Function Reference Manual for 8810A

Table of Contents

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 8810A.*

For information about general programming information please refer to the *Programmer's Reference Guide for 8810A.*

Reference CD

For electronic copies of the 8810A documentation, API-8810A Soft Panel application program, and source code for API-8810ADll and Soft Panel application please refer to the 8810A Product CD.

API-8810A Connect/Disconnect Routines	
API8810A_ConnectViaIEEE	Sets up and opens the connection to communicate to the 8810A via IEEE. The following IEEE languages are supported: • API-8810A Native • API-8810 Native (Legacy) • API-8810 SR103 (Legacy) • API-8810 HSR202 (Legacy) • API-8810 HSR203 (Legacy) • API-8810 Native FX2 (Legacy)
API8810A_ConnectViaUSB	Sets up and opens the connection to communicate to the 8810A via USB.
API8810A_ConnectViaEthernet	Sets up and opens the connection to communicate to the 8810A via Ethernet.
API8810A_DisconnectIEEE	Closes the IEEE connection.
API8810A_DisconnectUSB	Closes the USB connection.
API8810A_DisconnectEthernet	Closes the Ethernet connection.
API8810A_GetAPI8810AUSBDeviceCnt	Scan the USB Ports for the number of Cypress USB Devices. Note, after calling this function, call *API8810A_GetAPI8810ADeviceIDN()* routine to determine the device numbers for 8810A devices.
API8810A_GetAPI8810ADeviceIDN	Performs an IDN query on the Cypress USB Device and determines if the device being queried is an 8810A device. If it is, the IDN response which includes manufacture, model, and serial number is returned.
API-8810A Channel Routines	
API8810A_SetAPITrackHold	Sets the channel's track/latch (hold) state.
API8810A_GetAPITrackHold	Gets the channel's track/latch (hold) state.
API8810A_SetAPISignalMode	Sets the channel's signal mode (SYN/RSL).
API8810A_GetAPISignalMode	Gets the channel's signal mode.
API8810A_SetAPIReferenceSrc	Sets the channel's reference source mode (INT/EXT).
API8810A_GetAPIReferenceSrc	Gets the channel's reference source mode.
API8810A_SetAPIRatio	Sets the channel's ratio value.
API8810A_GetAPIRatio	Gets the channel's ratio value.
API8810A_SetAPIAutoBandwidth	Sets the channel's bandwidth to "auto" mode.
API8810A_SetAPIBandwidth	Sets the channel's bandwidth to "override" mode and set the channel's bandwidth value.
API8810A_GetAPIBandwidth	Gets the channel's bandwidth mode and bandwidth value.
API8810A_GetAPIAngle	Gets the channel's angle value.
API8810A_GetAPIAvgAngle	Gets the channel's angle average value.
API8810A_GetAPIVelocity	Gets the channel's angle velocity value.
API8810A_GetAPILineLineVolt	Gets the channel's line-to-line voltage value.
API8810A_GetAPINullVolt	Gets the channel's null voltage value.
API8810A_GetAPIRefVolt	Gets the channel's reference voltage value.

API8810A_GetAPIRefFreq	Gets the channel's reference frequency value.
API8810A_SetAPIAvgState	Sets the channel's angle averaging mode.
API8810A_GetAPIAvgState	Gets the channel's angle averaging mode.
API8810A_SetAPIAvgRate	Sets the channel's angle averaging rate.
API8810A_GetAPIAvgRate	Gets the channel's angle averaging rate.
API8810A_SetAPIAngLimitState	Sets the channel's angle limit testing mode.
API8810A_GetAPIAngLimitState	Gets the channel's angle limit testing mode.
API8810A_SetAPIAngLimitCompare	Sets the channel's angle limit testing comparison mode.
API8810A_GetAPIAngLimitCompare	Gets the channel's angle limit testing comparison mode.
API8810A_SetAPIAngUpperLimit	Sets the channel's upper angle limit value for angle limit testing.
API8810A_GetAPIAngUpperLimit	Gets the channel's upper angle limit value for angle limit testing.
API8810A_SetAPIAngLowerLimit	Sets the channel's lower angle limit value for angle limit testing.
API8810A_GetAPIAngLowerLimit	Gets the channel's lower angle limit value for angle limit testing.
API8810A_SetAPIAngLimitErrorStep	Sets the channel's angle step value for angle error comparison for angle limit testing.
API8810A_GetAPIAngLimitErrorStep	Gets the channel's angle step value for angle error comparison for angle limit testing.
API8810A_SetAPIDAOutput	Sets channel's data type (angle or velocity) to use for DA output.
API8810A_GetAPIDAOutput	Gets channel's data type (angle or velocity) to use for DA output.
API8810A_SetAPIDAUpperLimit	Sets the channel's upper angle or velocity limit value for DA output.
API8810A_GetAPIDAUpperLimit	Gets the channel's upper angle or velocity limit value for DA output.
API8810A_SetAPIDALowerLimit	Sets the channel's lower angle or velocity limit value for DA output.
API8810A_GetAPIDALowerLimit	Gets the channel's lower angle or velocity limit value for DA output.
API8810A_SetAPIDAUpperVoltage	Sets the channel's voltage value associated with the upper limit value for DA output.
API8810A_GetAPIDAUpperVoltage	Gets the channel's voltage value associated with the upper limit value for DA output
API8810A_SetAPIDALowerVoltage	Sets the channel's voltage value associated with the lower limit value for DA output.
API8810A_GetAPIDALowerVoltage	Gets the channel's voltage value associated with the lower limit value for DA output
API8810A_SetAPIDisplayAngDiffState	Sets the channel's angle difference display mode.
API8810A_GetAPIDisplayAngDiffState	Gets the channel's angle difference display mode.
API8810A_GetAPIAngleDiff	Gets the angle difference value between Channel 1 and Channel 2 input signal.

API-8810A Multiple Channel Query Routines	
API8810A_GetAPIAngles	Get both channels' angle values.
API8810A_GetAPIAvgStates	Get both channels' angle averaging modes and average rates.
API8810A_GetAPIAvgAngles	Get both channels' angle average values.
API8810A_GetAPIBandwidths	Get both channels' bandwidth mode and bandwidth values.
API8810A_GetAPISignalModes	Get both channels' signal modes.
API8810A_GetAPIRatios	Get both channels' ratio values.
API8810A_GetAPIReferenceSrcs	Get both channels' reference source modes.
API8810A_GetAPITrackHolds	Get both channels' track or hold states.
API8810A_GetAPIAngLimitStates	Get both channels' angle limit testing states and comparison modes.
API8810A_GetAPIVelocities	Get both channels' velocity values.
API8810A_GetAPILineLineVolts	Get both channels' line-to-line voltage values.
API8810A_GetAPINullVolts	Get both channels' null voltage values.
API8810A_GetAPIRefVolts	Get both channels' reference voltage values.
API8810A_GetAPIRefFreqs	Get both channels' reference frequency values.
API-8810A Internal Reference Routines	
API8810A_SetIntRefFreq	Sets the internal reference frequency value.
API8810A_GetIntRefFreq	Gets the internal reference frequency value.
API8810A_SetIntRefVolt	Sets the internal reference voltage value.
API8810A_GetIntRefVolt	Gets the internal reference voltage value.
API8810A_SetIntRefOutputState	Sets the internal reference output state.
API8810A_GetIntRefOutputState	Gets the internal reference output state.
API8810A_GetIntRefOverCurrentState	Gets the internal reference over current state.
API8810A_ResetIntRefOverCurrent	Resets the internal reference over current state.
API-8810A Command Routines	
API8810A_PerformGetID	Gets the Device ID.
API8810A_Reset	Resets the device.
API8810A_GetErrors	Gets the error message from the error queue.
API-8810A Configuration Routines	
API8810A_GetIEEELang	Gets the IEEE Language protocol configured in the 8810A.
API8810A_SetIEEELang	Sets the IEEE Language protocol to accept in the 8810A.
API8810A_GetCommState	Gets the communication settings.
API8810A_GoToLocal	Sets the device to Local mode.
API8810A_SetLocalLockout	Sets the device to Local Lockout mode.
API8810A_SetRemoteUSB	Sets the device to Remote USB mode.
API8810A_SetRemoteEthernet	Sets the device to Remote Ethernet mode.
API8810A_SetRemoteIEEE	Sets the device to Remote IEEE mode.
API8810A_SetRemoteJ1	Sets the device to Remote J1 mode.
API8810A_SetAngleDisplayFormat	Sets the Angle Display Format.
API8810A_GetAngleDisplayFormat	Gets the Angle Display Format

