

USBFS UART Code Example

v2.0

Features

- Enumerates as Virtual COM port.
- Sends received data backward to COM-port terminal
- Output virtual COM port settings to LCD (PSoC3/5LP only)

General Description

This code example demonstrates the USBUART implementation. It echoes received data to the Virtual COM port terminal.

Development Kit Configuration

The example project runs on the CY8CKIT-046 kit from Cypress Semiconductor. A description of the kit, along with more code examples and ordering information is 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-001	CY8C3866AXI-040/ CY8C5868AXI_LP035
CY8CKIT-046	CY8C4248BZI-L489
CY8CKIT-030	CY8C3866AXI-040
CY8CKIT-050	CY8C5868AXI_LP035

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

Table 2. Pin Assignment

Pin Name	Development Kit			
	CY8CKIT-001	CY8CKIT-046	CY8CKIT-030	CY8CKIT-050
\\USBFS:Dm\\	P15[7]	P13[1]	P15[7]	P15[7]
\\USBFS:Dp\\	P15[6]	P13[0]	P15[6]	P15[6]
\\LCD:LCDPort[6:0]\\	P2[6:0]	-	P2[6:0]	P2[6:0]

Note The example project control file handles the pin placement automatically.

Project Configuration

The example project consists of the USBUART and LCD (PSoC 3/5LP only) components. The project schematic is in Figure 1(PSoC 4200L) and Figure 2 (PSoC 3/5LP).

Figure 1. Example Project Design Schematic PSoC 4200L

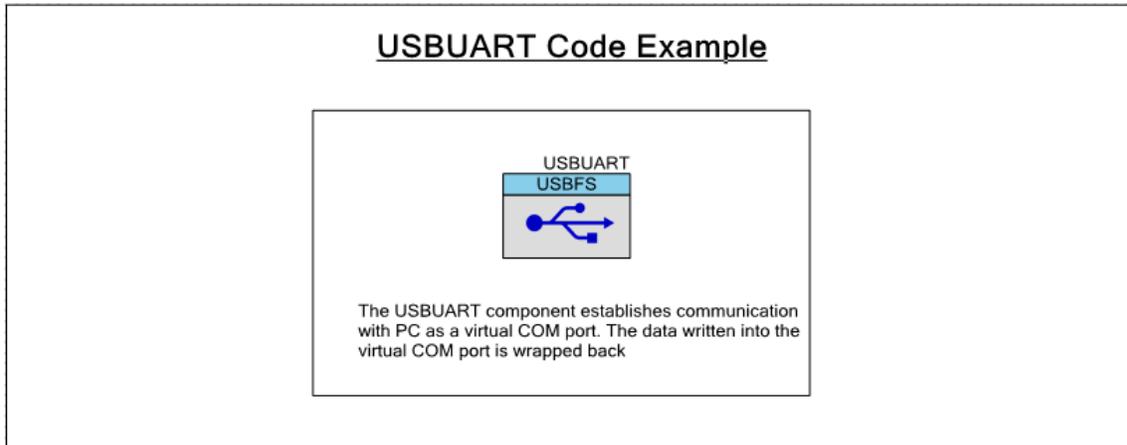
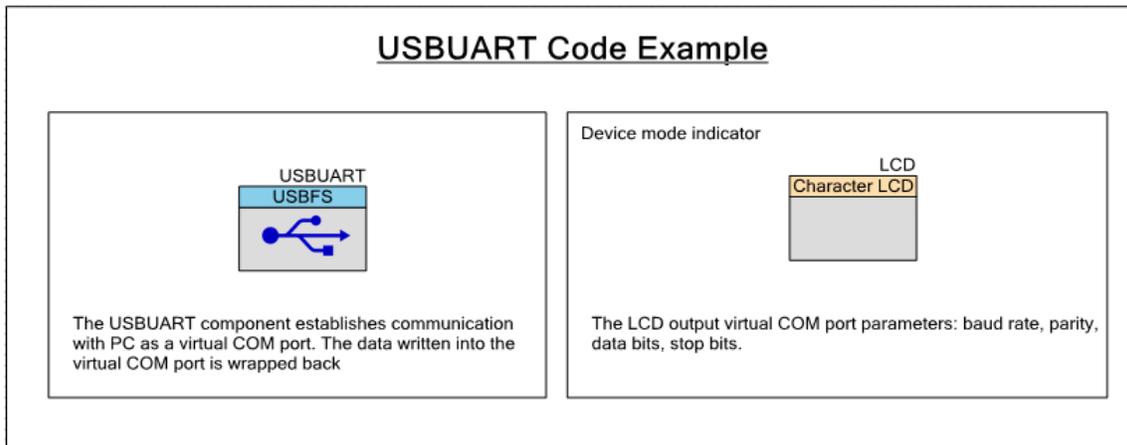


Figure 2. Example Project Design Schematic PSoC 3/5LP



The important USBFS component configuration Tabs are on the screenshots below.

