Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CE222967 – Excelon-Ultra QSPI F-RAM Access Using PSoC 6 MCU SMIF

Objective

CE222967 demonstrates accessing the Excelon™-Ultra QSPI F-RAM™ using PSoC® 6 MCU's Serial Memory Interface (SMIF) Component in memory mapped I/O (MMIO) mode.

Overview

CE222967 provides a code example that implements the QSPI host controller on the PSoC 6 MCU device using the SMIF Component and demonstrates accessing different features of the QSPI 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) 3.1.0

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

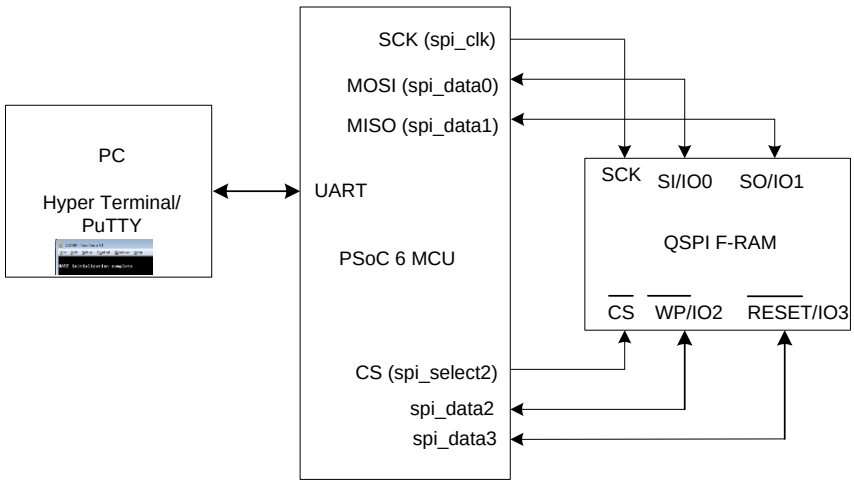
Associated Parts: All PSoC 6 MCU parts

Related Hardware: PSoC 6 WiFi-BT Pioneer Kit (CY8CKIT-062-WiFi-BT)

Hardware Setup

The hardware setup includes connecting the QSPI 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 QSPI control pins and connect to the QSPI pins of an external QSPI F-RAM. This example uses the PSoC 6 WiFi-BT Pioneer kit's default configuration. See the kit guide to ensure the kit is configured correctly.

Figure 1. Hardware Setup Block Diagram

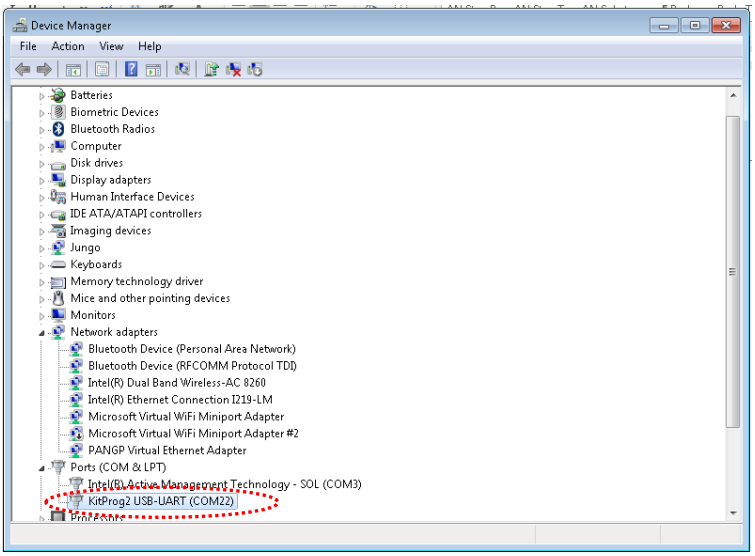


Software Setup

This section demonstrates the procedure to set up a serial (UART) connection using PuTTY on a 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 set up PuTTY to monitor the code example outputs on your PC.

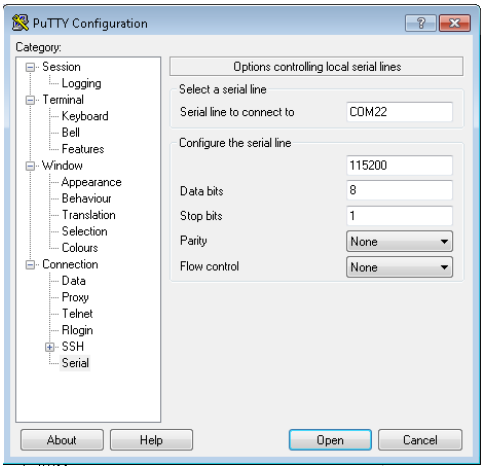
1. Connect the PSoC 6 Pioneer Kit to the PC using the USB cable. The kit enumerates as KitProg2 USB-UART and is available under the **Device Manager** > **Ports (COM & LPT)**. A communication port (COMx) is assigned to KitProg2 USB-UART; for example, COM22 is assigned to PSoC 6 Pioneer Kit on the sample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