API8810A_SetCh1Input	Sets the Channel 1 Input Connector configuration.
API8810A_GetCh1Input	Gets the Channel 1 Input Connector configuration.
API8810A_SetTouchscreenState	Sets the Touch screen mode.
API8810A_GetTouchscreenState	Gets the Touch screen mode.
API8810A_SetDisplayState	Sets the display to show on the device.
API8810A_GetDisplayState	Gets the display shown on the device.
API8810A_ResetDefaultValues	Sets the device to the default factory settings.
API-8810A Calibration Routines	
API8810A_GetCalState	Gets the calibration state.
API8810A_Calibrate	Calibrates the 8810A.
API8810A_SetAPIPeriodicCalState	Enables or disables the periodic calibration.
API8810A_GetAPIPeriodicCalState	Gets the periodic calibration state.
API-8810A Buffer Routines	
API8810A_GetSampleRate	Gets the sample rate for data buffering.
API8810A_SetSampleRate	Sets the sample rate for data buffering.
API8810A_GetSampleType	Gets the data type to sample for data buffering.
API8810A_SetSampleType	Sets the data type to sample for data buffering.
API8810A_GetPlotChan	Gets the channels to plot on the 8810A chart.
API8810A_SetPlotChan	Sets the channels to plot on the 8810A chart.
API8810A_GetAngleErrStep	Gets the angle step value for angle error comparison for data buffering.
API8810A_SetAngleErrStep	Sets the angle step value for angle error comparison for data buffering.
API8810A_GetLowerRange	Gets the expected lower range value for plotting on the 8810A chart.
API8810A_SetLowerRange	Sets the expected lower range value for plotting on the 8810A chart.
API8810A_GetUpperRange	Gets the expected upper range value for plotting on the 8810A chart.
API8810A_SetUpperRange	Sets the expected upper range value for plotting on the 8810A chart.
API8810A_GetRecordingState	Gets the recording state.
API8810A_SetRecordingState	Sets the recording state.
API8810A_GetBufferCnt	Gets the number of elements in the 8810A data buffer.
API8810A_GetBufferData	Gets the data elements in the 8810A data buffer. Data retrieval from the data buffer is available only via the USB or Ethernet interface.
API-8810A Miscellaneous Routines	
API8810A_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.
API8810A_LastCmdSent	Returns the last command set to the device by Dll.
API8810A_WriteCommand	Sends the freeform command to the 8810A.
API8810A_QueryCommand	Sends the freeform command to the 8810A and waits for a response.

2 API-8810A Connect/Disconnect Routines

The routines in this section handle IEEE, Ethernet and USB communications to the 8810A device.

2.1 API8810A_ConnectViaIEEE

Format:

```
_API8810AFUNC int API8810A_ConnectViaIEEE
(
   int apiNo,
   int nIEEEAddr,
   int nIEEELang
)
```

Function Description:

This function sets up and opens the connection to communicate to the 8810A via IEEE.
The IEEE supports the following language protocols:

- API-8810A Native
- API-8810 Native (Legacy)
- API-8810 SR103 (Legacy)
- API-8810 HSR202 (Legacy)
- API-8810 HSR203 (Legacy)

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nIEEEAddr - IEEE Address to be used to connect to 8810A. (0-30)
nIEEELang - Language Protocol to be used to communicate via IEEE to 8810A.

8810A Language Types:

API8810A_NATIVE	0
IEEE_API8810_NATIVE	1
IEEE_API8810_SR103	2
IEEE_API8810_HSR202	3
IEEE_API8810_HSR203	4
IEEE_API8810_MATECIIL	5
IEEE_API8810_FX2	6

Return Value:

API_SUCCESS – successfully connection via IEEE using specified address and language protocol
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_ADDRS - invalid IEEE Address parameter
API_ERROR_LANG - invalid 8810A Language parameter
API_ERROR_OPEN_API_SESSION - IEEE connection or configuration error

References for this function:

This function will make a call to the API8810A_SetIEEELang() routine to force the 8810A device to handle commands in the language specified.

2.2 API8810A_ConnectViaUSB

Format:

```
_API8810AFUNC int API8810A_ConnectViaUSB
(
   int apiNo,
   int nDeviceNo
)
```

Function Description:

This function sets up and opens the connection to communicate to the 8810A via USB.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nDeviceNo – Device Number to be used to connect to 8810A. (0-30)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_USB_CONNECTION– USB connection error

References for this function:

Prior to calling this function, make calls to the API8810A_GetAPI8810AUSBDeviceCnt() routine to determine the number of Cypress USB Devices detected in your system and the API8810A_GetAPI8810ADeviceIDN() routine to determine the device number (DeviceNo) associated with the Cypress USB Devices that are connected to 8810A via USB.

Sample Code:

The following sample code is available in the 8810A Software Package under the folder: ..\Driver\Source\API8810AUSBConnect. The sample code is written in C, compiled under Microsoft Visual .NET 2003 and invokes the routines in the API-8810A Dll that has been included in the software package.

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

#define _BUILD_API8810ADLL
#define __WIN32__
#include "API8810ADll.h"

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

/*****************************************************************************
*  Function:     main
*  Description:  Main routine for API8810A USB Connection application.
*                Scan for Cypress USB devices and opens each USB device to
*                determine which one is connected to a 8810A via IDN command.
*                Prompts the user for the USB Endpoint associated with 8810A
*                device and reads and displays the Signal Mode for Channel 1.
*  Parameters:   None
*  Return:       1 if successful.
*                -1 if any failure to API8810ADll calls.
*****************************************************************************/
int main()
{
   HANDLE hWnd;
   COORD bufferSize;
   BOOL bRetry;
   int nStatus;
   int nUSBDeviceCnt;
   int n8810ADeviceCnt;
   char sz8810AIDN[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 8810A
   */
   int aUSB_8810A_Endpoints[MAX_API];
   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("API8810A USB Connection");

   bRetry = TRUE;
   while (bRetry)
   {
      /* Initialize the array of Cypress USB Endpoints to -1 */
      for (i = 0; i < MAX_API; i++)
         aUSB_8810A_Endpoints[i] = -1;

      /* Scan for USB Devices */
      nStatus = API8810A_GetAPI8810AUSBDeviceCnt(&nUSBDeviceCnt);
      if (nStatus != API_SUCCESS)
      {
         printf("\nAPI8810A_GetAPI8810AUSBDeviceCnt Error: %d", nStatus);
         return -1;
      }
```

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

if (nUSBDeviceCnt > 0)
{
   /* Determine which USB devices are connected to 8810A */
   n8810ADeviceCnt = 0;
   for (i = 0; i < nUSBDeviceCnt; i++)
   {
      nStatus = API8810A_GetAPI8810ADeviceIDN(i, &sz8810AIDN[0]);
      if (nStatus != API_SUCCESS)
      {
         /* The USB device connected is not a 8810A */
         printf("\nUSB Endpoint: %d   NOT 8810A device", i);
      }
      else
      {
         /* Replace the \r\n with a terminator character (\0) */
         for (j = 0; j < (int)strlen(sz8810AIDN); j++)
         {
            if (sz8810AIDN[j] == '\r')
            {
               sz8810AIDN[j] = '\0';
               break;
            }
         }
         /* Track the endpoints that are connected to 8810A */
         aUSB_8810A_Endpoints[i] = i;
         n8810ADeviceCnt++;

         /* Display the IDN information */
         printf("\nUSB Endpoint: %d   IDN:%s", i, sz8810AIDN);
      }
   }

   /* Request 8810A USB device to connect to */
   bValidEntry = FALSE;
   while(!bValidEntry)
   {
      printf("\n\nPlease Enter USB Endpoint Device for 8810A to Connect: ");
      scanf("%d", &nConnectEndpoint);
      if (nConnectEndpoint < MAX_API)
      {
         if (aUSB_8810A_Endpoints[nConnectEndpoint] == -1)
            printf("Endpoint entered is connected to 8810A.\n");
         else
            bValidEntry = TRUE;
      }
      else
         printf("Endpoint entered is not valid");
   }

   /* Connect to 8810A */
   nStatus = API8810A_ConnectViaUSB(1, nConnectEndpoint);
   if (nStatus != API_SUCCESS)
   {
      printf("\nAPI8810A_ConnectViaUSB Error: %d", nStatus);
      return -1;
   }

   /* Get Channel 1 Mode Information */
   nStatus = API8810A_GetAPISignalMode(1, 1, &nSigMode);
   if (nStatus != API_SUCCESS)
   {
      printf("\nAPI8810A_GetAPISignalMode 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 = API8810A_DisconnectUSB(1);
         if (nStatus != API_SUCCESS)
         {
            printf("\nAPI8810A_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 3 Cypress USB Devices:
USB Endpoint: 0   NOT 8810A device
USB Endpoint: 1   IDN: NORTH ATLANTIC,8810AH-R,31164,5.13.4.5.4
USB Endpoint: 2   IDN: NORTH ATLANTIC,8810A-R,35524,5.18.102.102.15
Please Enter USB Endpoint Device to Connect: 1

Channel 1 Signal Mode = SYNCHRO

Please type q or Q to quit:
```

2.3 API8810A_ConnectViaEthernet

Format:

```
_API8810AFUNC int API8810A_ConnectViaEthernet
(
   int apiNo,
   char *szIPAddr,
   int nPort
)
```

Function Description:

This function sets up and opens the connection to communicate to the 8810A via Ethernet.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
szIPAddr - IP Address to be used to connect to 8810A.
nPort – Port to be used to connect to 8810A.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE – unable to perform ethernet login to 8810A
API_ERROR_ETHER_CONNECTION– Ethernet connection error

References for this function:

None.

2.4 API8810A_DisconnectIEEE

Format:

```
_API8810AFUNC int API8810A_DisconnectIEEE
(
   int apiNo
)
```

Function Description:

This function closes the connection to communicate to the 8810A via IEEE.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter

References for this function:

Prior to calling this function, make call to the API8810A_ConnectViaIEEE() routine to connected to 8810A via IEEE.

