

Objective

This example demonstrates interfacing with an external NOR flash memory in Quad Serial Peripheral Interface (QSPI) mode using the Serial Memory Interface (SMIF) block in PSoC® 6 MCU.

Requirements

Tool: PSoC Creator™ 4.2; Peripheral Driver Library (PDL) 3.1

Programming Language: C (Arm® GCC 5.4.1 and Arm MDK 5.22)

Associated Parts: All PSoC 6 MCU parts

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

Overview

This example writes 64 bytes of data to an external memory in Quad SPI mode. This example also checks the integrity of the read data against written data.

Hardware Setup

This example uses the PSoC 6 WiFi-BT Pioneer kit's default configuration. See the kit guide to ensure the kit is configured correctly.

Software Setup

This example uses a serial terminal program. Install one on your PC if you don't have one. The instructions use [Tera Term](#).

Operation

1. Connect the Pioneer board to your PC using the provided USB cable through the USB connector.
2. Open a terminal program and select the KitProg COM port. Set the other serial port parameters as follows:
 - Baud Rate: 115200
 - Data: 8 bits
 - Parity: None
 - Stop: 1 bit
 - Flow Control: None
3. 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. Flash for both CPUs is programmed in a single program operation.
4. Press and release the USER button on the kit when prompted on the terminal window. This is required only the first time the example is run on a kit to enable Quad mode in the memory. Enabling Quad mode is required for communicating in QSPI mode.
5. Observe the KIT_LED1 to determine the status of the SMIF operation.
 - LED is blinking: Successful operation
 - LED is on: Failed operation

Make sure that debug messages displayed in the terminal window are as expected. [Figure 1](#) is a snapshot of the serial terminal output.

Figure 1. Serial Terminal Output

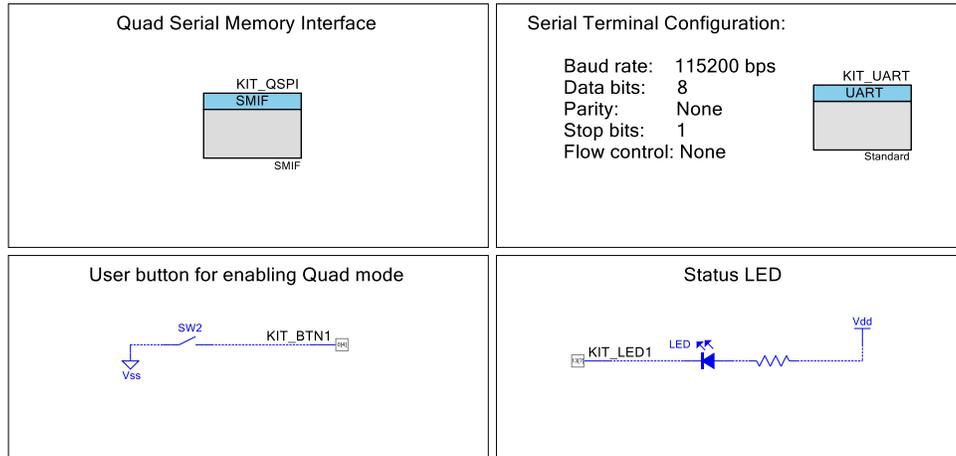
```

VT COM53 - Tera Term VT
File Edit Setup Control Window Help
SMIF code example has started
SMIF interrupt is initialized
SMIF hardware block is initialized
=====
1. Erasing 262144 bytes of memory.
2. Reading after Erase. Ensure that the data read is 0xFF for each byte.
Received Data (64 bytes):
0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
3. Writing data to memory.
Written Data (64 bytes):
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
4. Reading back for verification.
Received Data (64 bytes):
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0x0B 0x0C 0x0D 0x0E 0x0F
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1A 0x1B 0x1C 0x1D 0x1E 0x1F
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2A 0x2B 0x2C 0x2D 0x2E 0x2F
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F
=====
SUCCESS: Read data matches with written data!
=====
    
```

Design and Implementation

The QSPI resource implements a SPI-based communication for interfacing external memory devices with PSoC. QSPI resource is configured with four data lines and a single slave select line. The UART resource outputs debug information to a terminal window. A user LED indicates the status of read and write operation. [Figure 2](#) shows the design for this code example.

