

Programmer's Reference Guide for 5330A



Table of Contents

1 Introduction..... 3

2 Remote Setup..... 4

 2.1 Controlling Channel 1 Output Signal 5

3 Language Support 7

 3.1 Compatibility to 5330/5310 SRSs 9

 3.2 Language Independent Commands..... 10

 3.3 5330A Native..... 11

 3.4 5330 Native (Legacy) 24

 3.5 5310 Native (BCD) (Legacy)..... 29

 3.6 5310 Native (Binary) (Legacy)..... 31

4 5330A USB Protocol 33

5 5330A Ethernet Protocol..... 35

6 SRS5330A DLL..... 38

7 SRS-5330A Soft Panel Program..... 39

8 Cypress USB Driver Installation 46

1 Introduction

The 5330A Programmable Synchro/Resolver Simulator (SRS) Instrument provides two fully independent channels with 0.001° resolution and 0.003° (No Load)/ 0.004° (Full Load) accuracy. The 5330A allows all programming to be done via the touch-screen or mouse interface. In addition, remote operation capabilities are provided via IEEE-488, USB, Ethernet and J1 connection (50 pin DSUB connector in back of the unit).

Reference Documentation

For additional information about this instrument refer to the *Operation Manual for Model 5330A*. For additional information about the Application Programming Interface (API) provided in the SRS5330ADII refer to the *Function Reference Manual for 5330A*.

Reference CD

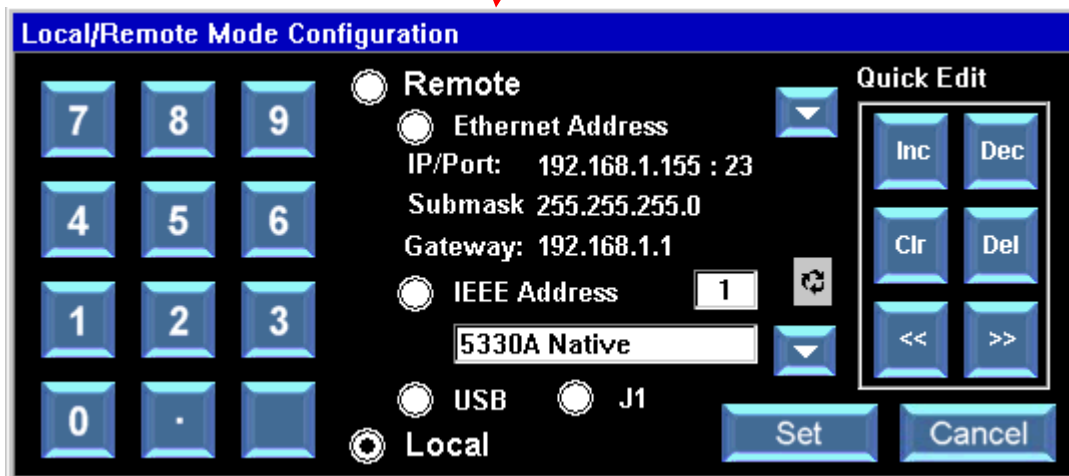
For electronic copies of the 5330A documentation, SRS-5330A Soft Panel application program, and source code for SRS5330ADII and Soft Panel application refer to the 5330A Product CD.

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:

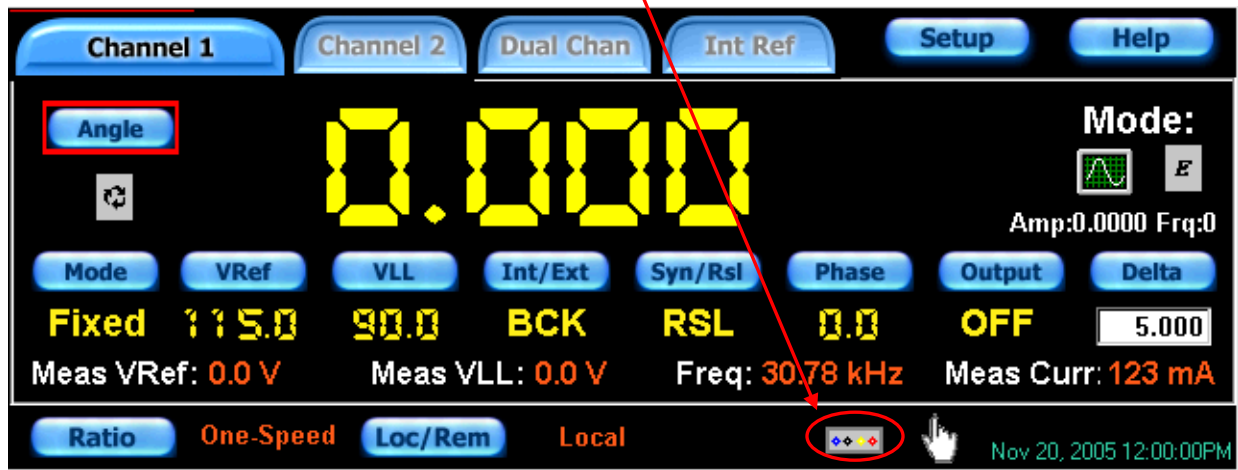



In “Local” mode, the configuration settings can be queried. The unit must be in one of the “Remote” modes (Ethernet, IEEE, USB or J1) before configuration settings can be changed remotely. Note, for remote programming via the IEEE interface, the language type must be selected, refer to section 3 on language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

2.1 Controlling Channel 1 Output Signal

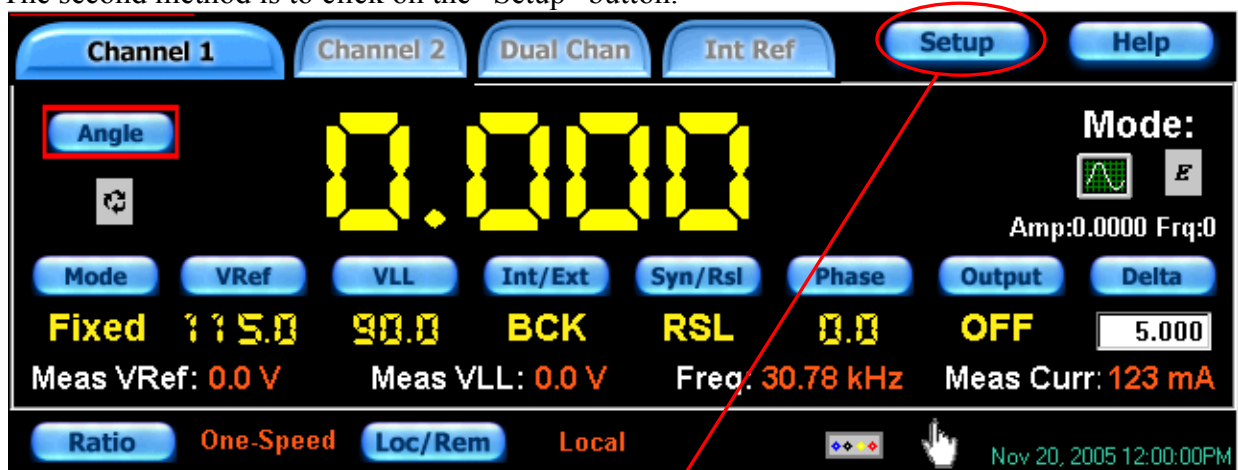
The channel 1 signal can be configured to output to the front panel connector or to the J1 connection in the back of the unit. The channel 2 signal outputs only to the J1 connection.

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



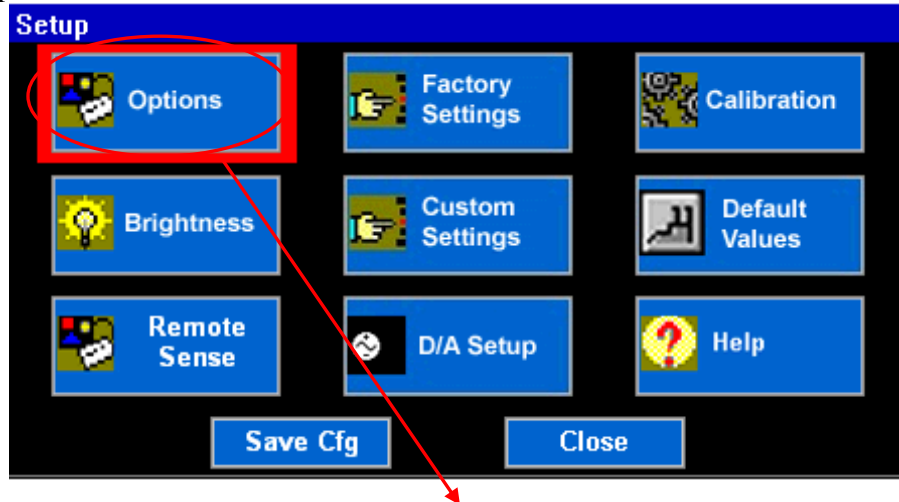
 Button configures Chan 1 Input to be read from the Front Connector or the Back (J1) Connector.

