CE95391 - PSoC® USB HID Bootloader

Objective

This example demonstrates use of the PSoC Creator Bootloader and Bootloadable Components in a USB-based bootloader system. It also demonstrates a PC-based bootloader host program.

Overview

This example uses two PSoC Creator projects – bootloader and bootloadable – to demonstrate the PSoC Creator bootloader system. The bootloader project communicates with a PC host via USB HID, to program an application image. The bootloadable project is the application image that is programmed.

Requirements

Tool: PSoC Creator 4.0 or higher

Programming Language: C (GCC 4.9)

Associated Parts: All PSoC 3, PSoC 4 L-series, and PSoC 5LP parts with USB

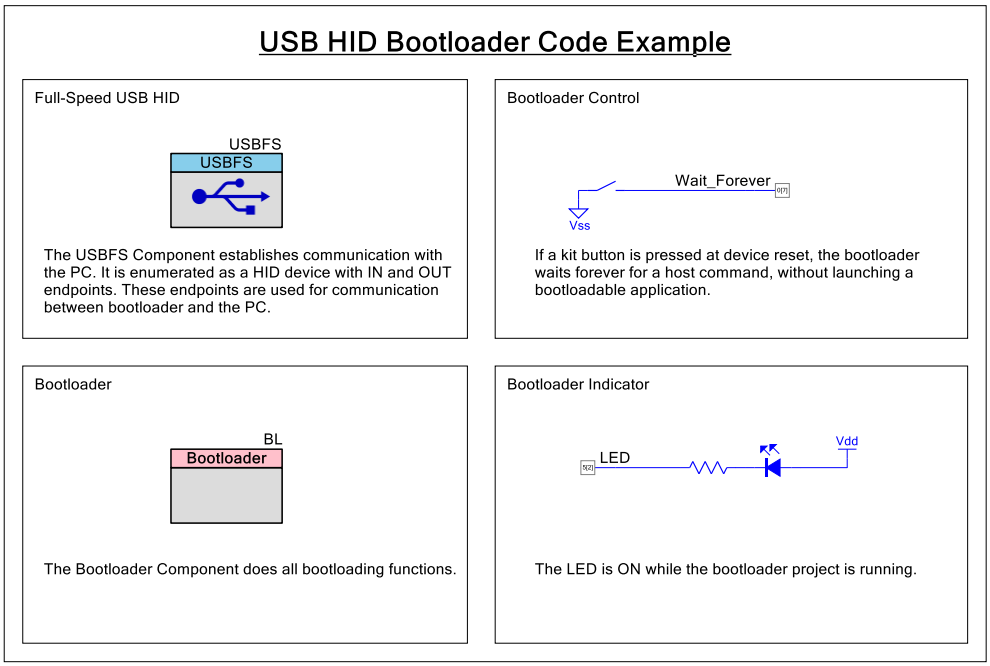
Related Hardware: CY8CKIT-030, CY8CKIT-046, CY8CKIT-050, CY8CKIT-059

Design

USB HID Bootloader

The bootloader project design features the PSoC Creator Bootloader and USBFS Components, as Figure 1 shows. The USBFS Component communicates with the PC host to get commands and a new application image. The Bootloader Component does flash programming, host command / response protocol, and launches the bootloadable application.

