Function Reference Manual for 5330A

Two Channel Programmable Synchro/Resolver Simulator
and
One Optional Reference Supply

Function Reference Manual
for
5330A

Table of Contents

1 Introduction 5
2 SRS-5330A Connect/Disconnect Routines 11
2.1 SRS5330A_ConnectViaIEEE 11
2.2 SRS5330A_ConnectViaUSB 12
2.3 SRS5330A_ConnectViaEthernet 16
2.4 SRS5330A_DisconnectIEEE 16
2.5 SRS5330A_DisconnectUSB 17
2.6 SRS5330A_DisconnectEthernet 17
2.7 SRS5330A_GetSRS5330AUSBDeviceCnt 17
2.8 SRS5330A_GetSRS5330ADeviceIDN 18
3 SRS-5330A Channel Routines 19
3.1 SRS5330A_SetSRSAngle 19
3.2 SRS5330A_GetSRSAngle 19
3.3 SRS5330A_SetSRSReferenceMode 20
3.4 SRS5330A_GetSRSReferenceMode 20
3.5 SRS5330A_SetSRSRefVolt 21
3.6 SRS5330A_GetSRSRefVolt 22
3.7 SRS5330A_SetSRSLineLineVolt 22
3.8 SRS5330A_GetSRSLineLineVolt 23
3.9 SRS5330A_SetSRSReferenceSrc 23
3.10 SRS5330A_GetSRSReferenceSrc 24
3.11 SRS5330A_SetSRSExtRefSrcLoc 25
3.12 SRS5330A_GetSRSExtRefSrcLoc 25
3.13 SRS5330A_SetSRSSignalMode 26
3.14 SRS5330A_GetSRSSignalMode 27
3.15 SRS5330A_SetSRSPhase 27
3.16 SRS5330A_GetSRSPhase 28
3.17 SRS5330A_SetSRSOutMode 28
3.18 SRS5330A_GetSRSOutMode 29
3.19 SRS5330A_GetSRSVrefWrap 30
3.20 SRS5330A_GetSRSVLLWrap 30
3.21 SRS5330A_GetSRSRefFreqWrap 31
3.22 SRS5330A_SetSRSRatio 31
3.23 SRS5330A_GetSRSRatio 32
3.24 SRS5330A_ResetOverCurrent 32
3.25 SRS5330A_GetOverCurrentStatus 33
4 SRS-5330A Dynamic Motion Routines 34
4.1 SRS5330A_SetSRSDynMotionMode 34
4.2 SRS5330A_GetSRSDynMotionMode 34
4.3 SRS5330A_SetSRSDynMotionAmp 35
4.4 SRS5330A_GetSRSDynMotionAmp 36
4.5 SRS5330A_SetSRSDynMotionFreq 36
4.6 SRS5330A_GetSRSDynMotionFreq 37

4.7 SRS5330A_SetSRSDynMotionRotRate 37
4.8 SRS5330A_GetSRSDynMotionRotRate 38
4.9 SRS5330A_SetSRSDynMotionRotStopAng 39
4.10 SRS5330A_GetSRSDynMotionRotStopAng 39
4.11 SRS5330A_SetSRSDynMotionStartStop 40
4.12 SRS5330A_GetSRSDynMotionStartStop 40
4.13 SRS5330A_SetSRSDynMotionAngStepSize 41
4.14 SRS5330A_SetSRSDynMotionIncCClock 42
4.15 SRS5330A_SetSRSDynMotionIncClockwise 42
5 SRS-5330A Output Isolation Relays Routines 44
5.1 SRS5330A_SetSRSCloseIsolationRelay 44
5.2 SRS5330A_SetSRSOpenIsolationRelay 44
6 SRS-5330A Output Remote Sense Routines 46
6.1 SRS5330A_SetSRSRmtSenseState 46
6.2 SRS5330A_GetSRSRmtSenseState 46
7 SRS-5330A Reference Generator Remote Sense Routines 48
7.1 SRS5330A_SetSRSRefGenLocSenseDirection 48
7.2 SRS5330A_GetSRSRefGenLocSenseDirection 48
7.3 SRS5330A_SetSRSRefGenRmtSenseState 49
7.4 SRS5330A_GetSRSRefGenRmtSenseState 50
8 SRS-5330A D/A Output Routines 51
8.1 SRS5330A_SetSRSDAOutput 51
8.2 SRS5330A_GetSRSDAOutput 51
8.3 SRS5330A_SetSRSDAUpperLimit 52
8.4 SRS5330A_GetSRSDAUpperLimit 52
8.5 SRS5330A_SetSRSDALowerLimit 53
8.6 SRS5330A_GetSRSDALowerLimit 54
8.7 SRS5330A_SetSRSDAUpperVoltage 54
8.8 SRS5330A_GetSRSDAUpperVoltage 55
8.9 SRS5330A_SetSRSDALowerVoltage 55
8.10 SRS5330A_GetSRSDALowerVoltage 56
9 SRS-5330A Binary Coded Decimal (BCD) Routines 57
9.1 SRS5330A_SetBCDMode 57
9.2 SRS5330A_GetBCDMode 57
10 SRS-5330A Internal Reference Routines 59
10.1 SRS5330A_SetIntRefFreq 59
10.2 SRS5330A_GetIntRefFreq 59
10.3 SRS5330A_SetIntRefVolt 60
10.4 SRS5330A_GetIntRefVolt 60
10.5 SRS5330A_SetIntRefOutputState 61
10.6 SRS5330A_GetIntRefOutputState 62
10.7 SRS5330A_GetIntRefOverCurState 62
10.8 SRS5330A_ResetIntRefOverCur 63
11 SRS-5330A Command Routines 65
11.1 SRS5330A_PerformGetID 65
11.2 SRS5330A_Reset 65

11.3 SRS5330A_GetErrors 66
11.4 SRS5330A_SetInternalSelfTest 66
11.5 SRS5330A_SetConfidenceTest 67
11.6 SRS5330A_SetCombCmdBcdLegacy 68
12 SRS-5330A Configuration Routines 69
12.1 SRS5330A_GetIEEELang 69
12.2 SRS5330A_SetIEEELang 69
12.3 SRS5330A_GetCommState 70
12.4 SRS5330A_GoToLocal 71
12.5 SRS5330A_SetLocalLockout 71
12.6 SRS5330A_SetRemoteUSB 72
12.7 SRS5330A_SetRemoteEthernet 72
12.8 SRS5330A_SetRemoteIEEE 73
12.9 SRS5330A_SetRemoteJ1 73
12.10 SRS5330A_SetAngleFormat 74
12.11 SRS5330A_GetAngleFormat 74
12.12 SRS5330A_SetCh1Output 75
12.13 SRS5330A_GetCh1Output 76
12.14 SRS5330A_SetTouchscreenState 76
12.15 SRS5330A_GetTouchscreenState 77
12.16 SRS5330A_ResetDefaultValues 77
13 SRS-5330A Calibration Routines 79
13.1 SRS5330A_GetCalState 79
13.2 SRS5330A_Calibrate 79
13.3 SRS5330A_SetClrCalibrate 80
13.4 SRS5330A_GetSRSBackGroundCalState 80
14 SRS-5330A Miscellaneous Routines 82
14.1 SRS5330A_MaxRetry 82
14.2 SRS5330A_LastCmdSent 82
14.3 SRS5330A_WriteCommand 83
14.4 SRS5330A_QueryCommand 83
15 Appendix A – SRS5330ADll Constant Values 84
16 Appendix B- Error Codes 86

1 Introduction

This document lists the functions and describes the purpose, format, input and output parameters, and possible errors for each function.

Reference Documentation
For information about the operation of this instrument please refer to the *Operation Manual for Model 5330A*.
For information about general programming information please refer to the *Programmer's Reference Guide for 5330A.*

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

SRS-5330A Connect/Disconnect Routines	
SRS5330A_ConnectViaIEEE	Sets up and opens the connection to communicate to the 5330A via IEEE. The following IEEE languages are supported: • SRS-5330A Native • SRS-5330 Native (Legacy) • SRS-5310 BCD (Legacy) • SRS-5310 Binary (Legacy)
SRS5330A_ConnectViaUSB	Sets up and opens the connection to communicate to the 5330A via USB.
SRS5330A_ConnectViaEthernet	Sets up and opens the connection to communicate to the 5330A via Ethernet.
SRS5330A_DisconnectIEEE	Closes the IEEE connection.
SRS5330A_DisconnectUSB	Closes the USB connection.
SRS5330A_DisconnectEthernet	Closes the Ethernet connection.
SRS5330A_GetSRS5330AUSBDeviceCnt	Scan the USB Ports for the number of Cypress USB Devices. Note, after calling this function, call *SRS5330A_GetSRS5330ADeviceIDN()* routine to determine the device numbers for 5330A devices.
SRS5330A_GetSRS5330ADeviceIDN	Performs an IDN query on the Cypress USB Device and determines if the device being queried is an 5330A device. If it is, the IDN response which includes manufacture, model, and serial number is returned.
SRS-5330A Channel Routines	
SRS5330A_SetSRSAngle	Sets the channel's angle value.
SRS5330A_GetSRSAngle	Gets the channel's angle value.
SRS5330A_SetSRSReferenceMode	Sets the channel's reference mode (FIXED/RATIO).
SRS5330A_GetSRSReferenceMode	Gets the channel's reference mode.
SRS5330A_SetSRSRefVolt	Sets the channel's reference voltage value.
SRS5330A_GetSRSRefVolt	Gets the channel's reference voltage value.
SRS5330A_SetSRSLineLineVolt	Sets the channel's line-to-line voltage value.
SRS5330A_GetSRSLineLineVolt	Gets the channel's line-to-line voltage value.
SRS5330A_SetSRSReferenceSrc	Sets the channel's reference source mode (INT/EXT).
SRS5330A_GetSRSReferenceSrc	Gets the channel's reference source mode.
SRS5330A_SetSRSExtRefSrcLoc	Sets the external source input mode. (EXTERNAL_BACK/EXTERNAL_FRONT) for Channel 1.
SRS5330A_GetSRSExtRefSrcLoc	Gets the external source input mode for Channel 1.
SRS5330A_SetSRSSignalMode	Sets the channel's signal mode (SYN/RSL).
SRS5330A_GetSRSSignalMode	Gets the channel's signal mode.

