAXFilt?

Returns: CC, maximum filtered power or voltage between the markers
Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:INTERval:MINFilt?

Description: For the specified channel, return the minimum filtered power or voltage in the time interval between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: READ[1|2|3|4]:INTERval:MINFilt?

Returns: CC, minimum filtered power or voltage between markers
Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:INTERval:MAXimum?

Description: For the specified channel, return the maximum instantaneous power or voltage between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: READ[1|2|3|4]:INTERval:AVERage?

Returns: CC, maximum instantaneous power or voltage between markers
Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:INTERval:MINimum?

Description: For the specified channel, return the minimum instantaneous power or voltage between marker 1 and marker 2. The units will be the same as the specified channel.

Syntax: READ[1|2|3|4]:INTERval:MINimum?

Returns: CC, minimum instantaneous power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:INTERval:PKAVG?

Description: For the specified channel, return the peak-to-average ratio of the power or voltage between marker 1 and marker 2. The units are dB for logarithmic channel units or percent for linear channel units.

Syntax: READ[1|2|3|4]:INTERval:PKAVG?

Returns: CC, peak-to-average ratio of power or voltage between markers

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:MARKer:AVERage?

Description: For the specified channel and marker, return the average power or voltage at the marker. The units are the same as the specified channel.

Syntax: READ[1|2|3|4]:MARKer[1|2]:AVERage?

Returns: CC, average power or voltage at marker

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated

READ:MARKer:DELTa?

Description: For the specified channel return the difference between MK1 and MK2. The units will be the same as marker units.

Syntax: READ[1|2|3|4]:MARKer:DELTa?

Returns: CC, (MK1 – MK2)

Where CC is the measurement condition code.

Valid Modes: All

READ:MARKer:MAXimum?

Description: For the specified channel and marker, return the maximum power or voltage at the marker. The units are the same as the specified channel.

Syntax: READ[1|2|3|4]:MARKer[1|2]:MAXimum?

Returns: CC, maximum power or voltage at marker

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated.

READ:MARKer:MINimum?

Description: For the specified channel and marker, return the minimum power or voltage at the marker. The units will be the same as the specified channel.

Syntax: READ[1|2|3|4]:MARKer[1|2]:MINimum?

Returns: CC, minimum power or voltage at marker

Where CC is the measurement condition code.

Valid Modes: Pulse and Modulated.

READ:MARKer:CURsor:PERcent?

Description: Return the CCDF cursor y-axis position in percent with respect to the value set by MARKer:POSItion:POWer (CCDF cursor mode is set to Power Ref). If CCDF cursor mode is Percent, returns user setting. See MARKer:POSItion:POWer and MARKer:POSItion:PERcent. Refer to the PMX40 Instruction Manual Statistical Mode Automatic Measurements section for more information.

Syntax: READ[1|2|3|4]:MARKer:CURsor:PERcent?

Returns: CC, percent CCDF

Where CC is the measurement condition code.

Valid Modes: Statistical.

READ:MARKer:CURsor:POWer?

Description: Return the CCDF cursor x-axis position in relative power with respect to the value set by MARKer:POSItion:PERcent (CCDF cursor mode is set to Percent). If CCDF cursor mode is Power Ref, returns user setting. See MARKer:POSItion:POWer and MARKer:POSItion:PERcent. Refer to the PMX40 Instruction Manual Statistical Mode Automatic Measurements section for more information.

Syntax: READ[1|2|3|4]:MARKer:CURsor:POWer?

Returns: CC, relative power (dBr) CCDF

Where CC is the measurement condition code.

Valid Modes: Statistical

READ:MARKer:RATio?

Description: For the specified channel return the ratio of MK1 to MK2. The units will be dB for logarithmic units or percent for linear units.

Syntax: READ[1|2|3|4]:Marker:RATio?

Returns: CC, MK1/MK2

Where CC is the measurement condition code.

Valid Modes: All

READ:MARKer:RDELTa?

Description: For the specified channel return the difference between MK2 and MK1. The units will be the same as marker units.

