USBFS Suspend Code Example

1.0

Features

- USB device suspend and resume
- Low-power modes

General Description

This example project demonstrates the ability of the USBFS component to detect a suspend condition on the USB bus and resume its operation when a resume condition is detected and the USB bus activity restored.

The PSoC operates in the active power mode before the host decides to suspend the USB device. The PSoC mode is changed to low power as soon as a suspend condition is detected to consume less power. The USB device maximum suspend current should not exceed 500uA to be compliant with the USB specification. When the host decides to wake up the USB device it drives a resume condition on the bus then PSoC wakes up and changes its power mode to active.

The LED is used to indicate the USB device state as well as the PSoC power mode. The LED is on when the USB bus is active and PSoC is in the active mode. The LED is off after a suspend condition is detected and PSoC is in the low-power mode.

NOTE Hereinafter, the red LED is referred to as the LED for the CY8CKIT-046 kit.

Development Kit Configuration

This example project is designed to run on the CY8CKIT-046 kit from Cypress Semiconductor. A description of the kit, along with more code examples and ordering information, can be found at http://www.cypress.com/go/cy8ckit-046.

The project requires configuration settings changes to run on other kits from Cypress Semiconductor. Table 1 is the list of the supported kits. To switch from CY8CKIT-046 to any other kit, change the project’s device with the help of Device Selector called from the project’s context menu.

Table 1. Development Kits vs Parts

Development Kit	Device
CY8CKIT-046	CY8C4248BZI-L489
CY8CKIT-030	CY8C3866AXI-040
CY8CKIT-050	CY8C5868AXI_LP035

Development Kit	Device
CY8CKIT-001	CY8C3866AXI-040/ CY8C5868AXI_LP035

The pins assignment for the supported kits is in Table 2.

Table 2. Pins Assignment

Pin Name	Development Kit			
	CY8CKIT-046	CY8CKIT-030	CY8CKIT-050	CY8CKIT-001
\USBFS:Dm\	P13[1]	P15[7]	P15[7]	P15[7]
\USBFS:Dp\	P13[0]	P15[6]	P15[6]	P15[6]
LED_RED	P5[2]	–	–	–
LED_GREEN	P5[3]	–	–	–
LED_BLUE	P5[4]	–	–	–
LED4	–	P6[3]	P6[3]	P6[3]

To handle hardware differences between the supported kits, separate TopDesigns and control files are added to the project. The control files monitor the pins assignment depending on the selected device. Manual placement of the pins overrides the control file directives, therefore the pins in the Design Wide Resource (DWR) file should be unlocked. All these files can be found in the **Components** tab of the workspace explorer.

