CE218964 - PSoC 6 MCU RTC Daily Alarm

Objective

This example demonstrates the RTC Alarm function of the PSoC® 6 MCU Real-Time Clock (RTC) 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, PSoC6 WiFi-Prototyping Kit

Overview

This code example demonstrates how to configure RTC registers for a daily alarm using the RTC driver API in the Peripheral Driver Library (PDL). A GPIO output is included for an LED to notify alarm expiration. A UART is used to show the current and alarm times. In this code example daily alarm is configured for Monday to Friday.

Hardware Setups

This example uses the PSoC 6 WiFi-BT Pioneer Kit in default configuration. Refer to the kit guide to ensure 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, PSoC 6 MCU KitProg Firmware Loader** section. If you do not upgrade, you will see an error like “unable to find CMSIS-DAP device” or “KitProg firmware is out of date”.

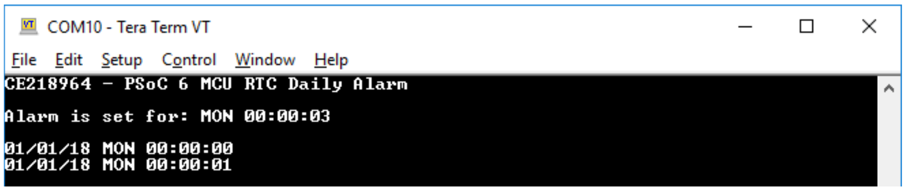
Software Setup

This code example requires a PC terminal software. 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 your terminal software and select the KitProg COM port. Set the other serial port parameters as follows:
   a. Baud rate: 115200bps
   b. Data: 8-bit
   c. Parity: None
   d. Stop: 1-bit
   e. Flow control: None
3. Import the code example into a new workspace. See KBA225201.
4. 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.
5. Confirm that the terminal program is working. A message should be printed every one second in the terminal window as shown in Figure 1.

Figure 1. UART Display Start Message



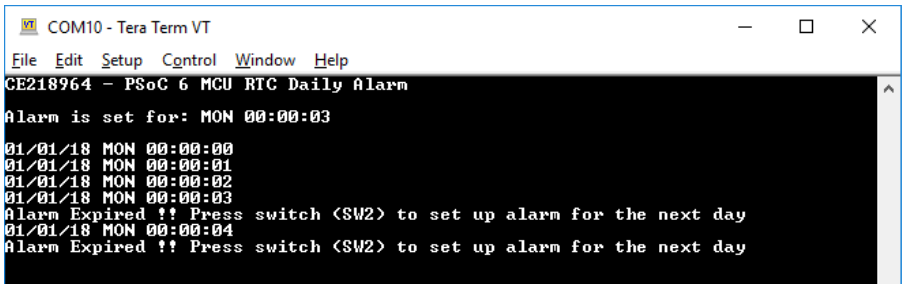
```
CE218964 - PSoC 6 MCU RTC Daily Alarm

Alarm is set for: MON 00:00:03

01/01/18 MON 00:00:00
01/01/18 MON 00:00:01
```

6. The following message should be printed after three second "`Alarm Expired !! Press SW2 for to set alarm for next day`"(Figure 2).

Figure 2. UART Display Message After RTC Alarm



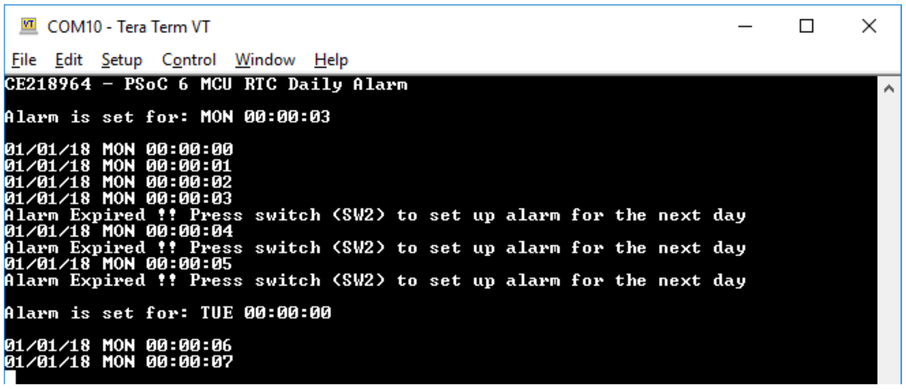
```
CE218964 - PSoC 6 MCU RTC Daily Alarm

Alarm is set for: MON 00:00:03

01/01/18 MON 00:00:00
01/01/18 MON 00:00:01
01/01/18 MON 00:00:02
01/01/18 MON 00:00:03
Alarm Expired !! Press switch (SW2) to set up alarm for the next day
01/01/18 MON 00:00:04
Alarm Expired !! Press switch (SW2) to set up alarm for the next day
```