The second method is to click on the “Setup” button.

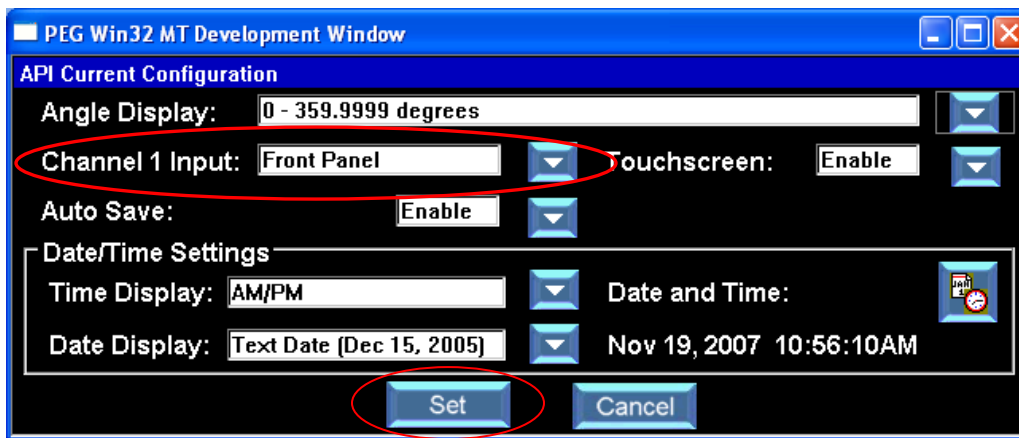


(See next page)

Click the “Options” button.



Select the “Front Panel” or “Back Connector” option for Channel 1 Output. Click on the “Set” button.



3 Language Support

The 5330A Unit is a direct replacement for all 5330's and 5310's. This unit supports the following languages:

	IEEE-488.1	USB	Ethernet
5330A Native	Supported	Supported	Supported
5330 Native (Legacy)	Supported	Not available	Not available
5310 Native (BCD) (Legacy)	Supported	Not available	Not available
5310 Native (Binary) (Legacy)	Supported	Not available	Not available

The following table lists the applicable IEEE-488 bus commands for the 5330A.

Mnemonic	ASCII	Hex	Function
GTL	SOH	01	Go To Local - This command instructs the SRS 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 SRS is addressed to listen, the SRS will initialize to the conditions listed under DCL.
DCL	DC4	14	Device Clear - When the SRS receives the DCL command it is initialized to the following state: SYNCHRO SRQ MODE - OFF GET MODE - OFF
GET	BS	08	Group Execute Trigger - When the GET command is received, and if the SRS is addressed to listen and has the GET mode switch on, data sent to the SRS 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 SRS is in remote or local operation.
SPE	CAN	18	Serial Poll Enable - After receipt of this command the SRS, when addressed to talk, will transmit the Status Byte.
SPD	EM	19	Serial Poll Disable - This command cancels the SPE command and allows the SRS, when it is addressed to talk, to send data.
UNL	/	3F	Unlisten - Unaddresses the SRS listen address.
UNT	-	5F	Untalk - Unaddresses the SRS talk address.

The following table lists the interface function capability codes for the 5330A.

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 5330/5310 SRSs

The 5330A will provide language compatibility to the following 5330/5310 systems:

- 5330 Native
- 5310 Native (BCD)
- 5310 Native (Binary)

Serial Poll (Ask Bryn)

When the IEEE language type selected is one of the 5330/5310 legacy languages, the status byte returned by the SRS indicates the status of the instrument. The bits of the status byte are defined as:

D7	D6	D5	D4	D3	D2	D1	D0
ERROR	RQS	0	0	0	0	0	RESOLVER

ERROR -When bit is set the SRS data is not stable.

RQS -When bit is set the SRS is asserting the SRQ line.

RESOLVER -When bit is set the SRS is programmed for RESOLVER mode. When cleared the SRS is set to SYNCHRO mode.

If the RQS bit is set, the remaining bits indicate the state of the SRS when the SRQ line was last asserted. If the RQS line is not set then the remaining bits indicate the state of the SRS at the time the status byte is read.

GET Mode When the G command is included in the programming string, the SRS will hold off applying the programming data until the GET (Group Executive Trigger) bus command is received. GET mode is cancelled once the bus command GET is received and must be reprogrammed if desired again.

3.2 Language Independent Commands

Note the following commands are case-sensitive.

Function	Syntax (commands must be sent with upper-case)	Comments
SRS 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 power-on or last saved configuration settings.
Language	SRSCMDLANG?<cr><lf>	Queries the IEEE Language setting. Query returns: ‘5330ANATIVE’, or ‘5330NATIVE’, or ‘5310_BCD’, ‘5310_BIN’
	SRSCMDLANG< 5330ANATIVE 5330NATIVE 5310_BCD 5310_BIN><cr><lf>	Sets the IEEE Language setting.

3.3 5330A Native

The SRS-5330A Native language is support via the IEEE-488.1, USB and Ethernet interfaces. The language provides remote programming access to the features available on the 5330A unit. Note the following commands are case-sensitive.

SRS CHANNELS		
Function	Syntax (commands must be sent with upper-case)	Comments
Angle	SRS<chan>ANGLE?<cr><lf>	Queries the angle setting for the channel. Query returns angle in degrees: Range: 0.0000 < value < 359.9999 or Bipolar Range: -180.0000 < value < +180.0000
	SRS<chan>ANGLE<value><cr><lf>	Sets the angle for the channel. Angle Data Range in degrees: Unipolar Range: 0.0000 < value < 359.9999 or Bipolar Range: -180.0000 < value < +180.0000
Reference Mode	SRS<chan>REFMODE?<cr><lf>	
	SRS<chan>REFMODE<FIXED RATIO><cr><lf>	
Reference Voltage	SRS<chan>REF_VOLT?<cr><lf>	Queries the reference voltage for the channel.
	SRS<chan>REF_VOLT<value><cr><lf>	
Line-to-Line Voltage	SRS<chan>LL_VOLT?<cr><lf>	Queries the line-to-line voltage for the channel.
	SRS<chan>LL_VOLT<value><cr><lf>	
Reference Source	SRS<chan>REF_SOURCE?<cr><lf>	Queries the reference source for the channel. Query returns: 'INT' or 'EXT'.

	SRS<chan>REF_SOURCE<INT EXT><cr><lf>	Sets the Internal/External reference source mode for the channel.
Signal Mode	SRS<chan>MODE?<cr><lf>	Queries the mode state of the channel. Query returns: 'RSL' or SYN'.
	SRS<chan>MODE<RSL SYN><cr><lf>	Sets the Rsl/Syn state for the channel.
Phase	SRS<chan>PHASE?<cr><lf>	
	SRS<chan>PHASE<value><cr><lf>	
Output Mode	SRS<chan>OUTMODE?<cr><lf>	
	SRS<chan>OUTMODE<ON OFF><cr><lf>	
Reference Voltage Wrap	SRS<chan>REF_VOLT_WRAP?<cr><lf>	
Line-to-Line Voltage Wrap	SRS<chan>LL_VOLT_WRAP?<cr><lf>	
Reference Frequency Wrap	SRS<chan>REF_FREQ?<cr><lf>	Queries the reference frequency for the channel.
Ratio	SRS<chan>RATIO?<cr><lf>	Queries the ratio setting for each channel. Query returns the '1' always for channel 1 and for channel 2, the ratio setting value: Ratio Range = 1 to 255.
	SRS<chan>RATIO<value><cr><lf>	Sets the ratio setting for each channel. Channel 1 can only be set to 1. Channel 2 can be set to any value between 1 and 255.

