**Please note that Cypress is an Infineon Technologies Company.**
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

**Continuity of document content**
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

**Continuity of ordering part numbers**
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

# CE224593 – PSoC 4 PWM

## Objective

This example shows how to use the PSoC® Creator™ Timer Counter Pulse Width Modulator (TCPWM) Component configured as a PWM in a PSoC 4 device.

## Requirements

**Tool:** PSoC Creator 4.2

**Programming Language:** C (Arm® GCC 5.4.1)

**Associated Parts:** PSoC 4 family

**Related Hardware:** CY8CKIT-042 PSoC 4 Pioneer Kit

## Overview

This example contains two projects that use the PWM Component. The TCPWM_PWM_LED_RateSwap project demonstrates how PWM drives an LED with the option of changing the blink rate from slow to fast and vice versa. The TCPWM_PWM_ThreePhase project demonstrates three LEDs driven by three PWMs that are 120 degrees out-of-phase from one another.

## Hardware Setup

No hardware setup is required for this code example.

## Software Setup

No software setup is required for this code example.

## Operation

1. Connect the USB cable between the PC and the PSoC 4 Pioneer Kit.
2. Build the project and program it into the PSoC 4 device. Choose **Debug** > **Program**. For more information on device programming, see PSoC Creator Help.
3. For the TCPWM_PWM_LED_RateSwap project: Each time SW2 is pressed, you will observe the LED's blink rate change.
4. For the TCPWM_PWM_ThreePhase project: The LED will cycle through colors to show the PWMs are 120 degrees out-of-phase.

# Design and Implementation

There are two projects in this example:

## TCPWM_PWM_LED_BlinkRateSwap

The TCPWM_PWM_LED_BlinkRateSwap example performs the following functions:

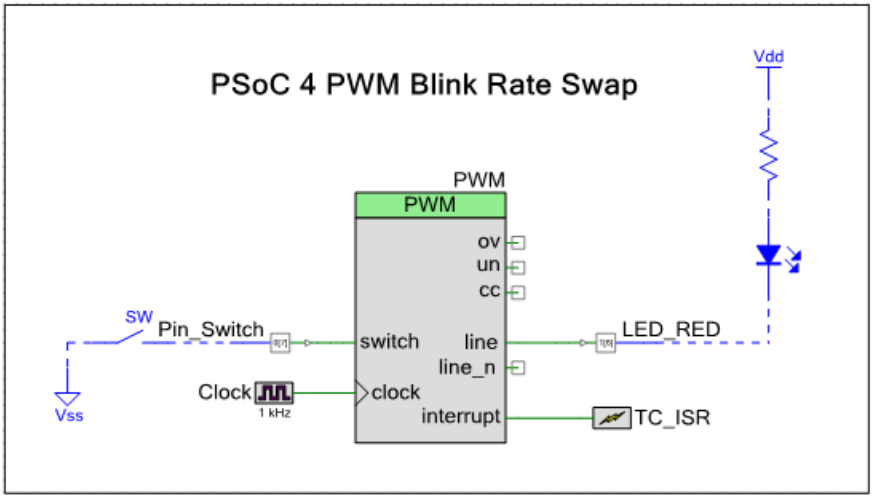
1. Configures the `TC_ISR_Handler` PWM interrupt handler function.
2. Starts PWM.
3. Initiates an interrupt when the count reaches terminal count.
4. When the switch (controlled by Pin_Switch) goes HIGH, the code example swaps the period register values. The period value controls the blink rate of LED_RED.

The `TC_ISR_Handler` function does the following:

1. Clears the interrupt for terminal count.
2. Toggles the LED ON/OFF state.

The top-level schematic of the PSoC Creator project is shown in Figure 1.

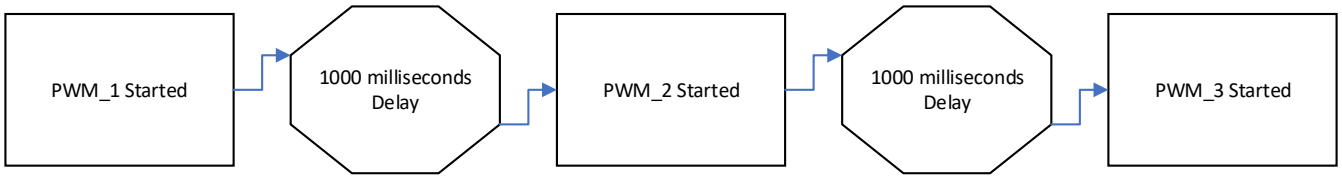
Figure 1. TCPWM_PWM_LED_RateSwap




## TCPWM_PWM_ThreePhase

Figure 2 shows the functions performed in the TCPWM_PWM_ThreePhase example.

