

## Objective

This example demonstrates the use of I2C SCB (Serial Communication Block) resource for PSoC® 6 MCU in master mode. Three applications show the use of SDK APIs to communicate with I<sup>2</sup>C and EzI2C slave, using ModusToolbox™ IDE.

## Requirements

**Tool:** ModusToolbox™ IDE 1.1

**Programming Language:** C

**Associated Parts:** All PSoC 6 MCU parts

**Related Hardware:** PSoC 6 BLE Pioneer Kit, PSoC 6 WiFi-BT Pioneer Kit, PSoC 6 WiFi-Prototyping Kit

## Overview

The I<sup>2</sup>C master for PSoC 6 MCU is designed to send command packets to control the user LED on the slave. Three applications developed in this example are: I<sup>2</sup>C master using high-level APIs, I<sup>2</sup>C master using low-level APIs and I<sup>2</sup>C master communication with EzI2C slave.

## Hardware Setup

This example uses the PSoC 6 WiFi-BT Pioneer Kit's default configuration. Refer to the kit guide to ensure the kit is configured correctly. You can also use PSoC 6 BLE Pioneer Kit or PSoC 6 WiFi-BT Stamp Board Kit by importing the application for that kit.

**Note:** The PSoC 6 BLE Pioneer kit and the PSoC 6 WiFi-BT Pioneer kit ship with KitProg2. ModusToolbox only works with KitProg3. Before using this code example, make sure that the kit is upgraded to KitProg3. See ModusToolbox Help > ModusToolbox IDE Documentation > User Guide; section PSoC 6 MCU KitProg Firmware Loader. If you do not upgrade, you will see an error like "unable to find CMSIS-DAP device" or "KitProg firmware is out of date".

Jumper wires are used to establish connection between the master and slave on WiFi-BT Pioneer Kit. P6[0] is connected to P9[0] and P6[1] is connected to P9[1] on the board.

## Operation

1. Connect the Pioneer board to your PC using the provided USB cable through the USB connector.
2. Import the application into a new workspace. If you are unsure how to import an application, see [KBA225201](#).
3. Build the application. Choose **Project > Build All**.
4. Program the PSoC 6 MCU device. Select the **mainapp** project. In the **QuickPanel**, scroll down and click the **Program Kitprog3** item.
5. Observe the KIT\_LED2 blink with an interval of 1 second.

## Debugging

You can debug the example to step through the code. Use the **Debug (KitProg3)** configuration. See [KBA224621](#) to learn how to start a debug session with ModusToolbox IDE.

## Design and Implementation

In all three applications, the Arm Cortex®-M4 (CM4) MCU controls both the Master and Slave SCB. Different pins are configured for SCL and SDA for Master and Slave. Master sends command packets to control the user LED (KIT\_LED2).

Master APIs are divided into two categories: **Master High-Level** and **Master Low-Level**. See the SDK documentation to know more about high-level and low-level functions.

The SCB I2C resource is used in all three I2C Master applications. The Master sends different command packets to the Slave every one second. A command packet has the information to turn OFF or ON the user LED (KIT\_LED2).

## Resources

Table 1 lists the resources used in this example, and how they are used in the design.

Table 1. ModusToolbox Resources

Resource	Alias	Purpose	Non – Default Settings
SCB2, SCB3	mI2C, sI2C	Two SCB peripheral blocks	Figure 1, Figure 2
SCB2	sEzI2C	Single SCB peripheral block	Figure 3
GPIO	KIT_LED2	KIT_LED2	Figure 4

## Parameter Settings

Non-default settings for each Resource are outlined in red in the following figures.

