CE221773 – PSoC 6 MCU: Hello World

Objective

This example demonstrates UART communication and blinks an LED using a TCPWM resource on PSoC® 6 MCU, using ModusToolbox™ IDE.

Requirements

Tool: ModusToolbox™ 1.1

Programming Language: C

Associated Parts: All PSoC 6 MCU parts

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

Overview

This example uses the Arm® Cortex-M4 (CM4) CPU of PSoC 6 MCU to execute two tasks: UART communication and LED control. At device reset, the default Cortex-M0+ (CM0+) application enables the CM4 CPU and configures CM0+ CPU to go to sleep. The CM4 CPU uses a UART resource to print a “Hello World” message in a UART terminal emulator. When the user presses the Enter key, the LED on the kit starts blinking.

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 other PSoC 6 kits by importing the application for that kit.

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 If you do not upgrade, you will see an error like “unable to find CMSIS-DAP device” or “KitProg firmware is out of date”.

Software Setup

This example uses a terminal emulator 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 KitProg3 COM port. Set the serial port parameters to 8N1 and 115,200 baud.
3. Import the application into a new workspace. If you are unsure how to import an application, 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 “Hello World!” is displayed on the UART terminal.
6. Press the **Enter** key. Confirm that the kit LED blinks at an approximate 1-Hz rate.

Debugging

You can debug the example to step through the code. Use **Debug (KitProg3)** configuration in the **Quick Panel**. If you are unfamiliar with how to start a debug session with ModusToolbox IDE, see KBA224621 in the Cypress community.

Design and Implementation

This example configures a TCPWM resource in Timer mode to blink the LED, and a serial communication block (SCB) resource to send a message and read serial input.

The TCPWM resource is connected to a clock operating at 2 kHz, with a compare value of 1000. It generates an interrupt on overflow/terminal count and this interrupt is used to toggle the state of the user LED on the kit. The interrupt configuration is done in the firmware.

The Serial Communication Block 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
Timer Counter (TCPWM)	Timer	Drives the user LED using an interrupt.	See Figure 1
SCB	KIT_UART	Prints a message to a terminal window.	See Figure 2
Digital Output Pin	KIT_LED2	Provides visual feedback.	See Figure 3
	KIT_UART_TX	Used for UART transmit (Tx).	See Figure 4
Digital Input Pin	KIT_UART_RX	Used for UART receive (Rx).	See Figure 5

Figure 1 to Figure 5 show non-default configuration settings for the resources.

