

Bootloader and Bootloadable Example Project

1.0

Features

- Bootloader with I²C communication interface
- Switching between bootloader and bootloadable applications

General Description

This example project demonstrates the basic operation of the Bootloader and Bootloadable components.

Development Kit Configuration

This example project is designed to run on the CY8CKIT-042 from Cypress Semiconductor. A full description of the kit, along with more example programs and ordering information, can be found at <http://www.cypress.com/go/cy8ckit-042>.

The project requires the following configuration settings changes in order to run on the CY8CKIT-040 from Cypress Semiconductor. A full description of the kit, along with more example programs and ordering information, can be found at <http://www.cypress.com/go/cy8ckit-040>.

In order to switch from CY8CKIT-042 to CY8CKIT-040 following steps should be performed:

1. Change the project's device from CY8C4245AXI-483 to CY8C4014LQI-422 with a Device Selector called from the project's context menu.
2. Assign the Pin Components to physical pins. In the Workspace Explorer window, double-click the project's design-wide resource file and assign the pins as Table 1 and Table 2 shows.

Table 1. Pin Assignment of the Bootloader_PSoC4_Example project

Pin Name	Development Kit	
	CY8CKIT-042	CY8CKIT-040
I2C_Slave:scl	P3[0]	P1[2]
I2C_Slave:sda	P3[1]	P1[3]
Bootloader_Status	P0[3]	P0[2]

Table 2. Pin Assignment of the Bootloadable_PSoC4_Example project

Pin Name	Development Kit	
	CY8CKIT-042	CY8CKIT-040
Bootloadable_Status	P0[2]	P1[1]

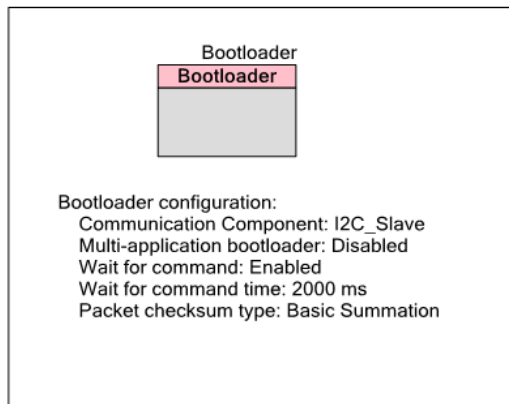
Bootloader Project Configuration

The example project consists of the Bootloader, SCB (in I2C Slave mode) and Pins components.

Bootloader

The Bootloader component allows you to update the device flash memory with a new code. The bootloader accepts and executes commands, then passes the command responses back to the communications component. The bootloader collects and arranges the received data and manages the actual writing of flash through a simple command/status register interface.

Figure 1. Bootloader

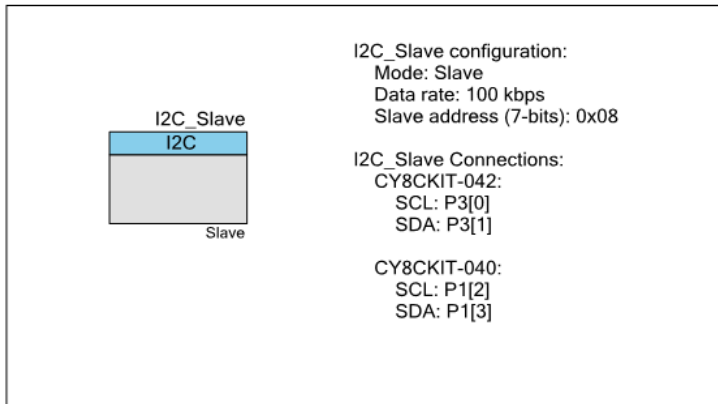


PSoC 4 SCB

The PSoC 4 SCB component is configured to implement I²C bus in Slave mode and manages the communications protocol to receive commands from an external system, and passes those commands to the bootloader. It also passes command responses from the bootloader back to the off-chip system.

The I²C connection depends on the development kit and is available in Table 1.

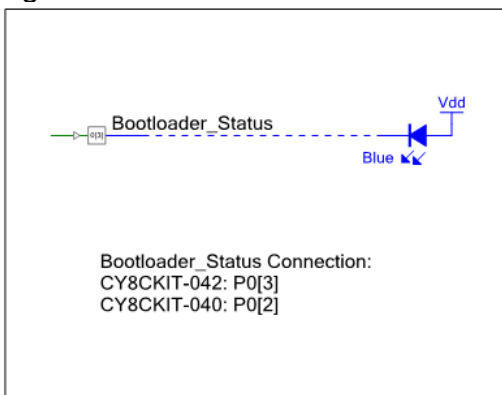
Figure 2. PSoC 4 SCB



Pins

The Pins component is used to indicate the bootloader application running status. The pin connection depends on the development kit and is available in Table 1.