SRS5330A_SetSRSPhase	Sets the channel’s phase angle value.
SRS5330A_GetSRSPhase	Gets the channel’s phase angle value.
SRS5330A_SetSRSOutMode	Sets the channel’s output mode (ON|OFF).
SRS5330A_GetSRSOutMode	Gets the channel’s output mode.
SRS5330A_GetSRSVrefWrap	Gets the channel’s reference voltage wrap value.
SRS5330A_GetSRSVLLWrap	Gets the channel’s line-to-line voltage wrap value.
SRS5330A_GetSRSRefFreqWrap	Gets the channel’s reference frequency wrap value.
SRS5330A_SetSRSRatio	Sets the channel’s ratio value.
SRS5330A_GetSRSRatio	Gets the channel’s ratio value.
SRS5330A_ResetOverCurrent	Resets the channel to the normal state if it is in the over-current state.
SRS5330A_GetOverCurrentStatus	Gets the channel’s over-current state.
SRS-5330A Dynamic Motion Routines	
SRS5330A_SetSRSDynMotionMode	Sets the channel’s dynamic motion mode (Static, Sine, Ramp, Square, Triangle, Continuous Rotation or Start/Stop Rotation).
SRS5330A_GetSRSDynMotionMode	Gets the channel’s dynamic motion mode.
SRS5330A_SetSRSDynMotionAmp	Sets the channel’s dynamic motion amplitude value.
SRS5330A_GetSRSDynMotionAmp	Gets the channel’s dynamic motion amplitude value.
SRS5330A_SetSRSDynMotionFreq	Sets the channel’s dynamic motion frequency value.
SRS5330A_GetSRSDynMotionFreq	Gets the channel’s dynamic motion frequency value.
SRS5330A_SetSRSDynMotionRotRate	Sets the channel’s dynamic motion rotation rate.
SRS5330A_GetSRSDynMotionRotRate	Gets the channel’s dynamic motion rotation rate.
SRS5330A_SetSRSDynMotionRotStopAng	Sets the channel’s dynamic motion rotation stop angle value.
SRS5330A_GetSRSDynMotionRotStopAng	Gets the channel’s dynamic motion rotation stop angle value.
SRS5330A_SetSRSDynMotionStartStop	Starts or Stops the channel’s dynamic motion.
SRS5330A_GetSRSDynMotionStartStop	Gets the channel’s dynamic motion state.
SRS5330A_SetSRSDynMotionAngStepSize	Sets the angle step for the Increment CCW and Increment CW commands. Note only support in IEEE_5330_NATIVE Language mode.
SRS5330A_SetSRSDynMotionIncCClock	Set the command to increment counter-clockwise. Note only support in IEEE_5330_NATIVE Language mode.
SRS5330A_SetSRSDynMotionIncClockwise	Set the command to increment clockwise. Note

	only support in IEEE_5330_NATIVE Language mode.
SRS-5330A Output Isolation Relays Routines	
SRS5330A_SetSRSCloseIsolationRelay	Closes the isolation relays. Note only support in IEEE_5330_NATIVE Language mode.
SRS5330A_SetSRSOpenIsolationRelay	Opens the isolation relays. Note only support in IEEE_5330_NATIVE Language mode.
SRS-5330A Output Remote Sense Routines	
SRS5330A_SetSRSRmtSenseState	Sets the channel’s remote sense state.
SRS5330A_GetSRSRmtSenseState	Gets the channel’s remote sense state.
SRS-5330A Reference Generator Remote Sense Routines	
SRS5330A_SetSRSRefGenLocSenseDirection	Sets the internal reference generator’s sense direction (BACK/FRONT).
SRS5330A_GetSRSRefGenLocSenseDirection	Gets the internal reference generator’s sense direction.
SRS5330A_SetSRSRefGenRmtSenseState	Sets the internal reference generator’s sense state (DISABLE (Local Sense) /ENABLE (Remote Sense)).
SRS5330A_GetSRSRefGenRmtSenseState	Gets the internal reference generator’s sense state.
SRS-5330A D/A Output Routines	
SRS5330A_SetSRSDAOutput	Sets channel’s data type (angle or velocity) to use for DA output.
SRS5330A_GetSRSDAOutput	Gets channel’s data type (angle or velocity) to use for DA output.
SRS5330A_SetSRSDAUpperLimit	Sets the channel’s upper angle or velocity limit value for DA output.
SRS5330A_GetSRSDAUpperLimit	Gets the channel’s upper angle or velocity limit value for DA output.
SRS5330A_SetSRSDALowerLimit	Sets the channel’s lower angle or velocity limit value for DA output.
SRS5330A_GetSRSDALowerLimit	Gets the channel’s lower angle or velocity limit value for DA output.
SRS5330A_SetSRSDAUpperVoltage	Sets the channel’s voltage value associated with the upper limit value for DA output.
SRS5330A_GetSRSDAUpperVoltage	Gets the channel’s voltage value associated with the upper limit value for DA output
SRS5330A_SetSRSDALowerVoltage	Sets the channel’s voltage value associated with the lower limit value for DA output.
SRS5330A_GetSRSDALowerVoltage	Gets the channel’s voltage value associated with the lower limit value for DA output
SRS-5330A Binary Coded Decimal (BCD) Routines	
SRS5330A_SetBCDMode	Sets the BCD (BCD/BIN) mode.
SRS5330A_GetBCDMode	Gets the BCD mode.
SRS-5330A Internal Reference Routines	

SRS5330A_SetIntRefFreq	Sets the internal reference frequency value.
SRS5330A_GetIntRefFreq	Gets the internal reference frequency value.
SRS5330A_SetIntRefVolt	Sets the internal reference voltage value.
SRS5330A_GetIntRefVolt	Gets the internal reference voltage value.
SRS5330A_SetIntRefOutputState	Sets the internal reference output state.
SRS5330A_GetIntRefOutputState	Gets the internal reference output state.
SRS5330A_GetIntRefOverCurState	Gets the internal reference over-current state.
SRS5330A_ResetIntRefOverCur	Resets the internal reference if it is in an over-current state.
SRS-5330A Command Routines	
SRS5330A_PerformGetID	Gets the Device ID.
SRS5330A_Reset	Resets the device.
SRS5330A_GetErrors	Gets the error message from the error queue.
SRS5330A_SetInternalSelfTest	Sends command to perform an internal self test on the 5330A. Note only support in IEEE_5330_NATIVE Language mode.
SRS5330A_SetConfidenceTest	Sends command to perform confidence test on the 5330A. Note only support in IEEE_5330_NATIVE Language mode.
SRS5330A_SetCombCmdBcdLegacy	Sends a combined command to set the angle, output mode (RSL/SYN), reference voltage and signal line-to-line voltage. Note only support in IEEE_5310_BCD Language mode.
SRS-5330A Configuration Routines	
SRS5330A_GetIEEELang	Gets the IEEE Language protocol configured in the 5330A.
SRS5330A_SetIEEELang	Sets the IEEE Language protocol to accept in the 5330A.
SRS5330A_GetCommState	Gets the communication settings.
SRS5330A_GoToLocal	Sets the device to Local mode.
SRS5330A_SetLocalLockout	Sets the device to Local Lockout mode.
SRS5330A_SetRemoteUSB	Sets the device to Remote USB mode.
SRS5330A_SetRemoteEthernet	Sets the device to Remote Ethernet mode.
SRS5330A_SetRemoteIEEE	Sets the device to Remote IEEE mode.
SRS5330A_SetRemoteJ1	Sets the device to Remote J1 mode.
SRS5330A_SetAngleFormat	Sets the Angle Format.
SRS5330A_GetAngleFormat	Gets the Angle Format
SRS5330A_SetCh1Output	Sets the Channel 1 Output Connector configuration.
SRS5330A_GetCh1Output	Gets the Channel 1 Output Connector configuration.
SRS5330A_SetTouchscreenState	Sets the Touch screen mode.
SRS5330A_GetTouchscreenState	Gets the Touch screen mode.
SRS5330A_ResetDefaultValues	Sets the device to the default factory settings.
SRS-5330A Calibration Routines	

SRS5330A_GetCalState	Gets the calibration state.
SRS5330A_Calibrate	Calibrates the 5330A.
SRS5330A_SetClrCalibrate	Sends command to clear all calibration on the 5330A. Note only support in IEEE_5330_NATIVE Language mode
SRS5330A_GetSRSBackGroundCalState	Gets the back-ground calibration state.
SRS-5330A Miscellaneous Routines	
SRS5330A_MaxRetry	Sets the number of retries for re-sending data after a timeout or problem sending or reading data from device. Default value for max retry is 0.
SRS5330A_LastCmdSent	Returns the last command set to the device by Dll.
SRS5330A_WriteCommand	Sends the freeform command to the 5330A.
SRS5330A_QueryCommand	Sends the freeform command to the 5330A and waits for a response.

2 SRS-5330A Connect/Disconnect Routines

The routines in this section handle IEEE, Ethernet and USB communications to the 5330A device.

2.1 SRS5330A_ConnectViaIEEE

Format:

```
_SRS5330AFUNC int SRS5330A_ConnectViaIEEE
(
   int srsNo,
   int nIEEEAddr,
   int nIEEELang
)
```

Function Description:

This function sets up and opens the connection to communicate to the 5330A via IEEE.

