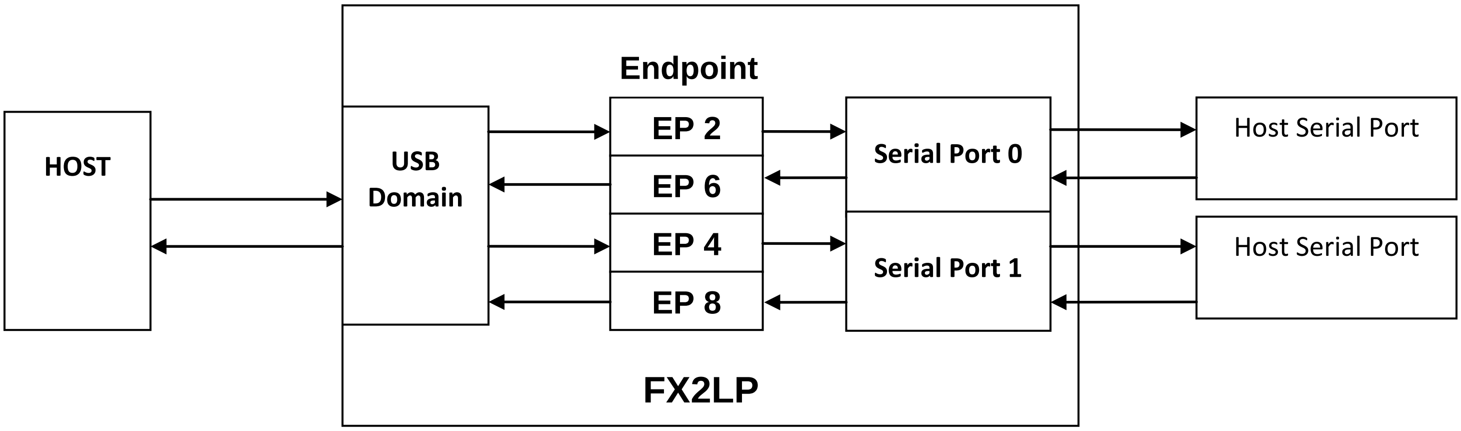
The document demonstrates the usage of both the serial ports together using the FX2LP CY3684 kit. The data from the serial ports is received and stored in the endpoints of FX2LP which can then be processed and retrieved from the USB side at any time using a vendor command (which is used to arm the endpoint) in the CyConsole. Similarly from the USB side when data is sent, it is stored in an endpoint FIFO, and can be sent via serial port anytime using a vendor command (which is used to trigger the transaction through serial port) and can be seen on HyperTerminal.

Firmware Description:

The written firmware will enable FX2LP to both send and receive data to and from the serial ports 0 and 1.
The data from the serial port 0 is received in Endpoint 6 and data in Endpoint 2 is sent to it.
The data from the serial port 1 is received in Endpoint 8 and data in Endpoint 4 is sent to it.

For implementing the serial ports, external Interrupts has been used. So that FX2LP can do any processing required in the TD_Poll() routine.

Hyperterminal is used to view and send the data through the serial ports. On the FX2LP side the data is sent and received using CyConsole.



The benefit of doing this is that the data received can be processed and sent via USB and data received through USB can be sent via Serial Port.

Serial Port 0 and 1 in FX2LP:

If we want to run both the ports on the same baud rate, we can use timer 1, as used in this example. The serial ports is configured in Asynchronous mode1 which has 8 data bits and no parity. The baud rate is chosen from the table 14-15 (of the TRM) as 9615 with CLKOUT of 48MHz.

Timer 1 Configuration and Registers:

Registers for Timer 1

1. We first select the clock for the timer , i.e. CLKOUT/12(for compatibility with standard 8051) or CLKOUT/4 . We choose CLKOUT/4 for Timer 1.
   CKCON= 0x10;
2. The timer is configured in mode 2.
   TMOD= 0x20;
3. The reload value is set to 0xB2.
   TH1=0xB2;
4. Enable the serial port 0/1 and set to mode 1,
   SCON0=0x50;
   SCON1=0x50;
5. The baud rates for both the ports are doubled.
   PCON=0x80;
   EICON=0x80;
6. Select Timer 1 for serial port 1 using register T2CON ;
   T2CON=0x00; //Selects timer 1 for Serial port 1.
   TCON=0x40; // Enables counting on timer 1.