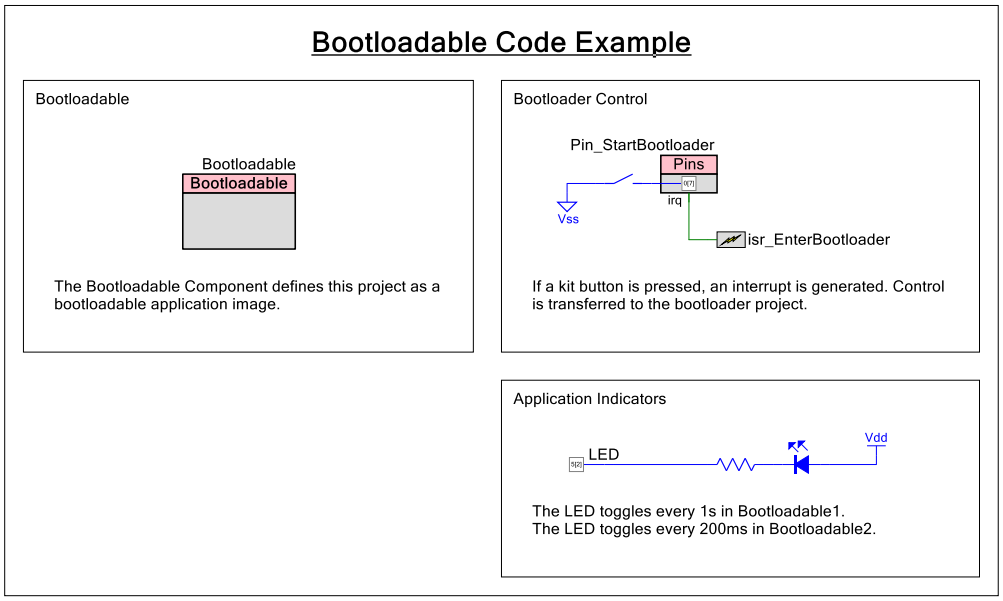
Figure 1. USB HID Bootloader Schematic (Pins and LED are Selected for CY8CKIT-046)



Bootloadable

The design of the bootloadable projects features the PSoC Creator Bootloadable Component, as Figure 2 shows. Other Components implement the desired application. This code example has two bootloadable projects, which perform slightly different functions.

Figure 2. Bootloadable Schematic (Pins and LED are Selected for CY8CKIT-046)



Design Considerations

This design can be extended by changing the Bootloader Component to implement a dual-application function. Then, two bootloadable projects can be programmed into flash.

This code example is designed for the following PSoC device families and associated kits:

- The PSoC 4 L-series family and associated CY8CKIT-046 kit
- The PSoC 5LP device family and associated CY8CKIT-059 kit

The design is easily portable to other PSoC 3, PSoC 4 L-series, and PSoC 5LP devices and kits, typically by just changing the LED or button pin assignments.

The PSoC Creator bootloader system is designed to be used with most communication channels, and it is easy to change this example to be an I^2C, UART, or SPI-based bootloader. For more information, see AN73854 - PSoC(R) 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders.

PSoC Creator Components

Table 1 and Table 2 list the PSoC Creator Components used in this example, as well as the hardware resources used by each Component.

Table 1. List of PSoC Creator Components for USB HID Bootloader Project

Component	Hardware Resources
BL	none
USBFS	1 USBFS
Wait_Forever	1 pin
LED	1 pin

Table 2. List of PSoC Creator Components for Bootloadable Projects

Component	Hardware Resources
Bootloadable	none
Pin_StartBootloader	1 pin
isr_EnterBootloader	1 interrupt vector
LED	1 pin

