CE220823 – PSoC 6 MCU SMIF Memory Write and Read Operation

Objective

This example demonstrates the write and read operations to the Serial Memory Interface (SMIF) in PSoC® 6 MCU.

Overview

This example writes and reads 256 bytes of data to external memory using SMIF quad mode. The example also checks the integrity of read data against written data.

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (Arm® GCC 5.4-2016-q2-update, Arm MDK 5.22)

Associated Parts: All PSoC 6 MCU parts

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Design

Figure 1 shows the design for this code example. The SMIF Component implements a SPI-based communication for interfacing external memory devices with PSoC. SMIF Component is configured with four data lines and single slave select line. The UART Component outputs debug information to a terminal window. It is configured for 8N1, transmit only, at 115.2 kbps. The design also uses two digital output pins to drive the RGB LED to indicate the status of data transfer.

Figure 1. SMIF Memory Write and Read Example Schematic

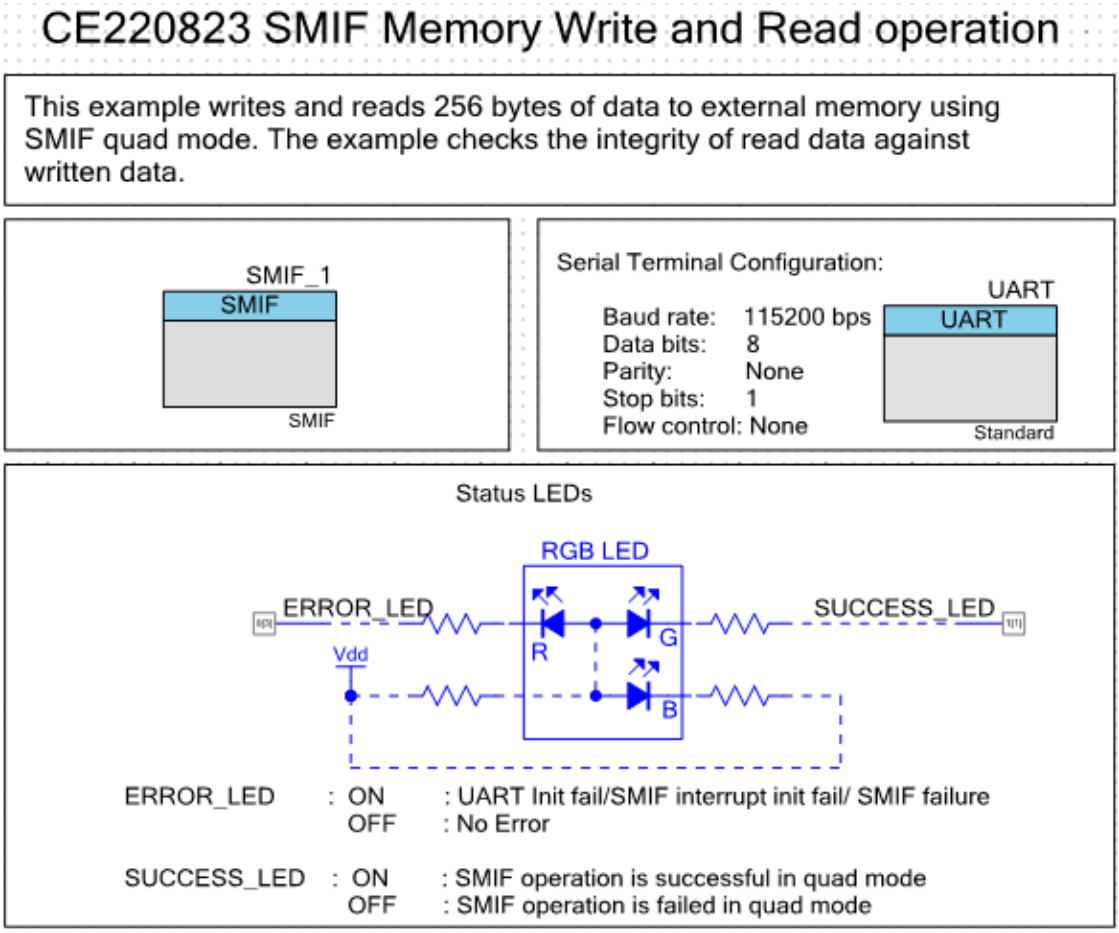


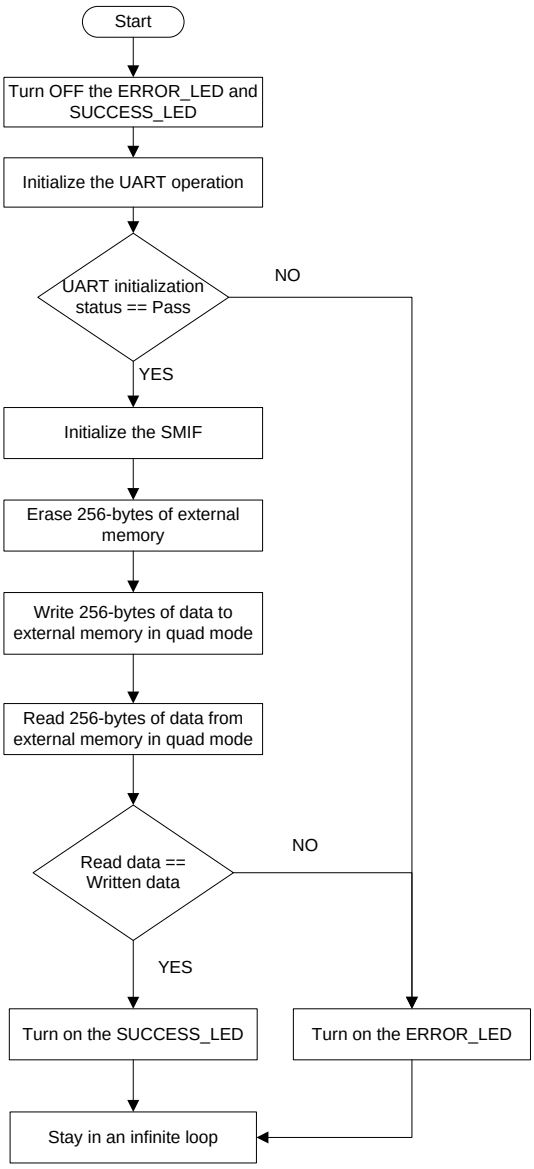
Table 1 explains the LED status indication.

Table 1. LED Status Indication

LED name	LED Status	Indication
ERROR_LED	ON	UART initialization failure or SMIF interrupt initialization failure or SMIF initialization failure or SMIF operation failure
	OFF	No Error
SUCCESS_LED	ON	SMIF operation succeeded in quad mode
	OFF	SMIF operation failed in quad mode

Figure 2 shows the firmware flow chart.

Figure 2. Firmware Flow Chart



The firmware uses source code (*cy_smif_memconfig.c* and *cy_smif_memconfig.h* files) generated from the SMIF Configuration Tool. This source code provides declarations for the SMIF driver memory configuration. For more information on the SMIF Configuration Tool, see Appendix A: SMIF Configuration Tool.

Design Considerations

This code example is designed to run on CY8CKIT-062-BLE with the PSoC 6 MCU device. To port the design to other PSoC 6 MCU family devices and kits, you must change the target device in Device Selector, and change the pin assignments in the ***cydwr*** settings. For single-core PSoC 6 MCU devices, port the code from *main_cm4.c* to *main.c* file as CM0+ CPU is not used in this code example.

