C++ TE_USB_FX2_SetData()

Description

TE_USB_FX2_SetData() reads data from the host computer and writes them to the USB FX2 microcontroller endpoint EP8 (0x08). This data is then passed over to the FPGA.

This function takes an already initialized CCyBulkEndPoint double pointer. The device has been previously selected by TE_USB_FX2_Open().

EP8's buffer is an HW buffer inside USB FX2 microcontroller, not the SW driver's buffer (on host computer) whose size is given by BufferSize parameter.

It is necessary that a FW/HW routine reads USB FX2 microcontroller's EP8 buffer.

If the EP8's buffer is not is not properly read/emptied by the FPGA(reference design)/other(custom design) the EP8's buffer will become full and no longer able to receive further packets. In this case the TE_USB_FX2_SetData() could experience strange behaviors. For example, a very low throughput (9-10 Mbyte/s even if a 22-24 Mbyte/s are expected) could be measured or the function TE_USB_FX2_SetData() fails returning false.

During USB write transmission test (TX: host computer perspective) the EP8's buffer reading is carried out by MicroBlaze (inside the FPGA); this behavior is setted by MicroBlaze API command FX22MB_REG0_START_RX (RX: USB FX2 microcontroller perspective).

```
SendFPGAcommand(USBDeviceList,FX22MB_REG0_START_RX);
```

Expected Data Throughput

The maximum data throughput expected (with a DataWriteLength= 120*10^6) is 24 Mbyte/s (PacketSize = BufferSize =102400) but in fact this value is variable between 22-29 Mbyte/s (the mean value seems 24 Mbyte/s); so if you measure this range of values, the data reception can be considered as normal.

⚠ The data throughput is variable in two ways:

- depends on the used host computer;
- varies with every function call (computer loading dependent).

DataRead Size Shall Not Be Too Large

TE_USB_FX2_SetData() seems unable to use too large arrays or, more precisely, this fact seems variable by changing host computer. To be safe, do not try to transfer in a single packet very large data (e.g. 120 millions of byte); transfer the same data with many packets instead (1,200 packets * 100,000 byte) and copy the data in a single large data array if necessary.

DataRead Size Shall Not Be Too Small

There are two reasons why DataWrite size shall not be too small.

The first reason is that PacketSize has a strong influence on DataThroughput. If PacketSize is too small (e.g. 512 byte), you can have very low DataThroughput (2.2 Mbyte/s) even if you use a large driver buffer (driver buffer size = 131,072 bytes). See section 6 TE_USB_FX2_CyAPI.dll: Data Transfer Throughput Optimization.

The second reason is that probably the FPGA imposes your minimum packet size. In a properly used write test mode (using FX22MB_REG0_START_RX and therefore attaching the FPGA), TE_USB_FX2_SetData() is unable to write less than 1024 byte. In a improperly used read test mode (not using FX22MB_REG0_START_RX and therefore detaching the FPGA), TE_USB_FX2_SetData() is able to write a packet size down to 64 byte. The same CyAPI method XferData() used (under the hood) in TE_USB_FX2_SendCommand() is able to read a packet size of 64 byte. These facts prove that the minimum packet size is imposed by FPGA. To be safe, we recommend to use this function with a size multiple of 1 kbyte.

Use of the code

For a code example see TE_USB_FX2_SetData_InstanceDriverBuffer().

Declaration

```
TE_USB_FX2_CYAPI int TE_USB_FX2_SetData (CCyBulkEndPoint *BulkOutEP, byte DataWrite, long DataWriteLength);
```

Function Call

Your application program shall call this function like this:

```
TE_USB_FX2_SetData (&BulkOutEP, DataWrite ,DataWriteLength);
```

Parameters

```
CCyBulkEndPoint **BulkOutEP
```

This parameter is used to pass to TE_USB_FX2_SetData() the parameter of BulkEndPoint used. This parameter is a double pointer to CcyBulkEndPoint. The double pointer is used because if single pointer is used the data modification of TE_USB_FX2_SetDataInstanceDriverBuffer() cannot be passed to TE_USB_FX2_SetData().

```
byte* DataWrite
```

C++ applications use directly TE_USB_FX2_CyAPI.dll based on CyAPI.lib. This parameter is passed by pointer to avoid copying back and forth large amount of data between these two DLLs. This parameter points the byte array that, after the function return, will contain the data written into buffer EP8 of USB FX2 microcontroller. The data contained in EP8 is generated by the host computer.

```
long DataWriteLength
```

This parameter is the length (in bytes) of the previous parameter.

Return Value

int : integer type

This function returns true (ST_OK = 0) if it is able to write the data to buffer EP8 within Timeout milliseconds. This function returns false (ST_ERROR = 1) otherwise.

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