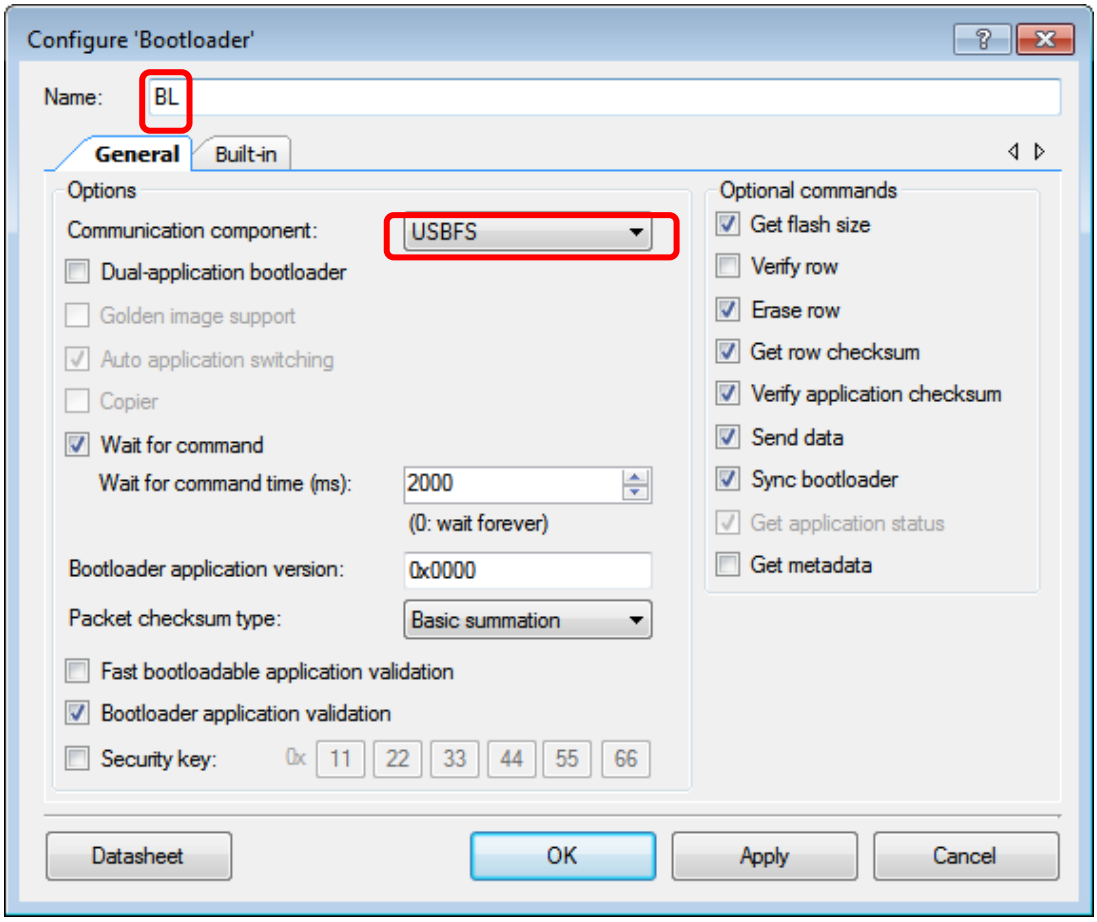
Parameter Settings

This section shows the changed configuration settings for the PSoC Creator Components in the code example projects.

Bootloader

Figure 3 shows the changed settings for the Bootloader Component. The Component Name change is optional.

Figure 3. Bootloader Component Configuration



USBFS

The USBFS Component is configured for HID bootloader by using an import file provided with PSoC Creator:

```
<PSoC Creator InstallDir> \ psoc \ content \ cycomponentlibrary \ CyComponentLibrary.cylib \
USBFS_v_x_xx \ Custom \ template \ Bootloader.root.xml
```

For details on how to import this file, see application note AN73053, PSoC USB HID Bootloader.

