

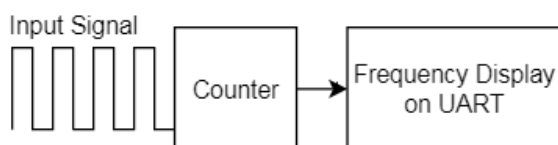
Objective

This code example demonstrates how to use the TCPWM Component in PSoC® 6 MCU to measure the frequency of a periodic digital signal.

Overview

Frequency measurement is counting the number of edges (rising, falling, or both) that occur within a known time interval. To find this number, this project uses a one second time window and determines the number of counts (rising edges) within that time window.

Note: This project measures digital signals. If you want to measure the frequency of other waveforms, the waveform must be converted to a digital signal before inputting it to the counter.



This code example assumes that you are familiar with the PSoC 6 MCU device and the PSoC Creator™ IDE. If you are new to PSoC 6 MCU, see the application note [AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy \(BLE\) Connectivity](#).

Requirements

Tool: PSoC Creator™ 4.2

Programming Language: C (ARM® GCC 5.4.1, ARM MDK 5.22)

Associated Parts: PSoC 6 MCU

Related Hardware: [CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit](#)

Design

This example uses three TCPWM Components.

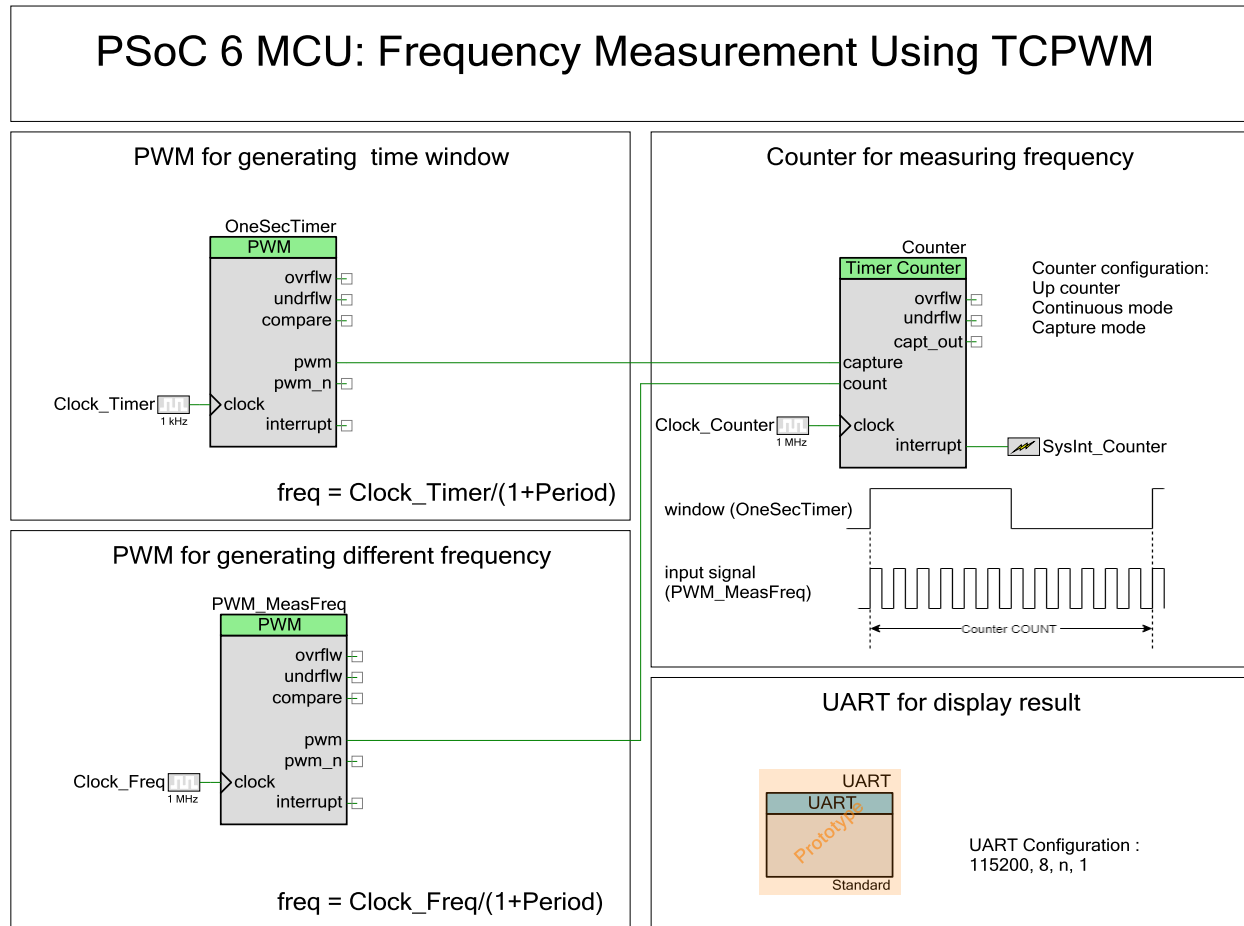
One is set up as a PWM (**OneSecTimer**), with a 1-kHz clock, and a count of 1000 (0-999). It generates a one-second time window.

