

## Objective

This example demonstrates the usage of the real-time clock (RTC) in PSoC® 6 MCU using ModusToolbox™ IDE.

## Requirements

**Tool:** ModusToolbox™ IDE 1.1

**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

This code example demonstrates how to get and set the time in the real-time clock (RTC), using the RTC driver API. The UART interface is used to input user data and print the result on the terminal.

## Hardware Setup

This example uses the kit's default configuration. See the kit guide to ensure that the kit is configured correctly.

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 If you do not upgrade, you will see an error like “unable to find CMSIS-DAP device” or “KitProg firmware is out of date”.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

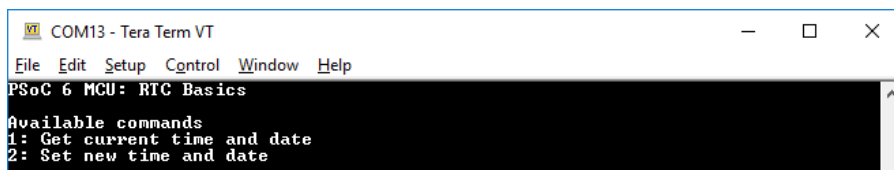
This example uses a terminal emulator. Install one if you don't have one. The instructions use [Tera Term](#).

## Operation

Follow the instructions that came with your kit to make sure that your kit is connected to your PC.

1. Open terminal software such as Tera Term and select the KitProg3's COM port with a baud rate setting of 115200 bps, data bits 8, parity none, and stop bit 1. Make sure the terminal sends a line feed (LF) character for a new line and enable local echo to see entered commands.
2. Add the code example to the IDE, in a new workspace. See [KBA225201](#).
3. Program the PSoC 6 MCU device. In the project explorer, select the **mainapp** project. In the Quick Panel, scroll to the **Launches** section and click the **Program (KitProg3)** configuration.
4. Confirm that the terminal program is working. It should show a message with the available commands, like [Figure 1](#).

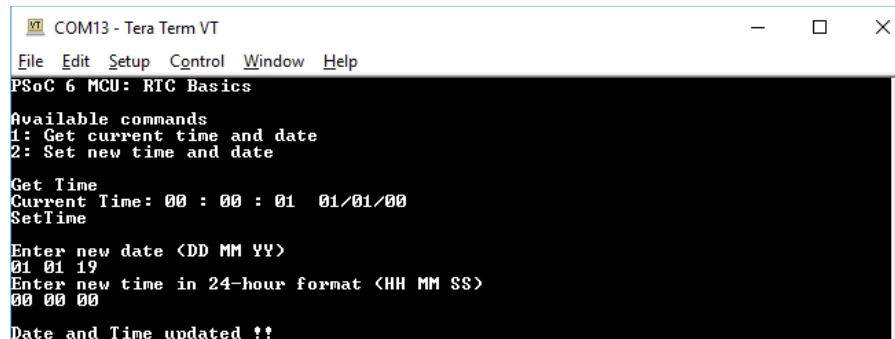
Figure 1. Available Commands in Terminal Window



5. Type "1". The terminal program shows the current time from the RTC, as shown in [Figure 2](#).

6. Type “2”. After that it will ask for new date and time. Enter the new date and time, and press **Enter** (Figure 2).

Figure 2. Display Terminal Messages



```

COM13 - Tera Term VT
File Edit Setup Control Window Help
PSoC 6 MCU: RTC Basics
Available commands
1: Get current time and date
2: Set new time and date
Get Time
Current Time: 00 : 00 : 01 01/01/00
SetTime
Enter new date <DD MM YY>
01 01 19
Enter new time in 24-hour format <HH MM SS>
00 00 00
Date and Time updated !!
  
```

## 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

This code example features the Real Time Clock resource and one UART resource. The RTC provides time and date information – second, minute, hour, day of the week, date, month, and year. The time and date information are updated every second with automatic leap year compensation performed by the RTC hardware block.

