

Dewesoft C++ DLL to Trenz Electronic C# DLL

Introduction

How to write C# programs using the new DLL starting from the old DLL.

There are some major differences between the two DLLs.

feature	Dewesoft C++ DLL	Trenz Electronic C# DLL
programming language	C++	C#
architecture	standard (TE0300DLL.dll)	stacked (TE_USB_FX2_CyUSB.dll requires Cypress CyUSB.dll);
Handles	present	absent
structures	embedded	defined in Cypress CyAPI.h
parameters*	less	more
freedom*	less	more
buffer size	2 Kbyte (fixed)	4 Kbyte or more (it can be changed)

Feature of Dewesoft C++ DLL and Trenz Electronic C# DLL

Function translation

Dewesoft C++ DLL	Trenz Electronic C# DLL
HANDLE m_handle = 0;	CyUSBDevice TE_USB_FX2_USBDevice = null; USBDeviceList USBdevList = new USBDeviceList(CyConst.DEVICES_CYUSB);  The handles are internally managed by CyUSB.dll and there is no need to expose them to the user.  CyUSBDevice TE_USB_FX2_USBDevice take the place of handles for C# programmers.
cout << endl << TE0300_ScanCards() << endl;	int NumberOfCardAttached = TE_USB_FX2.TE_USB_FX2.ScanCards(ref USBdevList); Console.WriteLine(" {0} ", NumberOfCardAttached);

TE0300_Open (&m_handle, 0)!=0	TE_USB_FX2. TE_USB_FX2 .TE_USB_FX2_Open (ref TE_USB_FX2_USBDevice, ref USBdevList, 0) == false  TE_USB_FX2_Open() In the code, it is possible to call TE_USB_FX2_Open() where TE0300_Open() is used.  TE_USB_FX2_Open() as SelectCard() TE_USB_FX2_Open(TE_USB_FX2_USBDevice, USBdevList, x) acts more as a SelectCard() function because the list of USB devices is already created in USBdevList. TE_USB_FX2_USBDevice is the selected device number x (0 in this case).
TE0300_Open (&m_handle, 1)!=0	TE_USB_FX2. TE_USB_FX2 .TE_USB_FX2_Open (ref TE_USB_FX2_USBDevice, ref USBdevList, 1) == false
TE0300_Close (&m_handle);	TE_USB_FX2. TE_USB_FX2 .TE_USB_FX2_Close (ref USBdevListToDispose); In the code, it is not recommended to call TE_USB_FX2_Close() where TE0300_Close() is used. This function differs from its homonym of the previous TE0300DLL.dll in that it does not close a Handle but disposes (erases) all USB devices in the list USBdevList. The reason of this behavior is due to the CyUSB.dll as explained by Cypress document CyUSB.NET.pdf, pages 132-133 and pages 139-140: "You should never invoke the Dispose() method of a USBDevice directly. Rather, the appropriate technique is to call the Dispose() method of the USBDeviceList object that contains the USBDevice objects".
TE0300_SendCommand (handle, cmd, cmd_length, reply, &reply_length, timeout)	TE_USB_FX2. TE_USB_FX2 .TE_USB_FX2_SendCommand (ref TE_USB_FX2_USBDevice, ref cmd, ref cmd_length, ref reply, ref reply_length, TIMEOUT_MS)

Translation example between the DLLs

Dewesoft C++ DLL	Trenz Electronic C# DLL
<code>void ResetFX2FifoStatus(HANDLE handle) {</code>	<code>static void ResetFX2FifoStatus (CyUSBDevice TE_USB_FX2_USBDevice) {</code>
	<code>if (TE_USB_FX2_USBDevice == null) { Console.WriteLine("Error,no device is selected"); return; }</code>
<code>cout << endl << "Resetting all FIFOs" << endl;</code>	<code>Console.WriteLine("Resetting all FIFOs");</code>
<code>byte cmd[64], reply[64]; int cmd_length = 64; int reply_length = 64;</code>	<code>byte[] cmd = new byte[64]; byte[] reply = new byte[64]; int cmd_length = 64; int reply_length = 64;</code>
	<code>uint TIMEOUT_MS = 100;</code>
<code>cmd[0] = 0xA4; cmd[1] = 0; //RESET all FIFOs</code>	<code>cmd[0] = (byte)FX2_Commands.RESET_FIFO_STATUS; cmd[1] = 0; //RESET all FIFOs</code>