2.5 API8810A_DisconnectUSB

Format:

```
_API8810AFUNC int API8810A_DisconnectUSB
(
   int apiNo
)
```

Function Description:

This function closes the connection to communicate to the 8810A via USB.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter

References for this function:

Prior to calling this function, make call to the API8810A_ConnectViaUSB() routine to connected to 8810A via USB.

2.6 API8810A_DisconnectEthernet

Format:

```
_API8810AFUNC int API8810A_DisconnectEthernet
(
   int apiNo
)
```

Function Description:

This function closes the connection to communicate to the 8810A via Ethernet.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter

References for this function:

Prior to calling this function, make call to the API8810A_ConnectViaEthernet() routine to connected to 8810A via Ethernet.

2.7 API8810A_GetAPI8810AUSBDeviceCnt

Format:

```
_API8810AFUNC int API8810A_GetAPI8810AUSBDeviceCnt
(
   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:

API_SUCCESS - function is successful

References for this function:

None.

2.8 API8810A_GetAPI8810ADeviceIDN

Format:

```
_API8810AFUNC int API8810A_ GetAPI8810ADeviceIDN
(
   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,8810A" 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 8810A.
pszIDN – pointer to location to return the IDN query response

Return Value:

API_SUCCESS - function is successful
API_ERROR_RANGE_GET – value retrieved is out-of-range
API_ERROR_DATA – data returned from 8810A is not valid for command sent
API_ERROR_USB_CONNECTION – USB connection error when opening device with Device number specified.

References for this function:

None.

3 API-8810A Channel Routines

The routines in this section handle retrieving channel information from the 8810A device and setting channel configurations.

3.1 API8810A_SetAPITrackHold

Format:

```
_API8810AFUNC int API8810A_SetAPITrackHold
(
   int apiNo,
   int nChanNo,
   bool bHold
)
```

Function Description:

This function sends the command to set 8810A channel specified to track or hold the angle data. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
bHold – command to track or hold channel angle

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function not supported (Track or Hold not supported in MATE/CIIL)
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.2 API8810A_GetAPITrackHold

Format:

```
_API8810AFUNC int API8810A_GetAPITrackHold
(
   int apiNo,
   int nChanNo,
   bool *pbHold
)
```

Function Description:

This function sends the command to get the track or hold state for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

```
nChanNo – 8810A Channel
pbHold – pointer to location to return the track or hold state for the
   channel
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function not supported (Track or Hold not
   supported in MATE/CIIL)
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_RANGE_GET – value retrieved is out-of-range
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.3 API8810A_SetAPISignalMode

Format:

```
_API8810AFUNC int API8810A_SetAPISignalMode
(
   int apiNo,
   int nChanNo,
   int nSigMode
)
```

Function Description:

This function sends the command to set 8810A channel specified to resolver or synchro mode. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nSigMode – signal mode to set channel
                  Mode Types:
                     RESOLVER          0
                     SYNCHRO           1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

3.4 API8810A_GetAPISignalMode

Format:

```
_API8810AFUNC int API8810A_GetAPISignalMode
```

```
(
   int apiNo,
   int nChanNo,
   int *pnSigMode
)
```

Function Description:

This function sends the command to get the signal mode for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pnSigMode – pointer to location to return the channel signal mode

Mode Types:	
RESOLVER	0
SYNCHRO	1

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_RANGE_GET – value retrieved is out-of-range
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.5 API8810A_SetAPIReferenceSrc

Format:

```
_API8810AFUNC int API8810A_SetAPIReferenceSrc
(
   int apiNo,
   int nChanNo,
   int nRefSrc
)
```

Function Description:

This function sends the command to set 8810A channel specified to internal or external reference mode. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nRefSrc – reference mode to set channel

Reference Source Types:	
INTERNAL	0
EXTERNAL	1

Return Value:

API_SUCCESS - function is successful

```
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

3.6 API8810A_GetAPIReferenceSrc

Format:

```
_API8810AFUNC int API8810A_GetAPIReferenceSrc
(
   int apiNo,
   int nChanNo,
   int *pnRefSrc
)
```

Function Description:

This function sends the command to get the reference mode for the 8810A channel.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pnRefSrc – pointer to location to return the reference mode
                  Reference Source Types:
                     INTERNAL          0
                     EXTERNAL          1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_RANGE_GET – value retrieved is out-of-range
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.7 API8810A_SetAPIRatio

Format:

```
_API8810AFUNC int API8810A_SetAPIRatio
(
   int apiNo,
   int nChanNo,
   int nRatio
)
```

Function Description:

This function sends the command to set the ratio value for the specified 8810A channel. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nRatio – ratio value set channel (1 – 255)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.8 API8810A_GetAPIRatio

Format:

```
_API8810AFUNC int API8810A_GetAPIRatio
(
   int apiNo,
   int nChanNo,
   int *pnRatio
)
```

Function Description:

This function sends the command to get the ratio value for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pnRatio – pointer to location to return the ratio value

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.9 API8810A_SetAPIAutoBandwidth

Format:

```
_API8810AFUNC int API8810A_SetAPIAutoBandwidth
(
   int apiNo,
   int nChanNo
)
```

Function Description:

This function sends the command to set the bandwidth to "auto" mode for the specified 8810A channel. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.10 API8810A_SetAPIBandwidth

Format:

```
_API8810AFUNC int API8810A_SetAPIBandwidth
(
   int apiNo,
   int nChanNo,
   int nBandwidth
)
```

Function Description:

This function sends the command to set the bandwidth to "override" mode with the bandwidth specified for the specified 8810A channel. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nBandwidth – Bandwidth to set the channel

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter

```
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

3.11 API8810A_GetAPIBandwidth

Format:

```
_API8810AFUNC int API8810A_GetAPIBandwidth
(
   int apiNo,
   int nChanNo,
   bool *pbAutoBW,
   int *pnBandwidth
)
```

Function Description:

```
This function sends the command to get the bandwidth mode and bandwidth
value for the 8810A channel.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pbAutoBW – pointer to location to return the bandwidth mode indicator
pnBandwidth – pointer to location to return the bandwidth value
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

3.12 API8810A_GetAPIAngle

Format:

```
_API8810AFUNC int API8810A_GetAPIAngle
(
   int apiNo,
   int nChanNo,
   double *pdAngle
)
```

Function Description:

```
This function sends the command or perform an IEEE read (with 8810 Legacy
languages) to get angle value for the 8810A channel.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdAngle – pointer to location to return the angle value
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

3.13 API8810A_GetAPIAvgAngle

Format:

```
_API8810AFUNC int API8810A_GetAPIAvgAngle
(
   int apiNo,
   int nChanNo,
   double *pdAvgAngle
)
```

Function Description:

```
This function sends the command or perform an IEEE read (with 8810 Legacy
languages) to get angle value for the 8810A channel. In API8810A_NATIVE
language mode, if the angle averaging is turned on, the average angle
value is returned by the unit, otherwise the angle value is returned by
the unit.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdAvgAngle – pointer to location to return the angle average value
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

3.14 API8810A_GetAPIVelocity

Format:

```
_API8810AFUNC int API8810A_GetAPIVelocity
(
   int apiNo,
   int nChanNo,
```

```
   double *pdVelocity
)
```

Function Description:

This function sends the command to get angle velocity value for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdVelocity – pointer to location to return the angle velocity value

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.15 API8810A_GetAPILineLineVolt

Format:

```
_API8810AFUNC int API8810A_GetAPILineLineVolt
(
   int apiNo,
   int nChanNo,
   double *pdLineLineVolt
)
```

Function Description:

This function sends the command to get line-to-line voltage value for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdLineLineVolt – pointer to location to return the line-to-line voltage value

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.16 API8810A_GetAPINullVolt

Format:

```
_API8810AFUNC int API8810A_GetAPINullVolt
(
   int apiNo,
   int nChanNo,
   double *pdNullVolt
)
```

Function Description:

This function sends the command to get null voltage value for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdNullVolt – pointer to location to return the null voltage value

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

Remarks:

Currently the 8810A device will always return 0 for the null voltage.