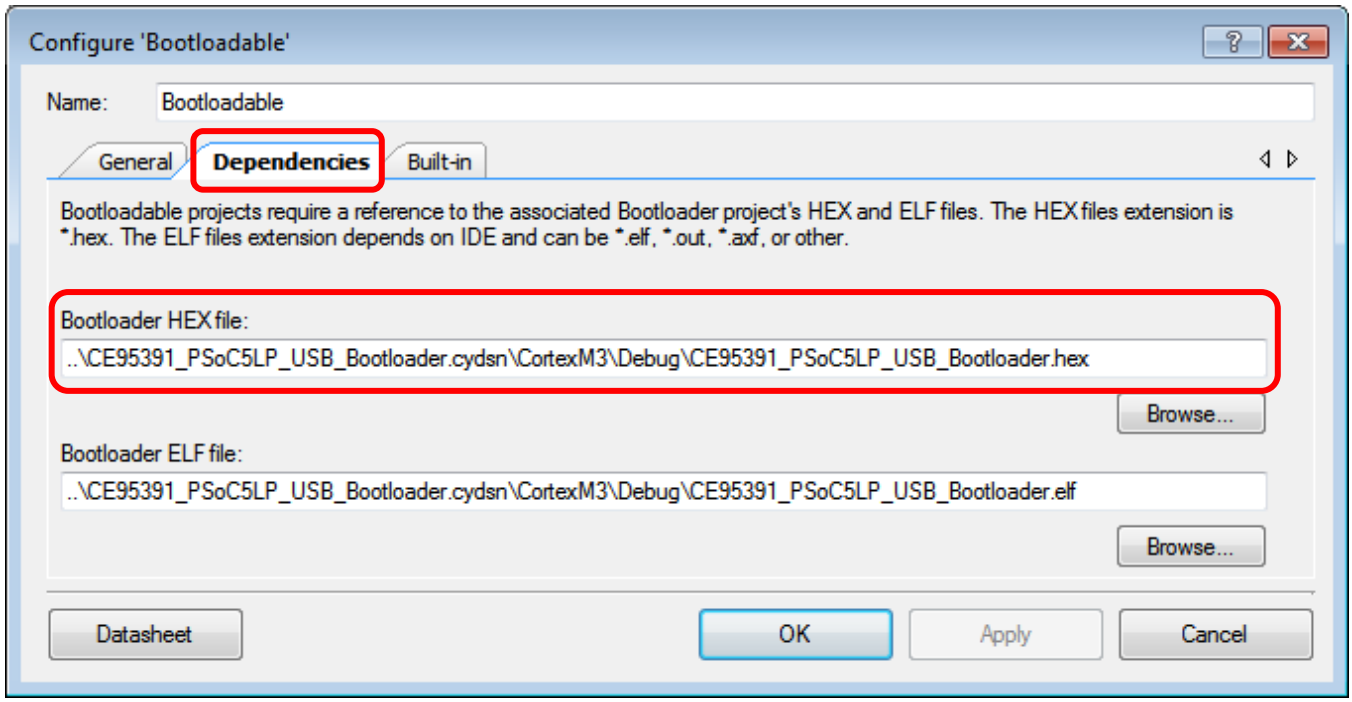
Bootloadable

A bootloadable project is always linked to the output *.hex* and *.elf* files of an associated bootloader project, as Figure 4 shows. Selecting the *.hex* file automatically selects the associated *.elf* file. Before configuring a bootloadable Component, you should completely build the associated bootloader project.

For this code example, the bootloader *.hex* file is at a relative path within this PSoC Creator workspace. However, this need not be the case; bootloader and bootloadable projects can be in different workspaces.

For more information on bootloader and bootloadable files, see AN73854, PSoC(R) 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders.

Figure 4. Bootloadable Component Configuration



Input Pins

In all of the projects, digital input Pin Components are used to read the state of a kit button. The buttons short to ground when pressed, therefore the Pin Component must be configured for resistive pull up, as Figure 5 shows. This causes the Pin input state to be 0 when the button is pressed and 1 when it is released.