Project Configuration

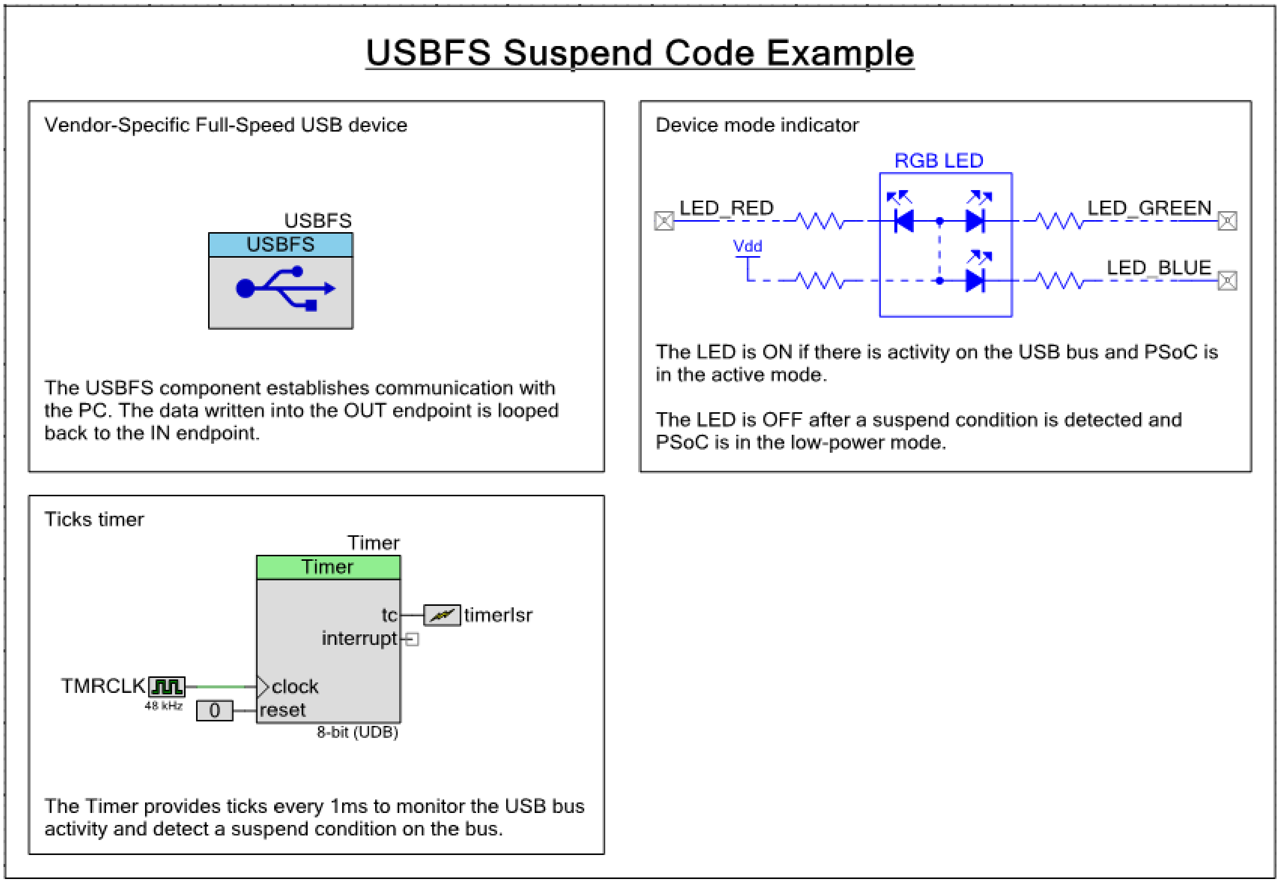
The example project consists of the following components: USBFS, Timer, and pins. The project schematic is in Figure 1.

The USBFS is used to establish communication with a PC. The USB device’s VID is 0x4B4 (this is Cypress Semiconductor vendor ID) and PID is 0x8051 (arbitrarily chosen). The appropriate string descriptors are also provided with the company name and example project name definition. The device is powered through the USB connection so the device configuration is bus-powered. The device has two BULK endpoints: EP1 IN and EP2 OUT. The OUT endpoint allows the host to write data into the device and the IN endpoint allows the host to read data from the device. Each endpoint maximum packet is 64 bytes, therefore the up to 64 bytes can be transferred.

The Timer function provides the system with ticks every 1ms. Every tick, the USB bus activity is checked to identify if the bus is idle. When the bus idle time is longer than 3ms, it is equal to the USB suspend condition.

The pin components are used to control the LED that indicates the USB device state as well as PSoC power mode. The LED is on when the USB bus is active and PSoC is in the active mode. The LED is off after a suspend condition is detected and PSoC is in the low-power mode. For PSoC4, the low-power mode is Deep Sleep and for PSoC 3/PSoC 5LP – Sleep. The LED’s off state in the suspend mode also ensures that the power consumption requirements of the USB device in the suspend mode (< 500uA) are met.

Figure 1. Example Project Design Schematic PSoC 4200



The USBFS component establishes communication with the PC. The data written into the OUT endpoint is looped back to the IN endpoint.