The CPU waits for the line feed (LF) character from the serial terminal. When it is received, the code parses the commands that have been sent.

- If the input command is ‘1’, display the current date and time.
- If the input command is ‘2’, update the RTC with new date and time.

## Resources and Settings

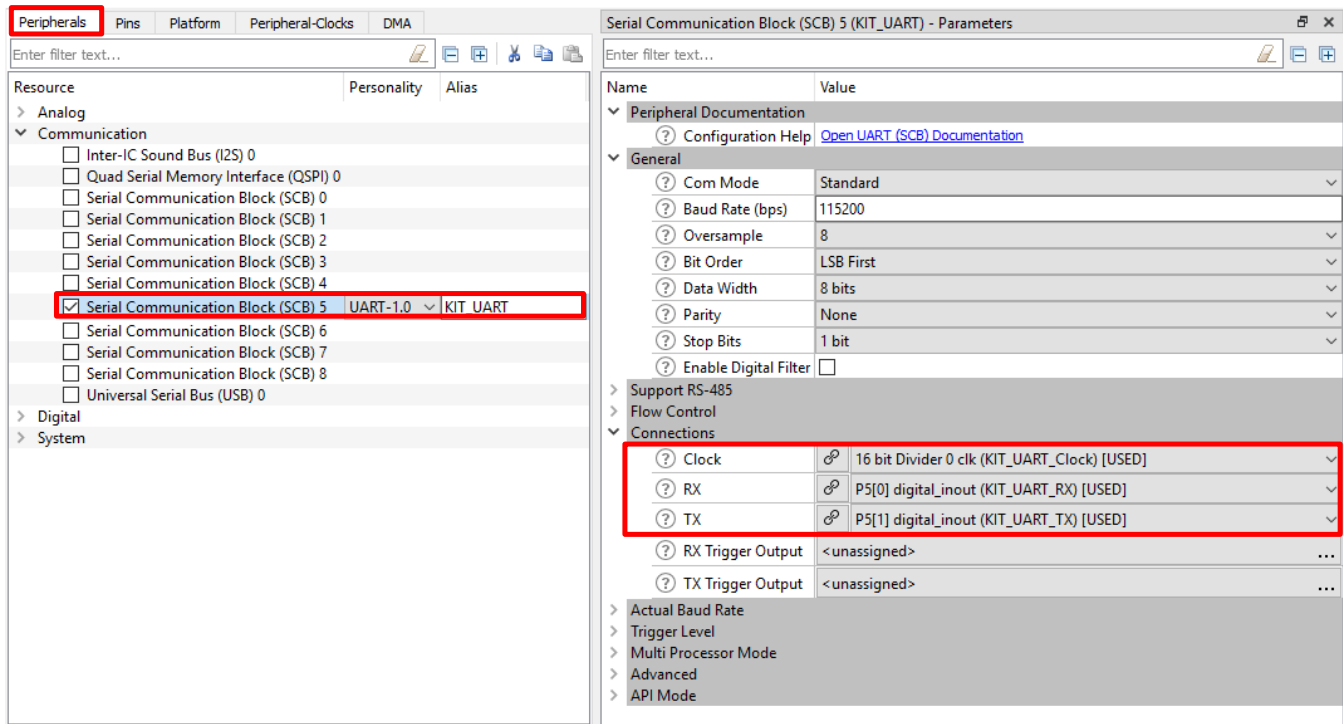
Table 1 lists the ModusToolbox resources used in this example, and how they are used in the design. For pin usage and configuration, open the **Pins** tab of the *design.modus* file.

Table 1. ModusToolbox Resources

Resource	Alias	Purpose	Non-default Settings
Real-Time Clock	RTC	Provides date and time information	Default
KIT_UART	KIT_UART	Prints messages on terminal program.	See <a href="#">Figure 3</a>
Digital Output Pin	KIT_UART_TX	Used for UART transmit (Tx)	See <a href="#">Figure 5</a>
	KIT_LED2	Provides visual feedback	See <a href="#">Figure 6</a>
Digital Input Pin	KIT_UART_RX	Used for UART receive (Rx)	See <a href="#">Figure 4</a>

[Figure 3](#) to [Figure 6](#) show non-default configuration settings for the resources.