Figure 5. Digital Input Pin Configuration for Resistive Pull Up

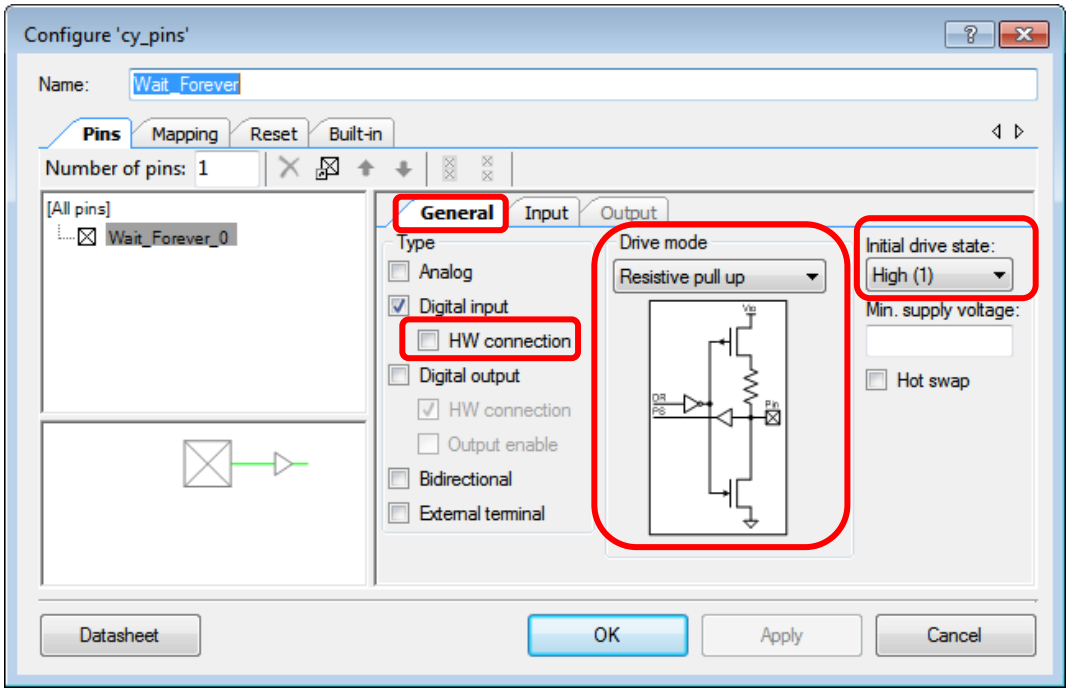
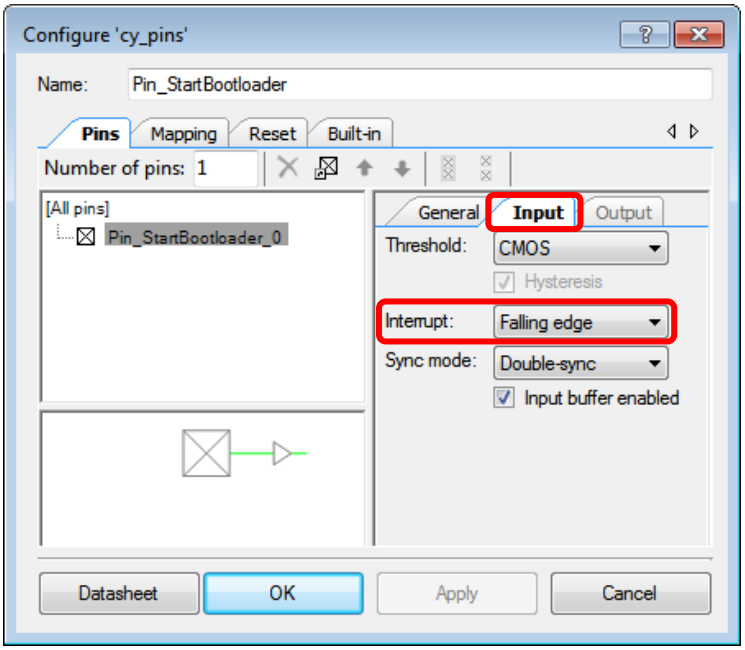


Figure 6 shows the configuration for the input Pin Component to generate an interrupt.

Figure 6. Digital Input Pin Configuration for Interrupt



Design-Wide

The design-wide clocks are configured for USB, as Figure 7 shows for the PSoC 4 L-series, and Figure 8 shows for PSoC 3 and PSoC 5LP.

