

SerialComm_LIB

Generated by Doxygen 1.8.20

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 SerialFileInfoStruct Struct Reference	5
4 File Documentation	7
4.1 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/SerialComm_LIB.c File Reference	7
4.1.1 Detailed Description	9
4.1.2 Macro Definition Documentation	10
4.1.2.1 insertTableColumn	10
4.1.3 Function Documentation	10
4.1.3.1 AddRowSerialConfigTableCB()	10
4.1.3.2 CloseSerialDevice()	11
4.1.3.3 CreateSerialConfigurationTable()	11
4.1.3.4 CreateSerialDebugPanel()	11
4.1.3.5 DisplayRS232Error()	11
4.1.3.6 FlushInQDevice()	12
4.1.3.7 FlushOutQDevice()	12
4.1.3.8 GetDeviceName()	12
4.1.3.9 getFileInfoIndexFromName()	13
4.1.3.10 GetInQLenForDeviceName()	13
4.1.3.11 GetSerialConfigurationPanelHandle()	13
4.1.3.12 InitializeSerialPortLib()	14
4.1.3.13 InitSerialDevice()	14
4.1.3.14 IsValidXMLSerial()	14
4.1.3.15 ReadSerialConfigurationFile()	14
4.1.3.16 ReadSerialDevice()	15
4.1.3.17 ReadSerialDeviceUntilTermChar()	15
4.1.3.18 SaveSerialConfigTableCB()	16
4.1.3.19 WriteSerialDevice()	16
4.1.3.20 WriteSerialDeviceRaw()	16
4.2 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/SerialComm_LIB.h File Reference	17
4.2.1 Detailed Description	18
4.2.2 Function Documentation	18
4.2.2.1 CloseSerialDevice()	18
4.2.2.2 FlushInQDevice()	19
4.2.2.3 FlushOutQDevice()	19
4.2.2.4 GetDeviceName()	19

4.2.2.5 GetInQLenForDeviceName()	20
4.2.2.6 GetSerialConfigurationPanelHandle()	20
4.2.2.7 InitializeSerialPortLib()	20
4.2.2.8 InitSerialDevice()	21
4.2.2.9 ReadSerialConfigurationFile()	21
4.2.2.10 ReadSerialDevice()	21
4.2.2.11 ReadSerialDeviceUntilTermChar()	22
4.2.2.12 WriteSerialDevice()	22
4.2.2.13 WriteSerialDeviceRaw()	23

Index	25
--------------	-----------

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

SerialFileInfoStruct	5
--	---

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/ SerialComm_LIB.c	
Serial communication library	7
C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/ SerialComm_LIB.h . . .	17

Chapter 3

Class Documentation

3.1 SerialFileInfoStruct Struct Reference

Public Attributes

- char **DeviceName** [MAXCHARARRAYLENGTH]
- char **Comport** [MAXCHARARRAYLENGTH]
- char **BaudRate** [MAXCHARARRAYLENGTH]
- char **Parity** [MAXCHARARRAYLENGTH]
- char **DataBits** [MAXCHARARRAYLENGTH]
- char **StopBits** [MAXCHARARRAYLENGTH]
- char **CTSMMode** [MAXCHARARRAYLENGTH]
- char **XonXoff** [MAXCHARARRAYLENGTH]
- char **Timeout** [MAXCHARARRAYLENGTH]
- int **PortOpen**

The documentation for this struct was generated from the following file:

- C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/[SerialComm_LIB.h](#)

Chapter 4

File Documentation

4.1 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/SerialComm_LIB.c File Reference

Serial communication library.

```
#include "cvixml.h"
#include <ansi_c.h>
#include <userint.h>
#include <utility.h>
#include <formatio.h>
#include "SerialComm_LIB.h"
#include <rs232.h>
```

Macros

- #define **insertTableColumn**(colIndex, paramIndex, cellType, colWidth)

Functions

- void **CreateSerialConfigurationTable** (int MainPanelHandle)
Create the Serial configuration table. Read from the specified file and populates the Serial config table.
- void **CreateSerialDebugPanel** (int MainPanelHandle)
Create the Serial debug Panel used to manually turn on and off signals.
- void **PopulateComboBox** (int RowNum)
Populates the combo box for newly added row.
- void **LoadSerialConfigFile** (void)
This function takes the data previously loaded into Serial struct and populates the test configuration table for each com port opened.
- int CVICALLBACK **SaveSerialConfigTableCB** (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback for the Save button in Serial Configuration panel.