Syntax: READ[1|2|3|4]:MARKer:RDELTa?

Returns: CC, (MK2-MK1)
Where CC is the measurement condition code.
Valid Modes: All

READ:MARKer:RRATio?

Description: For the specified channel return the ratio of MK2 to MK1. The units will be dB for logarithmic units or percent for linear units..
Syntax: READ[1|2|3|4]:MARKer:RRATio?
Returns: CC, MK2/MK1
Where CC is the measurement condition code.
Valid Modes: All

1.5.11 SENSe Subsystem

The purpose of the SENSe command subsystem is to directly configure device specific settings used to make measurements, generally parameters related to the RF power sensor and signal processing. The SENSe commands are used to configure the power meter for acquiring data. SENSe enables you to change measurement parameters such as filtering or averaging, sensor bandwidth, operating frequency and calfactors, and measurement gain or offset. The numeric suffix of the SENSe program mnemonic in the SENSe commands refers to a hardware measurement "channel" that is SENSe1 and SENSe2 represent the instrument's SENSOR 1 and SENSOR 2 signal paths, respectively. The SENSe commands generally DO NOT affect the data processing and display portion of the measurement (see the CALCulate subsystem).

SENSe:AVERage

Description: Set or return the number of traces averaged together to form the measurement result on the selected channel. Averaging can be used to reduce display noise on both the visible trace, and on marker and automatic pulse measurements. Trace averaging is a continuous process in which the measurement points from each sweep are weighted (multiplied) by an appropriate factor, and averaged into the existing trace data points. In this way, the most recent data will always have the greatest effect on the trace shape, and older measurements will be decayed at a rate determined by the averaging setting and trigger rate. Note that for timebase settings of 500 ns/div and faster, the PMX40 acquires samples using a technique called equivalent time or interleaved sampling. In this mode, not every pixel on the trace gets updated on each sweep, and the total number of sweeps needed to satisfy the AVERage setting will be increased by the sample interleave ratio of that particular timebase.
Syntax: SENSe[1|2|3|4]:AVERage <numeric_value>
Argument: <numeric_value> = Numeric_value from 1 to 16,384 (1 = no trace averaging)
Valid Modes: Pulse

SENSe:BANDwidth

Description: Set or return the sensor video bandwidth for the selected sensor or the trigger channel bandwidth if a trigger channel is selected. HIGH is the normal setting for most

measurements. The actual bandwidth is determined by the peak sensor model used. Use LOW bandwidth for additional noise reduction when measuring CW or signals with very low modulation bandwidth. If LOW bandwidth is used on signals with fast modulation, measurement errors will result because the sensor cannot track the fast changing envelope of the signal.

Syntax: SENSe[1|2|3|4]:BANDwidth <character data>

Argument: <character data> = { LOW, HIGH }

Valid Modes: All

SENSe:CORRection:CALFactor

Description: Set or return the frequency calfactor currently in use on the selected channel. Note setting a calfactor with this command will override the "automatic" frequency calfactor that was calculated and applied when the operating frequency was set, and setting the operating frequency will override this calfactor setting.

Syntax: SENSe[1|2|3|4]:CORRection:CALFactor <numeric_value>

Argument: <numeric_value> = –3.00 to 3.00 dB, units are optional

Valid Modes: All

SENSe:CORRection:DCYCle

Description: Set or return the duty cycle correction factor in percent currently in use on the selected channel. Syntax: SENSe[1|2|3|4]:CORRection:DCYCle

Argument: <numeric_value> = 0.01 to 100.00 Valid

Modes: CW Sensor (Modulated Mode)

SENSe:CORRection:FREQuency

Description: Set or return the RF frequency for the current sensor, and apply the appropriate frequency calfactor from the sensor's EEPROM table. Application of this calfactor cancels out the effect of variations in the flatness of the sensor's frequency response. If an explicit calfactor has been set, either manually or via the SENSe:CORRection:CALFactor command, entering a new frequency will override this calfactor and use only the "automatic" frequency calfactor.