Figure 2. KitProg2 USB-UART in Device Manager



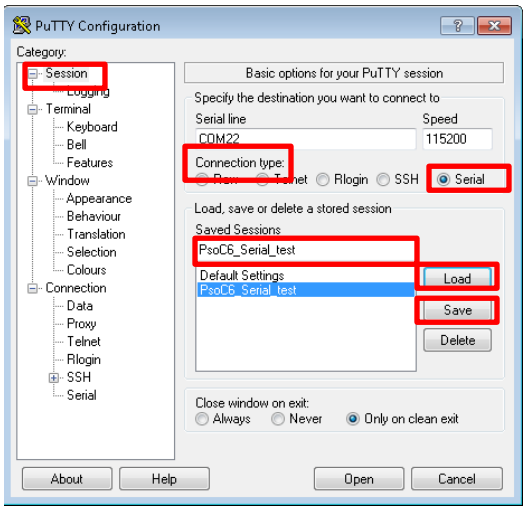
2. After you download and install PuTTY, double-click the PuTTY icon and select **Serial** under **Connection**.
3. A new window opens where you can select the communication port. Do the following in the **Options controlling local serial lines** section:
   a. Enter the PSoC 6 Port (COM & LPT), COMx, in **Serial line to connect to**. This code example uses **COM22**. Verify the COM setting for your setup and select the appropriate COMx.
   b. Enter **Speed (baud)**, **Data bits**, and **Stop bits**.
   c. Select **Parity** and **Flow control**.

Figure 3. Open New Connection



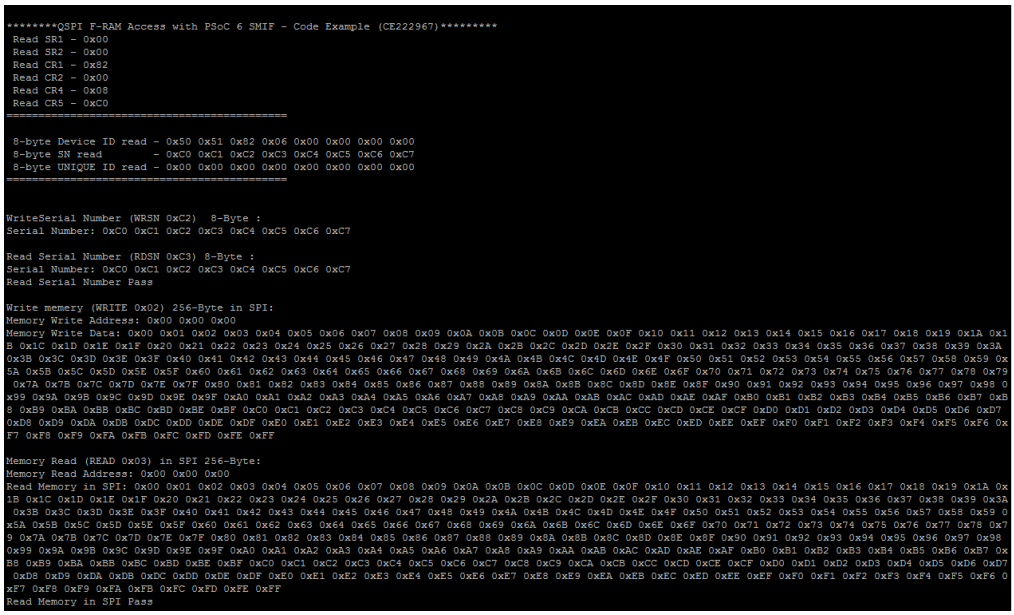
4. Select **Session** under **Category**. Select **Serial** as the **Connection type** as shown in Figure 4. You can save this current session and load the settings when required. Enter a name in **Saved Sessions** and click **Save**. Click **Open** to proceed.

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

This code example demonstrates accessing the QSPI F-RAM's standard write and read features including memory and user registers in Memory Mapped I/O (MMIO) mode. This code example does not support user APIs for execute-in-place (XIP), CRC, and ECC features of F-RAM. The following features of the QSPI F-RAM are demonstrated through this code example:

- Sets the device access mode to default SPI mode. This ensures that the part starts with a known SPI mode.
- Reads two Status and four Configuration registers (SR1, SR2, CR1, CR2, CR4, CR5)
- Reads the 8-byte device ID
- Writes 256 bytes into F-RAM at a given address, in SPI mode
- Reads 256 bytes from F-RAM at a given address, using the READ (0x02) opcode in SPI mode
- Reads 256 bytes from F-RAM at a given address, using the READ (0x02) opcode in DPI mode
- Reads 256 bytes from F-RAM at a given address, using the READ (0x02) opcode in QPI mode
- Reads 256 bytes from F-RAM at a given address, using the FastRead (0x0B) opcode in QPI mode
- Writes and read 256 bytes special sector using SSWR and SSRD commands in QPI mode

