Programmer's Reference Guide for 2250A

Table of Contents

1 Introduction

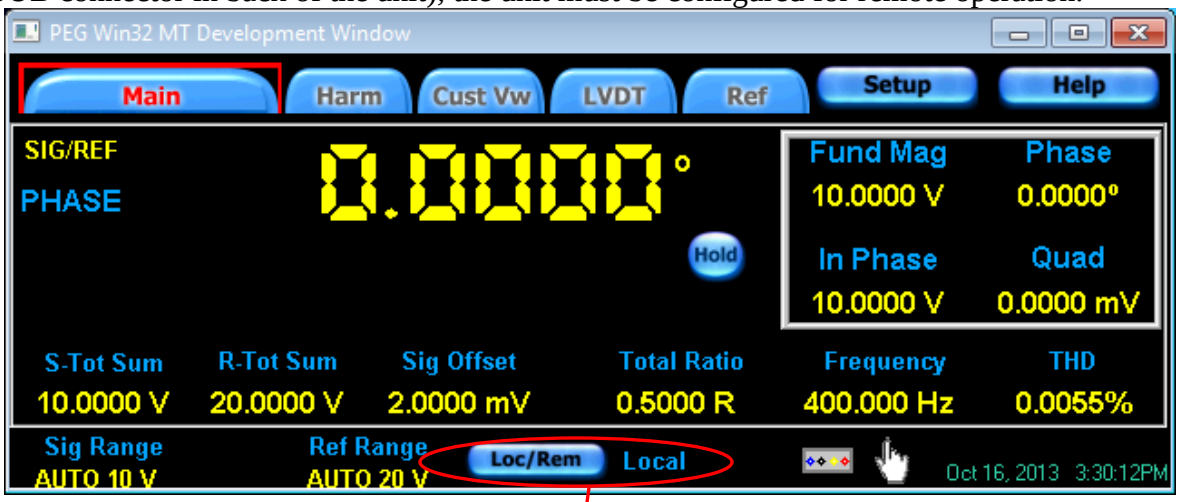
The Model 2250 Phase Angle Voltmeter is the most accurate and versatile AC measurement tool available. This instrument employs a microprocessor-based design, utilizing Fast-Fourier Transform techniques that combine many of the capabilities of today's network/waveform analyzers in a digital analyzing voltmeter configuration. Designed for the Synchro/Resolver and LVDT/RVDT marketplace, this Phase Angle Voltmeter accurately makes measurements of Phase Angle, In-Phase, Quadrature, Fundamental and Total, as well as impedance, power and harmonics. Isolated inputs allow null, ratio and gain measurements of key parameters. A reference phase offset facilitates bridging measurements. A sensitive null meter is included for precise nulling.

Reference Documentation
For additional information about this instrument refer to the *Operation Manual for Model 2250A*. For additional information about the Application Programming Interface (API) provided in the PAV2250ADll refer to the *Function Reference Manual for 2250A*.

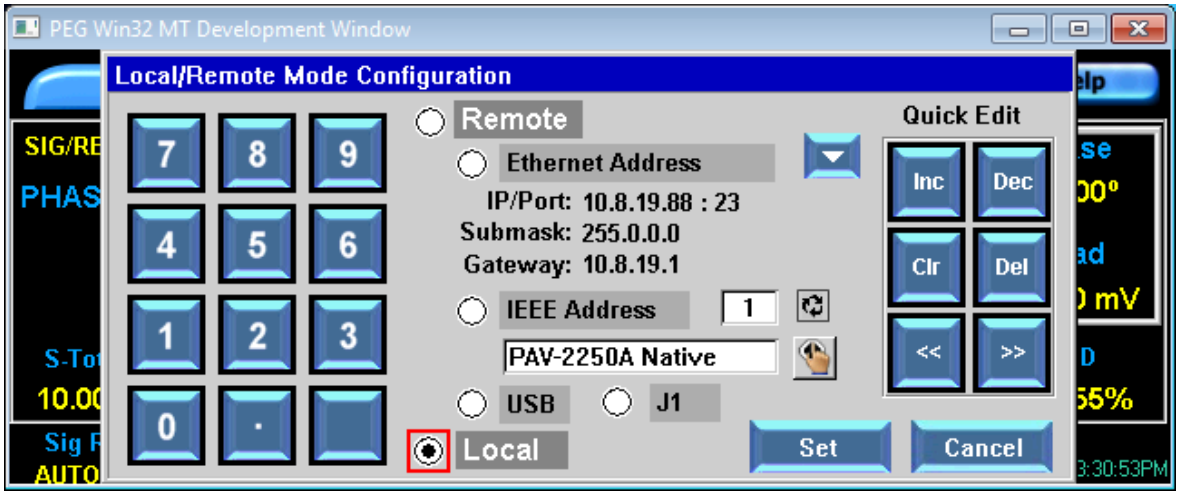
Reference CD
For electronic copies of the 2250A documentation, PAV-2250A Soft Panel application program, and source code for PAV-2250ADll and Soft Panel application refer to the 2250A Product CD.

2 Remote Setup

To enable remote operation capabilities via IEEE-488, USB, Ethernet and J1 connection (50 pin DSUB connector in back of the unit), the unit must be configured for remote operation.



Click on the button labeled "Loc/Rem" to view the Local/Remote Configuration screen:

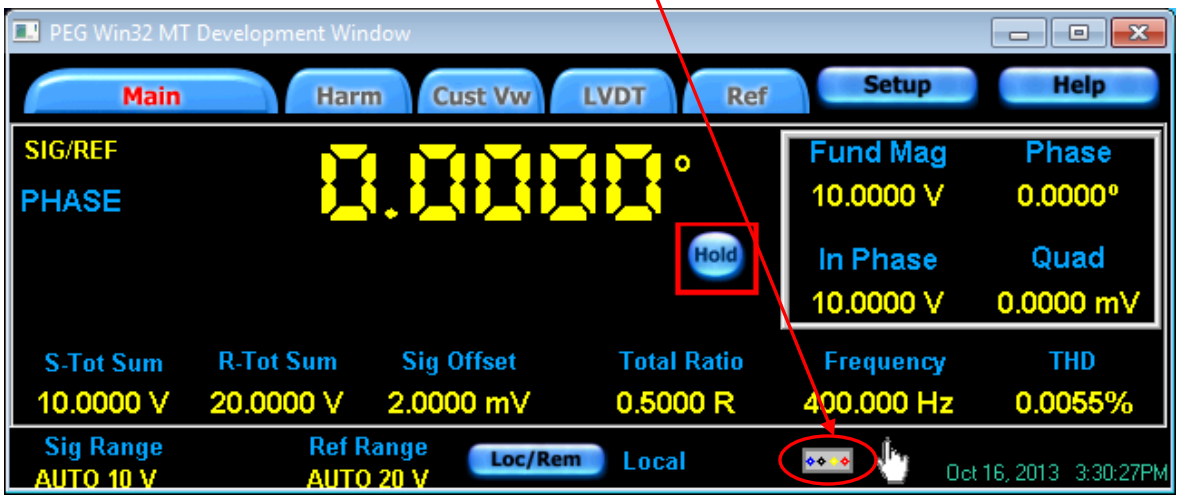


In "Local" mode, the configuration settings can be queried. The unit must be in one of the "Remote" modes (Ethernet, IEEE, USB or J1) before configuration settings can be changed remotely. Note, for remote programming via the IEEE interface, the language type must be selected, refer to section 3 on language support.

2.1 Controlling Channel 1 Signal Input

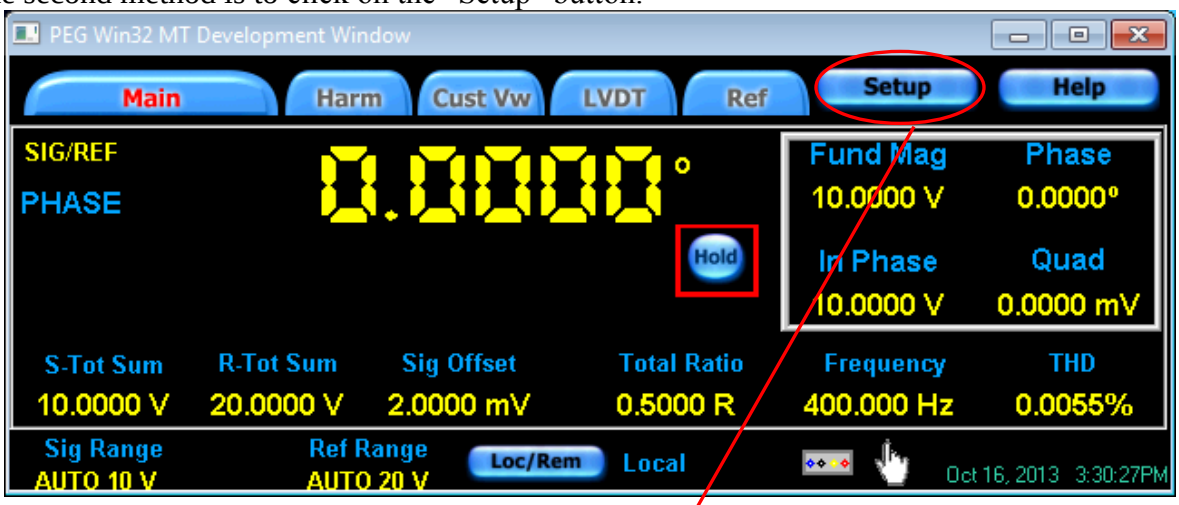
The signal input for channel 1 can be read from the front panel connector or from the J1 connection in the back of the unit. The signal input for channel 2 is read only from the J1 connection.

The configuration for channel 1 is configured two ways:
One method is to click the button shown below:



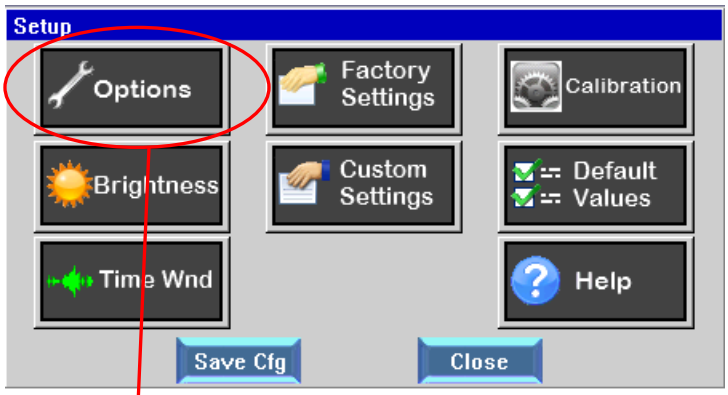
Button configures Chan 1 Input to be read from the Front Connector or the Back (J1) Connector.

The second method is to click on the "Setup" button.

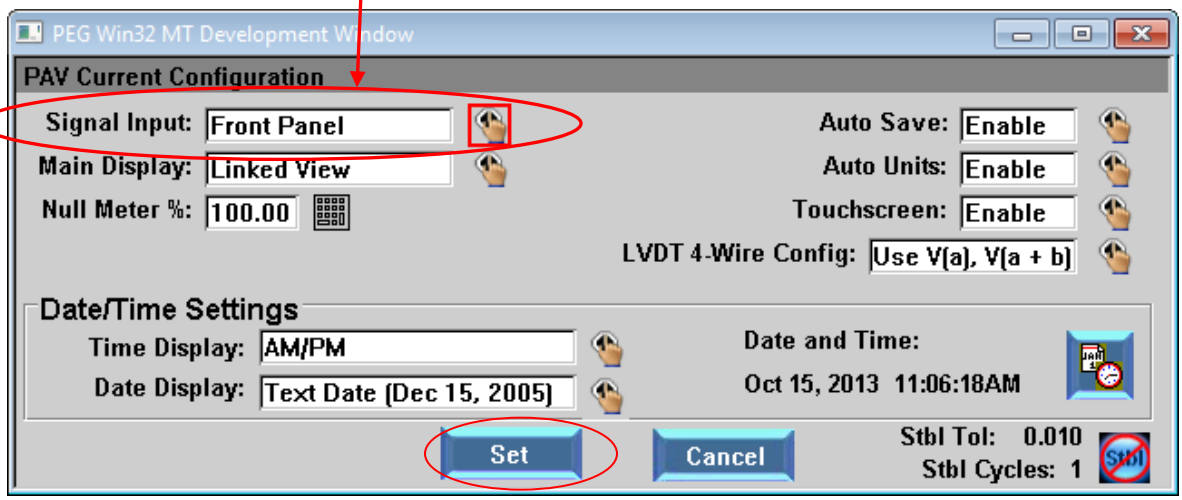


(See next page)

Click the "Options" button.



Select the "Front Panel" or "Back Connector" option for Channel 1 Input. Click on the "Set" button.



3 Language Support

The 2250A Unit is a direct replacement for all 2250's. This unit supports the following languages:

	IEEE-488.1	USB	Ethernet
PAV-2250A Native	**Supported**	**Supported**	**Supported**
PAV-2250 Native (Legacy)	**Supported**	Not available	Not available

The following table lists the applicable IEEE-488 bus commands for the PAV-2250A.

Mnemonic	ASCII	Hex	Function
GTL	SOH	01	Go To Local - This command instructs the PAV to go to local mode. All front panel controls are active.
SDC	EOT	04	Selected Device Clear - When the SDC command is received, and if the PAV is addressed to listen, the PAV will initialize to the conditions listed under DCL.
DCL	DC4	14	Device Clear - When the PAV receives the DCL command it is initialized to the following state: SYNCHRO DATA FREEZE - OFF SRQ MODE - OFF GET MODE – OFF
GET	BS	08	Group Execute Trigger - When the GET command is received, and if the PAV is addressed to listen and has the GET mode switch on, data sent to the PAV will be applied to the instrument.
LLO	DC1	11	Local Lockout - This command disables the front panel REM switch. It gives the controller complete control over whether the PAV is in remote or local operation.
SPE	CAN	18	Serial Poll Enable - After receipt of this command the PAV, when addressed to talk, will transmit the Status Byte.
SPD	EM	19	Serial Poll Disable - This command cancels the SPE command and allows the PAV, when it is addressed to talk, to send data.
UNL	/	3F	Unlisten - Unaddresses the PAV listen address.
UNT	-	5F	Untalk - Unaddresses the PAV talk address.

The following table lists the interface function capability codes for the PAV-2250A.

Code	Function
AH1	Acceptor handshake - complete capability
SH1	Source handshake - complete capability
T6	Talk capability - all except TON
TEO	Extended Talk capability – none
L4	Listen capability - all except LON
LEO	Extended Listen capability – none

SR1	Service request - complete capability
RL1	Remote/Local - complete capability
PPO	Parallel Poll - no capability
DC1	Device Clear - complete capability
DT1	Device Trigger - complete capability

3.1 Compatibility to 2250 APIs

The 2250A will provide language compatibility to the following 2250 systems:

- PAV-2250 Native

3.2 Language Independent Commands

Note the following commands are case-sensitive.

