Application Note

Capturing PWM 0% to 100% Duty Cycle with MSPM0 MCU

Chao Gao and Wayne Huang

ABSTRACT

This application note provides implementations to address applications that need to capture a PWM signal with a duty cycle ranging from 0% to 100%. When using the MSPM0 microcontroller (MCU) to capture this signal, the timer capture function is typically employed. However, the timer cannot capture the duty cycle at 0% or 100% because the timer relies on edge-triggered events, and these extreme duty cycles lack detectable edges.

Table of Contents

List of Figures

List of Tables

Trademarks

1 Introduction to MSPM0 Timer Capture and Compare Module

Based on the device, two types of timers are available: general-purpose timers (TIMG) and advanced control timers (TIMA). Both timers use a common timer architecture, which the capture function are the same. TIMG has up to two identical capture and compare blocks. TIMA has up to four identical capture and compare blocks present to support external or internal signals. However, TIMG14 can support up to four blocks as an exception.

Pulse width capture measures the high-time of a signal on CCP. The high time is the number of TIMCLK periods from rising edge to falling edge of the CCP input, and is useful for applications to measure the duty cycle of a PWM input signal. The counter is loaded at the positive edge and captured at the negative edge (capture event is generated).

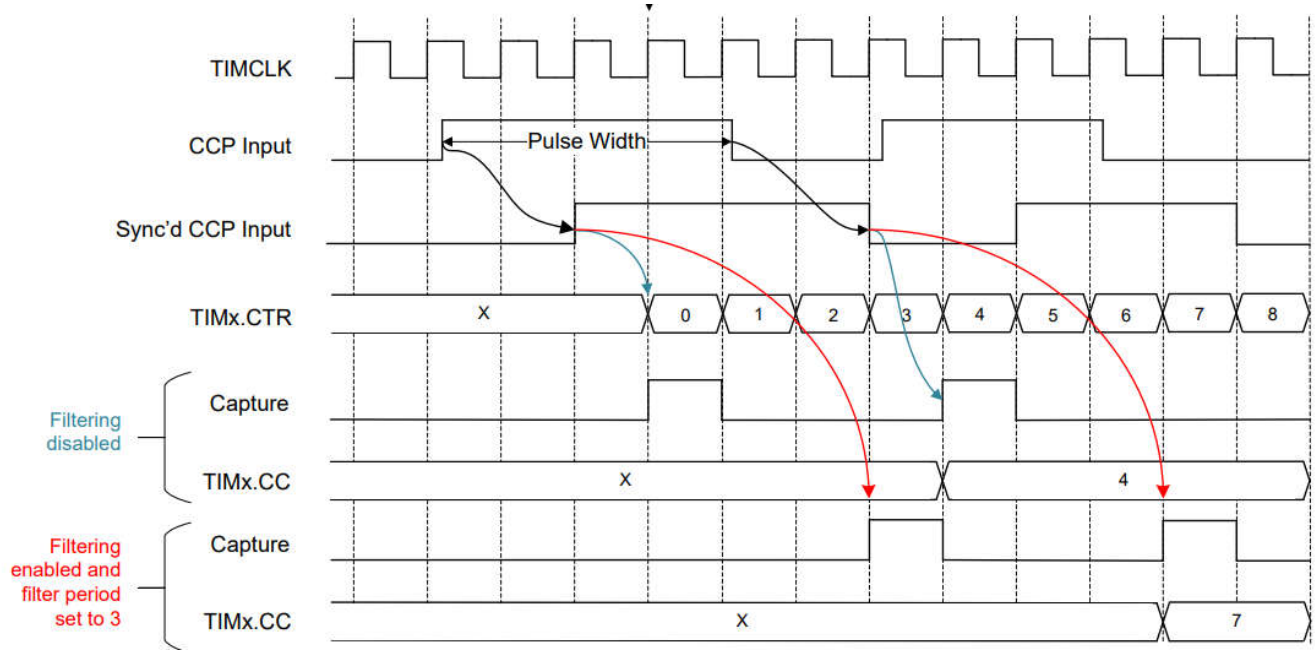

Figure 1-1. Pulse-Width Capture Mode

As mentioned, the MSPM0 timer relies on rising and falling edges to measure PWM duty cycles. The 0% and 100% duty cycles lack these edges so these cannot be captured directly.