7. Confirm Red LED (KIT_LED2) toggles every second.
8. Press switch SW2 (P0[4]). It should show the next alarm information as shown in Figure 3.

Figure 3. UART Display Message After Switch Is Pressed



```
CE218964 - PSoC 6 MCU RTC Daily Alarm

Alarm is set for: MON 00:00:03

01/01/18 MON 00:00:00
01/01/18 MON 00:00:01
01/01/18 MON 00:00:02
01/01/18 MON 00:00:03
Alarm Expired !! Press switch (SW2) to set up alarm for the next day
01/01/18 MON 00:00:04
Alarm Expired !! Press switch (SW2) to set up alarm for the next day
01/01/18 MON 00:00:05
Alarm Expired !! Press switch (SW2) to set up alarm for the next day

Alarm is set for: TUE 00:00:00

01/01/18 MON 00:00:06
01/01/18 MON 00:00:07
```

Design and Implementation

This code example features the Real Time Clock resource, one GPIO for LED alarm indicator, one GPIO for user switch and one UART resource.

The PSoC 6 MCU RTC is a hardware-based function; the alarm time can be configured by the alarm register fields. The daily alarm needs to enable the hour, minute, and second time fields. Each alarm field is paired with its own enable field. For example, the sec (second) field is paired with the secEn (second enable); the dayOfWeek field is paired with the dayOfWeekEn field. If an enable field is set, the field value will be used for matching the alarm time; otherwise the field value will be ignored.

In this code example, the alarm function uses the RTC alarm 1 interrupt. After an alarm has expired, the code prints the alarm expiration message and toggles the red LED (KIT_LED2) every second until the SW2 button is pressed. After the switch is pressed the alarm is configured for the next day.

Resources and Settings

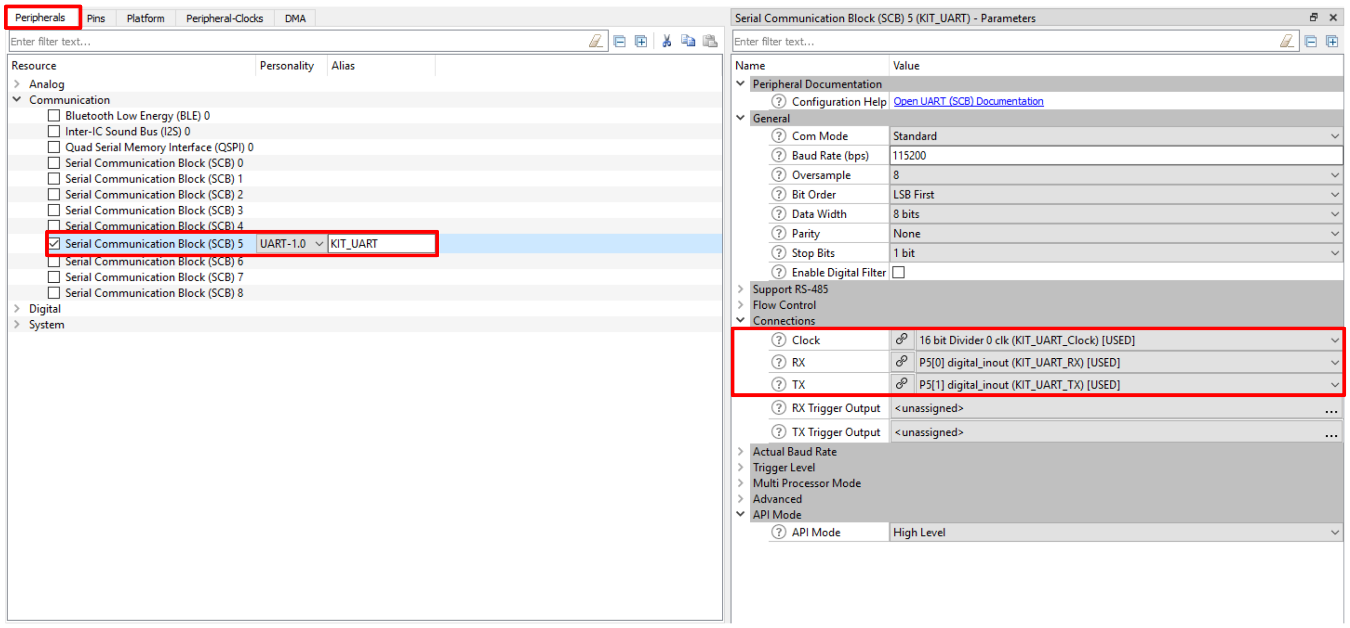
Table 1 lists the ModusToolbox resources used in this example, and how they are used in the design