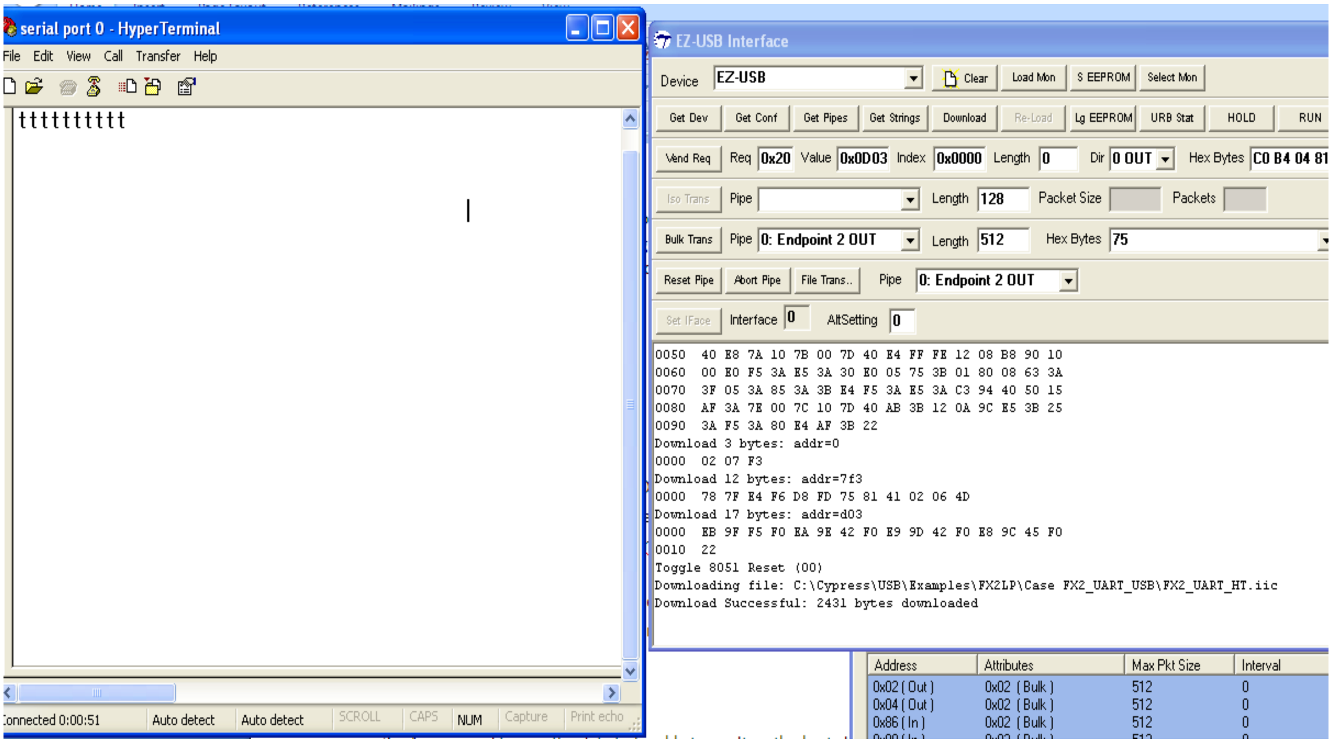
7. Enable the global interrupt EA and ES0 and ES1. (table 4-2 in TRM)
   EA =1;
   ES0 = 1;
   ES1 = 1;

Steps to follow to run the code:

To send data to FX2LP from Serial Port:

1. Download the code to the EEPROM of FX2LP using CyConsole. Open Hyperterminal session given along with the project. You need two serial ports, each associated with separate Hyperterminal session.

