CE221120 - PSoC 6 MCU SPI Master

Objective

This example demonstrates the use of SPI SCB (Serial Communication Block) resource for PSoC® 6 MCU in Master mode. Four different applications show the usage of APIs to communicate with an SPI slave, using ModusToolbox™ IDE.

Requirements

Tool: ModusToolbox™ IDE 1.0

Programming Language: C

Associated Parts: All PSoC 6 MCU parts

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

Overview

The SPI is designed to send command packets to control a user LED on the slave. The four different applications developed in this example are: SPI Master using high-level APIs, SPI Master using low-level APIs, SPI Master using User-ISR with low-level APIs, and SPI Master using DMA with low-level APIs.

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. You can also use PSoC 6 BLE Pioneer Kit or PSoC 6 WiFi-Prototyping Kit by modifying the application to use the corresponding device on the board, refer to reusing the example section.

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 If you do not upgrade, you will see an error like "unable to find CMSIS-DAP device" or "KitProg firmware is out of date".

Jumper wires are used to establish connection between the Master and Slave SCBs. P6[0] is connected to P9[0], P6[1] is connected to P9[1], P6[2] is connected to P9[2], and P6[3] is connected to P9[4].

Operation

1. Connect the Pioneer board to your PC using the provided USB cable through the USB connector.
2. Import the application into a new workspace. See KBA225201.
3. Build the application. Choose **Project** > **Build All**.
4. Program the PSoC 6 MCU device. Select the **mainapp** project. In the **QuickPanel**, scroll down, and click **Program Kitprog3**.
5. Observe the KIT_LED2 blink with an interval of 1 second.

Debugging

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

Design and Implementation

In all three applications, the Arm Cortex®-M4 (CM4) MCU controls both the Master and Slave SCB. Master sends command packets to control the user LED (KIT_LED2).

Master APIs are divided into two categories: Master High-Level and Master Low-Level. See the SDK documentation to know more about high-level and low-level functions. The Master sends command packets to the Slave every one second.

The “LowLevelUserIsrSPIMaster” CE is implemented using low-level API. This code example demonstrates a user defined ISR which is triggered when the SPI transfer is complete. This ISR can be used to do any post transfer operation or to check the status of the transfer.

The “LowLevelDMASPIMaster” CE is implemented using low-level API, where DMA is used to transfer command data from the SRAM to the SPI FIFO. The DMA transfers 12 bytes of command data from the SRAM to the SPI FIFO instead of the CPU performing this operation. Once the data is loaded on the FIFO, the command is transmitted.

Resources

Table 1 lists the resources used in this example, and how they are used in the design.