The IEEE supports the following language protocols:

- SRS-5330A Native
- SRS-5330 Native (Legacy)
- SRS-5310 Native BCD (Legacy)
- SRS-5310 Native Binary (Legacy)

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nIEEEAddr - IEEE Address to be used to connect to 5330A.(0-30)
nIEEELang - Language Protocol to be used to communicate via IEEE to 5330A.
                  5330A Language Types:
                     SRS_5330A_NATIVE     0
                     IEEE_5330_NATIVE      1
                     IEEE_5310_BCD         2
                     IEEE_5310_BINARY      3
```

Return Value:

```
SRS_SUCCESS – successfully connection via IEEE using specified address and
      language protocol
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_ADDRS - invalid IEEE Address parameter
SRS_ERROR_LANG - invalid 5330A Language parameter
SRS_ERROR_OPEN_SRS_SESSION - IEEE connection or configuration error
```

References for this function:

This function will make a call to the SRS5330A_SetIEEELang() routine to force the 5330A device to handle commands in the language specified.

2.2 SRS5330A_ConnectViaUSB

Format:

```
_SRS5330AFUNC int SRS5330A_ConnectViaUSB
(
   int srsNo,
   int nDeviceNo
)
```

Function Description:

This function sets up and opens the connection to communicate to the 5330A via USB.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nDeviceNo – Device Number to be used to connect to 5330A. (0-30)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_USB_CONNECTION– USB connection error
```

References for this function:

Prior to calling this function, make calls to the SRS5330A_GetSRS5330AUSBDeviceCnt() routine to determine the number of Cypress USB Devices detected in your system and the SRS5330A_GetSRS5330ADeviceIDN() routine to determine the device number (DeviceNo) associated with the Cypress USB Devices that are connected to 5330A via USB.

Sample Code:

The following sample code is available in the 5330A Software Package under the folder: ..\Driver\Source\SRS5330AUSBConnect. The sample code is written in C, compiled under Microsoft Visual .NET 2003 and invokes the routines in the SRS-5330A Dll that has been included in the software package.

```
#include <stdio.h>
#include <stdlib.h>
#include <Windows.h>
#include <Wincon.h>

#define _BUILD_SRS5330ADLL
#define __WIN32__
#include "SRS5330ADll.h"

/* Prototype definition for the Console Window */
extern WINBASEAPI HWND WINAPI GetConsoleWindow ();

/**************************************************************************
*  Function:     main
*  Description:  Main routine for SRS5330A USB Connection application.
*                Scan for Cypress USB devices and opens each USB device to
*                determine which one is connected to a 5330A via IDN command.
*                Prompts the user for the USB Endpoint associated with 5330A
*                device and reads and displays the Signal Mode for Channel 1.
*  Parameters:   None
*  Return:       1 if successful.
*                -1 if any failure to SRS5330ADll calls.
***************************************************************************/
int main()
{
   HANDLE hWnd;
   COORD bufferSize;
   BOOL bRetry;
   int nStatus;
   int nUSBDeviceCnt;
   int n5330ADeviceCnt;
   char sz5330AIDN[100];
   int i,j;
   /* This keeps track of the array of the Cypress USB Endpoints.
      We are only interested in the ones that are connected to the 5330A
   */
   int aUSB_5330A_Endpoints[MAX_SRS];
   BOOL bValidEntry;
   int nConnectEndpoint;
   int nSigMode;
   char buff[10];

   /* Get the console window */
   hWnd = GetConsoleWindow();

   /* Create a COORD to hold the buffer size and change the internal buffer size */
   bufferSize.X = 800;
   bufferSize.Y = 800;
   SetConsoleScreenBufferSize(hWnd, bufferSize);

   /* Move and resize the window */
   MoveWindow(hWnd, 5, 5, 800, 600, TRUE);

   /* Change the window title */
   SetConsoleTitle("SRS5330A USB Connection");

   bRetry = TRUE;
   while (bRetry)
   {
      /* Initialize the array of Cypress USB Endpoints to -1 */
      for (i = 0; i < MAX_SRS; i++)
         aUSB_5330A_Endpoints[i] = -1;

      /* Scan for USB Devices */
      nStatus = SRS5330A_GetSRS5330AUSBDeviceCnt(&nUSBDeviceCnt);
      if (nStatus != SRS_SUCCESS)
      {
         printf("\nSRS5330A_GetSRS5330AUSBDeviceCnt Error: %d", nStatus);
         return -1;
      }
```

```
printf("\nDetected %d Cypress USB Devices:", nUSBDeviceCnt);

if (nUSBDeviceCnt > 0)
{
   /* Determine which USB devices are connected to 5330A */
   n5330ADeviceCnt = 0;
   for (i = 0; i < nUSBDeviceCnt; i++)
   {
      nStatus = SRS5330A_GetSRS5330ADeviceIDN(i, &sz5330AIDN[0]);
      if (nStatus != SRS_SUCCESS)
      {
         /* The USB device connected is not a 5330A */
         printf("\nUSB Endpoint: %d   NOT 5330A device", i);
      }
      else
      {
         /* Replace the \r\n with a terminator character (\0) */
         for (j = 0; j < (int)strlen(sz5330AIDN); j++)
         {
            if (sz5330AIDN[j] == '\r')
            {
               sz5330AIDN[j] = '\0';
               break;
            }
         }
         /* Track the endpoints that are connected to 5330A */
         aUSB_5330A_Endpoints[i] = i;
         n5330ADeviceCnt++;

         /* Display the IDN information */
         printf("\nUSB Endpoint: %d   IDN:%s", i, sz5330AIDN);
      }
   }

   /* Request 5330A USB device to connect to */
   bValidEntry = FALSE;
   while(!bValidEntry)
   {
      printf("\n\nPlease Enter USB Endpoint Device for 5330A to Connect: ");
      scanf("%d", &nConnectEndpoint);
      if (nConnectEndpoint < MAX_SRS)
      {
         if (aUSB_5330A_Endpoints[nConnectEndpoint] == -1)
            printf("Endpoint entered is connected to 5330A.\n");
         else
            bValidEntry = TRUE;
      }
      else
         printf("Endpoint entered is not valid");
   }

   /* Connect to 5330A */
   nStatus = SRS5330A_ConnectViaUSB(1, nConnectEndpoint);
   if (nStatus != SRS_SUCCESS)
   {
      printf("\nSRS5330A_ConnectViaUSB Error: %d", nStatus);
      return -1;
   }

   /* Get Channel 1 Mode Information */
   nStatus = SRS5330A_GetSRSSignalMode(1, 1, &nSigMode);
   if (nStatus != SRS_SUCCESS)
   {
      printf("\nSRS5330A_GetSRSSignalMode Error: %d", nStatus);
      return -1;
   }

   /* Display Channel 1 Mode Information */
   if (nSigMode == RESOLVER)
```

```
            printf("\nChannel 1 Signal Mode = RESOLVER");
         else if (nSigMode == SYNCHRO)
            printf("\nChannel 1 Signal Mode = SYNCHRO");
         else
            printf("\nChannel 1 Signal Mode = UNKNOWN");

         /* Disconnect from Cypress USB Devices */
         nStatus = SRS5330A_DisconnectUSB(1);
         if (nStatus != SRS_SUCCESS)
         {
            printf("\nSRS5330A_DisconnectUSB Error: %d", nStatus);
            return -1;
         }

         bRetry = FALSE;
      }
      else
      {
         /* Request retry to find USB devices */
         bValidEntry = FALSE;
         while(!bValidEntry)
         {
            printf("\nPlease type 'y' or 'Y' to retry USB detection:  ");
            memset( buff,0x00,sizeof(buff));
            scanf("%s",buff);
            if ((buff[0] == 'y') || (buff[0] == 'Y'))
               bRetry = TRUE;
            else
               bRetry = FALSE;
            bValidEntry = TRUE;
         }
      }
   }

   /* User must hit a 'q' or 'Q' to exit program */
   bValidEntry = FALSE;
   while(!bValidEntry)
   {
      printf("\n\nPlease type q or Q to quit:  ");
      memset( buff,0x00,sizeof(buff));
      scanf("%s",buff);
      if ((buff[0] == 'q') || (buff[0] == 'Q'))
         bValidEntry = TRUE;
   }

   return 1;
}
```

Output:

```
Detected 1 Cypress USB Devices:
USB Endpoint: 0   IDN: NORTH ATLANTIC,5330A,123,5.2.0.0.0

Please Enter USB Endpoint Device for 5330A to Connect: 0

Channel 1 Signal Mode = RESOLVER

Please type q or Q to quit:
```

2.3 SRS5330A_ConnectViaEthernet

Format:

```
_SRS5330AFUNC int SRS5330A_ConnectViaEthernet
(
   int srsNo,
   char *szIPAddr,
   int nPort
)
```

Function Description:

```
This function sets up and opens the connection to communicate to the 5330A
via Ethernet.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
szIPAddr - IP Address to be used to connect to 5330A.
nPort – Port to be used to connect to 5330A.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE – unable to perform ethernet login to 5330A
SRS_ERROR_ETHER_CONNECTION– Ethernet connection error
```

References for this function:

```
None.
```

2.4 SRS5330A_DisconnectIEEE

Format:

```
_SRS5330AFUNC int SRS5330A_DisconnectIEEE
(
   int srsNo
)
```

Function Description:

```
This function closes the connection to communicate to the 5330A via IEEE.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
```

References for this function:

```
Prior to calling this function, make call to the SRS5330A_ConnectViaIEEE()
routine to connected to 5330A via IEEE.
```

2.5 SRS5330A_DisconnectUSB

Format:

```
_SRS5330AFUNC int SRS5330A_DisconnectUSB
(
   int srsNo
)
```

Function Description:

This function closes the connection to communicate to the 5330A via USB.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter

References for this function:

Prior to calling this function, make call to the SRS5330A_ConnectViaUSB() routine to connected to 5330A via USB.