2. Type a few characters. As you type, the ISR gets executed and the data gets stored in the EP6(or EP8) endpoint for data sent through serial port 0(or serial port1). (In the example below, serial port 1 is used)



```
tttttttttt
```

Device: EZ-USB

Vend Req — Req: 0x20 Value: 0x0D03 Index: 0x0000 Length: 0 Dir: 0 OUT Hex Bytes: C0 B4 04 81

Iso Trans — Pipe: Length: 128 Packet Size: Packets:

Bulk Trans — Pipe: 0: Endpoint 2 OUT Length: 512 Hex Bytes: 75

Pipe: 0: Endpoint 2 OUT

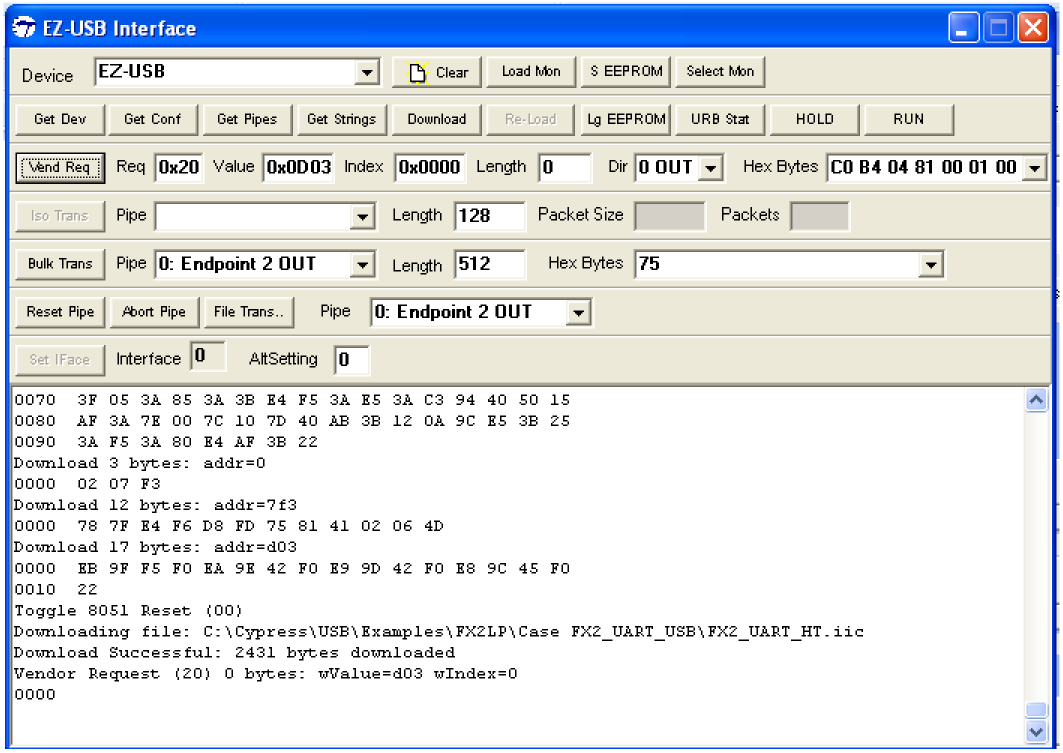
Interface: 0 AltSetting: 0

```
0050  40 E8 7A 10 7B 00 7D 40 E4 FF FE 12 08 B8 90 10
0060  00 E0 F5 3A E5 3A 30 E0 05 75 3B 01 80 08 63 3A
0070  3F 05 3A 85 3A 3B E4 F5 3A E5 3A C3 94 40 50 15
0080  AF 3A 7E 00 7C 10 7D 40 AB 3B 12 0A 9C E5 3B 25
0090  3A F5 3A 80 E4 AF 3B 22
Download 3 bytes: addr=0
0000  02 07 F3
Download 12 bytes: addr=7f3
0000  78 7F E4 F6 D8 FD 75 81 41 02 06 4D
Download 17 bytes: addr=d03
0000  EB 9F F5 F0 EA 9E 42 F0 E9 9D 42 F0 E8 9C 45 F0
0010  22
Toggle 8051 Reset (00)
Downloading file: C:\Cypress\USB\Examples\FX2LP\Case FX2_UART_USB\FX2_UART_HT.iic
Download Successful: 2431 bytes downloaded
```

Address	Attributes	Max Pkt Size	Interval
0x02 (Out)	0x02 (Bulk)	512	0
0x04 (Out)	0x02 (Bulk)	512	0
0x86 (In)	0x02 (Bulk)	512	0

3. As you need to arm the data to be able to see it on the host side (using CyConsole), Use a vendor command 0x10 (or 0x20) in case of serial port 0 (or serial port 1) to arm it using the EP6(or EP8) BCH/L registers.

