Dewesoft C++ DLL
to
Trenz Electronic C++ DLL
Porting Guide

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

1 Introduction

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_CyAPI.dll requires Cypress CyAPI.dll);
Handles	present	absent
structures	embedded	defined in Cypress CyAPI.h
parameters*	less	more
freedom*	less	more

* Example: in TE0300DLL.dll, the buffer size is fixed to 2 kbyte, while in TE_USB_FX2_CyAPI.dll you are free to choose 4 kbyte or more.

2 Function translation

Dewesoft C++ DLL	Trenz Electronic C++ DLL
`HANDLE m_handle = 0;`	`CCyUSBDevice *USBDeviceList = new CCyUSBDevice((HANDLE)0,CYUSBDRV_GUID,true);`
`cout << endl << TE0300_ScanCards() << endl;`	`cout << endl << TE_USB_FX2_ScanCards(USBDeviceList) << endl;`
`TE0300_Open(&m_handle, 0)!=0`	`TE_USB_FX2_Open(USBDeviceList, 0)!=0`
`TE0300_Open(&m_handle, 1)!=0`	`TE_USB_FX2_Open(USBDeviceList, 1)!=0`
`TE0300_Close(&m_handle);`	`TE_USB_FX2_Close(USBDeviceList);`
`TE0300_SendCommand(handle, cmd, cmd_length, reply, &reply_length, timeout)`	`TE_USB_FX2_SendCommand(USBDeviceList, cmd, cmd_length, reply, reply_length, timeout)`

Dewesoft C++ DLL	Trenz Electronic C++ DLL

//test code, not production code
 	
int packetlen = 512;
byte data[512];
 	
for (int i = 0; i < 10; i++)
{
  packetlen = 512;
  for (int j = 0; j < packetlen; j++)
    data[j] = j;
  if (TE0300_SetData(handle, data, packetlen, PI_EP8))
  {
    cout << "ERROR" << endl;
    return;
  }
}
	//test code, not production code
	
	int packetlen = 512;
	byte data[512];
	
	CCyBulkEndPoint *BulkOutEP = NULL;
	
	TE_USB_FX2_SetData_InstanceDriverBuffer (USBDeviceList, &BulkOutEP, PI_EP8, timeout, DeviceDriverBufferSize);
	
	for (int i = 0; i < 10; i++)
	{
	  packetlen = 512;
	  for (int j = 0; j < packetlen; j++)
	    data[j] = j;
	  if (TE_USB_FX2_SetData(&BulkOutEP, data, packetlen))
	  {
	    cout << "ERROR" << endl;
	    return;
	  }
	}

<table>
<tr><th>Dewesoft C++ DLL</th><th>Trenz Electronic C++ DLL</th></tr>
<tr><td><pre><code>int packetlen = 512;
byte data[512];

for (int i = 0; i < 10; i++)
{
  packetlen = 512;
  if (TE0300_GetData(handle, data, &packetlen, PI_EP6,
1000))
  {
    cout << "ERROR" << endl;
    return;
  }
  for (int j = 0; j < packetlen; j++)
    cout << data[j];
  cout << endl;
}</code></pre></td><td><pre><code>int packetlen = 512;byte data[512];

CCyBulkEndPoint *BulkInEP = NULL;

TE_USB_FX2_GetData_InstanceDriverBuffer (USBDeviceList,
&BulkInEP, PI_EP6, timeout, DeviceDriverBufferSize);

for (int i = 0; i < 10; i++)
{
  packetlen = 512;
  if (TE_USB_FX2_GetData(&BulkInEP, data, packetlen))
  {
    cout << "ERROR" << endl;
    return;
  }
  for (int j = 0; j < packetlen; j++)
    cout << data[j];
  cout << endl;
}</code></pre></td></tr>
</table>