Figure 3. USBUART Root Descriptor

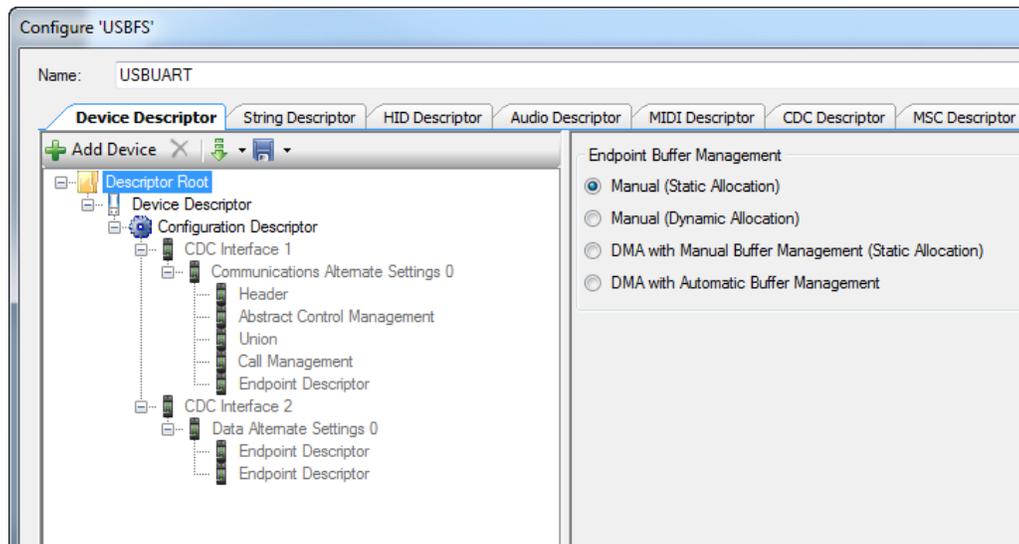


Figure 4. USBUART Device Descriptor

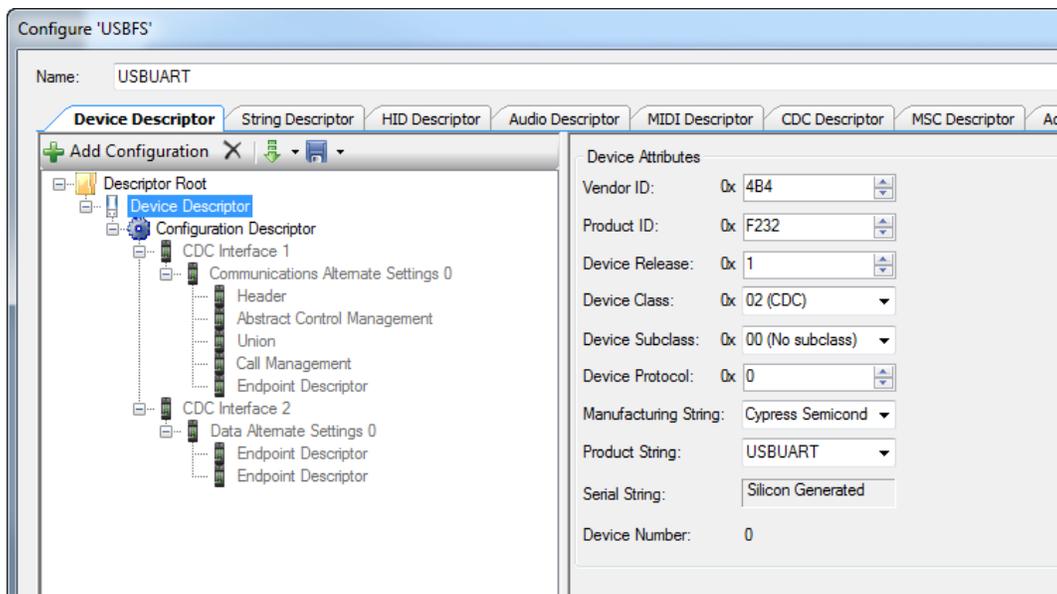


Figure 5. USBUART Configuration Descriptor

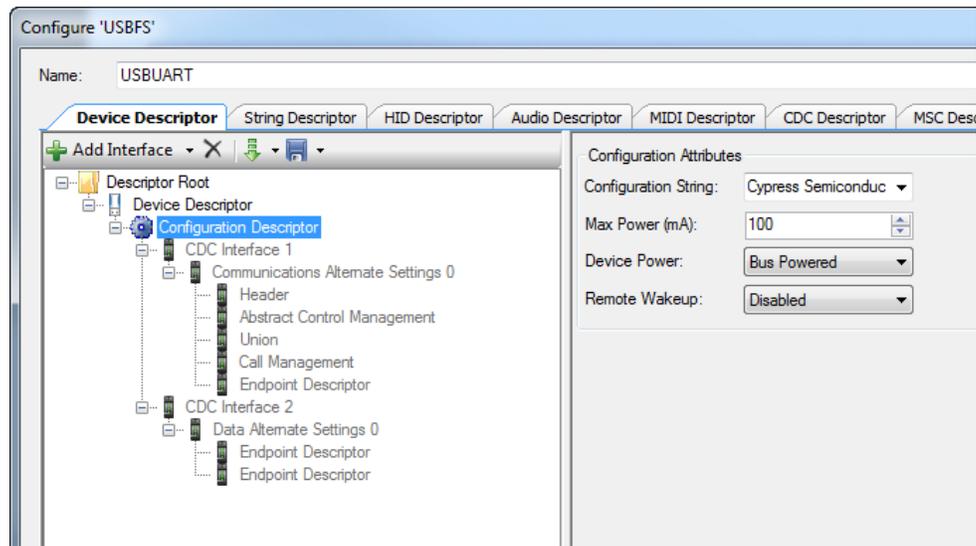


Figure 6. USBUART Communication Interface Descriptor

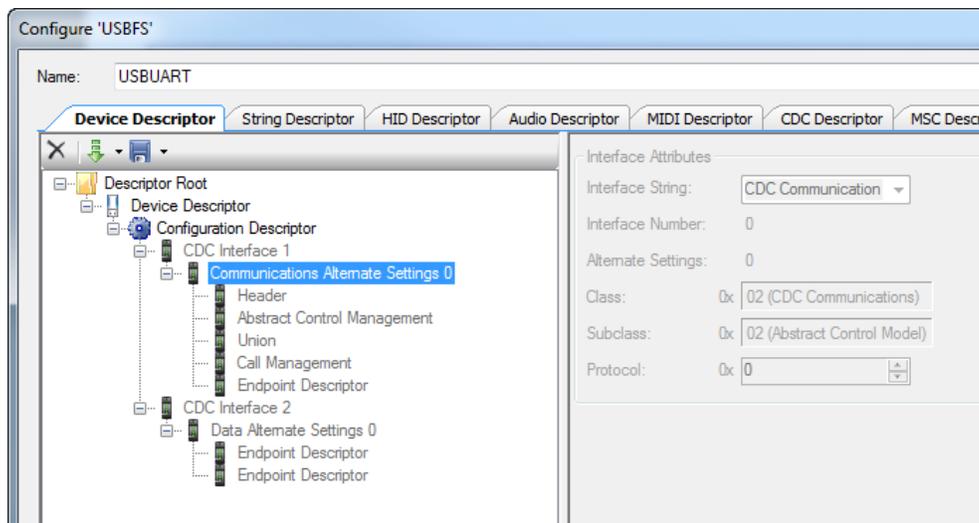


Figure 7. USBUART Notification Endpoint 1 Descriptor

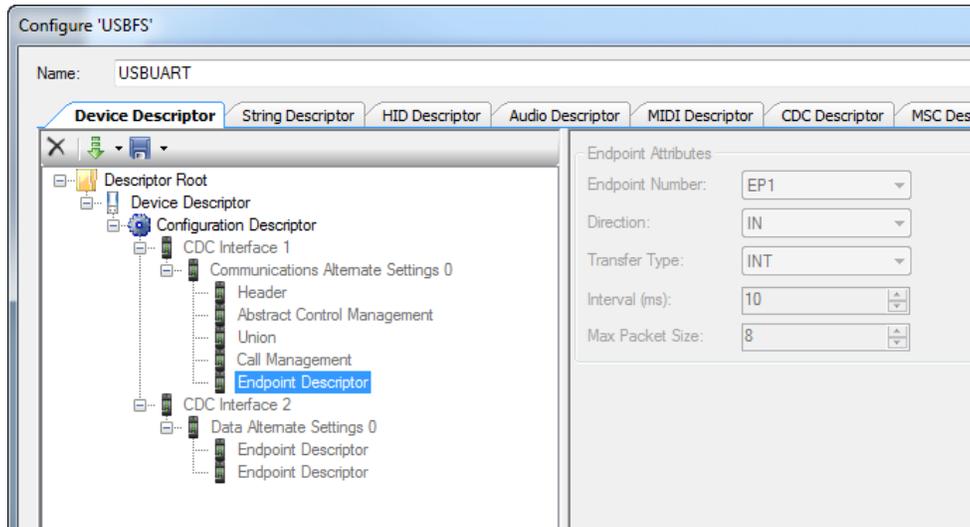


Figure 8. USBUART Data Interface Descriptor

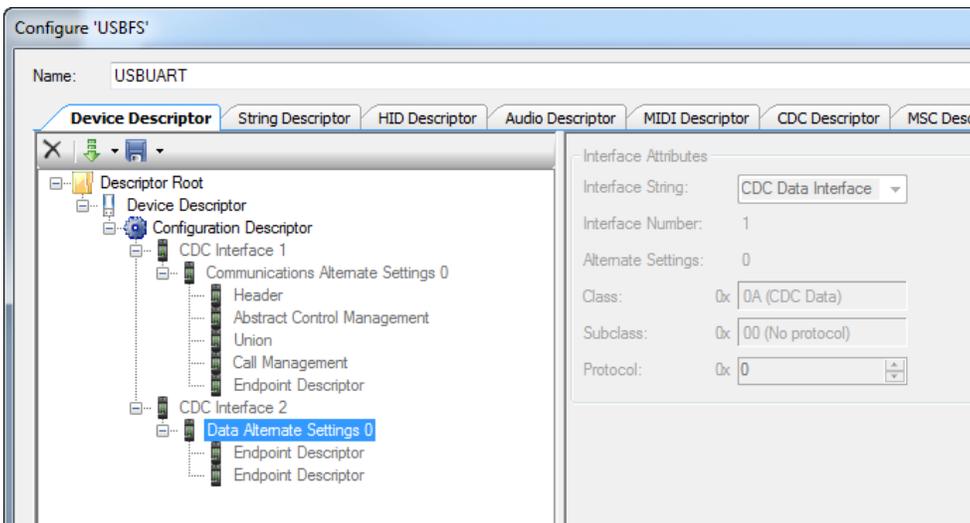


Figure 9. USBUART IN Data Endpoint 2 Descriptor

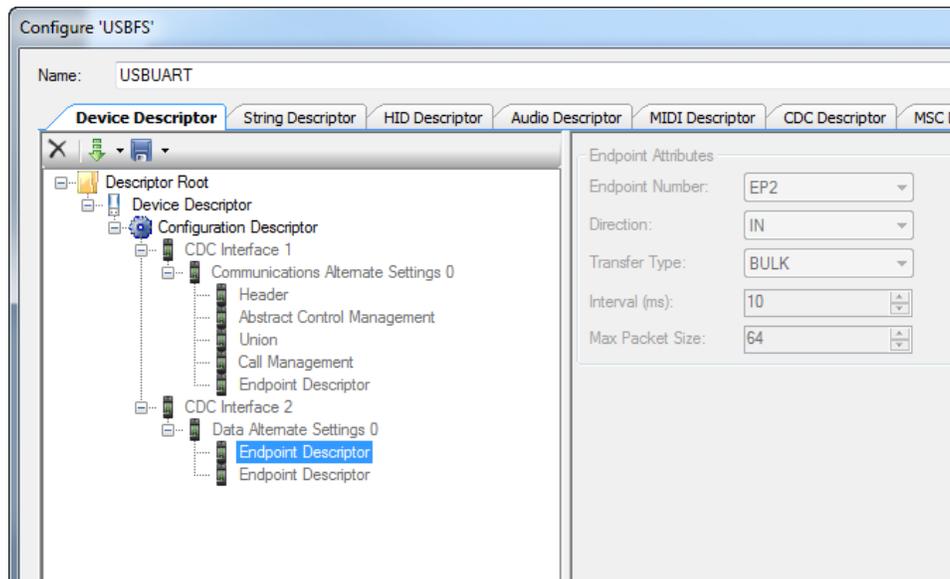
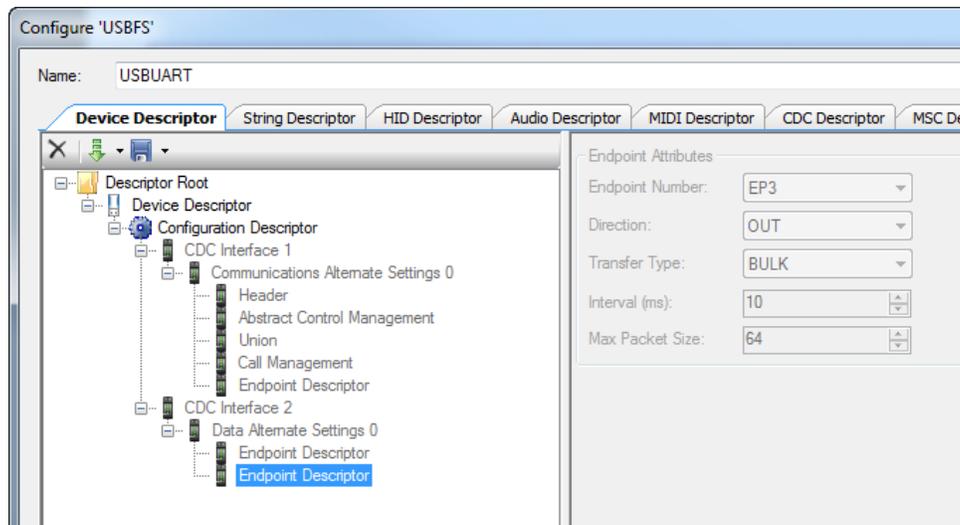


Figure 10. USBUART OUT Data Endpoint 3 Descriptor



Project Description

This project enumerates on the computer as a Virtual Com port. The project receives data from the COM-port terminal then sends the received data back. The LCD shows the line settings for PSoC3/5LP only. PSoC4200L kit (CY8CKIT-046) does not have an LCD.