Function	Syntax (commands must be sent with upper-case)	Comments
API COMMANDS		
Identification	*IDN?<cr><lf>	Queries the device for the ID.
Error Reporting	*ERR?<cr><lf>	Queries for any error messages on the error message queue. "No error" is returned when there are no errors on the queue.
Reset	*RST?<cr><lf>	Clears the error message queue and resets the device with default factory settings." Reset Complete" is returned when device is reset to the default settings. "Reset Not Performed" is returned if the device's remote configuration does not match communication connection mode.

Language	PAVCMD<b>LANG?<cr><lf>	Queries the IEEE Language setting. Query returns: 0 for PAV-2250 Native and 1 for PAV-2250 Legacy
	PAVCMD<b>LANG<b>TEXT?<cr><lf>	Queries the IEEE Language setting. Query returns: 'PAV-2250A Native', or 'PAV-2250 Legacy'
	PAVCMD<b>LANG<b>< 2250ANATIVE><cr><lf>	Sets the IEEE Language setting.
Stable	PAVCMD<b>STBL?<cr><lf>	Queries whether the PAV is considered to be STABLE. 0 indicates NOT STABLE; 1 indicates STABLE

3.3 PAV-2250A Native

The PAV-2250A Native language is supported via the IEEE-488.1, USB and Ethernet interfaces. The language provides remote programming access to the features available on the 2250A unit.

COMMAND FUNCTIONS		
Function	**Syntax**	**Comments**
Stable	PAVCMD<b>STBL?<cr><lf>	Queries whether the PAV is considered to be STABLE. 0 indicates NOT STABLE; 1 indicates STABLE
Data Retrieval	PAVCMD<b>DATA?<cr><lf>	Returns comma delimited string containing: - Total Ratio - Total Ref Volt - Total Sig Volt - THD - Frequency - Sample Rate Index - Ref Range - Sig Range

	PAVCMD<b>DATAR?<cr><lf>	Floating point values (4 bytes) each: * Total Ratio * Total Ref Volt * Total Sig Volt * THD * Frequency Unsigned short values (2 bytes) each: * Sample Rate Idx * Ref Range * Sig Range Harmonic data will be returned (4 bytes): (Ethernet - all 16 harmonics) (USB - only fundamental) * Phase * Amplitude * In Phase * Quadrature * AmplitudeRatio * In PhaseRatio * QuadratureRatio

HARMONIC FUNCTIONS		
Function	**Syntax**	**Comments**
Harmonic Full Data Retrieval	PAVHARM<harm>?<cr><lf>	Queries the given Harmonic (0 – 15) and returns a comma separated string containing: - Phase - Magnitude - In Phase - Quad
Phase Data Retrieval	PAVHARM<harm><b>PHASE?<cr><lf>	Returns the Phase component value for the given Harmonic (0 – 15).
Magnitude Data Retrieval	PAVHARM<harm><b>MAG?<cr><lf>	Returns the Magnitude component value for the given Harmonic (0 – 15).
InPhase Data Retrieval	PAVHARM<harm><b>IN_PHASE?<cr><lf>	Returns the InPhase component value for the given Harmonic (0 – 15).
Quad Data Retrieval	PAVHARM<harm><b>QUAD?<cr><lf>	Returns the Quad component value for the given Harmonic (0 – 15).
Harmonic Full Ratio Data Retrieval	PAVHARM<harm><b>RATIO?<cr><lf>	Queries the given Harmonic (0 – 15) and returns a comma separated string containing: - Phase - Magnitude Ratio - In Phase Ratio - Quad Ratio
Phase Ratio Data Retrieval	PAVHARM<harm><b>RATIO<b>PHASE?<cr><lf>	Returns the Phase component value for the given Harmonic (0 – 15).
Magnitude Ratio Data	PAVHARM<harm><b>RATIO<b>MAG?<cr><lf>	Returns the Magnitude Ratio

Retrieval		component value for the given Harmonic (0 – 15).
InPhase Ratio Data Retrieval	PAVHARM<harm><b>RATIO<b>IN_PHASE?<cr><lf>	Returns the InPhase Ratio component value for the given Harmonic (0 – 15).
Quad Ratio Data Retrieval	PAVHARM<harm><b>RATIO<b>QUAD?<cr><lf>	Returns the Quad Ratio component value for the given Harmonic (0 – 15).
Harmonic Full RAW (HEX) Data Retrieval	PAVHARM<harm_group><b>RAW?<cr><lf>	Returns the RAW (Hex) values for each of the Harmonic components of the specified harmonic group. Groups are 1 – 8 and each comprise of 2 Harmonics. Ex. The 1st harmonic group (1) returns data for the 1st and 2nd Harmonic. Group 2 returns data for the 3rd and 4th harmonic and so on.
Harmonic Ratio State	PAVHARM<b>RATIO_STATE?<cr><lf>	Queries the device for the current ratio state. When enabled, Ratio state displays the ratio values for Magnitude, In Phase and Quadrature. Return Values: 0 – Not in Ratio Mode 1 – In Ratio Mode
	PAVHARM<b>RATIO_STATE<b>SET<b><int_val><cr><lf>	Sets the Ratio State based on the value of <int_val>. 0 – Not in Ratio 1 – In Ratio
Harmonic Navigation	PAVHARM<b>PREV_GROUP<cr><lf>	Forces Harmonics View to display the Previous Harmonic group page. Each

		Group displays 5 Harmonics.
	PAVHARM<b>NEXT_GROUP<cr><lf>	Forces Harmonics View to display the Next Harmonic group page. Each Group displays 5 Harmonics.
	PAVHARM<b>VIEW_HARM<b><harm><cr><lf>	Forces Harmonics View to display the desired Harmonic specified by the value <harm>. Possible Values: 0- 15

REFERENCE RANGE FUNCTIONS		
Function	**Syntax**	**Comments**
Reference Auto Range	PAVCMD<b>REF_RANGE<b>AUTO<cr><lf>	Forces the reference to be in "Auto" range mode. (Instrument decides appropriate range)
	PAVCMD<b>REF_RANGE<b>AUTO? <cr><lf>	Queries for whether the reference is currently in "Auto" range mode.
Reference Range	PAVCMD<b>REF_RANGE<b>SET<b><rng_idx> <cr><lf>	Forces the reference to be in the specified Range. Range Index is passed in (0 – 12)
	PAVCMD<b>REF_RANGE? <cr><lf>	Queries for current Range setting information: Return values will have the word "AUTO" precede the actual range value when the PAV is in "AUTO" range mode and will have the word "OVR" precede the actual range when the actual range is over the configured range. Example: Auto Range with an actual range of 2 Volts will return: "AUTO 2.000". If the range is not "AUTO" and an Over range was not detected, the configured range will be returned such as: "2.000" for the 2 volt range.

	PAVCMD<b>REF_RANGE<b>INDEX_SETTINGS?<cr><lf>	Queries for current range setting information: Return values indicate whether or not the reference is in "Auto" range mode, whether or not there is a range mismatch (i.e. the actual range is different than the desired configured range), and the actual range index.
	PAVCMD<b>REF_RANGE<b>CONFIG_INDEX?<cr><lf>	Queries for the reference range index the PAV was configured with. (return values: 0 – 12 or 65535 for "AUTO")
	PAVCMD<b>REF_RANGE<b>ACTUAL_INDEX?<cr><lf>	Queries for the reference range index the PAV is actually operating in. (return values: 0 – 12)
Reference Range – Set to specific range	PAVCMD<b>REF_RANGE<b>50MV<cr><lf>	Sets reference range to 50MV.
	PAVCMD<b>REF_RANGE<b>100MV<cr><lf>	Sets reference range to 100MV.
	PAVCMD<b>REF_RANGE<b>200MV<cr><lf>	Sets reference range to 200MV.
	PAVCMD<b>REF_RANGE<b>500MV<cr><lf>	Sets reference range to 500MV.
	PAVCMD<b>REF_RANGE<b>1V<cr><lf>	Sets reference range to 1V.
	PAVCMD<b>REF_RANGE<b>2V<cr><lf>	Sets reference range to 2V.
	PAVCMD<b>REF_RANGE<b>5V<cr><lf>	Sets reference range to 5V.
	PAVCMD<b>REF_RANGE<b>10V<cr><lf>	Sets reference range to 10V.
	PAVCMD<b>REF_RANGE<b>20V<cr><lf>	Sets reference range to 20V.
	PAVCMD<b>REF_RANGE<b>50V<cr><lf>	Sets reference range to 50V.

	PAVCMD<b>REF_RANGE<b>100V<cr><lf>	Sets reference range to 100V.
	PAVCMD<b>REF_RANGE<b>200V<cr><lf>	Sets reference range to 200V.
	PAVCMD<b>REF_RANGE<b>500V<cr><lf>	Sets reference range to 500V.

SIGNAL RANGE FUNCTIONS		
Function	**Syntax**	**Comments**
Signal Auto Range	PAVCMD<b>SIG_RANGE<b>AUTO<cr><lf>	Forces the signal to be in "Auto" range mode. (Instrument decides appropriate range)
	PAVCMD<b>SIG_RANGE<b>AUTO? <cr><lf>	Queries for whether the signal is currently in "Auto" range mode.
Signal Range	PAVCMD<b>SIG_RANGE<b>SET<b><rng_idx> <cr><lf>	Forces the signal to be in the specified range. Range Index is passed in (0 – 12)
	PAVCMD<b>SIG_RANGE? <cr><lf>	Queries for current range setting information: Return values will have the word "AUTO" precede the actual range value when the PAV is in "AUTO" range mode and will have the word "OVR" precede the actual range when the actual range is over the configured range. Example: Auto Range with an actual range of 2 Volts will return: "AUTO 2.000". If the range is not "AUTO" and an Over range was not detected, the configured range will be returned such as: "2.000" for the 2 volt range.

	PAVCMD<b>SIG_RANGE<b>INDEX_SETTINGS?<cr><lf>	Queries for current range setting information: Return values indicate whether or not the reference is in "Auto" range mode, whether or not there is a range mismatch (i.e. the actual range is different than the desired configured range), and the actual range index.
	PAVCMD<b>SIG_RANGE<b>CONFIG_INDEX?<cr><lf>	Queries for the signal range index the PAV was configured with. (return values: 0 – 12 or 65535 for "AUTO")
Signal Range – Set to specific range	PAVCMD<b>SIG_RANGE<b>ACTUAL_INDEX?<cr><lf>	Queries for the signal range index the PAV is actually operating in. (return values: 0 – 12)
	PAVCMD<b>SIG_RANGE<b>50MV<cr><lf>	Sets signal range to 50MV.
	PAVCMD<b>SIG_RANGE<b>100MV<cr><lf>	Sets signal range to 100MV.
	PAVCMD<b>SIG_RANGE<b>200MV<cr><lf>	Sets signal range to 200MV.
	PAVCMD<b>SIG_RANGE<b>500MV<cr><lf>	Sets signal range to 500MV.
	PAVCMD<b>SIG_RANGE<b>1V<cr><lf>	Sets signal range to 1V.
	PAVCMD<b>SIG_RANGE<b>2V<cr><lf>	Sets signal range to 2V.
	PAVCMD<b>SIG_RANGE<b>5V<cr><lf>	Sets signal range to 5V.
	PAVCMD<b>SIG_RANGE<b>10V<cr><lf>	Sets signal range to 10V.
	PAVCMD<b>SIG_RANGE<b>20V<cr><lf>	Sets signal range to 20V.
	PAVCMD<b>SIG_RANGE<b>50V<cr><lf>	Sets signal range to 50V.

	PAVCMD<b>SIG_RANGE<b>100V<cr><lf>	Sets signal range to 100V.
	PAVCMD<b>SIG_RANGE<b>200V<cr><lf>	Sets signal range to 200V.
	PAVCMD<b>SIG_RANGE<b>500V<cr><lf>	Sets signal range to 500V.

CONFIGURATION FUNCTIONS		
Function	**Syntax**	**Comments**
IEEE Language	PAVCMD<b>LANG?cr><lf>	Queries the device for the IEEE language protocol it is currently configured with. (0 indicates NATIVE, 1 indicates LEGACY)
	PAVCMD<b>LANG<b>TEXT?<cr><lf>	Queries the device for the IEEE language protocol it is currently configured with. Actual language names are returned. (PAV-2250A Native and PAV-2250 Legacy)
	PAVCMD<b>LANG<b>SET<b><lang_idx><cr><lf>	Sets the IEEE language to the desired language (based on language index)
	PAVCMD<b>LANG<b>2250ANATIVE<cr><lf>	Sets the IEEE language to 2250ANATIVE.
	PAVCMD<b>LANG<b>2250LEGACY<cr><lf>	Sets the IEEE language to 2250LEGACY.
Communications	PAVCMD<b>COMM?<cr><lf>	Queries the device for the currently configured communication mode.
	PAVCMD<b>COMM<b>LOCAL<cr><lf>	Sets the communication mode to Local.
	PAVCMD<b>COMM<b>USB<cr><lf>	Sets the communication mode to USB.
	PAVCMD<b>COMM<b>ETHERNET<cr><lf>	Sets the communication mode to Remote Ethernet.

	PAVCMD<b>COMM<b>IEEE<cr><lf>	Sets the communication mode to IEEE.
	PAVCMD<b>COMM<b>J1<cr><lf>	Sets the communication mode to J1.

CALIBRATION FUNCTIONS		
Function	**Syntax**	**Comments**
Calibration	PAVCMD<b>CALIBRATE<cr><lf>	Forces PAV to perform calibration.
	PAVCMD<b>CALIBRATE?<cr><lf>	Queries the device for its current calibration state. Return Values: "CALIBRATING" – if PAV is currently calibrating. "READY" – if PAV is ready.

TIME WINDOW FUNCTIONS		
Function	**Syntax**	**Comments**
Time Window	PAVCMD<b>TIME_WND<b>AUTO<b><value><cr><lf>	Forces PAV Time Window to be in "Auto" mode.
	PAVCMD<b>TIME_WND<b>AUTO?<cr><lf>	Queries the device to determine if the Time Window is set to "Auto" or not. 1 is returned if "Auto" else 0.
	PAVCMD<b>TIME_WND<b>OVERRIDE<b>SET<b><float_val><cr><lf>	Forces PAV Time Window to a specific value denoted by <float_val>
	PAVCMD<b>TIME_WND<b>OVERRIDE?<cr><lf>	Queries the device for the Time Window override floating point interval value.

	PAVCMD<b>TIME_WND<b>ACTUAL?<cr><lf>	Queries the device for the Time Window actual floating point interval value.