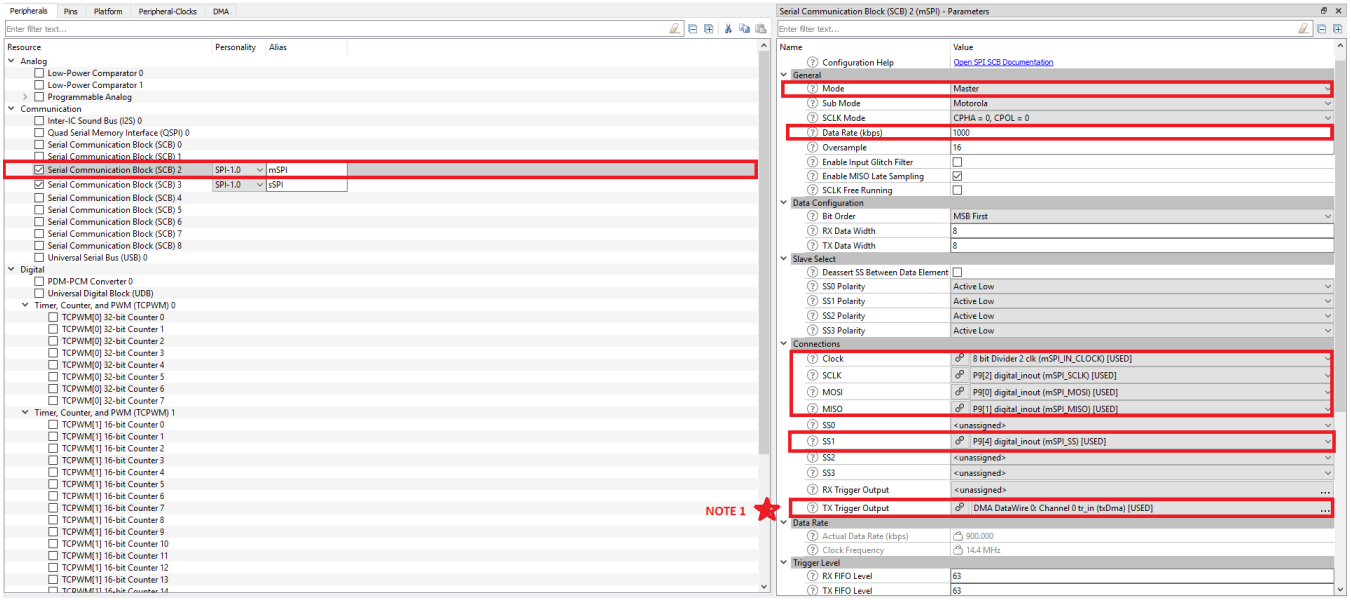
Table 1. ModusToolbox Resources

Resource	Alias	Purpose	Non – Default Settings
SCB2, SCB3	mSPI, sSPI	Two SCB peripheral blocks	Figure 1, Figure 2
GPIO	KIT_LED2	KIT_LED2	Figure 3
DMA Channel	txDma	Write data into SCB buffers	Figure 4

Parameter Settings

Non-default settings for each Resource are outlined in red in the following figures.

Figure 1. SPI Master Resource Configuration



Note 1: This option is only selected for Low level DMA SPI application.