Figure 7. PSoC 4 L-series USB Clock Configuration

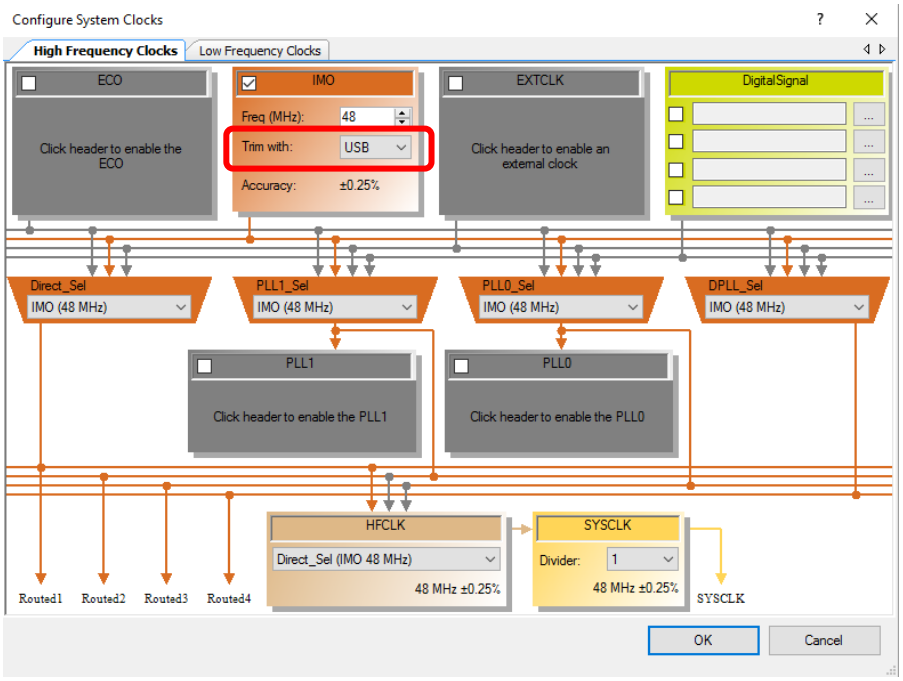
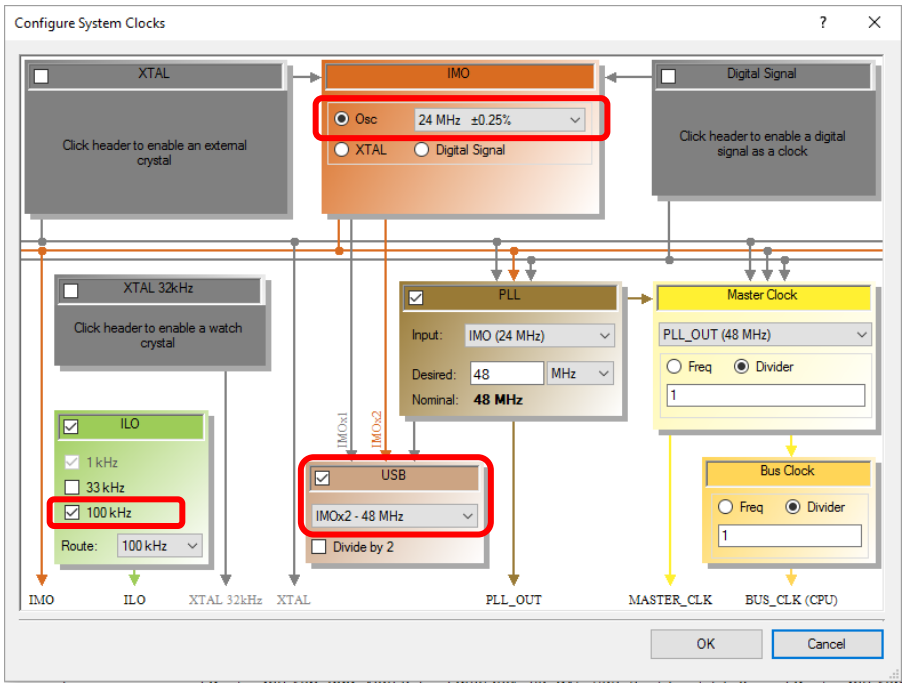