Table 1. ModusToolbox Resources

Resources	Alias	Purpose	Non-default Settings
Real Time Clock	RTC	Provide date and time information	Default
SCB	KIT_UART	Used for printing terminal messages	See Figure 4
Digital Output Pin	KIT_UART_TX	Used for UART transmit (Tx)	See Figure 5
	KIT_LED2	Provide visual feedback	See Figure 6
Digital Input Pin	KIT_UART_RX	Used for UART receive (Rx)	See Figure 7
	KIT_BTN2	Provide user interaction	See Figure 8

Figure 4 to Figure 8 shows the non-default configuration settings for the ModusToolbox resources.

Figure 4. UART Configuration

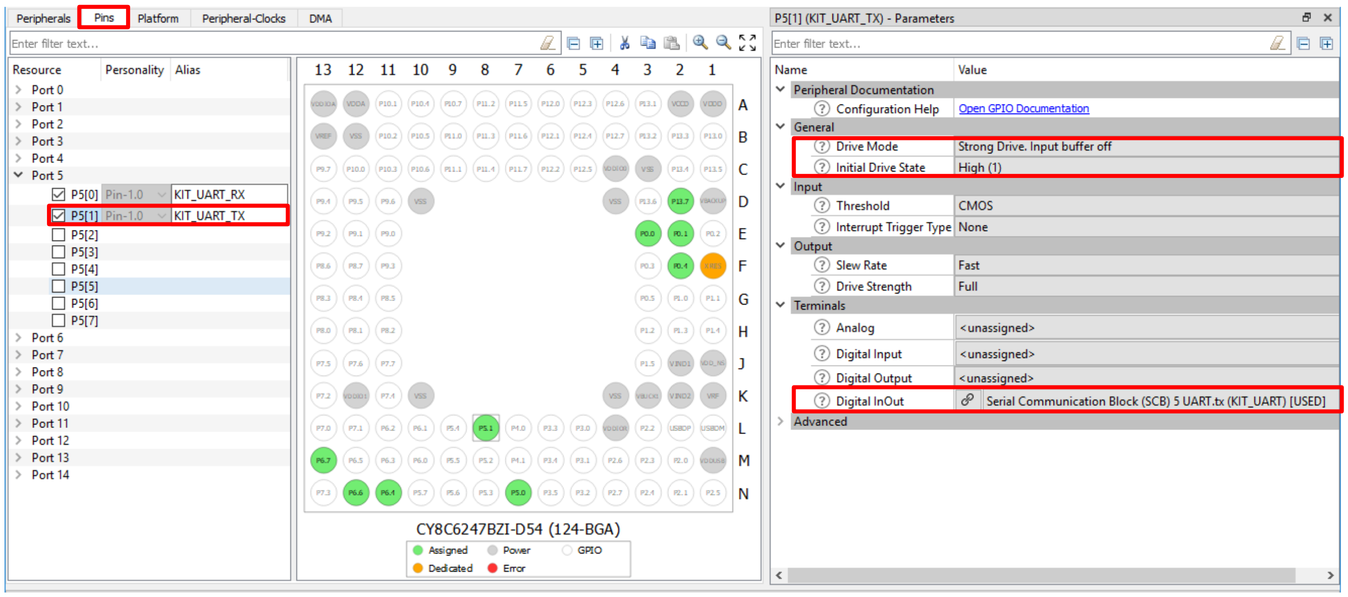


Resource	Personality	Alias
Analog		
Communication		
Bluetooth Low Energy (BLE) 0		
Inter-IC Sound Bus (I2S) 0		
Quad Serial Memory Interface (QSPI) 0		
Serial Communication Block (SCB) 0		
Serial Communication Block (SCB) 1		