Figure 2. SPI Slave Resource Configuration

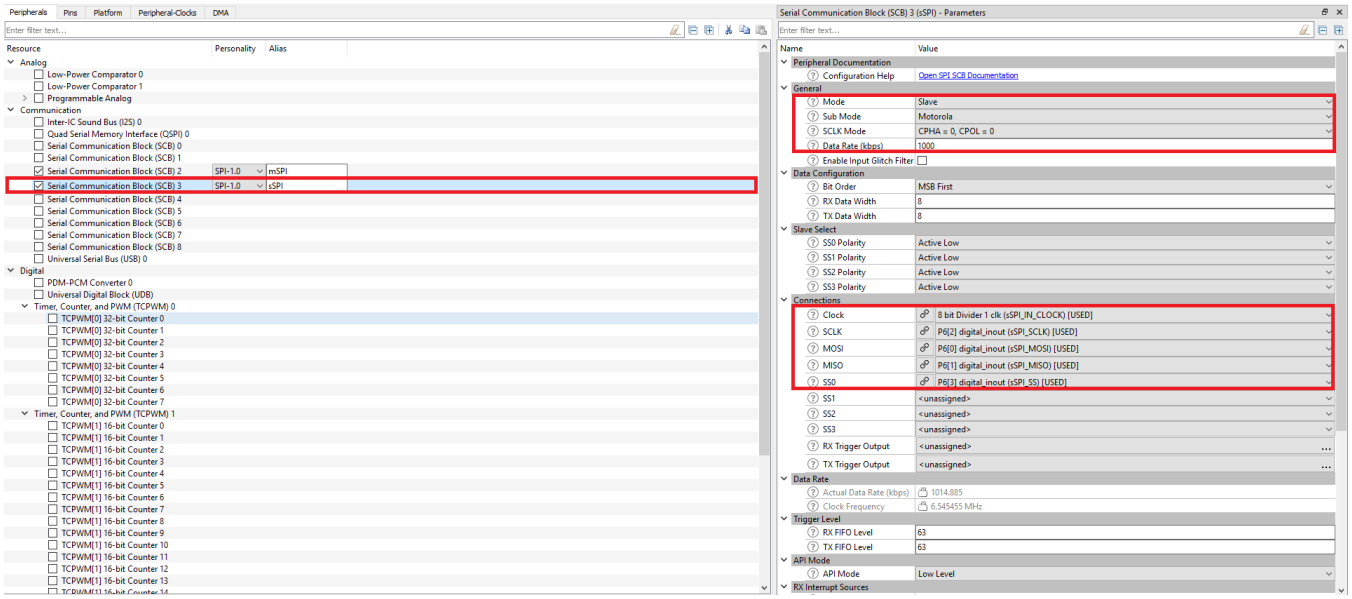


Figure 3. KIT_LED2 Resource Configuration

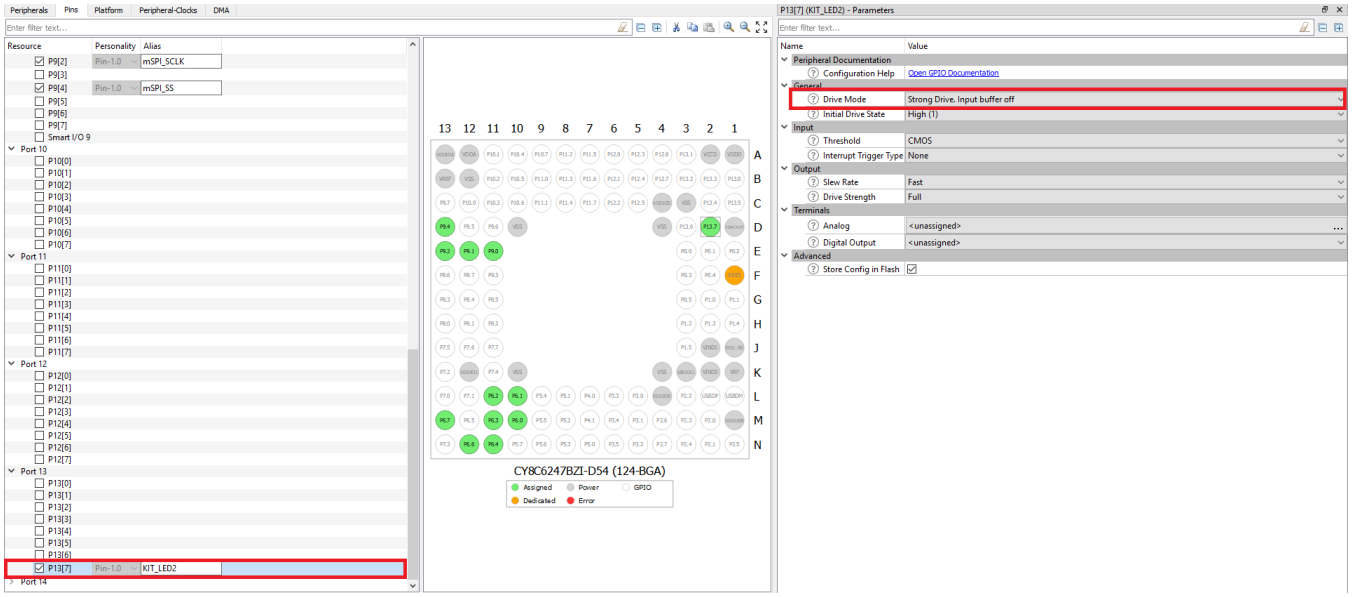
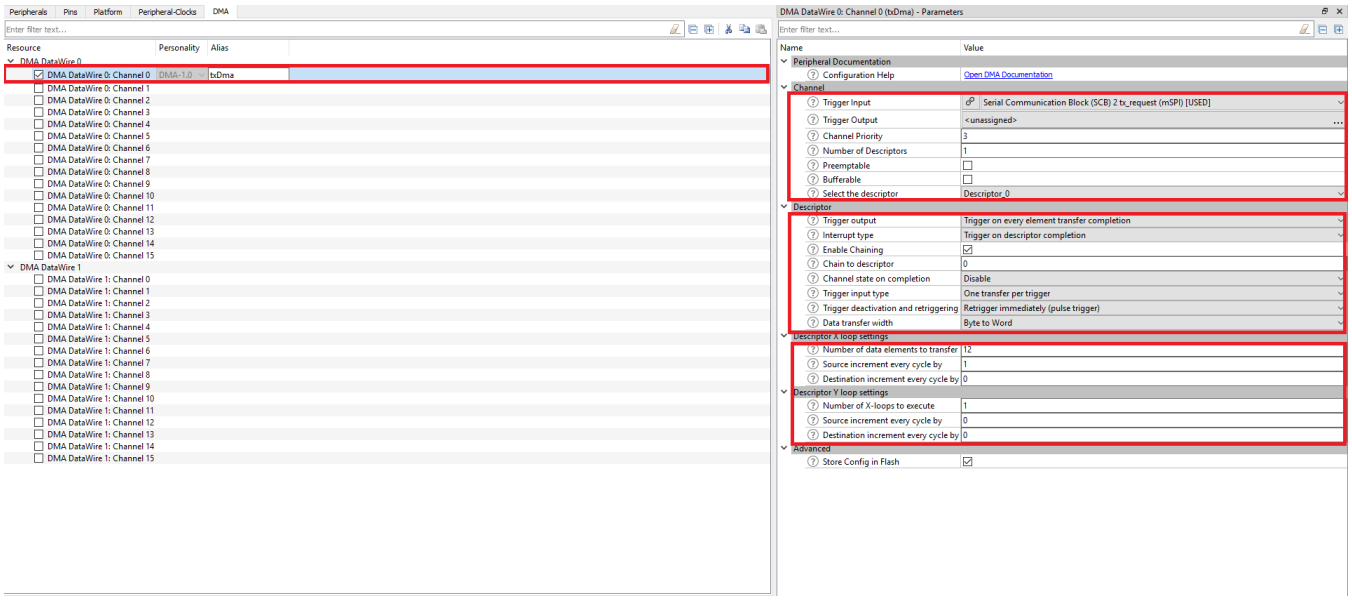