Syntax: SENSe[1|2|3|4]:CORRection:FREQuency <numeric_value>

Argument: <numeric_value> = 1e6 to 110.0e9 Hz (actual sensor may have narrower range; range may depend on channel bandwidth setting)

Valid Modes: All

SENSe:CORRection:OFFSet

Description: Set or return a measurement offset in dB for the selected sensor. This is used to compensate for external couplers, attenuators or amplifiers in the RF signal path ahead of the power sensor.

Syntax: SENSe[1|2|3|4]:CORRection:OFFSet <numeric_value>

Argument: <numeric_value> = –200.000 to 200.000 dB units are optional.

Valid Modes: Any

SENSe:FILTer:STATe

Description: Set or return the current setting of the integration filter on the selected channel. OFF provides no filtering, and can be used at high signal levels when absolute minimum settling time is required. ON allows a user-specified integration time, from 2 milliseconds to 15 seconds (see SENSe:FILTer:TIMe command). Note that setting the filter time will force the state to ON. AUTO uses a variable amount of filtering, which is set automatically by the power meter based on the current signal level to a value that gives a good compromise between measurement noise and settling time at most levels.

Syntax: SENSe[1|2|3|4]:FILTer:STATe <character data>

Argument: <character data> = {OFF, ON, AUTO}

Valid Modes: Modulated

SENSe:FILTer:TIMe

Description: Set or return the current length of the integration filter on the selected channel. If the filter state is set to AUTO, querying the time will return -0.01, and if set to OFF, a time query will return 0.00. Note that setting the filter time will force the state to ON.

Syntax: SENSe[1|2|3|4]:FILTer:TIMe <numeric_value>

Argument: <numeric_value> = 0.002 to 16.000 in 2 millisecond increments Valid

Modes: Modulated

SENSe:MBUF:ACQuire?

Description: Starts a new measurement buffer acquisition. This command blocks communication until the number of measurements for each enabled channel is equal to count, a time out has occurred, or the buffer is overrun before it could be read into memory.

Syntax: SENSe:MBUF:ACQuire?

Returns: Stop Reason numeric code. 1 = Count Reached, 2 = Timed Out, 3 = Buffer Overrun.

Valid Modes: Pulse

SENSe:MBUF:DELay:END

Description: Set or return the time in seconds added to the detected end of a burst for analysis. Typically a negative value is used to exclude the falling edge of a burst from the measurement..

Syntax: SENSe:MBUF:DELay:END <numeric value>

Argument: <numeric value> = -10e-6 to 100e-3

Valid Modes: Pulse

SENSe:MBUF:DELay:STARt

Description: Set or return the time in seconds added to the detected beginning of a burst for analysis. Typically used to exclude the rising edge of a burst from the measurement.

Syntax: SENSe:MBUF:DELay:STARt <numeric value>

Argument: <numeric value> = -10e-6 to 100e-3

Valid Modes: Pulse

SENSe:MBUF:ENABled:FLAG

Description: Set or return the bit mask value indicating which MBUF arrays will be filled during a SENSe:MBUF:ACQuire? query.

Syntax: SENSe:MBUF:ENABled <numeric value>

Argument: <numeric value> = decimal value representing a 6-bit binary value used to enable/disable mask bits

BIT	VALUE	DEFINITION
5	32	Enable Sequence Capture
4	16	Enable Start Time Capture
3	8	Enable Duration Capture
2	4	Enable Min. Measurements
1	2	Enable Average Measurements
0	1	Enable Max. Measurements

Valid Modes: Pulse

SENSe:MBUF:GATe:MODe

Description: Set or return the gate mode used to define the start and end of a measurement time interval.

Syntax: SENSe:MBUF:GATe:MODe <character data>

Argument: <character data> = {BURst, MARKer, EXTGATE, PERiodic, EXTTRIG}

Valid Modes: Pulse

SENSe#:MBUF:MEASurements:AVAilable?

Description: Gets the number of measurements available in the power meter's internal buffer.

Syntax: SENSe[1|2|3|4]:MBUF:MEASurements:AVailable?