Hardware Setup

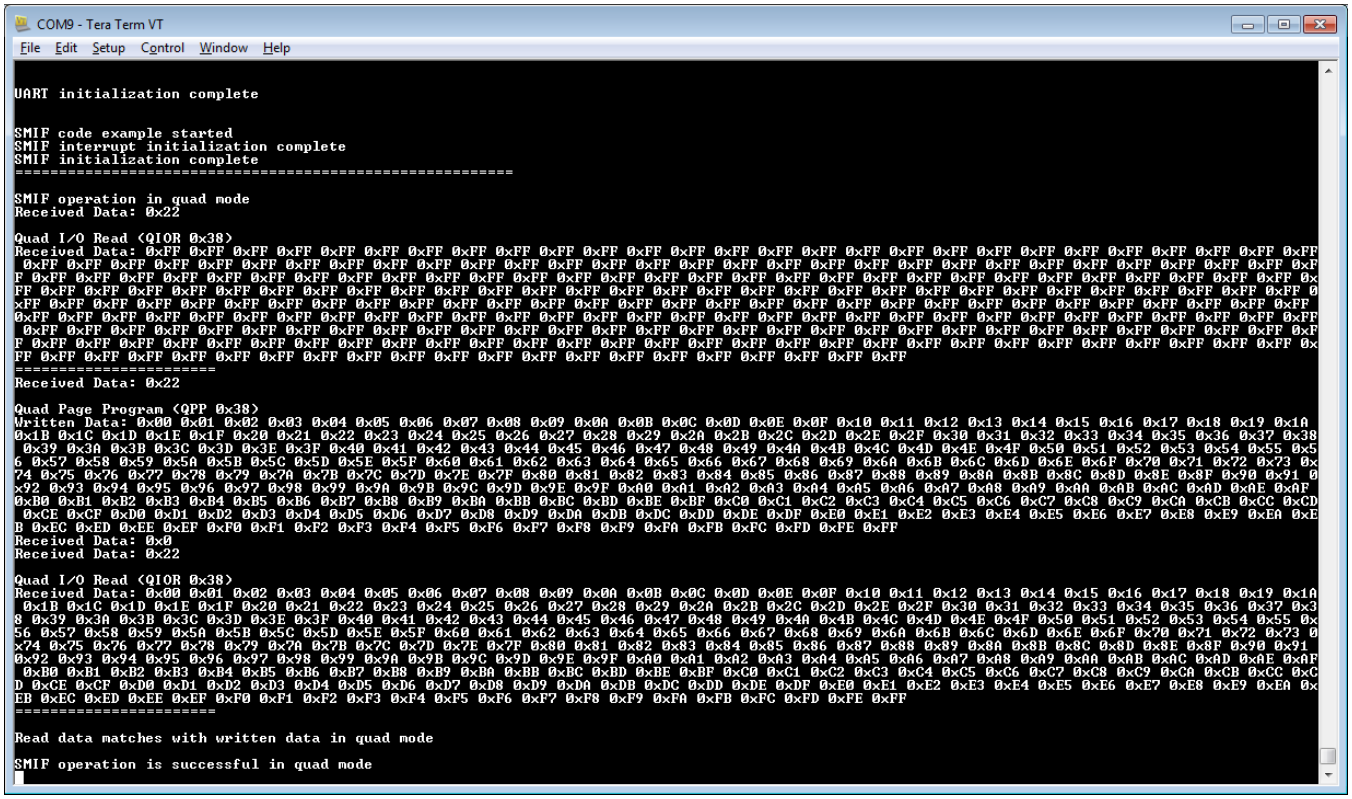
The code example works with the default settings on the CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit. If the settings are different from the default values, see the “Selection Switches” table in the kit guide to reset to the default settings.

Operation

1. Connect CY8CKIT-062-BLE to a USB port on your PC.
2. Open a serial port communication program, such as Tera Term and select the corresponding COM port. Configure the terminal to match UART: 115200 baud rate, 8N1, and Flow control – None. These settings must match the configuration of the PSoC Creator UART Component in the project.
3. Build and program the application into CY8CKIT-062-BLE. For more information on building a project or programming a device, see PSoC Creator Help.
4. Observe the LEDs to determine the status of the SMIF operation.
5. Make sure that debug messages display in the terminal window as expected.

Figure 3 is a snapshot of the debug UART terminal output.

Figure 3. Debug UART Terminal Output



Components

Table 2 lists the PSoC Creator Components used in this example and the hardware resources used by each Component.

Table 2. PSoC Creator Components

Component	Instance Name	Hardware Resources
Serial Memory Interface (SMIF_PDL)	SMIF_1	The mxsmif peripheral block
UART (SCB_UART_PDL)	UART	Single SCB peripheral block
General Purpose Input / Output (GPIO)	ERROR_LED, SUCCESS_LED	2 physical pins

Parameter Settings

Non-default settings for each Component is outlined in red in the following figures.

Figure 4 shows the SMIF_1 Component parameter settings.

