

## Objective

This example demonstrates the use of PSoC® 6 MCU TCPWM to count the number of external events.

## Overview

The example shows how to use the TCPWM to count a number of external events and display the results over UART.

## Requirements

**Tool:** PSoC Creator™ 4.2

**Programming Language:** C (ARM® GCC 5.4-2016-q2-update, ARM MDK 5.22)

**Associated Parts:** All PSoC 6 MCU parts

**Related Hardware:** CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

## Design

The design shown in Figure 1 consists of two TCPWM Components, EVENT\_COUNTER and TIMER. The EVENT\_COUNTER is configured to increment by one whenever there is a falling edge on its count input. The kit button SW2 is connected to the EVENT\_COUNTER count input. The TIMER is configured to generate an interrupt every ten seconds. The EVENT\_COUNTER counts the number of kit button presses every ten seconds and sends the data to serial terminal through UART. The ERROR\_RED\_LED switches ON if the UART initialization fails.

Figure 1. TCPWM Event Counter Example Schematic

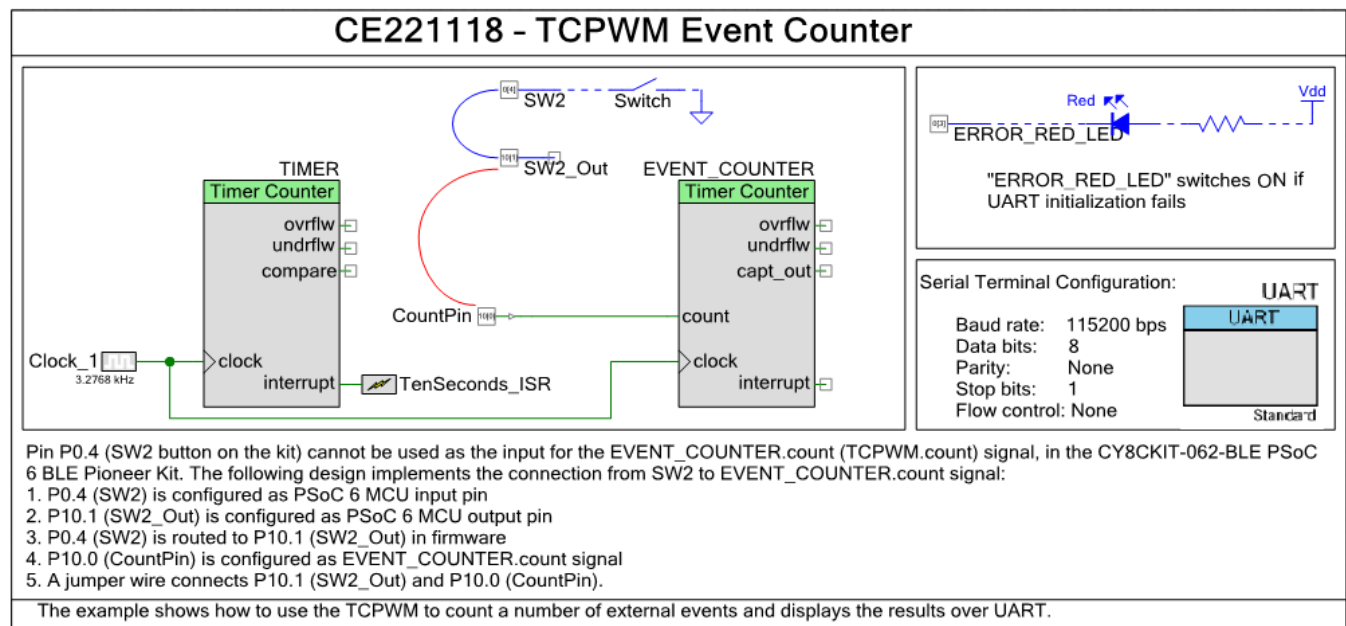
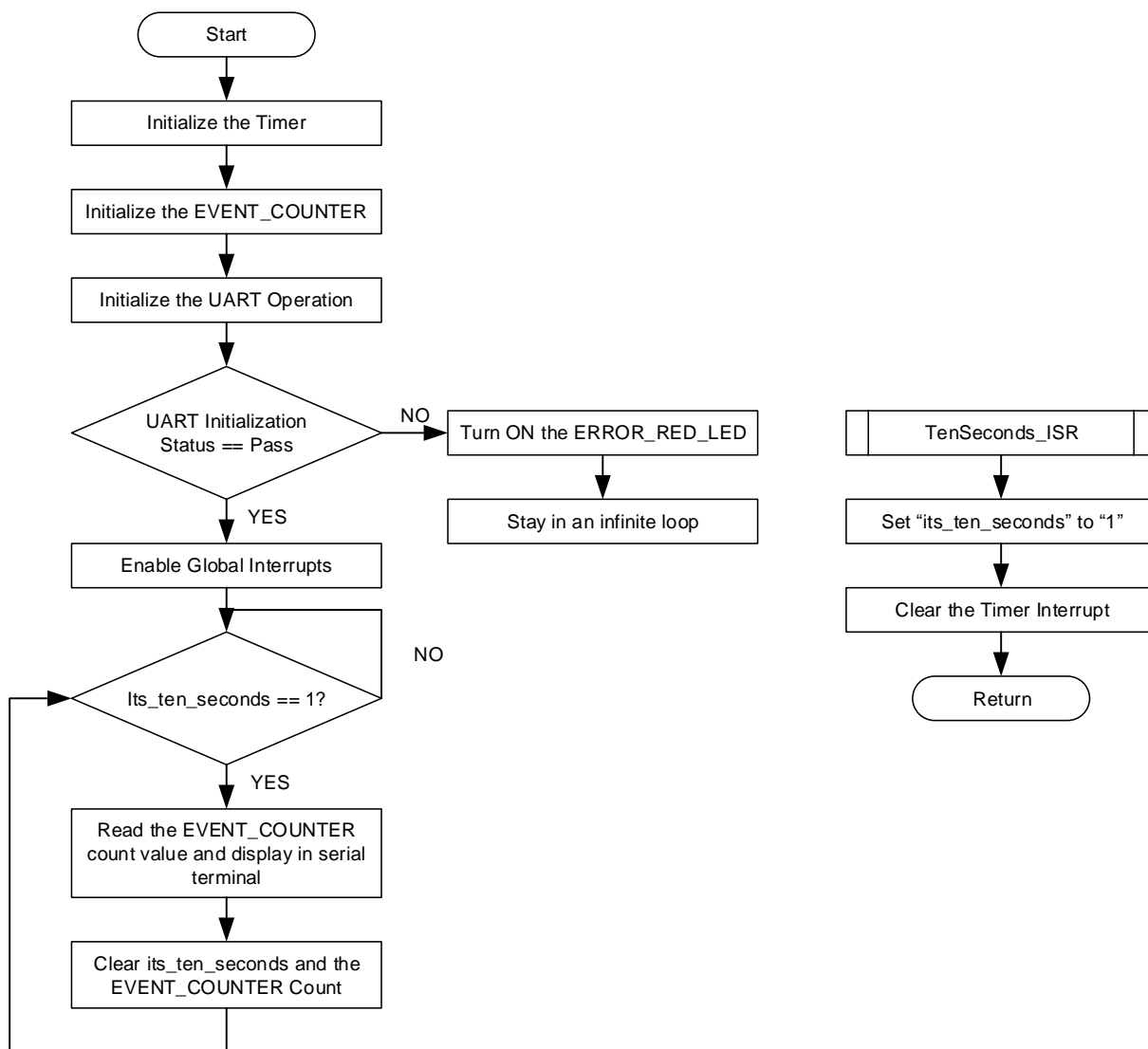


Figure 2 shows the firmware flowchart.