Figure 2. Design Schematic



The firmware uses source code (*cy_smif_memconfig.c* and *cy_smif_memconfig.h* files) generated from the QSPI Configurator. This source code defines data structures that hold the memory configuration.

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
SMIF	KIT_QSPI	Communicate with QSPI NOR Flash	See Figure 3 and Figure 4
SCB	KIT_UART	UART to transmit debug data	Default
Digital Output Pin	KIT_LED1	LED to indicate SMIF transfer status	See Figure 5
Digital Input Pin	KIT_BTN1	User button for enabling Quad mode	See Figure 6

For information on the hardware resources used by a Component, see the Component datasheet.

[Figure 3](#) through [Figure 6](#) highlight the non-default settings for each component in this example.

Figure 3. QSPI Component Configurator

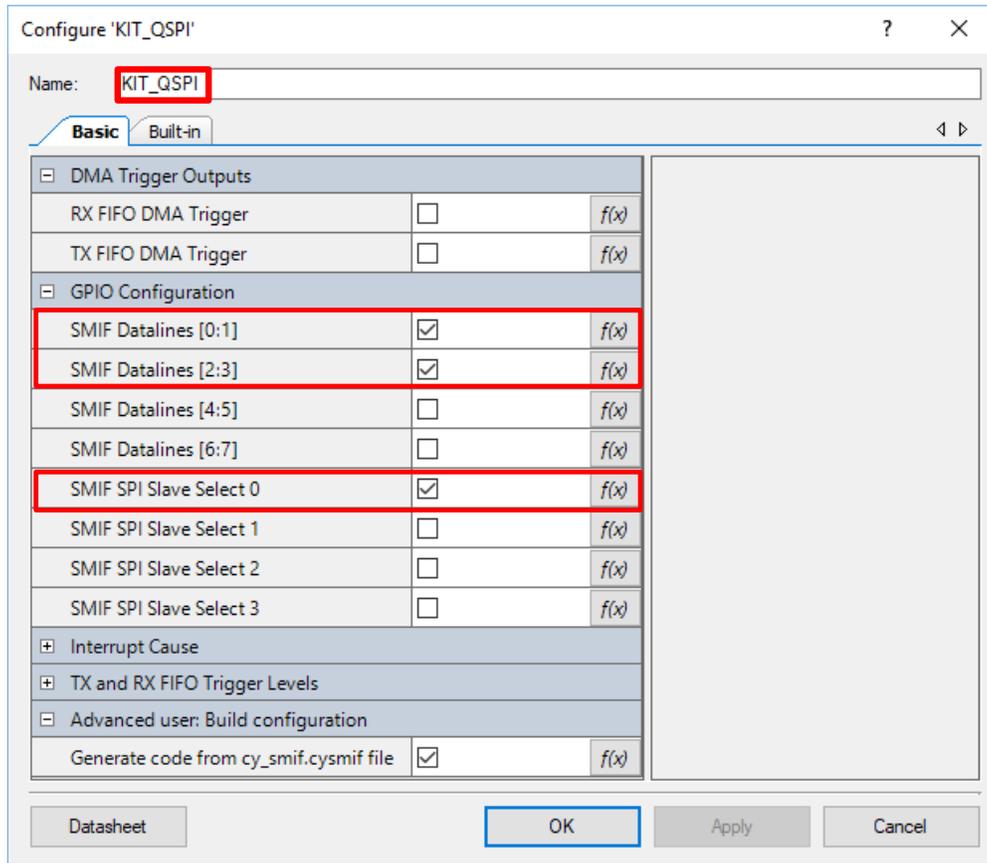
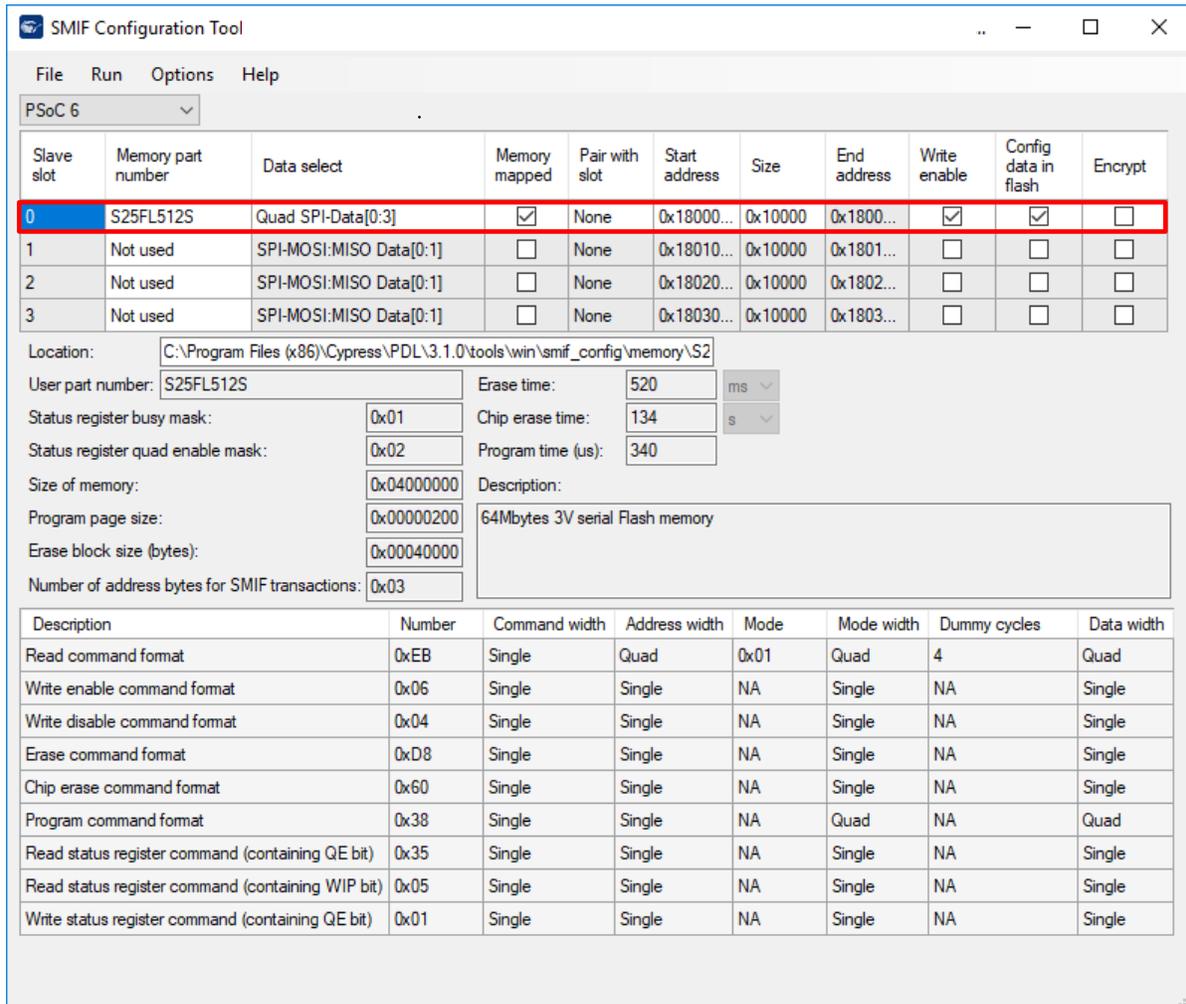