2.6 SRS5330A_DisconnectEthernet

Format:

```
_SRS5330AFUNC int SRS5330A_DisconnectEthernet
(
   int srsNo
)
```

Function Description:

This function closes the connection to communicate to the 5330A via Ethernet.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter

References for this function:

Prior to calling this function, make call to the SRS5330A_ConnectViaEthernet() routine to connected to 5330A via Ethernet.

2.7 SRS5330A_GetSRS5330AUSBDeviceCnt

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRS5330AUSBDeviceCnt
(
   int *pnUSBDeviceCnt
)
```

Function Description:

This function invokes the Cypress driver and returns the number of Cypress USB Devices detected with your computer system.

Parameters:

pnUSBDeviceCnt – pointer to location to return the number of Cypress USB Devices detected

Return Value:

SRS_SUCCESS - function is successful

References for this function:

None.

2.8 SRS5330A_GetSRS5330ADeviceIDN

Format:

```
_SRS5330AFUNC int SRS5330A_ GetSRS5330ADeviceIDN
(
   int nDeviceNo,
   char *pszIDN
)
```

Function Description:

This function opens the USB device specified by the device number and performs an IDN query (*IDN?\r\n) to retrieval information about the device. If the device responds with "NORTH ATLANTIC,5330A" or its associated model, this function will populate the IDN string with the information retrieved from the device.

Parameters:

nDeviceNo – USB device number to open and communication via USB with 5330A.

pszIDN – pointer to location to return the IDN query response

Return Value:

SRS_SUCCESS - function is successful

SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

SRS_ERROR_USB_CONNECTION – USB connection error when open device with Device number specified.

References for this function:

None.

3 SRS-5330A Channel Routines

The routines in this section handle retrieving channel information from the 5330A device and setting channel configurations.

3.1 SRS5330A_SetSRSAngle

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSAngle
(
   int srsNo,
   int nChanNo,
   double dAngle
)
```

Function Description:

```
This function sends the command to set the angle value for the specified
5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dAngle – angle value to set (Unipolar Range: 0 to 359.999 or Bipolar Range: -
         180.000 to 180.000)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

```
None.
```

3.2 SRS5330A_GetSRSAngle

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSAngle
(
   int srsNo,
   int nChanNo,
   double *pdAngle
)
```

Function Description:

```
This function sends the command to get angle value for the specified 5330A
channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdAngle – pointer to location to return the angle value
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.3 SRS5330A_SetSRSReferenceMode

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSReferenceMode
(
   int srsNo,
   int nChanNo,
   int nRefMode
)
```

Function Description:

This function sends the command to set the specified 5330A channel to either Reference Fixed or Reference Ratio-metric mode.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
nRefMode – reference mode to set channel
                  Mode Types:
                     FIXED         0
                     RATIO         1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.4 SRS5330A_GetSRSReferenceMode

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSReferenceMode
(
   int srsNo,
   int nChanNo,
   int *pnRefMode
)
```

Function Description:

```
This function sends the command to get the reference mode for the specified
5330A channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pnRefMode – pointer to location to return the channel reference mode
                  Mode Types:
                     FIXED          0
                     RATIO          1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

3.5 *SRS5330A_SetSRSRefVolt*

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSRefVolt
(
   int srsNo,
   int nChanNo,
   double dRefVolt
)
```

Function Description:

```
This function sends the command to set the reference voltage value for the
specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dRefVolt – reference voltage (Range: 0 to 115 volts)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
```

```
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.6 SRS5330A_GetSRSRefVolt

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSRefVolt
(
   int srsNo,
   int nChanNo,
   double *pdRefVolt
)
```

Function Description:

This function sends the command to get reference voltage value for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdRefVolt – pointer to location to return the reference voltage value
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.7 SRS5330A_SetSRSLineLineVolt

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSLineLineVolt
(
   int srsNo,
   int nChanNo,
   double dLineLineVolt
)
```

Function Description:

This function sends the command to set the Line-to-Line voltage value for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dLineLineVolt – line-to-line voltage to set to (Range: 0.0 to 90.0 volts)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.8 SRS5330A_GetSRSLineLineVolt

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSLineLineVolt
(
   int srsNo,
   int nChanNo,
   double *pdLineLineVolt
)
```

Function Description:

This function sends the command to get the Line-to-Line voltage value for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdLineLineVolt – pointer to location to return the line-to-line voltage value
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.9 SRS5330A_SetSRSReferenceSrc

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSReferenceSrc
(
   int srsNo,
   int nChanNo,
   int nRefSrc
)
```

Function Description:

```
This function sends the command to set 5330A channel specified to internal or
external reference mode.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
nRefSrc – reference source to set channel
                  Reference Source Types:
                     INTERNAL         0
                     EXTERNAL         1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.10 *SRS5330A_GetSRSReferenceSrc*

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSReferenceSrc
(
   int srsNo,
   int nChanNo,
   int *pnRefSrc
)
```

Function Description:

```
This function sends the command to get the reference mode for the specified
5330A channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pnRefSrc – pointer to location to return the reference source
                  Reference Source Types:
                     INTERNAL         0
                     EXTERNAL         1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

```
SRS_ERROR_DATA - data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.11 SRS5330A_SetSRSExtRefSrcLoc

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSExtRefSrcLoc
(
   int srsNo,
   int nChanNo,
   int nExtRefSrcLoc

)
```

Function Description:

This function sends the command to set the external source input for channel 1. The external source input could come from the front or back. Channel 2 is not configurable since its external source input can only come from the back.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo - 5330A Channel
nRefSrc - reference source to set channel
                  Reference Source Types:
                     EXTERNAL_BACK     0
                     EXTERNAL_FRONT    1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED - function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE - parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.12 SRS5330A_GetSRSExtRefSrcLoc

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSExtRefSrcLoc
(
   int srsNo,
   int nChanNo,
   int *pnExtRefSrc
)
```

Function Description:

```
This function sends the command to get the external source input for channel
1.  The external source input could come from the front or back.  Channel 2
is not configurable since its external source input can only come from the
back.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo – 5330A Channel
pnRefSrc – pointer to location to return the reference source
                  Reference Source Types:
                      EXTERNAL_BACK    0
                      EXTERNAL_FRONT   1
```

Return Value:

```
 SRS_SUCCESS - function is successful
 SRS_ERROR_SRSNO - invalid srsNo parameter
 SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
 SRS_ERROR_WRITE - unable to send command to 5330A
 SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.13 SRS5330A_SetSRSSignalMode

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSSignalMode
(
   int srsNo,
   int nChanNo,
   int nSigMode
)
```

Function Description:

```
This function sends the command to set specified 5330A channel to either
resolver or synchro mode.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
nSigMode – signal mode to set channel
                  Mode Types:
                     RESOLVER          0
                     SYNCHRO           1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.14 SRS5330A_GetSRSSignalMode

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSSignalMode
(
   int srsNo,
   int nChanNo,
   int *pnSigMode
)
```

Function Description:

This function sends the command to get the signal mode for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pnSigMode – pointer to location to return the channel signal mode
                  Mode Types:
                     RESOLVER          0
                     SYNCHRO           1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.15 SRS5330A_SetSRSPhase

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSPhase
(
   int srsNo,
   int nChanNo,
   double dPhase
)
```

Function Description:

This function sends the command to set the Phase value for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dPhase – phase angle to set to (Range: 0.0 to 359.999)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.16 SRS5330A_GetSRSPhase

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSPhase
(
   int srsNo,
   int nChanNo,
   double *pdPhase
)
```

Function Description:

This function sends the command to get the Phase value for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdPhase – pointer to location to return the phase value
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.17 SRS5330A_SetSRSOutMode

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSOutMode
(
   int srsNo,
   int nChanNo,
   bool bOn
)
```

Function Description:

This function sends the command to turn on/off the output of the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
bOn – true indicates turning the output on; false indicates turning the
         output off.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.18 *SRS5330A_GetSRSOutMode*

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSOutMode
(
   int srsNo,
   int nChanNo,
   bool *pbOn
)
```

Function Description:

This function sends the command to get the output state of the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pbOn – pointer to location to return the channel output state
                  Output State:
                     ON       true
                     OFF      false
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.19 SRS5330A_GetSRSVrefWrap

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSVrefWrap
(
   int srsNo,
   int nChanNo,
   double *pdVrefWrap
)
```

Function Description:

This function sends the command to get reference voltage wrap value for the specified 5330A channel.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdVrefWrap – pointer to location to return the reference voltage wrap value

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

3.20 SRS5330A_GetSRSVLLWrap

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSVLLWrap
(
   int srsNo,
   int nChanNo,
   double *pdVLLWrap
)
```

Function Description:

This function sends the command to get Line-to-Line voltage wrap value for the specified 5330A channel.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdVLLWrap – pointer to location to return the Line-to-Line voltage value

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A

SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

3.21 SRS5330A_GetSRSRefFreqWrap

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSRefFreqWrap
(
   int srsNo,
   int nChanNo,
   double *pdRefFreq
)
```

Function Description:

This function sends the command to get reference frequency wrap value for the specified 5330A channel.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdRefFreq – pointer to location to return the reference frequency value

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

3.22 SRS5330A_SetSRSRatio

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSRatio
(
   int srsNo,
   int nChanNo,
   int nRatio
)
```

Function Description:

This function sends the command to set the ratio value for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel

```
nRatio – ratio value set channel (1 – 255)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.23 SRS5330A_GetSRSRatio

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSRatio
(
   int srsNo,
   int nChanNo,
   int *pnRatio
)
```

Function Description:

This function sends the command to get the ratio value for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pnRatio – pointer to location to return the ratio value
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

3.24 SRS5330A_ResetOverCurrent

Format:

```
_SRS5330AFUNC int SRS5330A_ResetOverCurrent
(
   int srsNo,
   int nChanNo

)
```

Function Description:

This function sends the command to return the specified channel to normal state if it was in over-current state.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo – 5330A Channel
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

3.25 SRS5330A_GetOverCurrentStatus

Format:

```
_SRS5330AFUNC int SRS5330A_GetOverCurrentStatus
(
   int srsNo,
   int nChanNo,
   int *pnOverCurrStatus
)
```

Function Description:

This function sends the command to query the specified channel is in over-current state.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo – 5330A Channel
pnOverCurrStatus – pointer to location to return the over-current state
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

4 SRS-5330A Dynamic Motion Routines

The routines in this section handle retrieving and setting dynamic motion information for the channels in the 5330A device.