Figure 1. I2C Master Resource Parameter Settings

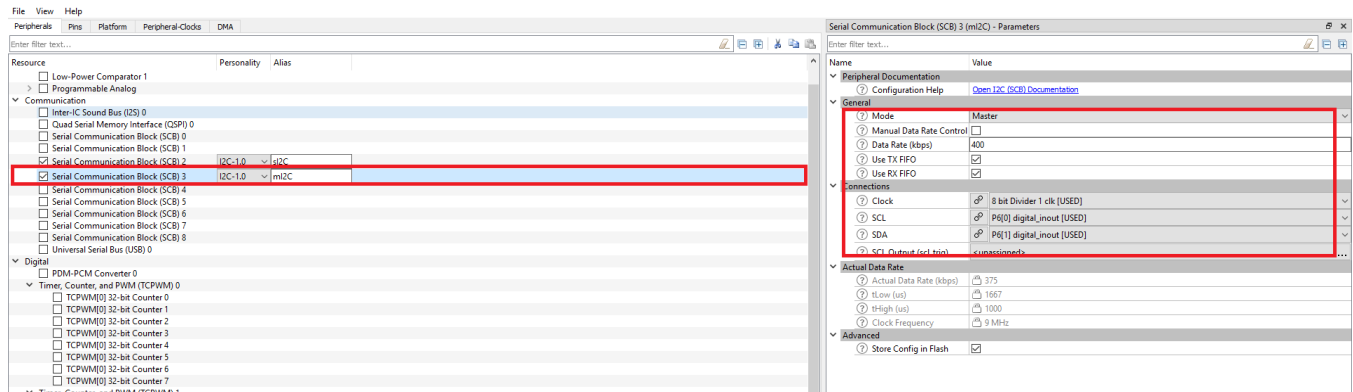


Figure 2. I2C Slave Resource Parameter Settings

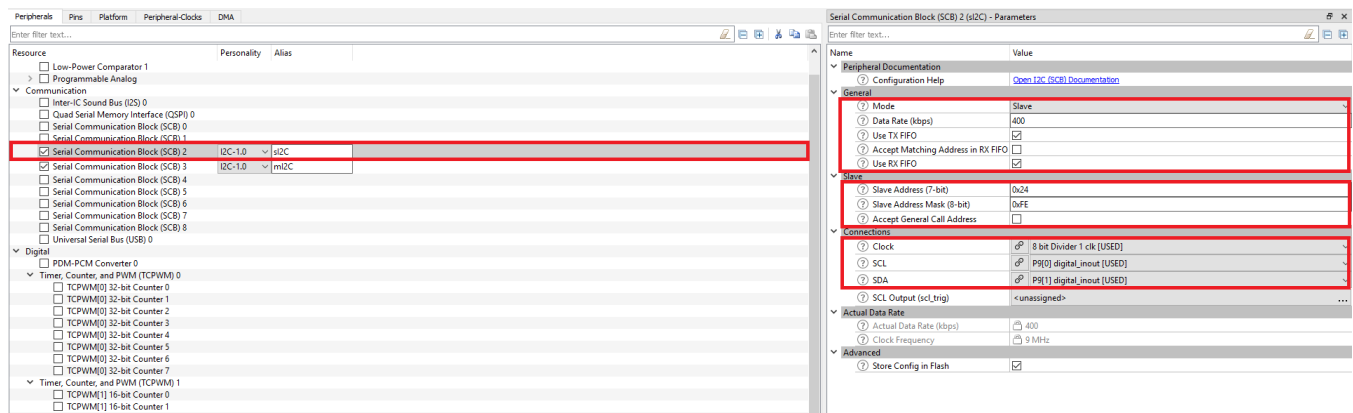


Figure 3. EzI2C Slave Resource Parameter Settings

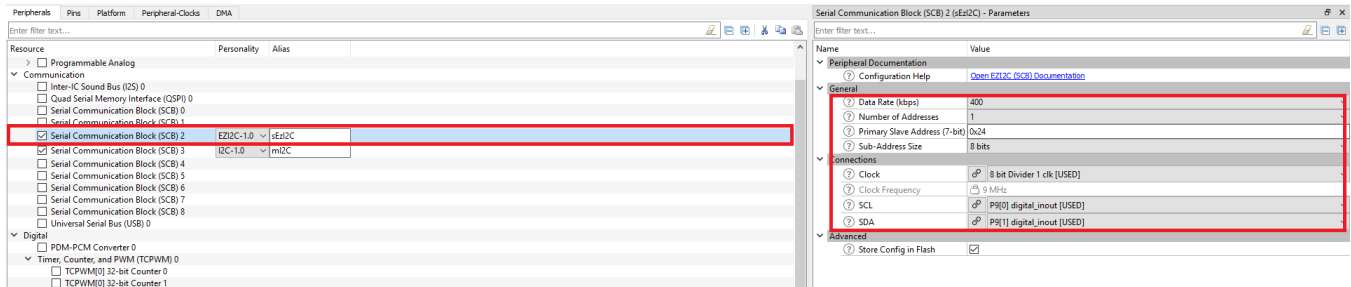
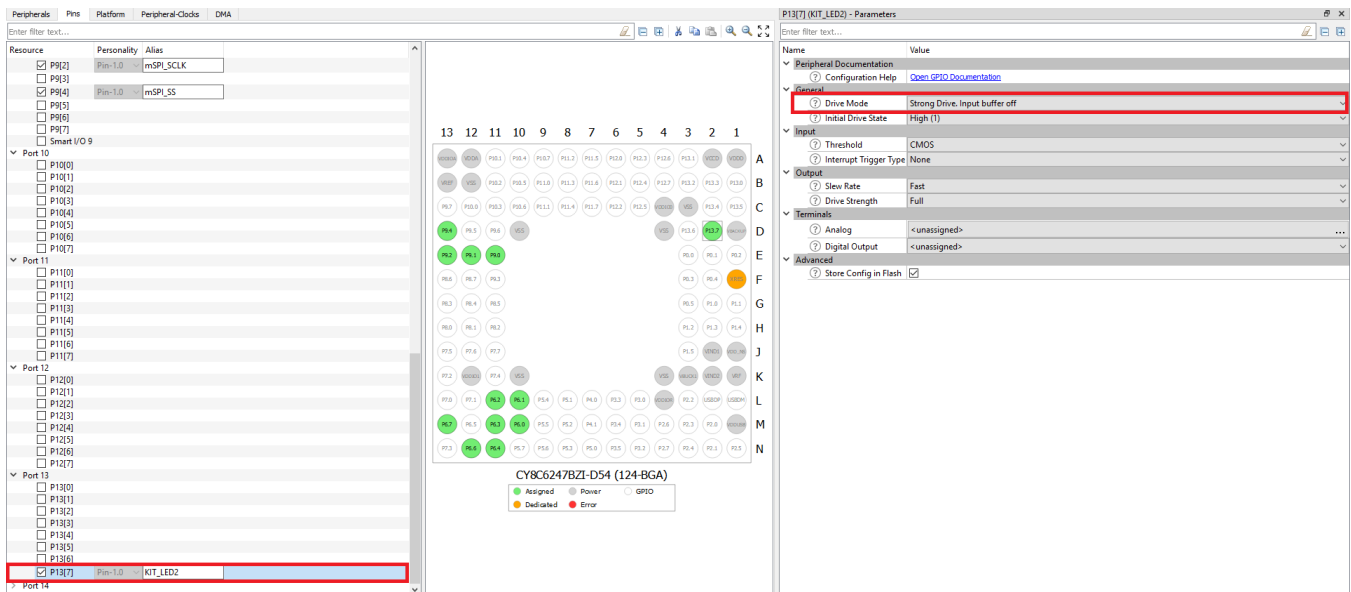


Figure 4. KIT\_LED2 Configuration



## Related Documents

For a comprehensive list of PSoC 6 MCU resources, see [KBA223067](#) in the Cypress community.

Application Notes	
<a href="#">AN221774</a> - Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first ModusToolbox application and PSoC Creator project.
<a href="#">AN210781</a> – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices.
<a href="#">AN215656</a> – 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	
<a href="#">CE218472</a> - PSoC 6 MCU Comparing External Voltages Using a Low-Power Comparator	
Visit the <a href="#">Cypress GitHub site</a> for a comprehensive collection of code examples using ModusToolbox IDE	
Device Documentation	
<a href="#">PSoC 6 MCU: PSoC 63 with BLE Datasheet</a>	<a href="#">PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual</a>
Development Kit Documentation	
<a href="#">CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit</a>	
<a href="#">CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit</a>	
<a href="#">CY8CPROTO-062-4343W PSoC 6 Wi-Fi BT Prototyping Kit</a>	
Tool Documentation	
<a href="#">ModusToolbox</a>	The Cypress IDE for IoT designers

## Cypress Resources

Cypress provides a wealth of data at [www.cypress.com](http://www.cypress.com) to help you to select the right device, and quickly and effectively integrate the device into your design.