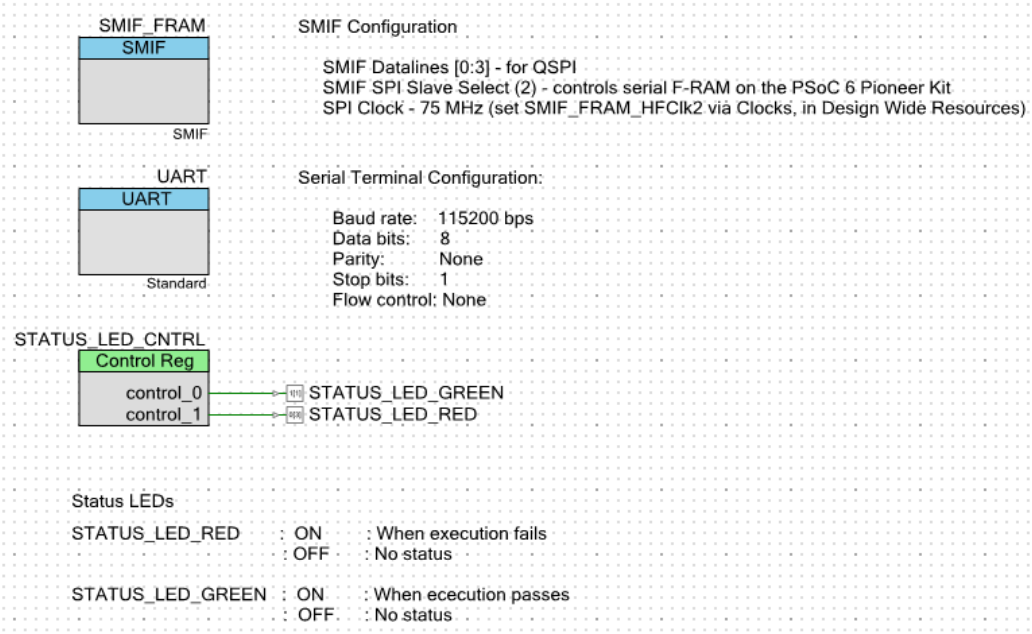
Do the following to execute the code example project. See the Design and Implementation section for more details.

1. Connect the CY8CKIT-062-WiFi-BT Pioneer kit to a USB port on your PC. Set V_{DD} select either 1.8 V or 3.3 V using the switch SW5 on PSoC 6 Pioneer kit. The Excelon-Ultra QSPI F-RAM supports wide operating voltage 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-WiFi-BT Kit or CY8CKIT-062-BLE Kit, which has the QSPI 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 quad data lines [Datalines[0:3]) GPIO configuration for interfacing with an external QSPI F-RAM with PSoC 6 MCU. The SMIF Component is configured with four data lines, single slave select line, and the SPI clock (SCK) at 75 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 the PSoC 6 Pioneer kit to display results.

Figure 6. CE222967 Design Schematic in PSoC Creator



Components and Settings

Table 1 lists 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 QSPI host controller in the design.
UART (SCB_UART_PDL)	UART	Handles 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 is 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)