Figure 2. Firmware Flowchart



## Design Considerations

This code example is designed to run on CY8CKIT-062-BLE with the PSoC 6 MCU device. To port the design to other PSoC 6 MCU family devices and kits, you must change the target device in Device Selector, and change the pin assignments in the *cydwr* settings. For single-core PSoC 6 MCU devices, port the code from *main\_cm4.c* to *main.c* file as ARM Cortex® M0+ CPU is not used in this code example.

Pin P0.4 (SW2 button on the kit) cannot be used as the input for the EVENT\_COUNTER.count (TCPWM.count) signal, in the CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit. The following design implements the connection from SW2 to EVENT\_COUNTER.count signal:

1. P0.4 (SW2) is configured as PSoC 6 MCU input pin
2. P10.1 (SW2\_Out) is configured as PSoC 6 MCU output pin
3. P0.4 (SW2) is routed to P10.1 (SW2\_Out) in firmware

4. P10.0 (CountPin) is configured as EVENT\_COUNTER.count signal
5. A jumper wire connects P10.1 (SW2\_Out) and P10.0 (CountPin).

## Hardware Setup

The code example works with the default settings on the CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit. If the settings are different from the default values, see the Selection Switches table in the [kit guide](#) to reset to the default settings.

## Operation

1. Connect CY8CKIT-062 BLE to a USB port on your PC.
2. Open a serial port communication program such as Tera Term and select the corresponding COM port. Configure the terminal to match the UART: 115200 baud rate, 8N1, and Flow control – None. These settings must match the configuration of the PSoC Creator UART Component in the project.
3. Connect an external hardware jumper wire between P10.1 and P10.0 in CY8CKIT-062 BLE.
4. Build and program the application into CY8CKIT-062 BLE. For more information on building a project or programming a device, see PSoC Creator Help.
5. Count the number of times you press and release the SW2 switch within ten seconds. The terminal window should display the same count.
6. Repeat step 5 multiple times.

## Components

Table 1 lists the PSoC Creator Components used in this example and the hardware resources used by each Component.

Table 1. PSoC Creator Components

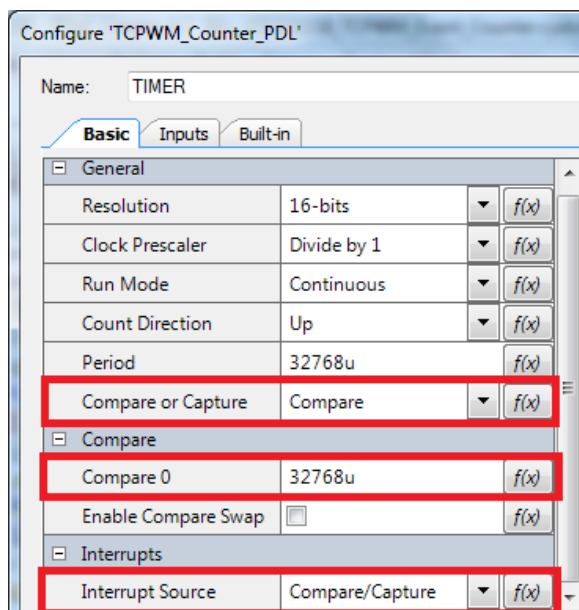
| Component                             | Instance Name                         | Hardware Resources   |
|---------------------------------------|---------------------------------------|--|
| Timer / Counter (TCPWM_Counter_PDL)   | TIMER, EVENT_COUNTER                  | Two Timer Counter Pulse Width Modulation peripheral blocks |
| UART (SCB_UART_PDL)                   | UART                                  | Single SCB peripheral block                                |
| Clock (SysClk_PDL)                    | Clock_1                               | A single clock divider                                     |
| System Interrupt (SysInt)             | TenSeconds_ISR                        | One entry in the device interrupt vector table             |
| General-Purpose Input / Output (GPIO) | ERROR_RED_LED, SW2, SW2_Out, CountPin | Four physical pins   |

## Parameter Settings

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

Figure 3 shows the TIMER Component parameter settings.