Serial Communication Block (SCB) 2		
Serial Communication Block (SCB) 3		
Serial Communication Block (SCB) 4		
✓ Serial Communication Block (SCB) 5	UART-1.0	KIT_UART
Serial Communication Block (SCB) 6		
Serial Communication Block (SCB) 7		
Serial Communication Block (SCB) 8		
Digital		
System		

Serial Communication Block (SCB) 5 (KIT_UART) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open UART (SCB) Documentation
General	
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	☐
Support RS-485	
Flow Control	
Connections	
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>
Actual Baud Rate	
Trigger Level	
Multi Processor Mode	
Advanced	
API Mode	
API Mode	High Level

Figure 5. GPIO Pin Configuration for UART Tx

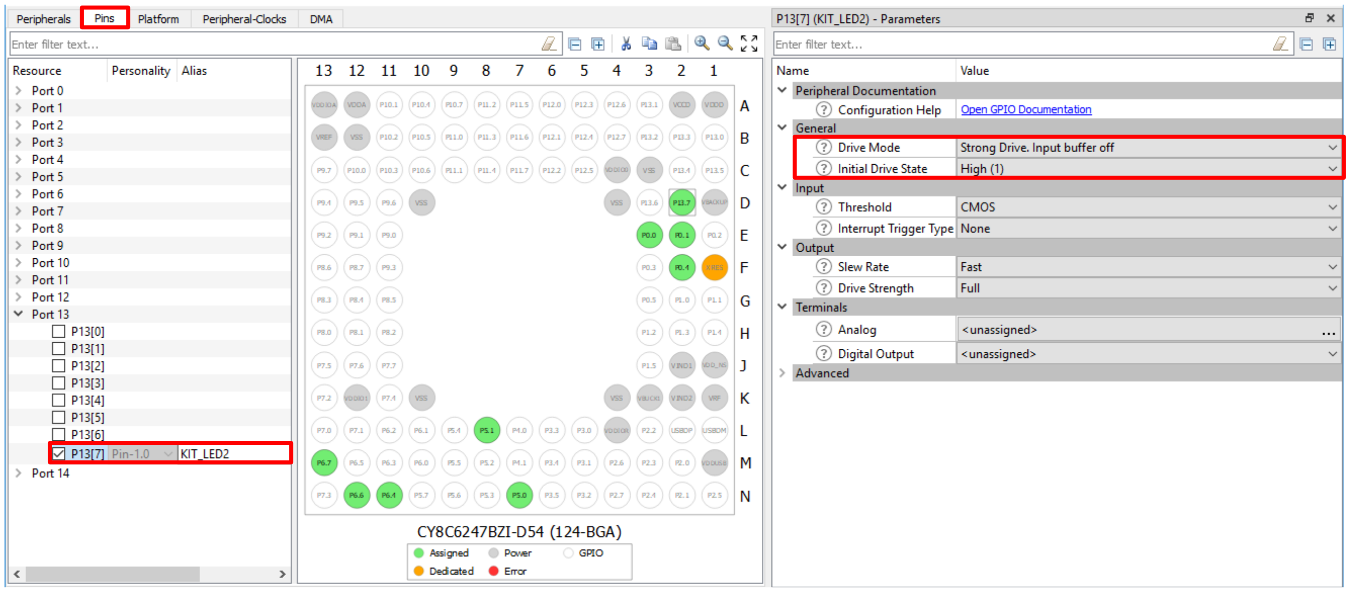


Resource	Personality	Alias
Port 5		
☑ P5[0]	Pin-1.0	KIT_UART_RX
☑ P5[1]	Pin-1.0	KIT_UART_TX

