Picard Industries PiUsb.dll for the Version 1.3 USB Motor

Connect to a Motor

```
void * __stdcall piConnectMotor(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 Motor label.

Returns

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

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
int FilterSerialNumber = 10;  // Serial number from Filter

      pUsb1 = piConnectMotor(&ErrorNumber,FilterSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );
```

Disconnect a Motor

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

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
int FilterSerialNumber = 10;  // Serial number from Filter

      pUsb1 = piConnectMotor(&ErrorNumber,FilterSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );

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

Home Motor

```
int __stdcall piHomeMotor(int Velocity, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

Velocity

The desired homing velocity. Valid values are 1 to 10.

Returns

Returns the error code. Valid return codes are:

```
     PI_NO_ERROR
     PI_DEVICE_NOT_FOUND
```

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
int Velocity = 3;  // 1 to 10 are valid velocities


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );

      ErrorNumber = piHomeMotor(Velocity,pUsb1);

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

Set Motor Velocity

```
int __stdcall piSetMotorVelocity(int Velocity, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

Velocity

The desired running velocity. Valid values are 1 to 10.

Returns

Returns the error code. Valid return codes are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
int Velocity = 3;  // 1 to 10 are valid velocities


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );

      ErrorNumber = piSetMotorVelocity (Velocity,pUsb1);

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

Halt Motor

```
int __stdcall piHaltMotor(void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

Returns

Returns the error code. Valid return codes are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );

      ErrorNumber = piHaltMotor(pUsb1);

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

Run Motor to a Position

```
int __stdcall piRunMotorToPosition( int Position, int Velocity, void *
devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

Position

The desired end postions. Valid values are 1 to 2000 for USB Motor and 1 to 8000 for USB Motor II.

Velocity

The desired running velocity. Valid values are 1 to 10.

Returns

Returns the error code. Valid return codes are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;

int Position = 1000;
int Velocity = 3;  // 1 to 10 are valid velocities


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
           AfxMessageBox( "Motor Connected." );

      ErrorNumber = piRunMotorToPosition(Position,Velocity,pUsb1);

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

Get Home Position Status

```
int __stdcall piGetMotorHomeStatus(BOOL * AtHome, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

AtHome

Returns AtHome set TRUE when at home position, FALSE otherwise

Returns

Returns the error code. Valid return codes are:

PI_NO_ERROR
PI_DEVICE_NOT_FOUND

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
BOOL AtHome;


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
           AfxMessageBox( "Motor Connected." );

      ErrorNumber = piGetMotorHomeStatus(&AtHome,Velocity,pUsb1);

      if (ErrorNumber == DEVICE_NOT_FOUND)
      {
            AfxMessageBox( "Motor was disconnected." );
            pUsb1 = NULL;  // Pointer is invalid after disconnecting
      }
      else
      {
            if (AtHome)
                  AfxMessageBox( "Motor is Homed." );
            else
                  AfxMessageBox( "Motor is not Homed." );
      }
```

Get Motor Moving Status

```
int __stdcall piGetMotorMovingStatus(BOOL * Moving, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

AtHome

Returns Moving set TRUE when stilll moving, FALSE otherwise

Returns

Returns the error code. Valid return codes are:

PI_NO_ERROR
PI_DEVICE_NOT_FOUND

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
BOOL Moving;


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );

      ErrorNumber = piGetMotorMovingStatus(&Moving,Velocity,pUsb1);

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

Get Motor Velocity

```
int __stdcall piGetMotorVelocity(int * ReportedVelocity, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

ReportedVelocity

Returns with value set to current velocity

Returns

Returns the error code. Valid return codes are:

```
PI_NO_ERROR
PI_DEVICE_NOT_FOUND
```

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
int ReportedVelocity;
CString ValueStr;


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
            AfxMessageBox( "Motor Connected." );

      ErrorNumber = piGetMotorVelocity(&ReportedVelocity,pUsb1);

      if (ErrorNumber == DEVICE_NOT_FOUND)
      {
            AfxMessageBox( "Motor was disconnected." );
            pUsb1 = NULL;  // Pointer is invalid after disconnecting
      }
      else
      {
            ValueStr.Format("%d",ReportedVelocity);
            SetDlgItemText(IDC_STATIC_VELOCITY,ValueStr);
      }
```

Get Motor Position

```
int __stdcall piGetMotorPosition(int * ReportedPosition, void * devicePtr);
```

Parameters

devicePtr

The device pointer that was returned from the piConnectMotor function.

ReportedPosition

Returns with value set to current position.

Returns

Returns the error code. Valid return codes are:

PI_NO_ERROR
PI_DEVICE_NOT_FOUND

Example

```
#include “PiUsb.h”

void * pUsb1;
int ErrorNumber;
int ReportedPosition;
CString ValueStr;


      pUsb1 = piConnectMotor(&ErrorNumber,MotorSerialNum);
      if (ErrorNumber == PI_DEVICE_NOT_FOUND)
            AfxMessageBox( "Unable to find Motor..." );
      else
           AfxMessageBox( "Motor Connected." );

      ErrorNumber = piGetMotorPosition (&ReportedPosition,pUsb1);

      if (ErrorNumber == DEVICE_NOT_FOUND)
      {
            AfxMessageBox( "Motor was disconnected." );
            pUsb1 = NULL;  // Pointer is invalid after disconnecting
      }
      else
      {
            ValueStr.Format("%d",ReportedPosition);
            SetDlgItemText(IDC_STATIC_POSITION,ValueStr);
      }
```