Example Project Execution Flow

To execute the USBFS component code example you need the following equipment:

- PSoC Kit (CY8CKIT-046/001/030/050)
- Software COM port console

Follow the procedure below:

1. Configure the development kit to operate as bus-powered (the power is supplied from the USB connector VBUS pad). The power scheme supposes that device operates from 5V.

Note CY8CKIT-046 is only bus-powered.

2. Connect the PSoC kit to the PC through the USB connector for programming.

Note CY8CKIT-001 needs a separate MiniProg3 for programming.

3. Build the project and program into the device. At this point, you can disconnect the USB cable from the programming connector.
4. Connect the computer USB cable to the development kit.
5. Install drivers for the device manually. In Windows 7, open the Device Manager, find the USBUART device in the **Other devices** branch of the tree. Open the context menu and select Update Driver Software. Browse to the USBFS_UART.inf file from the project root directory as a driver.

Note The USBFS_UART.inf supports only devices with the code example default VID\PID. If the VID\PID values are changed, you can use the generated USBUART_cdc.inf file from the \Generated_Source\ directory. However, this file is not digitally signed and you should disable the check of the driver signature in Windows to use it. (See the USBFS Bulk Wraparound code example datasheet for details)

6. Open the Device Manager and note the COM port number for the Cypress USB UART device.
7. Run any COM-port terminal (for example, Putty) and make a new connection to the noted COM port.
8. Type the message in the COM-port terminal.

Expected Results

You should see an echo response of the previously sent message in the COM-port terminal. In the case of PSoC3/5LP you should see such information on the LCD (Figure 11):

- Baud Rate
- Data Bits



- Parity: N(None)/O(Odd)/E(Even)/M(Mark)/S(Space)
- Stop Bits: 1/1.5/2
- State of DTR/RTS signals: ON/OFF

The output format is: BR:115200 7E1 where

BR:115200 – the baud rate

7 – the data bits

E – the parity

1 – the stop bits

Figure 11. PSoC3 CY8CKIT-030 Kit



© Cypress Semiconductor Corporation, 2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, 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 products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

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.

Use may be limited by and subject to the applicable Cypress software license agreement.

