CE222460 - SPI F-RAM Access Using PSoC 6 MCU SMIF

Objective

CE222460 demonstrates accessing the SPI F-RAM™ using PSoC® 6 MCU's Serial Memory Interface (SMIF) Component.

Overview

CE222460 provides a code example that implements the SPI host controller on PSoC 6 MCU using the SMIF Component and demonstrates accessing different features of the SPI F-RAM. The result is displayed by driving the status LED (RGB) which turns green when the result is a pass, and turns red when the result is a fail. The code example also enables the UART interface to connect to a PC to monitor the result.

Requirements

Tool: PSoC Creator™ 4.2; Peripheral Driver Library (PDL) 3.0.1

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

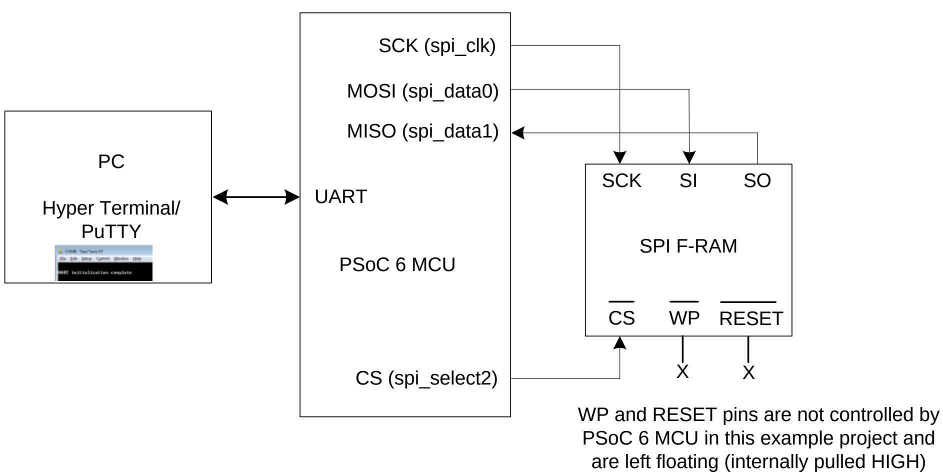
Associated Parts: All PSoC 6 MCU parts

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit with SPI/QSPI F-RAM

Hardware Setup

The hardware setup includes connecting the SPI F-RAM with PSoC 6 MCU. You can use either dedicated hardware as described in the Requirements section or can connect via jumper wires by tapping the SMIF SPI control pins and connect to the SPI pins of an external SPI F-RAM. This example uses the PSoC 6 BLE Pioneer kit's default configuration. Refer to the kit guide to ensure the kit is configured correctly.

Figure 1. Hardware Setup Block Diagram

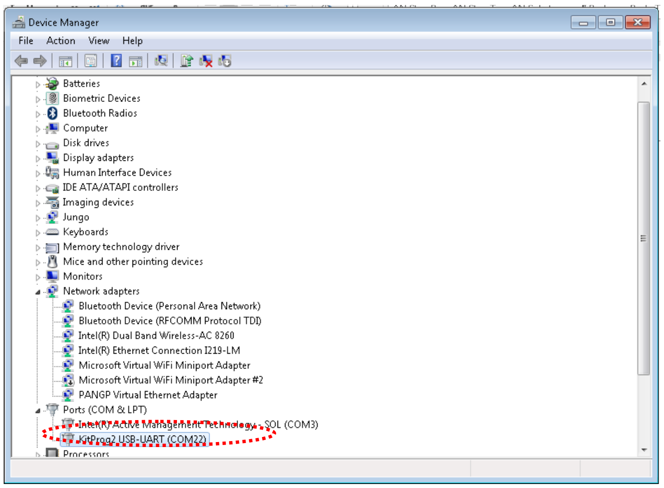


Software Setup

This section demonstrates the procedure to setup the serial (UART) connection using PuTTY on PC to communicate with the PSoC 6 Pioneer Kit. PuTTY is a free SSH and telnet client for Windows. You can download PuTTY from www.putty.org. Follow these instructions to determine the COM port number and setup the PuTTY to monitor the code example outputs on PC.