P5[1] (KIT_UART_TX) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open GPIO Documentation
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 InOut	Serial Communication Block (SCB) 5 UART.tx (KIT_UART) [USED]
Advanced	

Figure 6. GPIO Pin Configuration for LED Pin

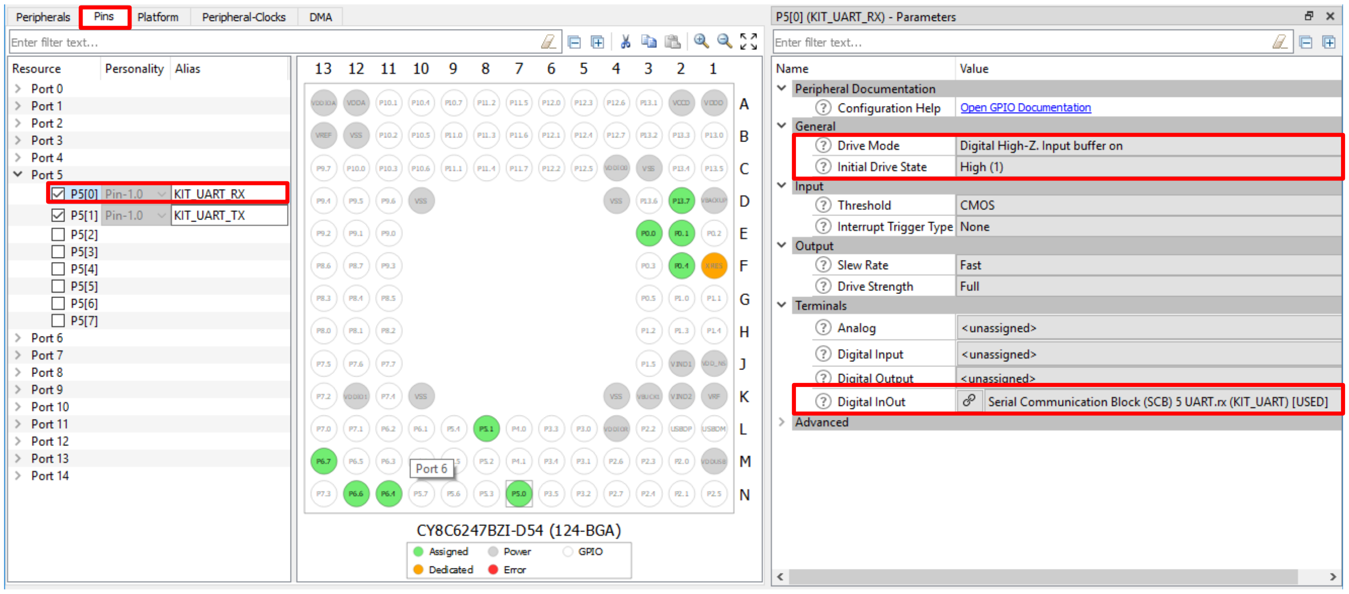


Resource	Personality	Alias
Port 13		
☑ P13[7]	Pin-1.0	KIT_LED2

P13[7] (KIT_LED2) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open GPIO Documentation
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. GPIO Pin Configuration for UART Rx