3.17 API8810A_GetAPIRefVolt

Format:

```
_API8810AFUNC int API8810A_GetAPIRefVolt
(
   int apiNo,
   int nChanNo,
   double *pdRefVolt
)
```

Function Description:

This function sends the command to get reference voltage value for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdRefVolt – pointer to location to return the reference voltage value

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

3.18 API8810A_GetAPIRefFreq

Format:

```
_API8810AFUNC int API8810A_GetAPIRefFreq
(
   int apiNo,
   int nChanNo,
   double *pdRefFreq
)
```

Function Description:

```
This function sends the command to get reference frequency value for the
8810A channel.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdRefFreq – pointer to location to return the reference frequency value
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

3.19 API8810A_SetAPIAvgState

Format:

```
_API8810AFUNC int API8810A_SetAPIAvgState
(
   int apiNo,
   int nChanNo,
   bool bAvgStateOn
)
```

Function Description:

```
This function sends the command to turn on or off the angle averaging mode
for the specified 8810A channel. Note the 8810A device will not accept the
```

command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
bAvgStateOn – Turn on or off angle averaging for the channel

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.20 API8810A_GetAPIAvgState

Format:

```
_API8810AFUNC int API8810A_GetAPIAvgState
(
   int apiNo,
   int nChanNo,
   bool *pbAvgStateOn
)
```

Function Description:

This function sends the command to get the angle averaging mode for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pbAvgStateOn – pointer to location to return the angle averaging mode

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.21 API8810A_SetAPIAvgRate

Format:

```
_API8810AFUNC int API8810A_SetAPIAvgRate
(
```

```
   int apiNo,
   int nChanNo,
   int nAvgRate
)
```

Function Description:

This function sends the command to set the angle averaging rate in msec for the specified 8810A channel. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nAvgRate – angle averaging rate to set for the channel (10-10000 msec)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.22 API8810A_GetAPIAvgRate

Format:

```
_API8810AFUNC int API8810A_GetAPIAvgRate
(
   int apiNo,
   int nChanNo,
   int *pnAvgRate
)
```

Function Description:

This function sends the command to get the angle averaging rate for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pnAvgRate – pointer to location to return the angle averaging rate

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.23 API8810A_SetAPIAngLimitState

Format:

```
_API8810AFUNC int API8810A_SetAPIAngLimitState
(
   int apiNo,
   int nChanNo,
   bool bAngLmtOn
)
```

Function Description:

This function sends the command to turn on or off the angle limit testing mode for the specified 8810A channel. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
bAngLmtOn – Turn on or off angle limit testing for the channel

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.24 API8810A_GetAPIAngLimitState

Format:

```
_API8810AFUNC int API8810A_GetAPIAngLimitState
(
   int apiNo,
   int nChanNo,
   bool *pbAngLmtOn
)
```

Function Description:

This function sends the command to get the angle limit testing mode for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pbAngLmtOn – pointer to location to return the angle limit testing mode

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.25 API8810A_SetAPIAngLimitCompare

Format:

```
_API8810AFUNC int API8810A_SetAPIAngLimitCompare
(
   int apiNo,
   int nChanNo,
   int nAngLimitCompare
)
```

Function Description:

```
This function sends the command to set 8810A channel specified with the
angle comparison mode for angle limit testing. Note the 8810A device will
not accept the command if its remote communication configuration does not
match the communication connection mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nAngLimitCompare – comparison mode for angle limit testing to set channel
                  Angle Limit Comparison Mode Types:
                     ANGLE_LIMIT_ABS_ANG     0
                     ANGLE_LIMIT_ANG_ERR     1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_RANGE – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

3.26 API8810A_GetAPIAngLimitCompare

Format:

```
_API8810AFUNC int API8810A_GetAPIAngLimitCompare
(
   int apiNo,
```

```
   int nChanNo,
   char *pszAngLimitCompare
)
```

Function Description:

This function sends the command to get the angle comparison mode for angle limit testing for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pszAngLimitCompare – pointer to location to return the angle comparison mode for angle limit testing

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.27 API8810A_SetAPIAngUpperLimit

Format:

```
_API8810AFUNC int API8810A_SetAPIAngUpperLimit
(
   int apiNo,
   int nChanNo,
   double dAngUpperLimit
)
```

Function Description:

This function sends the command to set 8810A channel specified with the upper angle limit value for angle limit testing. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
dAngUpperLimit – upper angle limit value to set channel for angle limit testing (0-360.0)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range

```
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

3.28 API8810A_GetAPIAngUpperLimit

Format:

```
_API8810AFUNC int API8810A_GetAPIAngUpperLimit
(
   int apiNo,
   int nChanNo,
   double *pdAngUpperLimit
)
```

Function Description:

```
This function sends the command to get the upper angle limit value for
angle limit testing for the 8810A channel.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdAngUpperLimit – pointer to location to return the upper angle limit
   value for angle limit testing
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.29 API8810A_SetAPIAngLowerLimit

Format:

```
_API8810AFUNC int API8810A_SetAPIAngLowerLimit
(
   int apiNo,
   int nChanNo,
   double dAngLowerLimit
)
```

Function Description:

```
This function sends the command to set 8810A channel specified with the
lower angle limit value for angle limit testing. Note the 8810A device
will not accept the command if its remote communication configuration does
not match the communication connection mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
dAngLowerLimit – lower angle limit value to set channel for angle limit
   testing (0-360.0)
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

3.30 API8810A_GetAPIAngLowerLimit

Format:

```
_API8810AFUNC int API8810A_GetAPIAngLowerLimit
(
   int apiNo,
   int nChanNo,
   double *pdAngLowerLimit
)
```

Function Description:

This function sends the command to get the lower angle limit value for angle limit testing for the 8810A channel.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdAngLowerLimit – pointer to location to return the lower angle limit
   value for angle limit testing
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.31 API8810A_SetAPIAngLimitErrorStep

Format:

```
_API8810AFUNC int API8810A_SetAPIAngLimitErrorStep
(
   int apiNo,
```

```
   int nChanNo,
   double dAngLimitErrorStep
)
```

Function Description:

This function sends the command to set 8810A channel specified with the angle step value for angle error comparison for angle limit testing. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
dAngLimitErrorStep – angle step value to set channel for angle limit testing (0-360.0)

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.32 API8810A_GetAPIAngLimitErrorStep

Format:

```
_API8810AFUNC int API8810A_GetAPIAngLimitErrorStep
(
   int apiNo,
   int nChanNo,
   double *pdAngLimitErrorStep
)
```

Function Description:

This function sends the command to get the angle step value for angle error comparison for angle limit testing for the 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdAngLimitErrorStep – pointer to location to return the angle step value for angle error comparison for angle limit testing.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.33 API8810A_SetAPIDAOutput

Format:

```
_API8810AFUNC int API8810A_SetAPIDAOutput
(
   int apiNo,
   int nChanNo,
   int nDAOutput
)
```

Function Description:

This function sends the command to set 8810A channel specified with the data type to use for DA output. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
nDAOutput – data type mode for DA output
                  DA Output Data Types:
                     DA_ANGLE_OUTPUT         0
                     DA_VELOCITY_OUTPUT      1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_RANGE – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

3.34 API8810A_GetAPIDAOutput

Format:

```
_API8810AFUNC int API8810A_GetAPIDAOutput
(
   int apiNo,
   int nChanNo,
   char *pszDAOutput
)
```

Function Description:

This function sends the command to get the data type to use for DA output for the 8810A channel.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pszDAOutput – pointer to location to return the data type for DA output
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.35 API8810A_SetAPIDAUpperLimit

Format:

```
_API8810AFUNC int API8810A_SetAPIDAUpperLimit
(
   int apiNo,
   int nChanNo,
   double dDAUpperLimit
)
```

Function Description:

```
This function sends the command to set 8810A channel specified with the
upper angle or velocity limit value for DA output. Note the 8810A device
will not accept the command if its remote communication configuration does
not match the communication connection mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
dDAUpperLimit – upper angle or velocity limit value to set channel for DA
   Output
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

3.36 API8810A_GetAPIDAUpperLimit

Format:

```
_API8810AFUNC int API8810A_GetAPIDAUpperLimit
(
   int apiNo,
```

```
   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 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdDAUpperLimit – pointer to location to return the upper angle or velocity limit for DA output.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.37 API8810A_SetAPIDALowerLimit

Format:

```
_API8810AFUNC int API8810A_SetAPIDALowerLimit
(
   int apiNo,
   int nChanNo,
   double dDALowerLimit
)
```

Function Description:

This function sends the command to set 8810A channel specified with the lower angle or velocity limit value for DA output. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
dDALowerLimit – lower angle or velocity limit value to set channel for DA Output

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

3.38 API8810A_GetAPIDALowerLimit

Format:

```
_API8810AFUNC int API8810A_GetAPIDALowerLimit
(
   int apiNo,
   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 8810A channel.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdDALowerLimit – pointer to location to return the lower angle or velocity limit for DA output.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

3.39 API8810A_SetAPIDAUpperVoltage

Format:

```
_API8810AFUNC int API8810A_SetAPIDAUpperVoltage
(
   int apiNo,
   int nChanNo,
   double dDAUpperVoltage
)
```

Function Description:

This function sends the command to set 8810A channel specified with the voltage value associated with the upper limit for DA output. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

```
nChanNo – 8810A Channel
dDAUpperVoltage – voltage value associated with upper limit value to set
   channel for DA Output
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

3.40 API8810A_GetAPIDAUpperVoltage

Format:

```
_API8810AFUNC int API8810A_GetAPIDAUpperVoltage
(
   int apiNo,
   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 8810A channel.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdDAUpperVoltage – pointer to location to return the voltage value
   associated with upper limit value for DA output.
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.41 API8810A_SetAPIDALowerVoltage

Format:

```
_API8810AFUNC int API8810A_SetAPIDALowerVoltage
(
   int apiNo,
   int nChanNo,
   double dDALowerVoltage
```

```
)
```

Function Description:

```
This function sends the command to set 8810A channel specified with the
voltage value associated with the lower limit for DA output. Note the
8810A device will not accept the command if its remote communication
configuration does not match the communication connection mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
dDALowerVoltage – voltage value associated with lower limit value to set
   channel for DA Output
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

3.42 API8810A_GetAPIDALowerVoltage

Format:

```
_API8810AFUNC int API8810A_GetAPIDALowerVoltage
(
   int apiNo,
   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 8810A channel.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChanNo – 8810A Channel
pdDALowerVoltage – pointer to location to return the voltage value
   associated with lower limit value for DA output.
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.43 API8810A_SetAPIDisplayAngDiffState

Format:

```
_API8810AFUNC int API8810A_SetAPIDisplayAngDiffState
(
   int apiNo,
   int nAngDiffState
)
```

Function Description:

This function sends the command to turn on or off the angle difference mode. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nAngDiffState – Turn on or off angle difference mode for the channel
                  Angle Difference Mode Types:
                     ANG_DIFF_DISABLED       0
                     ANG_DIFF_ENABLED        1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

Remarks:

When the 8810A device received the command to turn on the angle difference mode, it will automatically revert to the Dual Channel screen so that the angle difference data is visible on the 8810A display.

