Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

www.infineon.com

CE223368 – PSoC 6 MCU: USB CDC Echo Application

Objective

This example demonstrates how to configure the USB block in a PSoC® 6 MCU as a Communication Device Class (CDC). When configured as a CDC the PSoC 6 USB enumerates a Virtual COM port, which can be read and written by a terminal emulator program on a PC.

Requirements

Tool: PSoC Creator™ 4.2, Peripheral Driver Library (PDL) 3.1.0

Programming Language: C

Associated Parts: All PSoC 6 MCU parts with USB

Related Hardware: PSoC 6 Wi-Fi-BT Pioneer Kit

Overview

In this project, PSoC 6 MCU USB block is configured for CDC. This allows it to enumerate on the computer as a Virtual COM port. The project receives data from the terminal emulator then sends the received data back.

Right-click the USBFS Component in the PSoC Creator schematic of this project. Select **Launch USB Configurator** to create the USB descriptor associated with the PSoC device. In this example, the descriptor contains the CDC interface descriptor.

Hardware Setup

This example uses the kit's default configuration. Refer to the kit guide to ensure that the kit is configured correctly.

Software Setup

Install a terminal emulator like Tera Term or PuTTY on your computer. The example uses the terminal window to send and display messages.

Operation

1. Connect the kit board to your PC using the provided USB cable through the USB connector [J10 for PSoC 6 WiFi-BT Pioneer kit].
2. Build the project and program it into the PSoC 6 MCU device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help.
3. Connect another USB cable (or reuse the same cable used to program the kit) to the USB device connector [J28 for PSoC 6 WiFi-BT Pioneer kit].
4. On the PC verify that a new USB device was enumerated as a Virtual COM port.

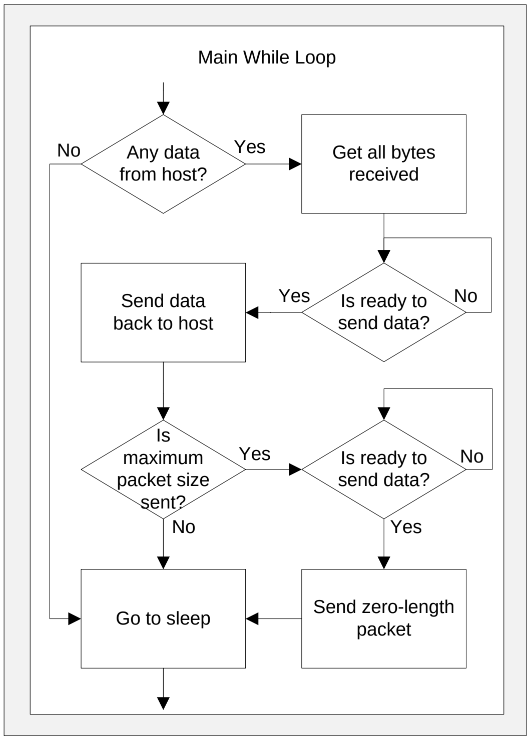
   Note: Windows 7 requires that you manually install the USB driver for this code example. Open Device Manager, find this USB device in the **Other devices** branch of the tree. Right-click this device and select **Update Driver Software**. Browse to the INF file from the project root as the driver to be installed.
5. Run the terminal emulator program and make a new connection to the Virtual COM port.
6. Type any message in the terminal and observe each key echoing back to the window.

Design and Implementation

In the main firmware routine, the USBFS block is configured to use the Communication Device Class (CDC). After enumeration, the device constantly checks if any data was received from the host. If any data is available, the application copies the received data to a buffer in the SRAM and use the same data to send back to the host.

Figure 1 shows the firmware flowchart of this code example.

Figure 1. Firmware Flowchart



If the device is not receiving or sending data, it goes to sleep and only wakes up at the next USB interrupt - which occurs every 1 millisecond. If the last sent packet is exactly the maximum packet size, it is followed by a zero-length packet to ensure that the end of the segment is properly identified by the terminal.

Components and Settings

Table 1 lists the PSoC Creator Components used in this example, how they are used in the design, and the non-default settings required so they function as intended.

