

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

Introduction

How to write C++ programs using the new [Simplified DLL](#) starting from the old [DEWESoft DLL](#).

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
buffer size	2 Kbyte (fixed)	4 Kbyte or more (it can be changed)

Features comparison



Both the Simplified Trenz Electronic C++ DLL and this porting guide are fully working developer versions but are not supported by Trenz Electronic till the official release, currently not yet planned.

Function Declarations

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

Exported function (from DLL)

- TE_USB_FX2_CYAPI [int](#) TE_USB_FX2_ScanCards ();
- TE_USB_FX2_CYAPI [int](#) TE_USB_FX2_Open ([int](#) CardNumber, [unsigned long](#) TimeOut, [int](#) DriverBufferSize);
- TE_USB_FX2_CYAPI [int](#) TE_USB_FX2_Close ();
- TE_USB_FX2_CYAPI [int](#) TE_USB_FX2_SendCommand (byte* Command, [long](#) CmdLength, byte* Reply, [long](#) ReplyLength, [unsigned long](#) Timeout);
- TE_USB_FX2_CYAPI [int](#) TE_USB_FX2_GetData (byte* DataRead, [long](#) DataReadLength);
- TE_USB_FX2_CYAPI [int](#) TE_USB_FX2_SetData (byte* DataWrite, [long](#) DataWriteLength);

Internal Function (not exported from DLL)

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().



This Simplified DLL is not thread safe.



This Simplified DLL is successfully used in the Python (using ctypes to import/export c types) program `Open_FUT` (Gen3) to program USB FX2 microcontroller's firmware and SPI Flash/FPGA's bitstream.

This Simplified DLL is full extern C (C compatible).

Function Translation

Dewesoft C++ DLL	Simplified Trenz Electronic C++ DLL
<code>HANDLE m_handle = 0;</code>	Nothing (you must charge the DLL)
<code>cout << endl << TE0300_ScanCards() << endl;</code>	<code>cout << endl << TE_USB_FX2_ScanCards() << endl;</code>
<code>TE0300_Open(&m_handle, 0)!=0</code>	<code>TE_USB_FX2_Open(0, TimeOut, DriverBufferSize)!=0</code>
<code>TE0300_Open(&m_handle, 1)!=0</code>	<code>TE_USB_FX2_Open(1, TimeOut, DriverBufferSize)!=0</code>
<code>TE0300_Close(&m_handle);</code>	<code>TE_USB_FX2_Close();</code>
<code>TE0300_SendCommand(handle, cmd, cmd_length, reply, &reply_length, timeout)</code>	<code>TE_USB_FX2_SendCommand(cmd, cmd_length, reply, reply_length, timeout)</code>
<code>TE0300_SetData(handle, data, packetlen, PI_EP8)</code>	<code>TE_USB_FX2_SetData(data, packetlen)</code>
<code>TE0300_GetData(handle, data, &packetlen, PI_EP6, 1000)</code>	<code>TE_USB_FX2_GetData(data, packetlen)</code>

Function translation between DLLs



The instantiation of driver buffer happens in `TE_USB_FX2_Open()`: the user must specify *TimeOut* and *DriverBufferSize*.



TimeOut, *DriverBufferSize*: it is possible to move these parameters to another function like `TE_USB_FX2_SetTimeOut` and `TE_USB_FX2_SetDriverBufferSize` or erase them (fix the value inside the DLL).

A future possible extension is 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
------------------	-------------------------------------

```

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
    SetData (data);
    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;
    }
    //StopWatch timer
    TheElapsedTime = ElapsedTime.Stop(false);
    //stops test
    SendFPGAcommand(handle,FX22MB_REG0_STOP);
    delete data;
}

```

```

void ReadData(unsigned int handle)
{
    int 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
    SetData (data);
    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;
        data_temp = &data[total_cnt];

        if (TE_USB_FX2_GetData(data_temp,packetlen))
        {
            cout << "ERROR" << endl;
            errors++;
            break;
        }
        total_cnt += packetlen;
    }
    //StopWatch timer
    TheElapsedTime = ElapsedTime.Stop(false);
    //stops test
    SendFPGAcommand(handle,FX22MB_REG0_STOP);
    delete data;
}

```

Read Data example

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_SetData(handle, data+total_cnt,
&packetlen, PI_EP8,TIMEOUT_MS))
        {
            cout << "ERROR" << endl;
            errors++;
            break;
        }
        total_cnt += packetlen;
    }
    //StopWatch timer
    TheElapsedTime = ElapsedTime.Stop(false);
    //stops test
    SendFPGAcommand(handle,FX22MB_REG0_STOP);
    delete data;
}

```

```

void WriteData(unsigned int handle)
{
    int 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(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;
        data_temp = &data[total_cnt];

        if (TE_USB_FX2_SetData(data_temp,packetlen))
        {
            cout << "ERROR" << endl;
            errors++;
            break;
        }
        total_cnt += packetlen;
    }
    //StopWatch timer
    TheElapsedTime = ElapsedTime.Stop(false);
    //stops test
    SendFPGAcommand(handle,FX22MB_REG0_STOP);
    delete data;
}

```

Write Data example