1. Connect PSoC 6 Pioneer Kit to the PC using USB cable. The kit enumerates as KitProg2 USB-UART and is available under the **Device Manager** > **Ports (COM & LPT)**. A communication port (COMx) is assigned to KitProg2 USB-UART; for example, COM22 is assigned to PSoC 6 Pioneer Kit on the sample setup, shown in Figure 2.

Figure 2. KitProg2 USB-UART in Device Manager

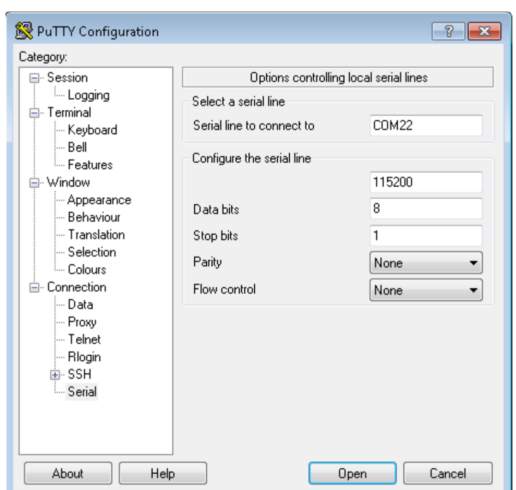


2. After you download and install PuTTY, double-click the PuTTY icon and select **Serial** under **Connection**.

3. A new window opens where the communication port can be selected. Do the following in the **Options controlling local serial lines** section:

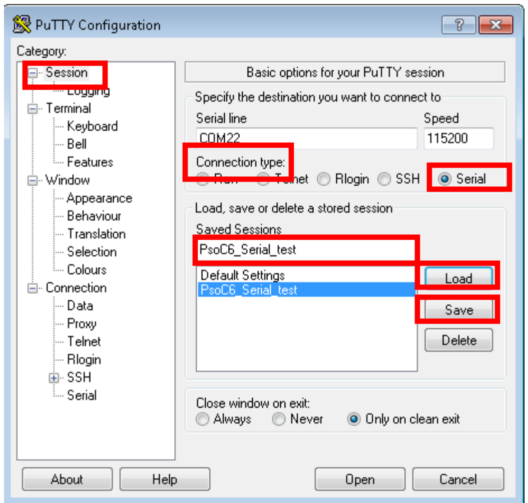
   - Enter the PSoC 6 Port (COM & LPT), COMx, in **Serial line to connect to**. This code example uses **COM22**. Verify the COM setting for your setup and select the appropriate COMx.
   - Enter **Speed (baud)**, **Data bits**, and **Stop bits**.
   - Select **Parity** and **Flow control**.

Figure 3. Open New Connection



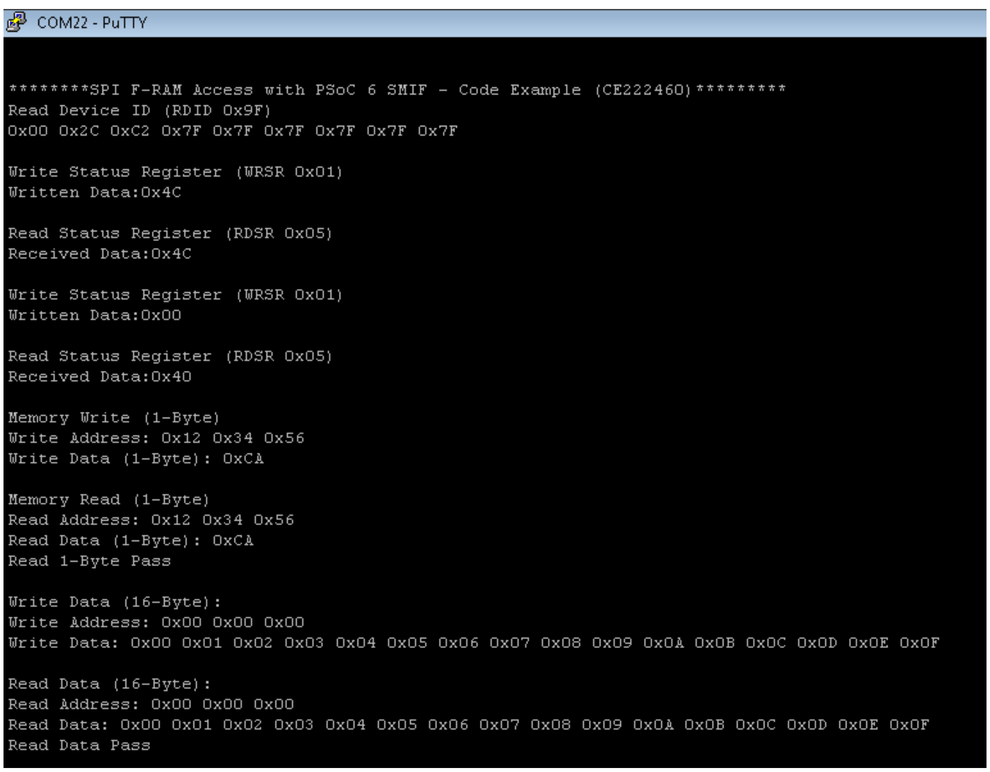
4. Select **Session** under **Category.** Select **Serial** under **Connection type** as shown in Figure 4. You can save this current session and **load** the settings when required. Enter a name in **Saved Sessions** and click **Save**. Click **Open** to proceed.