- int CVICALLBACK [SerialTableCBFunction](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Serial config table cb function for toggling between input and output.
- int CVICALLBACK [QuitSerialConfigTableCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback function for Quit/hide Serial Config panel.
- int CVICALLBACK [DelRowSerialConfigTableCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback function for delete row button. Delete the selected test row from Serial configuration table.
- int CVICALLBACK [AddRowSerialConfigTableCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback function for add test button. Adds a test row below the selected row in the Serial configuration table.
- int CVICALLBACK [InitSerialDebugCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback for Init Serial port button on Serial Debug Panel.
- int CVICALLBACK [QuitSerialDebugCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback for Quit/hide on Serial Debug panel.
- int CVICALLBACK [WriteSerialDebugCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback for Write Serial port button on Serial Debug panel.
- int CVICALLBACK [ReadSerialDebugCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback for Read Serial port button on Serial Debug Panel.
- int CVICALLBACK [ClearSerialDebugCB](#) (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
Callback for Clear Serial port button on Serial Debug Panel.
- void [DisplayRS232Error](#) (int RS232Error)
Display help information to the user based on RS232Error code.
- int [getFileIndexFromName](#) (char *DeviceName)
returns the index in the glbSerialFileInfo array for the deviceName
- int [SerialReadThread](#) (void *dummy)
Serial Read thread to continuously read the buffer.
- int [SaveBackupXmlFilenameSerial](#) (const char *filename)
Save backup copy of the existing Serial config xml file.
- int [HexToCharInString](#) (char *string)
- int [IsValidXMLSerial](#) (char *string)
Check if string is a valid XML tag.
- int [InitializeSerialPortLib](#) (char *SerialConfigurationFile, int MainPanelHandle, char errmsg[ERRLEN])
Initialize the serial library with configuration file.
- int [ReadSerialConfigurationFile](#) (char *filePath)
read the xml Serial configuration from specified path and populate glbSerialFileInfo structure
- int [GetSerialConfigurationPanelHandle](#) (void)
Get SerialConfiguration panel handle.
- int [GetSerialDebugPanelHandle](#) (void)
Get SerialDebug panel handle.
- int [GetDeviceName](#) (int index, char *devName, char errmsg[ERRLEN])
Get device name of specified index (0based)
- int [GetTotalSerialDevices](#) (void)
Get number of devices in SerialInfo.
- int [InitSerialDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Initialize a Serial Device.

- int [CloseSerialDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Close a Serial Device.
- int [WriteSerialDevice](#) (char *SerialDeviceName, char *data, char errmsg[ERRLEN])
Write To a specified serial device.
- int [WriteSerialDeviceRaw](#) (char *SerialDeviceName, char *data, int dataLen, char errmsg[ERRLEN])
Write To specified serial device with specified data length.
- int [ReadSerialDevice](#) (char *SerialDeviceName, char *ReadData, int numByteToRead, char errmsg[ERRLEN])
Read from specified serial device.
- int [ReadSerialDeviceUntilTermChar](#) (char *SerialDeviceName, char *ReadData, int numByteToRead, int terminationByte, char errmsg[ERRLEN])
Read from specified serial device until termination char is found or timeout or number of byte to read is reached.
- int [GetInQLenForDeviceName](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Get the in queue length for a specified device.
- char * [getSerialLibRevision](#) (void)
get library revision
- int **ReadMotor** (char *Motor)
- int [FlushInQDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Flush the IN queue for the specified serial device COM port.
- int [FlushOutQDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Flush the OUT queue for the specified serial device COM port.