Figure 3. TIMER Component Parameter Settings



Configure 'TCPWM\_Counter\_PDL'

Name: TIMER

**Basic** Inputs Built-in

**General**

|                    |             |      |
|--------------------|-------------|------|
| Resolution         | 16-bits     | f(x) |
| Clock Prescaler    | Divide by 1 | f(x) |
| Run Mode           | Continuous  | f(x) |
| Count Direction    | Up          | f(x) |
| Period             | 32768u      | f(x) |
| Compare or Capture | Compare     | f(x) |

**Compare**

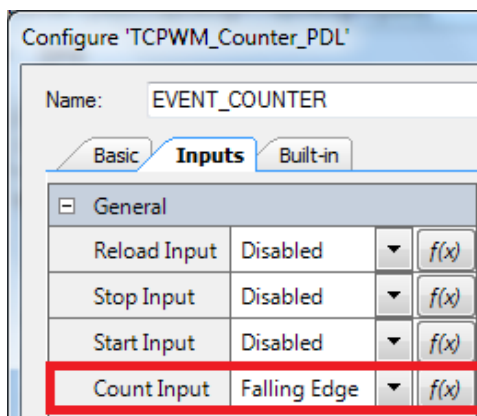
|                     |                          |      |
|---------------------|--------------------------|------|
| Compare 0           | 32768u                   | f(x) |
| Enable Compare Swap | <input type="checkbox"/> | f(x) |

**Interrupts**

|                  |                 |      |
|------------------|-----------------|------|
| Interrupt Source | Compare/Capture | f(x) |
|------------------|-----------------|------|

Figure 4 shows the EVENT\_COUNTER Component parameter settings.

Figure 4. EVENT\_COUNTER Component Parameter Settings



Configure 'TCPWM\_Counter\_PDL'

Name: EVENT\_COUNTER

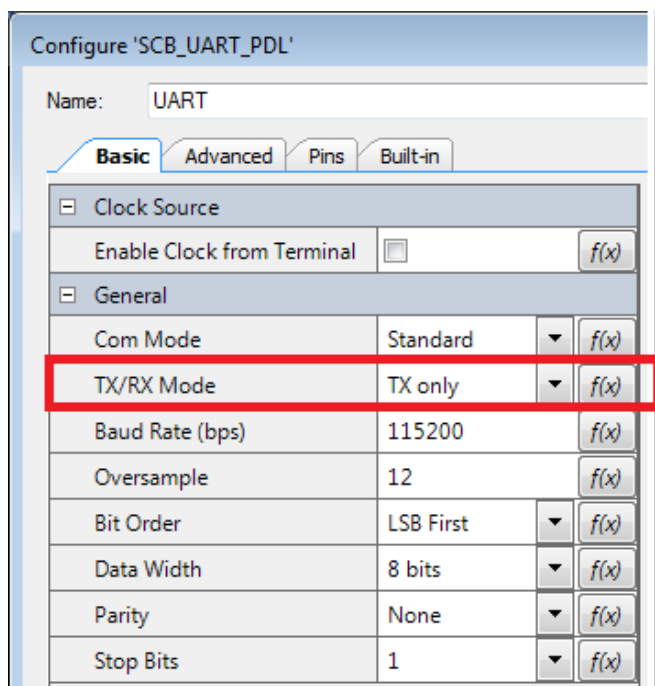
**Basic** **Inputs** Built-in

**General**

|              |              |      |
|--------------|--------------|------|
| Reload Input | Disabled     | f(x) |
| Stop Input   | Disabled     | f(x) |
| Start Input  | Disabled     | f(x) |
| Count Input  | Falling Edge | f(x) |

Figure 5 shows the UART Component parameter settings.