PAV SETUP OPTIONS FUNCTIONS		
Function	**Syntax**	**Comments**
Signal Input Option	PAVOPT<b>SIGNAL_INPUT?<cr><lf>	Queries the device for the current Signal Input Value. Return Values: 0 (Front Panel) or 1 (Back Panel).
	PAVOPT<b>SIGNAL_INPUT<b>TEXT?<cr><lf>	Queries the device for the current Signal Input Value. Return Values: "Front Panel" or "Back Panel".
	PAVOPT<b>SIGNAL_INPUT<b>SET<b><input_idx><cr><lf>	Forces the current Signal Input to either the Front Panel (0) or Back Panel (1). <input_idx> should be 0 for Front Panel or 1 for Back Panel.
	PAVOPT<b>SIGNAL_INPUT<b>FRONT <cr><lf>	Forces the current Signal Input to be "Front Panel".
	PAVOPT<b>SIGNAL_INPUT<b>BACK<cr><lf>	Forces the current Signal Input to be "Back Panel"
Main Display Option	PAVOPT<b>MAIN_DISPLAY?<cr><lf>	Queries the device for the current Main Display Index. Return Values: 0 – "Independent View" 1 – "Linked View"
	PAVOPT<b>MAIN_DISPLAY<b>TEXT?<cr><lf>	Queries the device for the current Main Display option text. Return Values: "Independent View" "Linked View"

	PAVOPT<b>MAIN_DISPLAY<b>SET<b><option_idx><cr><lf>	Forces the Main Display Option based on the value of <option_idx>. 0 – "Independent" 1 – "Linked View"
	PAVOPT<b>MAIN_DISPLAY<b>INDEPENDENT<cr><lf>	Forces the Main Display Option to "Independent"
	PAVOPT<b>MAIN_DISPLAY<b>LINKED<cr><lf>	Forces the Main Display Option to "LINKED"
Time Display Option	PAVOPT<b>TIME_DISPLAY?<cr><lf>	Queries the device for the current Time Display Option Index. Return Values: 0 – "AM/PM" 1 – "Military"
	PAVOPT<b>TIME_DISPLAY<b>TEXT?<cr><lf>	Queries the device for the current Time Display Option Text. Return Values: "AM/PM" "Military"
	PAVOPT<b>TIME_DISPLAY<b>SET<b><option_idx><cr><lf>	Forces the Time Display Option Index based on the value of <option_idx>. 0 – "AM/PM" 1 – "Military"
	PAVOPT<b>TIME_DISPLAY<b>AMPM<cr><lf>	Forces the Time Display Option to AM/PM.
	PAVOPT<b>TIME_DISPLAY<b>MILITARY<cr><lf>	Forces the Time Display Option to Military.
Date Display Option	PAVOPT<b>DATE_DISPLAY?<cr><lf>	Queries the device for the current Date Display Option Index. Return Values: 0 – "Text Date" 1 – "Numeric Date"

	PAVOPT<b>DATE_DISPLAY<b>TEXT?<cr><lf>	Queries the device for the current Date Display Option Text. Return Values: "Text Date" "Numeric Date"
	PAVOPT<b>DATE_DISPLAY<b>SET<b><option_idx><cr><lf>	Forces the Date Display Option Index based on the value of <option_idx>. 0 – "Text Date" 1 – "Numeric Date"
	PAVOPT<b>DATE_DISPLAY<b>TEXT<cr><lf>	Forces the Date Display Option to Text Date.
	PAVOPT<b>DATE_DISPLAY<b>NUMERIC<cr><lf>	Forces the Time Display Option to Numeric Date.
Auto Save Option	PAVOPT<b>AUTO_SAVE?<cr><lf>	Queries the device for the current Auto Save Option Index. Return Values: 0 – "Disabled" 1 – "Enabled"
	PAVOPT<b>AUTO_SAVE<b>TEXT?<cr><lf>	Queries the device for the current Auto Save Option Text. Return Values: "Disabled" "Enabled"
	PAVOPT<b>AUTO_SAVE<b>SET<b><option_idx><cr><lf>	Forces the Auto Save Option Index based on the value of <option_idx>. 0 – "Disabled" 1 – "Enabled"
	PAVOPT<b>AUTO_SAVE<b>ENABLE<cr><lf>	Forces the Auto Save Option to Enabled.
	PAVOPT<b>AUTO_SAVE<b>DISABLE<cr><lf>	Forces the Auto Save Option to Disabled.

Auto Units Option	PAVOPT<b>AUTO_UNITS?<cr><lf>	Queries the device for the current Auto Units Option Index. Return Values: 0 – "Disabled" 1 – "Enabled"
	PAVOPT<b>AUTO_UNITS<b>TEXT?<cr><lf>	Queries the device for the current Auto Units Option Text. Return Values: "Disabled" "Enabled"
	PAVOPT<b>AUTO_UNITS<b>SET<b><option_idx><cr><lf>	Forces the Auto Units Option Index based on the value of <option_idx>. 0 – "Disabled" 1 – "Enabled"
	PAVOPT<b>AUTO_UNITS<b>ENABLE<cr><lf>	Forces the Auto Units Option to Enabled.
	PAVOPT<b>AUTO_UNITS<b>DISABLE<cr><lf>	Forces the Auto Units Option to Disabled.
Touchscreen Option	PAVOPT<b>TOUCHSCREEN?<cr><lf>	Queries the device for the current Touchscreen Option Index. Return Values: 0 – "Disabled" 1 – "Enabled"
	PAVOPT<b>TOUCHSCREEN<b>TEXT?<cr><lf>	Queries the device for the current Touchscreen Option Text. Return Values: "Disabled" "Enabled"
	PAVOPT<b>TOUCHSCREEN<b>SET<b><option_idx><cr><lf>	Forces the Touchscreen Option Index based on the value of <option_idx>. 0 – "Disabled" 1 – "Enabled"
	PAVOPT<b>TOUCHSCREEN<b>ENABLE<cr><lf>	Forces the Touchscreen Option to Enabled.

	PAVOPT<b>TOUCSCREEN<b>DISABLE<cr><lf>	Forces the Touchscreen Option to Disabled.
LVDT 4 Wire Config	SEE "LVDT FUNCTIONS" TABLE for commands to change the LVDT 4-Wire Algorithm.	
Null Meter	PAVOPT<b>RANGE_PERCENT<b>SET<b><float_val><cr><lf>	Forces the Range Percent to the value of <float_val>. This value controls the end-points of the Null Meter.
	PAVOPT<b>RANGE_PERCENT?<cr><lf>	Queries the device for the current Null Meter Range Percent value. A Float value will be returned.

PAV CONFIGURATION FUNCTIONS		
Function	**Syntax**	**Comments**
Tab View	PAVCMD<b>TAB_VIEW?<cr><lf>	Queries the device for the current Tab View Index. Return Values: 0 – "Main Tab" 1 – "Reference Tab" 2 – "Harmonics Tab" 3 – "Quad Tab"
	PAVCMD<b>TAB_VIEW<b>TEXT?<cr><lf>	Queries the device for the current Tab View Text. Return Values: "Main" "Reference" "Harmonics" "QuadView"

	PAVCMD<b>TAB_VIEW<b>SET<b><option_idx><cr><lf>	Forces the Tab View Index based on the value of <option_idx>. 0 – "Main Tab" 1 – "Reference Tab" 2 – "Harmonics Tab" 3 – "Quad Tab"
	PAVCMD<b>TAB_VIEW<b>MAIN<cr><lf>	Forces the Tab View to Main Tab.
	PAVCMD<b>TAB_VIEW<b>REFERENCE<cr><lf>	(RESERVED FOR FUTURE CAPABILITY)
	PAVCMD<b>TAB_VIEW<b>HARMONICS<cr><lf>	Forces the Tab View to Harmonics Tab.
	PAVCMD<b>TAB_VIEW<b>QUAD<cr><lf>	Forces the Tab View to Quad Tab.
Read Mode	PAVCMD<b>READ_MODE?<cr><lf>	Queries the device for the current Read Mode Index. Return Values: 0 – "SIG/REF" 1 – "REF" 2 – "SIG" 3 – "REF/SIG"
	PAVCMD<b>READ_MODE<b>TEXT?<cr><lf>	Queries the device for the current Read Mode Text. Return Values: "SIG/REF" "REF" "SIG" "REF/SIG"
	PAVCMD<b>READ_MODE<b>SET<b><option_idx><cr><lf>	Forces the Read Mode Index based on the value of <option_idx>. 0 – "SIG/REF" 1 – "REF" 2 – "SIG" 3 – "REF_SIG"
	PAVCMD<b>READ_MODE<b>SIG_REF<cr><lf>	Forces the Read Mode to SIG/REF.
	PAVCMD<b>READ_MODE<b>REF_REF<cr><lf>	Forces the Read Mode to REF.

	PAVCMD<b>READ_MODE<b>SIG_SIG<cr><lf>	Forces the Read Mode to SIG.
	PAVCMD<b>READ_MODE<b>REF_SIG<cr><lf>	Forces the Read Mode to REF/SIG.
Main View	PAVCMD<b>MAIN_VIEW?<cr><lf>	Queries the device for the current Main View Index. Return Values: 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>MAIN_VIEW<b>TEXT?<cr><lf>	Queries the device for the current Main View Text. Return Values: "Fund Mag" "In Phase" "Quad" "Phase" "THD" "Sig Volt" "Ref Volt"
	PAVCMD<b>MAIN_VIEW<b>SET<b><option_idx><cr><lf>	Forces the Main View Index based on the value of <option_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>MAIN_VIEW<b>FUND_MAG<cr><lf>	Forces the Main View to Fund Mag.
	PAVCMD<b>MAIN_VIEW<b>IN_PHASE<cr><lf>	Forces the Main View to In Phase.
	PAVCMD<b>MAIN_VIEW<b>QUAD<cr><lf>	Forces the Main View to Quad.
	PAVCMD<b>MAIN_VIEW<b>PHASE<cr><lf>	Forces the Main View to Phase.

	PAVCMD<b>MAIN_VIEW<b>THD<cr><lf>	Forces the Main View to THD.
	PAVCMD<b>MAIN_VIEW<b>SIG_VOLT<cr><lf>	Forces the Main View to SIG_VOLT.
	PAVCMD<b>MAIN_VIEW<b>REF_VOLT<cr><lf>	Forces the Main View to REF_VOLT.
Hold Data State	PAVCMD<b>HOLD_DATA?<cr><lf>	Queries the device for the current Hold Data State. Return Values: 0 – Normal Operation 1 – Data Hold
	PAVCMD<b>HOLD_DATA<b>SET<b><int_val><cr><lf>	Forces the Data Hold State based on the value of <int_val>. 0 – Normal Operation (not holding data) 1 – Data Hold
Screen Brightness	PAVCMD<b>SCREEN_BRIGHTNESS?<cr><lf>	Queries the device for the current Screen Brightness. Return Values: 15 - 100 (Dim -> Bright)
	PAVCMD<b>SCREEN_BRIGHTNESS<b>SET<int_val><cr><lf>	Forces the Screen Brightness based on the value of <int_val>. 15 - 100 15 – lowest intensity 100 – highest intensity
Custom View 1	PAVCMD<b>CUST_VIEW1?<cr><lf>	Queries the device for the current CustView1 View Index. Return Values: 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"

	PAVCMD<b>CUST_VIEW1<b>TEXT?<cr><lf>	Queries the device for the current CustView1 View Text. Return Values: "Fund Mag" "In Phase" "Quad" "Phase" "THD" "Sig Volt" "Ref Volt"
	PAVCMD<b>CUST_VIEW1<b>SET<b><int_val><cr><lf>	Forces the CustView1 View Index based on the value of <option_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>CUST_VIEW1<b>FUND_MAG<cr><lf>	Forces the CustView1 to Fund Mag.
	PAVCMD<b>CUST_VIEW1<b>IN_PHASE<cr><lf>	Forces the CustView1 to In Phase.
	PAVCMD<b>CUST_VIEW1<b>QUAD<cr><lf>	Forces the CustView1 to Quad.
	PAVCMD<b>CUST_VIEW1<b>PHASE<cr><lf>	Forces the CustView1 to Phase.
	PAVCMD<b>CUST_VIEW1<b>THD<cr><lf>	Forces the CustView1 to THD.
	PAVCMD<b>CUST_VIEW1<b>SIG_VOLT<cr><lf>	Forces the CustView1 to SIG_VOLT.
	PAVCMD<b>CUST_VIEW1<b>REF_VOLT<cr><lf>	Forces the CustView1 to REF_VOLT.

Custom View 2	PAVCMD<b>CUST_VIEW2?<cr><lf>	Queries the device for the current CustView2 View Index. Return Values: 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>CUST_VIEW2<b>TEXT?<cr><lf>	Queries the device for the current CustView2 View Text. Return Values: "Fund Mag" "In Phase" "Quad" "Phase" "THD" "Sig Volt" "Ref Volt"
	PAVCMD<b>CUST_VIEW2<b>SET<b><int_val><cr><lf>	Forces the CustView2 View Index based on the value of <option_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>CUST_VIEW2<b>FUND_MAG<cr><lf>	Forces the CustView2 to Fund Mag.
	PAVCMD<b>CUST_VIEW2<b>IN_PHASE<cr><lf>	Forces the CustView2 to In Phase.
	PAVCMD<b>CUST_VIEW2<b>QUAD<cr><lf>	Forces the CustView2 to Quad.
	PAVCMD<b>CUST_VIEW2<b>PHASE<cr><lf>	Forces the CustView2 to Phase.
	PAVCMD<b>CUST_VIEW2<b>THD<cr><lf>	Forces the CustView2 to THD.

	PAVCMD<b>CUST_VIEW2<b>SIG_VOLT<cr><lf>	Forces the CustView2 to SIG_VOLT.
	PAVCMD<b>CUST_VIEW2<b>REF_VOLT<cr><lf>	Forces the CustView2 to REF_VOLT.
Custom View 3	PAVCMD<b>CUST_VIEW3?<cr><lf>	Queries the device for the current CustView3 View Index. Return Values: 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>CUST_VIEW3<b>TEXT?<cr><lf>	Queries the device for the current CustView3 View Text. Return Values: "Fund Mag" "In Phase" "Quad" "Phase" "THD" "Sig Volt" "Ref Volt"
	PAVCMD<b>CUST_VIEW3<b>SET<b><int_val><cr><lf>	Forces the CustView3 View Index based on the value of <option_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>CUST_VIEW3<b>FUND_MAG<cr><lf>	Forces the CustView3 to Fund Mag.
	PAVCMD<b>CUST_VIEW3<b>IN_PHASE<cr><lf>	Forces the CustView3 to In Phase.

	PAVCMD<b>CUST_VIEW3<b>QUAD<cr><lf>	Forces the CustView3 to Quad.
	PAVCMD<b>CUST_VIEW3<b>PHASE<cr><lf>	Forces the CustView3 to Phase.
	PAVCMD<b>CUST_VIEW3<b>THD<cr><lf>	Forces the CustView3 to THD.
	PAVCMD<b>CUST_VIEW3<b>SIG_VOLT<cr><lf>	Forces the CustView3 to SIG_VOLT.
	PAVCMD<b>CUST_VIEW3<b>REF_VOLT<cr><lf>	Forces the CustView3 to REF_VOLT.
Custom View 4	PAVCMD<b>CUST_VIEW4?<cr><lf>	Queries the device for the current CustView4 View Index. Return Values: 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"
	PAVCMD<b>CUST_VIEW4<b>TEXT?<cr><lf>	Queries the device for the current CustView4 View Text. Return Values: "Fund Mag" "In Phase" "Quad" "Phase" "THD" "Sig Volt" "Ref Volt"
	PAVCMD<b>CUST_VIEW4<b>SET<b><int_val><cr><lf>	Forces the CustView4 View Index based on the value of <option_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt"

	PAVCMD<b>CUST_VIEW4<b>FUND_MAG<cr><lf>	Forces the CustView4 to Fund Mag.
	PAVCMD<b>CUST_VIEW4<b>IN_PHASE<cr><lf>	Forces the CustView4 to In Phase.
	PAVCMD<b>CUST_VIEW4<b>QUAD<cr><lf>	Forces the CustView4 to Quad.
	PAVCMD<b>CUST_VIEW4<b>PHASE<cr><lf>	Forces the CustView4 to Phase.
	PAVCMD<b>CUST_VIEW4<b>THD<cr><lf>	Forces the CustView4 to THD.
	PAVCMD<b>CUST_VIEW4<b>SIG_VOLT<cr><lf>	Forces the CustView4 to SIG_VOLT.
	PAVCMD<b>CUST_VIEW4<b>REF_VOLT<cr><lf>	Forces the CustView4 to REF_VOLT.