Variables

- static int **libInitialized** = 0
- char **tempChecksum** [4] = {0}
- char **glbPathToSerialConfigFile** [256] = {0}
- int **glbSerialConfigurationPanelHandle** = 0
- int **glbSerialConfigTableHandle** = 0
- int **glbSerialDebugPanelHandle**
- int **glbNumOfComPorts** = 0
- int **glbSerialRingDebugMenuHandle**
- int **glbWriteBoxHandle**
- int **glbSerialThreadID** = 0
- int **glbSerialThread** = 0
- static int **glbSerialReadThreadHandle**
- int **glbReadBoxHandle**
- [SerialFileInfoStruct](#) **glbSerialFileInfo** [MAXNUMOFSERIALPORTS] = {0}
- char **glbSerialParamName** [9][20] = {"DeviceName","Comport","BaudRate","Parity","DataBits","StopBits","CTSMODE","XonXoff","Timeout"}

4.1.1 Detailed Description

Serial communication library.

Author

Arxtron

Copyright

Arxtron Technologies Inc. All Rights Reserved.

Date

11/9/2020

Library used to communicate to serial devices. Device configuration is loaded by an XML file. A panel for the configuration table and debug is created on initialization of this library and can be used by accessing the panel handles stored as global variables.

Version	Date	Author	Description
1.0.0	May 5, 2014	Arxtron	Initial Release
1.0.1	Nov 9, 2020	Jai Prajapati	Updated with library format

4.1.2 Macro Definition Documentation**4.1.2.1 insertTableColumn**

```
#define insertTableColumn(
    colIndex,
    paramIndex,
    cellType,
    colWidth )
```

Value:

```
InsertTableColumns (glbSerialConfigurationPanelHandle, glbSerialConfigTableHandle, colIndex, 1,
cellType);\
SetTableColumnAttribute (glbSerialConfigurationPanelHandle, glbSerialConfigTableHandle, colIndex,
ATTR_LABEL_TEXT, glbSerialParamName[paramIndex]);\
SetTableColumnAttribute (glbSerialConfigurationPanelHandle, glbSerialConfigTableHandle, colIndex,
ATTR_USE_LABEL_TEXT, 1);\
SetTableColumnAttribute (glbSerialConfigurationPanelHandle, glbSerialConfigTableHandle, colIndex,
ATTR_COLUMN_WIDTH, colWidth);\
SetTableColumnAttribute (glbSerialConfigurationPanelHandle, glbSerialConfigTableHandle, colIndex,
ATTR_CELL_TYPE, cellType)
```

4.1.3 Function Documentation**4.1.3.1 AddRowSerialConfigTableCB()**

```
int CVICALLBACK AddRowSerialConfigTableCB (
    int panel,
    int control,
    int event,
    void * callbackData,
    int eventData1,
    int eventData2 )
```

Callback function for add test button. Adds a test row below the selected row in the Serial configuration table.

REGION START CVI Callbacks

4.1.3.2 CloseSerialDevice()

```
int CloseSerialDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Close a Serial Device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to close
----	-------------------------	--------------------------------

4.1.3.3 CreateSerialConfigurationTable()

```
void CreateSerialConfigurationTable (
    int MainPanelHandle )
```

Create the Serial configuration table. Read from the specified file and populates the Serial config table.

Parameters

in	<i>MainPanelHandle</i>	Parent panel handle to create serial child panel under
----	------------------------	--

4.1.3.4 CreateSerialDebugPanel()

```
void CreateSerialDebugPanel (
    int MainPanelHandle )
```

Create the Serial debug Panel used to manually turn on and off signals.

Parameters

in	<i>MainPanelHandle</i>	Parent panel handle to create debug panel under
----	------------------------	---

4.1.3.5 DisplayRS232Error()

```
void DisplayRS232Error (
    int RS232Error )
```

Display help information to the user based on RS232Error code.

Parameters

in	<i>RS232Error</i>	Error code
----	-------------------	------------

4.1.3.6 FlushInQDevice()

```
int FlushInQDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Flush the IN queue for the specified serial device COM port.