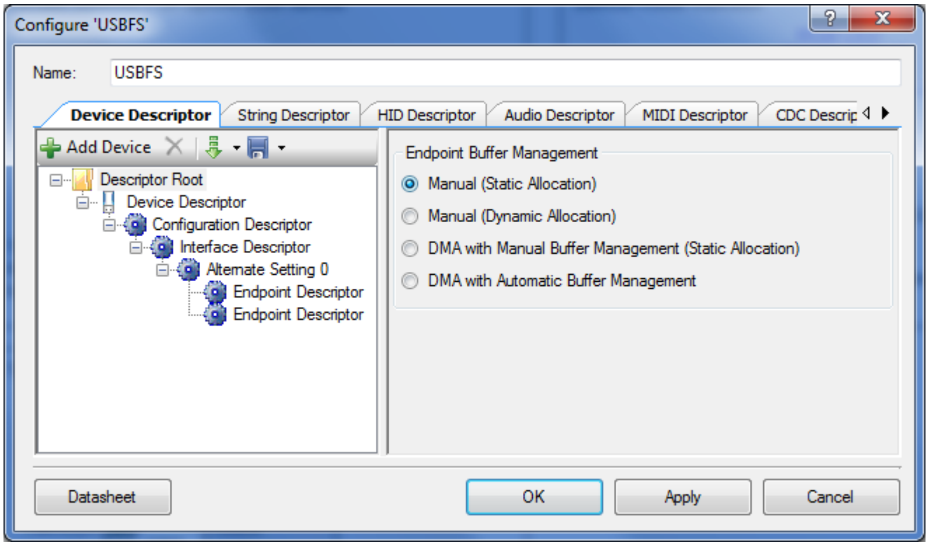
The LED is ON if there is activity on the USB bus and PSoC is in the active mode.

The LED is OFF after a suspend condition is detected and PSoC is in the low-power mode.

The Timer provides ticks every 1ms to monitor the USB bus activity and detect a suspend condition on the bus.

The important USBFS component configuration Tabs are shown below.

Figure 2. Descriptor Root



Name: USBFS

Endpoint Buffer Management

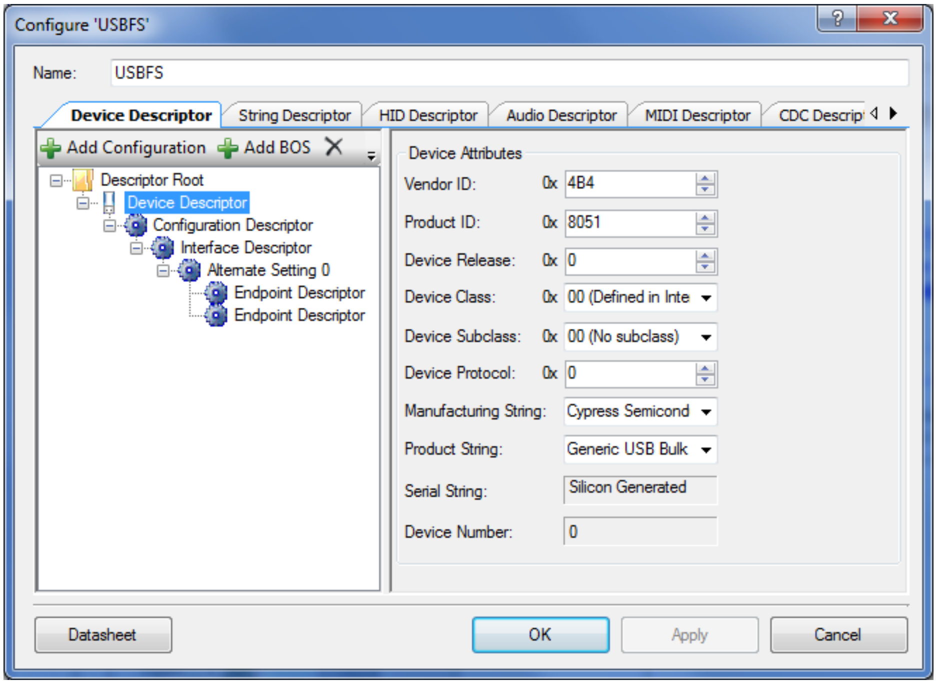
● Manual (Static Allocation)

○ Manual (Dynamic Allocation)

○ DMA with Manual Buffer Management (Static Allocation)