4.1 SRS5330A_SetSRSDynMotionMode

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionMode
(
   int srsNo,
   int nChanNo,
   int nMode
)
```

Function Description:

```
This function sends the command to set the dynamic motion mode of the
specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
nMode – dynamic motion mode to set channel
                  Mode Types:
                     DYNAMIC_MODE_STATIC         0
                     DYNAMIC_MODE_SINE           1
                     DYNAMIC_MODE_RAMP           2
                     DYNAMIC_MODE_SQUARE         3
                     DYNAMIC_MODE_TRIANGLE       4
                     DYNAMIC_MODE_ROT_CONT       5
                     DYNAMIC_MODE_ROT_STARTSTOP  6
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

```
None.
```

4.2 SRS5330A_GetSRSDynMotionMode

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDynMotionMode
(
   int srsNo,
   int nChanNo,
   int *pnMode
)
```

Function Description:

```
This function sends the command to get the dynamic motion mode of the
specified 5330A channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pnMode – pointer to location to return the channel dynamic motion mode
                  Mode Types:
                     DYNAMIC_MODE_STATIC         0
                     DYNAMIC_MODE_SINE           1
                     DYNAMIC_MODE_RAMP           2
                     DYNAMIC_MODE_SQUARE         3
                     DYNAMIC_MODE_TRIANGLE       4
                     DYNAMIC_MODE_ROT_CONT       5
                     DYNAMIC_MODE_ROT_STARTSTOP  6
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

4.3 SRS5330A_SetSRSDynMotionAmp

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionAmp
(
   int srsNo,
   int nChanNo,
   double dDynMotionAmp
)
```

Function Description:

```
This function sends the command to set the dynamic motion amplitude for the
specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dDynMotionAmp – dynamic motion amplitude to set (Range: 0 to 359.999)
```

Return Value:

```
SRS_SUCCESS - function is successful
```

```
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

4.4 SRS5330A_GetSRSDynMotionAmp

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDynMotionAmp
(
   int srsNo,
   int nChanNo,
   double *pdDynMotionAmp
)
```

Function Description:

This function sends the command to get the dynamic motion amplitude for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdDynMotionAmp – pointer to location to return the dynamic motion amplitude
                 value
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

4.5 SRS5330A_SetSRSDynMotionFreq

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionFreq
(
   int srsNo,
   int nChanNo,
   double dDynMotionFreq
)
```

Function Description:

This function sends the command to set the dynamic motion frequency value for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dDynMotionFreq – dynamic motion frequency to set (Range: 0 to 1000.0 Hz)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

4.6 SRS5330A_GetSRSDynMotionFreq

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDynMotionFreq
(
   int srsNo,
   int nChanNo,
   double *pdDynMotionFreq
)
```

Function Description:

This function sends the command to get the dynamic motion frequency value for the specified 5330A channel.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdDynMotionFreq – pointer to location to return the dynamic motion freq value

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

4.7 SRS5330A_SetSRSDynMotionRotRate

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionRotRate
(
   int srsNo,
   int nChanNo,
   double dRotRate
)
```

Function Description:

This function sends the command to set the rotation rate of the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dRotRate – rotation rate (Range: -100000 to 100000)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

4.8 SRS5330A_GetSRSDynMotionRotRate

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDynMotionRotRate
(
   int srsNo,
   int nChanNo,
   double *pdRotRate
)
```

Function Description:

This function sends the command to get the rotation rate of the specified 5330A channel.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdRotRate – pointer to location to return the rotation rate

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

4.9 SRS5330A_SetSRSDynMotionRotStopAng

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionRotStopAng
(
   int srsNo,
   int nChanNo,
   double dStopAng
)
```

Function Description:

This function sends the command to set the rotation stop angle value for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dStopAng – rotation stop angle (Range: 0 to 359.999)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

4.10 SRS5330A_GetSRSDynMotionRotStopAng

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDynMotionRotStopAng
(
   int srsNo,
   int nChanNo,
   double *pdStopAng
)
```

Function Description:

This function sends the command to get the rotation stop angle value for the specified 5330A channel.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdStopAng – pointer to location to return the rotation stop angle value

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

4.11 SRS5330A_SetSRSDynMotionStartStop

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionStartStop
(
   int srsNo,
   int nChanNo,
   int nStartStop
)
```

Function Description:

This function sends the command to set start/stop the dynamic motion of the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel]
nStartStop – start/stop command to send to the channel
                  Command Types:
                     START          0
                     STOP           1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

4.12 SRS5330A_GetSRSDynMotionStartStop

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDynMotionStartStop
(
   int srsNo,
   int nChanNo,
   bool *pbStopped
)
```

Function Description:

```
This function sends the command to get the dynamic motion state of the
specified 5330A channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pbStopped – pointer to location to return the rotation state
                  State Definitions:
                     true           Stopped
                     false          Rotating/Modulating
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

4.13 *SRS5330A_SetSRSDynMotionAngStepSize*

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionAngStepSize
(
   int srsNo,
   int nChanNo,
   double dAngStpSize
)
```

Function Description:

```
This function sends the command to set the angle step size and is used by the
Increment CCW and Increment CW commands for NAI 5330 Resolver/Synchro
Simulator.

Currently, this function is support only the 5330 Native Legacy Language and
not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote
communication configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo – 5330A Channel
dAngStpSize – angle step size: 0.1<= dAngStpSize <= 359.999
```

Return Value:

```
SRS_SUCCESS - function is successful
```

```
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

4.14 SRS5330A_SetSRSDynMotionIncCClock

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionIncCClock
(
   int srsNo
)
```

Function Description:

This function sends the command to set the increment counter-clockwise command for NAI 5330 Resolver/Synchro Simulator.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

4.15 SRS5330A_SetSRSDynMotionIncClockwise

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDynMotionIncClockwise
(
   int srsNo
)
```

Function Description:

This function sends the command to set the increment clockwise command for NAI 5330 Resolver/Synchro Simulator.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

5 SRS-5330A Output Isolation Relays Routines

5.1 *SRS5330A_SetSRSCloseIsolationRelay*

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSCloseIsolationRelay
(
   int srsNo
)
```

Function Description:

This function sends the command to set the close isolation relays command for NAI 5330 Resolver/Synchro Simulator. The 5330A accepts this command without error but will have no effect on setup of the unit.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it's remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

5.2 *SRS5330A_SetSRSOpenIsolationRelay*

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSOpenIsolationRelay
(
   int srsNo
)
```

Function Description:

This function sends the command to set the open isolation relays command for NAI 5330 Resolver/Synchro Simulator. The 5330A accepts this command without error but will have no effect on setup of the unit.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

6 SRS-5330A Output Remote Sense Routines

6.1 SRS5330A_SetSRSRmtSenseState

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSRmtSenseState
(
   int srsNo,
   int nChanNo,
   int nEnable
)
```

Function Description:

This function sends the command to set the remote sense feature of the selected channel in the SRS-5330A. No signals will be outputted if the remote sense state is ENABLE(REMOTE), but the sense lines are not connected to the load/drive lines.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo – 5330A Channel
nEnable – Enable remote sense.
                     DISABLE (LOCAL)     0
                     ENABLE (REMOTE)     1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

6.2 SRS5330A_GetSRSRmtSenseState

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSRmtSenseState
(
   int srsNo,
   int nChanNo,
   int *pnEnable
)
```

Function Description:

This function sends the command to get the remote sense feature of the selected channel in the SRS-5330A.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nChanNo – 5330A Channel
pnEnable – pointer to location to return remote sense enable value.
                     DISABLE (LOCAL)     0
                     ENABLE (REMOTE)     1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

7 SRS-5330A Reference Generator Remote Sense Routines

7.1 SRS5330A_SetSRSRefGenLocSenseDirection

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSRefGenLocSenseDirection
(
   int srsNo,
   int nDirection
)
```

Function Description:

This function sends the command to set the internal reference generator sense direction of the SRS-5330A. Note, there is no remote sense lines/function when the internal reference output is coming from the front connector.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nDirection – Reference generator sense direction.
                     BACK        0
                     FRONT       1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

7.2 SRS5330A_GetSRSRefGenLocSenseDirection

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSRefGenLocSenseDirection
(
   int srsNo,
   int *pnDirection
)
```

Function Description:

This function sends the command to get the internal reference generator sense direction of the SRS-5330A. Note, there is no remote sense lines/function when the internal reference output is coming from the front connector.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
```

```
pnDirection – pointer to location to return reference generator sense
direction.
                      BACK        0
                      FRONT       1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

7.3 SRS5330A_SetSRSRefGenRmtSenseState

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSRefGenRmtSenseState
(
   int srsNo,
   int nEnable

)
```

Function Description:

This function sends the command to set the internal reference generator sense mode of the SRS-5330A. Note, when the internal reference output is coming from the front connector, the sense state is always in LOCAL mode (Enable=Remote Sense, Disable=Local Sense). When the internal reference remote sense state is set to ENABLE(REMOTE), make sure the sense lines are connected to the intended load; otherwise, the reference output may go into over-current condition.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
nEnable – Enable remote sense.
                      DISABLE (LOCAL)     0
                      ENABLE (REMOTE)     1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

7.4 SRS5330A_GetSRSRefGenRmtSenseState

```
_SRS5330AFUNC int SRS5330A_SetSRSRefGenRmtSenseState
(
   int srsNo,
   int *pnEnable

)
```

Function Description:

This function sends the command to get the internal reference generator sense mode of the SRS-5330A. Note, when the internal reference output is coming from the front connector, the sense state is always in LOCAL mode (Enable=Remote Sense, Disable=Local Sense).

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A. (0-MAX_SRS-1)
pnEnable – pointer to location to return remote sense value.
                     DISABLE (LOCAL)     0
                     ENABLE (REMOTE)     1
```

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

8 SRS-5330A D/A Output Routines

The routines in this section handle retrieving and setting the D/A output configuration for the channels in the 5330A device.