A second is also set up as a PWM (**PWM_MeasFreq**) with a 1 MHz clock. Based on the “Period” and “Compare” values, it generates a signal at a particular frequency. By default, the Component is configured to generate a signal at 2500 Hz (2.5 KHz).

The third Component is set up as a counter (**Counter**). It receives the capture input from the OneSecTimer, and the input signal from the frequency generator. Once per second, it reads the count from the frequency generator. Firmware performs the calculation to identify the input frequency based on how the count changes each second.

A UART with baud rate of 115200 bps is configured to display the result in a terminal application.

Figure 1. PSoC Creator Schematic



Design Considerations

For a reliable design counter clock frequency should be more than 2x of the input signal. To measure an external signal, replace PWM_MeasFreq with a GPIO and provide the input signal to that GPIO pin.

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 1. PSoC Creator Component

Components	Instance name	Hardware Resources
Counter	Counter	1 TCPWM
PWM	OneSecTimer PWM_MeasFreq	2 TCPWM
Interrupt	SysInt_Counter	1 Interrupt
UART(SCB)	UART	1 SCB 2 GPIO Pin
Clock	Clock_Timer Clock_Freq Clock_Counter	3 peripheral clock dividers

Parameter Settings

Table 2 lists the non-default settings of all the components used in the design.

Table 2. Component Parameters

Component Instance Name	Settings (Non-Default)
OneSecTimer	[General tab] Period:999 Compare:500
PWM_MeasFreq	[General tab] Period:399u Compare:200u
Counter	[General tab] Capture input: Rising Edge Interrupt Source: Overflow/Underflow and Compare/Capture [Input tab] Count input: Rising edge
Clock_Timer	Frequency: 1 kHz
Clock_Freq	Frequency: 1 MHz
Clock_Counter	Frequency: 1 MHz

Design-Wide Resources

Table 3 shows the physical pins used.

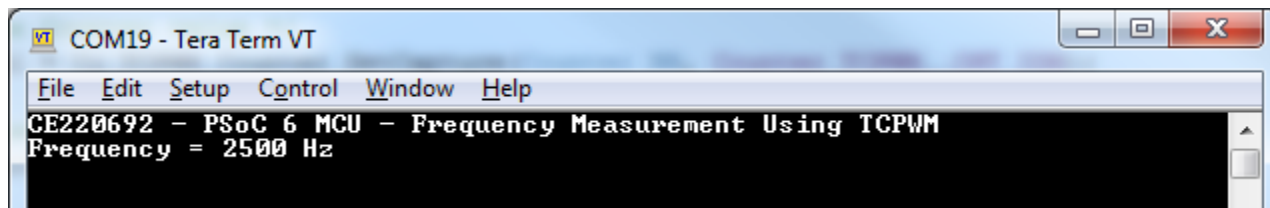
Table 3. Pin Names and Locations

Pin Name	Location
UART:rx	P5[0]
UART:tx	P5[1]

Operation

1. Plug the CY8CKIT-062 kit board into your computer's USB port.
2. Open a terminal software such as Tera Term and select the KitProg2's COM port with a baud rate setting of 115200 bps.
3. Build the project and program it into the PSoC 6 MCU device. Choose Debug > Program. For more information on device programming, see PSoC Creator Help. Flash for both CPUs is programmed in a single program operation.
4. On successful programming, the UART terminal displays the code example title and the frequency as shown in [Figure 2](#).

Figure 2. UART Terminal Displaying Starting Message and Result



5. To test the code example, change the **Period** and **Compare** values of the PWM_MeasFreq Component to generate a different frequency.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Introduction of PSoC 6 MCU with Bluetooth Low Energy (BLE)
PSoC Creator Component Datasheets	
UART	Provides asynchronous communication interface using SCB hardware
TCPWM	Supports configuration of the TCPWM hardware for Timer/Counter functionality
Clock	Supports local clock generation
Pins	Supports connection of hardware resources to physical pins
SysInt	Provides SysInt component settings
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE Pioneer Kit	

Document History

Document Title: CE220692 - PSoC 6 MCU: Frequency Measurement Using TCPWM

Document Number: 002-20692

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5896719	AJYA	09/26/2017	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	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](#)

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 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.