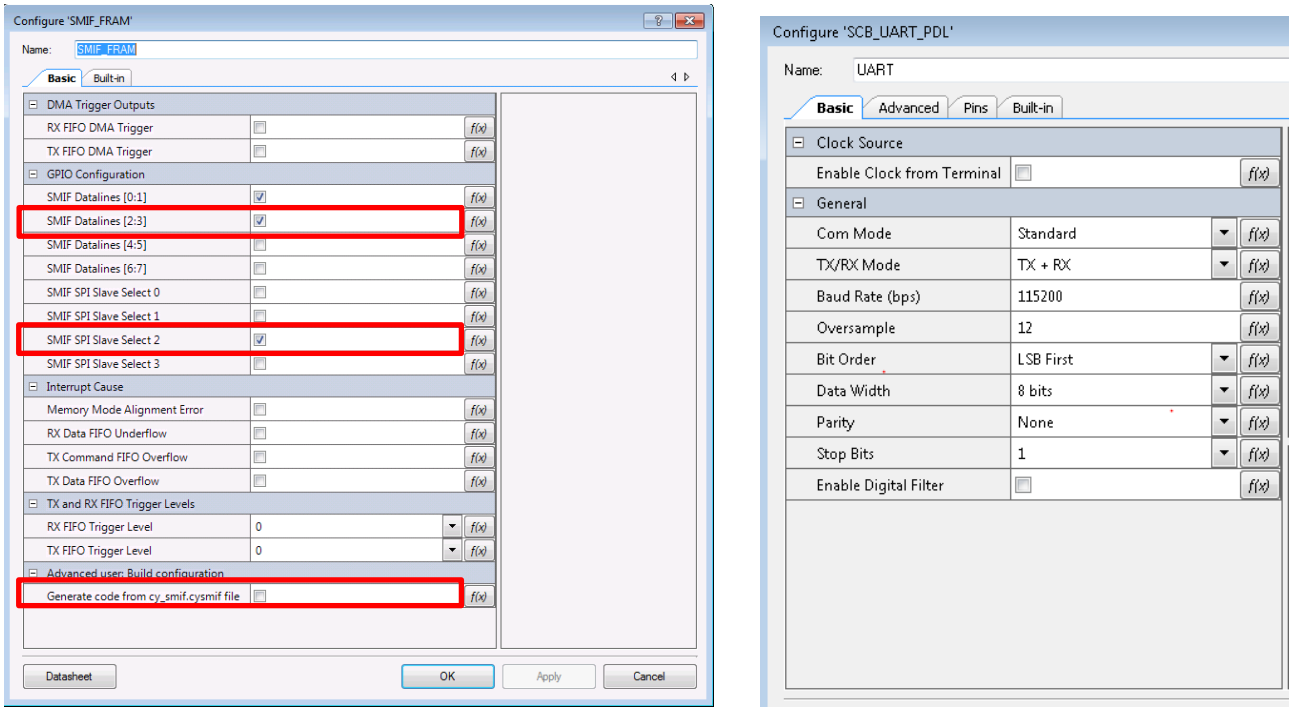
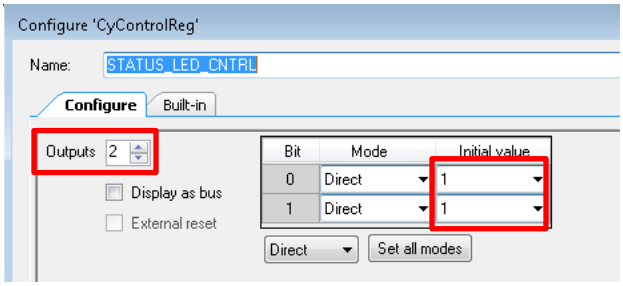


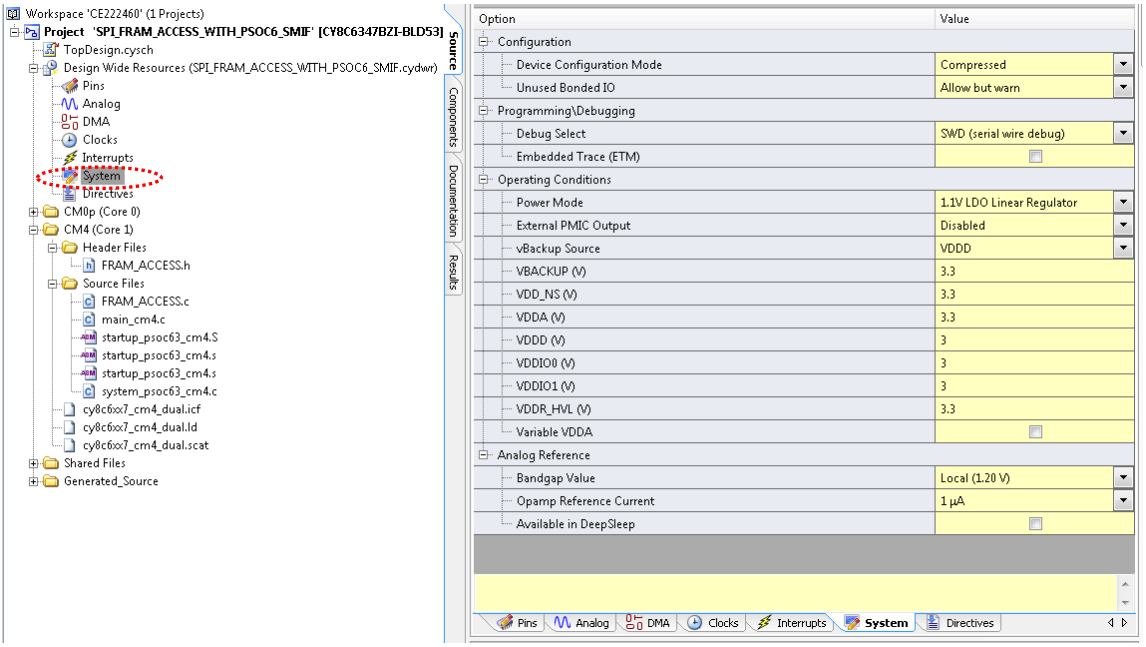
Figure 8. Settings for Control Register (Non-default settings are outlined)



