# CE220692 – PSoC 6 MCU: Frequency Measurement Using TCPWM

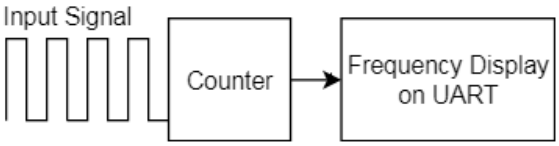
## Objective

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.

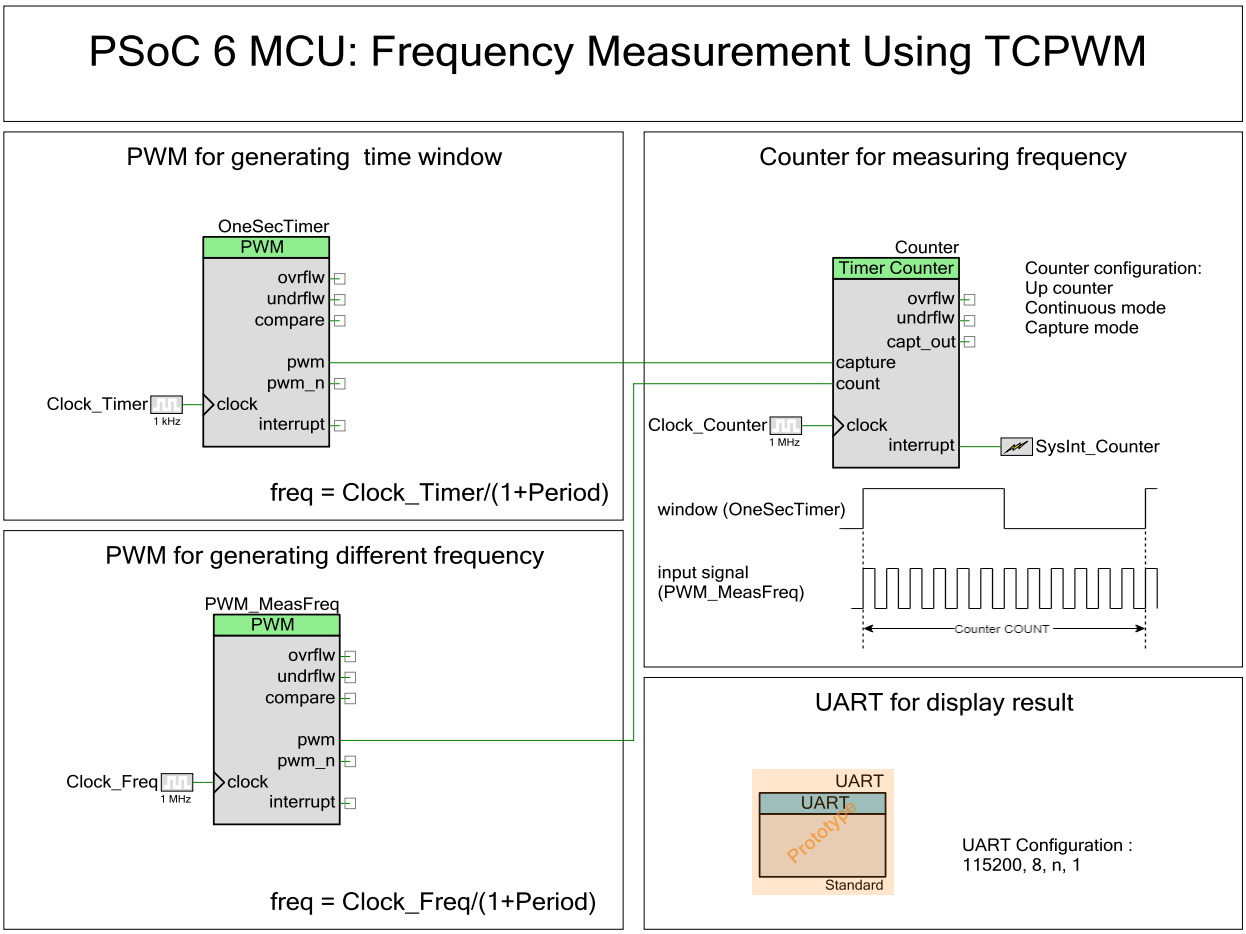
One is set up as a PWM (**OneSecTimer**), with a 1-kHz clock, and a count of 1000 (0-999). It generates a one-second time window.

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<br>PWM_MeasFreq | 2 TCPWM |
| Interrupt | SysInt_Counter | 1 Interrupt |
| UART(SCB) | UART | 1 SCB<br>2 GPIO Pin |
| Clock | Clock_Timer<br>Clock_Freq<br>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]**<br>Period:999<br>Compare:500 |
| PWM_MeasFreq | **[General tab]**<br>Period:399u<br>Compare:200u |
| Counter | **[General tab]**<br>Capture input: Rising Edge<br>Interrupt Source: Overflow/Underflow and Compare/Capture<br>[Input tab]<br>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.

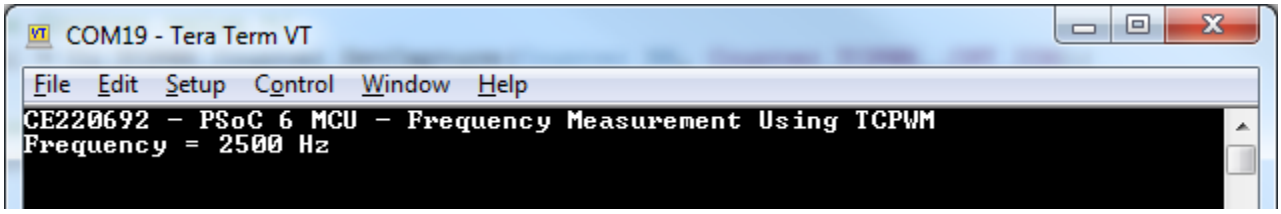
Table 3. Pin Names and Locations

| 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



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