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

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

1 Introduction

There are some major differences between the two DLLs.

feature	Dewesoft C++ DLL	Simplified Trenz Electronic C++ DLL
programming language	C, C++, Python	C, C++, Python
architecture	standard (TE0300DLL.dll)	Standard (the stacked nature of the solution is hidden) (TE_USB_FX2_CyAPI.dll requires Cypress CyAPI.lib);
Handles	present	absent
structures	embedded	Embedded (defined in Cypress CyAPI.h but invisible to user)
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 Declarations

```
#define TE_USB_FX2_CYAPI extern "C"  __declspec(dllexport)

//typedef int (WINAPI *_TE_USB_FX2_ScanCards)();
TE_USB_FX2_CYAPI int TE_USB_FX2_ScanCards ();

//typedef int (WINAPI *_TE_USB_FX2_Open)(unsigned int* PHandle, int CardNo);
TE_USB_FX2_CYAPI int TE_USB_FX2_Open (int CardNumber, unsigned long TimeOut, int DriverBufferSize);

//typedef int (WINAPI *_TE_USB_FX2_Close)(unsigned int* PHandle);
TE_USB_FX2_CYAPI int TE_USB_FX2_Close ();

//typedef int (WINAPI *_TE_USB_FX2_SendCommand)(unsigned int PHandle, byte* cmd, int cmd_len, byte* reply, int* reply_len, int timeout);
TE_USB_FX2_CYAPI int TE_USB_FX2_SendCommand ( byte* Command, long CmdLength, byte* Reply, long ReplyLength, unsigned long Timeout);

//typedef int (WINAPI *_TE_USB_FX2_GetData)(unsigned int PHandle, byte* data, int* len, PI_PipeNumber, int timeout);
TE_USB_FX2_CYAPI int TE_USB_FX2_GetData ( byte* DataRead, long DataReadLength);

//typedef int (WINAPI *_TE_USB_FX2_SetData)(unsigned int PHandle, byte* data, int len, PI_PipeNumber);
TE_USB_FX2_CYAPI int TE_USB_FX2_SetData ( byte* DataWrite, long DataWriteLength);
```

The two functions that follow appear in the header but are used only internally by the DLL (TE_USB_FX2_CyAPI.dll) and are not exported to the user:

```
int TE_USB_FX2_GetData_InstanceDriverBuffer (CCyUSBDevice *USBDeviceList, CCyBulkEndPoint **BulkInEPx, PI_PipeNumber PipeNo, unsigned long
Timeout, int BufferSize);

int TE_USB_FX2_SetData_InstanceDriverBuffer (CCyUSBDevice *USBDeviceList, CCyBulkEndPoint **BulkOutEPx, PI_PipeNumber PipeNo, unsigned long
Timeout, int BufferSize);
```

These two functions are called internally by function TE_USB_FX2_Open().

Internal note: With this declaration, TE_USB_FX2_CyAPI.dll has been successfully verified in a Python program (Open_FWUT) using ctypes (used to import/export c types): all functions have been verified (in the Python program Open_FWUT) apart from TE_USB_FX2_GetData() and TE_USB_FX2_SetData().

3 Function Translation

Dewesoft C++ DLL	Simplified Trenz Electronic C++ DLL
`HANDLE m_handle = 0;`	`Nothing (you must charge the DLL)`
`cout << endl << TE0300_ScanCards() << endl;`	`cout << endl << TE_USB_FX2_ScanCards() << endl;`
`TE0300_Open(&m_handle, 0)!=0`	`TE_USB_FX2_Open(0, TimeOut, DriverBufferSize)!=0`
`TE0300_Open(&m_handle, 1)!=0`	`TE_USB_FX2_Open(1, TimeOut, DriverBufferSize)!=0`
`TE0300_Close(&m_handle);`	`TE_USB_FX2_Close();`
`TE0300_SendCommand(handle, cmd, cmd_length, reply, &reply_length, timeout)`	`TE_USB_FX2_SendCommand( cmd, cmd_length, reply, reply_length, timeout)`

Internal notes:

TimeOut, *DriverBufferSize* : it is possible that these parameters can be moved to another function like TE_USB_FX2_SetTimeOut and TE_USB_FX2_SetDriverBufferSize or erased (internally fixed) if Trenz Electronic decides in this direction.

The instantiation of driver buffer happens in TE_USB_FX2_Open(): the user must specify *TimeOut* and *DriverBufferSize*.
A future extension is the possibility to set *TimeOut* = 1000 (1 ms) and *DriverBufferSize* = 131,072 if the respective value passed to the function is 0.