SRS DYNAMIC MOTION FUNCTIONS		
Dynamic Motion Mode	SRS<chan>DYNAMIC_MODE?<cr><lf>	
	SRS<chan>DYNAMIC_MODE<STATIC MOD_SINE MOD_RAMP MOD_SQUARE MOD_TRIANGLE ROT_CONTINUOUS ROT_STOPSTART><cr><lf>	
Dynamic Motion Amp	SRS<chan>MOD_AMP?<cr><lf>	
	SRS<chan>MOD_AMP<value><cr><lf>	
Dynamic Motion Freq	SRS<chan>MOD_FREQ?<cr><lf>	
	SRS<chan>MOD_FREQ<value><cr><lf>	
Dynamic Motion Rotation Rate	SRS<chan>ROT_RATE?<cr><lf>	Queries the angle velocity data for the channel. Query returns channel velocity in degrees per second: Range: -32767 < value < +32767.
	SRS<chan>ROT_RATE<value><cr><lf>	
Dynamic Motion Rotation Stop Angle	SRS<chan>STOP_ANG?<cr><lf>	
	SRS<chan>STOP_ANG<value><cr><lf>	
Dynamic Motion Start/Stop	SRS<chan><DYNAMIC_CHANGE?<cr><lf>	
	SRS<chan><DYNAMIC_START DYNAMIC_STOP><cr><lf>	
External Reference Source	SRS<chan>EXT_REF_SOURCE?<cr><lf>	<chan> can only equal "1". Channel 2 is not configurable, it only can go to the back.
	SRS<chan>EXT_REF_SOURCEFRONT BACK<cr><lf>	<chan> can only equal "1". Channel 2 is not configurable, it only can go to the back.
Overcurrent Status/Reset	SRS<chan>OVER_CUR?<cr><lf>	Query if Overcurrent condition has occurred. Returns "OVER_CUR" if an Overcurrent has occurred, otherwise it

		returns "NO OVER_CUR".
	SRS<chan>OVER_CURRESET<cr><lf>	Will return channel <chan> to normal status if it was in Overcurrent status.
Output Remote Sense	SRS<chan>REM_SENSE?<cr><lf>	Query if remote sense is enabled.
	SRS<chan>REM_SENSEON OFF<cr><lf>	Enables ("ON") or disables ("OFF") the sense lines on the channel output.
J1 BCD/ Binary Mode	SRSCMDJ1_MODE?<cr><lf>	Query if J1 is in BCD or Binary mode. Returns "BCD" or "BINARY"
	SRSCMDJ1_MODEBCD BINARY<cr><lf>	Sets J1 to BCD or Binary mode.

SRS D/A FUNCTIONS		
Digital-to-Analog Setup	SRS<chan>DAOUTPUT?<cr><lf>	Queries the D/A setup for the data to use for D/A voltage output conversion for the channel. Query returns 'ANG' or 'VEL'.
	SRS<chan>DAOUTPUT<ANG VEL><cr><lf>	Sets the D/A data to use either angle or velocity for D/A voltage output conversion for the channel.
	SRS<chan>DAHIDATA?<cr><lf>	Queries the D/A Upper Limit Data conversion for the channel. Query returns for Angle Data in degrees: Unipolar Range: 0.0000 < value < 359.9999 or Bipolar Range: -180.0000 < value < +180.0000 Query returns for Velocity Data in degrees/sec: Range: -10000<value<10000

	SRS<chan>DAHIDATA<value><cr><lf>	Sets the D/A Upper Limit Data conversion for the channel. Angle Data Range in degrees: Unipolar Range: 0.0000 < value < 359.9999 or Bipolar Range: -180.0000 < value < +180.0000 Velocity Data Range in degrees/sec: -10000<value<10000
	SRS<chan>DAHIVOLT?<cr><lf>	Queries the D/A Voltage conversion for the Upper Limit Conversion for the channel. Query returns voltage value in the range: -10.0<value<10.0
	SRS<chan>DAHIVOLT<value><cr><lf>	Sets the D/A Voltage conversion for the Upper Limit Conversion for the channel. Voltage range in volts: -10.0<value<10.0

	SRS<chan>DALODATA?<cr><lf>	<p>Queries the D/A Lower Limit Data conversion for the channel.</p> <p>Query returns for Angle Data in degrees: Unipolar Range: 0.0000 < value < 359.9999 or Bipolar Range: -180.0000 < value < +180.0000</p> <p>Query returns for Velocity Data in degrees/sec: Range: -10000<value<10000</p>
	SRS<chan>DALODATA<value><cr><lf>	<p>Sets the D/A Lower Limit Data conversion for the channel.</p> <p>Angle Data Range in degrees: Unipolar Range: 0.0000 < value < 359.9999 or Bipolar Range: -180.0000 < value < +180.0000</p> <p>Velocity Data Range in degrees/sec: -10000<value<10000</p>
	SRS<chan>DALOVOLT?<cr><lf>	<p>Queries the D/A Voltage conversion for the Lower Limit Conversion for the channel.</p> <p>Query returns voltage value in the range: -10.0<value<10.0</p>

	SRS<chan>DALOVOLT<value><cr><lf>	Sets the D/A Voltage conversion for the Lower Limit Conversion for the channel. Voltage range in volts: -10.0<value<10.0
--	---	--

INTERNAL REFERENCE GENERATOR		
Function	Syntax (commands must be sent with upper-case)	Comments
Reference Generator Frequency	REF_GENFREQ?<cr><lf>	Queries the frequency setting for the internal reference generator.
	REF_GENFREQ<value><cr><lf>	Sets the frequency setting for the internal reference generator. Frequency range is 47.0 to 20000.0 Hz.
Reference Generator Voltage	REF_GENVOLT?<cr><lf>	Queries the voltage setting for the internal reference generator.
	REF_GENVOLT<value><cr><lf>	Sets the voltage setting for the internal reference generator. Voltage range is 2.0 to 115.0 volts.
Reference Generator Output State	REF_GENSTATE?<cr><lf>	Queries the output state of the internal reference generator. Query returns: 'OPEN', or 'CLOSED'.
	REF_GENSTATE<OPEN CLOSE><cr><lf>	Sets the output state for the internal reference generator. The CLOSE state will out allow reference signals to be available at the output connectors. The OPEN state will prevent the reference signals from being outputted.

Reference Generator Sense Direction	REF_GENSENSE_DIR?<cr><lf>	Queries if the remote sense function for the Reference Generator is using the front connector or rear connector.
	REF_GENSENSE_DIRFRONT BACK<cr><lf>	Sets the remote sense function for the Reference Generator is using the front connector or rear connector.
Reference Generator Sense Enabled	REF_GENREM_SENSE?<cr><lf>	If remote sense is enabled.
	REF_GENREM_SENSEON OFF<cr><lf>	Enables ("ON") or disables ("OFF") the sense lines on the Reference Generator output.
Reference Generator Overcurrent Status/Reset	REF_GENOVER_CUR?<cr><lf>	Query if Overcurrent condition has occurred. Returns "OVER_CUR" if an Overcurrent has occurred, otherwise it returns "NO OVER_CUR".
	REF_GENOVER_CURRESET<cr><lf>	Will return Reference Generator to normal status if it was in Overcurrent status.

SRS CONFIGURATION		
Function	Syntax (commands must be sent with upper-case)	Comments
Communi- cation Setting	SRSCMD COMM?<cr><lf>	Queries the communication settings. Possible results are: Local Mode or Remote IEEE Addr: SRS-IEEE Language or Remote USB or Remote Ethernet or Remote J1 or Remote with Lockout via IEEE Addr: SRS-IEEE Language or Remote with Lockout via USB or Remote with Lockout via Ethernet or Remote with Lockout via J1
	SRSCMDCOMM<IEEE USB ETHERNET J1><cr><lf>	Sets the communication setting to communicate remotely via IEEE, USB, Ethernet or J1.
Go To Local	SRSCMDCOMM<LOCAL><cr><lf>	Sets the device to Local mode.
Local Lockout	SRSCMDCOMM<LOCKOUT><cr><lf>	Sets the device to Local Lockout mode.
Angle Format	SRSCMDANG_FMT?<cr><lf>	Queries the angle display format. Query returns: (‘0 to 360’, or ‘-180 to 180’, or ‘Deg, Min, Sec’)

	SRSCMDANG_FMT<360 180 MIN><cr><lf>	Sets the angle display format.
Channel 1 Output Connector	SRSCMDCH1OUTPUT?<cr><lf>	Queries the channel 1 output connector setting. Query returns: 'FRONT', or 'BACK'.
	SRSCMDCH1OUTPUT<FRONT BACK><cr><lf> >	Sets the channel 1 output connector setting to either the Front or Back connectors.
Default Values	SRSCMDRSTFRAM<cr><lf>	Sets the device to the factory default conditions.

Calibration	SRSCMDCALIBRATE?<cr><lf>	Queries the device for the calibration state. Query returns: 'CAL DONE' or 'CALIBRATING'
	SRSCMDCALIBRATE<cr><lf>	Calibrates the unit.
Background Calibration	SRSCMDBACKGND_CAL?<cr><lf>	Queries the device to determine if the background calibration is 'CHAN1 CALIBRATING' or 'CHAN2 CALIBRATING' or 'CHAN1 & CHAN2 CALIBRATING' or 'NOT CALIBRATING'
	SRSCMDBACKGND_CAL <ON OFF><cr><lf>	Enables or disables the background calibration.

3.4 5330 Native (Legacy)

The SRS-5330 Native language is only support via the IEEE-488.1. The language is available to provide backwards compatibility to the 5330 units. Only the features that were available for the 5330 are supported with this language.