Figure 4. Select Communication Type in PuTTY



5. The COM terminal window then displays the code example results as shown in Figure 5. You may have to reprogram PSoC 6 MCU with the code example hex file or reset PSoC 6 MCU (already programmed) to restart the code execution and monitor the result.

Figure 5. Result Displayed on PuTTY



Alternatively, you can run the HyperTerminal if supported on your PC to monitor the above result.

Operation

The code example executes the following functions on SPI F-RAM:

- Executes write and read access to the Status Register
- Executes random (1-Byte) memory write, read and verify from an address
- Executes burst (16-Byte) memory write, read and verify from a start address
- Executes burst (16-Byte) special sector write, burst read and verify from an address
- Executes Serial Number write, read and verify

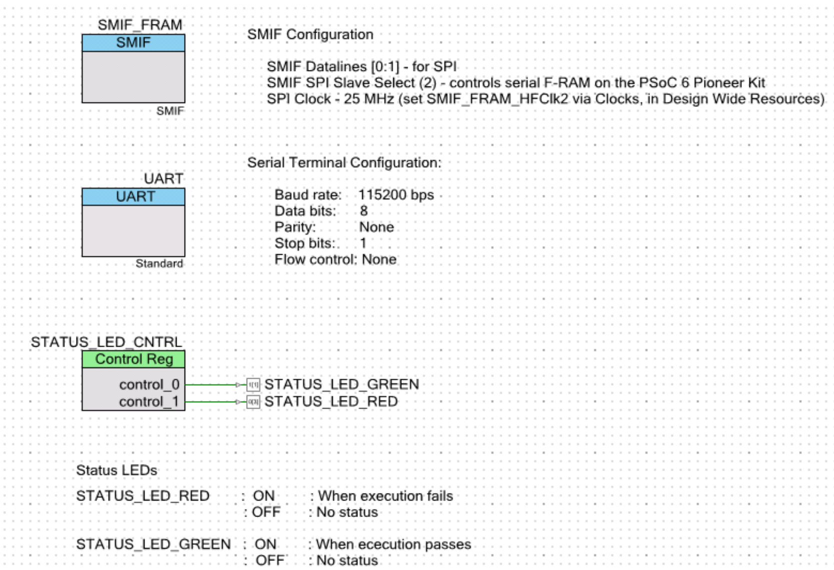
Do the following to execute the code example project. Refer to the Design and Implementation section for more details.

