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.

CE224594 – PSoC 4 Timer/Counter

Objective

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

Overview

This example contains three projects that use the Timer Counter Component. The Counter_Count project demonstrates the Timer to keep track of the number of button presses with LED to show the count changing. The Counter_Frequency_DutyCycle project measures the frequency and duty cycle of an input waveform using counter mode and prints the results over UART. The Counter_Periodic_Interrupt uses the timer mode to create periodic interrupts that blink an LED.

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

Hardware Setup

This code example is set up for CY8CKIT-042. If you are using a different kit, see .

Reusing this Example.

For CY8CKIT-042, the USB-UART bridge in KitProg2 module is used:

1. Connect the \UART:rx\ pin P0[4] to P12[7] on header J8.
2. Connect the \UART:tx\ pin P0[5] to P12[6] on header J8.

Other kits use different pins for the UART. Make sure that you select the pins that are right for your kit.

Software Setup

This design requires a terminal emulator such as PuTTY or Tera Term running on your computer.

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. Open a terminal emulator on your computer and configure the program to the appropriate COM port. Configure the baud rate to 115200, data bits to 8, no parity bits, stop bit as 1, and no control flow.
4. For Counter_Count project: Press the kit button SW2 and confirm that the count displayed on the terminal increments accordingly.
5. For Counter_Frequency_DutyCycle project: Change the period and compare register settings of the PWM to create different periods and duty cycles. Confirm that the frequency and duty cycle are correctly displayed on the terminal.
    a. An alternative to using the PWM is to use a digital input pin and feed in your own input waveform. Note that this requires an inverter so that Timer_2 can use the inverse of the input waveform.
6. For Counter_Interrupt project: confirm that the red LED turns on and off every two seconds. No terminal emulator is needed for this project

Design and Implementation

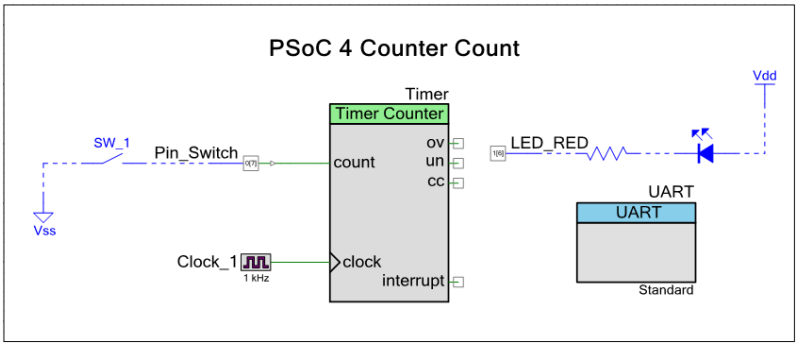
Counter_Count

In the Counter_Count example, the following functions are performed:

1. The Counter and UART Components start.
2. When the Timer Component detects a change in count, the LED flashes, and the UART displays a new count value.

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

Figure 1. Schematic, Counter_Count



Counter_Frequency_DutyCycle

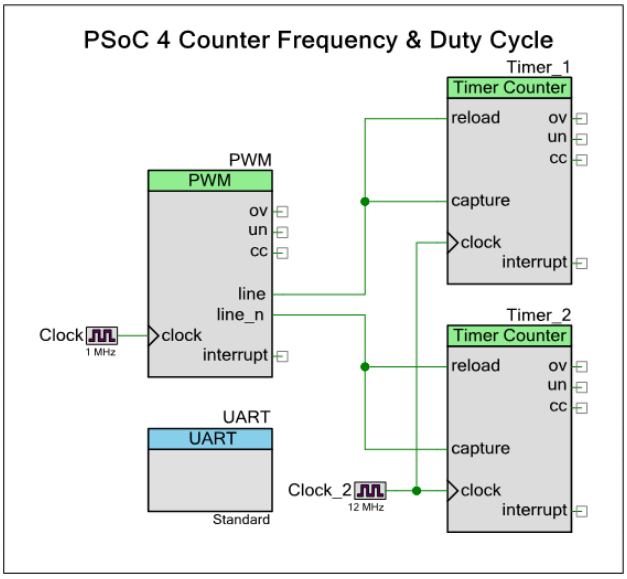
In the Counter_Frequency_DutyCycle example, the following functions are performed:

1. The Timer Counter, PWM, and UART Components start.
2. Timer 1 and Timer 2 capture both counts for the high and low parts of the waveform.
   a. Using these values, the period, frequency, and duty cycle are calculated and displayed.