VIEW CONFIGURATION FUNCTIONS		
Function	**Syntax**	**Comments**

View Configurations	PAVCMD<b>VIEW_INDEX_CONFIG<b><int_val>?<cr><lf>	Queries the device for the specified view's configuration information. View Index Values: 0 – "FundMag" 1 – "InPhase" 2 – "Quad" 3 – "Phase" 4 – "THD" 5 – "SigVolt" 6 – "RefVolt" 7 – "SigOffset" 8 – "TotalRatio" 9 – "Freq" 10 – "Main" Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>FUND_MAG?<cr><lf>	Queries the device for the FundMag view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale

	PAVCMD<b>VIEW_INDEX_CONFIG<b>IN_PHASE?<cr><lf>	Queries the device for the InPhase view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>QUAD?<cr><lf>	Queries the device for the Quad view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>PHASE?<cr><lf>	Queries the device for the Phase view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale

	PAVCMD<b>VIEW_INDEX_CONFIG<b>THD?<cr><lf>	Queries the device for the THD view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>SIG_VOLT?<cr><lf>	Queries the device for the SigVolt view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>REF_VOLT?<cr><lf>	Queries the device for the RefVolt view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale

	PAVCMD<b>VIEW_INDEX_CONFIG<b>SIG_OFFSET?<cr><lf>	Queries the device for the SigOffset view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>TOTAL_RATIO?<cr><lf>	Queries the device for the TotalRatio view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset Scale
	PAVCMD<b>VIEW_INDEX_CONFIG<b>FREQUENCY?<cr><lf>	Queries the device for the Frequency view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale

	PAVCMD<b>VIEW_INDEX_CONFIG<b>MAIN?<cr><lf>	Queries the device for the Main view configuration information. Returns a comma separated string containing the values: - View Name - Unit Index - Max Field Width - Offset - Scale

VIEW MAX FIELD WIDTH FUNCTIONS		
Function	**Syntax**	**Comments**
FundMag View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>FUND_MAG<b>SET<b><int_val><cr><lf>	Forces the FundMag view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>FUND_MAG?<cr><lf>	Queries the device for the FundMag view Max Field Width.
InPhase View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>IN_PHASE<b>SET<b><int_val><cr><lf>	Forces the InPhase view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>IN_PHASE?<cr><lf>	Queries the device for the InPhase view Max Field Width.
Quad View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>QUAD<b>SET<b><int_val><cr><lf>	Forces the Quad view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>QUAD?<cr><lf>	Queries the device for the Quad view Max Field Width.

Phase View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>PHASE<b>SET<b><int_val><cr><lf>	Forces the Phase view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>PHASE?<cr><lf>	Queries the device for the Phase view Max Field Width.
THD View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>THD<b>SET<b><int_val><cr><lf>	Forces the THD view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>THD?<cr><lf>	Queries the device for the THD view Max Field Width.
SigVolt View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>SIG_VOLT<b>SET<b><int_val><cr><lf>	Forces the SigVolt view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>SIG_VOLT?<cr><lf>	Queries the device for the SitVolt view Max Field Width.
RefVolt View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>REF_VOLT<b>SET<b><int_val><cr><lf>	Forces the RefVolt view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>REF_VOLT?<cr><lf>	Queries the device for the RefVolt view Max Field Width.
SigOffset View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>SIG_OFFSET<b>SET<b><int_val><cr><lf>	Forces the SigOffset view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>SIG_OFFSET?<cr><lf>	Queries the device for the SigOffset view Max Field Width.
TotalRatio View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>TOTAL_RATIO<b>SET<b><int_val><cr><lf>	Forces the TotalRatio view Max Field Width to the value passed in by <int_val>.

	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>TOTAL_RATIO?<cr><lf>	Queries the device for the TotalRatio view Max Field Width.
Frequency View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>FREQUENCY<b>SET<b><int_val><cr><lf>	Forces the Frequency view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>FREQUENCY?<cr><lf>	Queries the device for the Frequency view Max Field Width.
Main View Max Field Width	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>MAIN<b>SET<b><int_val><cr><lf>	Forces the Main view Max Field Width to the value passed in by <int_val>.
	PAVCMD<b>VIEW_MAX_FIELD_WIDTH<b>MAIN?<cr><lf>	Queries the device for the Main view Max Field Width.

VIEW UNITS FUNCTIONS		
Function	**Syntax**	**Comments**
FundMag View Units	PAVCMD<b>VIEW_UNITS<b>FUND_MAG<b>SET<b>V<cr><lf>	Forces the FundMag view units to V.
	PAVCMD<b>VIEW_UNITS<b>FUND_MAG<b>SET<b>MV<cr><lf>	Forces the FundMag view units to MV.
	PAVCMD<b>VIEW_UNITS<b>FUND_MAG<b>SET<b>PERCENT<cr><lf>	Forces the FundMag view units to %.
	PAVCMD<b>VIEW_UNITS<b>FUND_MAG<b>SET<b>DB<cr><lf>	Forces the FundMag view units to dB.
	PAVCMD<b>VIEW_UNITS<b>FUND_MAG<b>SET<b>Ratio<cr><lf>	Forces the FundMag view units to Ratio.
	PAVCMD<b>VIEW_UNITS<b>FUND_MAG?<cr><lf>	Queries the device for the FundMag view units. Return Values: 0 – "V" 1 – "mV" 2 – "Ratio" 3 – "Percent" 4 – "dB"

	PAVCMD<b>VIEW_UNITS<b>FUND_MAG<b>TEXT?<cr><lf>	Queries the device for the FundMag view units text. Return Values: "V" "mV" "Ratio" "Percent" "dB"
InPhase View Units	PAVCMD<b>VIEW_UNITS<b>IN_PHASE<b>SET<b>V<cr><lf>	Forces the InPhase view units to V.
	PAVCMD<b>VIEW_UNITS<b>IN_PHASE<b>SET<b>MV<cr><lf>	Forces the InPhase view units to MV.
	PAVCMD<b>VIEW_UNITS<b>IN_PHASE<b>SET<b>PERCENT<cr><lf>	Forces the InPhase view units to %.
	PAVCMD<b>VIEW_UNITS<b>IN_PHASE<b>SET<b>DB<cr><lf>	Forces the InPhase view units to dB.
	PAVCMD<b>VIEW_UNITS<b>IN_PHASE<b>SET<b>Ratio<cr><lf>	Forces the InPhase view units to Ratio.
	PAVCMD<b>VIEW_UNITS<b>IN_PHASE?<cr><lf>	Queries the device for the InPhase view units. Return Values: 0 – "V" 1 – "mV" 2 – "Ratio" 3 – "Percent" 4 – "dB"
	PAVCMD<b>VIEW_UNITS<b>IN_PHASE<b>TEXT?<cr><lf>	Queries the device for the InPhase view units text. Return Values: "V" "mV" "Ratio" "Percent" "dB"
Quad View Units	PAVCMD<b>VIEW_UNITS<b>QUAD<b>SET<b>V<cr><lf>	Forces the Quad view units to V.
	PAVCMD<b>VIEW_UNITS<b>QUAD<b>SET<b>MV<cr><lf>	Forces the Quad view units to MV.
	PAVCMD<b>VIEW_UNITS<b>QUAD<b>SET<b>PERCENT<cr><lf>	Forces the Quad view units to %.
	PAVCMD<b>VIEW_UNITS<b>QUAD<b>SET<b>DB<cr><lf>	Forces the Quad view units to dB.
	PAVCMD<b>VIEW_UNITS<b>QUAD<b>SET<b>Ratio<cr><lf>	Forces the Quad view units to Ratio.

	PAVCMD<b>VIEW_UNITS<b>QUAD?<cr><lf>	Queries the device for the Quad view units. Return Values: 0 – "V" 1 – "mV" 2 – "Ratio" 3 – "Percent" 4 – "dB"
	PAVCMD<b>VIEW_UNITS<b>QUAD<b>TEXT?<cr><lf>	Queries the device for the Quad view units text. Return Values: "V" "mV" "Ratio" "Percent" "dB"
Phase View Units	PAVCMD<b>VIEW_UNITS<b>PHASE<b>SET<b>360<cr><lf>	Forces the Phase view units to 360.
	PAVCMD<b>VIEW_UNITS<b>PHASE<b>SET<b>180<cr><lf>	Forces the Phase view units to 180.
	PAVCMD<b>VIEW_UNITS<b>PHASE?<cr><lf>	Queries the device for the Phase view units. Return Values: 5 – "380" 6 – "180"
	PAVCMD<b>VIEW_UNITS<b>PHASE<b>TEXT?<cr><lf>	Queries the device for the Phase view units text. Return Values: "360" "180"
THD View Units	PAVCMD<b>VIEW_UNITS<b>THD<b>SET<b>PERCENT<cr><lf>	Forces the THD view units to %.
	PAVCMD<b>VIEW_UNITS<b>THD<b>SET<b>DB<cr><lf>	Forces the THD view units to dB.
	PAVCMD<b>VIEW_UNITS<b>THD?<cr><lf>	Queries the device for the THD view units. Return Values: 3 – "Percent" 4 – "dB"

	PAVCMD<b>VIEW_UNITS<b>THD<b>TEXT?<cr><lf>	Queries the device for the THD view units text. Return Values: "Percent" "dB"
SigVolt View Units	PAVCMD<b>VIEW_UNITS<b>SIG_VOLT<b>SET<b>V<cr><lf>	Forces the SigVolt view units to V.
	PAVCMD<b>VIEW_UNITS<b>SIG_VOLT<b>SET<b>MV<cr><lf>	Forces the SigVolt view units to mV.
	PAVCMD<b>VIEW_UNITS<b>SIG_VOLT?<cr><lf>	Queries the device for the SigVolt view units. Return Values: 0 – "V" 1 – "mV"
	PAVCMD<b>VIEW_UNITS<b>SIG_VOLT<b>TEXT?<cr><lf>	Queries the device for the SigVolt view units text. Return Values: "V" "mV"
RefVolt View Units	PAVCMD<b>VIEW_UNITS<b>REF_VOLT<b>SET<b>V<cr><lf>	Forces the RefVolt view units to V.
	PAVCMD<b>VIEW_UNITS<b>REF_VOLT<b>SET<b>MV<cr><lf>	Forces the RefVolt view units to mV.
	PAVCMD<b>VIEW_UNITS<b>REF_VOLT?<cr><lf>	Queries the device for the RefVolt view units. Return Values: 0 – "V" 1 – "mV"
	PAVCMD<b>VIEW_UNITS<b>REF_VOLT<b>TEXT?<cr><lf>	Queries the device for the RefVolt view units text. Return Values: "V" "mV"
SigOffset View Units	PAVCMD<b>VIEW_UNITS<b>SIG_OFFSET<b>SET<b>V<cr><lf>	Forces the SigOffset view units to V.
	PAVCMD<b>VIEW_UNITS<b>SIG_OFFSET<b>SET<b>MV<cr><lf>	Forces the SigOffset view units to mV.
	PAVCMD<b>VIEW_UNITS<b>SIG_OFFSET?<cr><lf>	Queries the device for the SigOffset view units. Return Values: 0 – "V" 1 – "mV"

	PAVCMD<b>VIEW_UNITS<b>SIG_OFFSET<b>TEXT ?<cr><lf>	Queries the device for the SigOffset view units text. Return Values: "V" "mV"
TotalRatio View Units	PAVCMD<b>VIEW_UNITS<b>TOTAL_RATIO<b>SET <b>PERCENT<cr><lf>	Forces the Total Ratio view units to %.
	PAVCMD<b>VIEW_UNITS<b>TOTAL_RATIO<b>SET <b>DB<cr><lf>	Forces the Total Ratio view units to dB.
	PAVCMD<b>VIEW_UNITS<b>TOTAL_RATIO<b>SET <b>RATIO<cr><lf>	Forces the Total Ratio view units to Ratio.
	PAVCMD<b>VIEW_UNITS<b>TOTAL_RATIO?<cr><l f>	Queries the device for the Total Ratio view units. Return Values: 2 – "Ratio" 3 – "Percent" 4 – "dB"
	PAVCMD<b>VIEW_UNITS<b>TOTAL_RATIO<b>TE XT?<cr><lf>	Queries the device for the Total Ratio view units text. Return Values: "Ratio" "Percent" "dB"
Frequency View Units	PAVCMD<b>VIEW_UNITS<b>FREQUENCY<b>SET< b>HZ<cr><lf>	Forces the Frequency view units to Hz.
	PAVCMD<b>VIEW_UNITS<b>FREQUENCY<b>SET< b>KHZ<cr><lf>	Forces the Frequency view units to KHz.
	PAVCMD<b>VIEW_UNITS<b>FREQUENCY?<cr><lf>	Queries the device for the Frequency view units. Return Values: 7 – "Hz" 8 – "KHz"
	PAVCMD<b>VIEW_UNITS<b>FREQUENCY<b>TEXT ?<cr><lf>	Queries the device for the Frequency view units text. Return Values: "Hz" "KHz"

Main View Units	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>V<cr><lf>	Forces the Main view units to V.
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>MV<cr><lf>	Forces the Main view units to MV.
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>PERCENT<cr><lf>	Forces the Main view units to %.
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>DB<cr><lf>	Forces the Main view units to dB.
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>Ratio<cr><lf>	Forces the Main view units to Ratio.
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>360<cr><lf>	Forces the Main view units to 360.
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>SET<b>180<cr><lf>	Forces the Main view units to 180.
	PAVCMD<b>VIEW_UNITS<b>MAIN?<cr><lf>	Queries the device for the Main view units. Return Values: 0 – "V" 1 – "mV" 2 – "Ratio" 3 – "Percent" 4 – "dB" 5 – "360" 6 – "180"
	PAVCMD<b>VIEW_UNITS<b>MAIN<b>TEXT?<cr><lf>	Queries the device for the Main view units text. Return Values: "V" "mV" "Ratio" "Percent" "dB" "360" "180"

VIEW OFFSET FUNCTIONS		
Function	**Syntax**	**Comments**
FundMag Offset	PAVCMD<b>VIEW_OFFSET<b>FUND_MAG<b>SET<b><float_val><cr><lf>	Forces the FundMag offset to the value specified by <float_val>.