3.44 API8810A_GetAPIDisplayAngDiffState

Format:

```
_API8810AFUNC int API8810A_GetAPIDisplayAngDiffState
(
   int apiNo,
   int *pnAngDiffState
)
```

Function Description:

This function sends the command to get the angle difference mode setting.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
```

```
pnAngDiffState – pointer to location to return the angle difference mode
                  Angle Difference Mode Types:
                     ANG_DIFF_DISABLED       0
                     ANG_DIFF_ENABLED        1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

3.45 API8810A_GetAPIAngleDiff

Format:

```
_API8810AFUNC int API8810A_GetAPIAngleDiff
(
   int apiNo,
   double *pdAngleDiff
)
```

Function Description:

```
This function sends the command to get difference between angle value for
the Channel 1 and Channel 2 signal.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdAngleDiff – pointer to location to return the angle difference value
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

4 API-8810A Multiple Channel Query Routines

The routines in this section handle retrieving channel information from the 8810A device using one query call rather than individual query calls for each channel.

4.1 API8810A_GetAPIAngles

Format:

```
_API8810AFUNC int API8810A_GetAPIAngles
(
   int apiNo,
   double *pdAngle1,
   double *pdAngle2
)
```

Function Description:

This function sends the command to get the angle values for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdAngle1– pointer to location to return the angle value for Channel 1
pdAngle2– pointer to location to return the angle value for Channel 2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

4.2 API8810A_GetAPIAvgStates

Format:

```
_API8810AFUNC int API8810A_GetAPIAvgStates
(
   int apiNo,
   bool *pbAvgStateOn1,
   int *pnAvgRate1,
   bool *pbAvgStateOn2,
   int *pnAvgRate2
)
```

Function Description:

This function sends the command to get the angle averaging mode and the average rates for both 8810A channels.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pbAvgStateOn1 – pointer to location to return the angle averaging mode for
   Channel 1
pnAvgRate1– pointer to location to return the average rate value for
   Channel 1
pbAvgStateOn2 – pointer to location to return the angle averaging mode for
   Channel 2
pnAvgRate2– pointer to location to return the average rate value for
   Channel 2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

4.3 API8810A_GetAPIAvgAngles

Format:

```
_API8810AFUNC int API8810A_GetAPIAvgAngles
(
   int apiNo,
   double *pdAvgAngle1,
   double *pdAvgAngle2
)
```

Function Description:

```
This function sends the command to get the angle average values for both
8810A channels.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdAvgAngle1– pointer to location to return the angle average value for
   Channel 1
pdAvgAngle2– pointer to location to return the angle average value for
   Channel 2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

4.4 API8810A_GetAPIBandwidths

Format:

```
_API8810AFUNC int API8810A_GetAPIBandwidths
(
   int apiNo,
   bool *pbAutoBW1,
   int *pnBandwidth1,
   bool *pbAutoBW2,
   int *pnBandwidth2
)
```

Function Description:

This function sends the command to get the bandwidth modes and bandwidth values for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pbAutoBW1 – pointer to location to return the bandwidth mode for Channel 1
pnBandwidth1– pointer to location to return the bandwidth value for Channel 1
pbAutoBW2 – pointer to location to return the bandwidth mode for Channel 2
pnBandwidth2– pointer to location to return the bandwidth value for Channel 2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

4.5 API8810A_GetAPISignalModes

Format:

```
_API8810AFUNC int API8810A_GetAPISignalModes
(
   int apiNo,
   int *pnSigMode1,
   int *pnSigMode2
)
```

Function Description:

This function sends the command to get signal modes for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnSigMode1– pointer to location to return the signal mode for Channel 1

```
pnSigMode2– pointer to location to return the signal mode for Channel 2
                  Mode Types:
                     RESOLVER          0
                     SYNCHRO           1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

4.6 API8810A_GetAPIRatios

Format:

```
_API8810AFUNC int API8810A_GetAPIRatios
(
   int apiNo,
   int *pnRatio1,
   int *pnRatio2
)
```

Function Description:

```
This function sends the command to get ratio values for both 8810A
channels.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnRatio1– pointer to location to return the ratio value for Channel 1
pnRatio2– pointer to location to return the ratio value for Channel 2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

4.7 API8810A_GetAPIReferenceSrcs

Format:

```
_API8810AFUNC int API8810A_GetAPIReferenceSrcs
(
   int apiNo,
   int *pnRefSrc1,
```

```
   int *pnRefSrc2
)
```

Function Description:

This function sends the command to get reference modes for both 8810A channels.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnRefSrc1– pointer to location to return the reference mode for Channel 1
pnRefSrc2– pointer to location to return the reference mode for Channel 2
                  Reference Source Types:
                     INTERNAL          0
                     EXTERNAL          1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

4.8 API8810A_GetAPITrackHolds

Format:

```
_API8810AFUNC int API8810A_GetAPITrackHolds
(
   int apiNo,
   bool *pbHold1,
   bool *pbHold2
)
```

Function Description:

This function sends the command to get track or hold states for both 8810A channels.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pbHold1– pointer to location to return the track or hold state for Channel 1
pbHold2– pointer to location to return the track or hold state for Channel 2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

4.9 API8810A_GetAPIAngLimitStates

Format:

```
_API8810AFUNC int API8810A_GetAPIAngLimitStates
(
   int apiNo,
   bool *pbAngLmtOn1,
   int *pnAngLimitCmp1,
   bool *pbAngLmtOn2,
   int *pnAngLimitCmp2
)
```

Function Description:

This function sends the command to get angle limit testing states and comparison modes for both 8810A channels.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pbAngLmtOn1– pointer to location to return the angle limit testing state
   for Channel 1
pnAngLimitCmp1 – pointer to location to return the angle comparison mode
   for Channel 1
pbAngLmtOn2– pointer to location to return the angle limit testing state
   for Channel 2
pnAngLimitCmp2 – pointer to location to return the angle comparison mode
   for Channel 2
                  Angle Limit Comparison Mode Types:
                     ANGLE_LIMIT_ABS_ANG     0
                     ANGLE_LIMIT_ANG_ERR     1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

4.10 API8810A_GetAPIVelocities

Format:

```
_API8810AFUNC int API8810A_GetAPIVelocities
(
   int apiNo,
   double *pdVelocity1,
   double *pdVelocity2
)
```

Function Description:

This function sends the command to get the angle velocity values for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdVelocity1– pointer to location to return the velocity value for Channel 1
pdVelocity2– pointer to location to return the velocity value for Channel 2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

4.11 API8810A_GetAPILineLineVolts

Format:

```
_API8810AFUNC int API8810A_GetAPILineLineVolts
(
   int apiNo,
   double *pdLineLineVolt1,
   double *pdLineLineVolt2
)
```

Function Description:

This function sends the command to get the line-to-line voltage values for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdLineLineVolt1– pointer to location to return the line-to-line voltage value for Channel 1
pdLineLineVolt2– pointer to location to return the line-to-line voltage value for Channel 2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

4.12 API8810A_GetAPINullVolts

Format:

```
_API8810AFUNC int API8810A_GetAPINullVolts
(
   int apiNo,
   double *pdNullVolt1,
   double *pdNullVolt2
)
```

Function Description:

This function sends the command to get the null voltage values for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdNullVolt1– pointer to location to return the null voltage value for Channel 1
pdNullVolt2– pointer to location to return the null voltage value for Channel 2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

4.13 API8810A_GetAPIRefVolts

Format:

```
_API8810AFUNC int API8810A_GetAPIRefVolts
(
   int apiNo,
   double *pdRefVolt1,
   double *pdRefVolt2
)
```

Function Description:

This function sends the command to get the reference voltage values for both 8810A channels.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdRefVolt1– pointer to location to return the reference voltage value for Channel 1
pdRefVolt2– pointer to location to return the reference voltage value for Channel 2

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

4.14 API8810A_GetAPIRefFreqs

Format:

```
_API8810AFUNC int API8810A_GetAPIRefFreqs
(
   int apiNo,
   double *pdRefFreq1,
   double *pdRefFreq2
)
```

Function Description:

This function sends the command to get the reference frequency values for both 8810A channels.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdRefFreq1– pointer to location to return the reference frequency value
   for Channel 1
pdRefFreq2– pointer to location to return the reference frequency value
   for Channel 2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

5 API-8810A Internal Reference Routines

The routines in this section handle retrieving and setting configurations for the reference module in the 8810A device if available.