For the PSoC 6 MCU devices, see [KBA223067](#) in the Cypress community for a comprehensive list of PSoC 6 MCU resources.

## Document History

Document Title: CE220818 – PSoC 6 MCU I2C Master

Document Number: 002-25535

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6369716	YEKT	11/2/2018	New Code Example
*A	6484274	YEKT	2/20/2019	Code example updated for ModusToolbox 1.1

## 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	<a href="http://cypress.com/arm">cypress.com/arm</a>
Automotive	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
Interface	<a href="http://cypress.com/interface">cypress.com/interface</a>
Internet of Things	<a href="http://cypress.com/iot">cypress.com/iot</a>
Memory	<a href="http://cypress.com/memory">cypress.com/memory</a>
Microcontrollers	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
Power Management ICs	<a href="http://cypress.com/pmic">cypress.com/pmic</a>
Touch Sensing	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB Controllers	<a href="http://cypress.com/usb">cypress.com/usb</a>
Wireless Connectivity	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

### PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

### Cypress Developer Community

[Community](#) | [Projects](#) | [Videos](#) | [Blogs](#) | [Training](#) | [Components](#)

### Technical Support

[cypress.com/support](http://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, 2018-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("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 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.

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](http://cypress.com). Other names and brands may be claimed as property of their respective owners.