    (Serial port 1 is used, therefore vendor command used is 0x20)



Device EZ-USB

Vend Req: Req 0x20 Value 0x0D03 Index 0x0000 Length 0 Dir 0 OUT Hex Bytes C0 B4 04 81 00 01 00

Iso Trans: Pipe Length 128 Packet Size Packets

Bulk Trans: Pipe 0: Endpoint 2 OUT Length 512 Hex Bytes 75

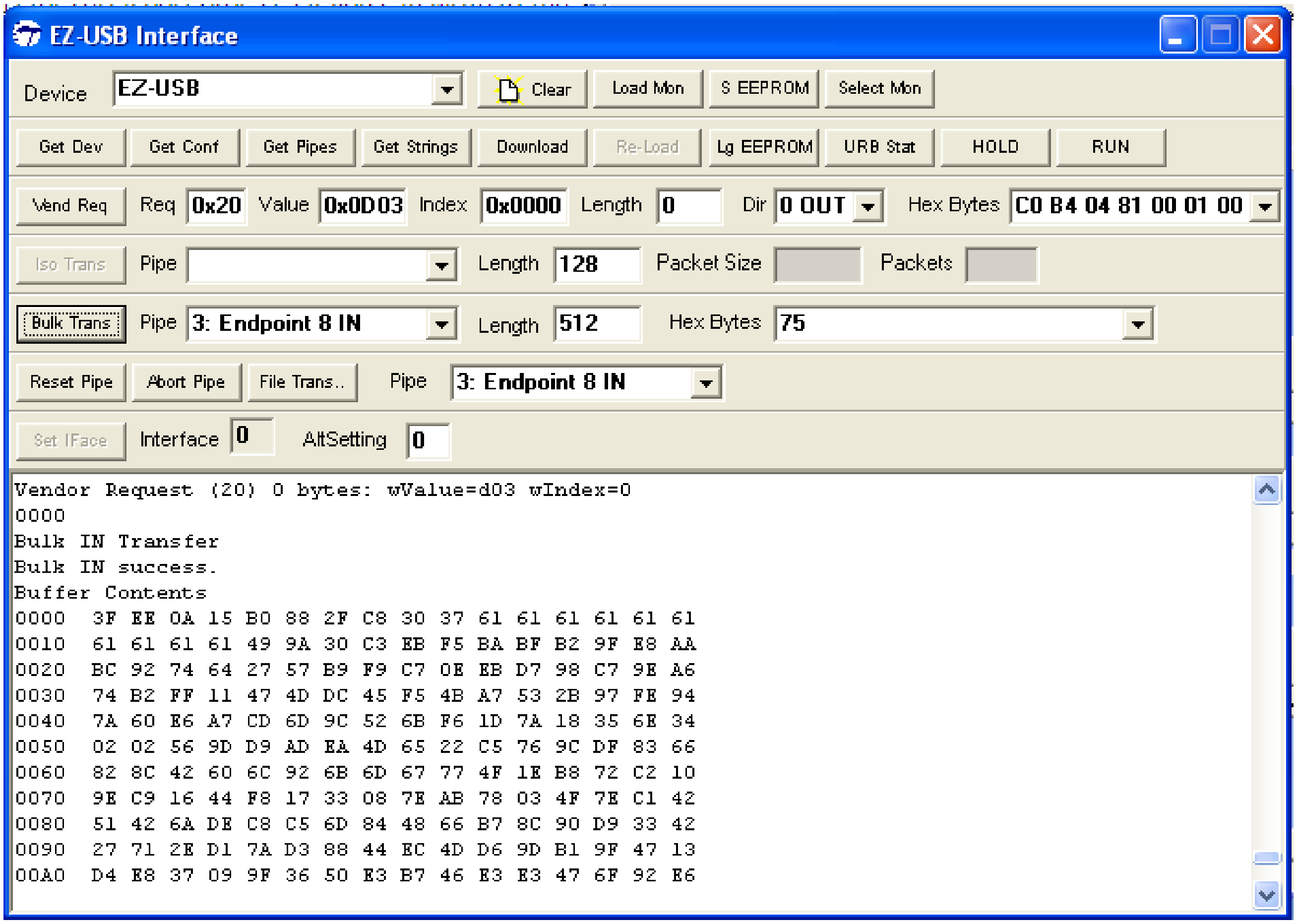
Pipe 0: Endpoint 2 OUT

Interface 0 AltSetting 0

```
0070  3F 05 3A 85 3A 3B E4 F5 3A E5 3A C3 94 40 50 15
0080  AF 3A 7E 00 7C 10 7D 40 AB 3B 12 0A 9C E5 3B 25
0090  3A F5 3A 80 E4 AF 3B 22
Download 3 bytes: addr=0
0000  02 07 F3
Download 12 bytes: addr=7f3
0000  78 7F E4 F6 D8 FD 75 81 41 02 06 4D
Download 17 bytes: addr=d03
0000  EB 9F F5 F0 EA 9E 42 F0 E9 9D 42 F0 E8 9C 45 F0
0010  22
Toggle 8051 Reset (00)
Downloading file: C:\Cypress\USB\Examples\FX2LP\Case FX2_UART_USB\FX2_UART_HT.iic
Download Successful: 2431 bytes downloaded
Vendor Request (20) 0 bytes: wValue=d03 wIndex=0
0000
```

4. After using the vendor command, you can do a Bulk IN transfer (from endpoint 6/8) in CyConsole to see the data sent via hyperterminal.
   (Serial Port 1 is used, so Bulk In Transfer done from EP8)

The whole process is repeated after this, you would see the data getting stored after the number of characters (as sent first)in the endpoint buffer. i.e. in this case you would see the data after 10 bytes as shown in the figure.



Device: EZ-USB

Req 0x20 Value 0x0D03 Index 0x0000 Length 0 Dir 0 OUT Hex Bytes C0 B4 04 81 00 01 00

