CE227032 – PSoC 6 MCU SMIF On-The-Fly Encryption in MMIO and XIP Modes

Objective

This example demonstrates on-the-fly encryption of data using Serial Memory Interface(SMIF) in MMIO and XIP modes. This example uses the CY15B104QSN Excelon™-Ultra 4-Mbit (512K × 8) Quad SPI F-RAM™ in SPI mode.

Requirements

Tool: ModusToolbox™ 1.1

Programming Language: C

Associated Parts: All PSoC® 6 MCU parts

Related Hardware: PSoC 6 WiFi-BT Pioneer Kit

Overview

This example demonstrates on-the-fly encryption of data using Serial Memory Interface(SMIF) in MMIO and XIP modes. This example uses the CY15B104QSN Excelon™-Ultra 4-Mbit (512K × 8) Quad SPI F-RAM in SPI mode that is available on the PSoC 6 WiFi-BT Pioneer Kit.

Plain data is encrypted in firmware and written to F-RAM in MMIO mode. The encrypted data is then read back, decrypted and compared with plain data for integrity. The firmware then shifts the SMIF operation to XIP mode. In this mode, plain data is directly written to the F-RAM. Encryption is performed on-the-fly by the SMIF block. Data is read back directly and compared with the written data for integrity check; decryption of data is performed on-the-fly by the SMIF block.

Hardware Setup

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

Note: The PSoC 6 WiFi-BT Pioneer kit ships 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 If you do not upgrade, you will see an error like "unable to find CMSIS-DAP device" or "KitProg firmware is out of date".

Software Setup

This example uses Tera Term as the terminal emulator program. Install one on your PC if you don't have one.

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 KitProg3 COM port. Set the serial port parameters to 8N1 and 115,200 baud.
3. Import the application into a new workspace. See KBA225201.
4. Program the PSoC 6 MCU device. Select the 'mainapp' project. In the **Quick Panel**, scroll down, and click **Program (Kitprog3)**.
5. After programming, the application starts automatically. Confirm that verbose messages regarding firmware operation are displayed on the UART terminal. Figure 1 is a snapshot of the serial terminal output.

Figure 1. Serial Terminal Output

Debugging

You can debug the example to step through the code. Use Debug (KitProg3) configuration in the Quick Panel. See KBA224621 in the Cypress community to learn how to start a debug session with ModusToolbox IDE.

Design and Implementation

Encryption and decryption of data in the SMIF block are based on the AES-128 forward block cipher. The secret private key is programmed into the CRYPTO_KEYx registers of the block. In the crypto block of SMIF, by applying AES-128 with 128-bit secret key to 128-bit plain text, you get 128-bit cipher text. Note that the AES-128 block cipher is on the address of the data and not on the data itself.

When using MMIO mode, plain text is encrypted by firmware. This is done by providing the address of the memory as plain text to the crypto block and then the data is XORed with the resulting cipher text. The encrypted data is then written to F-RAM. The data from F-RAM is similarly decrypted. See thecode snippet below. Data is encrypted and decrypted using the `Cy_SMIF_Encrypt` API function.

```
    PrintArray("[Info] Plain Data:\t", txBuffer, PACKET_SIZE);
    status = Cy_SMIF_Encrypt(KIT_QSPI_HW, CY15B104QSN_SlaveSlot_2.baseAddress, encryptedtxBuffer,
PACKET_SIZE,  &KIT_QSPI_context);

    /* Write the sample bytes in encryptedtxBuffer to the external memory address specified */
    WriteMemory(KIT_QSPI_HW, (cy_stc_smif_mem_config_t*) smifMemConfigs[0], &KIT_QSPI_context,
encryptedtxBuffer, PACKET_SIZE, extMemAddress);

    /* Read data from external memory into rxBuffer and decrypt it*/
    ReadMemory(KIT_QSPI_HW, (cy_stc_smif_mem_config_t*) smifMemConfigs[0], &KIT_QSPI_context, rxBuffer,
PACKET_SIZE, extMemAddress);
    status = Cy_SMIF_Encrypt(KIT_QSPI_HW, CY15B104QSN_SlaveSlot_2.baseAddress, rxBuffer, PACKET_SIZE,
&KIT_QSPI_context);
```

In XIP mode, the SMIF crypto block supports on-the-fly encryption and decryption. The address accessed using XIP mode is used as plain text; the resulting cipher text is then used to write to external F-RAM on-the-fly and is not software accessible. In XIP mode, the resulting ciphertext (of the encrypted address) is XORed with the memory transfer's read data or write data.