Figure 1. Timer Configuration

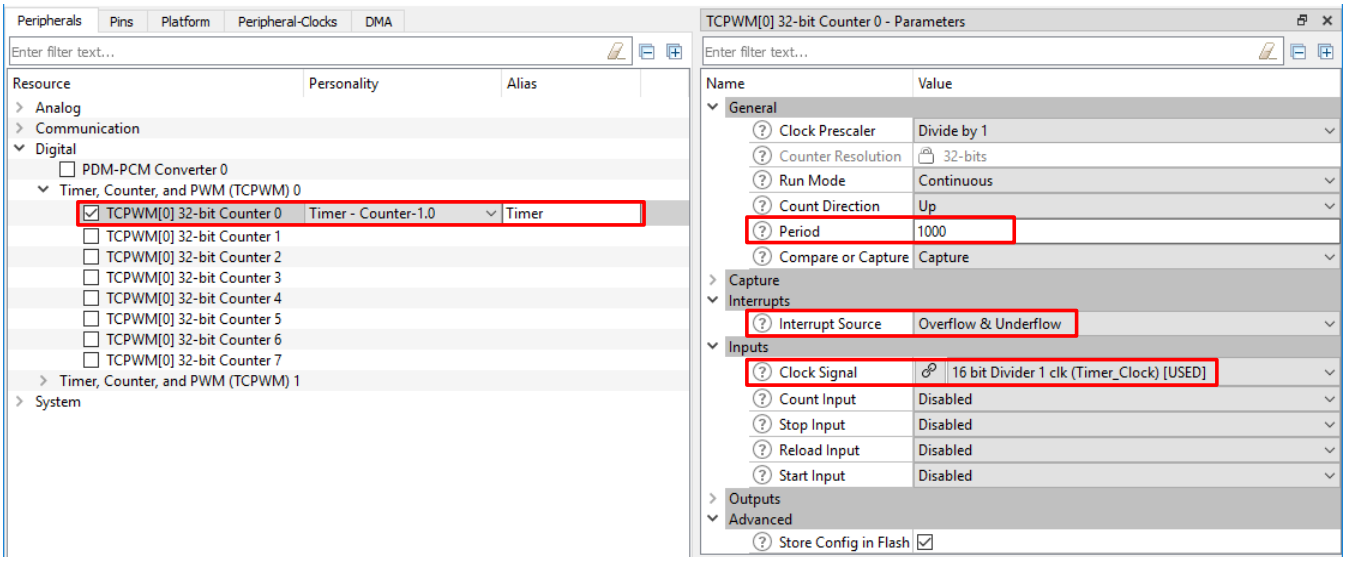

Figure 2. UART Configuration

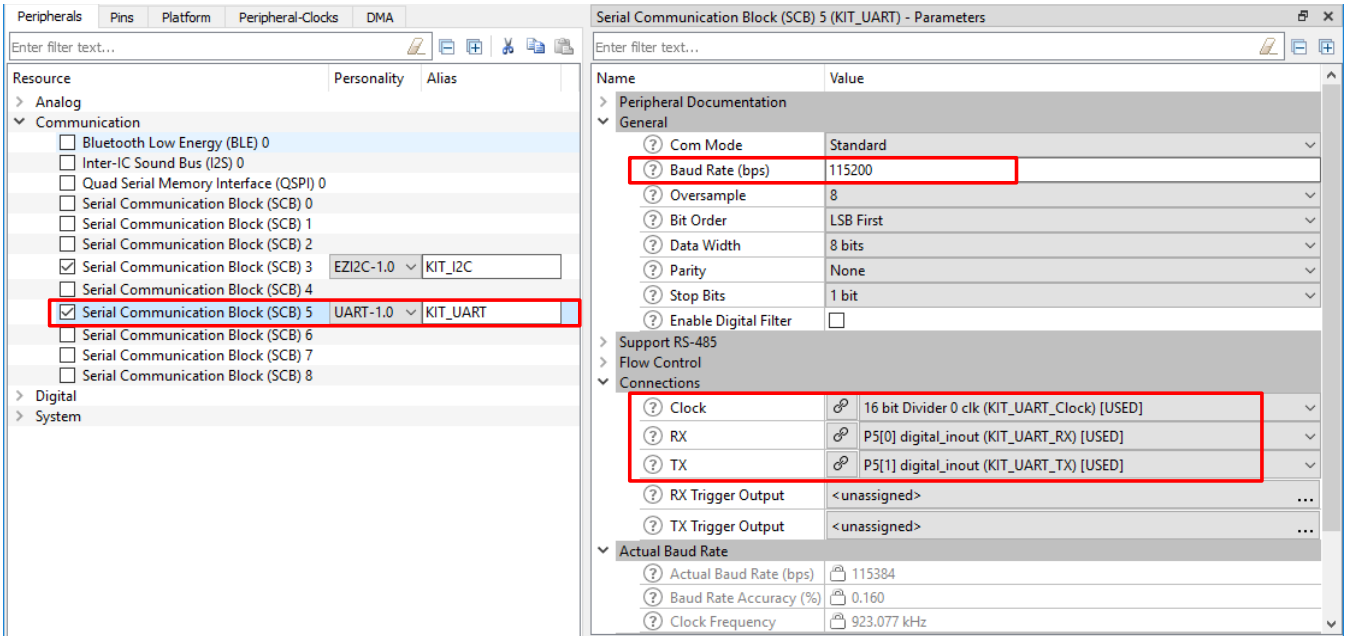

Figure 3. GPIO Pin configuration for LED