8.1 SRS5330A_SetSRSDAOutput

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDAOutput
(
   int srsNo,
   int nChanNo,
   int nDAOutput
)
```

Function Description:

```
This function sends the command to set the data type to use for DA output for
the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
nDAOutput – data type mode for DA output
                  DA Output Data Types:
                     DA_ANGLE_OUTPUT          0
                     DA_VELOCITY_OUTPUT       1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_RANGE – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

8.2 SRS5330A_GetSRSDAOutput

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDAOutput
(
   int srsNo,
   int nChanNo,
   char *pszDAOutput
)
```

Function Description:

```
This function sends the command to get the data type to use for DA output for
the specified 5330A channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pszDAOutput – pointer to location to return the data type for DA output
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

8.3 SRS5330A_SetSRSDAUpperLimit

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDAUpperLimit
(
   int srsNo,
   int nChanNo,
   double dDAUpperLimit
)
```

Function Description:

```
This function sends the command to set the upper angle or velocity limit
value for DA output for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dDAUpperLimit – upper angle or velocity limit value to set channel for DA
                Output
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

```
None.
```

8.4 SRS5330A_GetSRSDAUpperLimit

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDAUpperLimit
(
   int srsNo,
   int nChanNo,
   double *pdDAUpperLimit
)
```

Function Description:

```
This function sends the command to get the upper angle or velocity limit
value for DA output for the 5330A channel.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdDAUpperLimit – pointer to location to return the upper angle or velocity
   limit for DA output.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

8.5 *SRS5330A_SetSRSDALowerLimit*

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDALowerLimit
(
   int srsNo,
   int nChanNo,
   double dDALowerLimit
)
```

Function Description:

```
This function sends the command to set the lower angle or velocity limit
value for DA output for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dDALowerLimit – lower angle or velocity limit value to set channel for DA
   Output
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
```

```
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

8.6 SRS5330A_GetSRSDALowerLimit

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDALowerLimit
(
   int srsNo,
   int nChanNo,
   double *pdDALowerLimit
)
```

Function Description:

This function sends the command to get the lower angle or velocity limit value for DA output for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdDALowerLimit – pointer to location to return the lower angle or velocity
                 limit for DA output.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

8.7 SRS5330A_SetSRSDAUpperVoltage

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDAUpperVoltage
(
   int srsNo,
   int nChanNo,
   double dDAUpperVoltage
)
```

Function Description:

This function sends the command to set the voltage value associated with the upper limit value for DA output for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dDAUpperVoltage – voltage value associated with upper limit value to set
                  channel for DA Output
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

8.8 SRS5330A_GetSRSDAUpperVoltage

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDAUpperVoltage
(
   int srsNo,
   int nChanNo,
   double *pdDAUpperVoltage
)
```

Function Description:

This function sends the command to get the voltage value associated with the upper limit value for DA output for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdDAUpperVoltage – pointer to location to return the voltage value associated
                   with upper limit value for DA output.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

8.9 SRS5330A_SetSRSDALowerVoltage

Format:

```
_SRS5330AFUNC int SRS5330A_SetSRSDALowerVoltage
(
   int srsNo,
   int nChanNo,
   double dDALowerVoltage
)
```

Function Description:

This function sends the command to set the voltage value associated with the lower limit value for DA output for the specified 5330A channel.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
dDALowerVoltage – voltage value associated with lower limit value to set
                  channel for DA Output
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

8.10 SRS5330A_GetSRSDALowerVoltage

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSDALowerVoltage
(
   int srsNo,
   int nChanNo,
   double *pdDALowerVoltage
)
```

Function Description:

This function sends the command to get the voltage value associated with the lower limit value for DA output for the specified 5330A channel.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nChanNo – 5330A Channel
pdDALowerVoltage – pointer to location to return the voltage value associated
                   with lower limit value for DA output.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

9 SRS-5330A Binary Coded Decimal (BCD) Routines

9.1 SRS5330A_SetBCDMode

Format:

```
_SRS5330AFUNC int SRS5330A_SetBCDMode
(
   int srsNo,
   int nBcdMode
)
```

Function Description:

```
This function sends the command to set the BCD mode (BCD=0, BIN=1).
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nBcdMode – data type mode for BCD
                  BCD MODE Types:
                     BCD_MODE  0
                     BIN_MODE  1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

9.2 SRS5330A_GetBCDMode

Format:

```
_SRS5330AFUNC int SRS5330A_GetBCDMode
(
   int srsNo,
   int *nBcdMode
)
```

Function Description:

```
This function sends the command to get the BCD mode (BCD=0, BIN=1).
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nBcdMode – pointer to location to return the current BCD mode.
                  BCD MODE Types:
                     BCD_MODE  0
                     BIN_MODE  1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

10 SRS-5330A Internal Reference Routines

The routines in this section handle retrieving and setting configurations for the reference module in the 5330A device if available.

10.1 SRS5330A_SetIntRefFreq

Format:

```
_SRS5330AFUNC int SRS5330A_SetIntRefFreq
(
   int srsNo,
   double dFreq
)
```

Function Description:

```
This function sends the command to set the reference frequency for the
reference module (if available).

Note the 5330A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
dFreq – Frequency value to set the reference module
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

```
None.
```

10.2 SRS5330A_GetIntRefFreq

Format:

```
_SRS5330AFUNC int SRS5330A_GetIntRefFreq
(
   int srsNo,
   double *pdFreq
)
```

Function Description:

```
This function sends the command to get the reference frequency for the
reference module (if available).
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
```

```
pdFreq – pointer to location to return the frequency value to set the
               reference module
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

Remarks:

If the 5330A unit does not contain a reference module, the device will always return 400 Hz for the frequency value.

10.3 SRS5330A_SetIntRefVolt

Format:

```
_SRS5330AFUNC int SRS5330A_SetIntRefVolt
(
   int srsNo,
   double dVolt
)
```

Function Description:

This function sends the command to set the reference voltage for the reference module (if available).

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
dVolt – Voltage value to set the reference module
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

10.4 SRS5330A_GetIntRefVolt

Format:

```
_SRS5330AFUNC int SRS5330A_GetIntRefVolt
```

```
(
   int srsNo,
   double *pdVolt
)
```

Function Description:

This function sends the command to get the reference voltage for the reference module (if available).

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pdVolt – pointer to location to return the voltage value to set the
             reference module
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

Remarks:

If the 5330A unit does not contain a reference module, the device will always return 26 volts for the voltage value.

10.5 SRS5330A_SetIntRefOutputState

Format:

```
_SRS5330AFUNC int SRS5330A_SetIntRefOutputState
(
   int srsNo,
   int nOutputState
)
```

Function Description:

This function sends the command to set the reference output state for the reference module (if available).

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nOutputState – Output state value to set the reference module
                  Reference Output States:
                     INT_REF_OUT_NOT_AVAILABLE     0
                     INT_REF_OUT_AVAILABLE         1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

10.6 SRS5330A_GetIntRefOutputState

Format:

```
_SRS5330AFUNC int SRS5330A_GetIntRefOutputState
(
   int srsNo,
   int *pnOutputState
)
```

Function Description:

This function sends the command to get the reference output state for the reference module (if available).

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pnOutputState – pointer to location to return the reference output state
   to set the reference module
                  Reference Output States:
                     INT_REF_OUT_NOT_AVAILABLE     0
                     INT_REF_OUT_AVAILABLE         1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

Remarks:

If the 5330A unit does not contain a reference module, the device will always return INT_REF_OUT_NOT_AVAILABLE.

10.7 SRS5330A_GetIntRefOverCurState

Format:

```
_SRS5330AFUNC int SRS5330A_GetIntRefOverCurState
(
   int srsNo,
   int *pnOvrCurState
)
```

Function Description:

This function sends the command to get the internal reference over current state.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pnOvrCurState – pointer to location to return the internal reference over
current state.
                  Int. Ref. Over Current States:
                     NO_OVER_CURRENT    0
                     OVER_CURRENT       1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

10.8 SRS5330A_ResetIntRefOverCur

Format:

```
_SRS5330AFUNC int SRS5330A_ResetIntRefOverCur
(
   int srsNo,
   int nResetOvrCur
)
```

Function Description:

This function sends the command to reset the internal reference state if it was in over-current state.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nResetOvrCur – Output state value to set the reference module
                  Reference Output States:
                     INT_REF_NO_RESET_OVR_CUR    0
                     INT_REF_RESET_OVR_CUR       1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_RANGE - – parameter specified is out-of-range
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

11 SRS-5330A Command Routines

The routines in this section handle sending commands such as retrieval of the device ID and errors on the error queue, and resetting the 5330A device setting to factory default settings.

11.1 SRS5330A_PerformGetID

Format:

```
_SRS5330AFUNC int SRS5330A_PerformGetID
(
   int srsNo,
   char *pszID
)
```

Function Description:

```
This function sends the IDN command to get Device ID string for the
device. The ID returned includes the manufacturer (NORTH ATLANTIC), the
5330A module, serial number, and revision information.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pszID – pointer to location to return the device ID
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

11.2 SRS5330A_Reset

Format:

```
_SRS5330AFUNC int SRS5330A_Reset
(
   int srsNo,
   char *pszResults
)
```

Function Description:

```
This function sends the command to reset the 5330A device and set the
device setting back to the factory default settings.