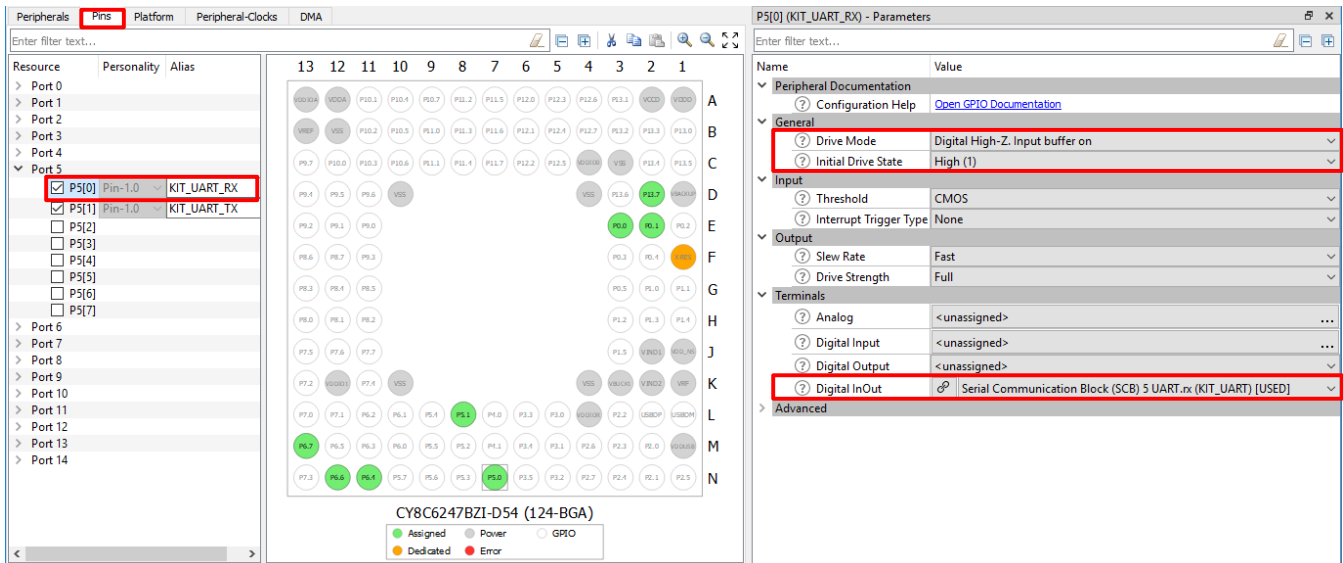
Figure 3. UART Configuration



**Serial Communication Block (SCB) 5 (KIT\_UART) - Parameters**

Name	Value
<b>Peripheral Documentation</b>	
Configuration Help	<a href="#">Open UART (SCB) Documentation</a>
<b>General</b>	
Com Mode	Standard
Baud Rate (bps)	115200
Oversample	8
Bit Order	LSB First
Data Width	8 bits
Parity	None
Stop Bits	1 bit
Enable Digital Filter	<input type="checkbox"/>
<b>Support RS-485</b>	
<b>Flow Control</b>	
<b>Connections</b>	
Clock	16 bit Divider 0 clk (KIT_UART_Clock) [USED]
RX	P5[0] digital_inout (KIT_UART_RX) [USED]
TX	P5[1] digital_inout (KIT_UART_TX) [USED]
RX Trigger Output	<unassigned>
TX Trigger Output	<unassigned>
<b>Actual Baud Rate</b>	
<b>Trigger Level</b>	
<b>Multi Processor Mode</b>	
<b>Advanced</b>	
<b>API Mode</b>	