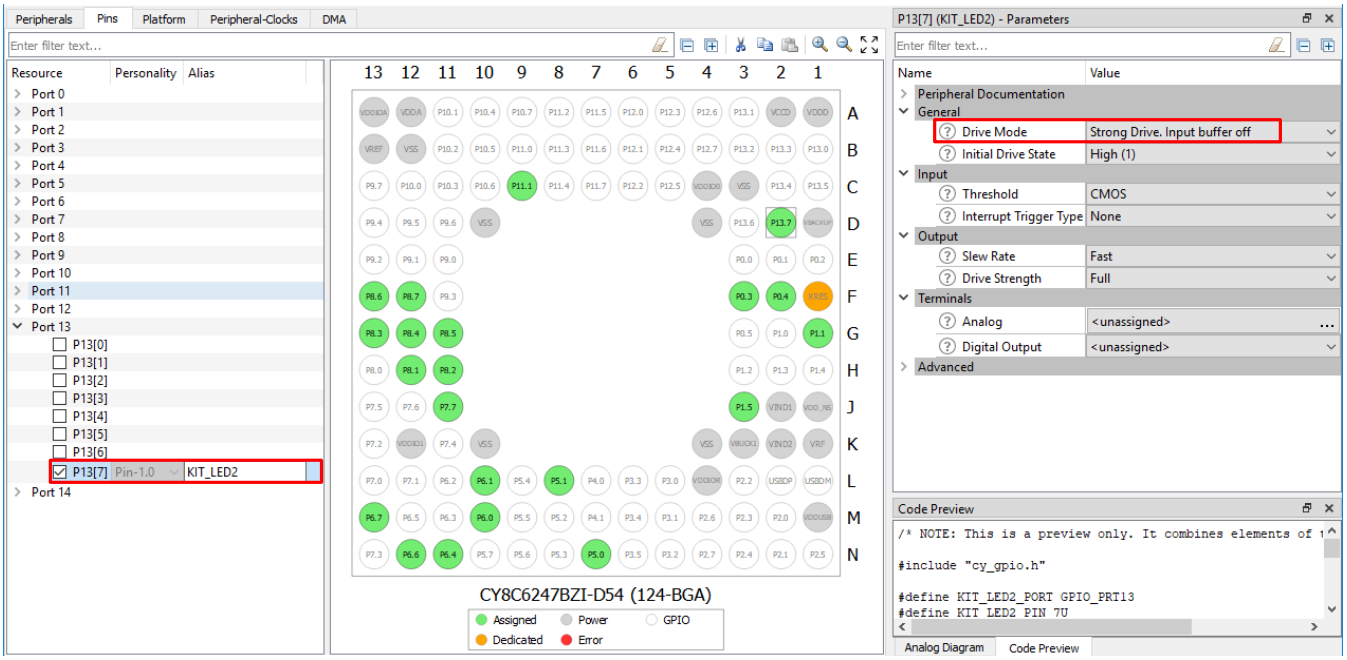

Figure 4. GPIO Pin Configuration for UART Tx

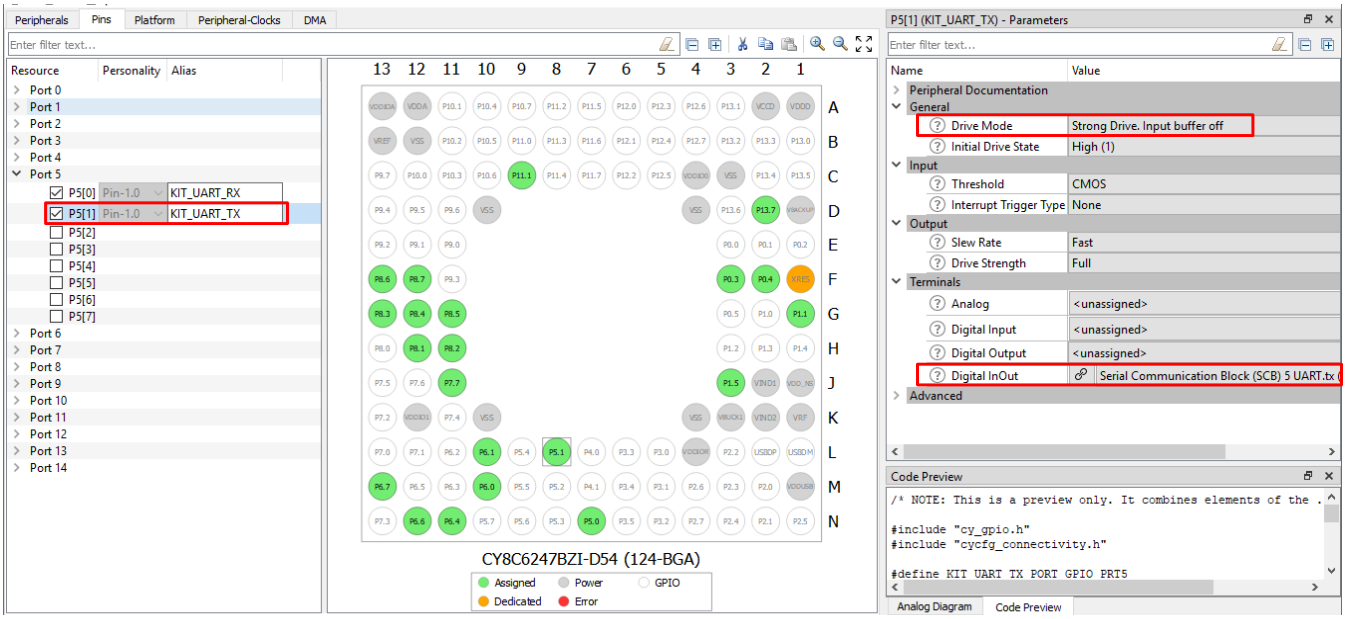

Figure 5. GPIO Pin Configuration for UART Rx