Parameters

in	<i>SerialDeviceName</i>	Name of device
----	-------------------------	----------------

4.1.3.7 FlushOutQDevice()

```
int FlushOutQDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Flush the OUT queue for the specified serial device COM port.

Parameters

in	<i>SerialDeviceName</i>	Name of device
----	-------------------------	----------------

4.1.3.8 GetDeviceName()

```
int GetDeviceName (
    int index,
    char * devName,
    char errmsg[ERRLEN] )
```

Get device name of specified index (0based)

Parameters

in	<i>index</i>	Index of serial file information
out	<i>devName</i>	Serial device name

4.1.3.9 getFileInfoIndexFromName()

```
int getFileInfoIndexFromName (
    char * DeviceName )
```

returns the index in the glbSerialFileInfo array for the deviceName

Parameters

in	<i>DeviceName</i>	Name of device to find
----	-------------------	------------------------

4.1.3.10 GetInQLenForDeviceName()

```
int GetInQLenForDeviceName (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Get the in queue length for a specified device.

Parameters

in	<i>SerialDeviceName</i>	Name of device
----	-------------------------	----------------

Returns

the in queue length or error code

- 0 -> COM Port status error

4.1.3.11 GetSerialConfigurationPanelHandle()

```
int GetSerialConfigurationPanelHandle (
    void )
```

Get SerialConfiguration panel handle.

Returns

glbSerialConfigurationPanelHandle

4.1.3.12 InitializeSerialPortLib()

```
int InitializeSerialPortLib (
    char * SerialConfigurationFile,
    int MainPanelHandle,
    char errmsg[ERRLEN] )
```

Initialize the serial library with configuration file.

Serial Port configuration table panel and the Serial debug panel are created

Parameters

in	<i>SerialConfigurationFile</i>	Path to config XML file
in	<i>MainPanelHandle</i>	Parent panel handle to create serial child panels under

4.1.3.13 InitSerialDevice()

```
int InitSerialDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Initialize a Serial Device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to initialize
----	-------------------------	-------------------------------------

4.1.3.14 IsValidXMLSerial()

```
int IsValidXMLSerial (
    char * string )
```

Check if string is a valid XML tag.

Parameters

in	<i>string</i>	
----	---------------	--

4.1.3.15 ReadSerialConfigurationFile()

```
int ReadSerialConfigurationFile (
    char * filePath )
```

read the xml Serial configuration from specified path and populate glbSerialFileInfo structure

Parameters

in	<i>filePath</i>	Path to serial configuration XML file
----	-----------------	---------------------------------------

Returns

The number of serial devices found in XML file

4.1.3.16 ReadSerialDevice()

```
int ReadSerialDevice (
    char * SerialDeviceName,
    char * ReadData,
    int numByteToRead,
    char errmsg[ERRLEN] )
```

Read from specified serial device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to read from
out	<i>ReadData</i>	
in	<i>numBytesToRead</i>	Number of bytes to read

Returns

The number of bytes read or negative errorcode

4.1.3.17 ReadSerialDeviceUntilTermChar()

```
int ReadSerialDeviceUntilTermChar (
    char * SerialDeviceName,
    char * ReadData,
    int numByteToRead,
    int terminationByte,
    char errmsg[ERRLEN] )
```

Read from specified serial device until termination char is found or timeout or number of byte to read is reached.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to read from
out	<i>ReadData</i>	
in	<i>numBytesToRead</i>	Number of bytes to read
in	<i>terminationByte</i>	Termination char to stop reading

Returns

The number of bytes read or negative error code

4.1.3.18 SaveSerialConfigTableCB()