Period is calculated by summing the total count for when the waveform is HIGH (From Timer_1) and when the waveform is LOW (From Timer_2). Frequency is calculated by dividing Timer Clock speed by Period. Duty Cycle is calculated by dividing total count for when the waveform is HIGH by the period.

The top-level schematic of the PSoC Creator project is shown in Figure 2.

Figure 2. Schematic, Counter_Frequency_DutyCycle



Counter_Periodic_Interrupt

In the Counter_Periodic_Interrupt example, the following functions are performed:

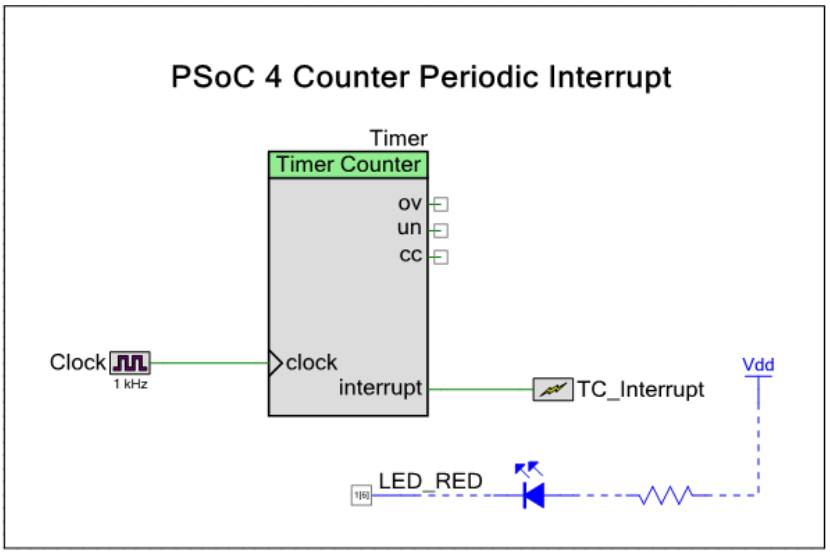
1. The Timer Counter Component is started
2. The timer counter interrupt handler function TC_InterruptHandler is configured.
3. An interrupt occurs when the timer's count reaches the terminal count, which is determined by the period set for timer.

The ISR_Timer 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 3.

Figure 3. Schematic, Counter_Periodic_Interrupt



Components and Settings

Table 1 lists the PSoC Creator Components 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
Timer Counter (TCPWM mode) [v2.10]	Timer	Handle the Timer/Counter operation	See Figure 4
Timer Counter (TCPWM mode) [v2.10]	Timer_1 & _2	Handle the Timer/Counter operation	See Figure 5
Timer Counter (TCPWM mode) [v2.10]	Timer (for periodic interrupt project)	Handle the Timer/Counter operation	Set period register to 2000.
PWM (TCPWM mode) [v2.10]	PWM	Handle the PWM operation	Set period register to 4999. Set compare register to 2499.
UART (SCB mode) [v4.0]	UART	Handle UART communication	None
Digital Input Pin	Pin_Switch	Handle the SW2 connection on device	See Figure 6