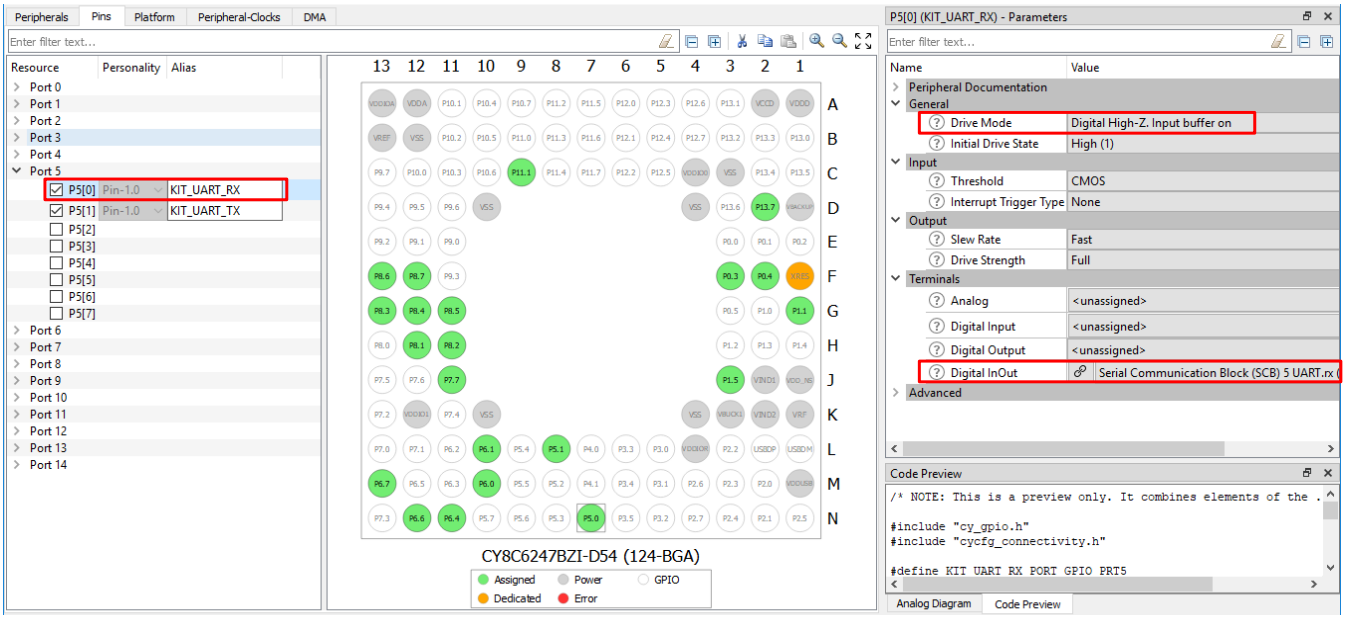

Figure 6 and Figure 7 show the Peripheral-Clock configuration for UART and TCPWM resources respectively.