```
int CVICALLBACK SaveSerialConfigTableCB (
    int panel,
    int control,
    int event,
    void * callbackData,
    int eventData1,
    int eventData2 )
```

Callback for the Save button in Serial Configuration panel.

This function create a backup of the current serial Configuration xml file and saves the new config into the file.

4.1.3.19 WriteSerialDevice()

```
int WriteSerialDevice (
    char * SerialDeviceName,
    char * data,
    char errmsg[ERRLEN] )
```

Write To a specified serial device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to write to
in	<i>data</i>	Data to write

Returns

The number of bytes written or negative error code

4.1.3.20 WriteSerialDeviceRaw()

```
int WriteSerialDeviceRaw (
    char * SerialDeviceName,
    char * data,
    int dataLen,
    char errmsg[ERRLEN] )
```

Write To specified serial device with specified data length.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to write to
in	<i>data</i>	Data to write
in	<i>dataLen</i>	Length of data

Returns

The number of bytes written or negative error code

4.2 C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/SerialComm_LIB.h File Reference

```
#include "cstddef.h"
#include "ArxtronToolslib.h"
```

Classes

- struct [SerialFileInfoStruct](#)

Macros

- #define **MAXNUMOFSERIALPORTS** 50
- #define **MAXCHARARRAYLENGTH** 400
- #define **SERIALLIBREV** "1.0.1"

Functions

- void **GetStandardErrMsg** (int error, char errmsg[ERRLEN])
- int CVICALLBACK **FunctionSelect** (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
- int CVICALLBACK **RunFunction** (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
- int [InitializeSerialPortLib](#) (char *SerialConfigurationFile, int MainPanelHandle, char errmsg[ERRLEN])
Initialize the serial library with configuration file.
- int [ReadSerialConfigurationFile](#) (char *filePath)
read the xml Serial configuration from specified path and populate glbSerialFileInfo structure
- int [GetSerialConfigurationPanelHandle](#) (void)
Get SerialConfiguration panel handle.
- int [GetSerialDebugPanelHandle](#) (void)
Get SerialDebug panel handle.
- int [InitSerialDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Initialize a Serial Device.
- int [CloseSerialDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Close a Serial Device.

- int [WriteSerialDevice](#) (char *SerialDeviceName, char *data, char errmsg[ERRLEN])
Write To a specified serial device.
- int [WriteSerialDeviceRaw](#) (char *SerialDeviceName, char *data, int dataLen, char errmsg[ERRLEN])
Write To specified serial device with specified data length.
- int [ReadSerialDevice](#) (char *SerialDeviceName, char *ReadData, int numByteToRead, char errmsg[ERRLEN])
Read from specified serial device.
- int [ReadSerialDeviceUntilTermChar](#) (char *SerialDeviceName, char *ReadData, int numByteToRead, int terminationByte, char errmsg[ERRLEN])
Read from specified serial device until termination char is found or timeout or number of byte to read is reached.
- int [FlushInQDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Flush the IN queue for the specified serial device COM port.
- int [FlushOutQDevice](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Flush the OUT queue for the specified serial device COM port.
- char * [getSerialLibRevision](#) (void)
get library revision
- int [GetTotalSerialDevices](#) (void)
Get number of devices in SerialInfo.
- int [GetDeviceName](#) (int index, char *devName, char errmsg[ERRLEN])
Get device name of specified index (0based)
- int [GetInQLenForDeviceName](#) (char *SerialDeviceName, char errmsg[ERRLEN])
Get the in queue length for a specified device.

4.2.1 Detailed Description

Author

Arxtron

Copyright

Arxtron Technologies Inc. All Rights Reserved.

Date

11/9/2020 8:54:00 AM

4.2.2 Function Documentation

4.2.2.1 CloseSerialDevice()

```
int CloseSerialDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Close a Serial Device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to close
----	-------------------------	--------------------------------

4.2.2.2 FlushInQDevice()

```
int FlushInQDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Flush the IN queue for the specified serial device COM port.

Parameters

in	<i>SerialDeviceName</i>	Name of device
----	-------------------------	----------------

4.2.2.3 FlushOutQDevice()

```
int FlushOutQDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Flush the OUT queue for the specified serial device COM port.