Figure 4. QSPI Configurator Tool



The screenshot shows the SMIF Configuration Tool window for a PSoC 6. The main configuration table is as follows:

Slave slot	Memory part number	Data select	Memory mapped	Pair with slot	Start address	Size	End address	Write enable	Config data in flash	Encrypt
0	S25FL512S	Quad SPI-Data[0:3]	<input checked="" type="checkbox"/>	None	0x18000...	0x10000	0x18000...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1	Not used	SPI-MOSI:MISO Data[0:1]	<input type="checkbox"/>	None	0x18010...	0x10000	0x18010...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Not used	SPI-MOSI:MISO Data[0:1]	<input type="checkbox"/>	None	0x18020...	0x10000	0x18020...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Not used	SPI-MOSI:MISO Data[0:1]	<input type="checkbox"/>	None	0x18030...	0x10000	0x18030...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

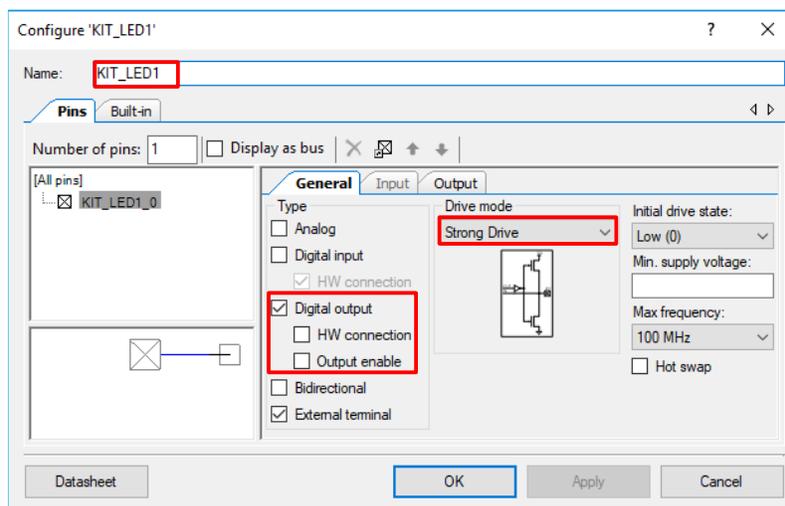
Additional configuration parameters shown below the table:

- Location: C:\Program Files (x86)\Cypress\PDL\3.1.0\tools\win\smif_config\memory\S2
- User part number: S25FL512S
- Erase time: 520 ms
- Status register busy mask: 0x01
- Chip erase time: 134 s
- Status register quad enable mask: 0x02
- Program time (us): 340
- Size of memory: 0x04000000
- Program page size: 0x00000200
- Erase block size (bytes): 0x00040000
- Number of address bytes for SMIF transactions: 0x03
- Description: 64Mbytes 3V serial Flash memory

A command format table is also present at the bottom of the tool:

Description	Number	Command width	Address width	Mode	Mode width	Dummy cycles	Data width
Read command format	0xEB	Single	Quad	0x01	Quad	4	Quad
Write enable command format	0x06	Single	Single	NA	Single	NA	Single
Write disable command format	0x04	Single	Single	NA	Single	NA	Single
Erase command format	0xD8	Single	Single	NA	Single	NA	Single
Chip erase command format	0x60	Single	Single	NA	Single	NA	Single
Program command format	0x38	Single	Single	NA	Quad	NA	Quad
Read status register command (containing QE bit)	0x35	Single	Single	NA	Single	NA	Single
Read status register command (containing WIP bit)	0x05	Single	Single	NA	Single	NA	Single
Write status register command (containing QE bit)	0x01	Single	Single	NA	Single	NA	Single

Figure 5. KIT_LED1 Pin Configuration



The screenshot shows the 'Configure KIT_LED1' dialog box. The 'Name' field is set to 'KIT_LED1'. Under the 'Pins' tab, 'KIT_LED1_0' is selected. The 'General' sub-tab is active, showing the following configuration:

- Type: Analog, Digital input, Digital output, Bidirectional
- Drive mode: Strong Drive (selected in a dropdown menu)
- Initial drive state: Low (0)
- Min. supply voltage: (empty field)
- Max frequency: 100 MHz
- Hot swap:

Other options include 'HW connection' (checked), 'Output enable' (unchecked), and 'External terminal' (checked). The 'Display as bus' checkbox is unchecked. Buttons for 'Datasheet', 'OK', 'Apply', and 'Cancel' are visible at the bottom.

Figure 6. KIT_BTN1 Pin Configuration

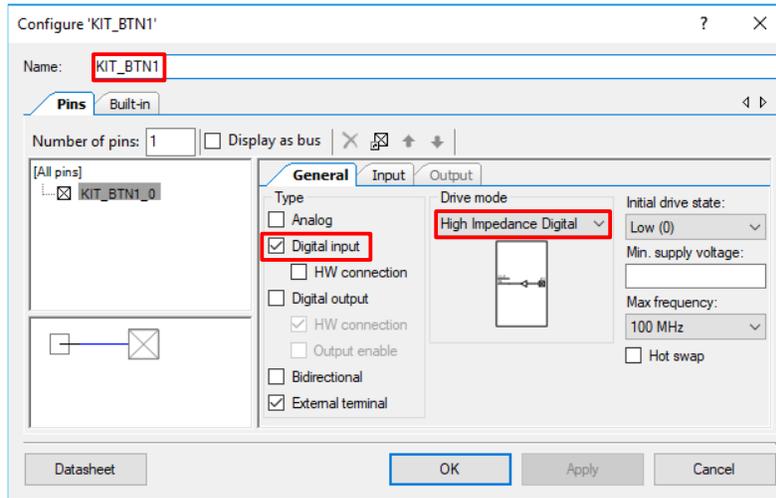


Figure 7 shows pin assignment for the code example.