Figure 6. Peripheral-Clock Configuration for UART

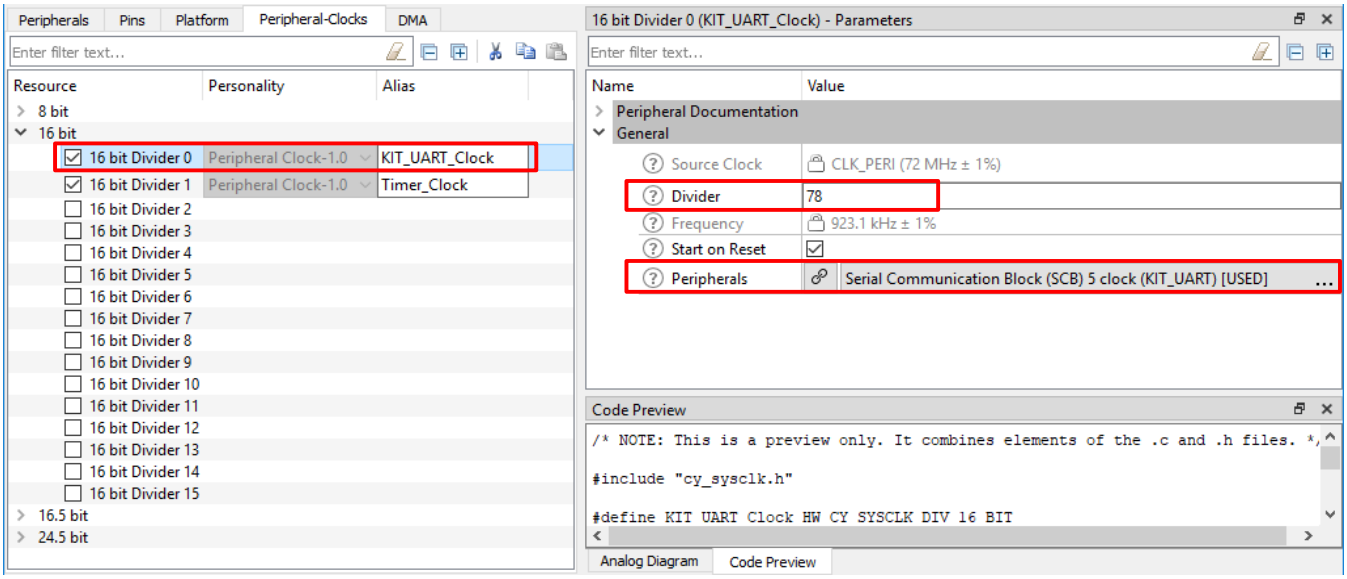

Figure 7. Peripheral-Clock Configuration for Timer

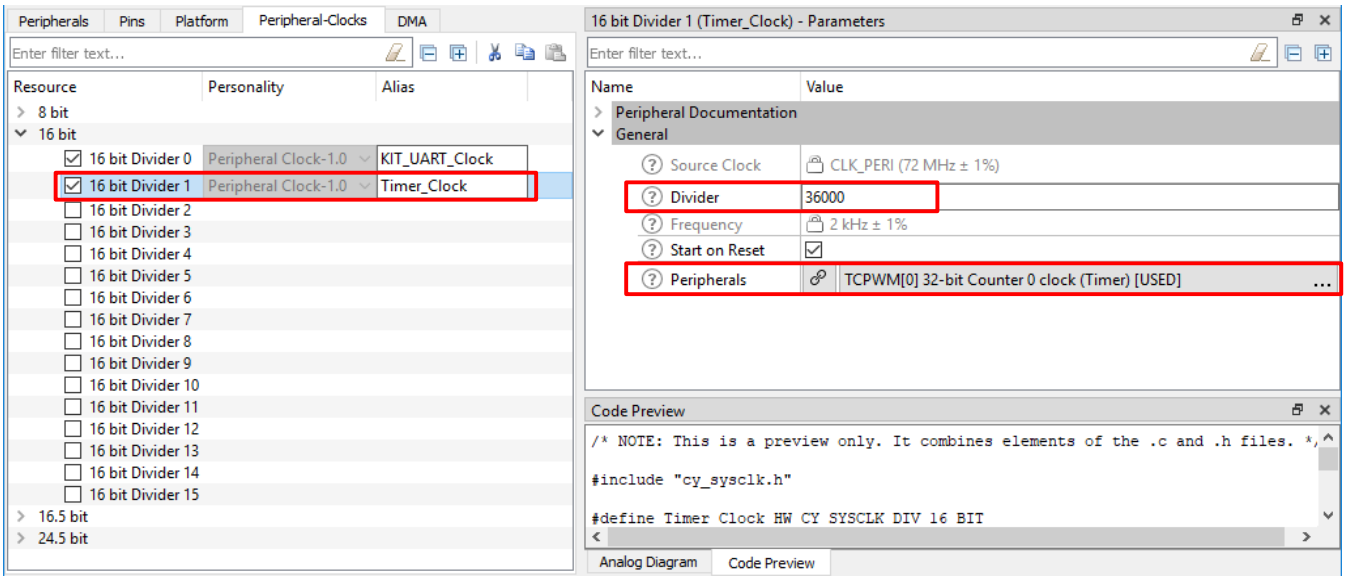

Table 2 shows the pin assignment for the application.

Table 2. Pin Assignments

Name	Port
LED	P13[7]
UART_RX	P5[0]
UART_TX	P5[1]

Reusing This Example

This example is designed for the CY8CKIT-062-WIFI-BT Pioneer Kit. To use the design on a different PSoC 6 MCU kit, import the application for that kit. If you are unsure how to import an application, see KBA225201. If changing to a different hardware, you may need to reassign pins.

Table 3. Device and Pin Mapping Table across PSoC 6 MCU Kits

Kit Name	Device Used	LED	UART_RX	UART_TX
CY8CKIT-062-WiFi-BT	CY8C6247BZI-D54	P13[7]	P5[0]	P5[1]
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	P13[7]	P5[0]	P5[1]
CY8CPROTO-062-4343W	CY8C624ABZI-D44	P13[7]	P5[0]	P5[1]

In some cases, a resource used by a code example (for example, a peripheral) 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 resource the device supports.

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
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	
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: CE221773 – PSoC 6 MCU: Hello World

Document Number: 002-23541

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6322373	SNVN	11/21/2018	New code example
*A	6344057	SNVN	02/06/2019	Code example updated for ModusToolbox 1.1

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, including Spansion LLC ("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 does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. 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. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, 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 or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

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.