Returns: <numeric>

Valid Modes: Pulse

SENSe:MBUF:MEASurements:CLEar

Description: Clears all cached average, min, max, duration, start time, and sequence numbers.

Syntax: SENSe:MBUF:MEASurements:CLEar

Argument: None

Valid Modes: Pulse

SENSe:MBUF:PERiod

Description: Set or return the period in seconds, that each timed mode measurement is started. Refer to the "SENSe:MBUF:PERiod:DURation" command to set the duration of the measurement.

Syntax: SENSe:MBUF:PERiod <numeric value>

Argument: <numeric value> = 100e-9 to 10

Valid Modes: Pulse

SENSe:MBUF:PERiod:DURation

Description: Set or return the time duration in seconds, that samples are captured during each timed mode MBUF acquisiton.

Syntax: SENSe:MBUF:PERiod:DURation <numeric value>

Argument: <numeric value> = 0.001 to 1000

Valid Modes: Pulse

SENSe:MBUF:QUALifier:END

Description: Set or return the minimum amount of time in seconds power remains below the trigger point to be counted as the end of a burst.

Syntax: SENSe:MBUF:QUALifier:END <numeric value>

Argument: <numeric value> = 0.0 to 10e-6

Valid Modes: Pulse

SENSe:MBUF:QUALifier:STARt

Description: Set or return the minimum amount of time in seconds power remains above the trigger point to be counted as the beginning of a burst.

Syntax: SENSe:MBUF:QUALifier:STARt <numeric value>

Argument: <numeric value> = 0.0 to 10e-6

Valid Modes: Pulse

SENSe:MBUF:SESsion:COUNt

Description: Set or return the count of elements for the MBUF arrays during a SENSe:MBUF:ACQuire? query.

Syntax: SENSe:MBUF:SESsion:COUNt <numeric data>

Argument: <numeric data> = 1 to 999999

Valid Modes: Pulse

SENSe:MBUF:SESsion:TIMEout

Description: Set or return the maximum time in seconds for the MBUF session during a SENSe:MBUF:ACQuire? query.

Syntax: SENSe:MBUF:SESsion:TIMEout <numeric value>

Argument: <numeric value> = 1e-3 to 1e3

Valid Modes: Pulse

SENSe:MBUF:START:MODe

Description: Set or return the mode used to start acquiring MBUF entries.

Syntax: SENSe:MBUF:GATe:MODe <character data>

Argument: <character data> = {IMMediate, EXTEnable, EXTStart}

Valid Modes: Pulse

SENSe:PULSe:DISTal

Description: Set or return the pulse amplitude percentage, which is used to define the end of a rising edge or beginning of a falling edge transition. Typically, this is 90% voltage or 81% power relative to the "top level" of the pulse. This setting is used when making automatic pulse risetime and falltime calculations returned by READ:ARRay:AMEASure:POWer.

Syntax: SENSe[1|2|3|4]:PULSe:DISTal <numeric_value>

Argument: <numeric_value> = 50.00 to 100.00 Valid

Modes: Pulse Mode only

SENSe:PULSe:ENDGT

Description: Set or return the point on a pulse, which is used to define the end of the pulse's "active" interval. This point is defined in percent of the total pulse duration, with 0% corresponding to the midpoint of the rising edge, and 100% corresponding to the midpoint of the falling edge, as defined by the mesial setting. For most pulse "on" average power measurements, it is desirable to exclude the rising and falling intervals, and only measure power over the active portion of the pulse. This is often known as time gating, and is used for the automatic pulse measurements returned by READ:ARRay:AMEASure:POWer.

Syntax: SENSe[1|2|3|4]:PULSe:ENDGT <numeric_value>

Argument: <numeric_value> = 60.00 to 100.00 Valid

Modes: Pulse Mode only

SENSe:PULSe:MESIal

Description: Set or return the pulse amplitude percentage, which is used to define the midpoint of a rising or falling edge transition. Typically, this is 50% voltage or 25% power relative to the "top level" of the pulse. This setting is used when making automatic pulse width and duty cycle calculations returned by FETCh:ARRay:AMEASure:POWer.