Pipe: Length 128 Packet Size Packets

Pipe 3: Endpoint 8 IN Length 512 Hex Bytes 75

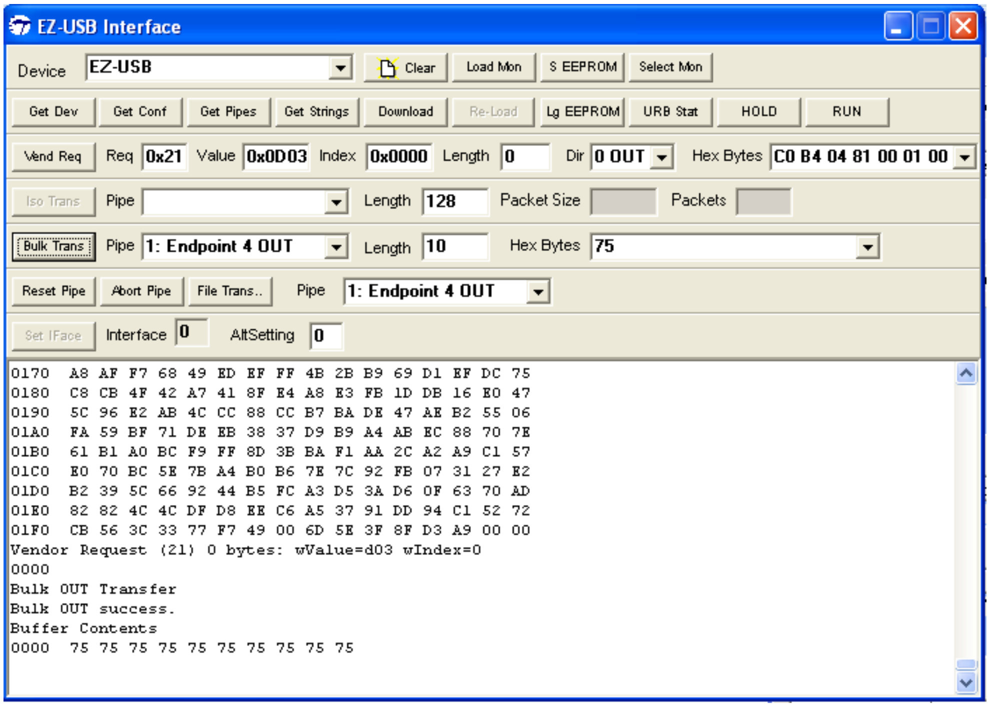
Pipe 3: Endpoint 8 IN

Interface 0 AltSetting 0

```
Vendor Request (20) 0 bytes: wValue=d03 wIndex=0
0000
Bulk IN Transfer
Bulk IN success.
Buffer Contents
0000  3F EE 0A 15 B0 88 2F C8 30 37 61 61 61 61 61 61
0010  61 61 61 61 49 9A 30 C3 EB F5 BA BF B2 9F E8 AA
0020  BC 92 74 64 27 57 B9 F9 C7 0E EB D7 98 C7 9E A6
0030  74 B2 FF 11 47 4D DC 45 F5 4B A7 53 2B 97 FE 94
0040  7A 60 E6 A7 CD 6D 9C 52 6B F6 1D 7A 18 35 6E 34
0050  02 02 56 9D D9 AD EA 4D 65 22 C5 76 9C DF 83 66