Figure 2. Functions of TCPWM_PWM_ThreePhase




The top-level schematic of the PSoC Creator project is shown in Figure 3.

The delay between the start of PWMs creates the 120 degrees out-of-phase alignment in the waveform shown in Figure 4.

The delay is determined by 1/3 (120 out of 360 degrees) the period of PWM: 1/3*3000 = 1000 ms delay.

Note that varying the PWM period also varies the LED blink rate.

Figure 3. TCPWM_PWM_ThreePhase

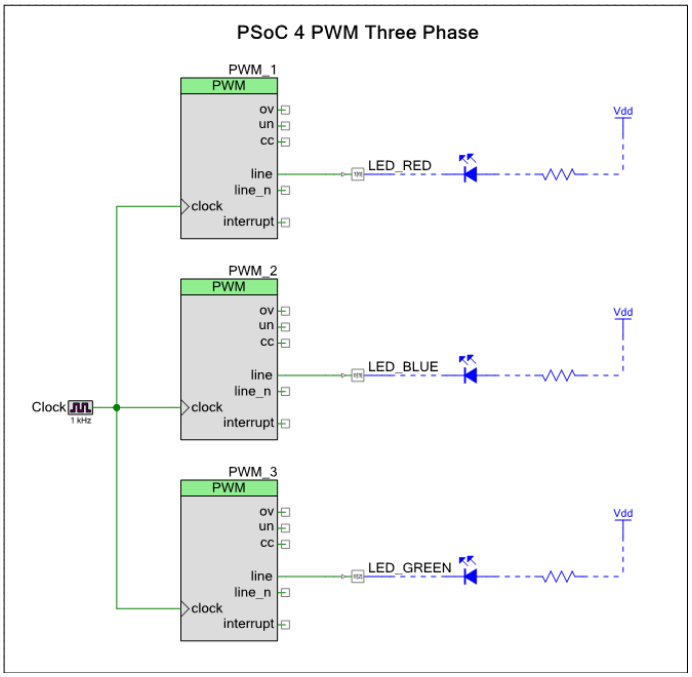
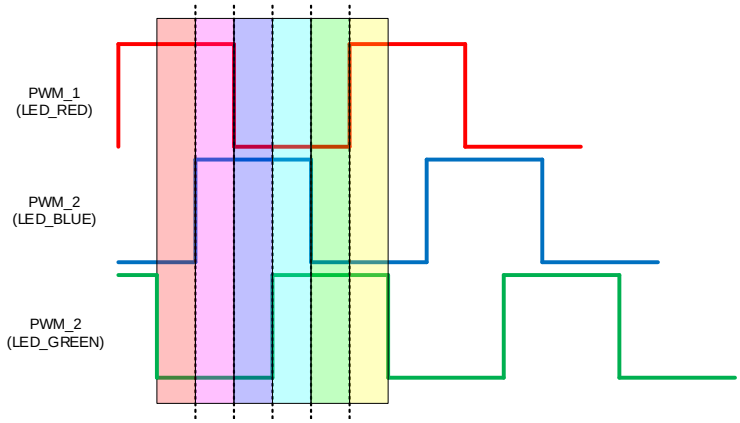



Figure 4. Expected LED Pattern (Red, Violet, Blue, Cyan, Green, Yellow)



## Components and Settings

Table 1 lists the PSoC Creator Components used in used in this example, how they are used in the design, and the non-default settings required so they function as intended.

Table 1. PSoC Creator Components

| Component | Instance Name | Purpose | Non-default Settings |
|---|---|---|---|
| Digital Input Pin | Pin_Switch | Handle the SW2 connection on device | See Figure 5 |
| PWM (TCPWM mode) [v2.10] | PWM | Handle the PWM operation | See Figure 6 |
| Digital Output Pin | LED_RED | Handle any LED color on device | Make sure external terminal is checked. |
| PWM (TCPWM mode) [v2.10] | PWM_1 (_2 & _3) | Handle the PWM operation | See Figure 7 |