Design-Wide Resources

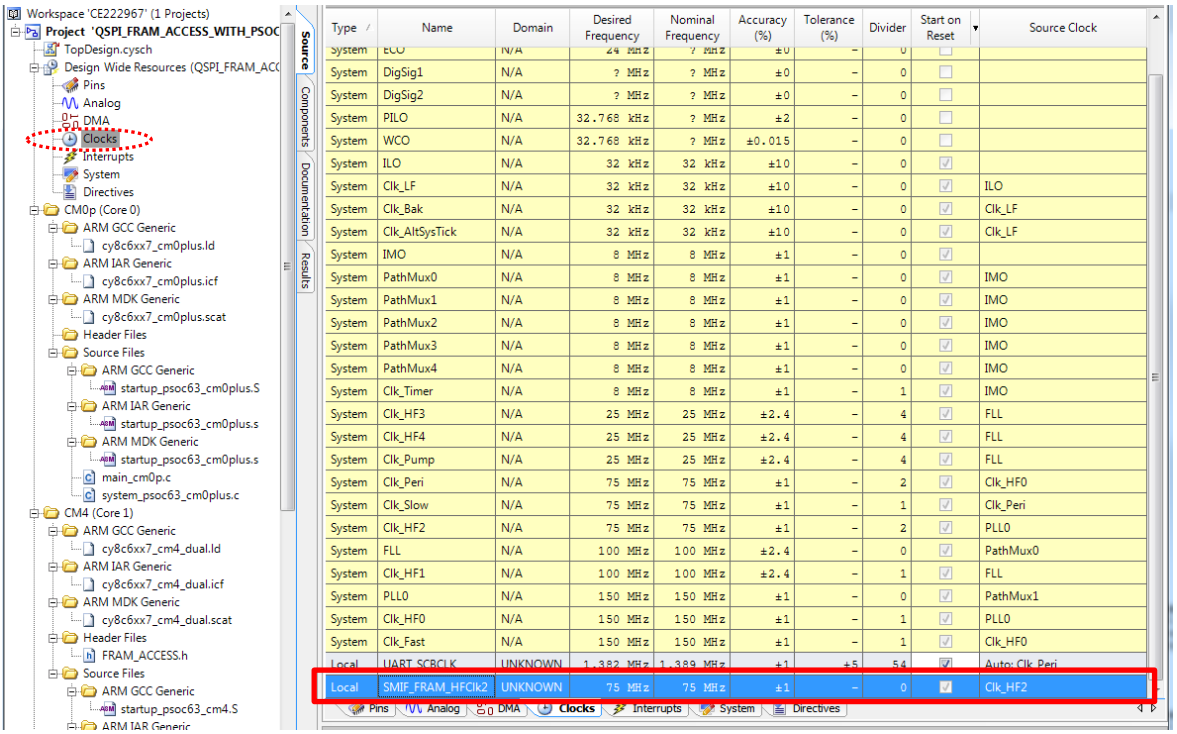
Make sure that V_{DDD} (**PSoC Creator** > **Design Wide Resources** > **System** tab) is set to 2.7 V or above, as shown in Figure 9, to drive the STATUS_LED. Also, make sure that the PSoC 6 MCU I/O voltage is set correctly to match the QSPI F-RAM operating range (V_{DD}/ V_{CC}).