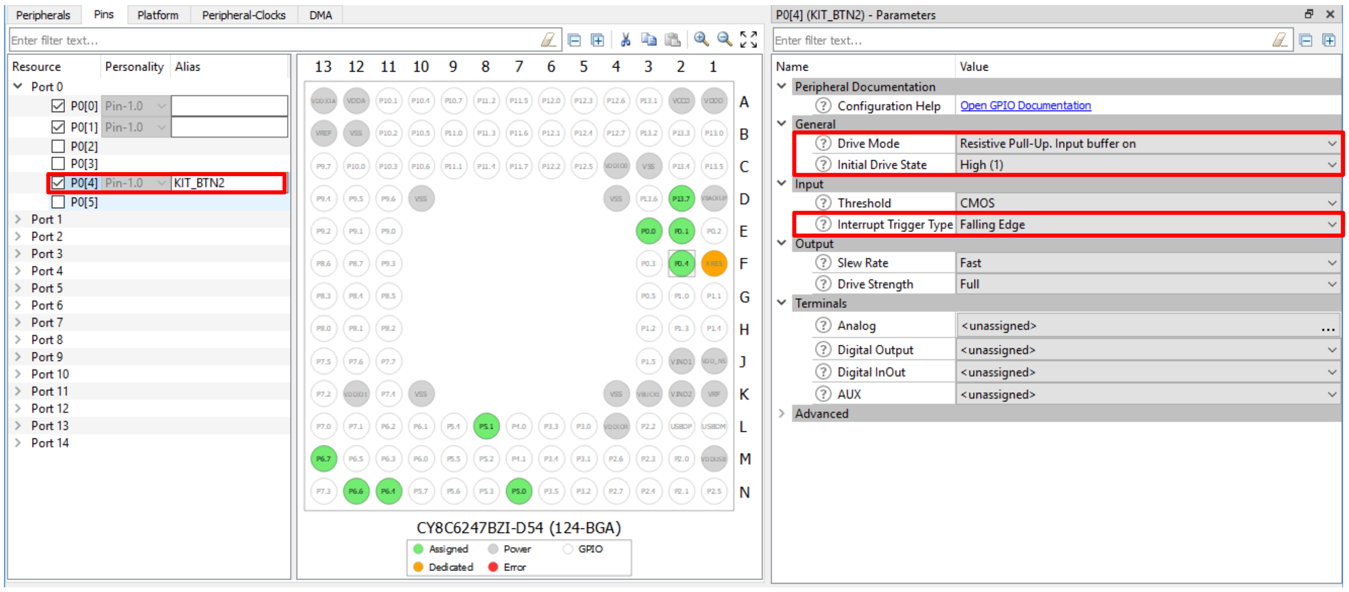


Resource	Personality	Alias
Port 0		
Port 1		
Port 2		
Port 3		
Port 4		
Port 5		
☑ P5[0]	Pin-1.0	KIT_UART_RX
☑ P5[1]	Pin-1.0	KIT_UART_TX
☐ P5[2]		
☐ P5[3]		
☐ P5[4]		
☐ P5[5]		
☐ P5[6]		
☐ P5[7]		
Port 6		
Port 7		
Port 8		
Port 9		
Port 10		
Port 11		
Port 12		
Port 13		
Port 14		

P5[0] (KIT_UART_RX) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open GPIO Documentation
General	
Drive Mode	Digital High-Z. Input buffer on
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 InOut	Serial Communication Block (SCB) 5 UART.rx (KIT_UART) [USED]
Advanced	