For more information on encryption in SMIF block, see the device TRM.

The Serial Communication Block (SCB) resource is configured as a UART at 115200 baud, 8N1. It is connected to a clock operating at 923 kHz to generate the correct baud rate. The RX is on pin P5[0] and TX is on P5[1] to match the pin usage on the kit.

The resources are configured and application code runs on the CM4 CPU.

To see all the settings, review the *design.modus* file in the application.

Resources and Settings

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
Quad Serial Memory Interface	KIT_QSPI	Communicates with SPI F-RAM	See Figure 2 See Figure 3 for FRAM configuration
SCB	KIT_UART	Prints messages to a terminal window	See Figure 4
Digital Output Pin	KIT_LED2	Provides visual feedback	See Figure 5
	KIT_UART_TX	Used for UART transmit (Tx)	See Figure 6
Digital Input Pin	KIT_UART_RX	Used for UART receive (Rx)	See Figure 7

Figure 2 to Figure 7 show non-default configuration settings for the resources.

Figure 2. QSPI Configuration

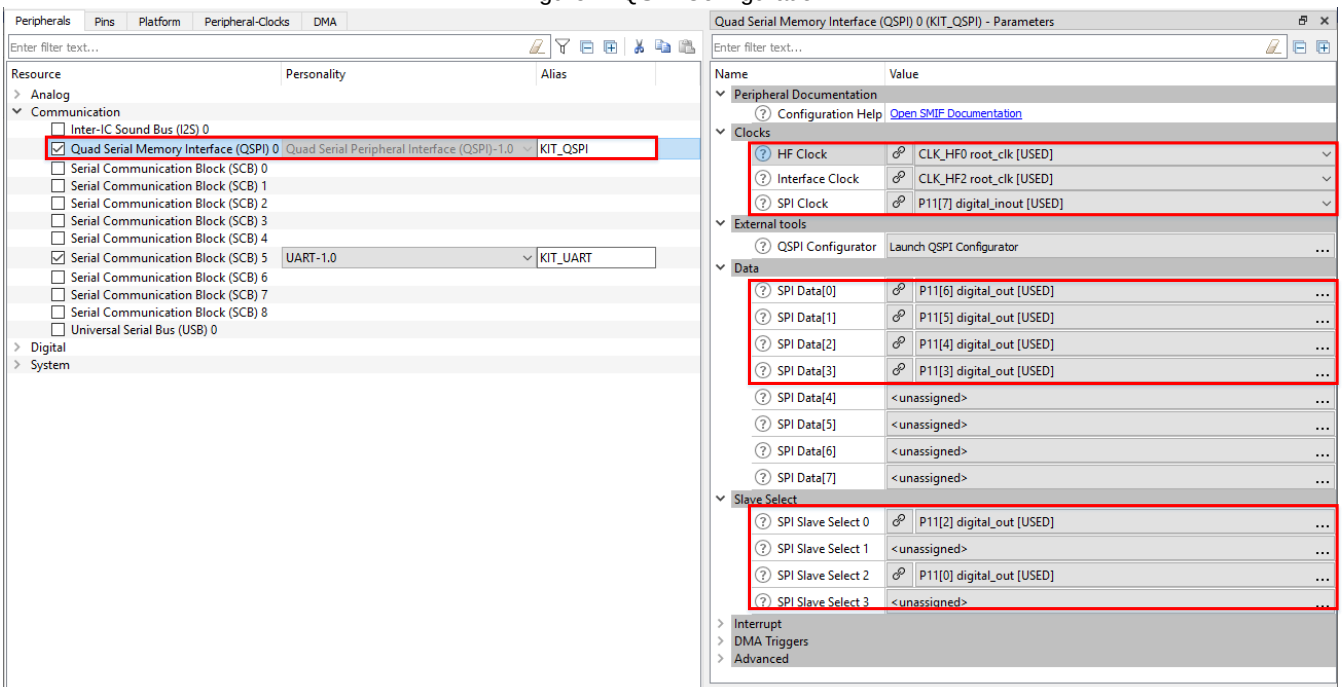
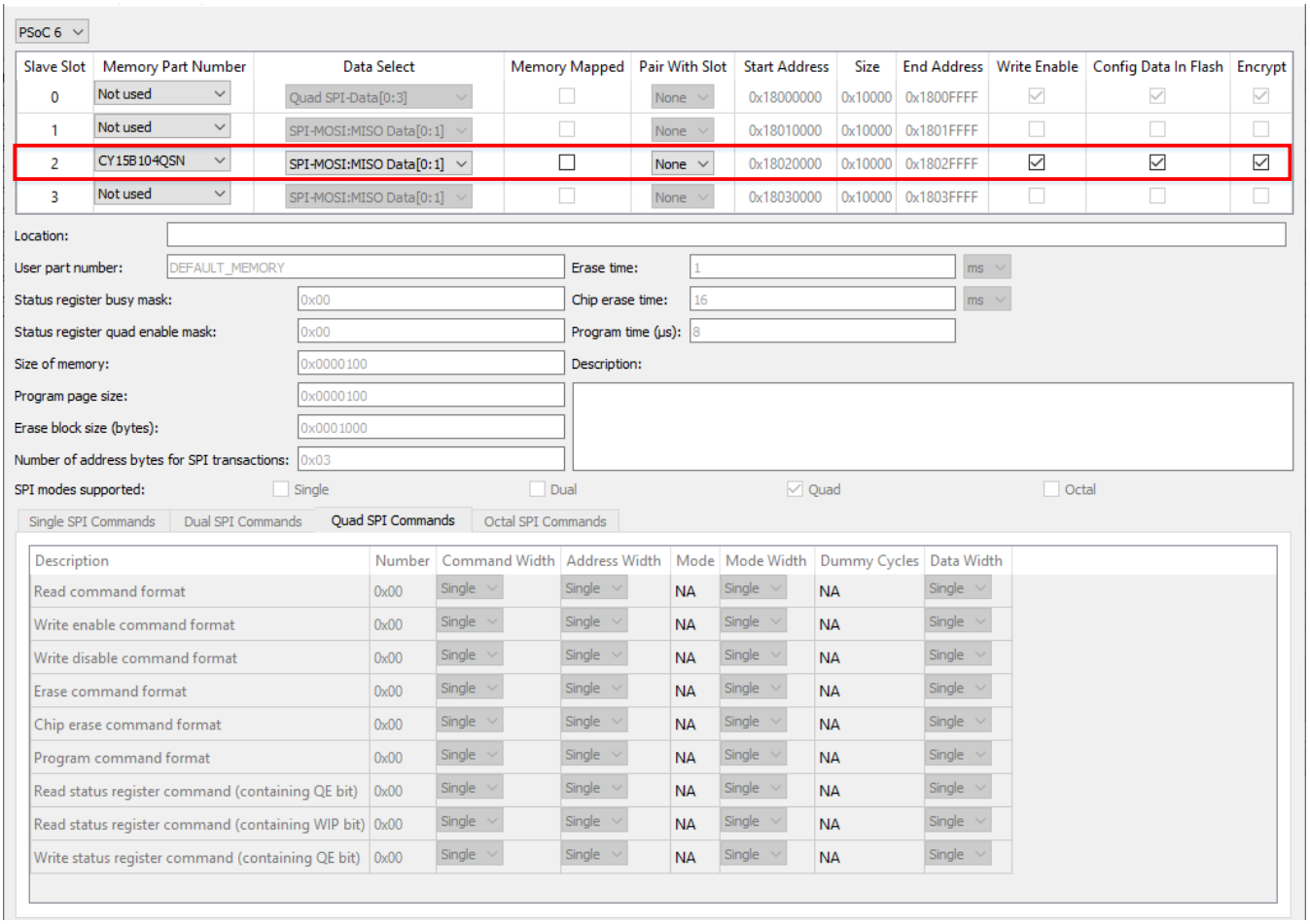


Figure 3. F-RAM Memory Configuration



Note: Memory mapping of F-RAM (operating in SPI mode) to PSoC 6 MCU's internal memory map is not supported by the QSPI configurator in ModusToolbox 1.1. Memory mapping is enabled by setting `CY_SMIF_FLAG_MEMORY_MAPPED` flag in the `CY15B104QSN_SlaveSlot_2.flags` variable. This variable is defined in the generated source file *cycfg_qspi_memslot.c*.