5330 units have only one channel. All 5330 backward compatible commands can only control Channel 1 of the 5330A.

Function	Syntax	Comments
Select Synchro Mode	FNCSYN:CH0<cr><lf> or FNCSYN:CH00<cr><lf>	Configures Channel 1 for Synchro Mode.
Select Resolver Mode	FNCRSL:CH0<cr><lf> or FNCRSL:CH00<cr><lf>	Configures Channel 1 for Resolver Mode.
Set Angle	FNCSYN:CH0 SETANGL<value><cr><lf> or FNCSYN:CH00 SETANGL<value><cr><lf> or FNCRSL:CH0 SETANGL<value><cr><lf> or FNCRSL:CH00 SETANGL<value><cr><lf>	Setting Synchro/Resolver is embedded into the Set Angle command. This mode has to be set every time angle is set. It can be set to the same value every time. Limits for angle: 0 <= <value> <= 360
Set Angle Rate	FNCSYN:CH0 SETANRT<value><cr><lf> or FNCSYN:CH00 SETANRT<value><cr><lf> or FNCRSL:CH0 SETANRT<value><cr><lf> or FNCRSL:CH00 SETANRT<value><cr><lf>	Setting Synchro/Resolver is embedded into the Set Angle Rate command. This mode has to be set every time angle is set. It can be set to the same value every time. Limits for angle rate: -12,600 <= <value> <= 12,600

Set Line to Line Voltage	<pre>FNCSYN:CH0 SETVOLT<value><cr><lf> or FNCSYN:CH00 SETVOLT<value><cr><lf> or FNCRSL:CH0 SETVOLT<value><cr><lf> or FNCRSL:CH00 SETVOLT<value><cr><lf></pre>	<p>Original 5330 documentation referred to this as "Voltage".</p> <p>Setting Synchro/Resolver is embedded into the Set Line to Line Voltage command. This mode has to be set every time VLL is set. It can be set to the same value every time.</p>
Set Reference Voltage	<pre>FNCSYN:CH0 SETREFV<value><cr><lf> or FNCSYN:CH00 SETREFV<value><cr><lf> or FNCRSL:CH0 SETREFV<value><cr><lf> or FNCRSL:CH00 SETREFV<value><cr><lf></pre>	<p>Just like 5330, the 5330A accepts this command without error but will have no effect on setup of the unit except for changes in Synchro/Resolver mode.</p> <p>Setting Synchro/Resolver is embedded into the Set Reference Voltage command. This mode has to be set every time Vref is set. It can be set to the same value every time.</p>
Set Frequency	<pre>FNCSYN:CH0 SETFREQ<value><cr><lf> or FNCSYN:CH00 SETFREQ<value><cr><lf> or FNCRSL:CH0 SETFREQ<value><cr><lf> or FNCRSL:CH00 SETFREQ<value><cr><lf></pre>	<p>Just like 5330, the 5330A accepts this command without error but will have no effect on setup of the unit except for changes in Synchro/Resolver mode.</p> <p>Setting Synchro/Resolver is embedded into the Set Frequency command. This mode has to be set every time Frequency is set. It can be set to the same value every time.</p>

Set Calibration	<pre>FNCSYN:CH0 SETCALA<value><cr><lf> or FNCSYN:CH00 SETCALA<value><cr><lf> or FNCRSL:CH0 SETCALA<value><cr><lf> or FNCRSL:CH00 SETCALA<value><cr><lf></pre>	<p>The 5330A calibration method is not backward compatible with the 5330. The 5330A accepts this command without error but will have no effect on setup of the unit except for changes in Synchro/Resolver mode.</p> <p>Setting Synchro/Resolver is embedded into the Set Calibration command. This mode has to be set every time Calibration is set. It can be set to the same value every time.</p>
Set Clear Calibration	<pre>FNCSYN:CH0 SETCALZ<value><cr><lf> or FNCSYN:CH00 SETCALZ<value><cr><lf> or FNCRSL:CH0 SETCALZ<value><cr><lf> or FNCRSL:CH00 SETCALZ<value><cr><lf></pre>	<p>The 5330A calibration method is not backward compatible with the 5330. The 5330A accepts this command without error but will have no effect on setup of the unit except for changes in Synchro/Resolver mode.</p> <p>Setting Synchro/Resolver is embedded into the Clear Calibration command. This mode has to be set every time the calibration is cleared. It can be set to the same value every time.</p>
Set Angle Step Size	<pre>FNCSYN:CH0 SETANGP<value><cr><lf> or FNCSYN:CH00 SETANGP<value><cr><lf> or FNCRSL:CH0 SETANGP<value><cr><lf> or FNCRSL:CH00 SETANGP<value><cr><lf></pre>	<p>The step size is used by the Increment CCW and Increment CW commands.</p> <p>Setting Synchro/Resolver is embedded into the Set Angle Step Size command. This mode has to be set every time angle is set. It can be set to the same value every time.</p> <p>Limits for angle step size: 0.1 <= <value> <= 359.999</p>
Increment Counter Clockwise	<pre>FNCSYN:CH0 SETINCC<cr><lf> or FNCSYN:CH00 SETINCC<cr><lf> or FNCRSL:CH0</pre>	<p>Setting Synchro/Resolver is embedded into the Increment Counter Clockwise command. This mode has to be set every time this command is sent. It can</p>

	SETINCC<cr><lf> or FNCRSL:CH00 SETINCC<cr><lf>	be set to the same value every time.
Increment Clockwise	FNCSYN:CH0 SETINCW<cr><lf> or FNCSYN:CH00 SETINCW<cr><lf> or FNCRSL:CH0 SETINCW<cr><lf> or FNCRSL:CH00 SETINCW<cr><lf>	Setting Synchro/Resolver is embedded into the Increment Clockwise command. This mode has to be set every time angle is set. It can be set to the same value every time.
Close Isolation Relays	CLS:CH0<cr><lf> or CLS:CH00<cr><lf>	Just like 5330, the 5330A accepts this command without error but will have no effect on setup of the unit.
Open Isolation Relays	OPN:CH0<cr><lf> or OPN:CH00<cr><lf>	Just like 5330, the 5330A accepts this command without error but will have no effect on setup of the unit.
Reset Command	RSTSYN:CH0<cr><lf> or RSTSYN:CH00<cr><lf> or RSTRSL:CH0<cr><lf> or RSTRSL:CH00<cr><lf>	The noun "SYN" or "RSL" must be present, but has no effect on setup of the unit. The unit is reset to: Mode: Synchro L-L Voltage: 11.8 Angle: 0.000 Angle Rate: 0
Command Status	STA<cr><lf>	After sending the Command Status, the unit will reply with an error (see 5330 Legacy Error messages) or "<cr><lf>"
Internal Self Test	IST<cr><lf>	The 5330A accepts this command without error but will have no effect on setup of the unit.
Confidence Test	CNF<cr><lf>	The 5330A accepts this command without error but will have no effect on setup of the unit.

5330 Legacy Error Messages:

FO7SRS00 (MOD): INVALID DATA FIELD
FO7SRS00 (MOD): MESSAGE OVERFLOW
FO7SRS00 (MOD): SYNTAX ERROR
FO7SRS00 (MOD): INVALID SEQUENCE
FO7SRS00 (MOD): DATA RANGE ERROR
FO7SRS00 (MOD):INVALID INX-FTH SEQUENCE
FO7SRS00 (MOD): MODIFIER MISMATCH
FO7SRS00 (MOD): NO SPACE
FO7SRS00 (MOD): R/D BIT NOT READY
FO7SRS00 (MOD): BIT FAIL
FO7SRS00 (MOD): ERROR IN EXPONENT RANGE
FO7SRS00 (MOD): INDICATES LOSS OF INPUT SIGNAL
FO7SRS00 (MOD): NO CARRIAGE RETURN
FO7SRS00 (MOD): INVALID MESSAGE COMMAND
FO7SRS00 (MOD): CHANNEL NUMBER ERROR
FO7SRS00 (MOD): SET CODE ERROR
FO7SRS00 (MOD): MODIFIER ERROR
FO7SRS00 (MOD): NOUN ERROR

3.5 5310 Native (BCD) (Legacy)

The SRS-5310 BCD language is only support via the IEEE-488.1. The language is available to provide backwards compatibility to the 5310 units with the Input Data Switch set to the BCD position. Only the features that were available for the 5310 are supported with this language.

5310 units have only one channel. All 5330 backward compatible commands can only control Channel 1 of the 5330A.