Figure 4. SMIF Component Parameter Settings

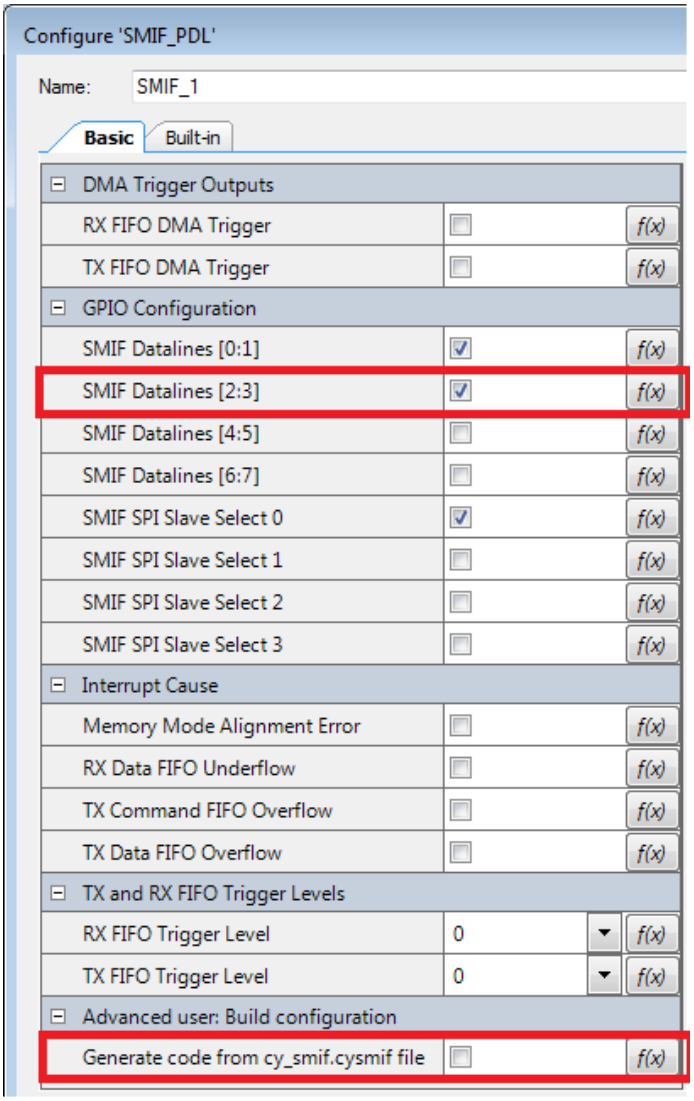
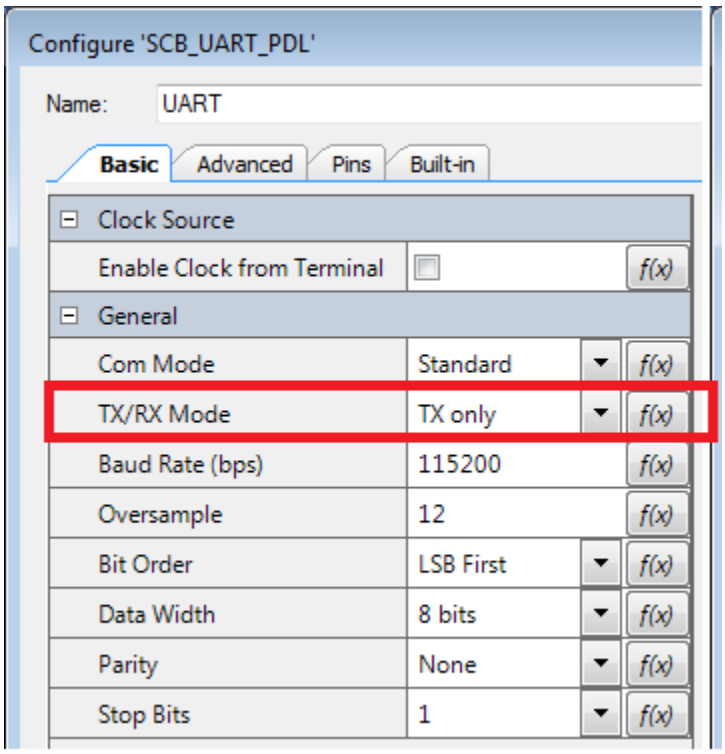


Figure 5 shows the UART Component parameter settings.

Figure 5. UART Component Parameter Settings



Design-Wide Resources

Make sure that V_{DDD} (**PSoC Creator** > **Design Wide Resources** tab > **System** tab) is set to 2.7 V or more to use SUCCESS_LED.

Table 3 lists the pin assignment for the code example.