Figure 5. Pin_Switch Parameter Settings

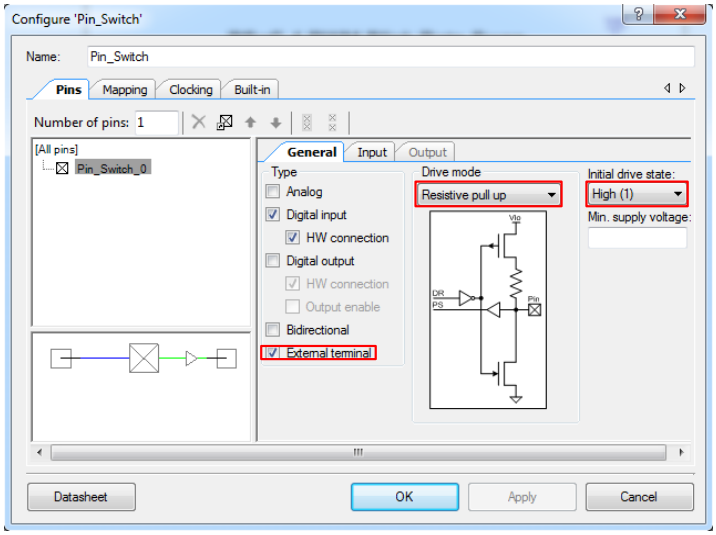


Figure 6. PWM Parameter Settings

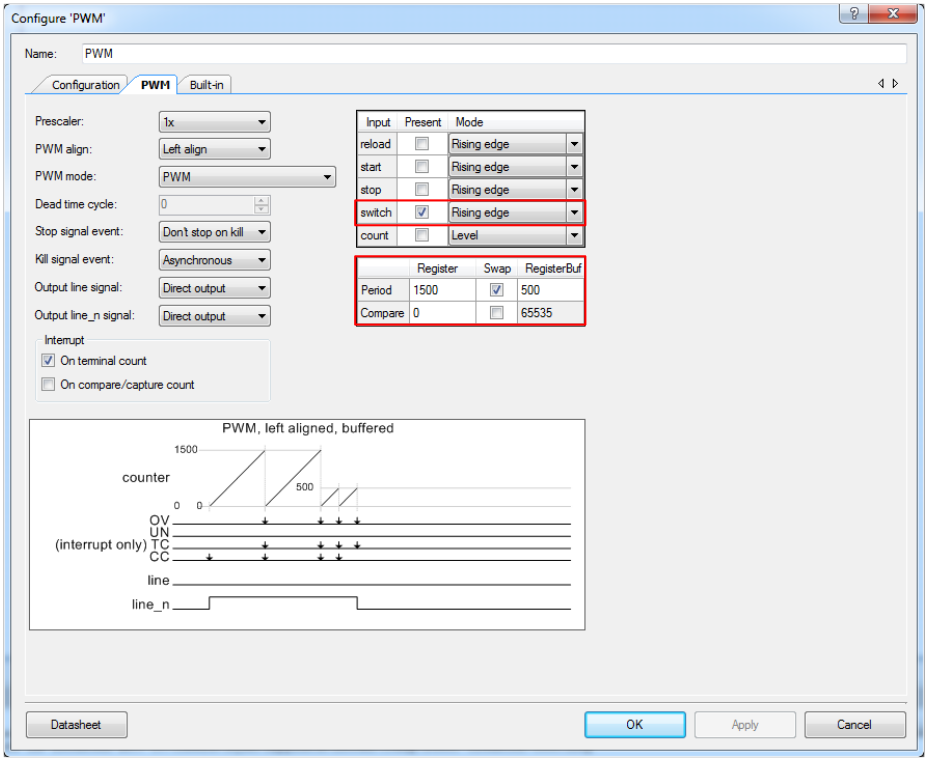
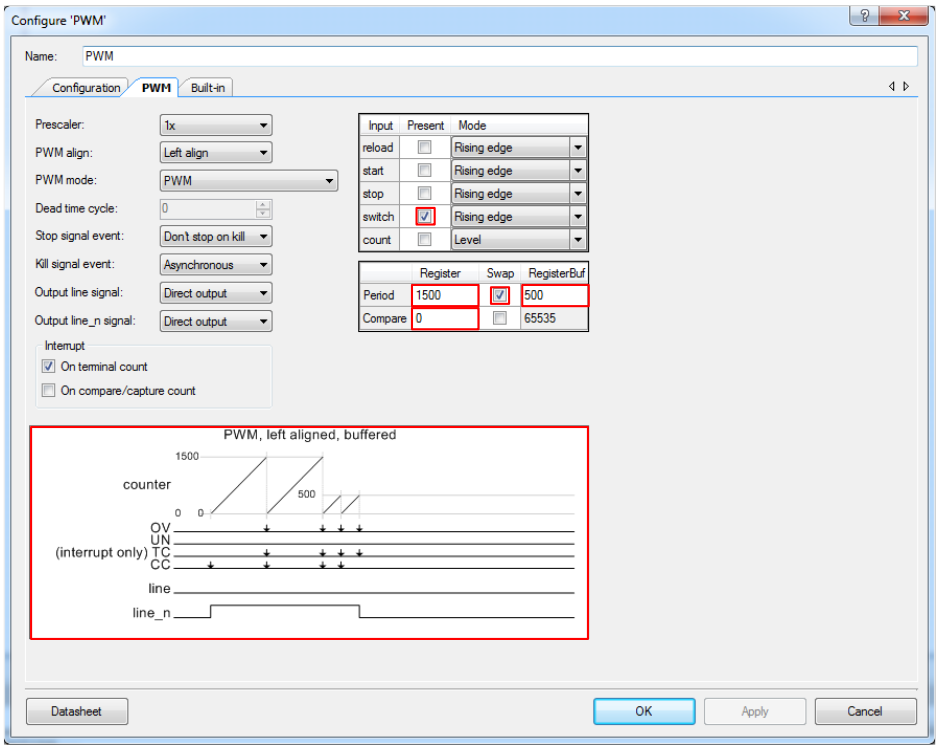