Syntax: SENSe[1|2|3|4]:PULSe:MESIal <numeric_value>

Argument: <numeric_value> = 10.00 to 90.00

Valid Modes: Pulse Mode only

SENSe:PULSe:PROXimal

Description: Set or return the pulse amplitude percentage, which is used to define the beginning of a rising edge or end of a falling edge transition. Typically, this is 10% voltage or 1% power relative to the "top level" of the pulse. This setting is used when making automatic pulse risetime and falltime calculations returned by FETCh:ARRay:AMEASure:POWer.

Syntax: SENSe[1|2|3|4]:PULSe:PROXimal <numeric_value>

Argument: <numeric_value> = 0.00 to 50.00

Valid Modes: Pulse Mode only

SENSe:PULSe:STARTGT

Description: Set or return the point on a pulse, which is used to define the beginning of the pulse's "active" interval. This point is defined in percent of the total pulse duration, with 0% corresponding to the midpoint of the rising edge, and 100% corresponding to the midpoint of the falling edge, as defined by the mesial setting. For most pulse "on" average power measurements, it is desirable to exclude the rising and falling intervals, and only measure power over the active portion of the pulse. This is often known as time gating, and is used for the automatic pulse measurements returned by FETCh:ARRay:AMEASure:POWer.

Syntax: SENSe[1|2|3|4]:PULSe:STARTGT <numeric_value>

Argument: <numeric_value> = 0.00 to 40.00

Valid Modes: Pulse Mode only

SENSe:PULSe:UNIT

Description: Set or return the units for entering the pulse distal, mesial and proximal levels. If the function is set to VOLTS, the pulse transition levels will be defined as the specified percentage in voltage. If set to WATTS, the levels are defined in percent power. Many pulse measurements call for 10% to 90% voltage (which equates to 1% to 81% power) for risetime and falltime measurements, and measure pulse widths from the half-power (–3dB, 50% power, or 71% voltage) points.

Syntax: SENSe[1|2|3|4]:PULSe:UNIT <character data>

Argument: <character data> = {WATTS, VOLTS}

Valid Modes: Pulse Mode only

SENSe:SENSOR:TYPE?

Description: Returns the sensor type for the selected channel.

Syntax: SENSe[1|2|3|4]:SENSOR:TYPE?

Returns: <character data> = { NONE, CW, VOLT, PEAK }

Valid Modes: All

1.5.12 SYSTem Subsystem

The SYSTem group of commands is used to control system-level functions not directly related to instrument measurement performance. SYSTem commands are used to return error codes or messages from the power meter error queue, control hardware features (backlight and key beep), access the internal clock/calendar, and configure communication parameters for the GPIB and LAN interfaces.

SYSTem:BEEP[:ENABle]

Description: Set or return the status of the audible keyboard beeper.

Syntax: SYSTem:BEEP[:ENABle] <Boolean>

Argument: <Boolean> = { 0, 1, OFF, ON }

Valid Modes: Any

SYSTem:COMMunicate:GPIB:ADDRess

Description: Set or return the GPIB bus address.

Syntax: SYSTem:COMMunicate:GPIB:ADDRess <numeric_value>

Argument: <numeric_value> = 1 to 30

Valid Modes: Any

SYSTem:COMMunicate:LAN:ADDRess

Description: Set or return the IP address for the Ethernet port.

Syntax: SYSTem:COMMunicate:LAN:ADDRess <character data>

Argument: <character data> = instrument IP address in nnn.nnn.nnn.nnn (“dot decimal”) format

Valid Modes: Any for queries. DHCP/AutoIP must be disabled (OFF) to set the instrument IP address.

SYSTem:COMMunicate:LAN:DHCP

Description: Set or return the state of DHCP/AutoIP system for the Ethernet port.