0060  82 8C 42 60 6C 92 6B 6D 67 77 4F 1E B8 72 C2 10
0070  9E C9 16 44 F8 17 33 08 7E AB 78 03 4F 7E C1 42
0080  51 42 6A DE C8 C5 6D 84 48 66 B7 8C 90 D9 33 42
0090  27 71 2E D1 7A D3 88 44 EC 4D D6 9D B1 9F 47 13
00A0  D4 E8 37 09 9F 36 50 E3 B7 46 E3 E3 47 6F 92 E6
```

To send data to Hyperterminal from FX2LP:

1. The data can be sent to the buffer using CyConsole Bulk OUT Transaction (in this case to Endpoint 4).



Device EZ-USB

Req 0x21 Value 0x0D03 Index 0x0000 Length 0 Dir 0 OUT Hex Bytes C0 B4 04 81 00 01 00

Pipe Length 128 Packet Size Packets

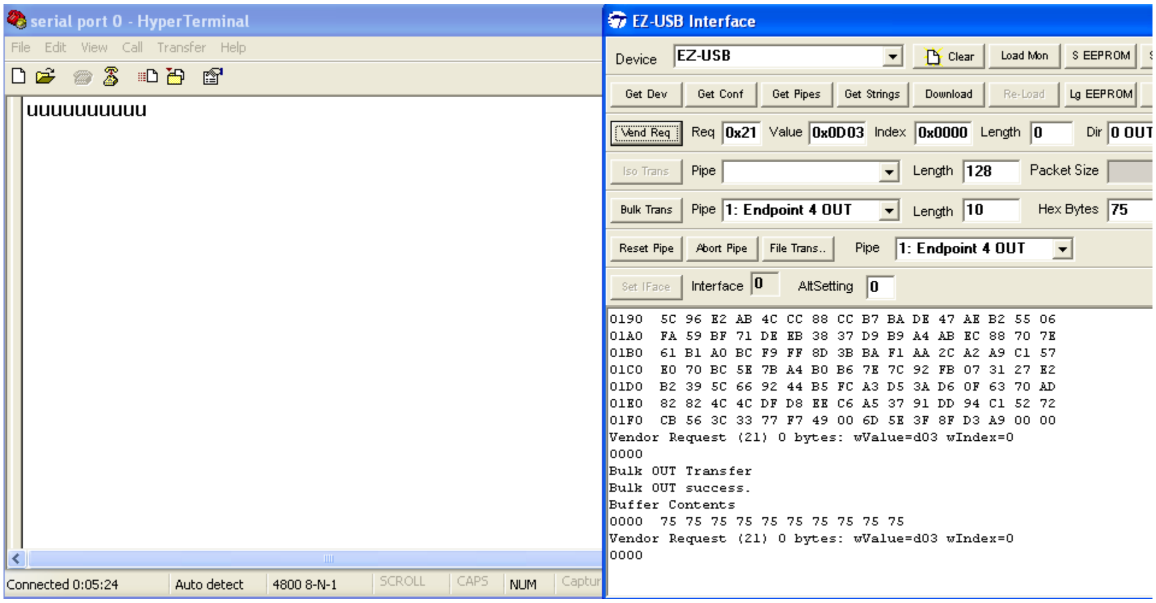
Pipe 1: Endpoint 4 OUT Length 10 Hex Bytes 75