Dewesoft C++ DLL	Trenz Electronic C++ DLL
void ReadData(unsigned int handle)	void ReadData(CCyUSBDevice *USBDeviceList, unsigned int
{	DeviceDriverBufferSize, int RX_PACKET_LEN, unsigned long
	TIMEOUT)
	{
int packetlen = RX_PACKET_LEN;	long packetlen = RX_PACKET_LEN;
unsigned int packets = 1200;	unsigned int packets = 1200;
byte * data;	byte * data;
	byte * data_temp = NULL;
unsigned int total_cnt = 0;	unsigned int total_cnt = 0;
unsigned int errors = 0;	unsigned int errors = 0;
data = new byte [RX_PACKET_LEN*packets];	data = new byte [RX_PACKET_LEN*packets];
//allocate memory	//allocate memory
ResetFX2FifoStatus(handle);	ResetFX2FifoStatus(USBDeviceList);
SendFPGAcommand(handle,FX22MB_REG0_START_TX);	SendFPGAcommand(USBDeviceList,FX22MB_REG0_START_TX);
//starts test	//starts test
	CCyBulkEndPoint *BulkInEP = NULL;
	TE_USB_FX2_GetData_InstanceDriverBuffer (USBDeviceList,
	CardNo, &BulkInEP, PI_EP6, TIMEOUT,
	DeviceDriverBufferSize);
ElapsedTime.Start(); //StopWatch start	ElapsedTime.Start(); //StopWatch start
for (unsigned int i = 0; i < packets; i++)	for (unsigned int i = 0; i < packets; i++)
{	{
packetlen = RX_PACKET_LEN;	packetlen = RX_PACKET_LEN;
if (TE0300_GetData(handle, data+total_cnt, &packetlen,	data_temp = &data[total_cnt];
PI_EP6,TIMEOUT_MS))	if (TE_USB_FX2_GetData(&BulkInEP,data_temp,packetlen))
{	{
cout << "ERROR" << endl;	cout << "ERROR read" << endl;
errors++;	errors++;
break;	break;
}	}
total_cnt += packetlen;	total_cnt += (packetlen);
}	}
TheElapsedTime = ElapsedTime.Stop(false);	TheElapsedTime = ElapsedTime.Stop(false);
//DEBUG StopWatch timer	//DEBUG StopWatch timer
SendFPGAcommand(handle,FX22MB_REG0_STOP);	SendFPGAcommand(USBDevice,FX22MB_REG0_STOP);
//stops test	//stops test
delete data;	delete data;
}	}

Dewesoft C++ DLL

```
void WriteData(unsigned int handle)
{


  int packetlen = TX_PACKET_LEN;
  unsigned int packets = 1200;
  byte * data;

  unsigned int total_cnt = 0;
  unsigned int errors = 0;
  data = new byte [TX_PACKET_LEN*packets];
  //allocate memory
  SetData (data);
  ResetFX2FifoStatus(handle);
  SendFPGAcommand(handle,FX22MB_REG0_START_RX);
  //starts test



  ElapsedTime.Start();
  //StopWatch start
  for (unsigned int i = 0; i < packets; i++)
  {
    packetlen = TX_PACKET_LEN;
    if (TE0300_GetData(handle, data+total_cnt, &packetlen,
PI_EP8,TIMEOUT_MS))
    {
      cout << "ERROR" << endl;
      errors++;
      break;
    }
    total_cnt += packetlen;
  }
  TheElapsedTime = ElapsedTime.Stop(false);
  //DEBUG StopWatch timer
  SendFPGAcommand(handle,FX22MB_REG0_STOP);
  //stops test
  delete data;
}
```

Trenz Electronic C++ DLL

```
void WriteData(CCyUSBDevice *USBDeviceList, unsigned int
DeviceDriverBufferSize, int TX_PACKET_LEN, unsigned long
TIMEOUT)
{
  long packetlen = TX_PACKET_LEN;
  unsigned int packets = 1200;
  byte * data;
  byte * data_temp = NULL;
  unsigned int total_cnt = 0;
  unsigned int errors = 0;
  data = new byte [TX_PACKET_LEN*packets];
  //allocate memory
  SetData (data);
  ResetFX2FifoStatus(USBDeviceList);
  SendFPGAcommand(USBDeviceList,FX22MB_REG0_START_RX);
  //starts test
  CCyBulkEndPoint *BulkOutEP = NULL;
  TE_USB_FX2_SetData_InstanceDriverBuffer ( USBDeviceList,
CardNo, &BulkOutEP, PI_EP8,TIMEOUT,DeviceDriverBufferSize);
  ElapsedTime.Start();
  //StopWatch start
  for (unsigned int i = 0; i < packets; i++)
  {
    packetlen = TX_PACKET_LEN;
    data_temp = &data[total_cnt];
     if (TE_USB_FX2_GetData(&BulkInEP,data_temp,packetlen))
     {
       cout << "ERROR read" << endl;
       errors++;
       break;
     }
     total_cnt += (packetlen);
  }
  TheElapsedTime = ElapsedTime.Stop(false);
  //DEBUG StopWatch timer
  SendFPGAcommand(USBDevice,FX22MB_REG0_STOP);
 //stops test
  delete data;
}
```

3 Document Change History

version	date	author	description
0.9	2012-06-01	SP, FDR	Public preview.
1.0			Initial release.