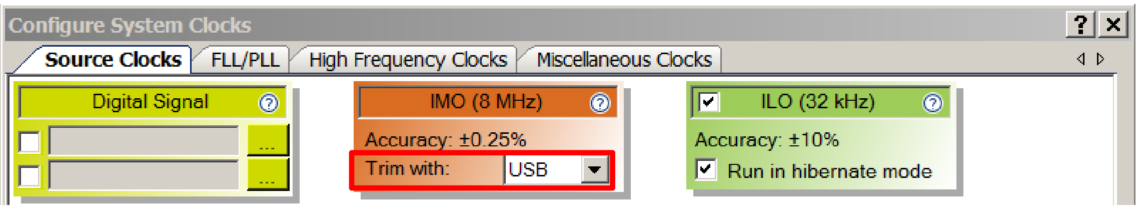
Table 1. PSoC Creator Components

Component	Instance Name	Purpose	Non-default Settings
USBFS	USBUART	Implements the CDC Device Class	All default

To visualize the USBFS descriptor, right-click the USBFS Component and select **Launch USB Configurator**. You can also refer to the *USBUART_cfg.h* file in the *Generated_Source* folder.

To achieve the 0.25% accuracy required by the USB bus, the IMO needs to be trimmed with USB, as shown in Figure 2. The IMO is configured in the Design Wide Resources clock tab.

Figure 2. Source Clocks Configuration



IMO (8 MHz)

Accuracy: ±0.25%

Trim with: USB

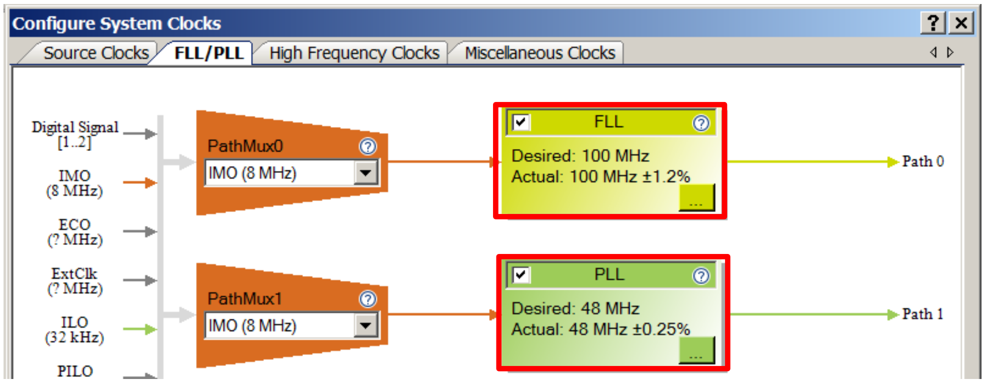
☑ ILO (32 kHz)

Accuracy: ±10%

☑ Run in hibernate mode

Figure 3 shows PLL/FLL configuration in the Design Wide Resources clock tab. Note that FLL and PLL are used in this application. FLL is used to clock the CPUs while PLL is used to clock the USBFS block. The reason to use the PLL to drive the USB is because it achieves the 0.25% accuracy requirement of the USB bus.

Figure 3. FLL/PLL Configuration



PathMux0: IMO (8 MHz)

PathMux1: IMO (8 MHz)