Figure 4. UART Configuration

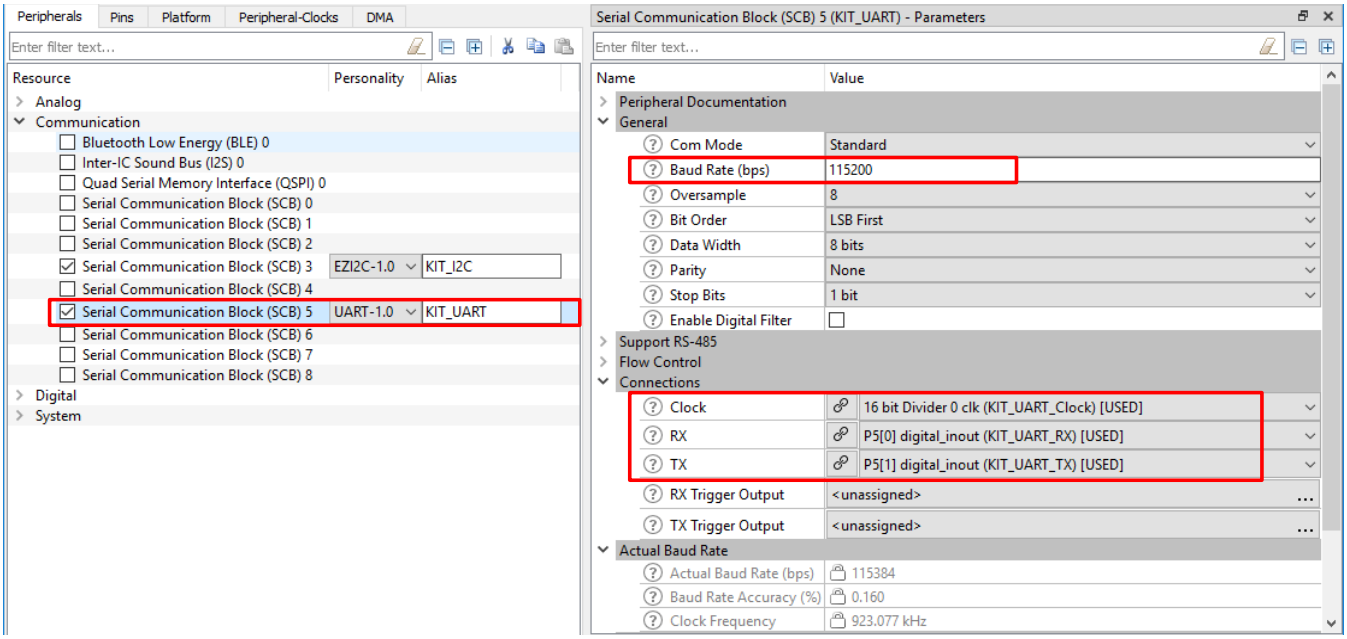


Figure 5. GPIO Pin configuration for LED

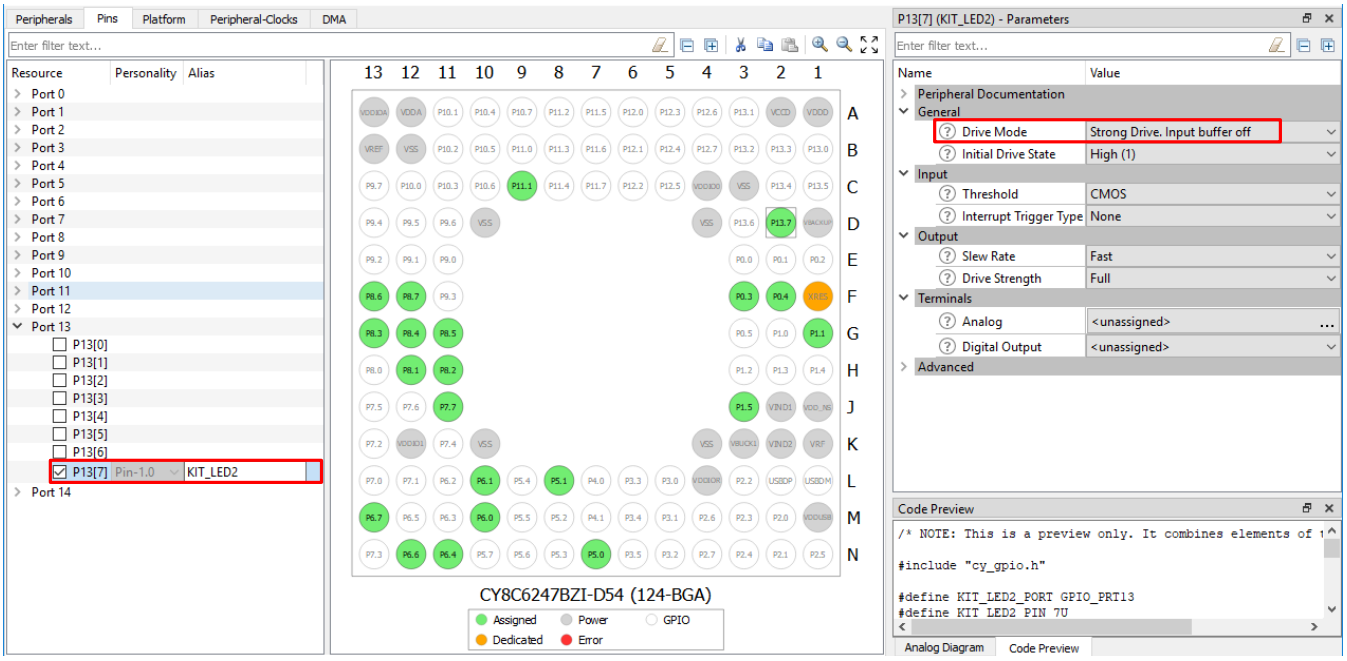


