Picard Industries PiUsb.dll for the USB Shutter

Version 1.5 – April 25, 2013

Library Overview

The PiUsb library is the software required to write your own programs for use with any Picard Industries USB product.

The PiUsb library can be used from any programming language or application that can call functions in a DLL (Dynamic Link Library). This includes:

- Visual C++
- Visual C#
- Visual Basic
- Delphi

The PiUsb library supports Windows 2000, Windows XP, Windows Vista, Windows 7, and Windows 8. It supports development of 32-bit and 64-bit applications.

How to use the Library

C++:

Copy these files to your project source code folder:

PiUsb.h
PiUsb.lib

PiUsb.h contains the function prototypes for all the functions available in the DLL. It also contains #define statements to define useful constants for your program. You should "#include" this file in any source code file that references the DLL.

PiUsb.lib defines the functions in the DLL. It is used by the linker to resolve the function references.

Copy this file to your executable folder:

PiUsb.dll

If you are building a 64-bit application, use the PiUsb.dll and PiUsb.lib files from the x64 folder.

Building 32-bit vs. 64-bit Applications

The PiUsb library supports development of both 32-bit and 64-bit applications. 32-bit applications can be used on 32-bit or 64-bit versions of Windows, while 64-bit applications can only be used on 64-bit versions of windows.

The files PiUsb.lib and PiUsb.dll are different for 32-bit vs. 64-bit applications.

If you are building a 32-bit application, use the 32 bit versions of the lib and dll files. These can be found in the installation folder of the sample application, typically

C:\Program Files\Picard Industries\USB Quad Shutter

They can also be found on the installation CD, in the folder

Quad Shutter dll and docs\Win32

If you are building a 64-bit application, use the 64 bit versions of the lib and dll files. These can be found in the x64 folder underneath the installation folder of the sample application, typically

C:\Program Files (x86)\Picard Industries\USB Quad Shutter\x64

They can also be found on the installation CD, in the folder

Quad Shutter dll and docs\x64,

The sample program as installed is a 32-bit application. If you wish you can build it as a 64-bit application in Visual C++.

Connect to a Shutter

```
void * __stdcall piConnectShutter(int * ErrorNumber, int SerialNum);
```

Parameters

ErrorNumber

Holds the error number upon completion of the call. Valid values are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

SerialNum

The serial number printed on the Shutter label.

Returns

Returns a pointer to the device. If the ErrorNumber = `PI_DEVICE_NOT_FOUND` then NULL is returned.

Example

```
#include "pi_USB.h"

void * pUsb1;
int ErrorNumber;
int ShutterSerialNumber = 10;  // Serial number from Shutter

      pUsb1 = piConnectShutter(&ErrorNumber,ShutterSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Shutter..." );
      else
            AfxMessageBox( "Shutter Connected." );
```

Disconnect a Shutter

```
void __stdcall piDisconnectShutter(void * devicePtr);
```

Parameters

devicePtr
The device pointer that was returned from the piConnectShutter function.

Example

```
#include “pi_USB.h”

void * pUsb1;
int ErrorNumber;
int ShutterSerialNumber = 10;  // Serial number from Shutter

      pUsb1 = piConnectShutter(&ErrorNumber,ShutterSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Shutter..." );
      else
           AfxMessageBox( "Shutter Connected." );

     if (pUsb1 != NULL)
     {
          piDisconnectShutter(pUsb1);
          pUsb1 = NULL;  // Pointer is invalid after disconnecting
     }
```

Set Shutter State

```
int __stdcall piSetShutterState(int ShutterState, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectShutter function.

ShutterState

The desired state as defined in piUSB.h Valid values are:

```
PI_SHUTTER_OPEN
PI_SHUTTER_CLOSED
```

Returns

Returns the error code. Valid return codes are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

Example

```
#include "pi_USB.h"

void * pUsb1;
int ErrorNumber;
int ShutterSerialNumber = 10;  // Serial number from Shutter


      pUsb1 = piConnectShutter(&ErrorNumber,ShutterSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Shutter..." );
      else
            AfxMessageBox( "Shutter Connected." );

      ErrorNumber = piSetShutterState(PI_SHUTTER_OPEN,pUsb1)

      if (ErrorNumber == DEVICE_NOT_FOUND)
      {
            AfxMessageBox( "Shutter was disconnected." );
            pUsb1 = NULL;  // Pointer is invalid after disconnecting
      }
      else
            AfxMessageBox( "Shutter Open." );
```

Get Shutter State

```
int __stdcall piGetShutterState(int * CurrentShutterState, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectShutter function.

CurrentShutterState

The current state of the Shutter as defined in piUSB.h Valid values are:

```
PI_SHUTTER_OPEN
PI_SHUTTER_CLOSED
```

Returns

Returns the error code. Valid return codes are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

Example

```
#include "pi_USB.h"

void * pUsb1;
int ErrorNumber;
int ShutterSerialNumber = 10;  // Serial number from Shutter
int CurrentShutterState;

      pUsb1 = piConnectShutter(&ErrorNumber,ShutterSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Shutter..." );
      else
           AfxMessageBox( "Shutter Connected." );

      ErrorNumber = piGetShutterState(&CurrentShutterState,pUsb1)

      if (ErrorNumber == DEVICE_NOT_FOUND)
      {
            AfxMessageBox( "Shutter was disconnected." );
            pUsb1 = NULL;  // Pointer is invalid after disconnecting
      }
      else if (CurrentShutterState == PI_SHUTTER_OPEN
            AfxMessageBox( "Shutter Open." );
      else
            AfxMessageBox( "Closed." );
```