○ DMA with Automatic Buffer Management

Figure 3. Device Descriptor



Name: USBFS

Vendor ID: 0x 4B4

Product ID: 0x 8051

Device Release: 0x 0

Device Class: 0x 00 (Defined in Inte

Device Subclass: 0x 00 (No subclass)

Device Protocol: 0x 0

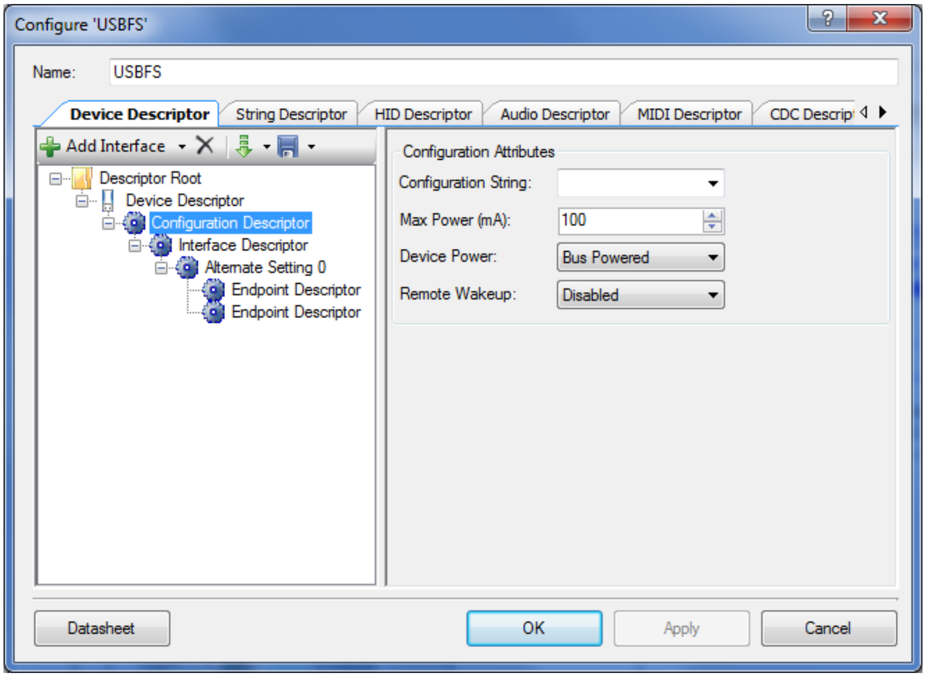
Manufacturing String: Cypress Semicond

Product String: Generic USB Bulk

Serial String: Silicon Generated

Device Number: 0

Figure 4. Configuration Descriptor



Name: USBFS

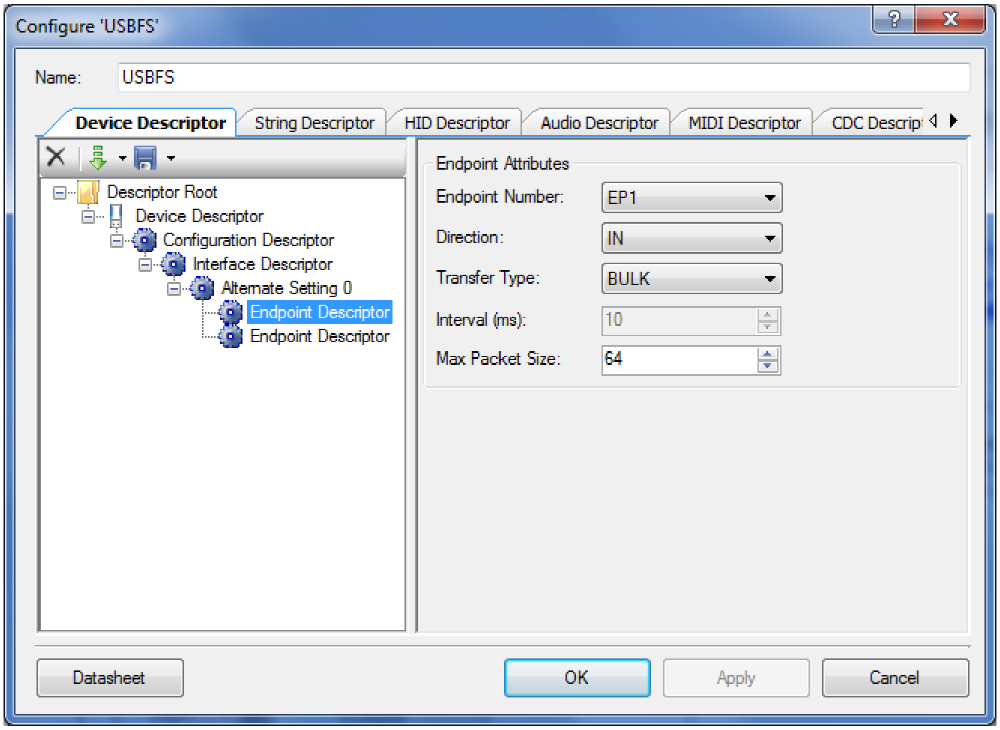
Configuration String:

Max Power (mA): 100

Device Power: Bus Powered

Remote Wakeup: Disabled

Figure 5. Endpoint 1 Descriptor



Configure 'USBFS'

Name: USBFS

- Descriptor Root
  - Device Descriptor
    - Configuration Descriptor
      - Interface Descriptor
        - Alternate Setting 0
          - Endpoint Descriptor
          - Endpoint Descriptor

Endpoint Attributes

Endpoint Number: EP1

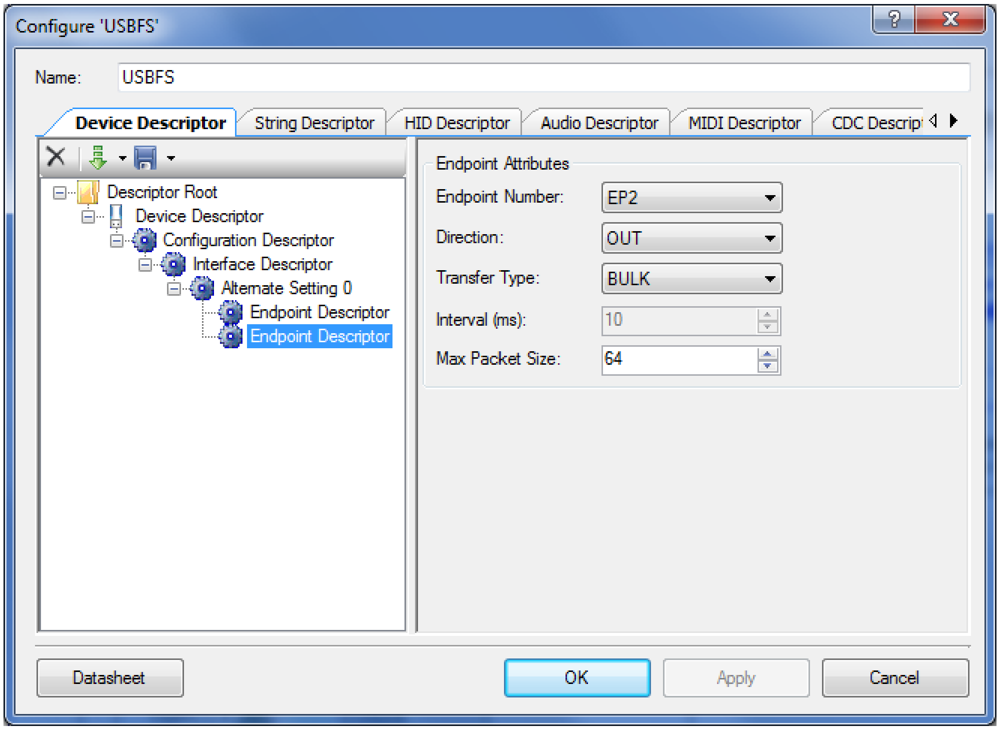
Direction: IN

Transfer Type: BULK

Interval (ms): 10

Max Packet Size: 64

Figure 6. Endpoint 2 Descriptor



Configure 'USBFS'