Figure 7. PWM Parameter Settings




For information on the hardware resources used by a Component, see the Component datasheet.

## Reusing this Example

This example is designed for the CY8CKIT-042 pioneer kit. To port this design to a different PSoC 4 device, kit, or both, do the following:

1. In PSoC Creator, select **Project** > **Device Selector** to change the target device. Select your device as listed in Table 2.
2. Make sure that the **SysClk Desired frequency** is set to 24 MHz after the device is changed.
3. In the PSoC Creator Workspace Explorer, select the **Clocks** interface listed under **Design Wide Resources**.
4. Set the **SysClk Desired Frequency** to 24 MHz, if it is not already.

Table 2. Development Kits and Associated Devices

| Development Kit | Device |
|---|---|
| CY8CKIT-041 | CY8C4146AZI-S433 |
| CY8CKIT-042 | CY8C4245AXI-483 |
| CY8CKIT-042-BLE | CY8C4247LQI-BL483 |
| CY8CKIT-044 | CY8C4247AZI-M485 |
| CY8CKIT-046 | CY8C4248BZI-L489 |
| CY8CKIT-048 | CY8C4A45AZI-483 |

In some cases, a resource used by a code example (for example, a Universal Digital 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 device supports.

## Related Documents

| | |
|---|---|
| **Application Notes** | |
| AN79953 Getting Started with PSoC® 4 | Describes PSoC 4 devices and shows how to build the associated code example |
| **Code Examples** | |
| CE224564 PSoC 4 TCPWM Counter/Timer | Demonstrates the use of a counter to keep track of the number of button presses to measure the frequency and duty cycle of an input waveform. |
| CE224595 PSoC 4 TCPWM QuadDec | Demonstrates the use of a Quadrature Decoder to detect the rotational direction as one waveform leads the other. Two PWMs are used to simulate input waveforms. |
| **PSoC Creator Component Datasheets** | |
| TCPWM | A multifunctional Component that can implement the following functionalities: PWM, Timer/Counter, and Quadrature Decoder. |
| General Purpose Input/Output (GPIO) | A multifunctional Component that allows hardware resources to connect to a physical port-pin and provides access to external signals through an appropriately configured physical I/O pin. |
| Interrupt | The Interrupt Component defines hardware-triggered interrupts. There are three types of system interrupt waveforms that can be processed by the interrupt controller: Level, Pulse, and Edge. |
| **Device Documentation** | |
| PSoC 4 Datasheets | PSoC 4 Technical Reference Manuals |
| **Development Kit (DVK) Documentation** | |
| CY8CKIT-042 PSoC® 4 Pioneer Kit | |
| PSoC 4 Kits | |
| **Tool Documentation** | |
| PSoC Creator | Go to the **Downloads** tab for Quick Start and User Guides |

# Document History

Document Title: CE224593 – PSoC 4 PWM

Document Number: 002-24593

| Revision | ECN | Orig. of Change | Submission Date | Description of Change |
|---|---|---|---|---|
| ** | 6290692 | SYAO | 08/31/2018 | New code example |
| *A | 6672859 | NRSH | 09/13/2019 | Updated document layout and small changes for clarity. |

## 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 | Code Examples | 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 ("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 shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. 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. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, 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 any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

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.