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



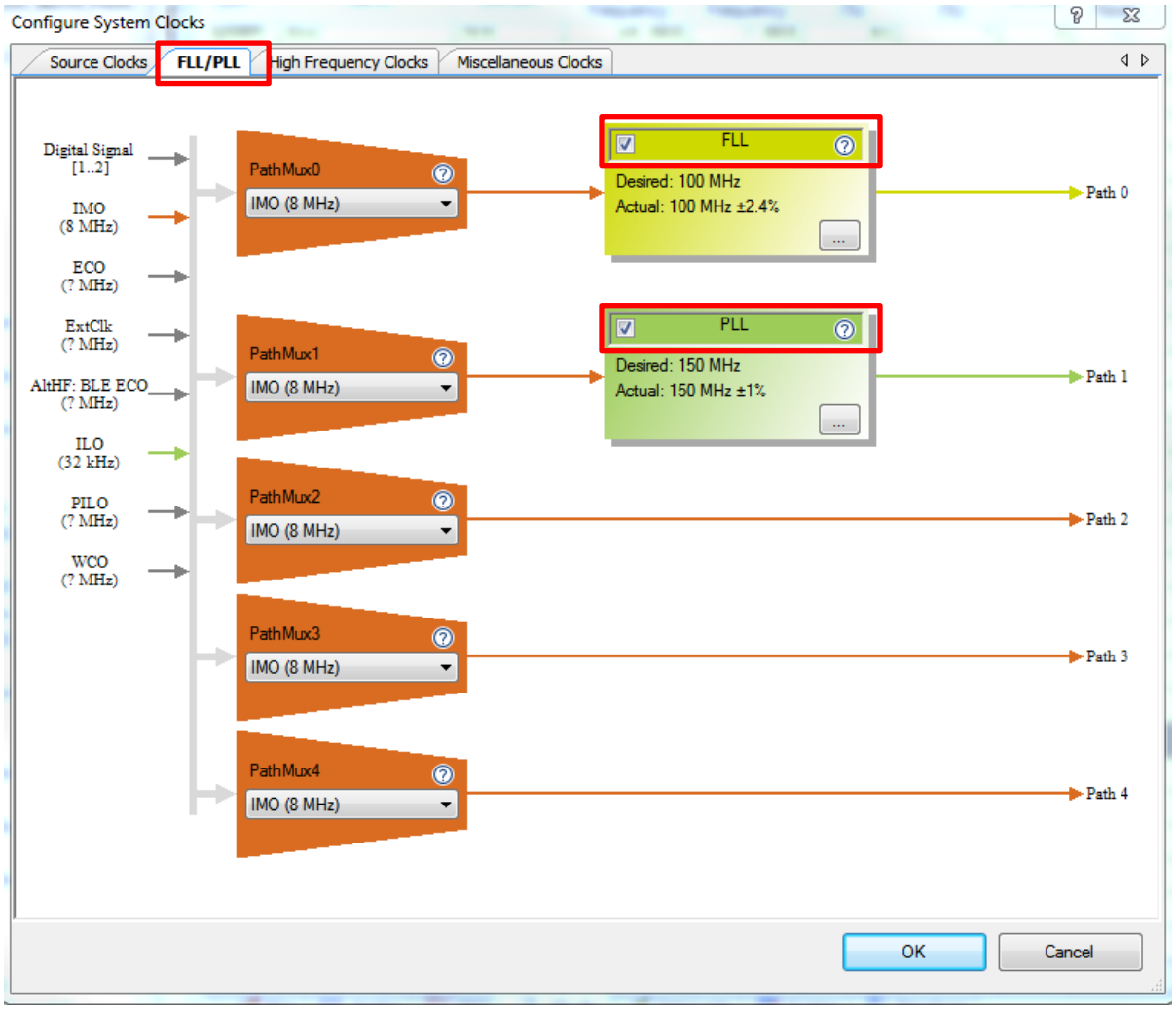
Make sure that the SMIF clock frequency is set to 80 MHz or below. This code example sets the SMIF_FRAM clock (SMIF_FRAM_HFClk2) to 75 MHz, as shown via Figure 10 and Figure 12. Go to **PSoC Creator** > **Design Wide Resources** and click **Clocks**.

Figure 10. SMIF_FRAM_HFClk2 Setting using Design Wide Resources – Step 1



Double-click anywhere in the type “system”, highlighted in yellow; for example, **Clk_HF0** or **Clk_Fast** row in Figure 10. A new **Configure System Clock** window opens. Select the **FLL/PLL** tab and check the FLL and PLL clock options with clock frequencies set as 100 MHz for the FLL and 150 MHz for the PLL, as shows in Figure 12.

Figure 11. FLL/PLL Clock Setting – Step 2



Select the **High Frequency Clocks** tab and select the appropriate clock path and clock divider from their drop-down options, as highlighted Figure 12, to achieve a frequency of 75 MHz. Click **OK**. You can use FLL/PLL tab to configure other frequency options for Path 0. The SMIF block currently supports maximum clock frequency of up to 80 MHz. See the PSoC 6 MCU: PSoC 63 with BLE Datasheet for more details on SMIF block features.