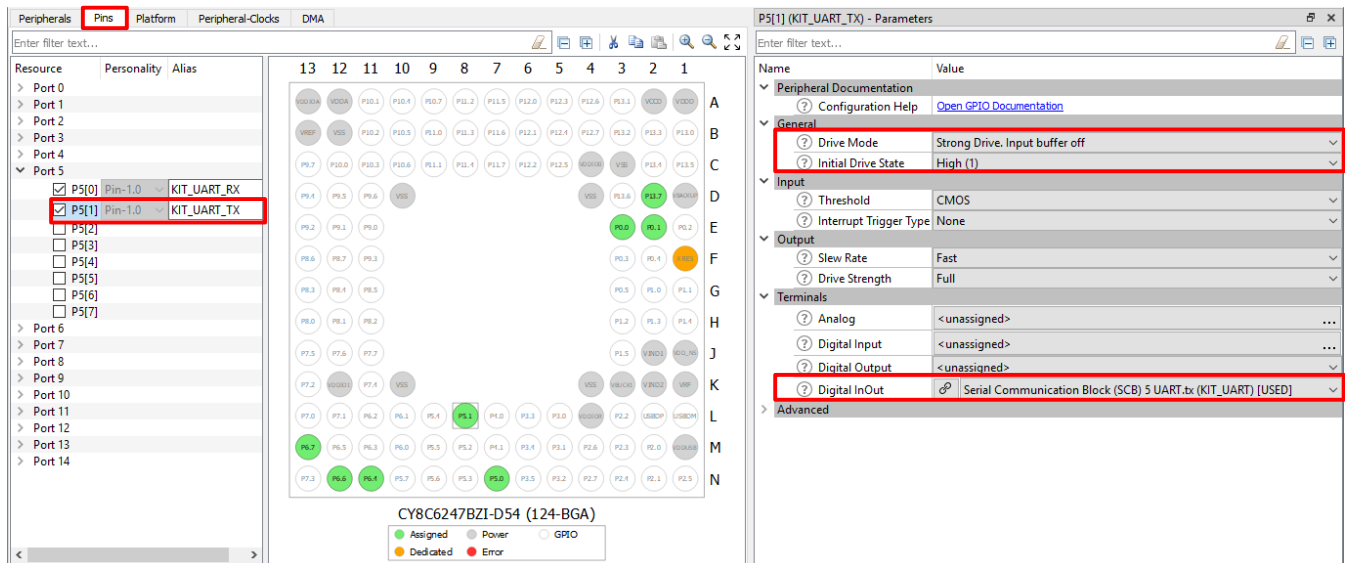
Figure 4. GPIO Pin Configuration for UART Rx



**P5[0] (KIT\_UART\_RX) - Parameters**

Name	Value
<b>Peripheral Documentation</b>	
Configuration Help	<a href="#">Open GPIO Documentation</a>
<b>General</b>	
Drive Mode	Digital High-Z, Input buffer on
Initial Drive State	High (1)
<b>Input</b>	
Threshold	CMOS
Interrupt Trigger Type	None
<b>Output</b>	
Slew Rate	Fast
Drive Strength	Full
<b>Terminals</b>	
Analog	<unassigned>
Digital Input	<unassigned>
Digital Output	<unassigned>
Digital InOut	Serial Communication Block (SCB) 5 UART rx (KIT_UART) [USED]
<b>Advanced</b>	

