

# TE\_USB\_FX2\_CyAPI.dll C++ examples

## First example: read test

See C++ FX22MB\_REG0\_START\_TX example.

## Second example: write test

See C++ FX22MB\_REG0\_START\_RX example.

## Third example: select module, read firmware version, read VID/PID

This example program implements a simple console that

1. creates an instance (USBDeviceList) initialized to null of the class CCyUSBDevice
2. reads and displays the number of Trenz Electronic modules (TE\_USB\_FX2\_ScanCards())
3. selects the first (0) Trenz Electronic module;an handle to the device driver is open but not exposed to the user
4. reads and displays VID and PID of the selected module
5. reads the firmware version of the selected module
6. selects the second (1) Trenz Electronic module;the previous handle is automatically closed and another handle (this time associated with the new Trenz Electronic module) to the device driver is opened but not exposed to the user. If the selected module is not attached to the USB bus, the previous handle is automatically closed but a new one is not opened.
7. reads the FPGA firmware version of the second (1) module

```
//This line creates a list of USB device that are used through Cypress driver
CCyUSBDevice *USBDeviceList = new CCyUSBDevice((HANDLE)0,CYUSBDRV_GUID,true);

unsigned long Timeout = 1000;

int NumberOfCardAttached = TE_USB_FX2_ScanCards(USBDeviceList);

cout << endl << NumberOfCardAttached << endl;

//If you want use the first Trenz Electronic module use 0

if (TE_USB_FX2_Open(USBDeviceList, 0)==0) cout << "Module is connected!" <<endl;
else cout << "Module is not connected!" <<endl;

//USBDevice->Open(CypressDeviceNumber); it is used in TE_USB_FX2_Open()

int vID = USBDeviceList->VendorID;

int pID = USBDeviceList->ProductID;

cout << "VID" << vID << endl;

cout << "PID" << pID << endl;

byte Command[64], Reply[64];

long CmdLength = 64;

long ReplyLength = 64;

Command[0] = 0x00;//comand read FX2 version
```

```

if (!TE_USB_FX2_SendCommand(USBDeviceList, Command, CmdLength, Reply, ReplyLength, Timeout))
{
    if (ReplyLength >= 4)
    {
        printf("Major version: %d \n", Reply[0]);

        printf("Minor version: %d \n", Reply[1]);

        printf("Device hi: %d \n", Reply[2]);

        printf("Device lo: %d \n", Reply[3]);

    }
}

else cout << "Error" << endl;

//If you want use the first Trenz Electronic module use 1

if (TE_USB_FX2_Open(USBDeviceList, 1)==0) cout << "Module is connected!" <<endl;
else cout << "Module is not connected!" <<endl;

byte Command1[64], Reply1[64];

long CmdLength1 = 64;

long ReplyLength1 = 64;

Command1[0] = SET_INTERRUPT; //0xB0;//comand SET_INTERRUPT

Command1[1] = MB_I2C_ADRESS; //0x3F;//I2C slave address

Command1[2] = I2C_BYTES;//12;//12 bytes payload

if (TE_USB_FX2_SendCommand(USBDeviceList, Command1, CmdLength1, Reply1, ReplyLength1, Timeout))
    cout << "Error Send Command SET INTERRUPT" << endl;

Command1[0] = I2C_WRITE; //0xAD;//comand I2C_WRITE

//Command1[1] = MB_I2C_ADRESS; //0x3F;//I2C slave address

//Command1[2] = I2C_BYTES; //12 bytes payload

Command1[3] = 0;

Command1[4] = 0;

Command1[5] = 0;

Command1[6] = FX22MB_REG0_GETVERSION; //1; //get FPGA version

if (TE_USB_FX2_SendCommand(USBDeviceList, Command1, CmdLength1, Reply1, ReplyLength1, Timeout))
    cout << "Error Send Command Get FPGA Version" << endl;

Command[0] = GET_INTERRUPT; //0xB1;//comand GET_INTERRUPT

if (!TE_USB_FX2_SendCommand(USBDeviceList, Command1, CmdLength1, Reply1, ReplyLength1, Timeout))
{

```

```
if ((ReplyLength1 > 4) &&(Reply1[0] != 0))
{
    printf("Major version: %d \n", Reply1[1]);
    printf("Minor version: %d \n", Reply1[2]);
    printf("Release version: %d \n", Reply1[3]);
    printf("Build version: %d \n", Reply1[4]);
}
}

else cout << "Error, GET INTERRUPT" << endl;
```