Name: USBFS

- Descriptor Root
  - Device Descriptor
    - Configuration Descriptor
      - Interface Descriptor
        - Alternate Setting 0
          - Endpoint Descriptor
          - Endpoint Descriptor

Endpoint Attributes

Endpoint Number: EP2

Direction: OUT

Transfer Type: BULK

Interval (ms): 10

Max Packet Size: 64

Project Description

In the main firmware routine, the USBFS component is configured for operation and started. The code waits until the USBFS device is enumerated and the LED is turned on. After enumeration, the device runs in the active mode and alternate between the two tasks:

- Task 1 (Bulk wraparound). The OUT endpoint buffer is checked for data availability and the received data is loaded into the IN endpoint buffer (if there is space). If the IN endpoint buffer is full, the data remains in the RAM buffer until new data is received or the IN endpoint buffer becomes empty.
- Task 2 (Detection of the USB bus suspend condition). The USB device is suspended if there is no activity on the bus for longer than 3ms. Normally, SOF (Start-Of-Frame) packets are sent by the host every 1ms. The USBFS component is not capable to track the suspend condition by itself. Therefore, a timer is placed in the design to provide ticks each 1ms. The USB bus activity is checked every tick to identify when USB bus is idle for 3ms.

After the USB suspend condition is detected, the PSoC leaves the active mode and enters the low power mode (for PSoC 4200L it is Deep Sleep whereas for PSoC 3/PSoC 5LP it is Sleep). This allows the device to reduce power consumption while the host does not communicate with it. Before going to the low power mode, the USBFS component stores its current configuration and sets up a wakeup source calling the USBFS_Suspend() function. The wakeup source is the falling edge on the USB Dp pin.

When the host wants to wake the device up after suspend, it does so by reversing the polarity of the signal on the data lines for at least 20ms. The signal is completed with a low-speed end of the packet signal. As soon as the Dp pin polarity is reverted, the PSoC wakes up because a wakeup event occurs (falling edge on Dp pin). The USBFS component configuration is restored calling the USBFS_Resume() function. The OUT endpoint is enabled to allow host communicate with the USB device. After all restore functions are completed for other components, the device returns into the active mode and executes its tasks.

Note that USBFS data endpoints require re-initialization after wakeup from the low-power mode including enabling OUT endpoints and loading IN endpoints. The data available in the data IN or OUT endpoints is not retained in the low-power mode and lost. It is the application responsibility to manage this properly.

Example Project Execution Flow

To execute the USBFS component code example follow the procedure:

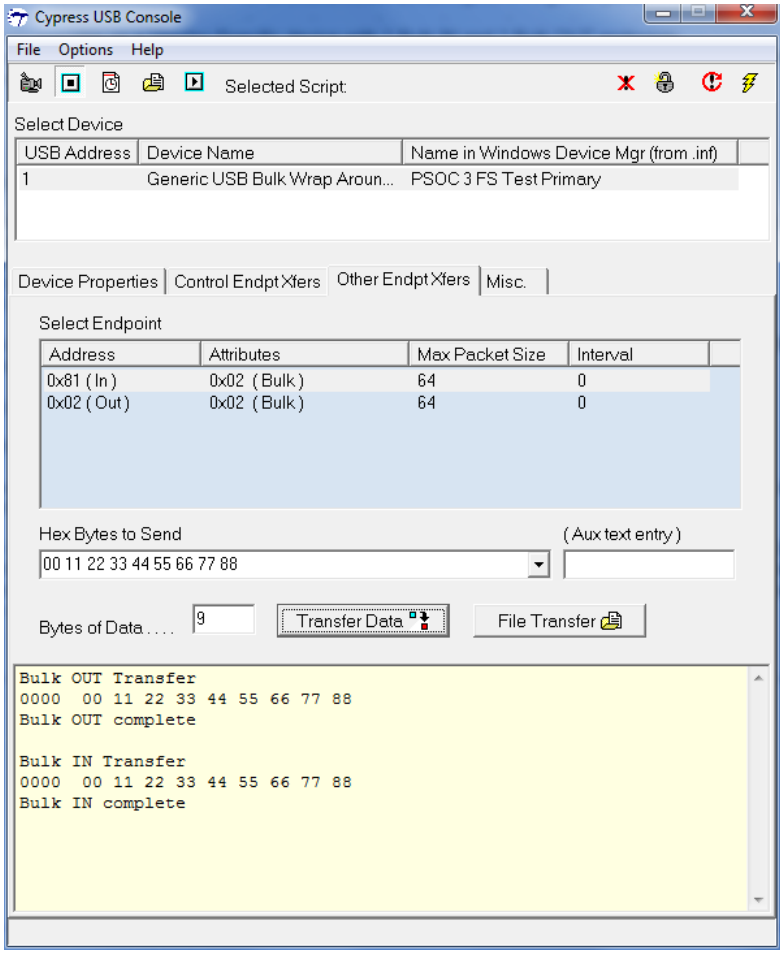
1. Connect the PSoC kit to the PC through a USB connector for programming.
2. Build the USBFS_Suspend example project and program it into the device. The USB cable can be disconnected from the programming connector at this point.
3. Install SuiteUSB 3.4 (follow the link for downloads and installation instructions) to get USB driver and CyConsole application which is required to in the following steps.