If DHCP/AutoIP is enabled (1 | ON), the instrument will attempt to obtain its IP Address, Subnet Mask, and Default Gateway from a DHCP (dynamic host configuration protocol) server on the network. If no DHCP server is found, the instrument will select its own IP Address, Subnet Mask, and Default Gateway values using the “AutoIP” protocol.

If DHCP/AutoIP is disabled (0 | OFF), the instrument will use the IP Address, Subnet Mask, and Default Gateway values that have been entered by the user.

Syntax: SYSTem:COMMunicate:LAN:DHCP <Boolean>

Argument: <Boolean> = { 0, 1, OFF, ON }

Valid Modes: Any

SYSTem:ERRor[:NEXT]?

Description: Returns the next queued error code number followed by a quoted ASCII text string describing the error. Note that errors are stored in a "first-in-first-out" queue, so if more than one error has occurred, repeating this command will report the errors in the sequence they happened. The action of reading an error removes that error from the queue, so once the most recent error has been read, further queries will report a code of zero, and "No Error"

Syntax: SYSTem:ERRor[:NEXT]?

Returns: <numeric error code>, "QUOTED ERROR DESCRIPTION"

Valid Modes: Any

SYSTem:ERRor:CODE?

Description: Returns the next queued error code number. Note that errors are stored in a "first-in-first-out" queue, so if more than one error has occurred, repeating this command will report the error codes in the sequence they happened. The action of reading an error removes that error from the queue, so once the most recent error has been read, any more queries will report a code of zero.

Syntax: SYSTem:ERRor:CODE?

Returns: <numeric error code>

Valid Modes: Any

SYSTem:ERRor:COUNt?

Description: Returns the number of errors that currently exist in the error queue. A value of 0 means that there are no errors in the queue. Therefore, either no errors have occurred, or all errors have been read.

Syntax: SYSTem:ERRor:COUNt?

Returns: <numeric error code>

Valid Modes: Any

SYSTem:PRESet

Description: Set PMX40 default parameters. Equivalent to selecting "Initialize" from *Measure > Meas. Settings* on the PMX40 meter.

Syntax: SYSTem:PRESet

Argument: None

Valid Modes: Any

SYSTem:VERSion?

Description: Return the SCPI version compliance claimed.

Syntax: SYSTem:VERSion?

Returns: <character data> = Version Code as <year.version> YYYY.V (will return 1999.0)

Valid Modes: Any

1.5.13 TRACe Data Array Commands

The TRACe group of commands is used to control the output of an acquired measurement array, which appears as a display trace when the power meter is in Graph mode. The TRACe commands allow outputting a channel's entire internal display trace (501 measurement points) as one large array, or selecting and returning the array in smaller portions. These commands are useful for capturing the displayed waveform and importing it into a database on the host.

TRACe [:AVERage]]:DATA[:NEXT]?

Description: Return a delimited array of power or voltage pixel average values corresponding to all or a portion of the graph mode display trace for the selected channel. The Average trace is the data shown when envelope display mode is inactive. Note that the pixel values are returned without regard to display vertical scale and center settings. The array will consist of COUNT trace pixel values, beginning at pixel number INDEX, up to the last pixel of the trace (index = 500), and will be returned in channel units. The selected channel must be "ON" to return measurement data.

Syntax: TRACe[1|2|3|4][:AVERage]]:DATA[:NEXT]?

Returns: P(index), P(index+1), P(index+2),…P(index+count) or

V(index), V(index+1), V(index+2),…V(index+count)

Valid Modes: All

TRACe:COUNt

Description: Set or return the number of trace points, which will be returned each time the TRACe:DATA? query is issued. At the completion of each read, INDEX is automatically incremented by COUNT. If COUNT is set to a number greater than the number of points remaining in the trace, the array will be truncated. Setting COUNT to 501 (and INDEX to zero each time) will return the entire trace array.