1. Connect the CY8CKIT-062-BLE Pioneer Kit to a USB port on your PC. Set the V_{DD} select either 1.8 V or 3.3 V using the switch SW5 on PSoC 6 Pioneer Kit. The SPI/QSPI F-RAM supports wide operating range V_{DD} = 1.8 V to 3.6 V.
2. Open a serial port communication program such as PuTTY and select the corresponding COM port. Configure the terminal to match the 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 the CY8CKIT-062-BLE Kit or CY8CKIT-062 Kit which has serial F-RAM mounted on it. For more information on building a project or programming a device, see *PSoC Creator Help*.
4. Observe the result status by monitoring RGB LED. The LED toggles GREEN when result is a pass and RED when result is a fail.
5. Observe the UART example header message printed in the terminal window. Figure 5 shows a snapshot of a sample UART terminal output.

Design and Implementation

Figure 6 shows the design for this code example. The SMIF Component implements a SPI-based communication for interfacing with an external SPI F-RAM with PSoC 6 MCU. The SMIF Component is configured with two data lines, single slave select line, and the SPI clock (SCK) at 25 MHz. The UART Component outputs debug information to a terminal window. It is configured for 8N1, transmit only, at 115.2 kbps. The code example also uses the Control Register Component to drive the RGB LED on PSoC 6 Pioneer kit to display results.

Figure 6. CE222460 Design Schematic in PSoC Creator



Cypress SPI F-RAMs support SPI clock frequency (SCK) up to 50 MHz.

Components and Settings

Table 1 lists all the PSoC Creator Components used in the three examples.

Table 1. PSoC Creator Components

Component	Instance Name	Purpose
SMIF(SMIF_PDL)	SMIF_FRAM	The smif peripheral block. Configures the SPI host controller in the design
UART (SCB_UART_PDL)	UART	Handle communication to the terminal window
Control Register (CyControlReg)	STATUS_LED_CNTRL	Drives the status (RGB) LED output

Parameter settings

Non-default settings for each Component are outlined in red in the following figures. Figure 7 shows the SMIF_FRAM Component parameter settings.

Figure 7. Settings for SMIF and UART Components (Non-default settings are outlined)

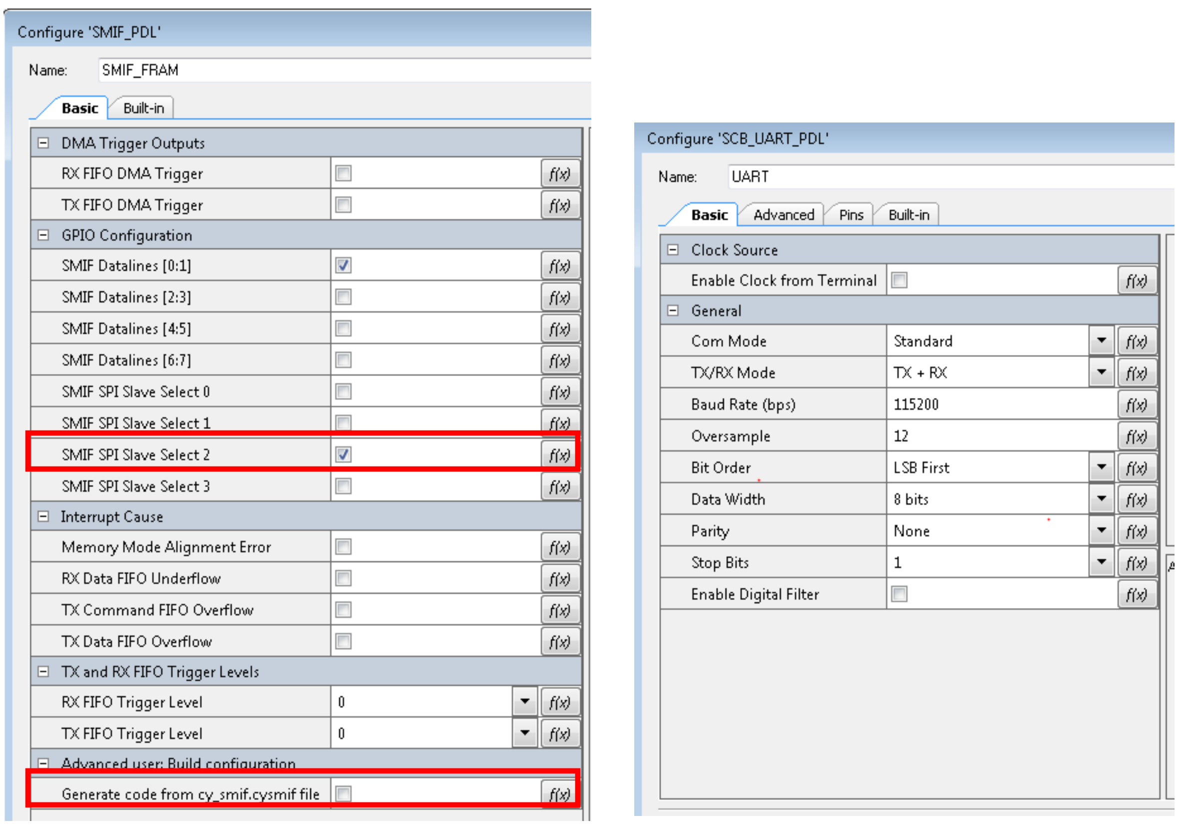
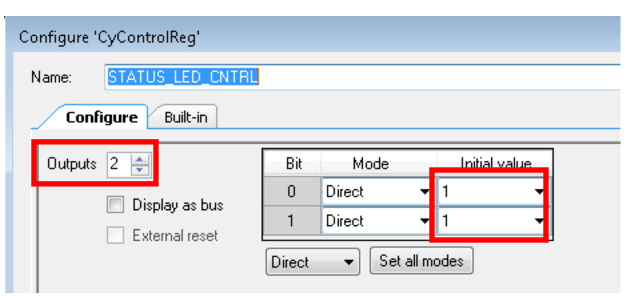