	PAVCMD<b>VIEW_OFFSET<b>FUND_MAG?<cr><lf>	Queries the device for the FundMag offset value.
InPhase Offset	PAVCMD<b>VIEW_OFFSET<b>IN_PHASE<b>SET<b><float_val><cr><lf>	Forces the InPhase offset to the value specified by <float_val>.
	PAVCMD<b>VIEW_OFFSET<b>IN_PHASE?<cr><lf>	Queries the device for the InPhase offset value.
Quad Offset	PAVCMD<b>VIEW_OFFSET<b>QUAD<b>SET<b><float_val><cr><lf>	Forces the Quad offset to the value specified by <float_val>.
	PAVCMD<b>VIEW_OFFSET<b>QUAD?<cr><lf>	Queries the device for the Quad offset value.
Phase Offset	PAVCMD<b>VIEW_OFFSET<b>PHASE<b>SET<b><float_val><cr><lf>	Forces the Phase offset to the value specified by <float_val>.
	PAVCMD<b>VIEW_OFFSET<b>PHASE?<cr><lf>	Queries the device for the Phase offset value.
SigVolt Offset	PAVCMD<b>VIEW_OFFSET<b>SIG_VOLT<b>SET<b><float_val><cr><lf>	Forces the SigVolt offset to the value specified by <float_val>.
	PAVCMD<b>VIEW_OFFSET<b>SIG_VOLT?<cr><lf>	Queries the device for the SigVolt offset value.
RefVolt Offset	PAVCMD<b>VIEW_OFFSET<b>REF_VOLT<b>SET<b><float_val><cr><lf>	Forces the RefVolt offset to the value specified by <float_val>.
	PAVCMD<b>VIEW_OFFSET<b>REF_VOLT?<cr><lf>	Queries the device for the RefVolt offset value.

VIEW SCALE FUNCTIONS		
Function	**Syntax**	**Comments**
FundMag Scale	PAVCMD<b>VIEW_SCALE<b>FUND_MAG<b>SET<b><float_val><cr><lf>	Forces the FundMag scale to the value specified by <float_val>.
	PAVCMD<b>VIEW_SCALE<b>FUND_MAG?<cr><lf>	Queries the device for the FundMag scale value.
InPhase Scale	PAVCMD<b>VIEW_SCALE<b>IN_PHASE<b>SET<b><float_val><cr><lf>	Forces the InPhase scale to the value specified by <float_val>.
	PAVCMD<b>VIEW_SCALE<b>IN_PHASE?<cr><lf>	Queries the device for the InPhase scale value.
Quad Scale	PAVCMD<b>VIEW_SCALE<b>QUAD<b>SET<b><float_val><cr><lf>	Forces the Quad scale to the value specified by <float_val>.
	PAVCMD<b>VIEW_SCALE<b>QUAD?<cr><lf>	Queries the device for the Quad scale value.
Phase Scale	PAVCMD<b>VIEW_SCALE<b>PHASE<b>SET<b><float_val><cr><lf>	Forces the Phase scale to the value specified by <float_val>.
	PAVCMD<b>VIEW_SCALE<b>PHASE?<cr><lf>	Queries the device for the Phase scale value.
SigVolt Scale	PAVCMD<b>VIEW_SCALE<b>SIG_VOLT<b>SET<b><float_val><cr><lf>	Forces the SigVolt scale to the value specified by <float_val>.
	PAVCMD<b>VIEW_SCALE<b>SIG_VOLT?<cr><lf>	Queries the device for the SigVolt scale value.
RefVolt Scale	PAVCMD<b>VIEW_SCALE<b>REF_VOLT<b>SET<b><float_val><cr><lf>	Forces the RefVolt scale to the value specified by <float_val>.
	PAVCMD<b>VIEW_SCALE<b>REF_VOLT?<cr><lf>	Queries the device for the RefVolt scale value.

VIEW INDEX FUNCTIONS		
Function	**Syntax**	**Comments**
Max Field Width	PAVCMD<b>VIEW_INDEX_MAX_FIELD_WIDTH<b><view_idx><b>SET<b><int_val><cr><lf>	Forces the view specified by <view_idx> Max Field Width to the value specified by <int_val>. Possible <view_idx> values: 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main"
	PAVCMD<b>VIEW_INDEX_MAX_FIELD_WIDTH<b><view_idx>? <cr><lf>	Queries the device for the Max Field Width value of the view specified by <view_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main"

Units	PAVCMD<b>VIEW_INDEX_UNITS<b><view_idx><b>SET<b><int_val><cr><lf>	Forces the view specified by <view_idx> units to the value specified by <int_val>. Possible <view_idx> values: 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main" Possible <int_val> values: 0 - "V" 1 - "mV" 2 - "Ratio" 3 - "Percent" 4 - "DB" 5 - "360" 6 - "180" 7 - "Hz" 8 - "KHz"

	PAVCMD<b>VIEW_INDEX_UNITS<b><view_idx>?<cr><lf>	Queries the device for the units value of the view specified by <view_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main" Return Values: 0 - "V" 1 - "mV" 2 - "Ratio" 3 - "Percent" 4 - "DB" 5 - "360" 6 - "180" 7 - "Hz" 8 - "KHz"

	PAVCMD<b>VIEW_INDEX_UNITS<b><view_idx><b>TEXT? <cr><lf>	Queries the device for the units text value of the view specified by <view_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main" Return Values: "V" "mV" "Ratio" "Percent" "DB" "360" "180" "Hz" "KHz"
Offset	PAVCMD<b>VIEW_INDEX_OFFSET<b><view_idx><b>SET<b><float_val><cr><lf>	Forces the view specified by <view_idx> offset to the value specified by <float_val>. Possible <view_idx> values: 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main"

	PAVCMD<b>VIEW_INDEX_OFFSET<b><view_idx>?<cr><lf>	Queries the device for the units value of the view specified by <view_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main"
Scale	PAVCMD<b>VIEW_INDEX_SCALE<b><view_idx><b>SET<b><float_val><cr><lf>	Forces the view specified by <view_idx> scale to the value specified by <float_val>. Possible <view_idx> values: 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main"

	PAVCMD<b>VIEW_INDEX_SCALE<b><view_idx>?<cr><lf>	Queries the device for the scale value of the view specified by <view_idx>. 0 – "Fund Mag" 1 – "In Phase" 2 – Quad" 3 – "Phase" 4 – "THD" 5 – "Sig Volt" 6 – "Ref Volt" 7 – "Sig Offset" 8 – "Total Ratio" 9 – "Freq" 10 – "Main"

REFVOLT FUNCTIONS		
Function	**Syntax**	**Comments**
Reference Voltage View	PAVCMD<b>REF?<cr><lf>	Queries the device for the reference voltage view index. Return Values: 1 – "Ref Total RMS (AC)" 2 – "Ref Total RMS (AC+DC)" 3 – "Ref DC"
	PAVCMD<b>REF <b>TEXT?<cr><lf>	Queries the device for the reference voltage view text. Return Values: "Ref Total RMS (AC)" "Ref Total RMS (AC+DC)" "Ref DC"

	PAVCMD<b>REF <b>SET<b><view_idx><cr><lf>	Forces the Reference Voltage View to the view specified by <view_idx>: 1 – "Ref Total RMS (AC)" 2 – "Ref Total RMS (AC+DC)" 3 – "Ref DC"
	PAVCMD<b>REF<b>TOTALRMS_AC<cr><lf>	Forces the Reference Voltage View to "Ref Total RMS (AC)"
	PAVCMD<b>REF<b>TOTALRMS_ACDC<cr><lf>	Forces the Reference Voltage View to "Ref Total RMS (AC+DC)"
	PAVCMD<b>REF<b>DC<cr><lf>	Forces the Reference Voltage View to "Ref DC"
	PAVCMD<b>REF<b>RMS?<cr><lf>	Queries the device for the Ref Total RMS AC value.
	PAVCMD<b>REF<b>ACDC?<cr><lf>	Queries the device for the Ref Total RMS AC+DC value.
	PAVCMD<b>REF<b>DC?<cr><lf>	Queries the device for the Ref DC value.

SIGVOLT FUNCTIONS		
Function	**Syntax**	**Comments**
Signal Voltage View	PAVCMD<b>SIG?<cr><lf>	Queries the device for the signal voltage view index. Return Values: 0 – "Sig Total Sum" 1 – "Sig Total RMS (AC)" 2 – "Sig Total RMS (AC+DC)" 3 – "Sig DC"

	PAVCMD<b>SIG <b>TEXT?<cr><lf>	Queries the device for the signal voltage view text. Return Values: "Sig Total Sum" "Sig Total RMS (AC)" "Sig Total RMS (AC+DC)" "Sig DC"
	PAVCMD<b>SIG <b>SET<b><view_idx><cr><lf>	Forces the Signal Voltage View to the view specified by <view_idx>: 0 – "Sig Total Sum" 1 – "Sig Total RMS (AC)" 2 – "Sig Total RMS (AC+DC)" 3 – "Sig DC"
	PAVCMD<b>SIG <b>TOTALSUM <cr><lf>	Forces the Signal Voltage View to "Sig Total Sum"
	PAVCMD<b>SIG <b>TOTALRMS_AC<cr><lf>	Forces the Signal Voltage View to "Sig Total RMS (AC)"
	PAVCMD<b>SIG <b>TOTALRMS_ACDC<cr><lf>	Forces the Signal Voltage View to "Sig Total RMS (AC+DC)"
	PAVCMD<b>SIG <b>DC<cr><lf>	Forces the Signal Voltage View to "Sig DC"
	PAVCMD<b>SIG<b>TOTAL_SUM?<cr><lf>	Queries the device for the Sig Total Sum value.
	PAVCMD<b> SIG<b>RMS?<cr><lf>	Queries the device for the Sig Total RMS AC value.
	PAVCMD<b>SIG<b>RMS_ACDC?<cr><lf>	Queries the device for the Sig Total RMS AC+DC value.

	PAVCMD<b>SIG<b>DC?<cr><lf>	Queries the device for the DC value.

INDEPENDENT COMPONENT FUNCTIONS		
Function	**Syntax**	**Comments**
Signal Offset	PAVCMD<b>SIG<b>OFFSET?<cr><lf>	Queries the device for the Signal Offset float value.
Total Ratio	PAVCMD<b>TOTAL_RATIO?<cr><lf>	Queries the device for the Total Ratio float value.
Frequency	PAVCMD<b>FREQ?<cr><lf>	Queries the device for the Frequency float value.
Sample Rate	PAVCMD<b>SAMPLE_RATE_INDEX?<cr><lf>	Queries the device for the Sample RaIndex int value.

LVDT FUNCTIONS		
Function	**Syntax**	**Comments**
Enable	PAVCMD<b>LVDT<b>ENABLED?<cr><lf>	Queries the device for whether LVDT calculations are enabled or disabled. – 0 = Disabled, 1=Enabled
	PAVCMD<b>LVDT<b>ENABLED<b>TEXT?<cr><lf>	Queries the device for whether LVDT calculations are enabled or disabled. – "Disabled" or "Enabled"
	PAVCMD<b>LVDT<b>ENABLE<cr><lf>	Forces the LVDT calculations to be enabled.
	PAVCMD<b>LVDT<b>DISABLE<cr><lf>	Forces the LVDT calculations to be disabled.
Position	PAVCMD<b>LVDT<b>POSITION?<cr><lf>	Queries the device for the LVDT Position.

Type	PAVCMD<b>LVDT<b>TYPE?<cr><lf>	Queries the device for the LVDT Type (2=2-Wire, 3=3-Wire, 4=4-Wire).
	PAVCMD<b>LVDT<b>TYPE<b>TEXT?<cr><lf>	Queries the device for the LVDT Type text value. ("2-Wire", "3-Wire" or "4-Wire")
	PAVCMD<b>LVDT<b>TYPE<b>SET<b><lvdt_type_idx><cr><lf>	Forces the LVDT Type to: 2 – "2-Wire" 3 – "3-Wire" 4 – "4-Wire"
	PAVCMD<b>LVDT<b>TYPE<b>2-WIRE<cr><lf>	Force the LVDT Type to 2-Wire.
	PAVCMD<b>LVDT<b>TYPE<b>3-WIRE<cr><lf>	Force the LVDT Type to 3-Wire.
	PAVCMD<b>LVDT<b>TYPE<b>4-WIRE<cr><lf>	Force the LVDT Type to 4-Wire.
Signal	PAVCMD<b>LVDT<b>SIGNAL?<cr><lf>	Queries the device for the LVDT Signal mode. (0="Fund", 1="Total")
	PAVCMD<b>LVDT<b>SIGNAL<b>TEXT?<cr><lf>	Queries the device for the LVDT Signal text value. ("Fund" or "Total")
	PAVCMD<b>LVDT<b>SIGNAL<b>SET<b><signal_idx><cr><lf>	Force the LVDT Signal to: 0 = "Fund" or 1 = "Total"
	PAVCMD<b>LVDT<b>SIGNAL<b>FUND <cr><lf>	Forces the LVDT Signal to "Fund"
	PAVCMD<b>LVDT<b>SIGNAL<b>TOTAL<cr><lf>	Forces the LVDT Signal to "Total"
VA	PAVCMD<b>LVDT<b>VA?<cr><lf>	Queries the device for LVDT VA value.
VB	PAVCMD<b>LVDT<b>VB?<cr><lf>	Queries the device for LVDT VB value.
Phase Offset	PAVCMD<b>LVDT<b>POFF?<cr><lf>	Queries the device for LVDT Phase Offset.
	PAVCMD<b>LVDT<b>POFF<b>SET<b><float_val><cr><lf>	Forces the LVDT phase offset to the desired value.

Scale	PAVCMD<b>LVDT<b>SCALE?<cr><lf>	Queries the device for LVDT Scale.
	PAVCMD<b>LVDT<b>SCALE<b>SET<b><float_val><cr><lf>	Forces the LVDT scale to the desired value.
LVDT 4-Wire Algorithm	PAVCMD<b>4-WIRE?<cr><lf>	Queries the device for the current LVDT 4-Wire Algorithm Option Index. Return Values: 0 – "V(a), V(a + b)" 1 – "V(a), V(b)"
	PAVCMD<b> 4-WIRE<b>TEXT?<cr><lf>	Queries the device for the current 4-Wire Algorithm Option Text. Return Values: "V(a) _ V(a+b)" "V(a) _ V(b)"
	PAVCMD<b>4-WIRE<b>SET<b><option_idx><cr><lf>	Forces the LVDT 4-Wire Algorithm Option Index based on the value of <option_idx>. 0 – "V(a), V(a+b)" 1 – "V(a), V(b)"
	PAVCMD<b> 4-WIRE<b>VA_VAPLUSVB<cr><lf>	Forces the LVDT 4-Wire Algorithm Option to "V(a), V(a+b)".
	PAVCMD<b>4-WIRE<b>VA_VB<cr><lf>	Forces the LVDT 4-Wire Algorithm Option to "V(a), V(b)".