Figure 12. SMIF_FRAM_HFClk2 Clock Frequency Setting – Step 3

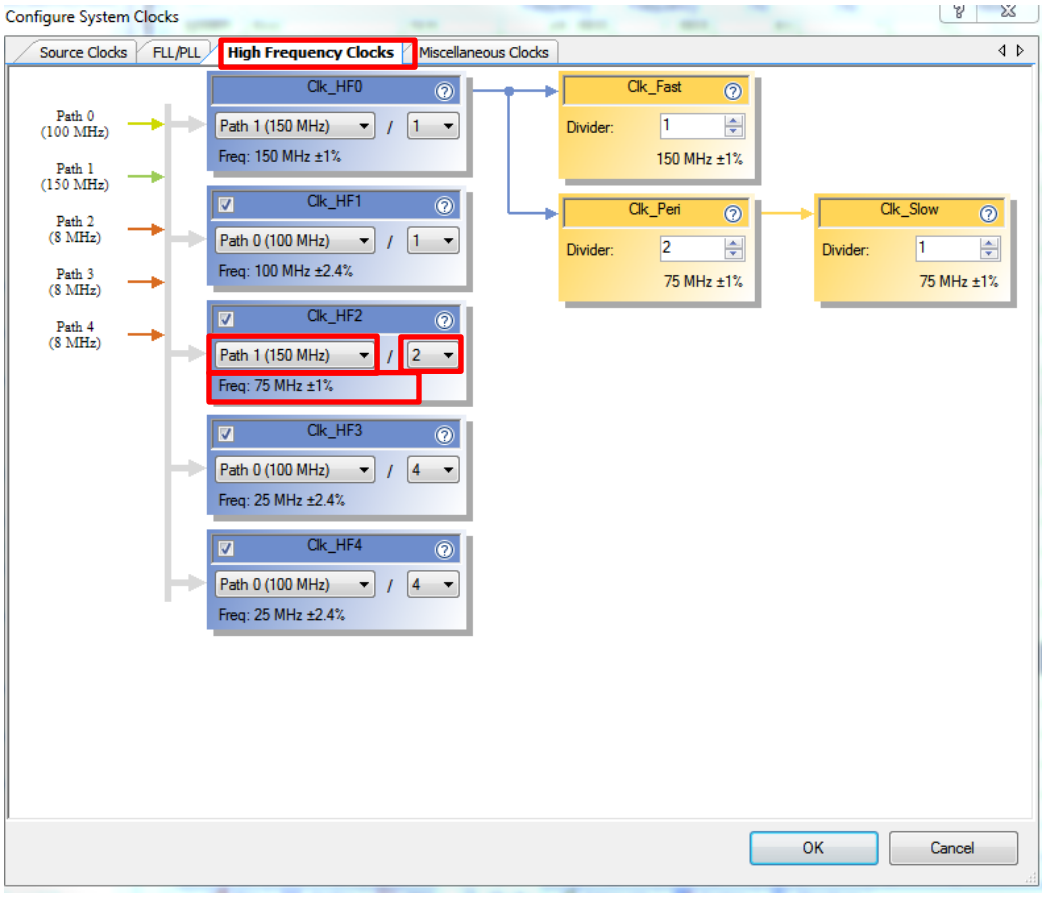
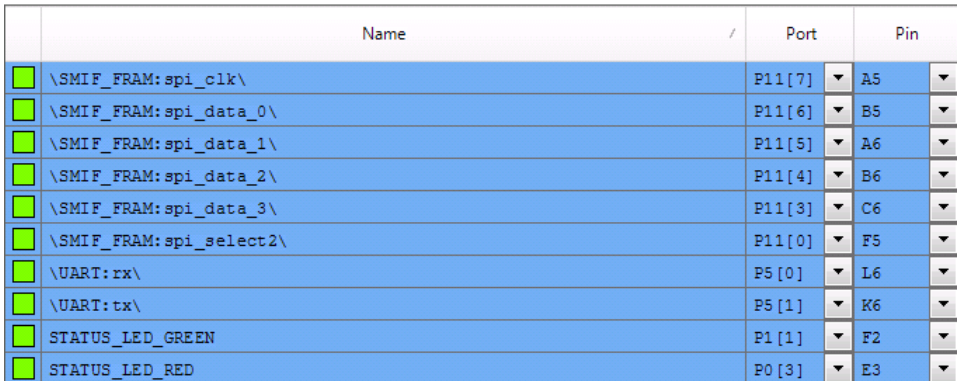


Figure 13 shows the pin assignment for the code example.

Figure 13. 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_data_2\	P11[4]	B6
	\SMIF_FRAM:spi_data_3\	P11[3]	C6
	\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