4. Add the device's **VendorID** (0x04B4) and **ProductID** (0x8051) to the **CYUSB.INF** file. After installation of the Cypress Suite USB installer, the driver file is located in the Driver subdirectory of the install directory (Default is C:\Program Files\Cypress\Cypress Suite USB 3.4.7\Driver\bin). The detailed instructions are in the **CyUSB.pdf** file shipped with the suite.
5. Connect the kit to the PC through a USB connector for communication. After the device is enumerated the LED is turned on notifying that the USB device is not suspended and PSoC is in the active mode.

   Note the USBFS device does not have signed driver. To make the device work for Windows 7, reboot PC as normally pressing F8 repeatedly while the boot process is running. When the boot menu appears, select **Disable Driver Signature Enforcement**. This option is temporary – on the next reboot, driver signing will be activated again.
6. To communicate with the USB device, run Cypress USB Console application (CyConsole). Write up to 64 bytes into the OUT endpoint then read back the same data from the IN endpoint.

   The host fails to read the IN endpoint when data has not been loaded in its buffer. This can happen in two cases: the OUT endpoint has not been written before read data from the IN endpoint or the loaded data is lost after resume due to entering the low-power mode (the endpoint data memory is not retained).

Figure 7. Cypress USB Console Output



Select Device

USB Address	Device Name	Name in Windows Device Mgr (from .inf)
1	Generic USB Bulk Wrap Aroun...	PSOC 3 FS Test Primary

Select Endpoint

Address	Attributes	Max Packet Size	Interval
0x81 (In)	0x02 (Bulk)	64	0
0x02 (Out)	0x02 (Bulk)	64	0

Hex Bytes to Send: 00 11 22 33 44 55 66 77 88

(Aux text entry)

Bytes of Data 9

```
Bulk OUT Transfer
0000  00 11 22 33 44 55 66 77 88
Bulk OUT complete

Bulk IN Transfer
0000  00 11 22 33 44 55 66 77 88
Bulk IN complete
```

7. To put the device into suspend, go to the Start menu and put the PC into the sleep mode. This makes the USB traffic stop and PSoC enters the low-power mode, which is notified by the LED turn off.
8. Use a mouse or key board to wake up the PC from the sleep mode. The host generates a resume condition on the USB bus and PSoC exits low power and restores communication through the USB (the device is not enumerated again).
9. After the PC is awakened, the communication with the USBFS device can be repeated as described in step 5.

Expected Results

If you follow the instructions above, the Vendor-Specific USB Full-Speed device is enumerated by the PC with BULK IN and OUT endpoints and then you can communicate with the USB device: receive data from the PC via the OUT endpoint and to loop that data back using the IN endpoint using Cypress USB Console application.

The USB device detects when the host drives a suspend or resume condition on the bus and changes its power mode appropriately. The LED is off when the device is suspended and on if not. The USB device configuration is restored after it is suspended and it can continue communicating with the host.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.