Figure 6. GPIO Pin Configuration for UART Tx

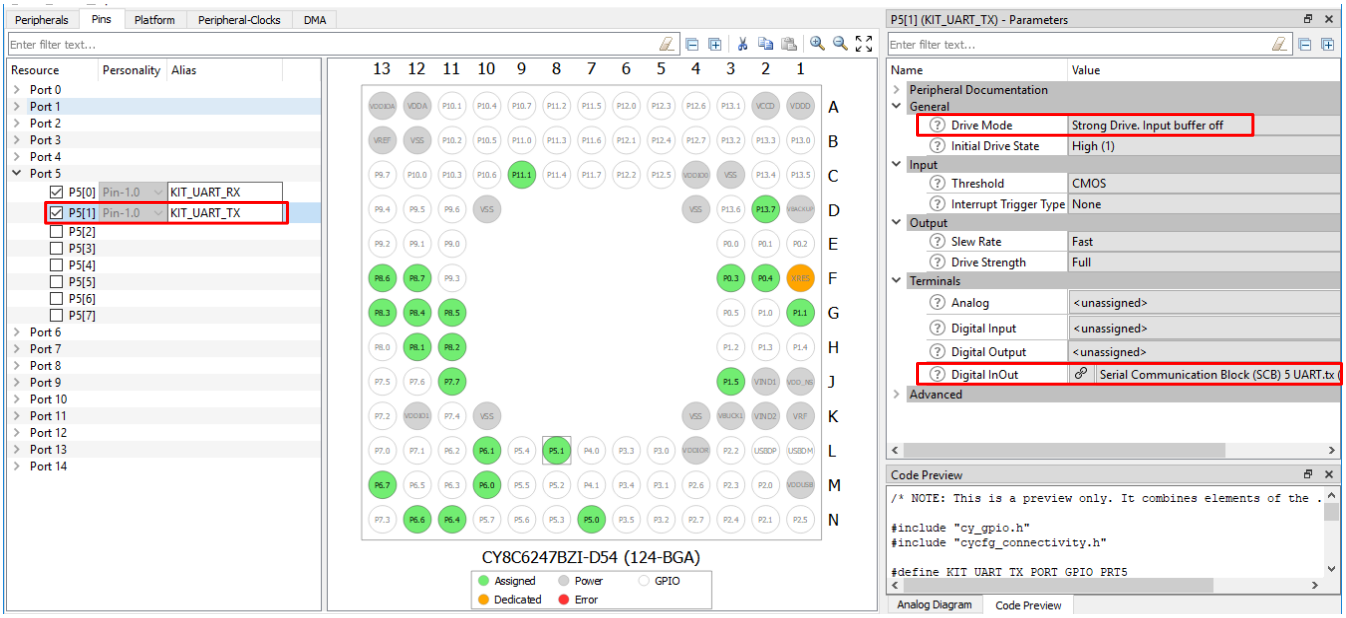


Figure 7. GPIO Pin Configuration for UART Rx