Figure 3. Pins



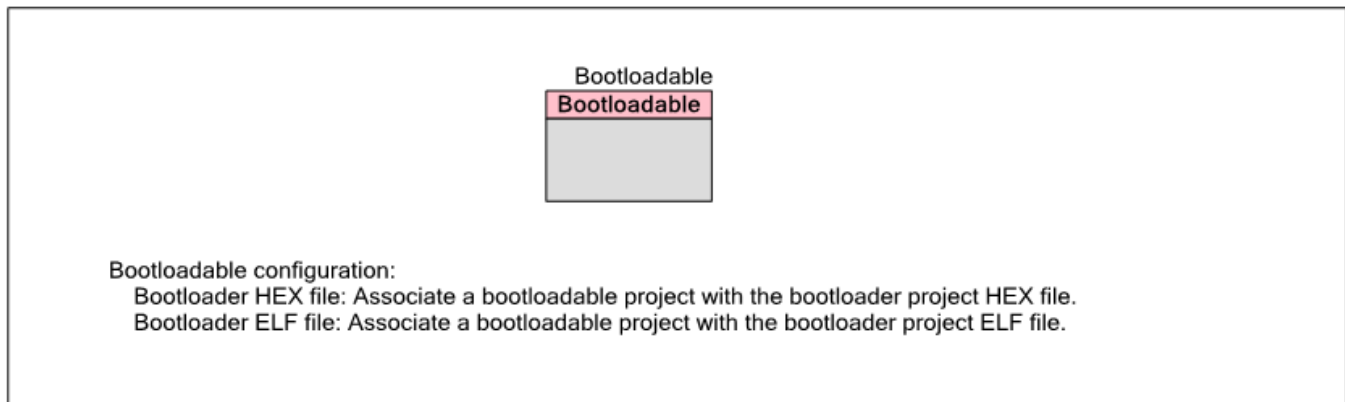
Bootloadable Project Configuration

The example project consists of the Bootloadable and Pin components.

Bootloadable

The Bootloadable component allows specifying additional parameters for the bootloadable project.

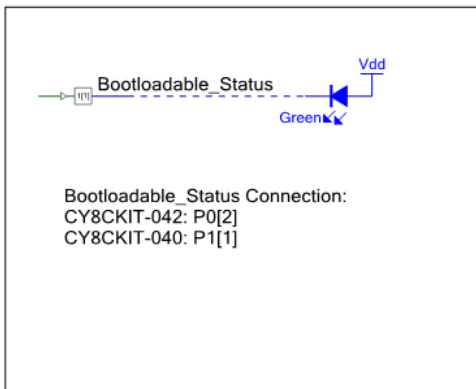
Figure 4. Bootloadable



Pins

The Pins component is used to indicate the bootloadable application running status. The pin connection depends on the development kit and is available in Table 2.

