

# TCPWM (QuadDec mode) example project

2.0

## Features

- Project uses TCPWM component with Quadrature Decoder mode configuration
- Indicate capture interrupts on LED

## General Description

This example project demonstrates the TCPWM component usage in the Quadrature Decoder mode.

## Development kit configuration

1. Use the CY8CKIT-042 Kit with the default configuration and the CY8CKIT-040 Kit with changed project configuration settings.
2. Connect P2[0] to P0[1].
3. Connect the logic low (Gnd) or the logic high (3.3V) signal to P0[0].
4. Build the project and program the hex file on to the target device.
5. Power cycle the device and observe the results on the LEDs.

In order to configure the project for CY8CKIT-040 the following steps should be performed:

1. Change the project's device from PSoC 4200 to PSoC 4000.  
Use Device Selector from the project's context menu.
2. Change assignment of the pin component to physical pin.

In the Workspace Explorer window, double-click the project's design-wide resource file and assign the pin for LED\_GREEN to P1[1] (see Table 1).

Table 1. Pin assignment of QuadDecExample project

Pin Name	Development Kit	
	CY8CKIT-042	CY8CKIT-040
LED_GREEN	P0[2]	P1[1]
PhiA_Out	P2[0]	P2[0]
PhiA_In	P0[1]	P0[1]
PhiB_In	P0[0]	P0[0]

## Project configuration

The example project consists of the following components: TCPWM, Clock, two digital output pins, two digital input pins, and Interrupt. The TCPWM is used as the Quadrature Decoder mode with the 1x Decoding mode. One of the output pins is used for PhiA signal generation. Another output pin is used to reflect a capture event when interrupts happen. The top design schematic is shown in Figure 1.

### Parameters used:

- Quadrature Decoder mode
- Encoding mode = 1x
- Interrupt mode = Capture

The TCPWM (Quadrature Decoder mode) datasheet example project

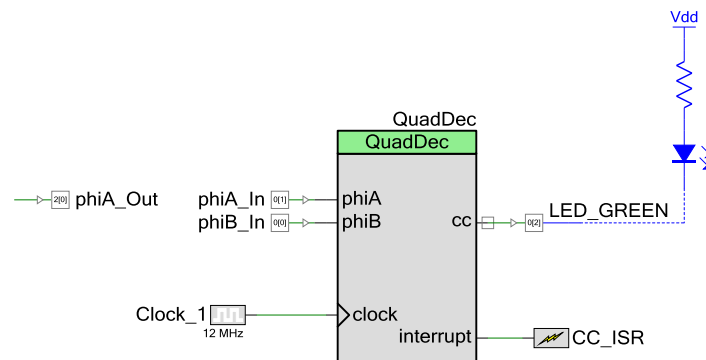


Figure 1. Top design schematic.

The TCPWM component GUI configuration (Figure 2, Figure 3):