Note the 5330A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pszID – pointer to location to return the results of the reset command.
```

```
               Return results:
                  "Reset Complete" – reset has been successful.
                  "Reset Not Performed" – reset has not been successful.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

11.3 SRS5330A_GetErrors

Format:

```
_SRS5330AFUNC int SRS5330A_GetErrors
(
   int srsNo,
   char *pszErrors
)
```

Function Description:

This function sends the ERR command to get error from the error queue for the device. "No error" is returned when there are no errors on the queue.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pszErrors – pointer to location to return the error string
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

11.4 SRS5330A_SetInternalSelfTest

Format:

```
_SRS5330AFUNC int SRS5330A_SetInternalSelfTest
(
   int srsNo

)
```

Function Description:

This function sends the command to perform an internal self test on the 5330A. The 5330A accepts this command without error but will have no effect on setup of the unit.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

11.5 SRS5330A_SetConfidenceTest

Format:

```
_SRS5330AFUNC int SRS5330A_SetConfidenceTest
(
   int srsNo
)
```

Function Description:

This function sends the command to perform a confidence test on the 5330A. The 5330A accepts this command without error but will have no effect on setup of the unit.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5330_NATIVE

```
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

11.6 SRS5330A_SetCombCmdBcdLegacy

Format:

```
_SRS5330AFUNC int SRS5330A_SetCombCmdBcdLegacy
(
   int srsNo,
   double dAngle,
   int nSigMode,
   double dRefVolt,
   double dVll

)
```

Function Description:

This function sends a combined command to set the angle, output mode(RSL/SYN), ref input(115/26V) and output voltage(90/26/11.8V) to 5330A in 5310 BCD language.

Currently, this function is support only the 5310 BCD Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
dAngle – angle value to set (Unipolar Range: 0 to 359.999)
nSigMode - signal mode to set channel. Resolver = 0, SYNCHRO = 1.
dRefVolt - 115V, 26V.
dVll - 90V, 26V, 11.8V.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with IEEE_5310_BCD
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12 SRS-5330A Configuration Routines

The routines in this section handle sending commands to set and retrieve the configuration settings of the 5330A device.

12.1 SRS5330A_GetIEEELang

Format:

```
_SRS5330AFUNC int SRS5330A_GetIEEELang
(
   int srsNo,
   char *pszIEEELang
)
```

Function Description:

```
This function sends the command to get the IEEE language protocol set in
the 5330A.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pszIEEELang – pointer to location to return the IEEE protocol string
   Return values:
      "5330ANATIVE"   – SRS-5330A Native
      "5330NATIVE"    - SRS-5330 Native (Legacy)
      "5310NATIVEBCD" - SRS-5310 BCD (Legacy)
      "5310NATIVEBIN" – SRS-5310 Binary (Legacy)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

12.2 SRS5330A_SetIEEELang

Format:

```
_SRS5330AFUNC int SRS5330A_SetIEEELang
(
   int srsNo,
   int nIEEELang
)
```

Function Description:

```
This function sends the command to set the IEEE protocol language to
accept when communicating via IEEE.
```

```
Note the 5330A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nIEEELang - Language Protocol to be used to communicate via IEEE to 5330A.
                  5330A Language Types:
                     SRS_5330A_NATIVE           0
                     IEEE_5330_NATIVE            1
                     IEEE_5310_BCD               2
                     IEEE_5310_BINARY            3
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

```
None.
```

12.3 SRS5330A_GetCommState

Format:

```
_SRS5330AFUNC int SRS5330A_GetCommState
(
   int srsNo,
   char *pszCommState
)
```

Function Description:

```
This function sends the command to get the communication mode set in the
5330A.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pszCommState – pointer to location to return the communication mode string
   Return values:
         “Local Mode”
         “Remote IEEE Addr:  SRS-IEEE Language”
         “Remote USB”
         “Remote Ethernet”
         “Remote J1”
         “Remote with Lockout via IEEE Addr:  SRS-IEEE Language”
         “Remote with Lockout via USB”
         “Remote with Lockout via Ethernet”
         “Remote with Lockout via J1”
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
```

```
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

12.4 SRS5330A_GoToLocal

Format:

```
_SRS5330AFUNC int SRS5330A_GoToLocal
(
   int srsNo
)
```

Function Description:

This function sends the command to set the communication mode to Local mode. In Local mode, remote “set” commands will not be accepted.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.5 SRS5330A_SetLocalLockout

Format:

```
_SRS5330AFUNC int SRS5330A_SetLocalLockout
(
   int srsNo
)
```

Function Description:

This function sends the command to lockout configuration setting such as signal mode, reference mode, and ratio setting from the 5330A front panel.

Note the 5330A device will not accept the command if device is set to Local mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.6 SRS5330A_SetRemoteUSB

Format:

```
_SRS5330AFUNC int SRS5330A_SetRemoteUSB
(
   int srsNo
)
```

Function Description:

This function sends the command to set the communication mode to Remote USB mode. In Remote USB mode, remote “set” commands will be accepted if the command is received from the USB interface.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.7 SRS5330A_SetRemoteEthernet

Format:

```
_SRS5330AFUNC int SRS5330A_SetRemoteEthernet
(
   int srsNo
)
```

Function Description:

This function sends the command to set the communication mode to Remote Ethernet mode. In Remote Ethernet mode, remote “set” commands will be accepted if the command is received from the Ethernet interface.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.8 SRS5330A_SetRemoteIEEE

Format:

```
_SRS5330AFUNC int SRS5330A_SetRemoteIEEE
(
   int srsNo
)
```

Function Description:

This function sends the command to set the communication mode to Remote IEEE mode. In Remote IEEE mode, remote "set" commands will be accepted if the command is received from the IEEE interface.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.9 SRS5330A_SetRemoteJ1

Format:

```
_SRS5330AFUNC int SRS5330A_SetRemoteJ1
(
   int srsNo
)
```

Function Description:

This function sends the command to set the communication mode to Remote J1 mode. In Remote J1 mode, remote "set" commands will be accepted if the command is received from the J1 interface.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.10 SRS5330A_SetAngleFormat

Format:

```
_SRS5330AFUNC int SRS5330A_SetAngleFormat
(
   int srsNo,
   int nFormat
)
```

Function Description:

This function sends the command to set the angle format.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nFormat – angle format on the 5330A
                  Angle Format Types:
                     ANGLE_FMT_360          0
                     ANGLE_FMT_180          1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported with SRS_5330A_NATIVE,
   IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.11 SRS5330A_GetAngleFormat

Format:

```
_SRS5330AFUNC int SRS5330A_GetAngleFormat
(
   int srsNo,
   int *pnFormat
)
```

Function Description:

This function sends the command to get the angle format set in the 5330A.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pnFormat – pointer to location to return the angle format
                  Angle Format Types:
                     ANGLE_FMT_360          0
                     ANGLE_FMT_180          1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

12.12 SRS5330A_SetCh1Output

Format:

```
_SRS5330AFUNC int SRS5330A_SetCh1Output
(
   int srsNo,
   int nCh1OutputConnector
)
```

Function Description:

This function sends the command to set the Channel 1 output connector configuration.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nCh1OutputConnector – Channel 1 output connector configuration on the
                       5330A
                  Channel 1 Output Configuration Types:
                     CH1_OUTPUT_FRONT_CONNECTOR     0
                     CH1_OUTPUT_BACK_CONNECTOR      1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.13 SRS5330A_GetCh1Output

Format:

```
_SRS5330AFUNC int SRS5330A_GetCh1Output
(
   int srsNo,
   int *pnCh1OutputConnector
)
```

Function Description:

```
This function sends the command to get the Channel 1 output connector
configuration set in the 5330A.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pnCh1OutputConnector – pointer to location to return the Channel 1 output
                      connector configuration
                  Channel 1 Output Configuration Types:
                     CH1_OUTPUT_FRONT_CONNECTOR     0
                     CH1_OUTPUT_BACK_CONNECTOR      1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

12.14 SRS5330A_SetTouchscreenState

Format:

```
_SRS5330AFUNC int SRS5330A_SetTouchscreenState
(
   int srsNo,
   int nTouchscreenState
)
```

Function Description:

```
This function sends the command to set the touch screen configuration.

Note the 5330A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
nTouchscreenState – Touch screen configuration on the 5330A
                  Touch screen Configuration Types:
                     TOUCHSCREEN_DISABLED     0
```

```
                              TOUCHSCREEN_ENABLED      1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

12.15SRS5330A_GetTouchscreenState

Format:

```
_SRS5330AFUNC int SRS5330A_GetTouchscreenState
(
   int srsNo,
   int *pnTouchscreenState
)
```

Function Description:

This function sends the command to get the touch screen configuration set in the 5330A.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pnTouchscreenState – pointer to location to return the touch screen
   configuration
                  Touchscreen Configuration Types:
                     TOUCHSCREEN_DISABLED     0
                     TOUCHSCREEN_ENABLED      1
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

None.

12.16SRS5330A_ResetDefaultValues

Format:

```
_SRS5330AFUNC int SRS5330A_ResetDefaultValues
(
   int srsNo
)
```

Function Description:

This function sends the command to set the device setting back to the factory default settings.

Note the 5330A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

13 SRS-5330A Calibration Routines

The routines in this section handle sending commands to calibrate the 5330A device and retrieve the calibration state of the 5330A device.