Dewesoft C++ DLL	Simplified 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];`
	
	`for (int i = 0; i < 10; i++)`
	`{`
	`  packetlen = 512;`
	`  for (int j = 0; j < packetlen; j++)`
	`  data[j] = j;`
	`  if (TE_USB_FX2_SetData(data, packetlen))`
	`  {`
	`  cout << "ERROR" << endl;`
	`  return;`
	`  }`
	`}`

Dewesoft C++ DLL	Simplified Trenz Electronic C++ DLL

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;
}
	int packetlen = 512;
	byte data[512];
	
	for (int i = 0; i < 10; i++)
	{
	  packetlen = 512;
	  if (TE_USB_FX2_GetData(data, packetlen))
	
	  {
	  cout << "ERROR" << endl;
	  return;
	  }
	  for (int j = 0; j < packetlen; j++)
	  cout << data[j];
	  cout << endl;
	}

Dewesoft C++ DLL	Simplified Trenz Electronic C++ DLL

```
void ReadData(unsigned int handle)
{
  int packetlen = RX_PACKET_LEN;
  unsigned int packets = 1200;
  byte * data;

  unsigned int total_cnt = 0;
  unsigned int errors = 0;
  data = new byte [RX_PACKET_LEN*packets];
  //allocate memory

  ResetFX2FifoStatus(handle);
  SendFPGAcommand(handle,FX22MB_REG0_START_TX);
  //starts test

  ElapsedTime.Start(); //StopWatch start
  for (unsigned int i = 0; i < packets; i++)
  {
    packetlen = RX_PACKET_LEN;

    if (TE0300_GetData(handle, data+total_cnt, &packetlen,
PI_EP6,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;
}
```

```
void ReadData()
{
  long packetlen = RX_PACKET_LEN;
  unsigned int packets = 1200;
  byte * data;
  byte * data_temp = NULL;
  unsigned int total_cnt = 0;
  unsigned int errors = 0;
  data = new byte [RX_PACKET_LEN*packets];
  //allocate memory

  ResetFX2FifoStatus();
  SendFPGAcommand(FX22MB_REG0_START_TX);
  //starts test

  ElapsedTime.Start(); //StopWatch start
  for (unsigned int i = 0; i < packets; i++)
  {
    packetlen = RX_PACKET_LEN;
    data_temp = &data[total_cnt];
     if (TE_USB_FX2_GetData(data_temp,packetlen))
     {

       cout << "ERROR read" << endl;
       errors++;
       break;
     }
     total_cnt += (packetlen);
  }
  TheElapsedTime = ElapsedTime.Stop(false);
  //DEBUG StopWatch timer
  SendFPGAcommand(FX22MB_REG0_STOP);
 //stops test
  delete data;
}
```

Dewesoft C++ DLL	Simplified Trenz Electronic 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;
}
	void WriteData()
	{
	  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();
	  SendFPGAcommand(FX22MB_REG0_START_RX);
	  //starts test
	  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(data_temp,packetlen))
	  {
	  cout << "ERROR read" << endl;
	  errors++;
	  break;
	  }
	  total_cnt += (packetlen);
	  }
	  TheElapsedTime = ElapsedTime.Stop(false);
	  //DEBUG StopWatch timer
	  SendFPGAcommand(FX22MB_REG0_STOP);
	 //stops test
	  delete data;
	}

4 A Note on the Header File TE_USB_FX2_CyAPI.h

Two different header files exist.

One is used for the creation of TE_USB_FX2_CyAPI.dll DLL.

Another header file (with the same name) is exported for the creation of application programs that use TE_USB_FX2_CyAPI.dll DLL.

The latter has some additional lines with regard to the former. This happens to solve some problems with the include files in applications.

The lines of difference are the followings (added to TE_USB_FX2_CyAPI.h used for applications).

```
#pragma once
//#include <WinDef.h> NO, it fails at 32 bit
#include <windows.h>
#include "CyAPI.h"
typedef unsigned char byte;
```

Internal note: The file to use is in C:\Project3264bit\TE_USB_FX2_CyAPI\FileToExportForApplication

5 Procedure for the Creation of a New Project Using Simplified TE_USB_FX2_CyAPI.dll

See document “C++ TE_USB_FX2 API reference manual”

http://www.trenz-electronic.de/download/d0/Trenz_Electronic/d1/TE-USB-Suite/d2/generation_3/d3/APIs.html
section 3.0 “API Functions” (“Exported function list” excluded).

6 Appendix A : Open the Visual Studio 2010 project

See document “C++ TE_USB_FX2 API reference manual”

http://www.trenz-electronic.de/download/d0/Trenz_Electronic/d1/TE-USB-Suite/d2/generation_3/d3/APIs.html
section 7 “Appendix A : Open the Visual Studio 2010 project”.

7 Appendix B : Use of pdb File (Symbolic Debugging)

See document “C++ TE_USB_FX2 API reference manual”
http://www.trenz-electronic.de/download/d0/Trenz_Electronic/d1/TE-USB-Suite/d2/generation_3/d3/APIs.html
section 8 “Appendix B : use of pdb file (symbolic debugging)”.

8 Document Change History

version	date	author	description
0.1	2012-09-24	SP, FDR	Release Preview.
0.2	2013-04-09	FDR	Updated documentation links.