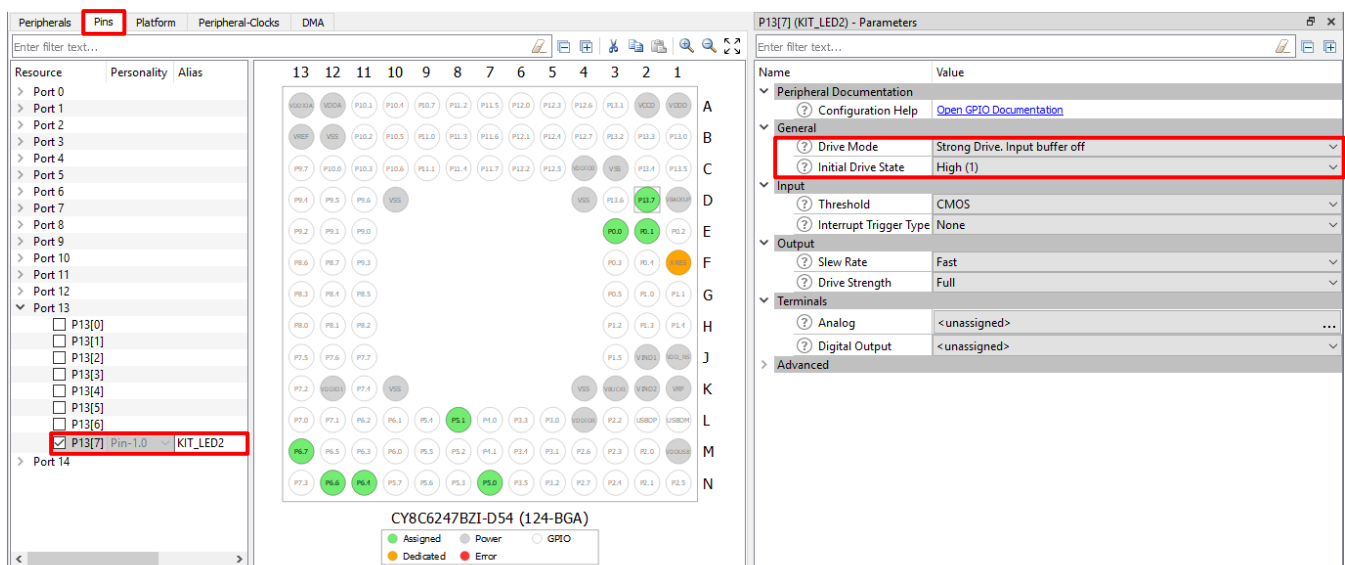
Figure 5. GPIO Pin Configuration for UART Tx



The screenshot displays the PSoC Creator interface for configuring a GPIO pin. On the left, the 'Pins' tab is active, showing a list of resources. P5[1] is selected and assigned to KIT\_UART\_TX. The central pin matrix shows the physical layout of the CY8C6247BZI-D54 (124-BGA) package. On the right, the 'Parameters' window for P5[1] (KIT\_UART\_TX) is shown. The configuration is as follows:

Name	Value
Peripheral Documentation	Configuration Help <a href="#">Open GPIO Documentation</a>
General	
Drive Mode	Strong Drive, Input buffer off
Initial Drive State	High (1)
Input	
Threshold	CMOS
Interrupt Trigger Type	None
Output	
Slew Rate	Fast
Drive Strength	Full
Terminals	
Analog	<unassigned>
Digital Input	<unassigned>
Digital Output	<unassigned>
Digital In/Out	Serial Communication Block (SCB) 5 UART.tx (KIT_UART) [USED]
Advanced	