Syntax: TRACe[1|2|3|4]:COUNt <numeric_value>

Argument: <numeric_value> = 1 to 501

Valid Modes: All

TRACe:INDEX

Description: Set or return the array index for the first trace point to be returned next time the TRACe:DATA? query is issued. Index 0 is the start of the trace buffer, and corresponds to the leftmost pixel on the graph display. Index 500 is the last point, and is the rightmost pixel. Each time a block of data is read, INDEX is automatically incremented by the COUNT value, so the full array can be split up into blocks of manageable size and read

with successive TRACe:DATA? queries. INDEX must be reset to zero for each new trace that is to be dumped, whether or not all the points have been read.

Syntax: TRACe[1|2|3|4]:INDEX <numeric_value>

Argument: <numeric_value> = 0 to 500

Valid Modes: All

1.5.14 TRIGger Subsystem

The TRIGger group of commands is used to control synchronization of data acquisition with external events. TRIGger commands generally affect only Pulse Mode.

TRIGger:CDF:COUNt

Description: Set or return the terminal count (sample population size) in millions of samples for Statistical Mode acquisition. When the terminal count is reached, the CCDF is considered "complete", and the instrument will halt acquisition if INITiate:CONTinuous is set to OFF. If INITiate:CONTinuous is ON, sample acquisition will continue in the manner specified by the TRIGger:CDF:DECImate setting.

Syntax: TRIGger:CDF:COUNt <numeric_value>

Argument: <numeric_value> = 1 to 4000 Valid Modes:

Statistical Mode only.

TRIGger:CDF:DECImate

Description: Set or return the termination action when running continuously in Statistical Mode. This action occurs when the terminal count is reached (as defined by TRIGger:CDF:COUNt) or the terminal running time is reached (as defined by TRIGger:CDF:TIMe).

Syntax: TRIGger:CDF:DECImate <DECIMATE, RESTART, STOP>

Argument: <Boolean> = { DECIMATE, RESTART, STOP }

Valid Modes: Statistical Mode only

TRIGger:CDF:TIMe

Description: Set or return the terminal running time in seconds for Statistical Mode acquisition. When the terminal time is reached, the CCDF is considered "complete", and the instrument will halt acquisition if INITiate:CONTinuous is set to OFF. If INITiate:CONTinuous is ON, sample acquisition will continue in the manner specified by the TRIGger:CDF:DECImate setting.

Syntax: TRIGger:CDF:TIMe <numeric_value>

Argument: <numeric_value> = 1 to 3600 Valid

Modes: Statistical Mode only

TRIGger:DELay

Description: Set or return the trigger delay time in seconds with respect to the trigger for the trigger display location in the LEFT position. Positive values cause the actual trigger to occur after the trigger condition is met. This places the trigger event to the left of the trigger

point on the display, and is useful for viewing events during a pulse, some fixed delay time after the rising edge trigger. Negative trigger delay places the trigger event to the right of the trigger point on the display, and is useful for looking at events before the trigger edge.

Syntax: TRIGger:DELay <numeric_value>

Argument: <numeric value> -378e-3 to 1250e-3, range varies with timebase setting

Valid Modes: Pulse mode only

TRIGger:HOLDoff

Description: Set or return the trigger holdoff time in seconds. Trigger holdoff is used to disable the trigger for a specified amount of time after each trigger event. The holdoff time starts immediately after each valid trigger edge, and will not permit any new triggers until the time has expired. When the holdoff time is up, the trigger re-arms, and the next valid trigger event (edge) will cause a new sweep. This feature is used to help synchronize the power meter with burst waveforms such as a TDMA or GSM frame. The trigger holdoff resolution is 10 nanoseconds, and it should be set to a time that is just slightly shorter than the frame repetition interval.

Syntax: TRIGger:HOLDoff <numeric_value>

Argument: <numeric_value> = 0.0 to 1.0 seconds in 10e-9 increments, 0.0 = no holdoff

Valid Modes: Pulse mode only

TRIGger:LEVel

Description: Set or return the trigger level for synchronizing data acquisition with a pulsed input signal or external trigger pulses. The internal trigger level entered should include any global offset and will also be affected by the frequency cal factor. The available internal trigger level range is sensor dependent. For internal trigger sources, the trigger level is set and returned in dBm. The external trigger is set and returned in volts. Note that there is a small amount of hysteresis built into the trigger system, and the signal should have at least one dB greater swing in each direction past the trigger level setting, and somewhat more at low levels. Note that explicitly setting the trigger level while TRIGger:MODe is set to AUTOPKPK will cancel the AUTOPKPK setting, and force the trigger mode back to AUTO. In AUTOPKPK the Trigger Level menu will display “AUTO LEVEL”.