Figure 7. Pin Assignment

	Name	Port
	\KIT_QSPI:spi_clk\	P11[7]
	\KIT_QSPI:spi_data_0\	P11[6]
	\KIT_QSPI:spi_data_1\	P11[5]
	\KIT_QSPI:spi_data_2\	P11[4]
	\KIT_QSPI:spi_data_3\	P11[3]
	\KIT_QSPI:spi_select0\	P11[2]
	\KIT_UART:rx\	P5[0]
	\KIT_UART:tx\	P5[1]
	KIT_BTN1	P0[4]
	KIT_LED1	P13[7]

Reusing This Example

This example is designed for the supported kit(s). 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. For single-core PSoC 6 MCU devices, port the code from *main_cm4.c* to *main.c*. If you change the device or design, you may need to adjust the source files that come with the example. These are not automatically generated and may not work with a modified design.

In some cases a resource used by a code example (for example, an IP block) 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

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

For a comprehensive list of PSoC 3, PSoC 4, and PSoC 5LP resources, see [KBA86521](#) in the Cypress community.

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator 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
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
UART	Supports UART communication
Serial Memory Interface	Supports Single/Dual/Quad/Octal SPI Memories
Device Documentation	
PSoC 6 MCU Datasheets	PSoC 6 Technical Reference Manuals
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit	
CY8CPROTO-062-4343W PSoC 6 Wi-Fi BT Prototyping Kit	
CY8CPROTO-063 BLE PSoC 6 BLE Prototyping Kit	
Tool Documentation	
PSoC Creator	Look in the downloads tab for Quick Start and User Guides
Peripheral Driver Library (PDL)	Get the latest version for use with PSoC Creator. Look in the <PDL install folder>/doc for the User Guide and the API Reference

Cypress Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right device, and quickly and effectively integrate the device into your design.

The following is an abbreviated list of resources related to this code example:

- **Overview:** [MCU Portfolio](#), [PSoC & MCU Roadmap](#)
- **Product Selectors:** [PSoC 1](#), [PSoC 3](#), [PSoC 4](#), [PSoC 5LP](#), or [PSoC 6](#). In addition, [PSoC Creator](#) includes a device selection tool.
- **Datasheets:** Describe and provide electrical specifications for MCU and PSoC device families.
- **Application Notes:** Cover a broad range of topics, from basic to advanced level.
- **Code Examples:** for [PSoC 3](#), [PSoC 4](#), and [PSoC 5LP](#); or for [PSoC 6](#).
- **PSoC Technical Reference Manuals (TRM):** Provide detailed descriptions of the architecture and registers for a PSoC device family.
- **Training Videos:** These videos provide guidance on getting started with various Cypress product families and tools.
- **PSoC 6 MCU Training Videos:** Provide guidance on getting started.
- **CapSense Design Guides:** Learn how to design capacitive touch-sensing applications.
- **Development Kits:** Some examples include:
 - [PSoC 6 BLE Pioneer Kit](#) is a low-cost hardware platform that enables design and debug of the PSoC 63 series. It comes with an E-Ink display shield board.
 - [PSoC 6 WiFi-BT Pioneer Kit](#) supports the PSoC 62 series MCU along with Wi-Fi and BT connectivity
 - [CY8CKIT-042](#) and [CY8CKIT-040](#), Pioneer kits, are easy-to-use and inexpensive development platforms. These kits include connectors for Arduino™ compatible shields and Digilent® Pmod™ daughter cards.
 - [CY8CKIT-049](#) is a series of very low-cost prototyping platform for sampling PSoC 4 devices.
 - [CY8CKIT-030](#) and [CY8CKIT-050](#) are designed for analog performance. They enable you to evaluate, develop, and prototype high-precision analog, low-power, and low-voltage applications powered by PSoC 3 and PSoC 5LP, respectively.
 - [CY8CKIT-001](#) is a common development platform for all PSoC family devices.
- The [MiniProg3](#) device provides an interface for flash programming and debug.

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
*C	6554903	AJYA	04/23/2019	Code example has been revised completely for clarity and correctness

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](#) | [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, 2017-2019. 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.