Figure 4. txDma Resource Configuration



Reusing This Example

This example is configured for the PSoC 6 WiFi-BT Pioneer Kit. To port the design to a different PSoC 6 MCU device, create a new project selecting the required kit. If you want to manually edit the same application, right-click the application project and choose **Change Device** (you will need to reassign the DMA channels).

See the Parameter Settings for information on resource configuration.

Table 2: Device-Specific Resource Allocation

Kit name	Device Used	txDma
CY8CKIT-062-WiFi-BT	CY8C6247BZI-D54	DMA DataWire0: Channel 0
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	DMA DataWire0: Channel 0
CY8CPROTO-062-4343W	CY8C624ABZI-D44	DMA DataWire0: Channel 20

Related Documents

For a comprehensive list of PSoC 6 MCU resources, see KBA223067 in the Cypress community.

Application Notes	
AN221774 - Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first ModusToolbox application and PSoC Creator project.
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices.
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.
Code Examples	
CE218472 - PSoC 6 MCU Comparing External Voltages Using a Low-Power Comparator	
Visit the Cypress GitHub site for a comprehensive collection of code examples using ModusToolbox IDE	
Device Documentation	
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	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 PSoC 6 MCU devices, see KBA223067 in the Cypress community for a comprehensive list of PSoC 6 MCU resources.

Document History

Document Title: CE221120 - PSoC 6 MCU SPI Master

Document Number: 002-25577

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6374719	YEKT	11/05/2018	Initial Public Release

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