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.

CE226423 – PSoC 6 MCU: USB HID Mouse Application

Objective

This example demonstrates how to configure the USB block in a PSoC® 6 MCU as a Human Interface Device (HID). The device enumerates as a 3-button mouse.

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

This code example demonstrates how to setup a USB HID descriptor to implement a 3-button mouse. After the device enumerates, firmware moves the mouse cursor from your computer from the right to the left, and vice-versa.

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 3-button mouse HID descriptor.

Hardware Setup

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

Operation

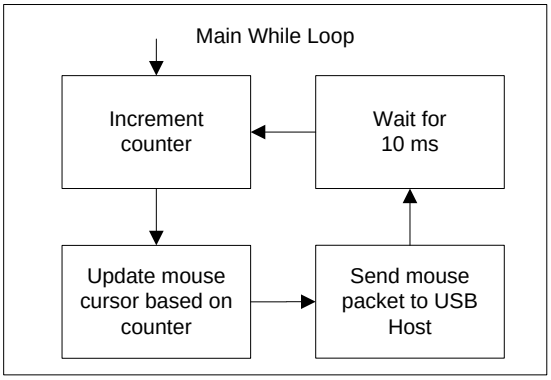
1. Connect the kit board to your PC using the provided USB cable through the USB connector [J10 for PSoC 6 WiFi-BT Pioneer kit].
2. Build the project and program it into the PSoC 6 MCU device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help.
3. Connect 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 mouse device. The mouse's cursor shall move from left to right, and vice-versa.

Design and Implementation

In the main firmware routine, the USBFS block is configured to use the HID Device Class. After enumeration, the device sends a packet to the host every 10 milliseconds. Each packet contains three bytes. The second and third bytes define the Y and X increments to move the mouse cursor. The first byte defines the mouse button states. In this example, only the second byte is used to move the cursor horizontally.

Figure 1 shows the firmware flowchart of this code example.

Figure 1. Firmware Flowchart



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	USBFS	Implements the HID Device Class	All default

To see the USBFS descriptor, right-click the USBFS Component and select **Launch USB Configurator**. You can also refer to the *USBFS_cfg.h* file in the *Generated_Source* folder. For more details on how to create a HID descriptor, read the USB Configurator Guide.

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

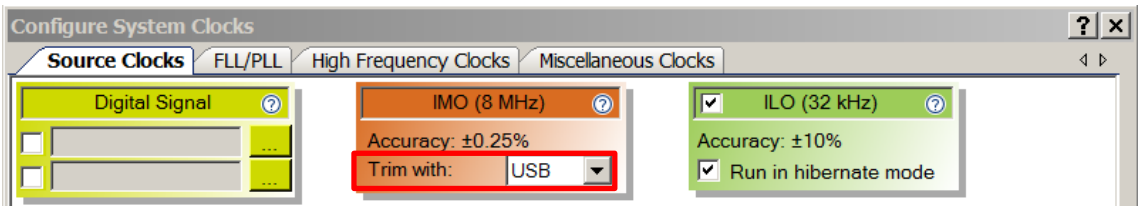


Figure 3 shows PLL/FLL configuration in the Design Wide Resources clock tab. Note that FLL and PLL are used in this application. FLL is used to clock the CPUs while PLL is used to clock the USBFS block. The reason to use the PLL to drive the USB is because it achieves the 0.25% accuracy requirement of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

Figure 3. FLL/PLL Configuration

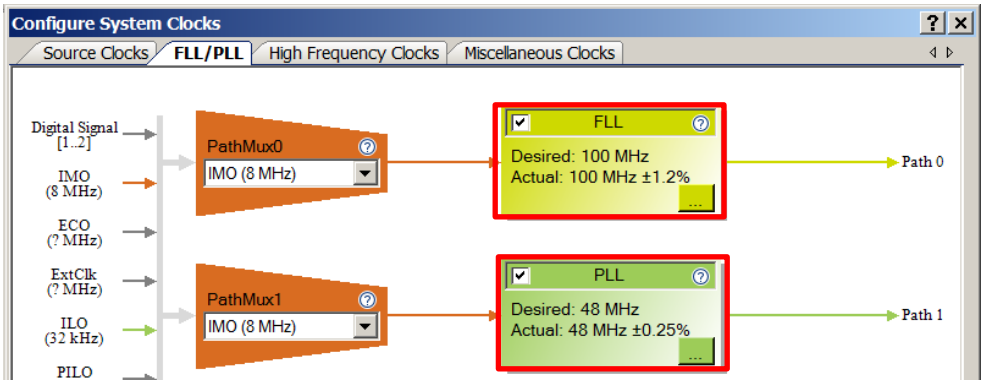
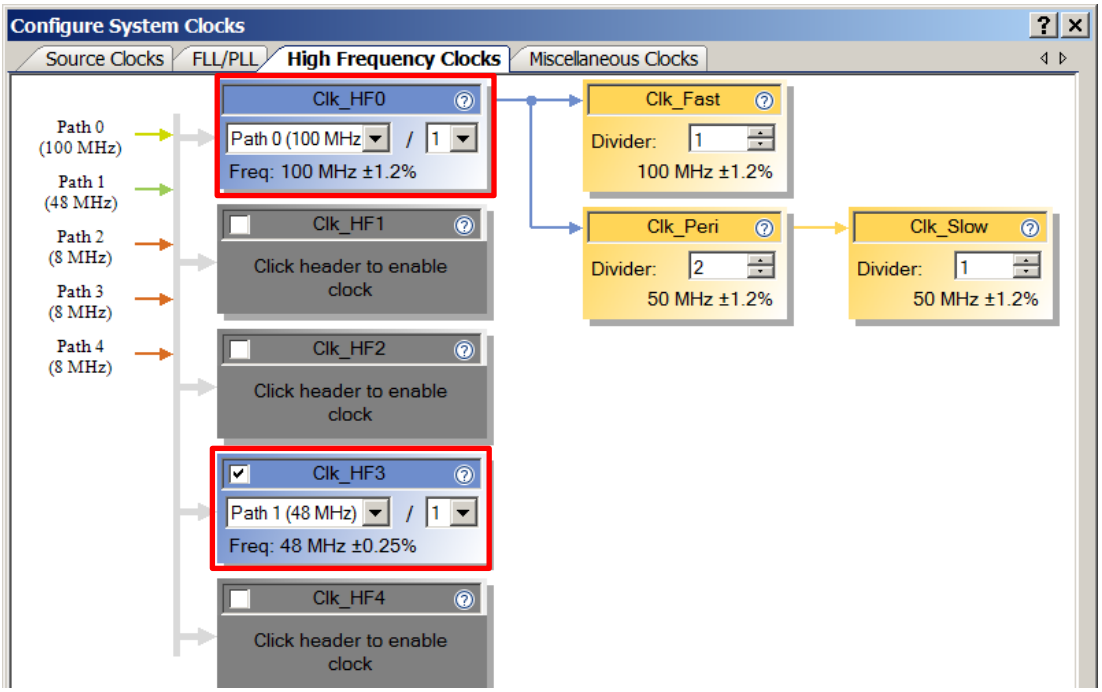


Figure 4 shows the high-frequency clock settings.

Figure 4. High Frequency Clock Configuration



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: CE226423 – PSoC 6 MCU: USB HID Mouse Application

Document Number: 002-26423

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6472562	RLOS	02/15/2019	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 | Code Examples | 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

© Cypress Semiconductor Corporation, 2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

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 shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. 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. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.