5.1 API8810A_SetIntRefFreq

Format:

```
_API8810AFUNC int API8810A_SetIntRefFreq
(
   int apiNo,
   double dFreq
)
```

Function Description:

This function sends the command to set the reference frequency for the reference module (if available). Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
dFreq – Frequency value to set the reference module

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

5.2 API8810A_GetIntRefFreq

Format:

```
_API8810AFUNC int API8810A_GetIntRefFreq
(
   int apiNo,
   double *pdFreq
)
```

Function Description:

This function sends the command to get the reference frequency for the reference module.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdFreq – pointer to location to return the frequency value to set the
   reference module

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

Remarks:

```
If the 8810A unit does not contain a reference module, the device will
always return 400 Hz for the frequency value.
```

5.3 API8810A_SetIntRefVolt

Format:

```
_API8810AFUNC int API8810A_SetIntRefVolt
(
   int apiNo,
   double dVolt
)
```

Function Description:

```
This function sends the command to set the reference voltage for the
reference module (if available). Note the 8810A device will not accept the
command if its remote communication configuration does not match the
communication connection mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
dVolt – Voltage value to set the reference module
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

5.4 API8810A_GetIntRefVolt

Format:

```
_API8810AFUNC int API8810A_GetIntRefVolt
(
   int apiNo,
   double *pdVolt
)
```

Function Description:

This function sends the command to get the reference voltage for the reference module.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdVolt – pointer to location to return the voltage value to set the reference module

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

Remarks:

If the 8810A unit does not contain a reference module, the device will always return 26 volts for the voltage value.

5.5 API8810A_SetIntRefOutputState

Format:

```
_API8810AFUNC int API8810A_SetIntRefOutputState
(
   int apiNo,
   int nOutputState
)
```

Function Description:

This function sends the command to set the reference output state for the reference module (if available). Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
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:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

5.6 API8810A_GetIntRefOutputState

Format:

```
_API8810AFUNC int API8810A_GetIntRefOutputState
(
   int apiNo,
   int *pnOutputState
)
```

Function Description:

This function sends the command to get the reference output state for the reference module.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
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:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

Remarks:

If the 8810A unit does not contain a reference module, the device will always return INT_REF_OUT_NOT_AVAILABLE.

5.7 API8810A_GetIntRefOverCurrentState

Format:

```
_API8810AFUNC int API8810A_GetIntRefOverCurrentState
(
   int apiNo,
   bool *pbIntRefOverCurrent
)
```

Function Description:

This function sends the command to get the reference over-current state for the reference module.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pbIntRefOverCurrent – pointer to location to return the reference over-
current state to set the reference module
                  Reference Over-current States:
                     true - Reference is in over-current state
                     false - Reference is not in over-current state
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

Remarks:

```
If the 8810A unit does not contain a reference module, the device will
always return false.
```

5.8 API8810A_ResetIntRefOverCurrent

Format:

```
_API8810AFUNC int API8810A_ResetIntRefOverCurrent
(
   int apiNo
)
```

Function Description:

```
This function sends the command to reset the reference over-current state
for the reference module (if available).

Note the 8810A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

6 API-8810A 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 8810A device setting to factory default settings.

6.1 API8810A_PerformGetID

Format:

```
_API8810AFUNC int API8810A_PerformGetID
(
   int apiNo,
   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 8810A module, serial number, and revision information.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszID – pointer to location to return the device ID

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

6.2 API8810A_Reset

Format:

```
_API8810AFUNC int API8810A_Reset
(
   int apiNo,
   char *pszResults
)
```

Function Description:

This function sends the command to reset the 8810A device and set the device setting back to the factory default settings. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
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:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

6.3 API8810A_GetErrors

Format:

```
_API8810AFUNC int API8810A_GetErrors
(
   int apiNo,
   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:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszErrors – pointer to location to return the error string

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

7 API-8810A Configuration Routines

The routines in this section handle sending commands to set and retrieve the configuration settings of the 8810A device.

7.1 API8810A_GetIEEELang

Format:

```
_API8810AFUNC int API8810A_GetIEEELang
(
   int apiNo,
   char *pszIEEELang
)
```

Function Description:

This function sends the command to get the IEEE language protocol set in the 8810A.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszIEEELang – pointer to location to return the IEEE protocol string
Return values:
"API8810A Native" - API-8810A Native
"API8810 Native" - API-8810 Native (Legacy)
"API8810 SR103" - API-8810 SR103 (Legacy)
"API8810 HSR202" - API-8810 HSR202 (Legacy)
"API8810 HSR203" - API-8810 HSR203 (Legacy)
"API8810 MATE/CIIL" – API-8810 MATE/CIIL (Legacy)
"API8810 FX2" - API-8810 FX2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

7.2 API8810A_SetIEEELang

Format:

```
_API8810AFUNC int API8810A_SetIEEELang
(
   int apiNo,
   int nIEEELang
)
```

Function Description:

This function sends the command to set the IEEE protocol language to accept when communicating via IEEE. Note the 8810A device will not accept

the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nIEEELang - Language Protocol to be used to communicate via IEEE to 8810A.
                  8810A Language Types:
                     API8810A_NATIVE          0
                     IEEE_API8810_NATIVE      1
                     IEEE_API8810_SR103       2
                     IEEE_API8810_HSR202      3
                     IEEE_API8810_HSR203      4
                     IEEE_API8810_MATECIIL    5
                     IEEE_API8810_FX2         6
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.3 API8810A_GetCommState

Format:

```
_API8810AFUNC int API8810A_GetCommState
(
   int apiNo,
   char *pszCommState
)
```

Function Description:

This function sends the command to get the communication mode set in the 8810A.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszCommState – pointer to location to return the communication mode string
   Return values:
         "Local Mode"
         "Remote IEEE Addr:  API-IEEE Language"
         "Remote USB"
         "Remote Ethernet"
         "Remote J1"
         "Remote with Lockout via IEEE Addr:  API-IEEE Language"
         "Remote with Lockout via USB"
         "Remote with Lockout via Ethernet"
         "Remote with Lockout via J1"
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
```

```
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

7.4 API8810A_GoToLocal

Format:

```
_API8810AFUNC int API8810A_GoToLocal
(
   int apiNo
)
```

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:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.5 API8810A_SetLocalLockout

Format:

```
_API8810AFUNC int API8810A_SetLocalLockout
(
   int apiNo
)
```

Function Description:

This function sends the command to lockout configuration setting such as track or hold, signal mode, reference mode, and ratio setting from the 8810A front panel. Note the 8810A device will not accept the command if device is set to Local mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

API_SUCCESS - function is successful

```
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.6 API8810A_SetRemoteUSB

Format:

```
_API8810AFUNC int API8810A_SetRemoteUSB
(
   int apiNo
)
```

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:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.7 API8810A_SetRemoteEthernet

Format:

```
_API8810AFUNC int API8810A_SetRemoteEthernet
(
   int apiNo
)
```

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:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

API_SUCCESS - function is successful

```
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.8 API8810A_SetRemoteIEEE

Format:

```
_API8810AFUNC int API8810A_SetRemoteIEEE
(
   int apiNo
)
```

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:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.9 API8810A_SetRemoteJ1

Format:

```
_API8810AFUNC int API8810A_SetRemoteJ1
(
   int apiNo
)
```

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:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

API_SUCCESS - function is successful

```
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.10 API8810A_SetAngleDisplayFormat

Format:

```
_API8810AFUNC int API8810A_SetAngleDisplayFormat
(
   int apiNo,
   int nDisplayFormat
)
```

Function Description:

This function sends the command to set the angle display format. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nDisplayFormat – angle format to display on the 8810A
                  Angle Display Format Types:
                     ANGLE_FMT_360          0
                     ANGLE_FMT_180          1
                     ANGLE_FMT_MIN          2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported with API8810A_NATIVE,
    IEEE_API8810_NATIVE, IEEE_API8810_FX2 and IEEE_API8810_SR103
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.11 API8810A_GetAngleDisplayFormat

Format:

```
_API8810AFUNC int API8810A_GetAngleDisplayFormat
(
   int apiNo,
   int *pnDisplayFormat
)
```

Function Description:

This function sends the command to get the angle display format set in the 8810A.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnDisplayFormat – pointer to location to return the angle display format
                  Angle Display Format Types:
                     ANGLE_FMT_360          0
                     ANGLE_FMT_180          1
                     ANGLE_FMT_MIN          2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_RANGE_GET – value retrieved is out-of-range
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

7.12 API8810A_SetCh1Input

Format:

```
_API8810AFUNC int API8810A_SetCh1Input
(
   int apiNo,
   int nCh1InputConnector
)
```

Function Description:

This function sends the command to set the Channel 1 input connector configuration. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nCh1InputConnector – Channel 1 input connector configuration on the 8810A
                  Channel 1 Input Configuration Types:
                     CH1_INPUT_FRONT_CONNECTOR     0
                     CH1_INPUT_BACK_CONNECTOR      1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.13 API8810A_GetCh1Input

Format:

```
_API8810AFUNC int API8810A_GetCh1Input
(
   int apiNo,
   int *pnCh1InputConnector
)
```

Function Description:

This function sends the command to get the Channel 1 input connector configuration set in the 8810A.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnCh1InputConnector – pointer to location to return the Channel 1 input connector configuration

Channel 1 Input Configuration Types:	
CH1_INPUT_FRONT_CONNECTOR	0
CH1_INPUT_BACK_CONNECTOR	1

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

7.14 API8810A_SetTouchscreenState

Format:

```
_API8810AFUNC int API8810A_SetTouchscreenState
(
   int apiNo,
   int nTouchscreenState
)
```

Function Description:

This function sends the command to set the touch screen configuration. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nTouchscreenState – Touch screen configuration on the 8810A

Touch screen Configuration Types:	
TOUCHSCREEN_DISABLED	0
TOUCHSCREEN_ENABLED	1

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.15 API8810A_GetTouchscreenState

Format:

```
_API8810AFUNC int API8810A_GetTouchscreenState
(
   int apiNo,
   int *pnTouchscreenState
)
```

Function Description:

This function sends the command to get the touch screen configuration set in the 8810A.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnTouchscreenState – pointer to location to return the touch screen
   configuration
                  Touchscreen Configuration Types:
                     TOUCHSCREEN_DISABLED     0
                     TOUCHSCREEN_ENABLED      1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

7.16 API8810A_SetDisplayState

Format:

```
_API8810AFUNC int API8810A_SetDisplayState
(
   int apiNo,
   int nDisplayState
)
```

Function Description:

```
This function sends the command to set the main display state.
Note the 8810A device will not accept the command if its remote
communication configuration does not match the communication connection
mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nDisplayState – Display configuration on the 8810A
                  Display Configuration Types:
                     CHAN1_DISPLAY               0
                     CHAN2_DISPLAY               1
                     CHAN1_ANALOG_DISPLAY        2
                     CHAN2_ANALOG_DISPLAY        3
                     DUAL_DISPLAY                4
                     REFERENCE_DISPLAY           5
                     CHARTING_DISPLAY            6
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_RANGE– data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

7.17 API8810A_GetDisplayState

Format:

```
_API8810AFUNC int API8810A_GetDisplayState
(
   int apiNo,
   int *pnDisplayState
)
```

Function Description:

```
This function sends the command to get the main display configuration on
the 8810A.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnDisplayState – pointer to location to return the main screen
   configuration state.
                  Main Screen Configuration Types:
                     CHAN1_DISPLAY               0
                     CHAN2_DISPLAY               1
                     CHAN1_ANALOG_DISPLAY        2
                     CHAN2_ANALOG_DISPLAY        3
                     DUAL_DISPLAY                4
                     REFERENCE_DISPLAY           5
                     CHARTING_DISPLAY            6
```

```
                    UNKNOWN_DISPLAY            -1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_RANGE_GET – value retrieved is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

7.18 API8810A_ResetDefaultValues

Format:

```
_API8810AFUNC int API8810A_ResetDefaultValues
(
   int apiNo
)
```

Function Description:

This function sends the command to set the device setting back to the factory default settings. This command will also reset the calibration values and a calibration of the device is recommended after issuing this command. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_RANGE– data returned from 8810A is not valid for command sent
```

References for this function:

None.

8 API-8810A Calibration Routines

The routines in this section handle sending commands to calibrate the 8810A device and retrieve the calibration state of the 8810A device.

8.1 API8810A_GetCalState

Format:

```
_API8810AFUNC int API8810A_GetCalState
(
   int apiNo,
   char *pszCalState
)
```

Function Description:

This function sends the command to get the calibration state of the 8810A.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszCalState – pointer to location to return the calibration state string
   Return values:
      "CAL DONE"
      "CALIBRATING"

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

8.2 API8810A_Calibrate

Format:

```
_API8810AFUNC int API8810A_Calibrate
(
   int apiNo
)
```

Function Description:

This function sends the command to calibrate the 8810A. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

8.3 API8810A_SetAPIPeriodicCalState

Format:

```
_API8810AFUNC int API8810A_SetAPIPeriodicCalState
(
   int apiNo,
   int nPeriodicCalState
)
```

Function Description:

This function sends the command to turn on or off the periodic calibration for both channels. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nPeriodicCalState – Turn on or off periodic calibration for both channels
                  Periodic Calibration Types:
                     PERIODIC_CAL_DISABLED     0
                     PERIODIC_CAL_ENABLED      1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

8.4 API8810A_GetAPIPeriodicCalState

Format:

```
_API8810AFUNC int API8810A_GetAPIPeriodicCalState
(
   int apiNo,
   int *pnPeriodicCalState
)
```

Function Description:

This function sends the command to get the periodic calibration setting.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnPeriodicCalState – pointer to location to return the periodic
     calibration state
                  Periodic Calibration Types:
                     PERIODIC_CAL_DISABLED     0
                     PERIODIC_CAL_ENABLED      1
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

9 API-8810A Buffer Routines

The routines in this section handle sending commands to set and get the configuration for channel buffering on the 8810A.

9.1 API8810A_GetSampleRate

Format:

```
_API8810AFUNC int API8810A_GetSampleRate
(
   int apiNo,
   int *pnSampleRate,
   int *pnSampleRateUnit
)
```

Function Description:

This function sends the command to get the sample rate set for angle, angle error, or angle velocity buffering in the 8810A.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnSampleRate – pointer to location to return the sample rate
pnSampleRateUnit – pointer to location to return the sample rate units
                  Sample Rate Unit Types:
                     SAMPLE_RATE_MSEC       0
                     SAMPLE_RATE_SEC        1
                     SAMPLE_RATE_MIN        2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

9.2 API8810A_SetSampleRate

Format:

```
_API8810AFUNC int API8810A_SetSampleRate
(
   int apiNo,
   int nSampleRate,
   int nSampleRateUnit
)
```

Function Description:

This function sends the command to set the sample rate set for angle, angle error, or angle velocity buffering in the 8810A. Note the 8810A

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

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nSampleRate – Sample Rate to set for data buffering
nSampleRateUnit – Sample Rate Unit to set for data buffering
                  Sample Rate Unit Types:
                     SAMPLE_RATE_MSEC       0
                     SAMPLE_RATE_SEC        1
                     SAMPLE_RATE_MIN        2
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

9.3 API8810A_GetSampleType

Format:

```
_API8810AFUNC int API8810A_GetSampleType
(
   int apiNo,
   char *pszSampleType
)
```

Function Description:

```
This function sends the command to get the data type being buffering in
the 8810A.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszSampleType – pointer to location to return the data type being buffered
   Return values:
         "ANG"
         "ANGERR"
         "VEL"
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

9.4 API8810A_SetSampleType

Format:

```
_API8810AFUNC int API8810A_SetSampleType
(
   int apiNo,
   int nSampleType
)
```

Function Description:

This function sends the command to set the data type to buffer in the 8810A. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nSampleType – Sample Rate to set for data buffering

Sample Data Types:	
BUF_ANGLE	0
BUF_ANGERR	1
BUF_VELOCITY	2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

9.5 API8810A_GetPlotChan

Format:

```
_API8810AFUNC int API8810A_GetPlotChan
(
   int apiNo,
   char *pszPlotChan
)
```

Function Description:

This function sends the command to get the channel being plotted in the 8810A chart.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszPlotChan – pointer to location to return the channel being plotted
Return values:
"BOTH"
"CH1"

"CH2"

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

9.6 API8810A_SetPlotChan

Format:

```
_API8810AFUNC int API8810A_SetPlotChan
(
   int apiNo,
   int nPlotChan
)
```

Function Description:

This function sends the command to set the channel to plot in the 8810A. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nPlotChan – Channel to plot

Channel Plot Types:	
PLOT_BOTH_CHAN	0
PLOT_CHAN1	1
PLOT_CHAN2	2

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
     API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

9.7 API8810A_GetAngleErrStep

Format:

```
_API8810AFUNC int API8810A_GetAngleErrStep
(
   int apiNo,
```

```
   double *pdAngErrStep
)
```

Function Description:

This function sends the command to get the angle step value for angle error comparison for 8810A data buffering.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdAngErrStep – pointer to location to return the angle step value for angle error comparison for 8810A data buffering.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

9.8 API8810A_SetAngleErrStep

Format:

```
_API8810AFUNC int API8810A_SetAngleErrStep
(
   int apiNo,
   double dAngErrStep
)
```

Function Description:

This function sends the command to set the angle step value for angle error comparison for 8810A data buffering. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
dAngErrStep – Angle step value for angle error comparison for data buffering

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

9.9 API8810A_GetLowerRange

Format:

```
_API8810AFUNC int API8810A_GetLowerRange
(
   int apiNo,
   double *pdLowerRange
)
```

Function Description:

This function sends the command to get the expected lower range value for 8810A charting.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdLowerRange – pointer to location to return expected lower range value for 8810A charting.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

9.10 API8810A_SetLowerRange

Format:

```
_API8810AFUNC int API8810A_SetLowerRange
(
   int apiNo,
   double dLowerRange
)
```

Function Description:

This function sends the command to set the expected lower range value for 8810A charting. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
dLowerRange – Expected lower range value for 8810A charting

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE

```
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

```
None.
```

9.11 API8810A_GetUpperRange

Format:

```
_API8810AFUNC int API8810A_GetUpperRange
(
   int apiNo,
   double *pdUpperRange
)
```

Function Description:

```
This function sends the command to get the expected upper range value for
8810A charting.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pdUpperRange – pointer to location to return expected upper range value
   for 8810A charting.
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

```
None.
```

9.12 API8810A_SetUpperRange

Format:

```
_API8810AFUNC int API8810A_SetUpperRange
(
   int apiNo,
   double dUpperRange
)
```

Function Description:

```
This function sends the command to set the expected upper range value for
8810A charting. Note the 8810A device will not accept the command if its
remote communication configuration does not match the communication
connection mode.
```

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
dUpperRange – Expected upper range value for 8810A charting
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
```

References for this function:

None.