Project Folder and Files Details

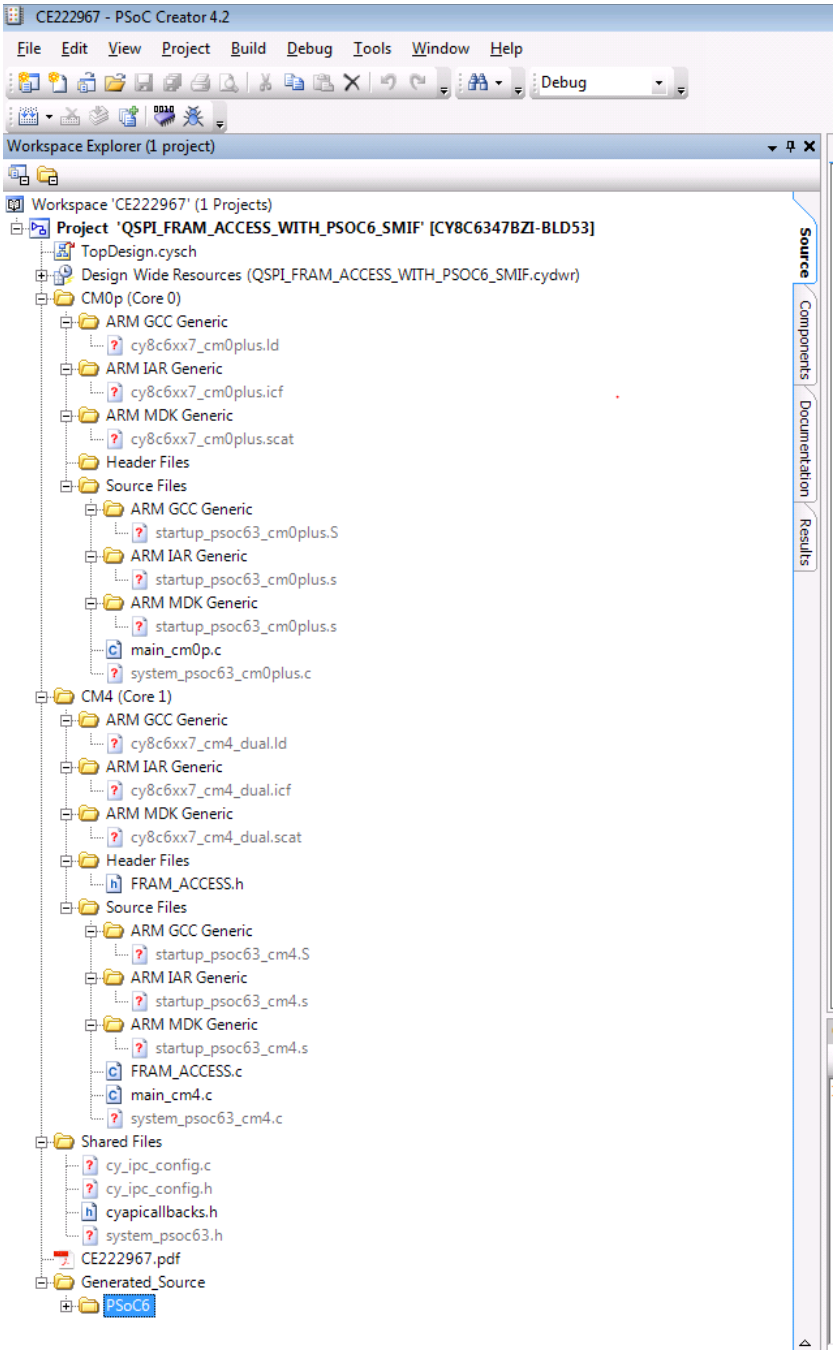
Figure 14 shows the project folder and files structure. The project files mark with '?' are the files missing in the code example provided. These files are automatically compiled and added to the project after a successful build.

- *FRAM_ACCESS.h* – This file declares the function prototype and constants for the QSPI F-RAM access.
- *FRAM_ACCESS.c* – This file defines user functions and APIs for the QSPI F-RAM access. Functions and APIs defined in this file use low-level driver, defined in the smif driver file *cy_smif.c*. The *cy_smif.c* file is in the workspace:

Project (QSPI_FRAM_ACCESS_WITH_PSOC6_SMIF) > **Generated_Source** > **PSoC6** > **pdl > drivers** >**peripheral**> **smif**

main_cm4.c – This is the *main.c* file where a few APIs are called to demonstrate accessing the QSPI F-RAM using SMIF.

Figure 14. Folder and File Structure for Code Example



Reusing This Example

This example is designed for the CY8CKIT-062-WiFi-BT Pioneer Kit with serial F-RAM mounted on it. To port the design to a different PSoC 6 MCU device, kit, or both, 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.
Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	This application note helps you explore the PSoC 6 MCU with BLE architecture and development tools and shows you how to create your first project using PSoC Creator, export the project to a third-party integrated development environment (IDE), and continue your firmware development.
PSoC Creator Component Datasheets	
UART	UART communications interface
Serial Memory 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
QSPI F-RAM (CY15B104QSN) Datasheet	1.8 V to 3.6 V (3 V typical), 108 MHz QSPI FRAM datasheet
Development Kit (DVK) Documentation	
PSoC 6 WiFi-BT Pioneer Kit (CY8CKIT-062-WiFi-BT)	

Document History

Document Title: CE222967 – Excelon-Ultra QSPI F-RAM Access Using PSoC 6 MCU SMIF

Document Number: 002-22967

Revision	ECN	Submission Date	Description of Change
**	6073046	02/28/2018	New Code Example
*A	6282523	8/15/2018	1. Updated the “PowerUpMemoryDefaultSPI” function in main.c using the “write any register (WRAR)” opcode to write to configuration registers 2. Updated the copyright notice in the main.c, FRAM_ACCESS.c, and FRAM_ACCESS.h files 3. Updated CONFIG_REG2_ADDR initialization in Global variables and functions section in main.c. Changed the configuration register 2 (CR2) address to volatile register address 0x700003
*B	6629848	07/15/2019	Fixed the broken hyperlink for the PDL, SMIF, and QSPI FRAM product page link Updated the Related Hardware to PSoC 6 WiFi-BT Pioneer Kit

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, 2018-2019. 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.