13.1 SRS5330A_GetCalState

Format:

```
_SRS5330AFUNC int SRS5330A_GetCalState
(
   int srsNo,
   char *pszCalState
)
```

Function Description:

```
This function sends the command to get the calibration state of the 5330A.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pszCalState – pointer to location to return the calibration state string
   Return values:
         "CAL DONE"
         "CALIBRATING"
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent
```

References for this function:

```
None.
```

13.2 SRS5330A_Calibrate

Format:

```
_SRS5330AFUNC int SRS5330A_Calibrate
(
   int srsNo
)
```

Function Description:

```
This function sends the command to calibrate the 5330A.

Note the 5330A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
         SRS5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

13.3 SRS5330A_SetClrCalibrate

Format:

```
_SRS5330AFUNC int SRS5330A_SetClrCalibrate
(
   int srsNo,
   double dCal

)
```

Function Description:

This function sends the command to clear all calibration the 5330A. The 5330A accepts this command without error but will have no effect on setup of the unit.

Currently, this function is support only the 5330 Native Legacy Language and not supported in 5330A Native Language.

Note the 5330A device will not accept the command if it’s remote communication configuration does not match the communication connection mode.

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
dCal – value to pass to the CALZ command.
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with
IEEE_5330_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
```

References for this function:

None.

13.4 SRS5330A_GetSRSBackGroundCalState

Format:

```
_SRS5330AFUNC int SRS5330A_GetSRSBackGroundCalState
(
   int srsNo,
   int *pnBackGroundCalState
)
```

Function Description:

This function sends the command to get the back-ground calibration state. The back-ground calibration state changes when the following occurred:

1. Internal reference frequency changes.
2. Internal reference voltage changes.
3. Channel output mode(RSL/SYN) changes.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
pnBackGroundCalState – pointer to location to return the back-ground calibration state.

Periodic Calibration Types:

BACK_GND_NOT_CALING	0
BACK_GND_CHAN1_CALING	1
BACK_GND_CHAN2_CALING	2
BACK_GND_CH1_AND_CH2_CALING	3

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_FUNC_NOT_SUPPORTED – function supported only with SRS_5330A_NATIVE
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

14 SRS-5330A Miscellaneous Routines

The routines in this section handle setting or retrieving information from the SRS5330ADll and sending freeform commands and queries to the 5330A device.

14.1 SRS5330A_MaxRetry

Format:

```
_SRS5330AFUNC int SRS5330A_MaxRetry
(
   int nMaxRetry
)
```

Function Description:

```
This function sets the maximum retries to send a command or read a
response that will be made when communicating via IEEE. The default value
is 0.
```

Parameters:

```
nMaxRetry – maximum retries for IEEE communication
```

Return Value:

```
SRS_SUCCESS - function is successful
```

References for this function:

```
None.
```

14.2 SRS5330A_LastCmdSent

Format:

```
_SRS5330AFUNC int SRS5330A_LastCmdSent
(
   int srsNo,
   char szLastCommand[]
)
```

Function Description:

```
This function returns the last command sent via IEEE, USB or Ethernet to
the 5330A device.
```

Parameters:

```
srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
szLastCommand – last command sent to 5330A
```

Return Value:

```
SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
```

References for this function:

```
None.
```

14.3 SRS5330A_WriteCommand

Format:

```
_SRS5330AFUNC int SRS5330A_WriteCommand
(
   int srsNo,
   char szCommand[]
)
```

Function Description:

This function sends the command to the 5330A device.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
szCommand – command to send to 5330A

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE - unable to send command to 5330A

References for this function:

None.

14.4 SRS5330A_QueryCommand

Format:

```
_SRS5330AFUNC int SRS5330A_QueryCommand
(
   int srsNo,
   char szCommand[],
   char *pszResponse
)
```

Function Description:

This function sends the command to the 5330A device and waits for the 5330A to respond.

Parameters:

srsNo - Logical SRS number assigned to connection with 5330A.(0-MAX_SRS-1)
szCommand – command to send to 5330A
pszResponse – pointer to location to return the 5330A response to the command sent.

Return Value:

SRS_SUCCESS - function is successful
SRS_ERROR_SRSNO - invalid srsNo parameter
SRS_ERROR_WRITE - unable to send command to 5330A
SRS_ERROR_DATA – data returned from 5330A is not valid for command sent

References for this function:

None.

15 Appendix A – SRS5330ADll Constant Values

```
/* Maximum number of SRSs Driver can communicate with */
#define  MAX_SRS                    12

/* SRS Language Types */
#define SRS_5330A_NATIVE            0
#define IEEE_5330_NATIVE            1
#define IEEE_5310_BCD               2
#define IEEE_5310_BINARY            3

/* SRS Communication Type */
#define NO_CONNECTION               0
#define IEEE_CONNECTION             1
#define USB_CONNECTION              2
#define ETHERNET_CONNECTION         3

/* Mode Type */
#define RESOLVER                    0
#define SYNCHRO                     1

/* Dynamic Mode Types */
#define DYNAMIC_MODE_STATIC         0
#define DYNAMIC_MODE_SINE           1
#define DYNAMIC_MODE_RAMP           2
#define DYNAMIC_MODE_SQUARE         3
#define DYNAMIC_MODE_TRIANGLE       4
#define DYNAMIC_MODE_ROT_CONT       5
#define DYNAMIC_MODE_ROT_STARTSTOP  6

/* Reference Mode Type */
#define FIXED                       0
#define RATIO                       1

/* Reference Source Type */
#define INTERNAL                    0
#define EXTERNAL                    1

#define EXTERNAL_BACK               0
#define EXTERNAL_FRONT              1

/* Remote Sense Enable */
#define RMT_SENSE_DISABLE           0
#define RMT_SENSE_ENABLE            1

/* Reference Generator Dir */
#define REF_GEN_DIR_BACK            0
#define REF_GEN_DIR_FRONT           1

/* Remote Sense Enable */
#define NO_OVER_CURRENT             0
#define OVER_CURRENT                1


/* Internal Reference Output State Type */
```

```
#define INT_REF_OUT_NOT_AVAILABLE   0
#define INT_REF_OUT_AVAILABLE       1

/* Internal Reference Reset Over Current */
#define INT_REF_NO_RESET_OVR_CUR    0
#define INT_REF_RESET_OVR_CUR       1


/* Angle Display Format Type */
#define ANGLE_FMT_360               0
#define ANGLE_FMT_180               1

/* DA Format Type */
#define DA_ANGLE_OUTPUT             0
#define DA_VELOCITY_OUTPUT          1

/* DA Format Type */
#define BCD_MODE                    0
#define BIN_MODE                    1

/* Channel 1 Output Type */
#define CH1_OUTPUT_FRONT_CONNECTOR  0
#define CH1_OUTPUT_BACK_CONNECTOR   1

/* Touchscreen Enable State Type */
#define TOUCHSCREEN_DISABLED        0
#define TOUCHSCREEN_ENABLED         1

/* Periodic Calibration State Type */
#define PERIODIC_CAL_DISABLED       0
#define PERIODIC_CAL_ENABLED        1

/* If output is Rotating/Modulating at this time */
#define CHANGING                    0
#define STOPPED                     1

/* Parameters for Stopping or Starting Rotation/Modulation */
#define START                       0
#define STOP                        1

/* Back-Ground Calibration State Type */
#define BACK_GND_NOT_CALING         0
#define BACK_GND_CHAN1_CALING       1
#define BACK_GND_CHAN2_CALING       2
#define BACK_GND_CH1_AND_CH2_CALING 3

#define TOTAL_BIN_5310BINARY        0x3FFFF
```

16 Appendix B- Error Codes

Error Mnemonic	Value	Meaning
SRS_SUCCESS	0	Function is successful
SRS_ERROR_OPEN_SRS_SESSION	1	IEEE connection or configuration error
SRS_ERROR_SRSNO	2	Invalid srsNo parameter
SRS_ERROR_ADDRS	3	Invalid IEEE Address parameter
SRS_ERROR_LANG	4	Invalid 5330A Language parameter
SRS_ERROR_DATA	5	Data returned from 5330A is not valid for command sent
SRS_ERROR_RANGE	6	Parameter specified is out-of-range
SRS_ERROR_WRITE	7	Unable to send command to 5330A
SRS_ERROR_USB_CONNECTION	8	USB connection error
SRS_ERROR_ETHER_CONNECTION	9	Ethernet connection error
SRS_ERROR_FUNC_NOT_SUPPORTED	10	Function not support in selected language for 5330A communication

Revision History

Revision ID	Revision Date	Description	Author
1.0.0.0	Feb 5, 2010	Initial Release	Gc
2.0.0.100	Feb 24, 2011	Added the following routines to the document: **SRS-5330A Channel Routines** SRS5330A_SetSRSExtRefSrcLoc SRS5330A_GetSRSExtRefSrcLoc SRS5330A_ResetOverCurrent SRS5330A_GetOverCurrentStatus **SRS-5330A Dynamic Motion Routines** SRS5330A_SetSRSDynMotionAngStepSize SRS5330A_SetSRSDynMotionIncCClock SRS5330A_SetSRSDynMotionIncClockwise **SRS-5330A Output Isolation Relays Routines** SRS5330A_SetSRSCloseIsolationRelay SRS5330A_SetSRSOpenIsolationRelay **SRS-5330A Output Remote Sense Routines** SRS5330A_SetSRSRmtSenseState SRS5330A_GetSRSRmtSenseState **SRS-5330A Reference Generator Remote Sense Routines** SRS5330A_SetSRSRefGenLocSenseDirection SRS5330A_GetSRSRefGenLocSenseDirection SRS5330A_SetSRSRefGenRmtSenseState SRS5330A_GetSRSRefGenRmtSenseState **SRS-5330A Binary Coded Decimal (BCD) Routines** SRS5330A_SetBCDMode SRS5330A_GetBCDMode **SRS-5330A Internal Reference Routines** SRS5330A_GetIntRefOverCurState SRS5330A_ResetIntRefOverCur **SRS-5330A Command Routines** SRS5330A_SetInternalSelfTest SRS5330A_SetConfidenceTest SRS5330A_SetCombCmdBcdLegacy **SRS-5330A Calibration Routines** SRS5330A_SetClrCalibrate SRS5330A_GetSRSBackGroundCalState	Gc
3.0.0.1	Oct 8, 2012	No changes to API. Updated document revision to correspond to SSK release providing Cypress USB Driver for Windows 7.	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	gc

		Cypress USB Driver.	