Syntax: TRIGger:LEVel <numeric_value>

Argument: <numeric_value> = –40.0 to +20 (plus offset, if any) (internal trigger sources)

<numeric_value> = +/-5.0 (external trigger) Valid

Modes: Pulse mode only

TRIGger:MODE

Description: Set or return the trigger mode for synchronizing data acquisition with pulsed signals. NORM mode will cause a sweep to be triggered each time the power level crosses the preset trigger level in the direction specified by TRIGger:SLOPe. If there are no edges that cross this level, no data acquisition will occur. AUTO mode operates in much the same way as NORM mode, but will automatically generate a trace if no trigger edges are detected for a period of time (100 to 500 milliseconds, depending on timebase). This will keep the trace updating even if the pulse edges stop. The AUTOPKPK mode operates the same as AUTO mode, but will adjust the trigger level to halfway between the highest and lowest power or voltage levels detected. This aids in maintaining synchronization with a pulse signal of varying level. Note that a setting of PKTOPK will be overridden and forced back to AUTO if a TRIGger:LEVel is set. The FREERUN mode force traces at a high rate to assist in locating the signal.

Syntax: TRIGger:MODE <character data>

Argument: <character data> = {AUTO, AUTOPKPK, NORMAL, FREERUN}

Valid Modes: Pulse mode only

TRIGger:POSition

Description: Set or return the position of the trigger event on displayed sweep. Assuming zero trigger delay, setting the position to LEFT causes the entire trace to be post-trigger. Setting it to RIGHT causes the entire trace to be pre-trigger. And setting to MIDDLE will display both the pre- and post- trigger portions of the trace. Note that the TRIGger:DELay setting is in addition to this setting, and will cause the trigger position to appear in a different location. Setting the trigger position places the PMX40 in trigger position mode, which overrides the variable vernier settings of the TRIGger:VERNier command.

Syntax: TRIGger:POSition <character data>

Argument: <character data> = {LEFT, MIDDLE, RIGHT}

Valid Modes: Pulse mode only

TRIGger:SLOPe

Description: Set or return the trigger slope or polarity. When set to POS, trigger events will be generated when a signal's rising edge crosses the trigger level threshold. When NEG, trigger events are generated on the falling edge of the pulse.

Syntax: TRIGger:SLOPe <character data>

Argument: <character data> = {NEG, POS}

Valid Modes: Pulse mode only

TRIGger:SOURce

Description: Set or return the trigger source used for synchronizing data acquisition. The CH1 and CH2 (2 channel PMX40) settings use the signal from the associated sensor. EXT setting uses the signal applied to the rear panel TRIG IN connector.

Syntax: TRIGger:SOURce <character data>

Argument: <character data> = {CH1, CH2, CH3, CH4, IND, EXT}

Valid Modes: Pulse mode only

TRIGger:VERNier

Description: Set or return the fine position of the trigger event on displayed sweep. The position is given in divisions relative to the left edge of the screen, so with zero trigger delay, setting the vernier control to 0.0 causes the entire trace to be post-trigger. Setting it to 10.0 causes the entire trace to be pre-trigger. And setting to 5.0 will display both the pre- and post-trigger portions of the trace. Note that the TRIGger:DELay setting is in addition to this setting, and will cause the trigger position to appear in a different location. Setting the trigger vernier places the PMX40 in trigger vernier mode, which overrides the fixed “Left, Middle, Right” settings of the TRIGger:POSition command.

Syntax: TRIGger:VERNier <numeric_value>

Argument: <numeric_value> = –30.0 to 30.0 Valid

Modes: Pulse mode only

END OF PMX40 RF POWER METER Programming Reference