C++ TE_USB_FX2_SetData_InstanceDriverBuffer()

Description

Brief Description

This function instantiate the driver buffer (host computer buffer) for a single TE USB FX2 module write connection.

> ⚠ This function has not been included in TE_USB_FX2_SetData() for throughput reasons; if the driver buffer instantiation were repeated at every data reception, the data throughput would be halved.
>
> This function shall be used only one time to instantiate the driver buffer; after instantiation, TE_USB_FX2_SetData() can be used repeatedly without re-instantiating the driver buffer.

Longer Description

This function takes an already initialized USB device list (USBDevice previously selected by TE_USB_FX2_Open()) and a not initialized CCyBulkEndPoint double pointer, BulkOutEP. This function selects the endpoint to use: you shall choose EP8 (0x08) (endpoints EP4(0x04) or EP2(0x02) are also theoretically possible).

> ⚠ Currently, only endpoint 0x08 is actually implemented in Trenz Electronic USB FPGA modules, so that endpoints EP2 and EP4 cannot be written or , more precisely, they are not even connected to the FPGA. That is why attempting to write them causes a function failure after Timeout expires.

Description of internal procedure

TE_USB_FX2_SetData_InstanceDriverBuffer() function instantiates the class used by CyAPI to use Bulk EndPoint (CCyBulkEndPoint, see pages 9 to 11) and initializes the parameters of this class instantiation.

The parameters are :

1. Timeout
2. XMODE_DIRECT (this parameter set the driver to single buffering, instead the slower double buffering)
3. DeviceDriverBufferSize.

The last parameter force the instantiation of the driver buffer (SW side, on the host computer) for the endpoint 0x86; this buffer has a size in byte given by DeviceDriverBufferSize. This value is of great importance because the data throughput is strongly influenced by this parameter (see Data Transfer Throughput Optimization).

Use of the code

Declaration

```
TE_USB_FX2_CYAPI int TE_USB_FX2_SetData_InstanceDriverBuffer(CCyUSBDevice *USBDeviceList, CCyBulkEndPoint
**BulkOutEP, PI_PipeNumber PipeNo,unsigned long Timeout, int BufferSize);
```

Function Call

Your application program shall call this function like this:

```
TE_USB_FX2_SetData_InstanceDriverBuffer (USBDeviceList, &BulkOutEP, PipeNo, Timeout, BufferSize);
```

Parameters

```
CCyUSBDevice *USBDeviceList
```

CCyUSBDevice is a type defined in CyUSB.dll. Its name is misleading because it is not a class that represents a single USB device, but it rather represents a list of USB devices. CCyUSBDevice is the list of devices served by the CyUSB.sys driver (or a derivative like TE_USB_FX2_xx.sys). This parameter is passed by pointer. See page 7 and pages 23-49 of CyAPI.pdf (Cypress CyAPI Programmer's Reference).

```
CCyBulkEndPoint **BulkOutEP
```

This parameter is a double pointer to CCyBulkEndPoint. This parameter is used to pass the used BulkEndPoint parameter to TE_USB_FX2_SetData(). The double pointer is used because, if single pointer were used, the data modification of TE_USB_FX2_SetDataInstanceDriverBuffer() could not be passed over to TE_USB_FX2_SetData.()

```
PI_PipeNumber PipeNo
```

This parameter is the value that identifies the endpoint used for data transfer. It is called PipeNumber because it identifies the buffer (pipe) used by the USB FX2 microcontroller.

```
unsigned long Timeout
```

It is the integer time value in milliseconds assigned to the synchronous method XferData() of data transfer used by CyAPI.lib. TimeOut is the time that is allowed to to the function for sending/receiving the data packet passed to the function; this Timeout shall be large enough to allow data/command transmission/reception.Otherwise the transmission/reception will fail. See Timeout Setting.

```
int BufferSize
```

It is the dimension (in bytes) of the driver buffer (SW) used in data transmission of a ***single endpoints (EP8 0x08 in this case);*** the total buffer size is the sum of BufferSize of every endpoint used. BufferSize has a strong influence on DataThroughput. If BufferSize is too small, DataThroughput can be 1/3-1/2 of the maximum value (from a maximum value of 24 Mbyte/s for read transactions to an actual value of 18 Mbyte/s). See 6 TE_USB_FX2_CyAPI.dll: Data Transfer Throughput Optimization.

Return Value

int : integer type

This function returns true (ST_OK=0) if the selected BulkEndPoint exists in the firmware (it is able to instantiate the driver buffer). This function returns false (ST_ERROR=1) otherwise.

```
enum ST_Status
{
     ST_OK = 0,
     ST_ERROR = 1
};
```

Code example

```
int TX_PACKET_LEN = 51200;//102400;
int packetlen = TX_PACKET_LEN;
unsigned int packets = 500;//1200;//1200;
unsigned int DeviceDriverBufferSize = 102400;
unsigned long TIMEOUT= 200;
byte * data;
byte * data_temp = NULL;
unsigned int total_cnt = 0;
unsigned int errors = 0;
data = new byte [TX_PACKET_LEN*packets]; //allocate memory
PI_PipeNumber PipeNo = PI_EP8;
//test starts
SendFPGAcommand(USBDeviceList,FX22MB_REG0_START_RX);
CCyBulkEndPoint *BulkOutEP = NULL;
TE_USB_FX2_SetData_InstanceDriverBuffer (USBDeviceList, &BulkOutEP, PipeNo, TIMEOUT, DeviceDriverBufferSize);
ElapsedTime.Start(); //StopWatch start
for (unsigned int i = 0; i < packets; i++)
{
        packetlen = TX_PACKET_LEN;
        data_temp = &data[total_cnt];
        //cout << "Address &BulkInEP" << &BulkInEP << endl;
        //cout << "Address BulkInEP" << BulkInEP << endl;
        //cout << "Address *BulkInEP" << (*BulkInEP) << endl;
        if (TE_USB_FX2_SetData(&BulkOutEP, data_temp, packetlen))
        {
                cout << "ERROR read" << endl;
                errors++;
                break;
        }
        total_cnt += packetlen;
}
//StopWatch timer stops
TheElapsedTime = ElapsedTime.Stop(false);
//test stops
SendFPGAcommand(USBDeviceList,FX22MB_REG0_STOP);
```