Function	Syntax	Comments
Combination Command	<angle><mode><vref> <vll><cr><lf>	One command sets Angle, Syn/Res mode, Reference Voltage, and Line to Line Voltage. <angle> Is always six digits wide. Least significant digit is the hundredths place. Examples: 270°="27000", 90°="09000", 123.45°="12345". <mode> is always one digit wide. 0 = Resolver 1 = Synchro <vref> is always one digit wide. 0 = 115v 1 = 26v <vll> is always one digit wide 1 = 11.8v 2 = 26v 3 = 90v Example of full command: "27000111" sets 270 degrees, Synchro, Vref =26v, Vll = 11.8v
Set 100s-Place of Angle	I<value><cr><lf>	Set the hundredths place of output angle. Limits: 0 <= <value> <= 3
Set 10s-Place of Angle	N<value><cr><lf>	Set the tens place of output angle. Limits: 0 <= <value> <= 9
Set 1s-Place of Angle	P<value><cr><lf>	Set the ones place of output angle. Limits: 0 <= <value> <= 9
Set 0.1s-Place of Angle	U<value><cr><lf>	Set the tenths place of output angle. Limits: 0 <= <value> <= 9

Set .01s-Place of Angle	T<value><cr><lf>	Set the hundredths place of output angle. Limits: 0 <= <value> <= 9
Set Tracking Rate	S<sign><value><cr><lf>	Set the tracking rate angle <sign> can equal "+" or "-". Can be omitted if the tracking rate is positive. <value> Is in the format XXX.X. This is the value of rate in deg/sec.
Set End Angle	E<value><cr><lf>	Set end angle delimiter. <value> Is always six digits wide. Least significant digit is the hundredths place. Examples: 270°="27000", 90°="09000", 123.45°="12345".

3.6 5310 Native (Binary) (Legacy)

The SRS-5310 Binary language is only support via the IEEE-488.1. The language is available to provide backwards compatibility to the 5310 units with the Input Data Switch set to the BIN position. Only the features that were available for the 5310 are supported with this language.

5310 units have only one channel. All 5330 backward compatible commands can only control Channel 1 of the 5330A.

Function	Syntax	Comments
Combination Command	<angle><mode><vref> <vll><cr><lf>	<p>One command sets Angle, Syn/Res mode, Reference Voltage, and Line to Line Voltage.</p> <p><angle> Is always six digits wide. It is a hex value in the range 000000 to 03FFFF. Examples: 0°="00000", 270°="30000", 90°="10000", 22.5°="04000".</p> <p><mode> is always one digit wide. 0 = Resolver 1 = Synchro</p> <p><vref> is always one digit wide. 0 = 115v 1 = 26v</p> <p><vll> is always one digit wide 1 = 11.8v 2 = 26v 3 = 90v</p> <p>Example of full command: "30000111" sets 270 degrees, Synchro, Vref =26v, Vll = 11.8v</p>
Set Bits 1-2 of Angle	I<value><cr><lf>	<p>Set bits 1 and 2 (most significant bits) of the angle.</p> <p>Limits: 0 <= <value> <= 3</p>
Set Bits 3-6 of Angle	N<value><cr><lf>	<p>Set bits 3 to 6 of output angle.</p> <p>Limits: 0 <= <value> <= F</p>
Set Bits 7-10 of Angle	P<value><cr><lf>	<p>Set bits 7 to 10 of output angle.</p> <p>Limits: 0 <= <value> <= F</p>

Set Bits 11-14 of Angle	U<value><cr><lf>	Set bits 11 to 14 of output angle. Limits: 0 <= <value> <= F
Set Bits 15-18 of Angle	T<value><cr><lf>	Set bits 15 to 18 of output angle. Limits: 0 <= <value> <= F
Set Tracking Rate	S<sign><value><cr><lf>	Set the tracking rate angle <sign> can equal "+" or "-". Can be omitted if the tracking rate is positive. <value> Is in the format XXX.X. This is the value of rate in deg/sec. This field is always in bcd.
Set End Angle	E<value><cr><lf>	Set end angle delimiter. <value> Is always six digits wide. Least significant digit is the hundredths place. Examples: 270°="27000", 90°="09000", 123.45°="12345". This field is always in bcd.

4 5330A USB Protocol

The 5330A USB interface supports only the SRS-5330A Native Language. Sending commands via the USB interface require the following protocol:

Number of Bytes to be sent (2 bytes)	Command ID (5330 (i.e. 0x14D2)) (2 bytes)	Data
---	--	------

The following is code snippets from the SRS5330AD11 (USBComm.cpp) that makes calls to the Cypress CyAPI.lib file to sending commands to the 5330A:

```
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 = 0x14D2; // 5330 (0x14D2) 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 (0x14D2) for 5330
    // 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 (note, max returned from Cypress USB is 64 bytes
            BytesToRead = 64;
            unsigned char  aReceiveBuffer[64];

            for (int i = 0; i < 64; 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 < 2));
}
else
{
    ReinitUSB();
}
}

return bSuccess;
}
```

5 5330A Ethernet Protocol

The 5330A Ethernet interface supports only the SRS-5330A 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 `SRS5330ADll (Ethernet.cpp)` makes calls to the Winsock API to create and connect a TCP/IP socket to send commands to the 5330A. Note the code utilizes the `PingHost()` call to make sure that the IP address specified for the 5330A 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
    WSStartup(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 5330A, device log-in is required. 5330A Ethernet login is accomplished by sending “`NAII\r\n`” command via the Ethernet connection to the 5330A.

The following code snippets, `Ethernet_WriteMsg()`, `SendEthernetMsg()` and `ReadEthernetMsg()` from the SRS5330ADII (Ethernet.cpp) makes calls to the Winsock API to send and receive messages to and from the 5330A.

```
#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;
}
```

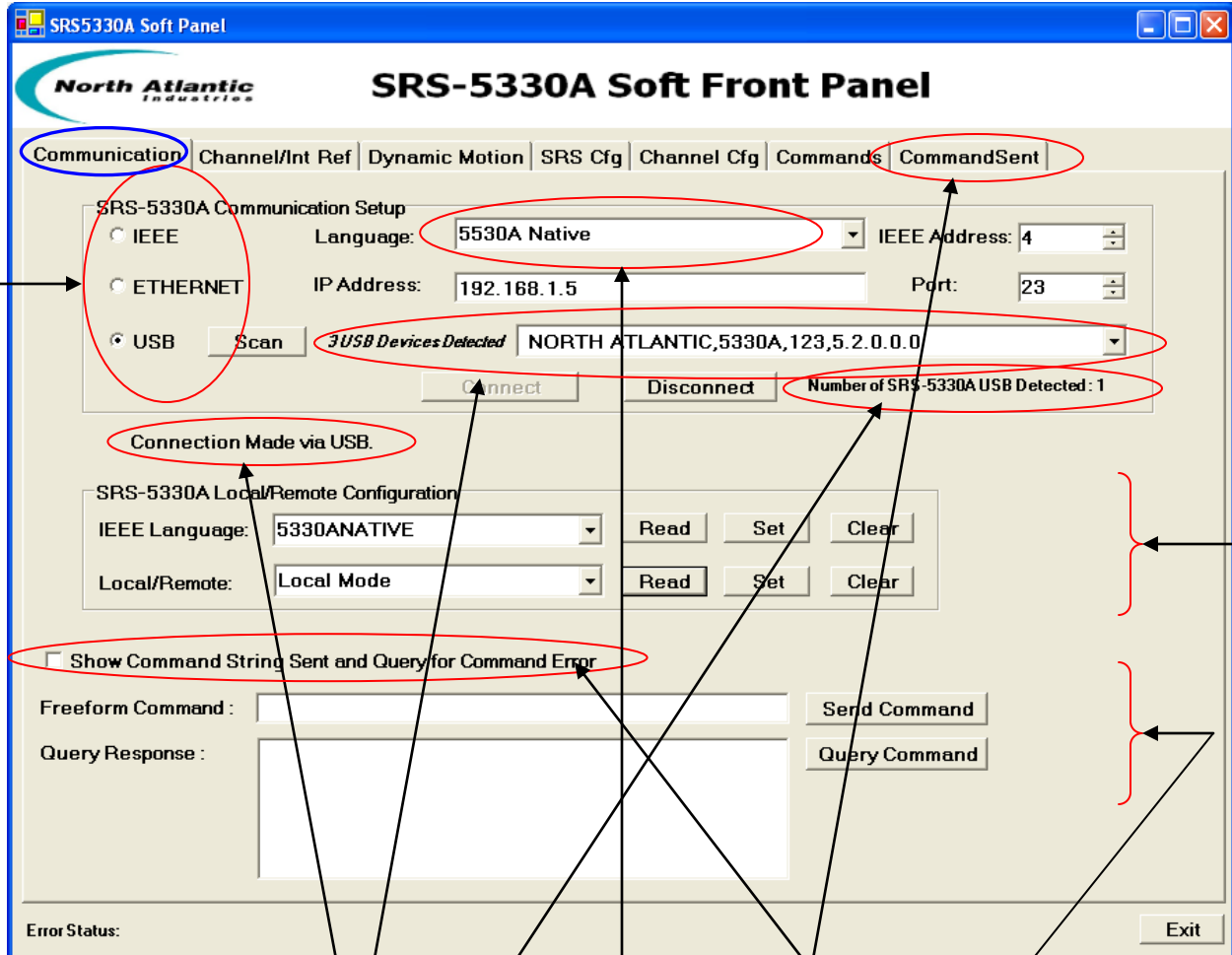
6 SRS5330A DLL

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 5330A*.