Table 3. Pin Names and Location

Pin Name	Location
SMIF_1:spi_clk	P11[7]
SMIF_1:spi_data_0	P11[6]
SMIF_1:spi_data_1	P11[5]
SMIF_1:spi_data_2	P11[4]
SMIF_1:spi_data_3	P11[3]
SMIF_1:spi_select0	P11[2]
UART:tx	P5[1]
ERROR_LED	P0[3]
SUCCESS_LED	P1[1]

Appendix A: SMIF Configuration Tool

PSoC Creator supports a stand-alone application, SMIF Configuration Tool, which enables a user to configure the SMIF through a GUI-based interface. This application is invoked from the SMIF Component in PSoC Creator. Figure 6 and Figure 7 show how to configure the memory device interfaced with SMIF. This tool will generate a **.cysmif* file with these configuration details. Follow these steps to generate SMIF driver memory configuration (**.cysmif*, *cy_smif_memconfig.c,* and *cy_smif_memconfig.h*) files from the SMIF Configuration Tool.

1. Make sure that the "Generate code from cymem file" parameter is selected in the SMIF Component
2. Remove *cy_smif_memconfig.c* and *cy_smif_memconfig.h* files from the project workspace.
3. Right-click the SMIF Component in **PSoC Creator schematics** window and click **SMIF Configuration Tool**.
4. Configure the memory part number to match the device on the kit.
5. Click **File** and save **.cysmif* in the **example** root folder.
6. Click **Options > Configurations…** to select the example root folder as the output folder.
7. Click **Run > Generate Source Code** to generate the *cy_smif_memconfig.c* and *cy_smif_memconfig.h* files. These files provide definitions of the SMIF driver memory configuration.
8. Close the SMIF Configuration Tool.
9. Build the application. PSoC Creator generates the *cy_smif_memslot.c* and *cy_smif_memslot.h* files, the source code for the memory-level API of the SMIF driver. These APIs are used by the firmware to implement SMIF operation.