Figure 8. GPIO Pin Configuration for switch



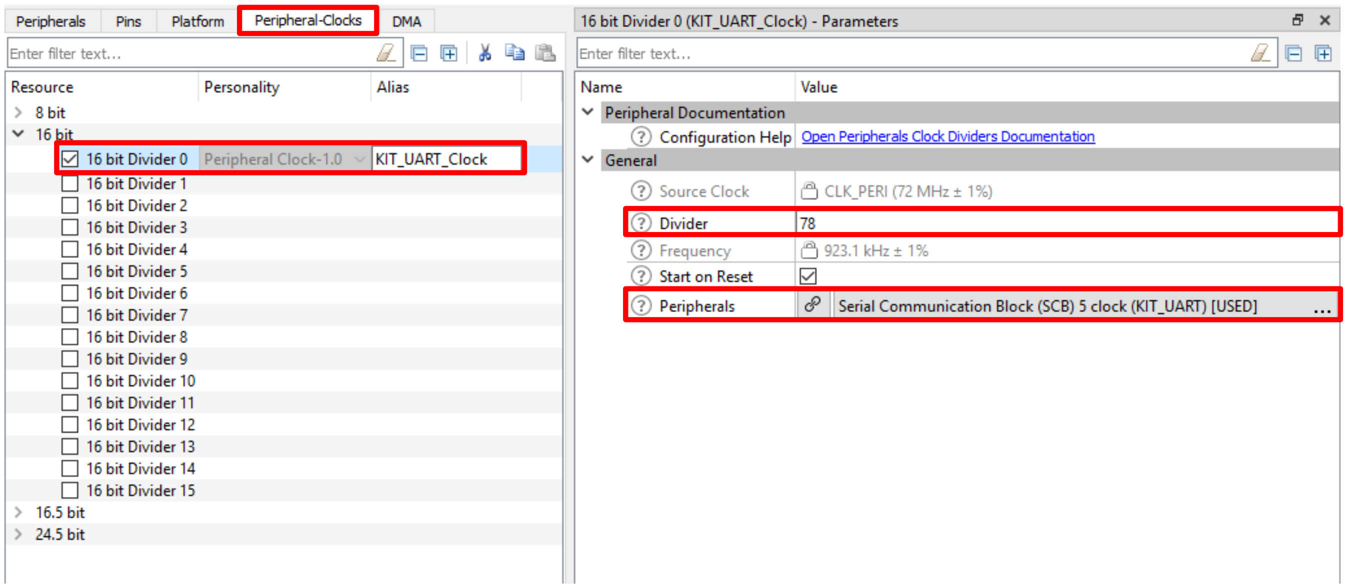
Resource	Personality	Alias
Port 0		
☑ P0[0]	Pin-1.0	
☑ P0[1]	Pin-1.0	
☐ P0[2]		
☐ P0[3]		
☑ P0[4]	Pin-1.0	KIT_BTN2
☐ P0[5]		
Port 1		
Port 2		
Port 3		
Port 4		
Port 5		
Port 6		
Port 7		
Port 8		
Port 9		
Port 10		
Port 11		
Port 12		
Port 13		
Port 14		

P0[4] (KIT_BTN2) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open GPIO Documentation
General	
Drive Mode	Resistive Pull-Up. Input buffer on
Initial Drive State	High (1)
Input	
Threshold	CMOS
Interrupt Trigger Type	Falling Edge
Output	
Slew Rate	Fast
Drive Strength	Full
Terminals	
Analog	<unassigned>
Digital Output	<unassigned>
Digital InOut	<unassigned>
AUX	<unassigned>
Advanced	

Figure 9 shows the Peripheral-Clock configuration for resource.

Figure 9. Peripheral-Clock Configuration for UART



Resource	Personality	Alias
8 bit		
16 bit		
☑ 16 bit Divider 0	Peripheral Clock-1.0	KIT_UART_Clock
☐ 16 bit Divider 1		
☐ 16 bit Divider 2		
☐ 16 bit Divider 3		
☐ 16 bit Divider 4		
☐ 16 bit Divider 5		
☐ 16 bit Divider 6		
☐ 16 bit Divider 7		
☐ 16 bit Divider 8		
☐ 16 bit Divider 9		
☐ 16 bit Divider 10		
☐ 16 bit Divider 11		
☐ 16 bit Divider 12		
☐ 16 bit Divider 13		
☐ 16 bit Divider 14		
☐ 16 bit Divider 15		
16.5 bit		
24.5 bit		