7 SRS-5330A Soft Panel Program

A Soft Panel application written in C#, compiled under Microsoft Visual .NET 2003 that invokes the routines in the SRS-5330A 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 SRS-5330A.

Connection Status

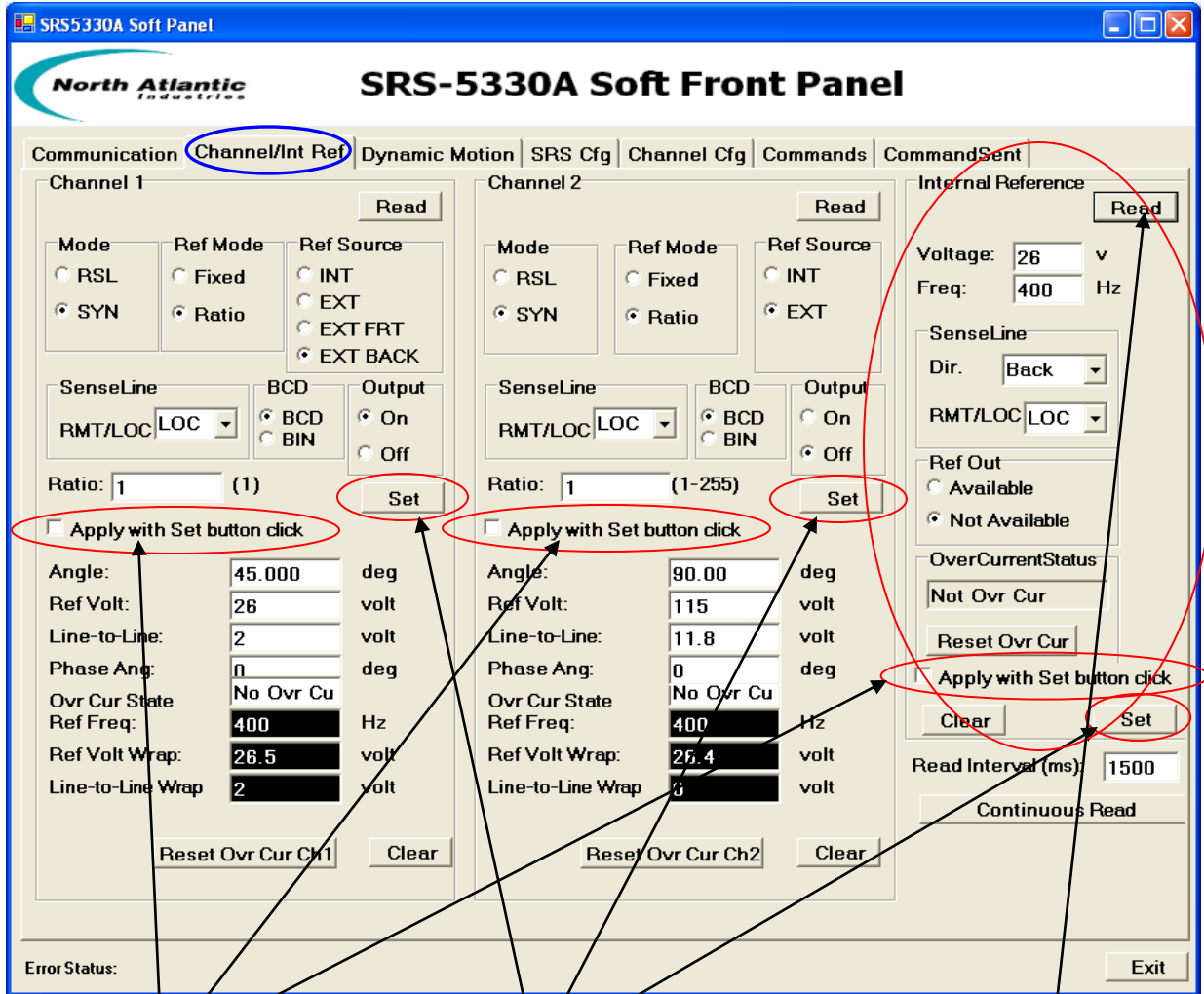
Choose the IEEE Language to communicate with SRS-5330A.

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

Writes/Queries Freeform commands.

Gets/Sets the SRS-5330A Communication Setup. Note, communication to SRS-5330A must first be established.

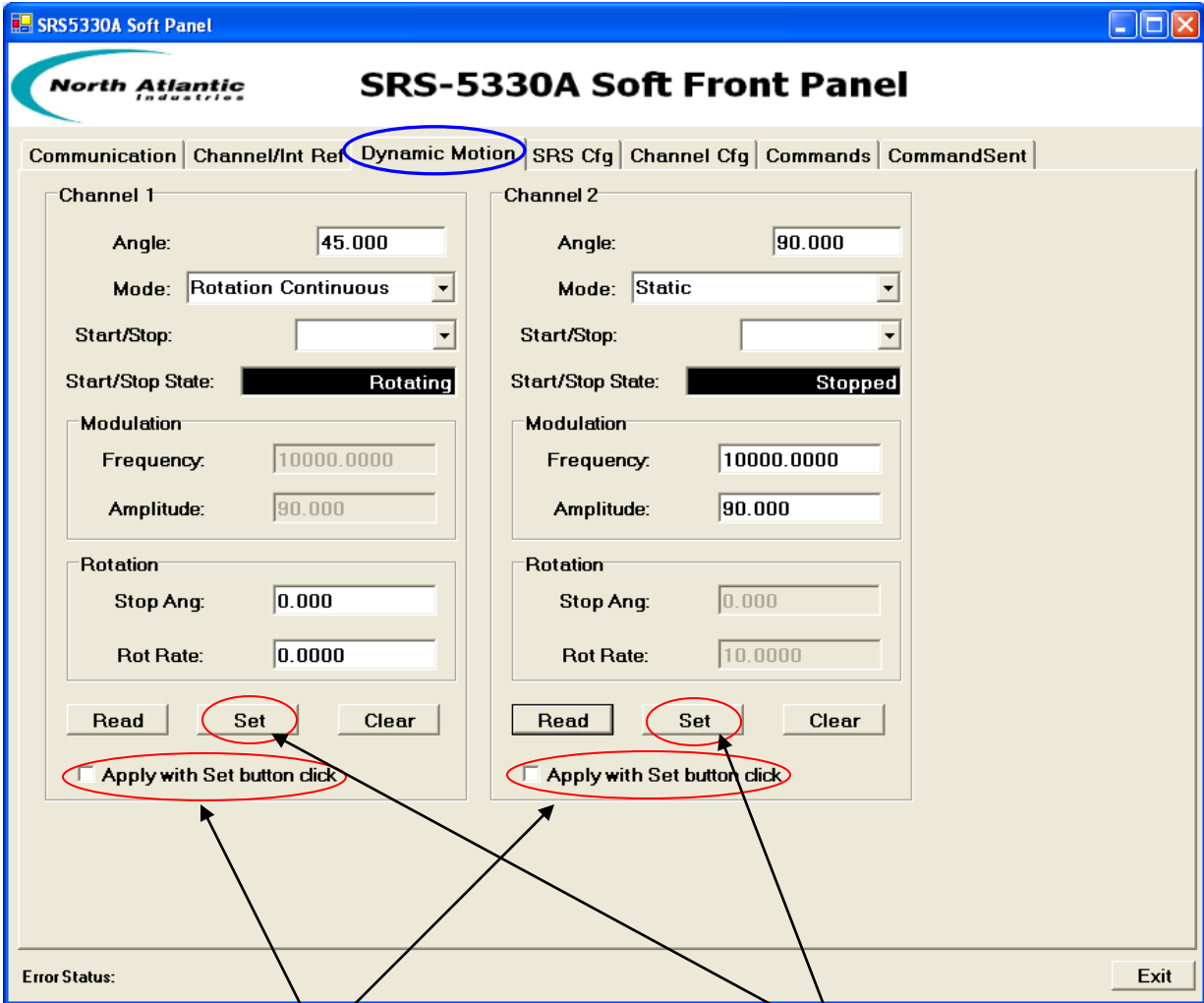
Scan detects the number of Cypress USB Devices connected to your computer. "IDN" is used to determine which USB connections are for SRS-5330A devices.