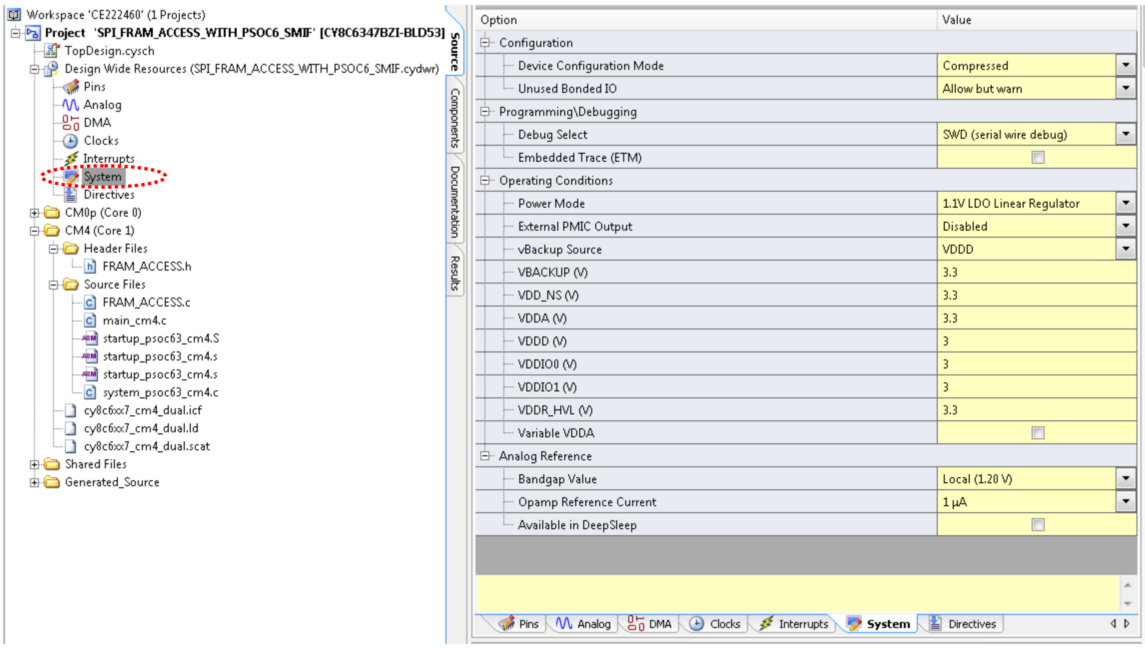
Figure 8. Settings for Control Register (Non-default settings are outlined)