REFERENCE COMPONENT FUNCTIONS		
Function	**Syntax**	**Comments**
Ref Total	PAVCMD<b>REF<b>TOTAL?<cr><lf>	Queries the device for the ReferenceTotal float value.

Ref Fund/Mag	PAVCMD<b>REF<b>FUND_MAG?<cr><lf>	Queries the device for the Reference Fundamental (Magnitude) float value.
Ref InPhase	PAVCMD<b>REF<b>IN_PHASE?<cr><lf>	Queries the device for the Reference In-Phase float value.
Ref Quad	PAVCMD<b>REF<b>QUAD?<cr><lf>	Queries the device for the Reference Quad float value.
Ref Phase	PAVCMD<b>REF<b>PHASE?<cr><lf>	Queries the device for the Reference Phase float value.
Ref THD	PAVCMD<b>REF<b>THD?<cr><lf>	Queries the device for the Reference THD float value.
Ref Total Ratio	PAVCMD<b>REF<b>TOTAL_RATIO?<cr><lf>	Queries the device for the Reference Total Ratio float value.
Ref Fund/Mag Ratio	PAVCMD<b>REF<b>FUND_MAG_RATIO?<cr><lf>	Queries the device for the Reference Fundamental/Magnitude Ratio float value.
Ref InPhase Ratio	PAVCMD<b>REF<b>IN_PHASE_RATIO?<cr><lf>	Queries the device for the Reference InPhase Ratio float value.
Ref Quad Ratio	PAVCMD<b>REF<b>QUAD_RATIO?<cr><lf>	Queries the device for the Reference Quad Ratio float value.

REFERENCE GENERATOR FUNCTIONS		
Function	**Syntax**	**Comments**
	REF_GEN<b>FREQ?<cr><lf>	Queries the device for the Reference Generator's Frequency value.
	REF_GEN<b>FREQ<b><float_val><cr><lf>	Forces the Reference Generator's Frequency to the desired value.

	REF_GEN<b>OVER_CUR?<cr><lf>	Queries the device for whether the Reference Generator is in an Over Current situation. "OVER_CUR" is returned when in over current. "NO OVER_CUR" is returned if not in over current.
	REF_GEN<b>OVER_CUR<b>RESET<cr><lf>	Tells the device to RESET the over current flag.
	REF_GEN<b>REM_SENSE?<cr><lf>	Queries the device for whether or not Remote Sense is enabled. "ON" is returned if enabled. "OFF" is returned if disabled.
	REF_GEN<b>REM_SENSE<b>ON<cr><lf>	Tells the device to enable Remote Sense.
	REF_GEN<b>REM_SENSE<b>OFF<cr><lf>	Tells the device to disable Remote Sense.
	REF_GEN<b>SENSE_DIR?<cr><lf>	Queries the device for which direction remote sense is configured for: "FRONT" is returned if configured for front. "BACK" is returned if configured for back.
	REF_GEN<b>SENSE_DIR<b>FRONT<cr><lf>	Tells the device to configure remote sense for FRONT.
	REF_GEN<b>SENSE_DIR<b>BACK<cr><lf>	Tells the device to configure remote sense for BACK.

	REF_GEN<b>STATE?<cr><lf>	Queries the device for the current state of the Reference Generator. "COLSE" is returned if the Reference Generator is currently outputting. "OPEN" is returned if the Reference Generator is not outputting.
	REF_GEN<b>STATE<b>CLOSE<cr><lf>	Tells the device to have the Reference Generator enable the output.
	REF_GEN<b>STATE<b>OPEN<cr><lf>	Tells the device to have the Reference Generator disable the output.
	REF_GEN<b>VOLT?<cr><lf>	Queries the device for the current Reference Generator voltage.
	REF_GEN<b>VOLT<b><float_val><cr><lf>	Tells the device to configure the Reference Generator to the desired Voltage.
	REF_GEN<b>MEAS_CUR?<cr><lf>	Queries the device for the Reference Generator's Measured Current value.

3.4 PAV-2250 Native (Legacy)

The PAV-2250 Native language (referenced from here forward as PAV-2250Legacy) is only supported via the IEEE-488.1. The language is available to provide backwards compatibility to the 2250 units. Only the features that were available for the 2250 are supported with this language.

PAV2250-LEGACY FNC FUNCTIONS		
Function	**Syntax**	**Comments**
Switches Main View Display	FNC<b>RMSV<cr><lf>	Forces Main View to display Sig Volt View – Total Sum
	FNC<b>FUND<cr><lf>	Forces Main View to Fundamental Magnitude (FundMag)
	FNC<b>INPH<cr><lf>	Forces Main View to display In Phase.
	FNC<b>QUAD<cr><lf>	Forces Main View to display Quad.
	FNC<b>PANG<cr><lf>	Forces Main View to display Phase.
	FNC<b>DSTR<cr><lf>	Forces Main View to display THD.

PAV2250-LEGACY FTH FUNCTIONS		
Function	**Syntax**	**Comments**
Switches Main View Display	FTH<b>FREQ<cr><lf>	Queries the device for the current Frequency value.
	FTH<b>MAIN<cr><lf>	Queries the device for the current value that is displayed on the "Main View". Use one of the "FNC" commands to initially assign Main View to desired setting.

PAV2250-LEGACY SET FUNCTIONS		
Function	**Syntax**	**Comments**

Switches	SET<b>HARM<b><harm><cr><lf>	Forces view to the desired Harmonic designated by <harm>. NOTE – this a 1-based value (not zero based). Valid values are 1 – 16.
	SET<b>FREQ<b><float_val><cr><lf>	Forces Frequency to the value <float_val>
	SET<b>FREQ<b>OFF<cr><lf>	This is a no-op in the 2250A.
	SET<b>RREF<b>ON<cr><lf>	Forces the Read Mode of the 2250A to be in "Ref" mode.
	SET<b>RREF<b>OFF<cr><lf>	Forces the Read Mode of the 2250A to return to "Sig/Ref" mode.
	SET<b>TKHD<b>ON<cr><lf>	Forces the "Track Hold' feature to be on which freezes the current values.
	SET<b>TKHD<b>OFF<cr><lf>	Forces the "Track Hold" feature to be off allowing the system to once again acquire updates.
	SET<b>POFF<b><float_val><cr><lf>	Forces the Phase to have an Offset of the value specified in <float_val>.
	SET<b>VARI<b><float_val><cr><lf>	Forces the view currently displayed in the Main view to have a Scaling value as specified in <float_val>.
	SET<b>OFFS<b><float_val><cr><lf>	Forces the view currently displayed in the Main view to have an Offset value as specified in <float_val>.

	SET<b>P180<b>ON<cr><lf>	If "Phase" is displayed as the Main View, this call ensures the data will be displayed in 180 mode. If "Phase" is not the Main View, this call is a "no-op".
	SET<b>P180<b>OFF<cr><lf>	If "Phase" is displayed as the Main View, this call ensures the data will be displayed in 360 mode. If "Phase" is not the Main View, this call is a "no-op".
	SET<b>DEVI<b><float_val><cr><lf>	PAV2250A currently does not support this feature.
	SET<b>DECI<b>ON<cr><lf>	Forces all views capable of going into dB mode to go into dB mode. (Magnitude, InPhase, Quad, Total Ratio and THD)
	SET<b>DECI<b>OFF<cr><lf>	Forces all views capable of going into dB mode to go into Ratio mode. (Magnitude, InPhase, Quad, Total Ratio and THD)
	SET<b>DECI<b><float_val><cr><lf>	Forces all views capable of going into dB mode or Ratio mode to do so based upon the value. 0 indicates Ratio and 1 indicates dB mode.
	SET<b>CALB<cr><lf>	Forces the unit to perform calibration.
	SET<b>ACAL<b>ON<cr><lf>	Currently this is a no-op in the 2250A.
	SET<b>ACAL<b>OFF<cr><lf>	Currently this is a no-op in the 2250A.

	SET<b>RTIO<b>ON<cr><lf>	Forces all views capable of Ratio mode to be in Ratio. (Magnitude, InPhase, Quad, Total Ratio and THD) Ratio mode also forces the PAV-2250A to be in "Auto Range" for Signal and Reference.
	SET<b>RTIO<b>OFF<cr><lf>	Forces all views that were in Ratio mode to now be in Absolute mode which forces each view to their default display unit.
	SET<b>GET<b>ON<cr><lf>	Signals the PAV-2250A to start buffering all incoming commands (does not execute them until a GPIB command to GET is performed)
	SET<b>GET<b>OFF<cr><lf>	Signals the PAV-2250A to stop buffering incoming commands. All future commands will be executed as they are received one at a time.
	SET<b>AVRG<b><float_val><cr><lf>	Forces the unit's Time Window to the specified (<float_val>) interval. Values will be averaged across this time window.
	SET<b>AVRG<b>OFF<cr><lf>	Forces the unit to be in "Auto" Time Window. The unit will decide the appropriate value.

	SET<b>SRQ<b>ON<cr><lf>	Instructs the unit to assert SRQ only if the data is considered to be stable.
	SET<b>SRQ<b>OFF<cr><lf>	Instructs the unit to not assert SRQ.
	SET<b>SRQ<b>CONT<cr><lf>	Instructs the unit to assert SRQ continuously regardless of whether the data is stable.

PAV2250-LEGACY SRX FUNCTIONS		
Function	**Syntax**	**Comments**
Volt	SRX<b>VOLT<b>AUTO<cr><lf>	Forces the unit to be in "Auto" mode for both the Signal and Reference.
	SRX<b>VOLT<b><float_val><cr><lf>	Forces the unit to be in the appropriate Signal range mode based upon the value of <float_val>. The Reference range is forced to be "Auto".
Ratio	SRX<b>RTIO<b>AUTO<cr><lf>	Currently this is a no-op in the 2250A.

	SRX<b>RTIO<b><float_val><cr><lf>	Forces the PAV into one of the Legacy Ratio Ranges: - Range 0 :if <float_val> <= 0.01f - Range 1 if <float_val> <= 0.1f - Range 2 if <float_val> <= 1.0f - Range 3 if <float_val> <= 10.0f - Range 4 if <float_val> <= 100.0f - Range 5 if <float_val> <= 1000.0f

4 PAV-2250A USB Protocol

The 22500A USB interface supports only the PAV-2250A Native Language. Sending commands via the USB interface require the following protocol:

Number of Bytes to be sent (2 bytes)	Command ID (2210 (i.e. 0x08CA) (2 bytes)	Data

The following is code snippets from the PAV2250ADll (USBComm.cpp) that makes calls to the Cypress CyAPI.lib file to sending commands to the 2250A:

```
bool USB_WriteMsg(char* szMsg, bool bExpectReply, char*pszReply)
{
        bool bSuccess = false;

   unsigned short usTotalBytes = (unsigned short)strlen(szMsg) + 4;  // Length of Message + 2
bytes for Bytes sent + 2 bytes for Command
   unsigned short usCommand = 0x08CA;         // 2250 (0x08CA) Command

   char           szData[256];
   LONG           nDataCnt = 0;
        LONG           BytesToRead = 0;
   byte           loByte, hiByte;

   //*****************************************
   // Format data to be sent
   //  Protocol:
   //       (16 bits) Number of bytes to be sent
   //       (16 bits) Command ID (0x08CA) for 2250
   //       szMsg - data message
   //*****************************************

   // Low byte of Total Bytes to send
   loByte = (byte)(usTotalBytes & 0x00FF);
   // High byte of Total Bytes to send
   hiByte = (byte)(usTotalBytes >> 8);

   szData[nDataCnt++] = loByte;
   szData[nDataCnt++] = hiByte;

   // Low byte of Command
   loByte = (byte)(usCommand & 0x00FF);
   // High byte of Command
   hiByte = (byte)(usCommand >> 8);

   szData[nDataCnt++] = loByte;
   szData[nDataCnt++] = hiByte;

   // Message Data
   for (int i = 0; i < (int)strlen(szMsg); i++)
      szData[nDataCnt++] = szMsg[i];

        if(glb_pUSBDevice)
        {
                if(glb_pUSBDevice->IsOpen())
                {
                        short   numOfTries = 0;
                        do
                        {
```

```
            // Write Data Message
                                if (glb_pUSBDevice->BulkOutEndPt)
                                {
                                        if (!glb_pUSBDevice->BulkOutEndPt-
>XferData((PUCHAR)&szData, nDataCnt))
                {
                   ReinitUSB();
                   break;
                }
            }

            // Get Reply if one is expected
            if (bExpectReply)
            {
                                    if (glb_pUSBDevice->BulkInEndPt)
                                    {
                  // Read data
                  BytesToRead = MAX_REPLY_BUFFER;
                       unsigned char   aReceiveBuffer[MAX_REPLY_BUFFER];

                  for (int i = 0; i < MAX_REPLY_BUFFER; i++)
                     aReceiveBuffer[i] = 0;

                  glb_pUSBDevice->BulkInEndPt->TimeOut = 10000;  // 10 second timeout
                                bSuccess = glb_pUSBDevice->BulkInEndPt->XferData(aReceiveBuffer,
BytesToRead);
                                        numOfTries++;
                                        if(!bSuccess)
                                        {
                                           Wait(500);
                                        }
                  else
                  {
                     strcpy(pszReply, (char *)aReceiveBuffer);
                  }
                        }
               else
                  bSuccess = true;
            }
            else
               bSuccess = true;
                    }while((!bSuccess) && (numOfTries < USB_MAX_RETRY));
               }
               else
               {
         ReinitUSB();
               }
        }

        return bSuccess;
}
```

5 PAV-2250A Ethernet Protocol

The 2250A Ethernet interface supports only the PAV-2250A Native Language. Sending commands via the Ethernet interface requires the creation and connection via a TCP/IP socket.

The following code snippet, `CreateClientSocket()` from the PAV2250ADll (Ethernet.cpp) makes calls to the Winsock API to create and connect a TCP/IP socket to send commands to the 2250A. Note the code utilizes the `PingHost()` call to make sure that the IP address specified for the 2250A is reachable before attempting to create the socket. This avoids waiting for the socket timeout in the `connect()` call if the device is not reachable. The code snippet, `CloseClientSocket()` closes the socket connection.

```
int CreateClientSocket(char *pszIPAddr, int nPort, SOCKET* s)
{
   WSADATA      wsaData;
   SOCKET        sock;
   SOCKADDR_IN   ServerAddr;

   int result;

   /* Before trying to make a connection to the server, ping it to make sure it's reachable */
   result = PingHost(pszIPAddr);
   if (result != 0)
      return ETHER_CANNOT_ESTABLISH_CONNECTION;

   // Initialize Winsock version 2.2
   WSAStartup(MAKEWORD(2,2), &wsaData);

   // Create a new socket to make a TCP client connection
   sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
   setsockopt( sock, SOL_SOCKET, SO_RCVTIMEO, (char*)&RECEIVE_TIMEOUT, sizeof(int) );
   setsockopt( sock, SOL_SOCKET, SO_SNDTIMEO, (char*)&SEND_TIMEOUT, sizeof(int) );

   // set to no_delay to insure quick ack
   result = setsockopt( sock, IPPROTO_TCP, TCP_NODELAY, (char*)&NO_DELAY, sizeof(int) );

   // Setup a SOCKADDR_IN structure that will be used to connect
   // to the listening server on the Port.
   ServerAddr.sin_family = AF_INET;
   ServerAddr.sin_port = htons(nPort);
   ServerAddr.sin_addr.s_addr = inet_addr(pszIPAddr);

   // Make a connection to the server with socket sock
   connect(sock, (const struct sockaddr *)&ServerAddr, sizeof(ServerAddr));
   *s = sock;
   Socket = sock;  // put it into global socket
   return ETHER_SUCCESS;
}

int CloseClientSocket(SOCKET s)
{
  closesocket(s);
  WSACleanup();
  return ETHER_SUCCESS;
}
```

After a socket connection is made to the 2250A, device log-in is required. 2250A Ethernet login is accomplished by sending **"NAII\r\n"** command via the Ethernet connection to the 2250A.