Figure 6. GPIO Pin Configuration for LED Pin

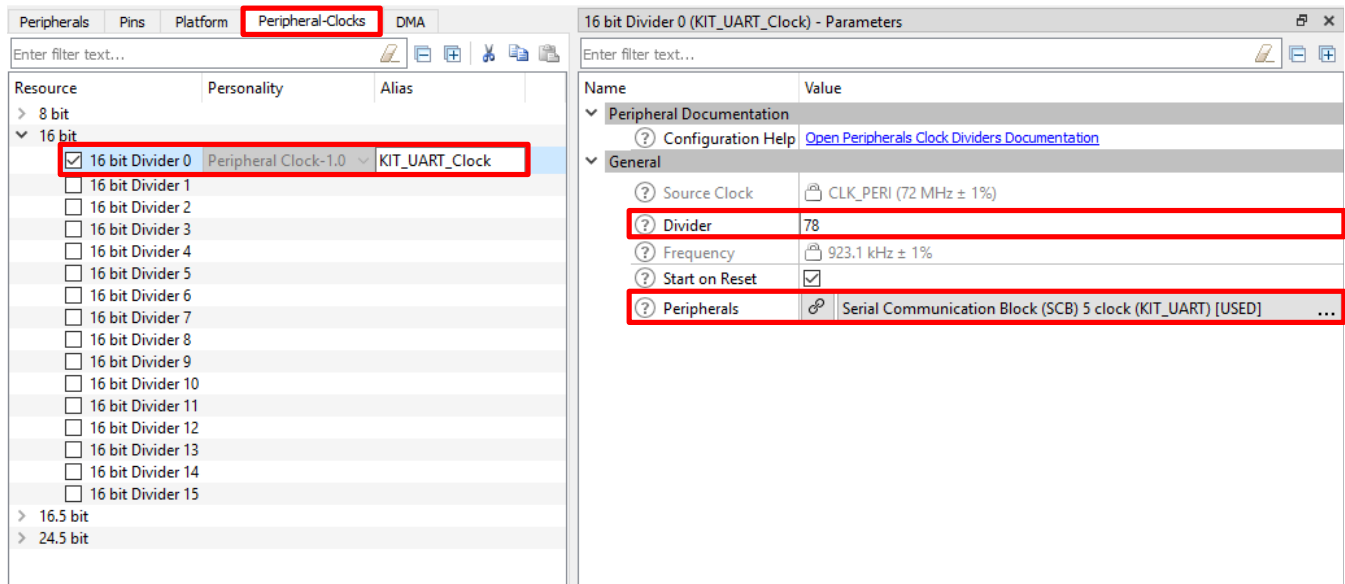


The screenshot displays the PSoC Creator interface for configuring a GPIO pin. On the left, the 'Pins' tab is active, showing a list of resources. P13[7] is selected and assigned to KIT\_LED2. The central pin matrix shows the physical layout of the CY8C6247BZI-D54 (124-BGA) package. On the right, the 'Parameters' window for P13[7] (KIT\_LED2) is shown. The configuration is as follows:

Name	Value
Peripheral Documentation	Configuration Help <a href="#">Open GPIO Documentation</a>
General	
Drive Mode	Strong Drive, Input buffer off
Initial Drive State	High (1)
Input	
Threshold	CMOS
Interrupt Trigger Type	None
Output	
Slew Rate	Fast
Drive Strength	Full
Terminals	
Analog	<unassigned>
Digital Output	<unassigned>
Advanced	

Figure 7 shows the Peripheral-Clock configuration for UART resource to get 115200 bps baud rate.

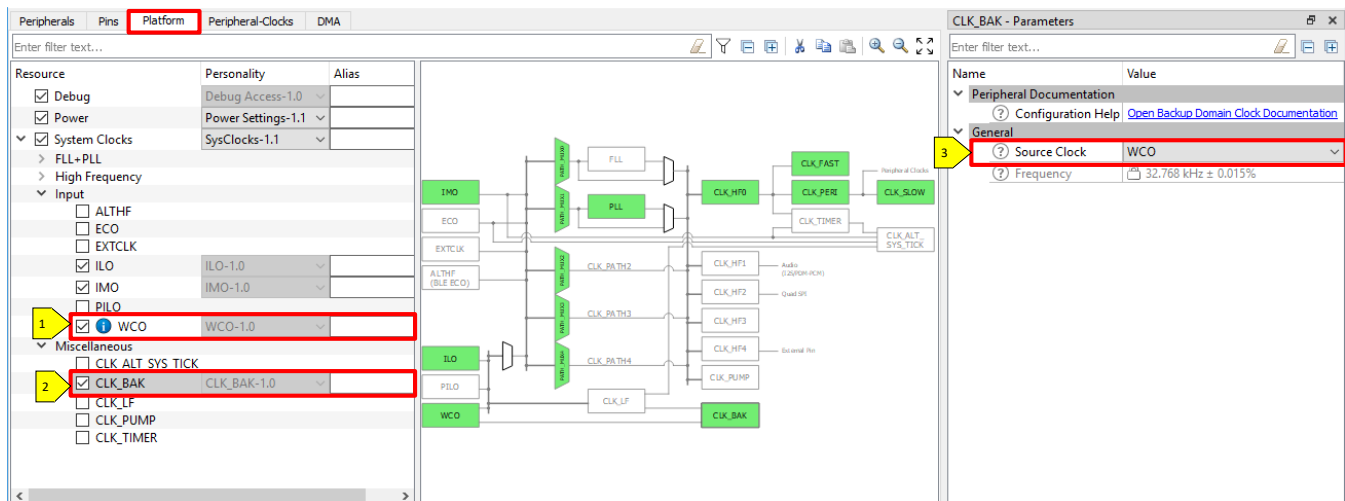
Figure 7. Peripheral-Clock Configuration for UART