Figure 6. SMIF Configuration 1

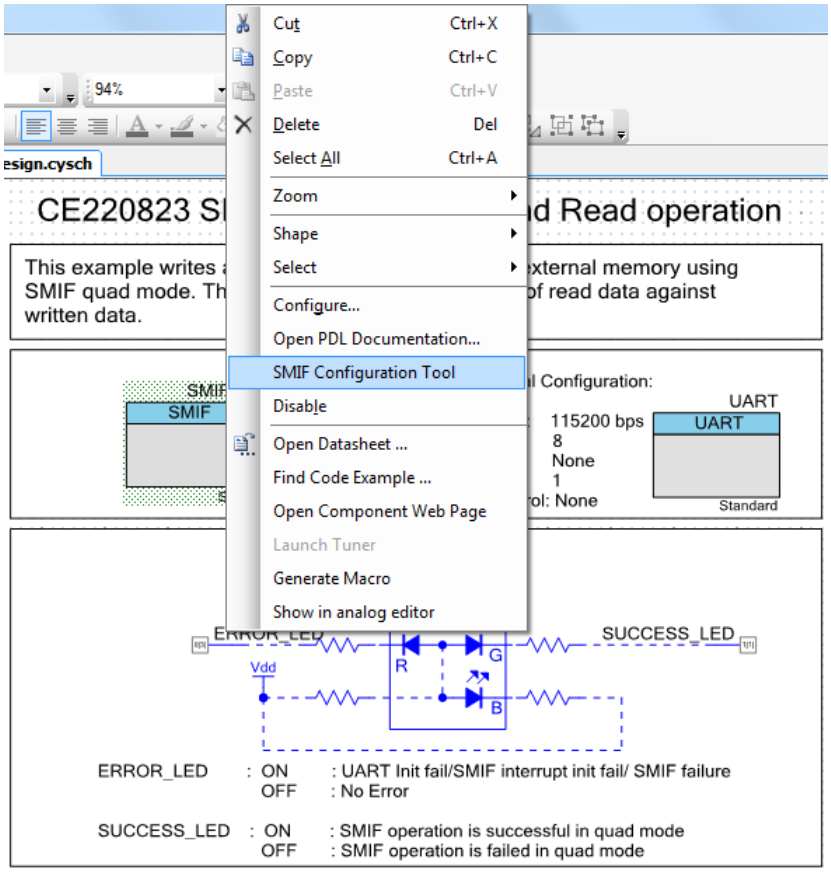
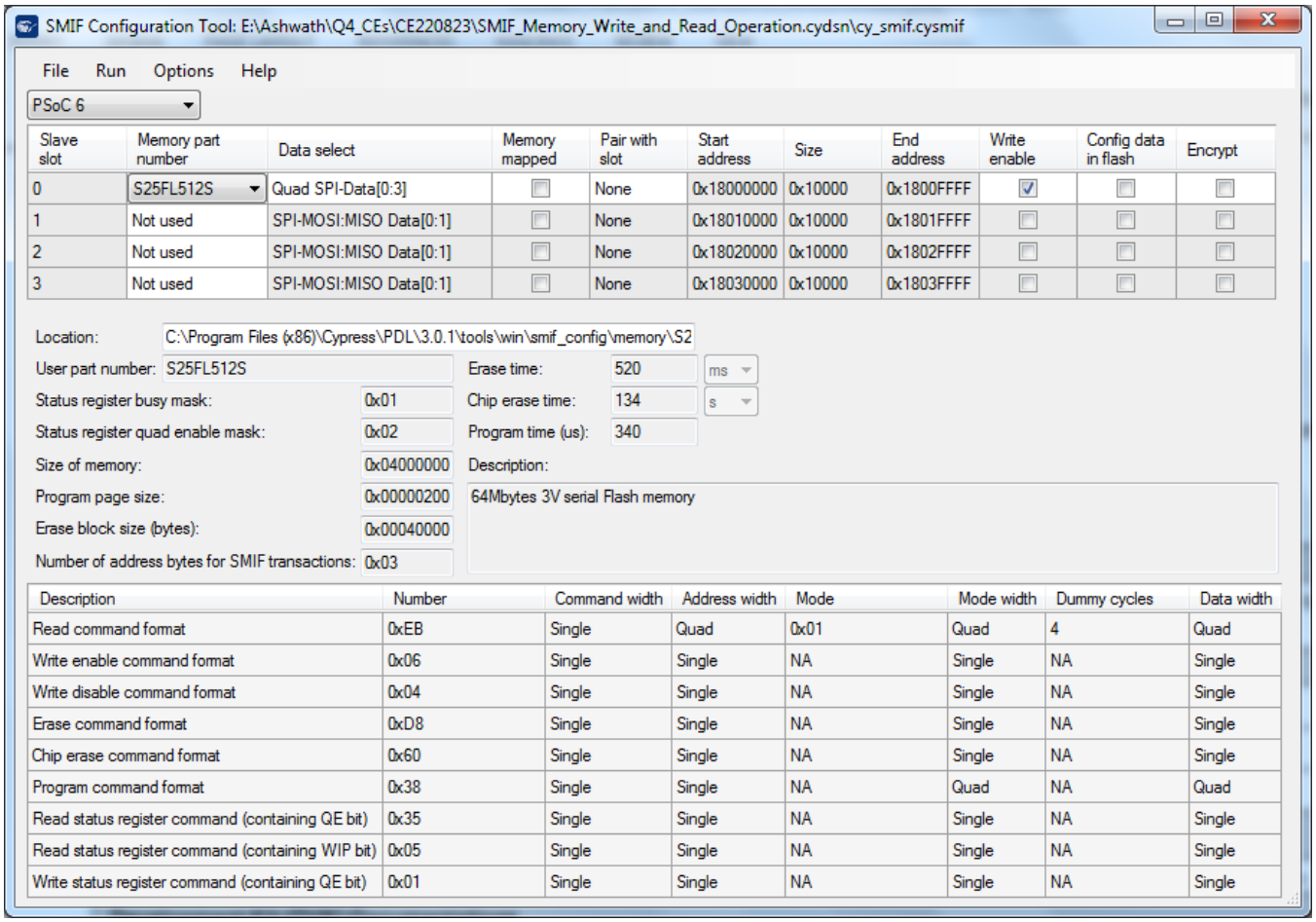


Figure 7. SMIF Configuration 2



Related Documents

Table 4 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component/user module datasheets.

Table 4. Related Documents

Application Notes	
AN210781 Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 63 with Bluetooth Low Energy (BLE) Connectivity and how to build your first PSoC Creator project
PSoC Creator Component Datasheets	
Serial Memory Interface	Supports Single/Dual/Quad/Octal SPI Memories
UART	Supports UART communication
General-Purpose Input / Output	Supports Analog, Digital I/O and Bidirectional signal types
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE220823 - PSoC 6 MCU SMIF Memory Write and Read Operation

Document Number: 002-20823

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5856613	VJYA	08/17/2017	Initial public release
*A	5918188	VJYA	11/03/2017	Updated project name
*B	6003180	VJYA	12/22/2017	Updated Figure 6 and Figure 7

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

Forums | WICED IOT 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

© Cypress Semiconductor Corporation, 2017. 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. Other names and brands may be claimed as property of their respective owners.