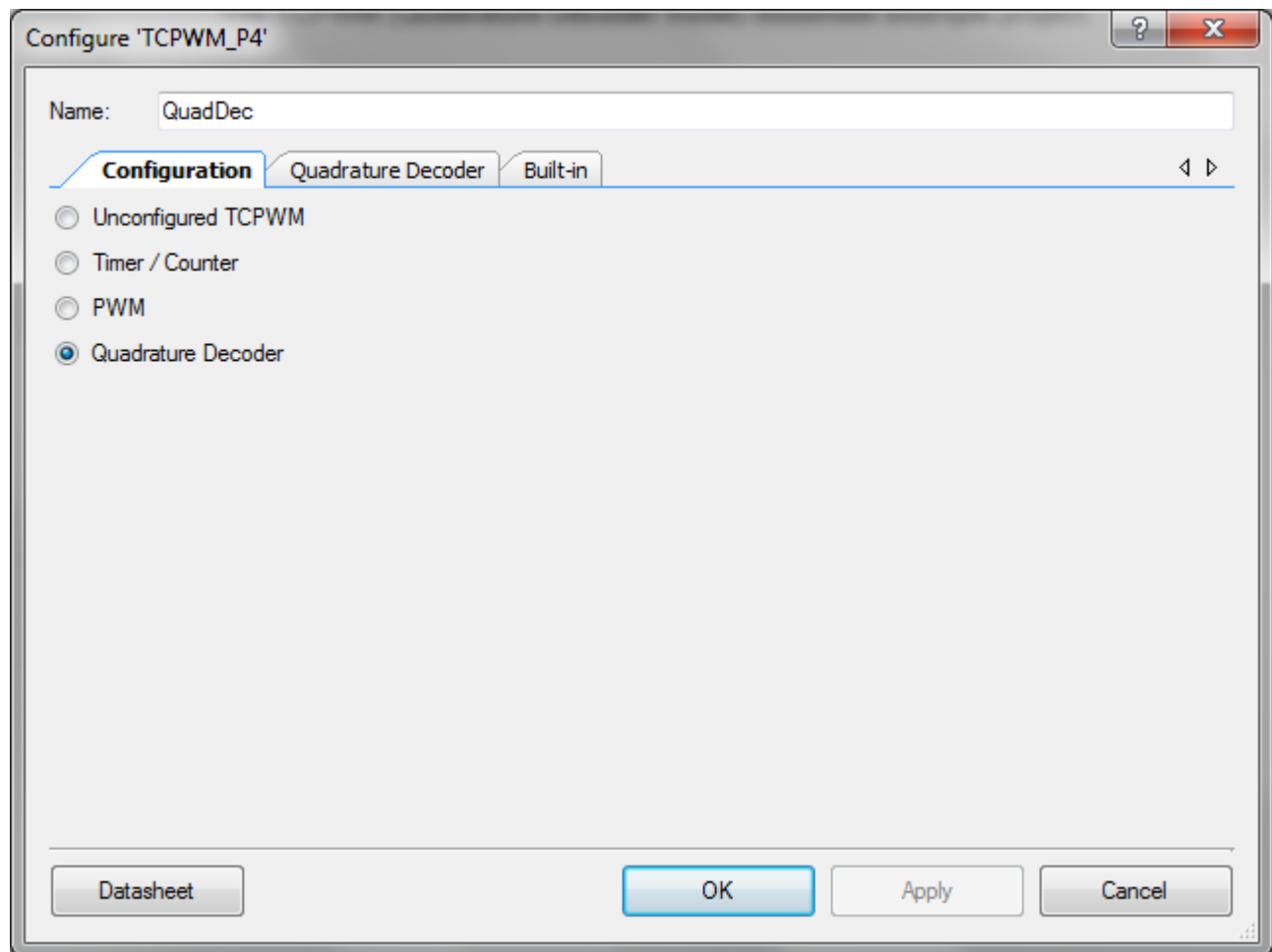


Figure 2. TCPWM Component Configuration Tab

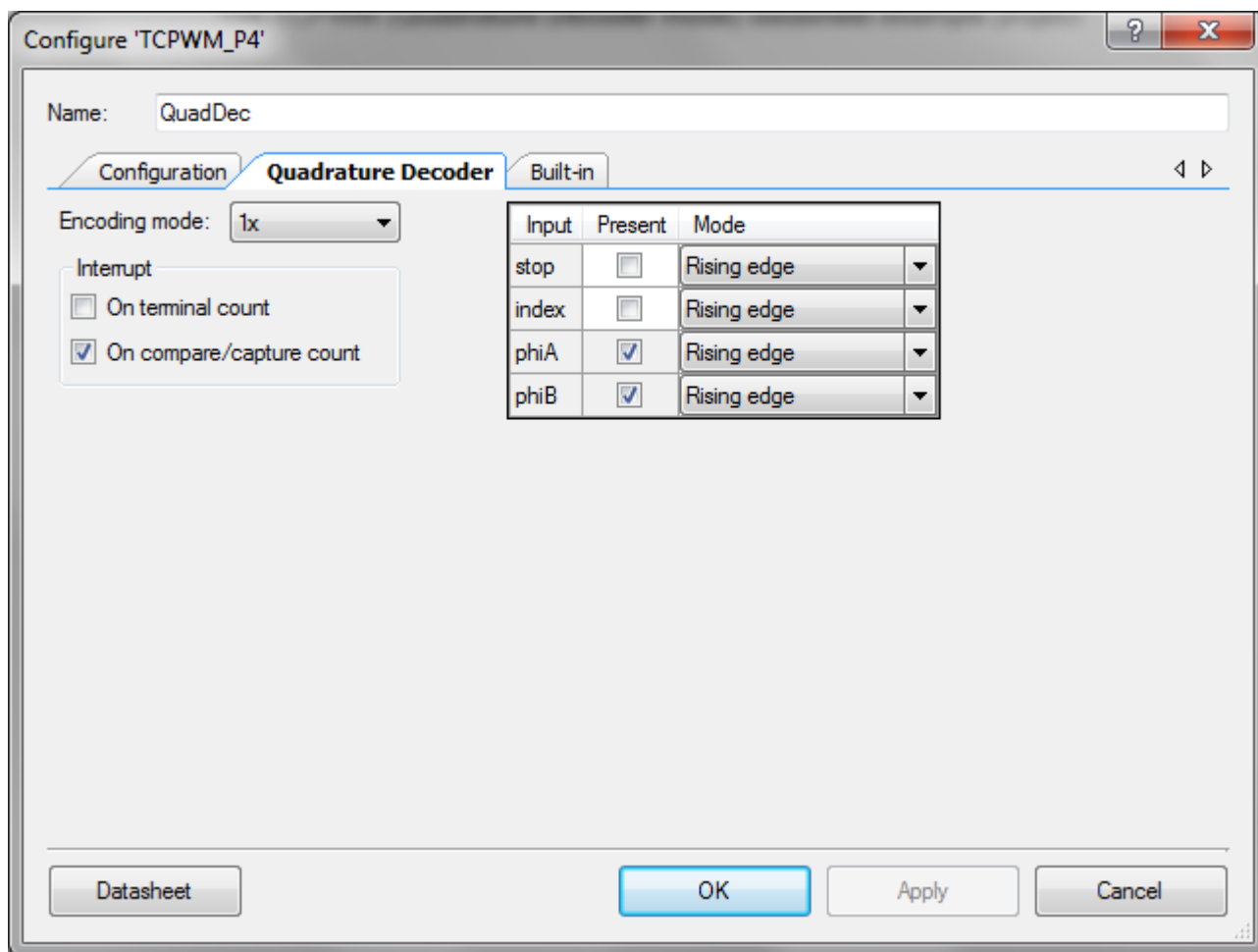


Figure 3. The TCPWM Component Quadrature Decoder Tab

## Project description

The counter is initialized with the mid-point counter value (0x8000) on an index event. A positive edge on phiA increments the counter when phiB is 0 and decrements the counter when phiB is 1. In the project, the TCPWM counts from 0x8000u down to 0u or from 0x8000 to 0xFFFFu. If the counter reaches the 0xFFFFu value, the interrupt happens, the green color LED blinks for 200ms, and the counter is initialized with the mid-point value (0x8000u).

The index event is generated by the API (QuadDec\_TriggerCommand(QuadDec\_MASK, QuadDec\_CMD\_RELOAD)).

The phiA signal is generated by the PhiAbGeneration function in the main.c file.

If the phiB signal connected to the logic low signal, a simulation is emulated, when the shaft is rotating in the clockwise direction (A leads B). If the phiB signal is connected to the logic high signal, a situation is emulated, when the shaft is rotating in the counter-clockwise direction (B leads A).

## Expected results

The green color LED is blinking when the counter value is captured (0xFFFF).

© Cypress Semiconductor Corporation, 2009-2014. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.