Figure 8. PSoC 3 and PSoC 5LP USB Clock Configuration



Operation

1. Connect a USB cable between the host PC and the kit *programming USB connector*. Note that some kits plug directly into the USB port.
2. Program the USB HID Bootloader project into the kit. Confirm that the kit LED is on constantly, indicating that the bootloader is running.
3. Unplug the USB cable, and plug it into the kit *general usage USB connector*. If the kit is plugged into a USB port, remove it. The kit should be able to be powered from the USB connection.
4. Run the PSoC Creator Bootloader Host program (menu item **Tools** > **Bootloader Host...**).
5. Select a Bootloadable1 project *.cyacd* file and program it using the bootloader. Confirm that the LED blinks at a 1-second rate, indicating that the bootloadable is running.
6. Press the kit button to switch from bootloadable to bootloader. Confirm that the LED stays on (the bootloader is running).
7. Select a bootloadable2 project *.cyacd* file and program it using the bootloader. Confirm that the LED now blinks at a 200-ms rate, indicating that the other bootloadable is running.
8. Press the kit button to switch from bootloadable to bootloader. Confirm that the LED stays on (the bootloader is running).

The remaining steps are optional:

9. Start programming a bootloadable project *.cyacd* file, as in steps 5 and 7, but during programming disconnect the USB cable.
10. Repeat step 5 or 7 and confirm the successful completion of bootloading.

Related Documents

For more information, refer to the documents listed in Table 3.

Table 3. Related Documents

Application Notes		
AN73854	PSoC 3, PSoC 4, and PSoC 5LP Introduction to Bootloaders	Describes the bootloader architecture used in PSoC 3, PSoC 4 and PSoC 5LP
AN73503	PSoC 3 and PSoC 5LP USB HID Bootloader	Shows how to build a USB HID -based bootloader for PSoC 3 and PSoC 5LP
AN60317	PSoC 3 and PSoC 5LP I^2C Bootloader	Shows how to build an I^2C-based bootloader for PSoC 3 and PSoC 5LP
AN89611	PSoC 3 and PSoC 5LP Getting Started with Chip-Scale Packages	Describes the I2C bootloader that is factory-installed in PSoC 3 and PSoC 5LP CSP devices
AN86526	PSoC 4 I^2C Bootloader	Shows how to build an I^2C-based bootloader for PSoC 4 family devices
AN68272	PSoC 3 and PSoC 5LP UART Bootloader	Shows how to build a UART-based bootloader for PSoC 3 and PSoC 5LP
AN84401	PSoC 3 and PSoC 5LP SPI Bootloader	Shows how to build a SPI-based bootloader for PSoC 3 and PSoC 5LP
AN2100	PSoC 1 Bootloader	Describes a UART-based bootloader for PSoC 1
Code Examples		
CE69310	PSoC 1 I^2C Bootloader	
CE82634	Host Code for Bootloading PSoC 1 Via I^2C	
PSoC Creator Component Datasheets		
Bootloader and Bootloadable	Describes the bootloader and bootloadable components required for implementing a bootloader system	
Full Speed USB (USBFS)	Provides a USB full-speed compliant device framework	

Device Documentation	
PSoC 3 Datasheets	PSoC 3 Technical Reference Manuals
PSoC 4L Datasheets	PSoC 4L Technical Reference Manuals
PSoC 5LP Datasheets	PSoC 5LP Technical Reference Manuals
Development Kit (DVK) Documentation	
PSoC 3 and PSoC 5LP Kits	
PSoC 4 Kits	

Document History

Document Title: CE95391 - PSoC® USB HID Bootloader

Document Number: 001-95391

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5093124	SJLE	01/19/2016	New code example
*A	5591285	SJLE	01/20/2017	Updated Figure 4 for the updated location of .hex and .elf outputs.
*B	5740313	AESATP12	05/26/2017	Updated logo and copyright.

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

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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

.