Figure 5. UART Component Parameter Settings



Configure 'SCB\_UART\_PDL'

Name: UART

**Basic** Advanced Pins Built-in

☐ Clock Source

Enable Clock from Terminal ☐ f(x)

☐ General

Com Mode Standard ▼ f(x)

**TX/RX Mode TX only ▼ f(x)**

Baud Rate (bps) 115200 f(x)

Oversample 12 f(x)

Bit Order LSB First ▼ f(x)

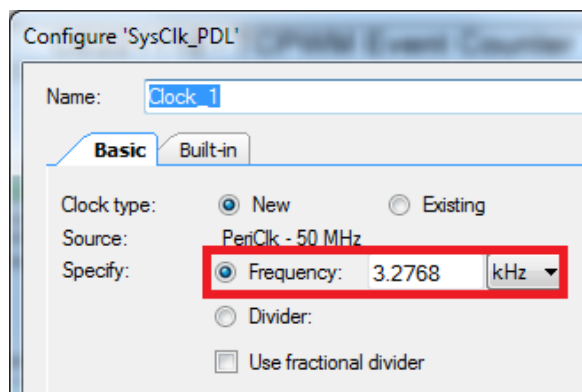
Data Width 8 bits ▼ f(x)

Parity None ▼ f(x)

Stop Bits 1 ▼ f(x)

Figure 6 shows the Clock\_1 Component parameter settings.

Figure 6. Clock\_1 Component Parameter Settings



Configure 'SysClk\_PDL'

Name: Clock\_1

**Basic** Built-in

Clock type: ☒ New ☐ Existing

Source: PeriClk - 50 MHz

Specify: **☒ Frequency: 3.2768 kHz ▼**

☐ Divider:

☐ Use fractional divider

## Design-Wide Resources

Table 2 shows the pin assignment for the code example.

Table 2. Pin Names and Locations

| Pin Name      | Location |
|---------------|----------|
| UART:tx       | P5[1]    |
| CountPin      | P10[0]   |
| ERROR_RED_LED | P0[3]    |
| SW2           | P0[4]    |

| Pin Name | Location |
|----------|----------|
| SW2_Out  | P10[1]   |

## Related Documents

|   |   |
|---|---|
| <b>Application Notes</b>  |   |
| <a href="#">AN210781</a> Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity | Describes PSoC 63 with Bluetooth Low Energy (BLE) Connectivity and how to build your first PSoC Creator project |
| <b>PSoC Creator Component Datasheets</b>  |   |
| <a href="#">TCPWM</a>   | Supports 16- or 32-bit Timer/Counter  |
| <a href="#">Clock</a>   | Supports programmable clock dividers  |
| <a href="#">System Interrupt</a>  | Interrupt vectoring and control   |
| <a href="#">UART</a>  | Supports UART communication   |
| <a href="#">General-Purpose Input / Output</a>  | Supports Analog, Digital I/O and Bidirectional signal types   |
| <b>Device Documentation</b>   |   |
| <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>                            |
| <b>Development Kit (DVK) Documentation</b>  |   |
| <a href="#">CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit</a>  |   |

## Document History

Document Title: CE221118 – PSoC 6 MCU TCPWM Event Counter

Document Number: 002-21118

| Revision | ECN     | Orig. of Change | Submission Date | Description of Change |
|----------|---------|-----------------|-----------------|-----------------------|
| **       | 5894785 | VJYA            | 11/03/2017      | 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="#">cypress.com/arm</a>        |
| Automotive                    | <a href="#">cypress.com/automotive</a> |
| Clocks & Buffers              | <a href="#">cypress.com/clocks</a>     |
| Interface                     | <a href="#">cypress.com/interface</a>  |
| Internet of Things            | <a href="#">cypress.com/iot</a>        |
| Memory                        | <a href="#">cypress.com/memory</a>     |
| Microcontrollers              | <a href="#">cypress.com/mcu</a>        |
| PSoC                          | <a href="#">cypress.com/psoc</a>       |
| Power Management ICs          | <a href="#">cypress.com/pmic</a>       |
| Touch Sensing                 | <a href="#">cypress.com/touch</a>      |
| USB Controllers               | <a href="#">cypress.com/usb</a>        |
| Wireless Connectivity         | <a href="#">cypress.com/wireless</a>   |

## PSoC® Solutions

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

## Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [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, 2017. 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. 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.