Design-Wide Resources

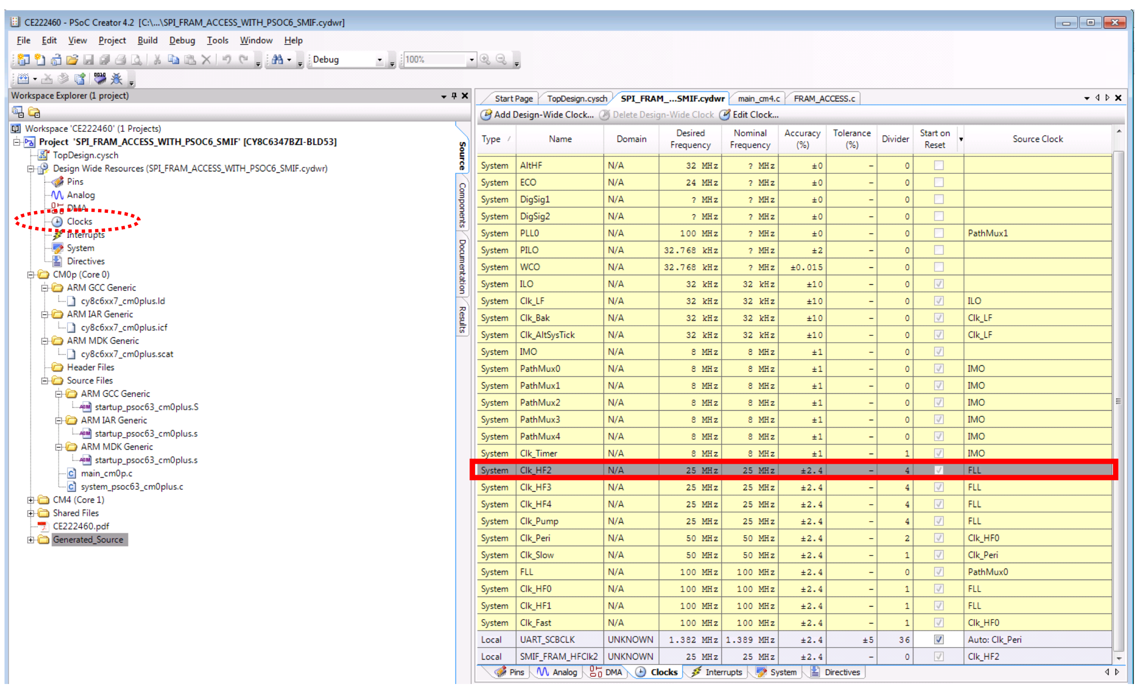
Make sure that V_{DDD} (**PSoC Creator** > **Design Wide Resources** > **System** tab) is set to 2.7 V or above, as shown in Figure 9, to drive the STATUS_LED. Also, make sure that PSoC 6 MCU I/O voltage is set correctly to match the SPI F-RAM operating range (V_{DD}/ V_{CC}).

Figure 9. V_{DD} Setting using Design Wide Resources



Make sure that SMIF SPI clock frequency is set to 50 MHz or below as shown via Figure 10 and Figure 11. Go to **PSoC Creator** > **Design Wide Resources** > **Clocks** and click on it.

Figure 10. SMIF SPI Clock Setting using Design Wide Resources – Step 1



Double-click anywhere in **Clk_HF2** row, as highlighted in Figure 11. A new **Configure System Clocks** window opens. Select the **High Frequency Clocks** tab and use appropriate clock path and the clock divider from their drop-down options, as highlighted in Figure 11, to achieve 50 MHz or below frequency. This code example has set the SPI clock frequency as 25 MHz. Select **OK**. You can use **FLL/PLL** tab to configure other frequency options for Path 0.