The following code snippets, `Ethernet_WriteMsg()`, `SendEthernetMsg()` and `ReadEthernetMsg()` from the PAV2250ADll (Ethernet.cpp) makes calls to the Winsock API to send and receive messages to and from the 2250A.

```
#define MSG_MAX_SIZE         1500      /* Maximum number of bytes to send */
#define RECV_MSG_MAX_SIZE    1500      /* Maximum number of bytes that can be read */

bool Ethernet_WriteMsg(SOCKET s, char* szMsg, bool bExpectReply, char* pszReply)
{
   bool bSuccess = false;
   char aReceiveBuffer[RECV_MSG_MAX_SIZE];
   int nBytesRead = 0;

   if (SendEthernetMsg(s, &szMsg[0], strlen(szMsg)) == ETHER_SEND_ERROR)
      return bSuccess;

   if (bExpectReply)
   {
      if (ReadEthernetMsg(s, RECV_MSG_MAX_SIZE, aReceiveBuffer, &nBytesRead) == ETHER_RECV_ERROR)
         return bSuccess;

      strncpy(pszReply, (char *)aReceiveBuffer, nBytesRead);
   }

   bSuccess = true;
   return bSuccess;
}

int SendEthernetMsg(SOCKET s, char *pszMessage, int nMessageLen)
{
   int ret;
   char sendbuff[MSG_MAX_SIZE];
   int nLeft;
   int nIndex;
   int status = 0;

   // Copy the data to be sent to the buffer
   for (nIndex = 0; nIndex < nMessageLen; nIndex++)
      sendbuff[nIndex] = pszMessage[nIndex];

   nLeft = nMessageLen;
   nIndex = 0;

   while (nLeft > 0)
   {
      ret = send(s, &sendbuff[nIndex], nLeft, 0);
      // It seems we sent some data
      if (ret != SOCKET_ERROR)
      {
         nLeft -= ret;
         nIndex += ret;
      }
      // got SOCKET_ERROR
      else
      {
         status = ETHER_SEND_ERROR;
         break;
      }
   }

   if (nLeft > 0)
      status = ETHER_SEND_ERROR; /* ERROR */
   else
      status = ETHER_SUCCESS;  /* SUCCESS */
   return status;
}
```

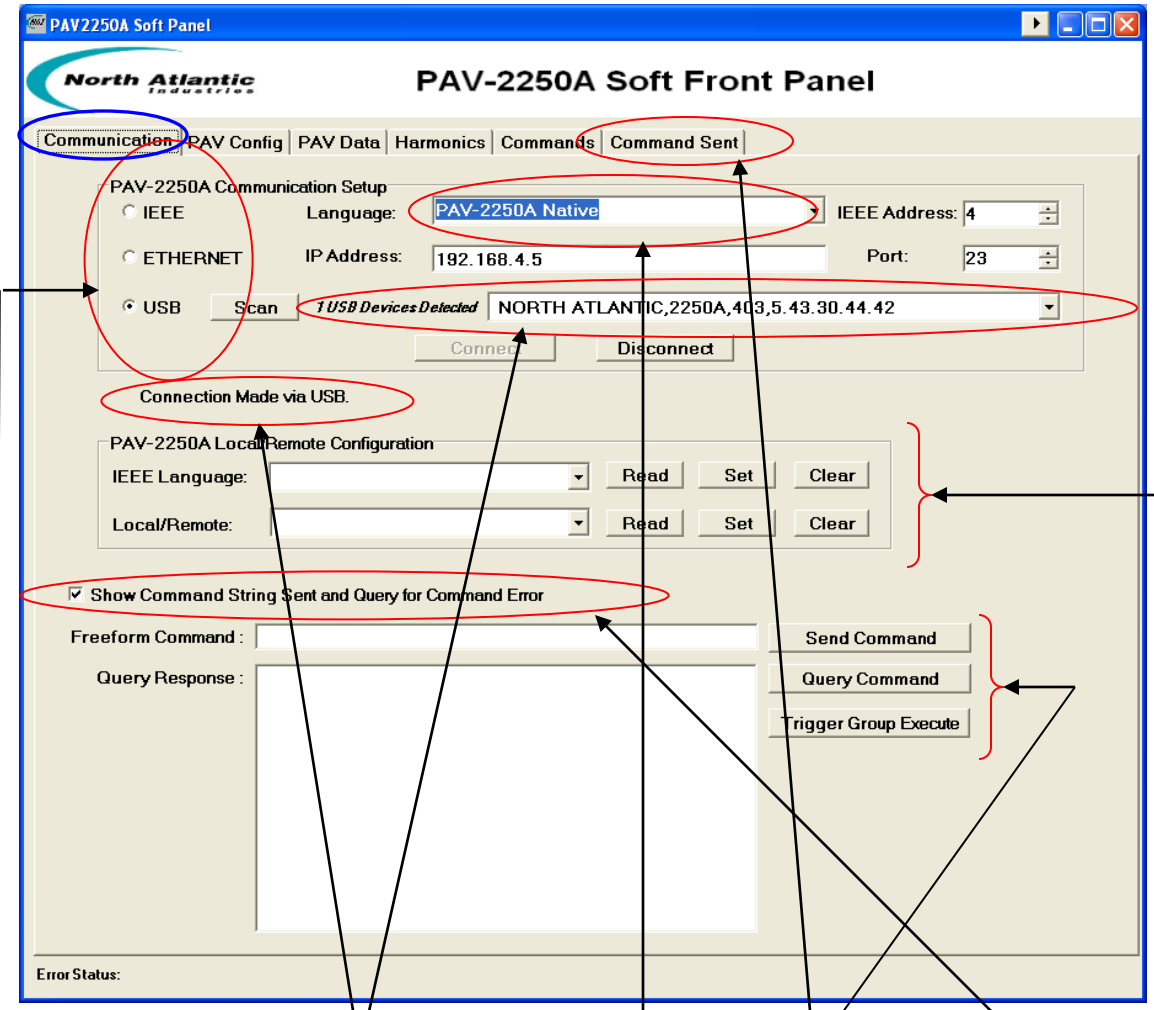
```
int ReadEthernetMsg(SOCKET s, int nMessageLenToBeRead, char *pszMessage, int *nMessageLen)
{
   int ret;
   int nLeft;
   int nIndex;
   int status = 0;

   nLeft = nMessageLenToBeRead;
   nIndex = 0;

   while (nLeft > 0)
   {
      ret = recv(s, pszMessage, nLeft, 0);
      // It seems we got some data
      if (ret != SOCKET_ERROR)
      {
         nLeft -= ret;
         nIndex += ret;
         pszMessage += ret;

         // We don't know the exact size of each message
         // for API we know that it won't exceed RECV_MSG_MAX_SIZE bytes
         nMessageLenToBeRead = nLeft;
         nLeft = 0;
      }
      // got SOCKET_ERROR
      else
      {
         status = ETHER_RECV_ERROR;
         break;
       }
    }

   if (nIndex > 0)
   {
      *nMessageLen = nIndex;
      status = ETHER_SUCCESS;  /* SUCCESS */
   }
   else
      status =  ETHER_RECV_ERROR; /* ERROR */
   return status;
}
```

6 PAV-2250A DLL

A dynamic link library (DLL) written in C, compiled under Microsoft Visual .NET 2003 has been included in the software package to provide a program interface that handles the language syntax to communicate with the unit. The function lists provided in this Dynamic-link library (DLL) is described in *Function Reference Manual for 2250A*.

7 PAV-2250A Soft Panel Program

A Soft Panel application written in C#, compiled under Microsoft Visual .NET 2003 that invokes the routines in the PAV-2250A Dll has been included in the software package. Note, the Microsoft .NET Framework 1.1 must be installed on your machine prior to running the Soft Panel application. The .NET Framework Version 1.1 Redistributable Package can be downloaded from the Microsoft Web site: http://www.microsoft.com/downloads



Choose the communication interface to the PAV-2250A.

Connection Status

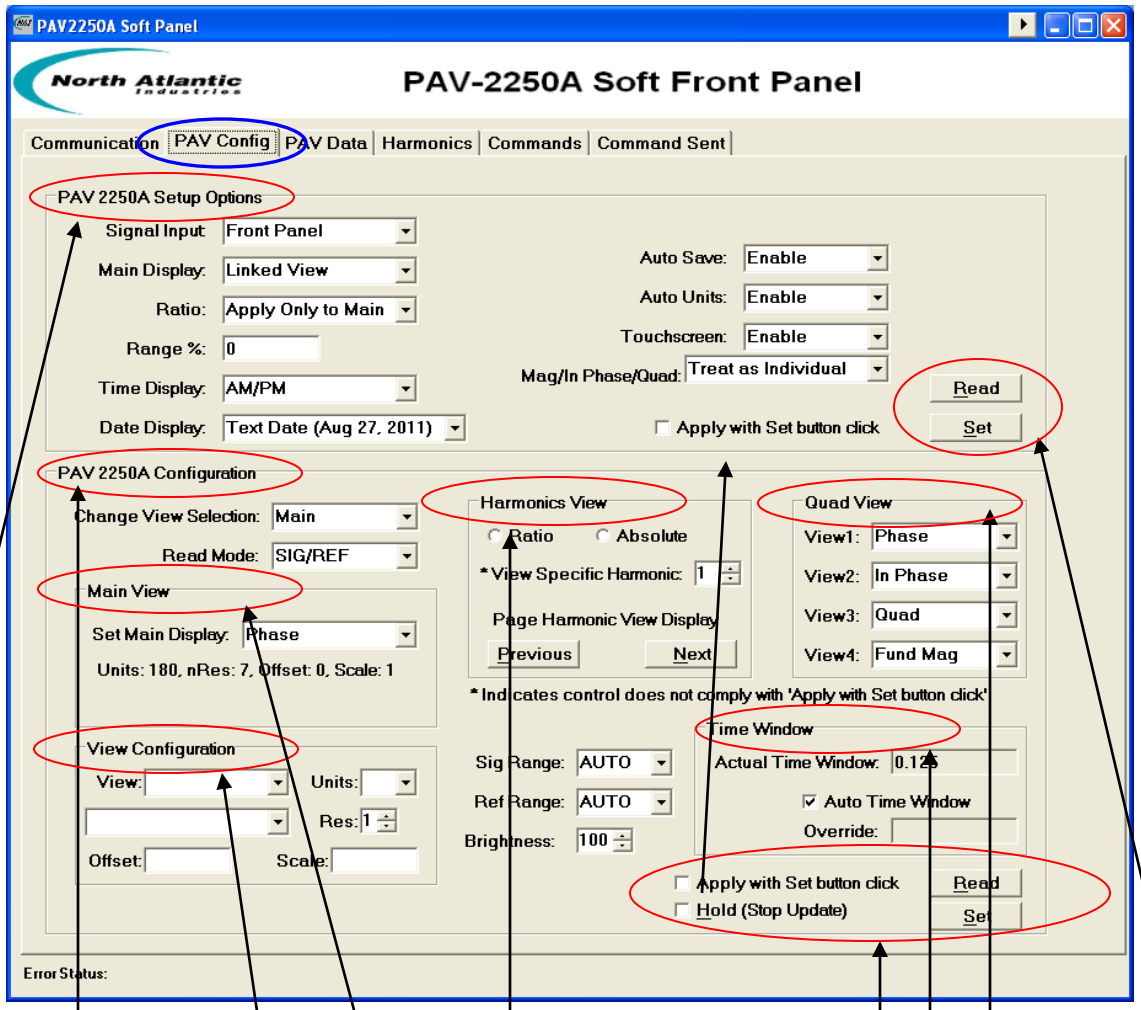
Choose the IEEE Language to communicate with PAV-2250A.

Option to show the command string and any errors in the tab labeled "Command Sent"

Writes/ Queries Freeform commands.

Gets/Sets the PAV-2250A Communication Setup. Note, communication to PAV-2250A must first be established.

Scan detects the number of Cypress USB Devices connected to your computer. "IDN" is used to determine which USB connections are for PAV-2250A devices.



Setup Options control various user preferences such as whether "Auto Units" is turned on and the format of the "Time Display" – AM/PM or Military.

Harmonics View section shows those items that are configurable on the Harmonics Tab. Viewing a specific Harmonic, displaying in Ratio/Absolute mode and paging through Harmonic groups are all controllable from this section.

Quad View section allows the user to configure what view to display for each of the 4 available view slots.