<pre> if (TE0300_SendCommand (handle, cmd, cmd_length, reply, &reply_length, 1000)) cout << "Error" << endl; </pre>	<pre> if (TE_USB_FX2.TE_USB_FX2.TE_USB_FX2_SendCommand (ref TE_USB_FX2_USBDevice, ref cmd, ref cmd_length, ref reply, ref reply_length, TIMEOUT_MS) == false) Console.WriteLine("Error Send Command Reset all fifos"); </pre>
<pre> cmd[0] = 0xA0;//command INITIALIZE cmd[1] = 1;//FIFO mode </pre>	<pre> cmd[0] = (byte)FX2_Commands.INITIALIZE; cmd[1] = 1;//FIFO mode </pre>
<pre> if (TE0300_SendCommand (handle, cmd, cmd_length, reply, &reply_length, 1000)) cout << "Error" << endl; } </pre>	<pre> if (TE_USB_FX2.TE_USB_FX2.TE_USB_FX2_SendCommand (ref TE_USB_FX2_USBDevice, ref cmd, ref cmd_length, ref reply, ref reply_length, TIMEOUT_MS) == false) Console.WriteLine("Error Switch Mode Fifo Mode"); } </pre>

Reset FX2 FIFO Status example

Dewesoft C++ DLL	Trenz Electronic C# DLL
<pre> void ReadData(unsigned int handle) { </pre>	<pre> static void ReadData (CyUSBDevice TE_USB_FX2_USBDevice, int BUFFER_SIZE, uint TIMEOUT_MS) { </pre>
	<pre> if (TE_USB_FX2_USBDevice == null) { Console.WriteLine("Error,no device is selected"); return; } </pre>
<pre> int packetlen = RX_PACKET_LEN; unsigned int packets = 1200; byte * data; data = new byte [RX_PACKET_LEN*packets]; unsigned int total_cnt = 0; unsigned int errors = 0; </pre>	<pre> int packetlen = RX_PACKET_LEN; int packets = 1200; byte[] data = new byte[packetlen*packets]; byte[] buffer = new byte[packetlen]; int total_cnt = 0; int errors = 0; </pre>
	<pre> int PI_EP6 = 6; bool bResultXfer = false; test_cnt = 0; total_cnt = 0; </pre>
<pre> ResetFX2FifoStatus(handle); //starts test SendFPGAcommand(handle, FX22MB_REG0_START_TX); </pre>	<pre> ResetFX2FifoStatus(TE_USB_FX2_USBDevice); //starts test SendFPGAcommand(ref TE_USB_FX2_USBDevice, MB_Commands.FX22MB_REG0_START_TX, TIMEOUT_MS); </pre>
<pre> //StopWatch start ElapsedTime.Start(); </pre>	<pre> //StopWatch start Stopwatch stopWatch = new Stopwatch(); stopWatch.Start(); </pre>
<pre> for (unsigned int i = 0; i < packets; i++) { packetlen = RX_PACKET_LEN; </pre>	<pre> for (int i = 0; i < packets; i++) { packetlen = RX_PACKET_LEN; </pre>
<pre> if (TE0300_GetData(handle, data+total_cnt, &packetlen, PI_EP6,TIMEOUT_MS)) { </pre>	<pre> bResultXfer = TE_USB_FX2.TE_USB_FX2.TE_USB_FX2_GetData (ref TE_USB_FX2_USBDevice, ref buffer, ref packetlen, PI_EP6, TIMEOUT_MS,BUFFER_SIZE); Buffer.BlockCopy(buffer,0, data, total_cnt, packetlen); if (bResultXfer == false) { </pre>