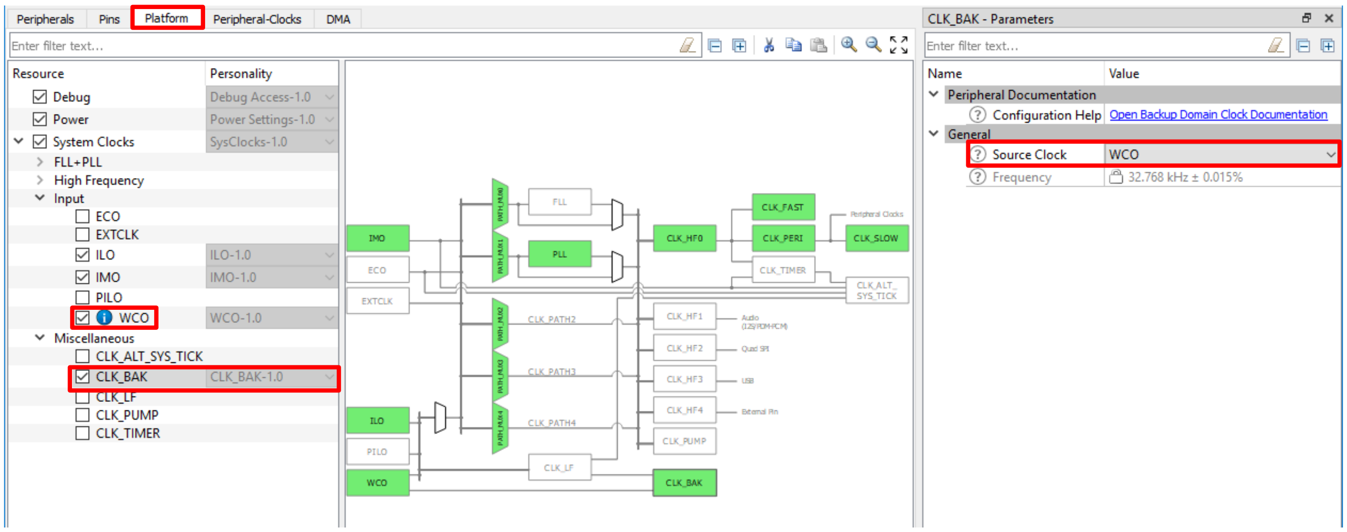
16 bit Divider 0 (KIT_UART_Clock) - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open Peripherals Clock Dividers Documentation
General	
Source Clock	CLK_PERI (72 MHz ± 1%)
Divider	78
Frequency	923.1 kHz ± 1%
Start on Reset	☑
Peripherals	Serial Communication Block (SCB) 5 clock (KIT_UART) [USED]

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 10 shows the backup clock configuration used in this project.

Figure 10. Backup Clock Configuration



Resource	Personality
☑ Debug	Debug Access-1.0
☑ Power	Power Settings-1.0
☑ System Clocks	SysClocks-1.0
FLL+PLL	
High Frequency	
Input	
☐ ECO	
☐ EXTCLK	
☑ ILO	ILO-1.0
☑ IMO	IMO-1.0
☐ PILO	
☑ WCO	WCO-1.0
Miscellaneous	
☐ CLK_ALT_SYS_TICK	
☑ CLK_BAK	CLK_BAK-1.0
☐ CLK_LF	
☐ CLK_PUMP	
☐ CLK_TIMER	

CLK_BAK - Parameters

Name	Value
Peripheral Documentation	
Configuration Help	Open Backup Domain Clock Documentation
General	
Source Clock	WCO
Frequency	32.768 kHz ± 0.015%

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_LED	KIT_UART_RX	KIT_UART_TX	KIT_BTN2
CY8CKIT-062-WiFi-BT	CY8C6247BZI-D54	P13[7]	P5[0]	P5[1]	P0[4]
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	P13[7]	P5[0]	P5[1]	P0[4]
CY8CPROTO-062-4343W	CY8C624ABZI-D44	P13[7]	P5[0]	P5[1]	P0[4]

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 ModusToolbox application and 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	
CE216825 - PSoC® 6 Real-Time Clock Basics	
CE218542 - PSoC 6 MCU Custom Tick Timer Using RTC Alarm	
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 the PSoC 6 MCU devices, see KBA223067 in the Cypress community for a comprehensive list of PSoC 6 MCU resources.

Document History

Document Title: CE218964 - PSoC 6 MCU RTC Daily Alarm

Document Number: 002-25550

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6371319	AJYA	11/22/2018	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

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