## Design Considerations

It is necessary to provide a 32.768-kHz clock for the RTC function in the backup power domain. For accurate RTC operation, it is recommended that you use a Watch Crystal Oscillator (WCO). Figure 8 shows the backup clock configuration used in this project.

Figure 8. Backup Clock Configuration



## Reusing This Example

This example is designed for the supported kits. To port the design to a different PSoC 6 MCU device, right-click an application project and choose **Change Device**. If changing to a different kit, you may need to reassign pins.

Table 2. Device and Pin Mapping Table across PSoC 6 MCU Kits

Kit Name	Device Used	KIT_LED2	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, an IP block) 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 device supports.

## Related Documents

For a comprehensive list of PSoC 6 MCU resources, see [KBA223067](#) in the Cypress community.

Application Notes	
<a href="#">AN210781</a> – 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
<a href="#">AN221774</a> – Getting Started with PSoC 6 MCU	Describes PSoC 6 MCU devices and how to build your first ModusToolbox application and PSoC Creator project
<a href="#">AN215656</a> – 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 <a href="#">Cypress GitHub site</a> for a comprehensive collection of code examples using ModusToolbox IDE	
Device Documentation	
<a href="#">PSoC 6 MCU: PSoC 63 with BLE Datasheet</a>	<a href="#">PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual</a>
Development Kit Documentation	
<a href="#">CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit</a>	
<a href="#">CY8CKIT-062-WiFi-BT PSoC 6 WiFi-BT Pioneer Kit</a>	
<a href="#">CY8CPROTO-062-4343W PSoC 6 Wi-Fi BT Prototyping Kit</a>	
<a href="#">CY8CPROTO-063 BLE PSoC 6 BLE Prototyping Kit</a>	
Tool Documentation	
<a href="#">ModusToolbox</a>	ModusToolbox simplifies development for IoT designers. It delivers easy-to-use tools and a familiar microcontroller (MCU) integrated development environment (IDE) for Windows, macOS, and Linux.

## Cypress Resources

Cypress provides a wealth of data at [www.cypress.com](http://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: CE216825 - PSoC 6 MCU Real-Time Clock Basics

Document Number: 002-25740

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6409432	AJYA	02/20/2019	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	<a href="http://cypress.com/arm">cypress.com/arm</a>
Automotive	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
Interface	<a href="http://cypress.com/interface">cypress.com/interface</a>
Internet of Things	<a href="http://cypress.com/iot">cypress.com/iot</a>
Memory	<a href="http://cypress.com/memory">cypress.com/memory</a>
Microcontrollers	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
Power Management ICs	<a href="http://cypress.com/pmic">cypress.com/pmic</a>
Touch Sensing	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB Controllers	<a href="http://cypress.com/usb">cypress.com/usb</a>
Wireless Connectivity	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

## PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

## Cypress Developer Community

[Community](#) | [Code Examples](#) | [Projects](#) | [Videos](#) | [Blogs](#)  
| [Training](#) | [Components](#)

## Technical Support

[cypress.com/support](http://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, 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](http://cypress.com). Other names and brands may be claimed as property of their respective owners.