<pre> cout << "Error Get Data" << endl; errors++; break; } total_cnt += packetlen; } //StopWatch timer TheElapsedTime = ElapsedTime.Stop(false); //stops test SendFPGAcommand(handle,FX22MB_REG0_STOP); delete data; } </pre>	<pre> Console.WriteLine("Error Get Data"); errors++; break; } total_cnt += packetlen; } //StopWatch timer stopWatch.Stop(); TimeSpan ts = stopWatch.Elapsed; //stops test SendFPGAcommand(ref TE_USB_FX2_USBDevice, MB_Commands.FX22MB_REG0_STOP, TIMEOUT_MS); //Garbage Collector } </pre>
---	---

Read data example

Dewesoft C++ DLL	Trenz Electronic C# DLL
<pre> void WriteData(unsigned int handle) { </pre>	<pre> static void WriteData (CyUSBDevice TE_USB_FX2_USBDevice, int BUFFER_SIZE, uint TIMEOUT_MS) { </pre>
	<pre> if (TE_USB_FX2_USBDevice == null) { Console.WriteLine("Error,no device is selected"); return; } </pre>
<pre> int packetlen = TX_PACKET_LEN; unsigned int packets = 1200; byte * data; data = new byte [TX_PACKET_LEN*packets]; unsigned int total_cnt = 0; unsigned int errors = 0; </pre>	<pre> int packetlen = TX_PACKET_LEN; int packets = 1200; byte[] data = new byte[packetlen*packets]; byte[] buffer = new byte[packetlen]; int total_cnt = 0; int errors = 0; </pre>
	<pre> int PI_EP8 = 8; bool bResultXfer = false; test_cnt = 0; total_cnt = 0; </pre>
<pre> ResetFX2FifoStatus(handle); //starts test SendFPGAcommand(handle, FX22MB_REG0_START_RX); </pre>	<pre> ResetFX2FifoStatus(TE_USB_FX2_USBDevice); //starts test SendFPGAcommand(ref TE_USB_FX2_USBDevice, MB_Commands.FX22MB_REG0_START_RX, TIMEOUT_MS); </pre>
<pre> //StopWatch start ElapsedTime.Start(); </pre>	<pre> //StopWatch start Stopwatch stopWatch = new Stopwatch(); stopWatch.Start(); </pre>
<pre> for (unsigned int i = 0; i < packets; i++) { packetlen = TX_PACKET_LEN; </pre>	<pre> for (int i = 0; i < packets; i++) { packetlen = TX_PACKET_LEN; </pre>
<pre> if (TE0300_SetData(handle, data+total_cnt, &packetlen, PI_EP6,TIMEOUT_MS)) { </pre>	<pre> Buffer.BlockCopy(data,total_cnt, buffer, 0, packetlen); bResultXfer = TE_USB_FX2.TE_USB_FX2_SetData (ref TE_USB_FX2_USBDevice, ref buffer, ref packetlen, PI_EP8, TIMEOUT_MS,BUFFER_SIZE); if (bResultXfer == false) { </pre>

```

        cout << "Error Set Data" << endl;
        errors++;
        break;
    }
    total_cnt += packetlen;
}
//StopWatch timer
TheElapsedTime = ElapsedTime.Stop(false);

//stops test
SendFPGAcommand(handle,FX22MB_REG0_STOP);

delete data;
}

```

```

        Console.WriteLine("Error Set Data");
        errors++;
        break;
    }
    total_cnt += packetlen;
}
//StopWatch timer
stopWatch.Stop();
TimeSpan ts = stopWatch.Elapsed;
//stops test
SendFPGAcommand(ref TE_USB_FX2_USBDevice,
MB_Commands.FX22MB_REG0_STOP, TIMEOUT_MS);
//Garbage Collector
}

```

Write data example