Figure 4. Timer Parameter Settings

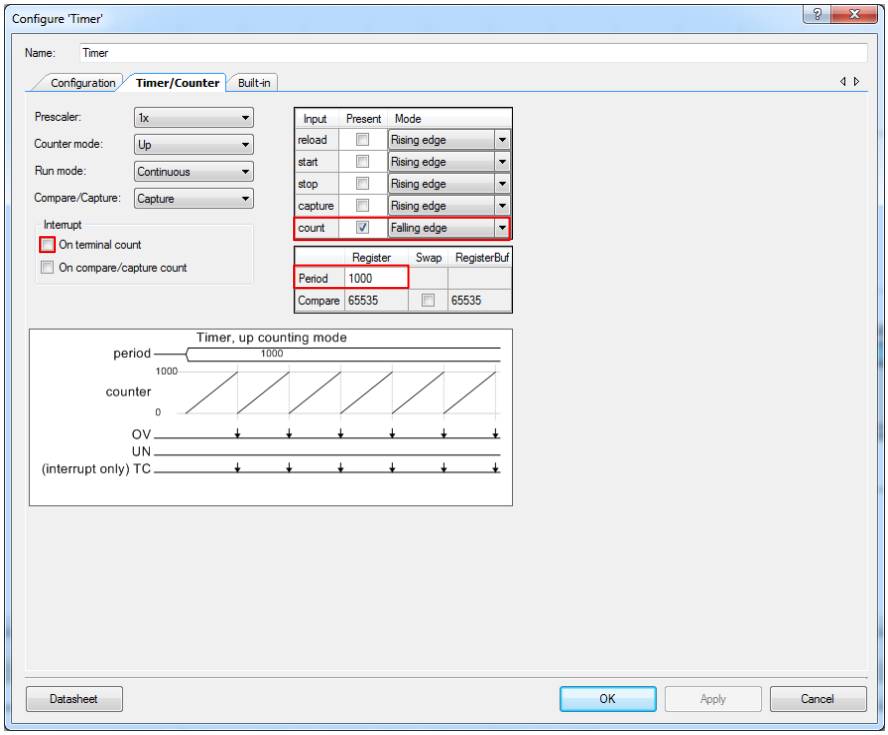


Figure 5. Tiimer_1 (2) Parameter Settings

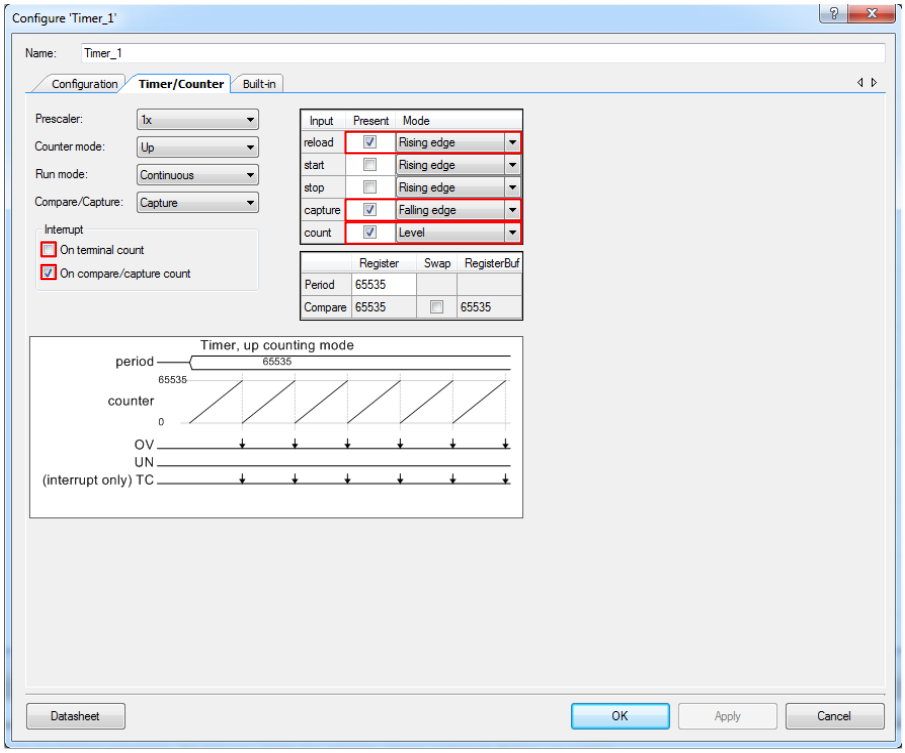
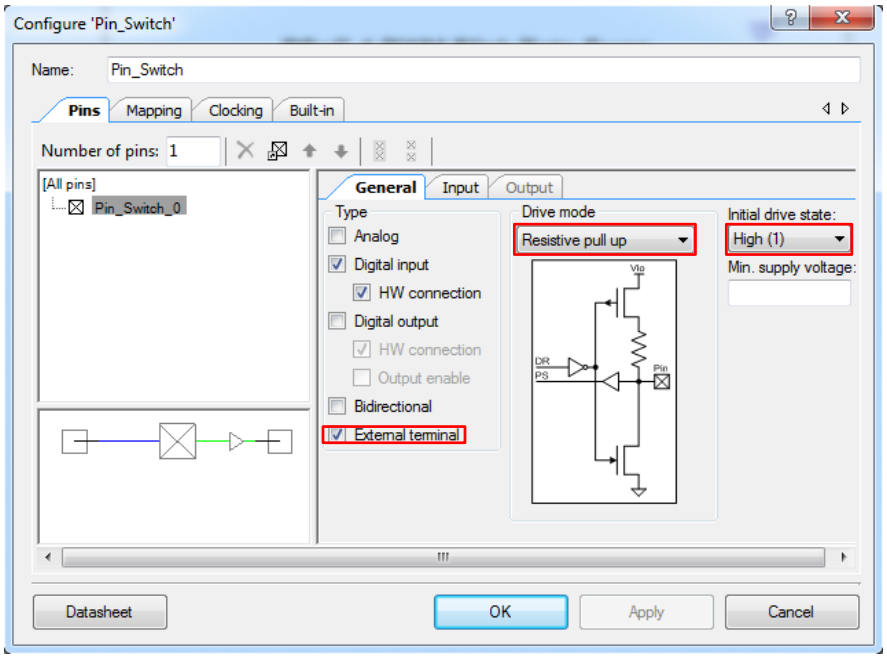


Figure 6. Pin_Switch Parameter Settings



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

5. Route \UART:tx\ and \UART:rx\ to the pins listed in Table 3. For the CY8CKIT-048, install jumper wires for \UART:tx\ and \UART:rx\ to P12[6] and P12[7] on header J16, respectively.

Table 3. Pin Assignments for Different Kits

Pin Name	Development Kit					
	CY8CKIT-041	CY8CKIT-042	CY8CKIT-042-BLE	CY8CKIT-044	CY8CKIT-046	CY8CKIT-048
\UART:rx\	P0[4]	P0[4]	P1[4]	P7[0]	P3[0]	P0[4]
\UART:tx\	P0[5]	P0[5]	P1[5]	P7[1]	P3[1]	P0[5]

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 device supports.

Related Documents

Application Notes	
AN79953 Getting Started with PSoC® 4	Describes PSoC 4 and shows how to build the attached code example
Code Examples	
CE224593 PSoC4 TCPWM PWM	Demonstrates PWM driving an LED with changing blink rates and also three PWM 120 degrees out of phase from each other driving LEDs.
CE224595 PSoC 4 TCPWM QuadDec	Demonstrates the use of QuadDec to detect the direction of count and direction status is displayed using a LED. Two PWMs are used to simulate the rotational direction as one leads the other.
PSoC Creator Component Datasheets	
TCPWM	Multifunctional component that can implement the following functionalities: PWM, Timer/Counter, and Quadrature Decoder.
SCB	A multifunction hardware block that implements the following communication Components: I2C, SPI, UART, and EZI2C
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 IO 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 Manual
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: CE224594 – PSoC 4 Timer/Counter

Document Number: 002-24594

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6371623	SYAO	12/20/2018	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 Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709