Parameters

in	<i>SerialDeviceName</i>	Name of device
----	-------------------------	----------------

4.2.2.4 GetDeviceName()

```
int GetDeviceName (
    int index,
    char * devName,
    char errmsg[ERRLEN] )
```

Get device name of specified index (0based)

Parameters

in	<i>index</i>	Index of serial file information
out	<i>devName</i>	Serial device name

4.2.2.5 GetInQLenForDeviceName()

```
int GetInQLenForDeviceName (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Get the in queue length for a specified device.

Parameters

in	<i>SerialDeviceName</i>	Name of device
----	-------------------------	----------------

Returns

the in queue length or error code

- 0 -> COM Port status error

4.2.2.6 GetSerialConfigurationPanelHandle()

```
int GetSerialConfigurationPanelHandle (
    void )
```

Get SerialConfiguration panel handle.

Returns

glbSerialConfigurationPanelHandle

4.2.2.7 InitializeSerialPortLib()

```
int InitializeSerialPortLib (
    char * SerialConfigurationFile,
    int MainPanelHandle,
    char errmsg[ERRLEN] )
```

Initialize the serial library with configuration file.

Serial Port configuration table panel and the Serial debug panel are created

Parameters

in	<i>SerialConfigurationFile</i>	Path to config XML file
in	<i>MainPanelHandle</i>	Parent panel handle to create serial child panels under

4.2.2.8 InitSerialDevice()

```
int InitSerialDevice (
    char * SerialDeviceName,
    char errmsg[ERRLEN] )
```

Initialize a Serial Device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to initialize
----	-------------------------	-------------------------------------

4.2.2.9 ReadSerialConfigurationFile()

```
int ReadSerialConfigurationFile (
    char * filePath )
```

read the xml Serial configuration from specified path and populate glbSerialFileInfo structure

Parameters

in	<i>filePath</i>	Path to serial configuration XML file
----	-----------------	---------------------------------------

Returns

The number of serial devices found in XML file

4.2.2.10 ReadSerialDevice()

```
int ReadSerialDevice (
    char * SerialDeviceName,
    char * ReadData,
    int numByteToRead,
    char errmsg[ERRLEN] )
```

Read from specified serial device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to read from
out	<i>ReadData</i>	
in	<i>numBytesToRead</i>	Number of bytes to read

Returns

The number of bytes read or negative errorcode

4.2.2.11 ReadSerialDeviceUntilTermChar()

```
int ReadSerialDeviceUntilTermChar (
    char * SerialDeviceName,
    char * ReadData,
    int numByteToRead,
    int terminationByte,
    char errmsg[ERRLEN] )
```

Read from specified serial device until termination char is found or timeout or number of byte to read is reached.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to read from
out	<i>ReadData</i>	
in	<i>numBytesToRead</i>	Number of bytes to read
in	<i>terminationByte</i>	Termination char to stop reading

Returns

The number of bytes read or negative error code

4.2.2.12 WriteSerialDevice()

```
int WriteSerialDevice (
    char * SerialDeviceName,
    char * data,
    char errmsg[ERRLEN] )
```

Write To a specified serial device.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to write to
in	<i>data</i>	Data to write

Returns

The number of bytes written or negative error code

4.2.2.13 WriteSerialDeviceRaw()