Figure 11. SMIF SPI Clock Setting using Design Wide Resources – Step 2

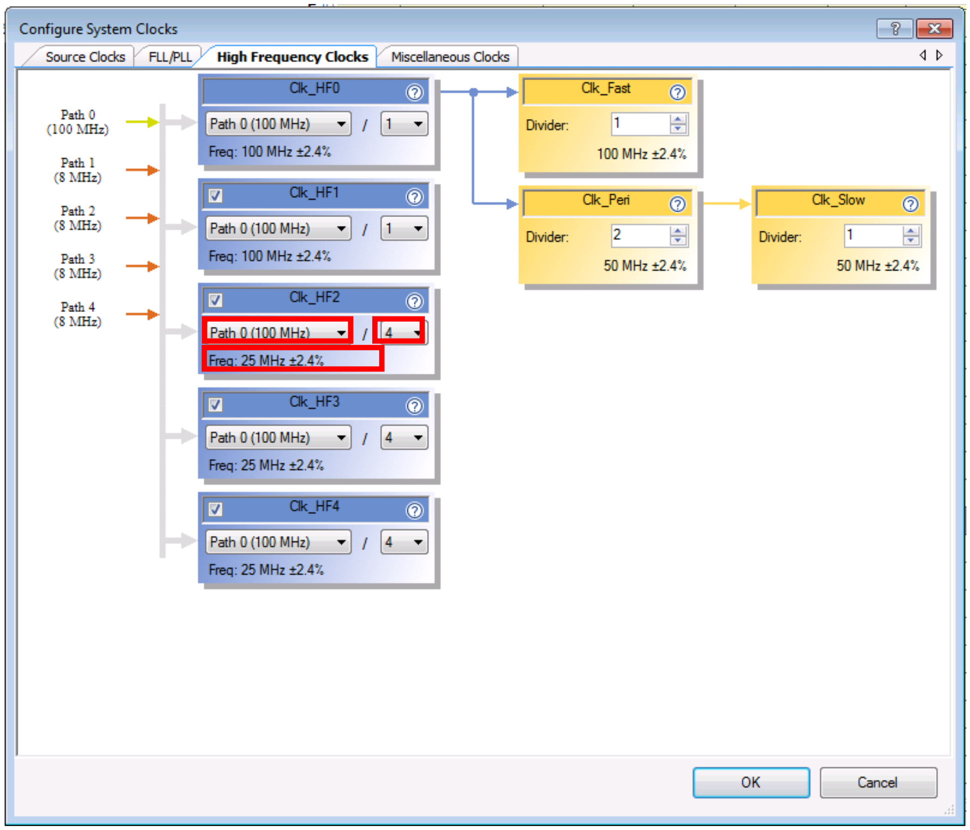


Figure 12 shows the pin assignment for the code example.

Figure 12. PSoC 6 Pin Assignments for Code Example

Name	Port	Pin
\SMIF_FRAM:spi_clk\	P11[7]	A5
\SMIF_FRAM:spi_data_0\	P11[6]	B5
\SMIF_FRAM:spi_data_1\	P11[5]	A6
\SMIF_FRAM:spi_select2\	P11[0]	F5
\UART:rx\	P5[0]	L6
\UART:tx\	P5[1]	K6
STATUS_LED_GREEN	P1[1]	F2
STATUS_LED_RED	P0[3]	E3

Reusing This Example

This example is designed for the CY8CKIT-062-BLE Pioneer Kit with serial F-RAM mounted on it. To port the design to a different PSoC 6 MCU device and/or kit, change the target device using Device Selector and update the pin assignments in Design Wide Resources Pins settings. For single-CPU PSoC 6 MCU devices, port the code from *main_cm4.c* to *main.c*.

Related Documents

Application Notes/Code Examples	
CE220823 – PSoC 6 MCU SMIF Memory Write and Read Operation	This example demonstrates the write and read operations to the Serial Memory Interface (SMIF) in PSoC 6 MCU.
AN304 – SPI Guide for F-RAM™	AN304 provides the functional description, timing, and example code for SPI F-RAMs.
PSoC Creator Component Datasheets	
UART	UART communications interface
SMIF	Serial Memory Interface
Control Register	Allows the firmware to set values for to use for digital signals
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
SPI F-RAM (CY15B104QN) Datasheet	3V, 50 MHz SPI, 4Mb SPI FRAM datasheet
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE222460 – SPI F-RAM Access Using PSoC 6 MCU SMIF

Document Number: 002-22460

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5997527	ZSK	01/03/2018	New Code Example

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