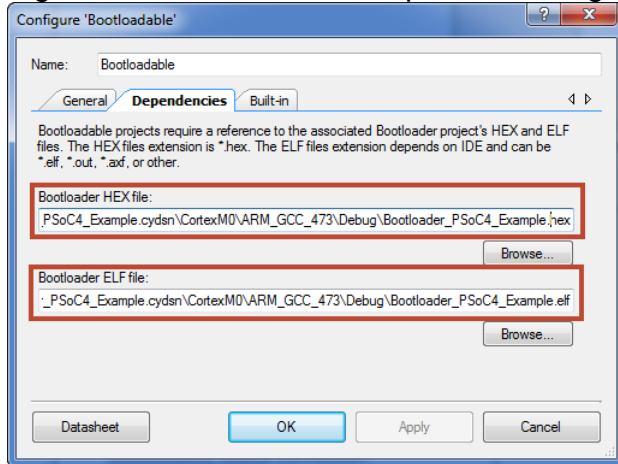
Figure 5. Pins



Projects Description

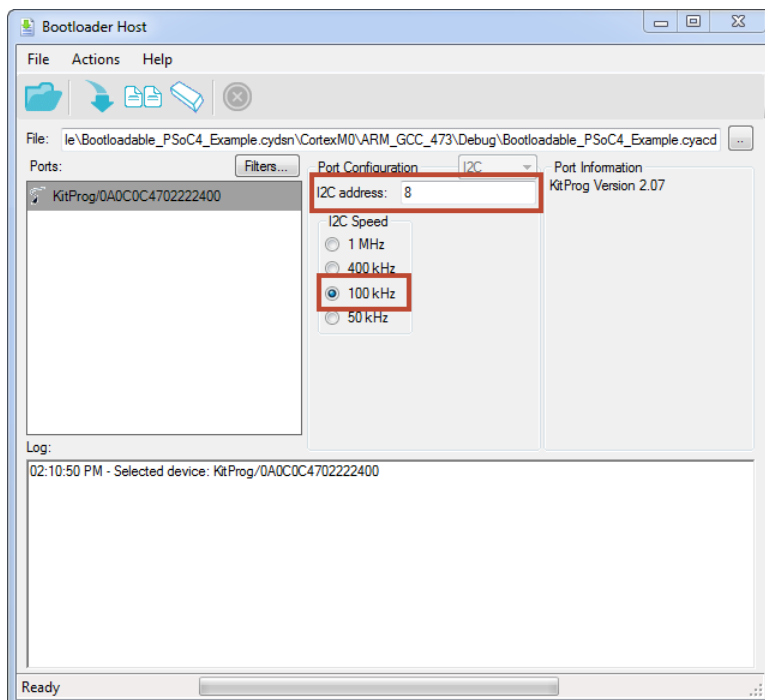
1. Build the *Bootloader_PSoC4_Example* project and program it into a device on the CY8CKIT-042 or CY8CKIT-040 kit.
2. Open the top design schematic of the *Bootloadable_PSoC4_Example* project. Specify path to the bootloader project HEX and ELF files, by double-clicking on the Bootloadable component and going to the **Dependencies** tab and link Bootloadable to the *Bootloader_PSoC4_Example.hex* file, as
3. Figure 6 shows.

Figure 6. Bootloadable Component Configuration



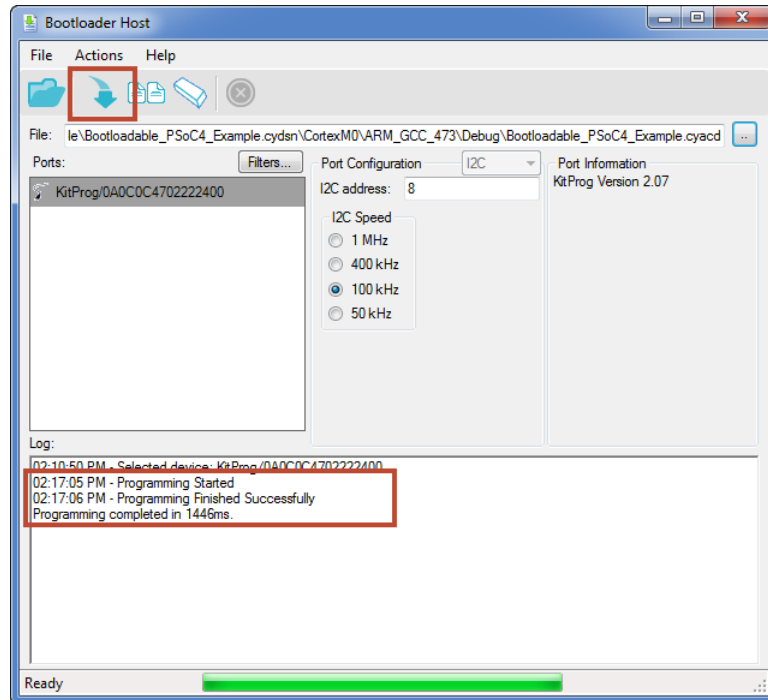
4. Open the Bootloader Host tool by navigating to Tools > Bootloader Host in PSoC Creator.
5. Make sure that the bootloader host application's I²C configuration, shown in Figure 7, is the same as the bootloader project's I2C Component configuration (Figure 2).

Figure 7. Bootloader Host Application



6. Press the File button and choose the bootloadable file Bootloadable_PSoC4_Example.cyacd in the bootloadable project's Debug/Release folder:
..\Bootloadable_PSoC4_Example.cydsn\CortexM0\ARM_GCC_473\
7. To bootload the device, click the **Program** button. You should get a screen similar to Figure 8.

Figure 8. Bootloading Bootloadable Application



8. After the bootloadable project is downloaded successfully, a software reset occurs, and the device starts executing the new application. The green LED turns on.

Expected Results

Program the device with the bootloader project. Transfer bootloadable application as described in the section above. Once started, the bootloadable application runs for 7 seconds and switches back to the bootloader application. While the bootloader application is running the blue LED turns on and while the bootloadable application is running the green LED turns on.



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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2014. 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.

This 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.