```
int WriteSerialDeviceRaw (
    char * SerialDeviceName,
    char * data,
    int dataLen,
    char errmsg[ERRLEN] )
```

Write To specified serial device with specified data length.

Parameters

in	<i>SerialDeviceName</i>	Name of serial device to write to
in	<i>data</i>	Data to write
in	<i>dataLen</i>	Length of data

Returns

The number of bytes written or negative error code

Index

AddRowSerialConfigTableCB
SerialComm_LIB.c, [10](#)

C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/SerialComm_LIB.c,
[7](#)

C:/Users/jai_prajapati/Documents/SourceLibraries/Serial_LIB/SerialComm_LIB/SerialComm_LIB.h,
[17](#)

CloseSerialDevice
SerialComm_LIB.c, [10](#)
SerialComm_LIB.h, [18](#)

CreateSerialConfigurationTable
SerialComm_LIB.c, [11](#)

CreateSerialDebugPanel
SerialComm_LIB.c, [11](#)

DisplayRS232Error
SerialComm_LIB.c, [11](#)

FlushInQDevice
SerialComm_LIB.c, [12](#)
SerialComm_LIB.h, [19](#)

FlushOutQDevice
SerialComm_LIB.c, [12](#)
SerialComm_LIB.h, [19](#)

GetDeviceName
SerialComm_LIB.c, [12](#)
SerialComm_LIB.h, [19](#)

getFileInfoIndexFromName
SerialComm_LIB.c, [13](#)

GetInQLenForDeviceName
SerialComm_LIB.c, [13](#)
SerialComm_LIB.h, [20](#)

GetSerialConfigurationPanelHandle
SerialComm_LIB.c, [13](#)
SerialComm_LIB.h, [20](#)

InitializeSerialPortLib
SerialComm_LIB.c, [13](#)
SerialComm_LIB.h, [20](#)

InitSerialDevice
SerialComm_LIB.c, [14](#)
SerialComm_LIB.h, [21](#)

insertTableColumn
SerialComm_LIB.c, [10](#)

IsValidXMLSerial
SerialComm_LIB.c, [14](#)

ReadSerialConfigurationFile
SerialComm_LIB.c, [14](#)
SerialComm_LIB.h, [21](#)

ReadSerialDevice
SerialComm_LIB.c, [15](#)
SerialComm_LIB.h, [21](#)

ReadSerialDeviceUntilTermChar
SerialComm_LIB.c, [15](#)
SerialComm_LIB.h, [22](#)

SaveSerialConfigTableCB
SerialComm_LIB.c, [16](#)

SerialComm_LIB.c
AddRowSerialConfigTableCB, [10](#)
CloseSerialDevice, [10](#)
CreateSerialConfigurationTable, [11](#)
CreateSerialDebugPanel, [11](#)
DisplayRS232Error, [11](#)
FlushInQDevice, [12](#)
FlushOutQDevice, [12](#)
GetDeviceName, [12](#)
getFileInfoIndexFromName, [13](#)
GetInQLenForDeviceName, [13](#)
GetSerialConfigurationPanelHandle, [13](#)
InitializeSerialPortLib, [13](#)
InitSerialDevice, [14](#)
insertTableColumn, [10](#)
IsValidXMLSerial, [14](#)
ReadSerialConfigurationFile, [14](#)
ReadSerialDevice, [15](#)
ReadSerialDeviceUntilTermChar, [15](#)
SaveSerialConfigTableCB, [16](#)
WriteSerialDevice, [16](#)
WriteSerialDeviceRaw, [16](#)

SerialComm_LIB.h
CloseSerialDevice, [18](#)
FlushInQDevice, [19](#)
FlushOutQDevice, [19](#)
GetDeviceName, [19](#)
GetInQLenForDeviceName, [20](#)
GetSerialConfigurationPanelHandle, [20](#)
InitializeSerialPortLib, [20](#)
InitSerialDevice, [21](#)
ReadSerialConfigurationFile, [21](#)
ReadSerialDevice, [21](#)
ReadSerialDeviceUntilTermChar, [22](#)
WriteSerialDevice, [22](#)
WriteSerialDeviceRaw, [22](#)

SerialFileInfoStruct, [5](#)

WriteSerialDevice
SerialComm_LIB.c, [16](#)
SerialComm_LIB.h, [22](#)

WriteSerialDeviceRaw

SerialComm_LIB.c, [16](#)

SerialComm_LIB.h, [22](#)