☑ FLL

Desired: 100 MHz

Actual: 100 MHz ±1.2%

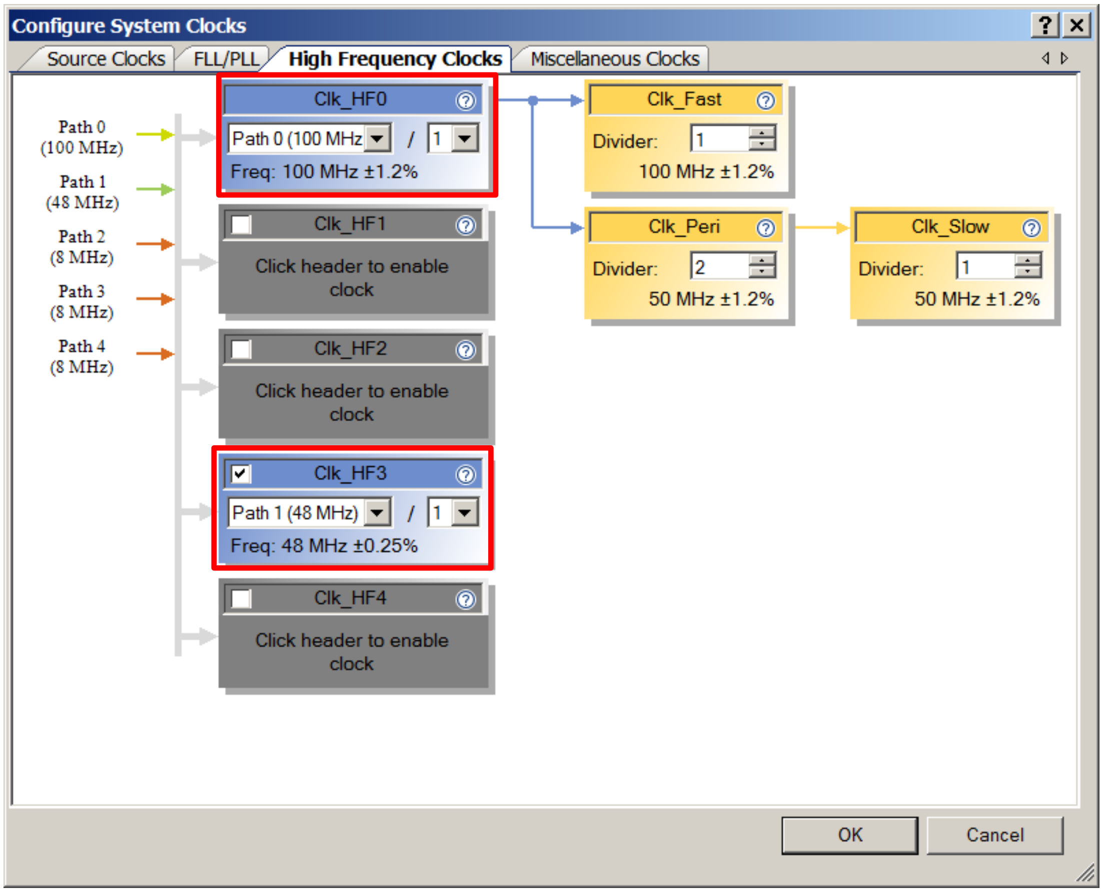
☑ PLL

Desired: 48 MHz

Actual: 48 MHz ±0.25%

Figure 4 shows the high-frequency clock settings.

Figure 4. High Frequency Clock Configuration



Clk_HF0: Path 0 (100 MHz / 1
Freq: 100 MHz ±1.2%

☑ Clk_HF3: Path 1 (48 MHz) / 1
Freq: 48 MHz ±0.25%

Clk_Fast: Divider: 1
100 MHz ±1.2%

Clk_Peri: Divider: 2
50 MHz ±1.2%

Clk_Slow: Divider: 1
50 MHz ±1.2%

For information on the hardware resources used by a Component, see the Component datasheet.

Reusing This Example

This example is designed for the CY8CKIT-062-WiFi-BT pioneer kit. To port the design to a different PSoC 6 MCU device and/or kit, change the target device using the Device Selector and update the pin assignments in the Design Wide Resources Pins settings as needed.

In some cases, a resource used by a code example is not supported on another device. In that case, the example will not work. If you build the code targeted at such a device, you will get errors. See the device datasheet for information on what a particular device supports.

Related Documents

Application Notes	
AN221774 – Getting Started with PSoC 6 MCU	Describes the PSoC 6 MCU devices and how to build your first PSoC project.
AN215656 – PSoC 6 MCU Dual-CPU System Design	Describes the dual-CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-CPU design.
Code Examples	
Visit the Cypress Code Example site for a comprehensive collection of code examples using PSoC Creator IDE.	
Device Documentation	
PSoC 6 MCU Datasheets	PSoC 6 MCU Technical Reference Manuals
Development Kit Documentation	
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit	

Document History

Document Title: CE223368 – PSoC 6 MCU: USB CDC Echo Application

Document Number: 002-23368

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6418191	RLOS	12/21/2018	New code example

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.