9.13 API8810A_GetRecordingState

Format:

```
_API8810AFUNC int API8810A_GetRecordingState
(
   int apiNo,
   char *pszRecState
)
```

Function Description:

This function sends the command to get the recording state for 8810A data buffering.

Parameters:

```
apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pszRecState – pointer to location to return the recording state for 8810A
   data buffering
   Return values:
         "NOT RECORDING"
         "RECORDING"
```

Return Value:

```
API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with
    API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent
```

References for this function:

None.

9.14 API8810A_SetRecordingState

Format:

```
_API8810AFUNC int API8810A_SetRecordingState
(
   int apiNo,
   int nRecState
)
```

Function Description:

This function sends the command to start recording, stop recording or clear the data buffer. Note the 8810A device will not accept the command if its remote communication configuration does not match the communication connection mode.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nRecState – Buffer Recording command state

Buffer Recording Command Types:

BUFFER_REC_STOP	0
BUFFER_REC_START	1
BUFFER_REC_CLEAR	2

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_RANGE_SET – parameter specified is out-of-range
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

9.15 API8810A_GetBufferCnt

Format:

```
_API8810AFUNC int API8810A_GetBufferCnt
(
   int apiNo,
   int *pnBufCnt
)
```

Function Description:

This function sends the command to get the number of data elements in 8810A data buffer.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
pnBufCnt – pointer to location to return the number of data elements in data buffer.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

9.16 API8810A_GetBufferData

Format:

```
_API8810AFUNC int API8810A_GetBufferData
(
   int apiNo,
   int nChannel,
   int nStartRec,
   int nEndRec,
   char *pszBufData
)
```

Function Description:

This function sends the command to get data elements in 8810A data buffer. The number of data elements is the same for Channel 1 and Channel 2. Note data buffer retrieval is available only via USB or Ethernet. Via USB, the maximum number of records returned for each call is 5 data elements. Via Ethernet, the maximum number of records returned for each call is 150 data elements.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
nChannel – Channel to retrieve data
nStartRec – Record number (1 for the first record) to first element to retrieve
nEndRec – Record number of last element to retrieve
pszBufData – pointer to location to return data elements in data buffer that has been retrieved.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_FUNC_NOT_SUPPORTED – function supported only with API8810A_NATIVE
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

Prior to calling this function, call API8810A_GetBufferCnt() to determine the number of elements in the data buffer. The number of elements is the same for Channel 1 and Channel 2.

Remarks:

Data retrieval of buffered data from the 8810A is available only via the USB (5 data elements maximum with each call) or Ethernet (150 data elements maximum with each call) interface.

10 API-8810A Miscellaneous Routines

The routines in this section handle setting or retrieving information from the API8810ADll and sending freeform commands and queries to the 8810A device.

10.1 API8810A_MaxRetry

Format:

```
_API8810AFUNC int API8810A_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:

API_SUCCESS - function is successful

References for this function:

None.

10.2 API8810A_LastCmdSent

Format:

```
_API8810AFUNC int API8810A_LastCmdSent
(
   int apiNo,
   char szLastCommand[]
)
```

Function Description:

This function returns the last command sent via IEEE, USB or Ethernet to the 8810A device.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
szLastCommand – last command sent to 8810A

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter

References for this function:

None.

10.3 API8810A_WriteCommand

Format:

```
_API8810AFUNC int API8810A_WriteCommand
(
   int apiNo,
   char szCommand[]
)
```

Function Description:

This function sends the command to the 8810A device.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
szCommand – command to send to 8810A

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A

References for this function:

None.

10.4 API8810A_QueryCommand

Format:

```
_API8810AFUNC int API8810A_QueryCommand
(
   int apiNo,
   char szCommand[],
   char *pszResponse
)
```

Function Description:

This function sends the command to the 8810A device and waits for the 8810A to respond.

Parameters:

apiNo - Logical API number assigned to connection with 8810A (1-MAX_API)
szCommand – command to send to 8810A
pszResponse – pointer to location to return the 8810A response to the command sent.

Return Value:

API_SUCCESS - function is successful
API_ERROR_APINO - invalid apiNo parameter
API_ERROR_WRITE - unable to send command to 8810A
API_ERROR_DATA – data returned from 8810A is not valid for command sent

References for this function:

None.

11 Appendix A – API8810ADll Constant Values

```
/* Maximum number of APIs Driver can communicate with */
#define  MAX_API 12

/* API Language Types */
#define API8810A_NATIVE          0
#define IEEE_API8810_NATIVE      1
#define IEEE_API8810_SR103       2
#define IEEE_API8810_HSR202      3
#define IEEE_API8810_HSR203      4
#define IEEE_API8810_MATECIIL    5
#define IEEE_API8810_FX2         6

/* API 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

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

/* Internal Reference Output State Type */
#define INT_REF_OUT_NOT_AVAILABLE   0
#define INT_REF_OUT_AVAILABLE       1

/* Angle Display Format Type */
#define ANGLE_FMT_360               0
#define ANGLE_FMT_180               1
#define ANGLE_FMT_MIN               2

/* Angle Limit Format Type */
#define ANGLE_LIMIT_ABS_ANG         0
#define ANGLE_LIMIT_ANG_ERR         1

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

/* Channel 1 Input Type */
#define CH1_INPUT_FRONT_CONNECTOR   0
#define CH1_INPUT_BACK_CONNECTOR    1

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

/* Angle Difference Display State Type */
#define ANG_DIFF_DISABLED           0
```

```
#define ANG_DIFF_ENABLED            1

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

/* Buffer Data Type */
#define BUF_ANGLE                   0
#define BUF_ANGERR                  1
#define BUF_VELOCITY                2

/* Channel Plot Type */
#define PLOT_BOTH_CHAN              0
#define PLOT_CHAN1                  1
#define PLOT_CHAN2                  2

/* Sample Rate Unit Type */
#define SAMPLE_RATE_MSEC            0
#define SAMPLE_RATE_SEC             1
#define SAMPLE_RATE_MIN             2

/* Buffer Recording Command Type */
#define BUFFER_REC_STOP             0
#define BUFFER_REC_START            1
#define BUFFER_REC_CLEAR            2

/* Display Command Type */
#define CHAN1_DISPLAY               0
#define CHAN2_DISPLAY               1
#define CHAN1_ANALOG_DISPLAY        2
#define CHAN2_ANALOG_DISPLAY        3
#define DUAL_DISPLAY                4
#define REFERENCE_DISPLAY           5
#define CHARTING_DISPLAY            6
#define UNKNOWN_DISPLAY            -1
```

12 Appendix B- Error Codes

Error Mnemonic	Value	Meaning
API_SUCCESS	0	Function is successful
API_ERROR_OPEN_API_SESSION	1	IEEE connection or configuration error
API_ERROR_APINO	2	Invalid apiNo parameter
API_ERROR_ADDRS	3	Invalid IEEE Address parameter
API_ERROR_LANG	4	Invalid 8810A Language parameter
API_ERROR_DATA	5	Data returned from 8810A is not valid for command sent
API_ERROR_RANGE_SET	6	Parameter specified is out-of-range
API_ERROR_RANGE_GET	7	Value retrieved is out-of-range
API_ERROR_WRITE	8	Unable to send command to 8810A
API_ERROR_USB_CONNECTION	9	USB connection error
API_ERROR_ETHER_CONNECTION	10	Ethernet connection error
API_ERROR_FUNC_NOT_SUPPORTED	11	Function not support in selected language for 8810A communication

Revision History

Revision ID	Revision Date	Description	Author
2.0.0.2	Dec 21,2007	Initial Release	gc
2.0.0.3	May 30, 2008	Added API8810A_SetAPIPeriodicCalState and API8810A_GetAPIPeriodicCalState APIs	gc
2.0.0.4	July 31, 2008	Modified the API Soft Panel Screen (p42) to include the additional button that allows the user to read and log the channels' angle data directly to the specified file	gc
2.0.0.15	Mar 30, 2009	Added analog display mode, ability to adjust resolution of angle display, output angles in radians, mil-radians, angle offset option and recall of channel display (ch 1, ch 2 or dual) after power off.	gc
2.0.0.18	Oct 23, 2009	Added API8810A_GetIntRefOvercurrentState and API8810A_ResetIntRefOvercurrentState	gc
2.0.0.19	Mar 17, 2010	For SR103, HSR202, and HSR203, the return angle does not begin with the '<' character, only the 8810A NATIVE has the '<' character.	gc
2.0.0.20	Apr 08, 2010	Added 8810FX2 mode to support automatic setup for the 8810A to the 8810-FX2 configuration	gc
2.0.0.21	Jul 01, 2010	Fixed problem with the IEEE address and Ethernet addresses getting set to factory default values for the *RST? and APICMD RSTFRAM commands.	gc
3.0.0.1	Oct 03, 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 Cypress USB Driver.	gc