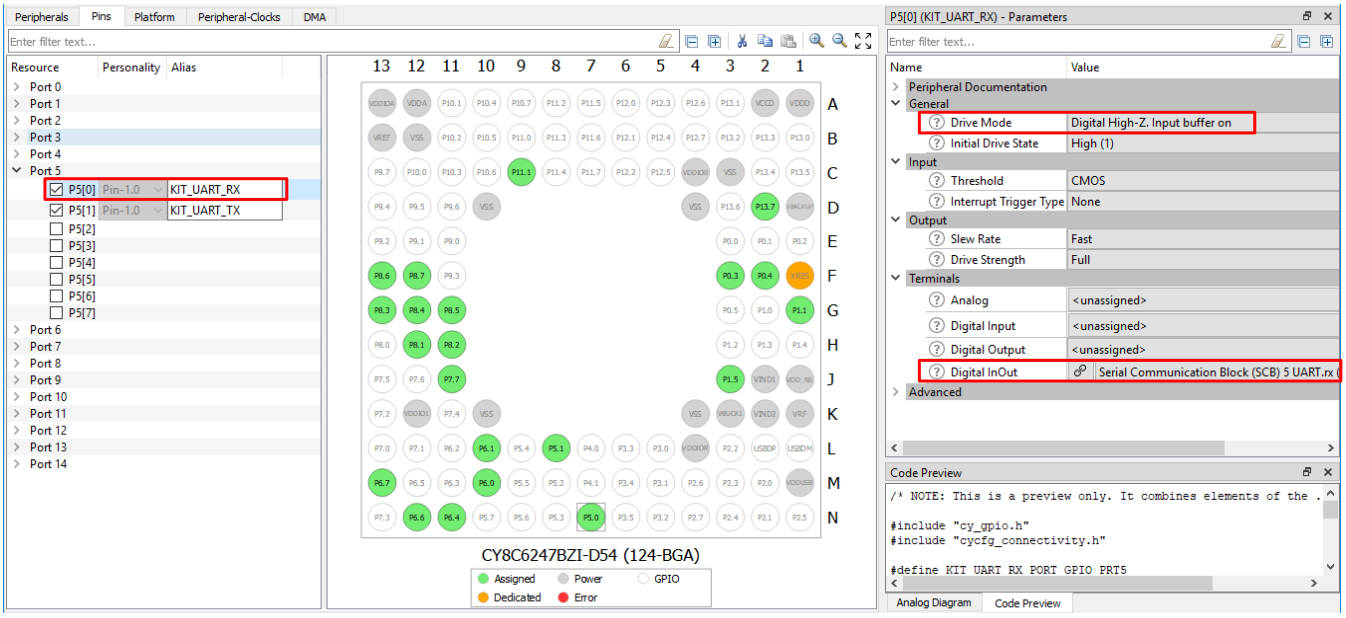


Figure 8 and Figure 9 show the Clock configuration for UART and QSPI resources respectively.