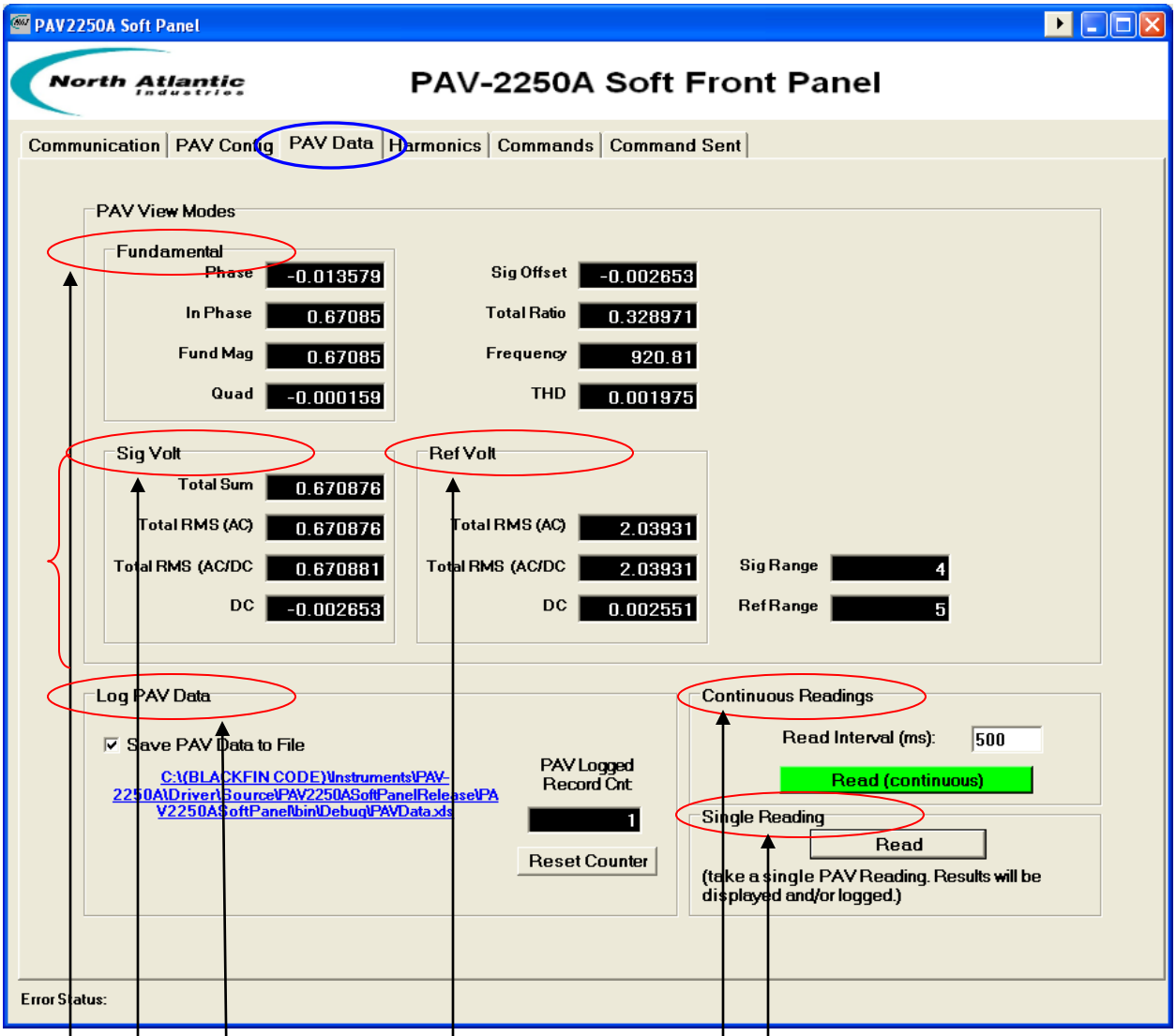
Configuration section allows the user to remotely control and configure all available view options - from specifying what to display on the main screen to configuring specific ranges for Signal and Reference and many other options.

Main View section allows for the configuration of what to display on the "Main View" along with the desired read mode (Sig/Ref, Sig/Sig, Ref/Ref, or Ref/Sig)

Force values to be "Read" from PAV or force PAV to reflect current selections ("Set"). NOTE: if "Apply with Set button click" is not checked PAV will automatically reflect selections as they are made.

View Configuration allows the user to control view aspects such as "units", "Max Field Width", "offset" and "scale"

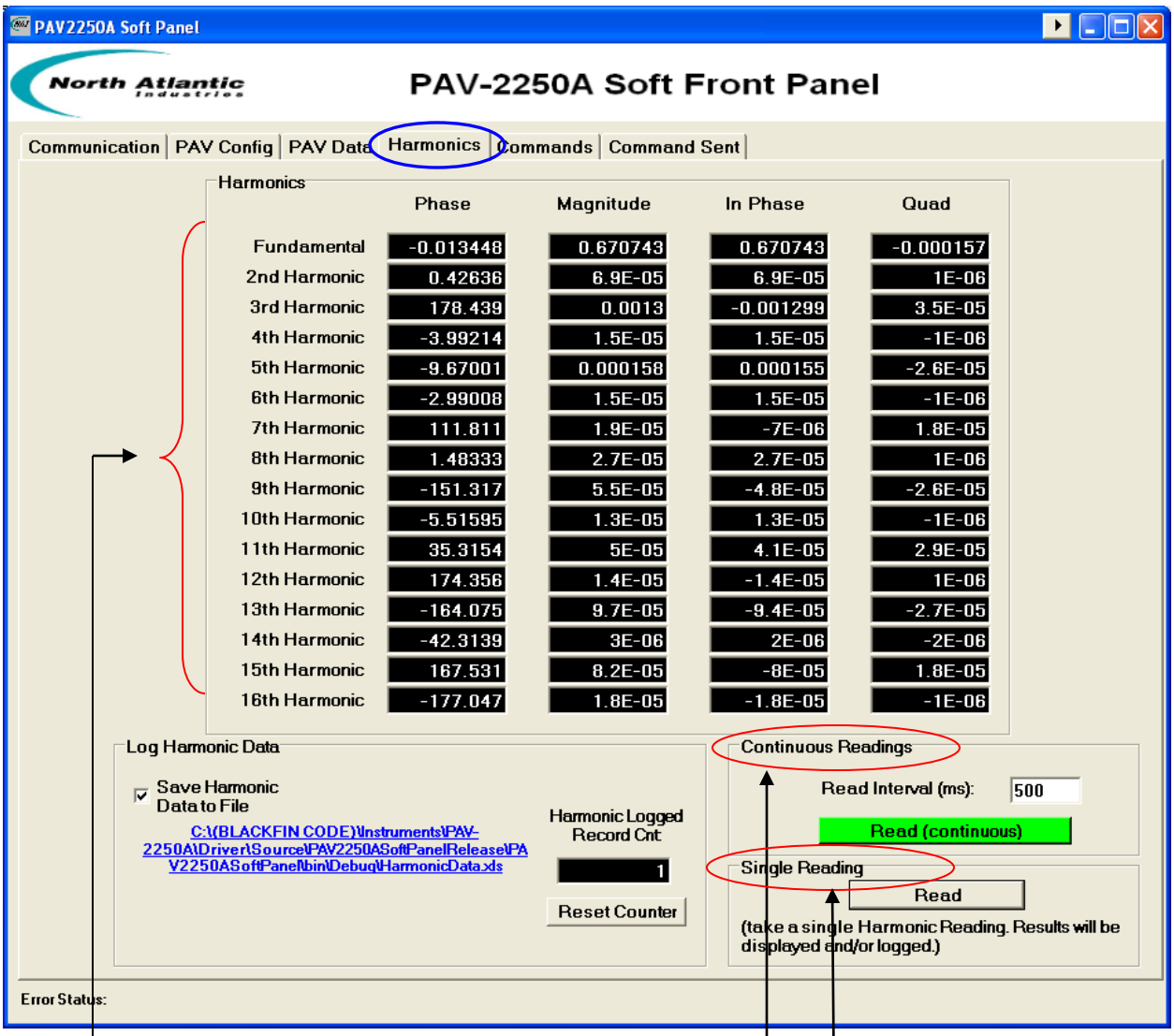
Time Window allows for the configuration of how often the data will be refreshed.



Retrieves Data for each of the view modes.

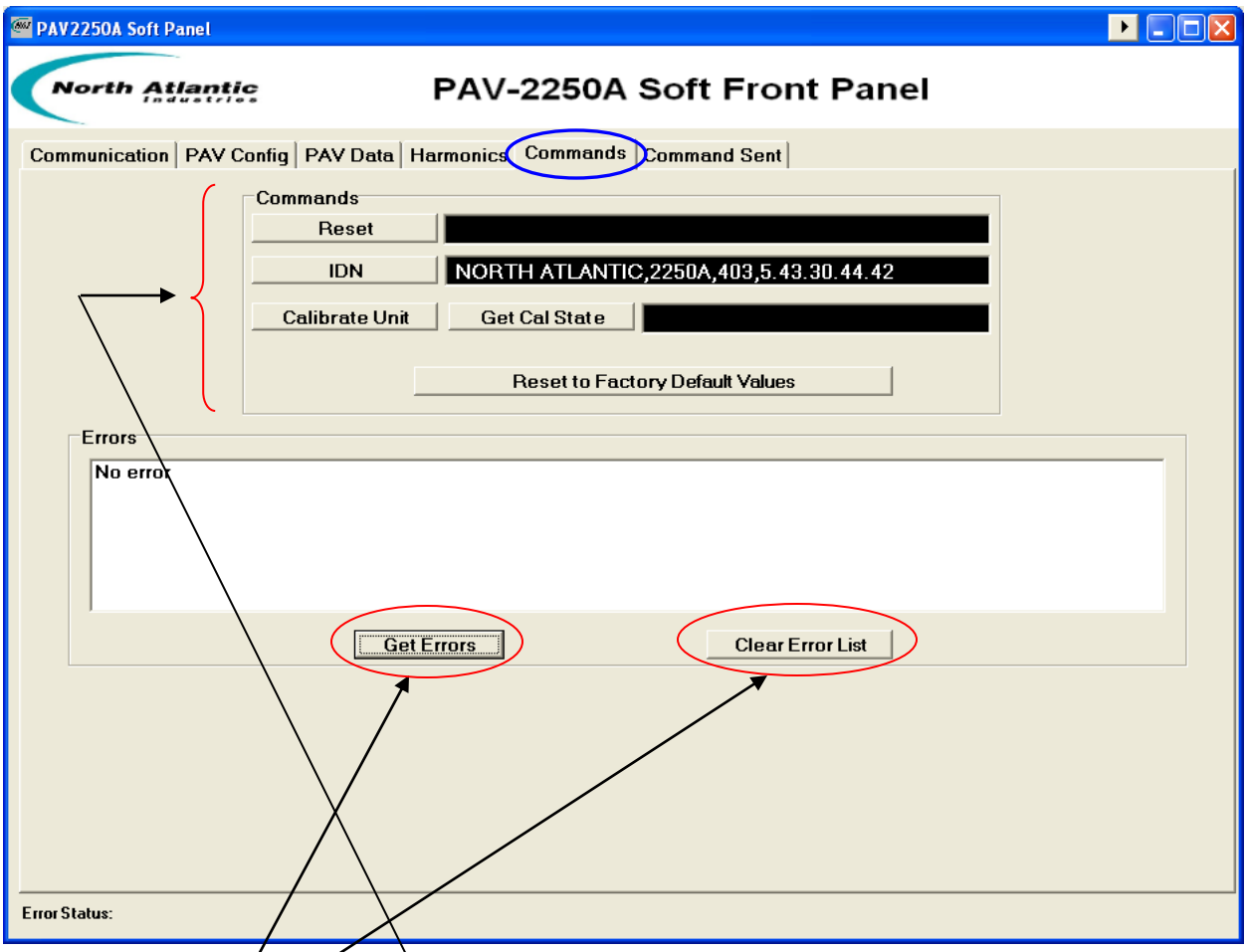
Log PAV Data saves the displayed values out to an Excel spreadsheet for further analysis.

Provides the user with the option to take continuous readings at a specified read interval or manually take a single reading at a time.



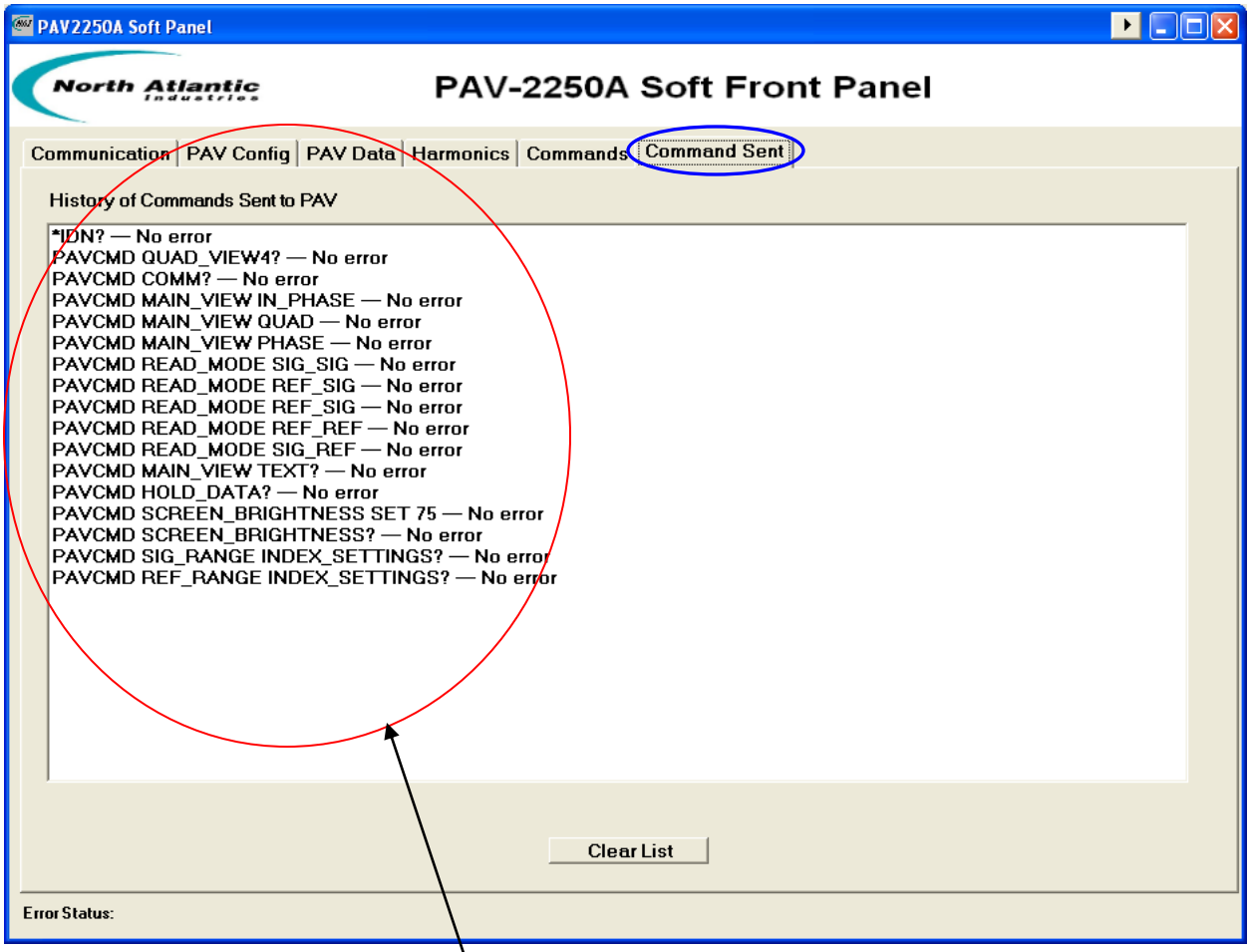
Log Harmonic Data saves the displayed harmonic values out to an Excel spreadsheet for further analysis.

Provides the user with the option to take continuous readings at a specified read interval or manually take a single reading at a time.



"No error" is returned when there is no error on the Error Queue.

"Reset", "Set Local Lockout", "Calibrate Unit" and "Reset to Factory Default Values" are allowed only when the PAV-2250A unit is set for Remote mode with specified communication interface (IEEE, USB, or Ethernet)



Commands sent to the PAV-2250A as well as the results from performing a call to the PAV-2250A Dll's PAV2250A_GetErrors() method to retrieve any messages from the Error Queue.

8 Cypress USB Driver Installation

In order to communicate with PAV2250A unit via the USB 2.0 interface, the Cypress USB Driver must be installed. Please refer to the document labeled “Cypress USB Installation” for detailed instruction on how to install the driver on your Windows PC.

Revision History

Revision ID	Revision Date	Description	Author
1.0.0.0	Mar 15, 2013	Initial Release	dt