When checked, the changes will apply when the Set button is clicked.

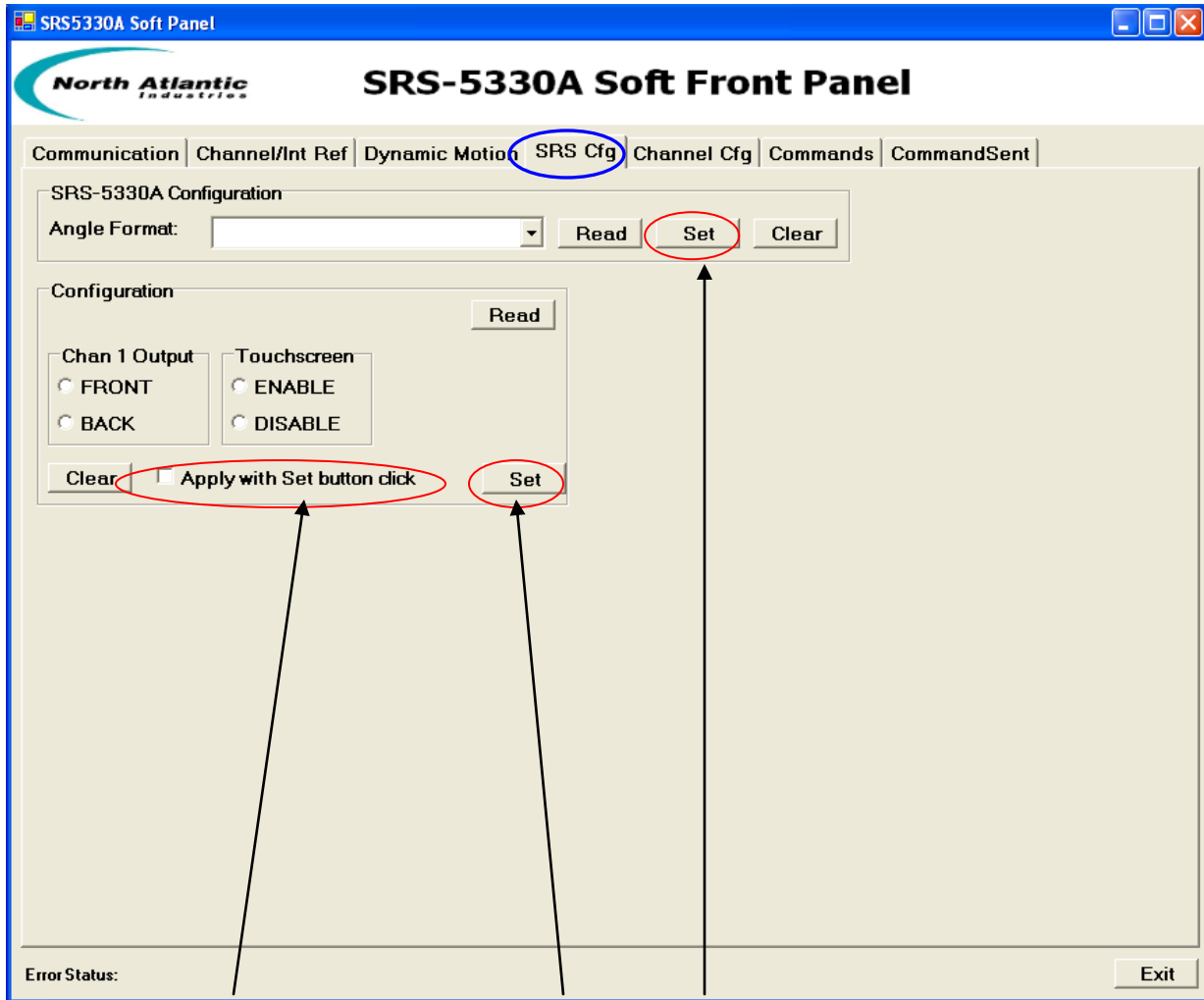
“Sets” are allowed only when the SRS-5330A unit is set for Remote mode with specified communication interface (IEEE, USB, or Ethernet)

On units where Internal Reference is not available, the SRS-5330A will return the default settings:
 Voltage: 26 v
 Frequency: 400 Hz
 Ref Out: Not Available



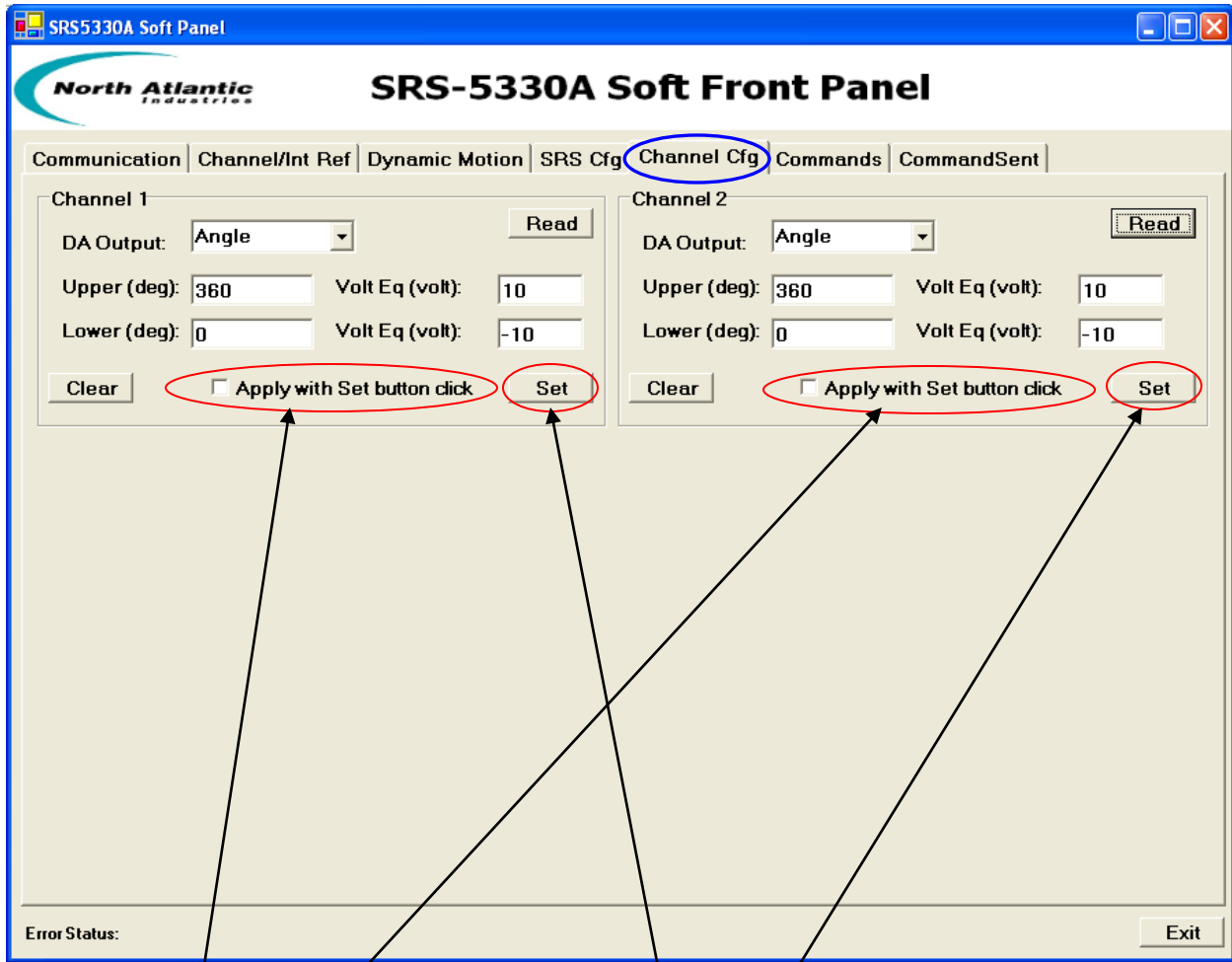
When checked, the changes will apply when the Set button is clicked.