Pipe 1: Endpoint 4 OUT

Interface 0 AltSetting 0

```
0170  A8 AF F7 68 49 ED EF FF 4B 2B B9 69 D1 EF DC 75
0180  C8 CB 4F 42 A7 41 8F E4 A8 E3 FB 1D DB 16 E0 47
0190  5C 96 E2 AB 4C CC 88 CC B7 BA DE 47 AE B2 55 06
01A0  FA 59 BF 71 DE EB 38 37 D9 B9 A4 AB EC 88 70 7E
01B0  61 B1 A0 BC F9 FF 8D 3B BA F1 AA 2C A2 A9 C1 57
01C0  E0 70 BC 5E 7B A4 B0 B6 7E 7C 92 FB 07 31 27 E2
01D0  B2 39 5C 66 92 44 B5 FC A3 D5 3A D6 0F 63 70 AD
01E0  82 82 4C 4C DF D8 EE C6 A5 37 91 DD 94 C1 52 72
01F0  CB 56 3C 33 77 F7 49 00 6D 5E 3F 8F D3 A9 00 00
Vendor Request (21) 0 bytes: wValue=d03 wIndex=0
0000
Bulk OUT Transfer
Bulk OUT success.
Buffer Contents
0000  75 75 75 75 75 75 75 75 75 75
```

2. To trigger sending data, set the TI flag for each serial port using the Vendor Command 0x11 (or 0x21) for serial port 0(or 1). Or you can write to the SBUF register, which triggers the data flow. (In this example, Vendor command 0x21)



```
uuuuuuuuuu
```

Connected 0:05:24 | Auto detect | 4800 8-N-1 | SCROLL | CAPS | NUM | Captur

Device EZ-USB

Req 0x21 Value 0x0D03 Index 0x0000 Length 0 Dir 0 OUT

Pipe Length 128 Packet Size

Pipe 1: Endpoint 4 OUT Length 10 Hex Bytes 75

Pipe 1: Endpoint 4 OUT

Interface 0 AltSetting 0

```
0190  5C 96 E2 AB 4C CC 88 CC B7 BA DE 47 AE B2 55 06
01A0  FA 59 BF 71 DE EB 38 37 D9 B9 A4 AB EC 88 70 7E
01B0  61 B1 A0 BC F9 FF 8D 3B BA F1 AA 2C A2 A9 C1 57
01C0  E0 70 BC 5E 7B A4 B0 B6 7E 7C 92 FB 07 31 27 E2
01D0  B2 39 5C 66 92 44 B5 FC A3 D5 3A D6 0F 63 70 AD
01E0  82 82 4C 4C DF D8 EE C6 A5 37 91 DD 94 C1 52 72
01F0  CB 56 3C 33 77 F7 49 00 6D 5E 3F 8F D3 A9 00 00
Vendor Request (21) 0 bytes: wValue=d03 wIndex=0
0000
Bulk OUT Transfer
Bulk OUT success.
Buffer Contents
0000  75 75 75 75 75 75 75 75 75 75
Vendor Request (21) 0 bytes: wValue=d03 wIndex=0
0000
```

3. As the TI flag gets set, the ISR will start sending first ten bytes of data from EP2 (or EP4) endpoint.

Note:

1. The code is only to introduce the concepts of external interrupts used, the use of timers for the baud rate generation to run a serial port.
2. The code is checked using two computers.
3. For details on how to use and implement ISR for the Serial Port, please refer to the code example: **CE59058 – External Interrupts in FX2LP**.
4. With the baud rate of 9600, the code does not work sometimes. **If we take a lower baud rate, the code works properly.** These are issues with the timing signals, as the actual baud rate of the serial ports is 9615 (with an error of .16 %) in FX2LP and on the hyperterminal side its 9600. One can also try changing the cable.

   For lower Baud Rate of 4807, TH1 register must be changed to TH1= 0x64; and the hyperterminal baud rate selected should be 4800.
5. **The Hyperterminal Settings** :

   Baud Rate: 9600
   Flow Control: None (other settings remain as default)
   And you can select the ASCII setup to echo the character on the terminal.
6. Please find the project attached.