Figure 8. Peripheral-Clock Configuration for UART

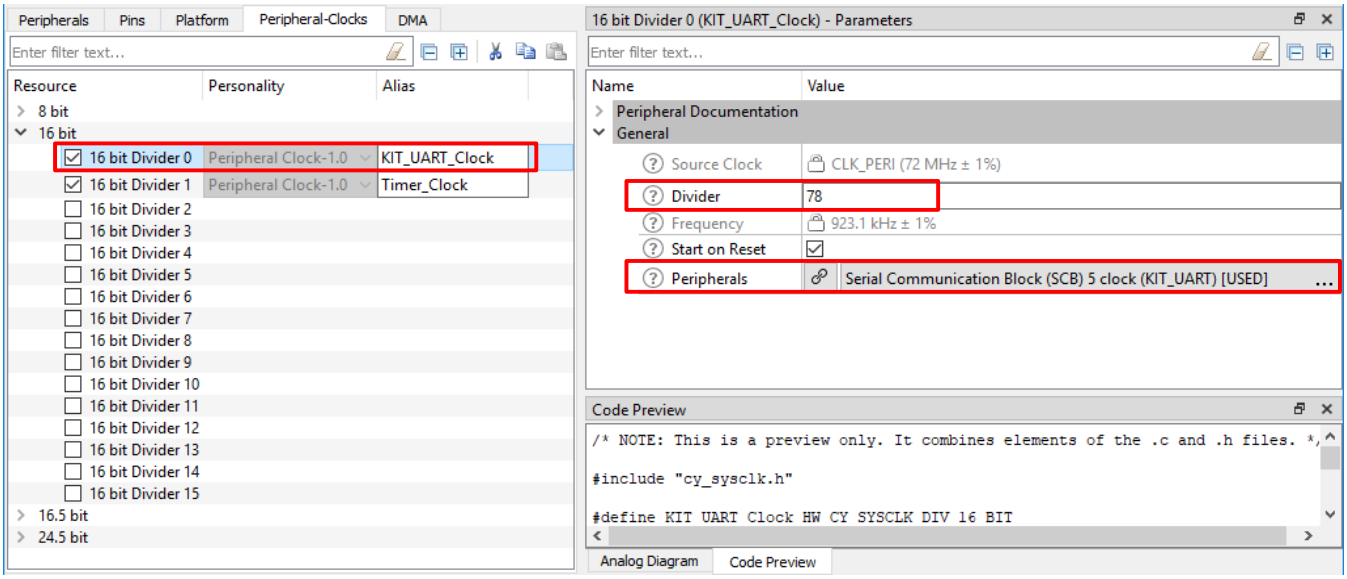
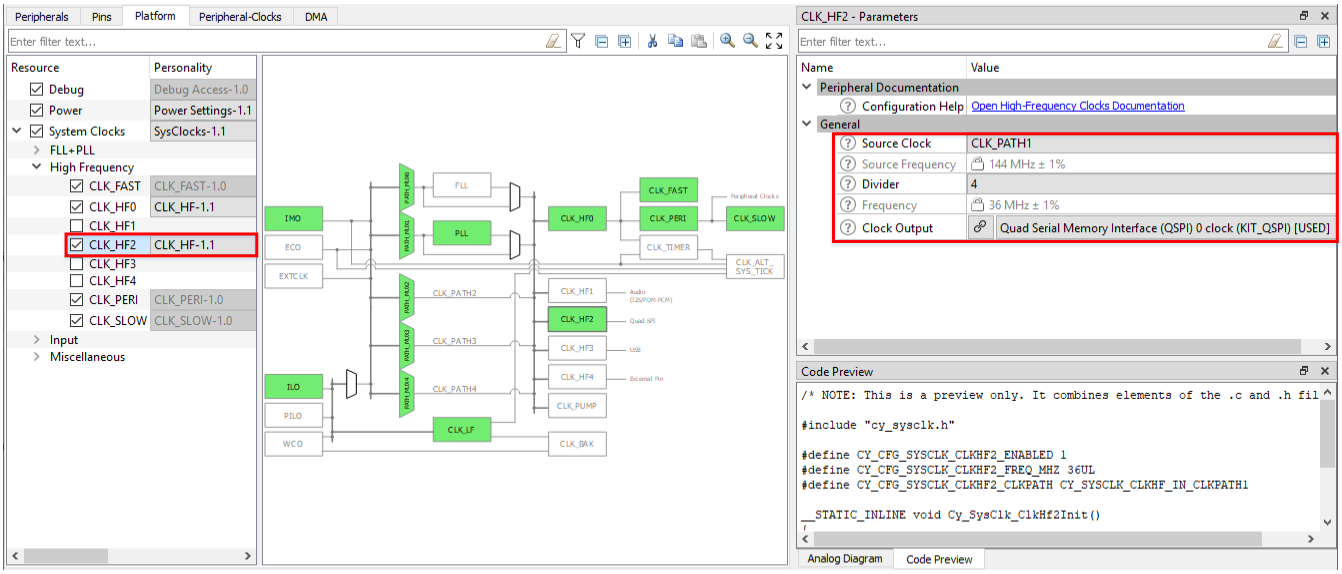


Figure 9. Clock Configuration for SMIF



Related Documents

For a comprehensive list of PSoC 6 MCU resources, see KBA223067 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
AN221774 – Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first ModusToolbox application and PSoC Creator project
Code Examples	
Visit the Cypress GitHub site for a comprehensive collection of code examples using ModusToolbox IDE	
Device Documentation	
PSoC 6 MCU: PSoC 62 Datasheet	PSoC 6 MCU: PSoC 62 Architecture Technical Reference Manual (TRM)
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
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	
Tool Documentation	
ModusToolbox IDE	The Cypress IDE for IoT designers

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.

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: CE227032 – PSoC 6 MCU SMIF On-The-Fly Encryption in MMIO and XIP Modes

Document Number: 002-27032

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6591895	SNVN	06/10/2019	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

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709