“Sets” are allowed only when the SRS-5330A unit is set for Remote mode with specified communication interface (IEEE, USB, or Ethernet)



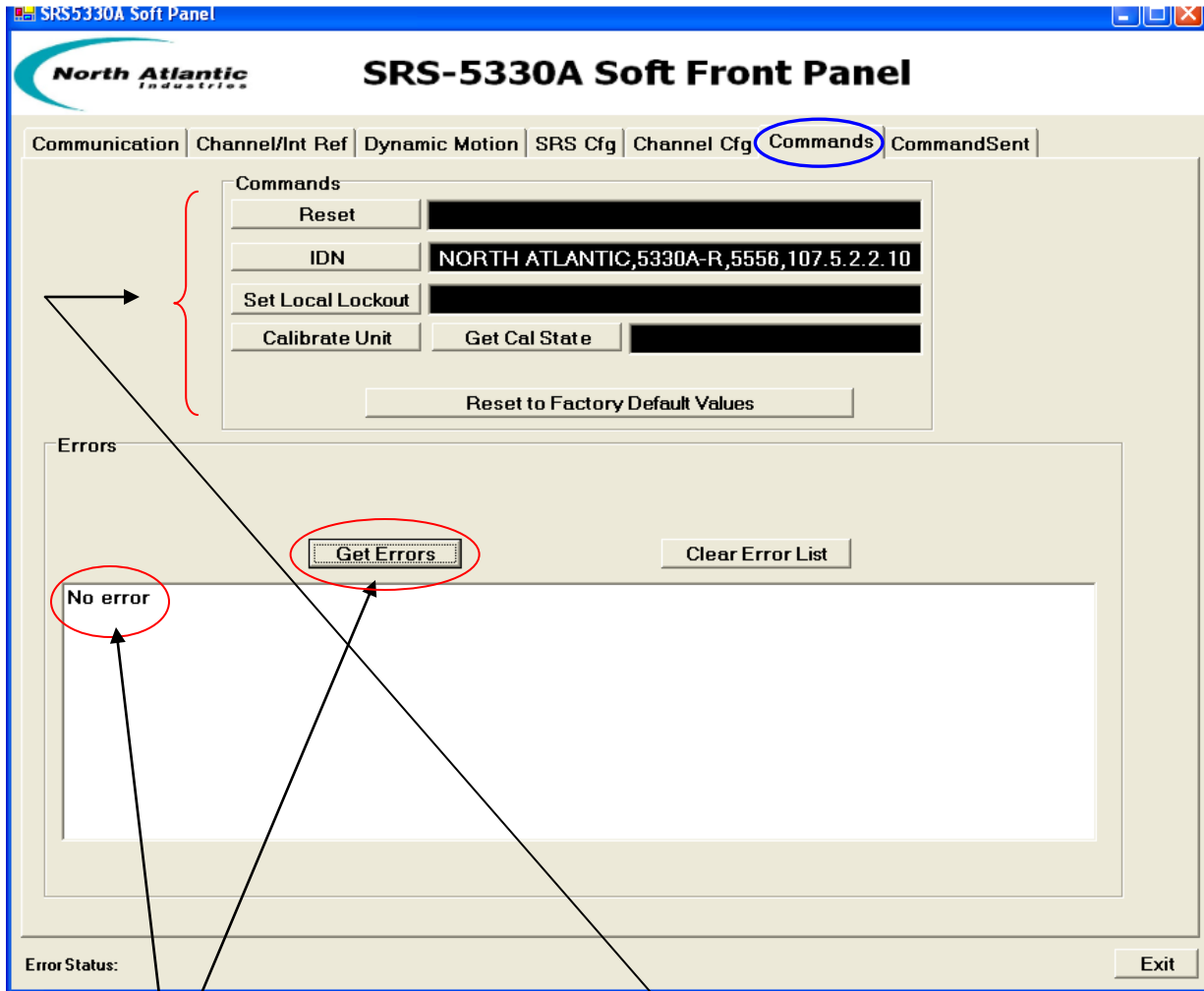
When checked, the changes will apply when the Set button is clicked.

“Sets” are allowed only when the SRS-5330A unit is set for Remote mode with specified communication interface (IEEE, USB, or Ethernet)



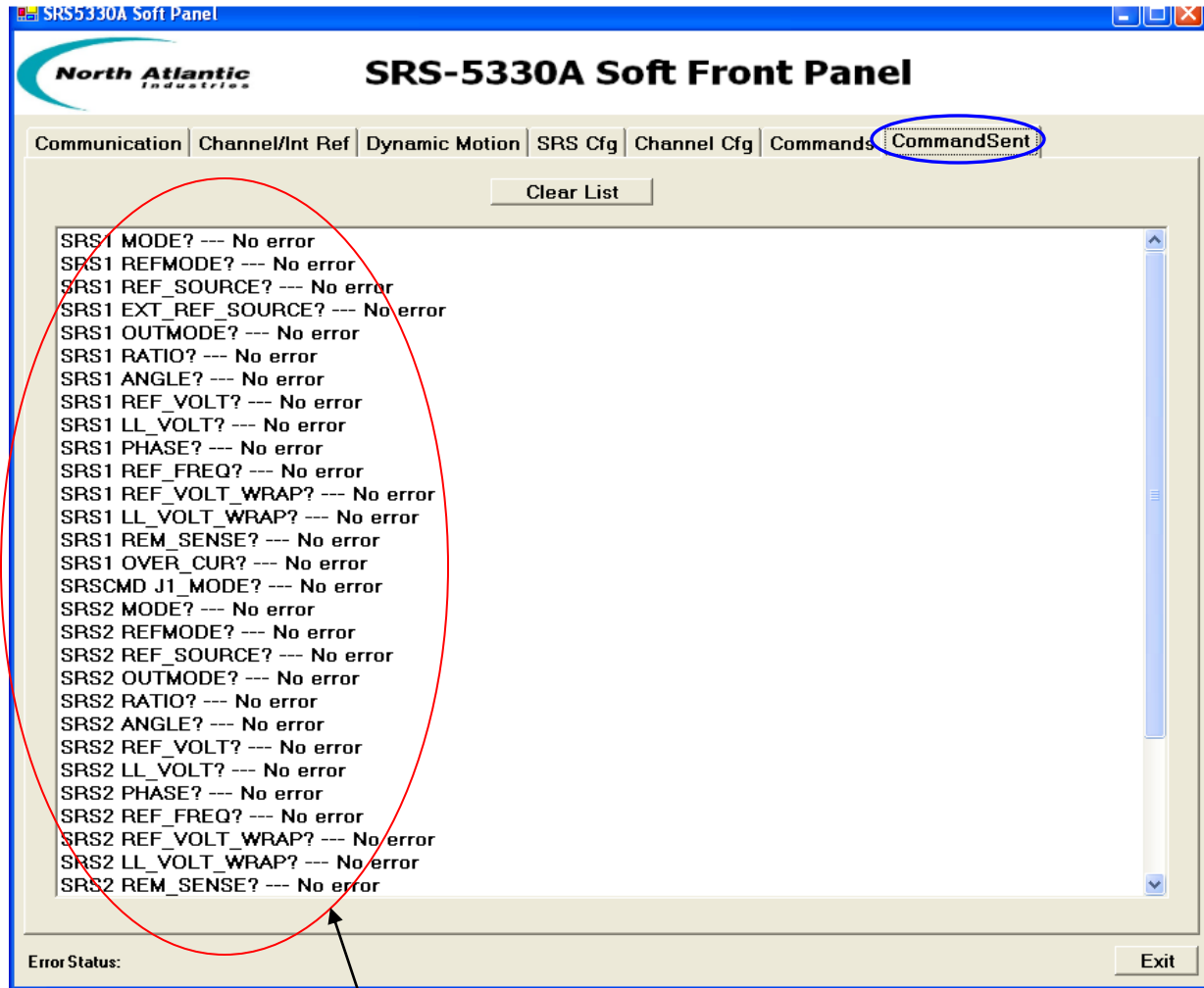
When checked, the changes will apply when the Set button is clicked.

“Sets” are allowed only when the SRS-5330A unit is set for Remote mode with specified communication interface (IEEE, USB, or Ethernet)



“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 SRS-5330A unit is set for Remote mode with specified communication interface (IEEE, USB, or Ethernet)



Commands sent to the SRS-5330A as well as the results from performing a call to the SRS5330A Dll's SRS5330A_GetErrors() method to retrieve any messages from the Error Queue.

8 Cypress USB Driver Installation

In order to communicate with API 8810A 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	Feb 05, 2010	Initial Release	gc
2.0.0.0	Dec 3, 2010	Added commands	bpm
2.0.0.1	Jan 17, 2011	Added Overcurrent Status/Reset Command	bpm
2.0.0.100	Feb 24, 2011	Updated screens for release 2.0.0.100	gc
3.0.0.1	Oct 8, 2011	Updated the Cypress USB Driver Installation section to refer to the “Cypress USB Installation” document. SRS5330A DII and SRS5330A Soft Panel Program are built with Microsoft Visual .NET 2010.	gc
3.0.0.2	Nov 16, 2012	No changes to API. Updated document revision to correspond to SSK release 3.0.0.2 which changed the Windows XP and Windows 7 folder names for the Cypress USB Driver.	gc