2 Implementations for Capturing 0% or 100% Duty Cycle

2.1 Application Conditions

To capture 0% or 100% duty cycles, the PWM frequency must be fixed and known—or changed to a known value. If the frequency varies unpredictably, detecting these extreme duty cycles becomes impossible.

2.2 Implementation Introduction

To capture a fixed-frequency PWM duty cycle, an additional timeout timer running at the same frequency as the PWM can be used. This timer generates a timeout event when a 0% or 100% duty cycle occurs and then to detect high or low of the PWM signal.

Since the MSPM0 timer lacks an input signal state monitor register, other peripherals must be used to detect the signal state. The software flow for this application note is illustrated below:

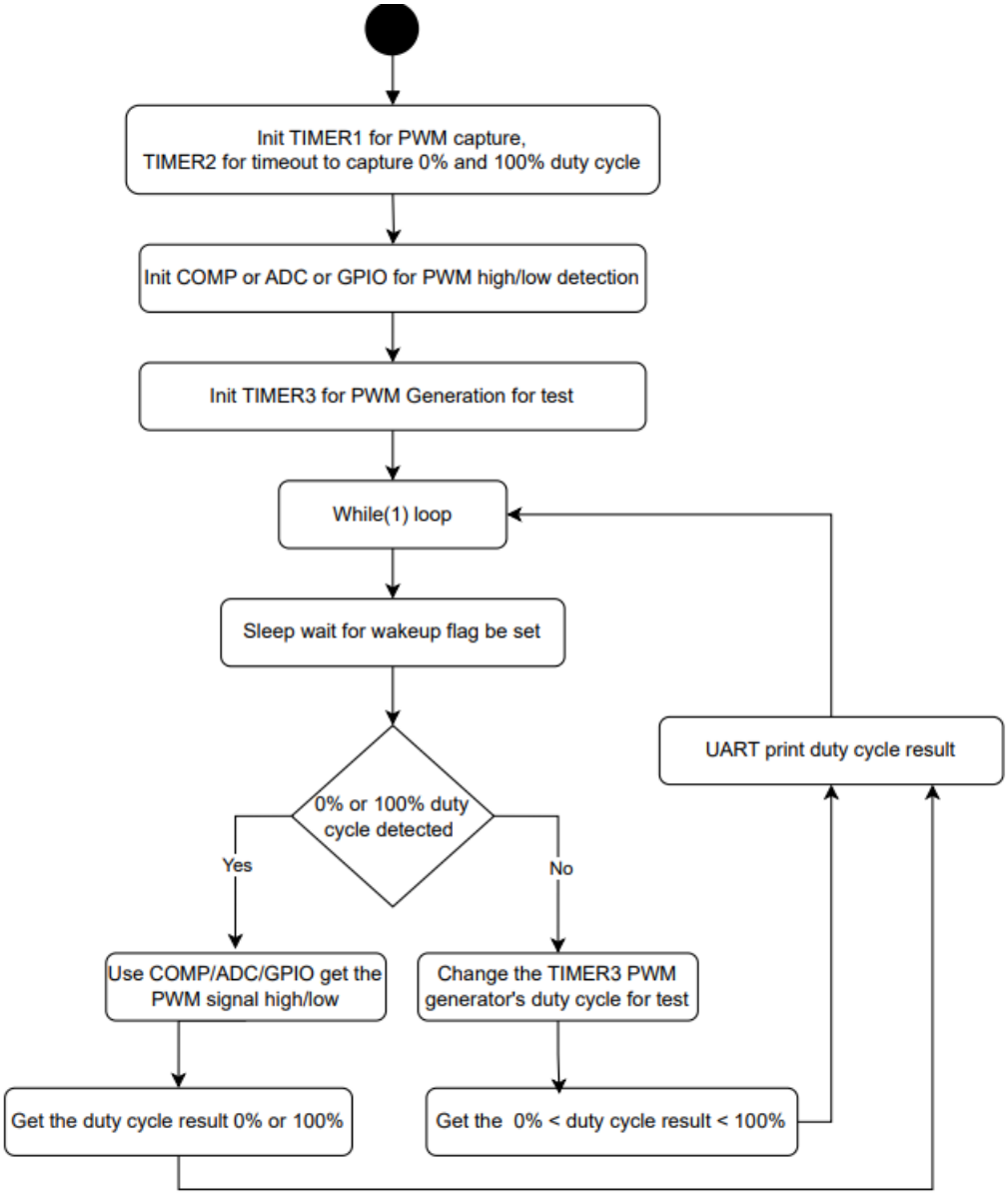

Figure 2-1. Software Flow of Main Thread

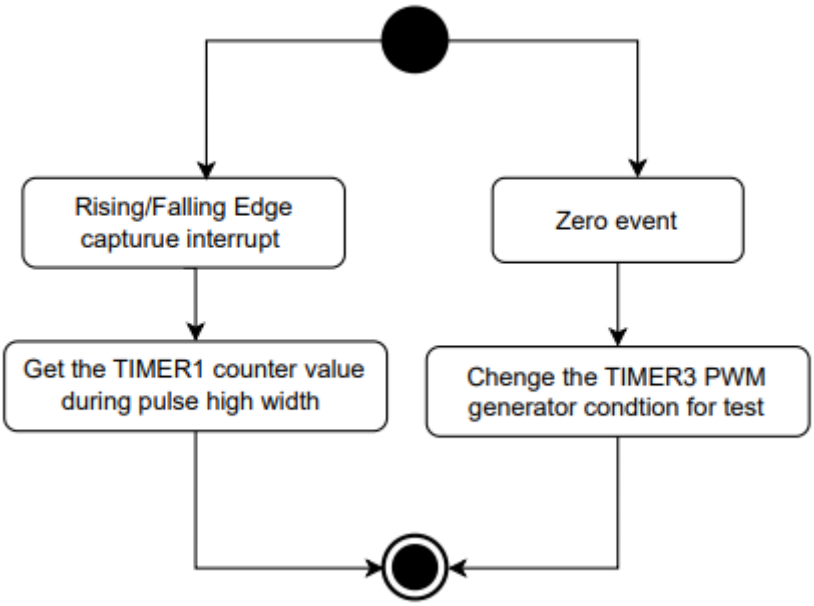

Figure 2-2. TIMER1 ISR

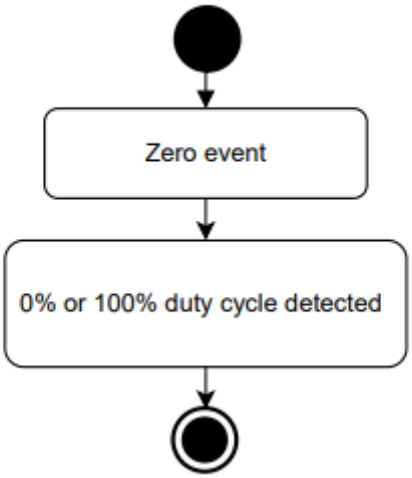

Figure 2-3. TIMER3 ISR

2.2.1 Use COMP Detect High or Low

If the MSPM0 used has a comparator (COMP) and is not used for other functions, then this can be used for 0% or 100% duty cycle high or low detection. The internal connection can be connected as below in the demo code.

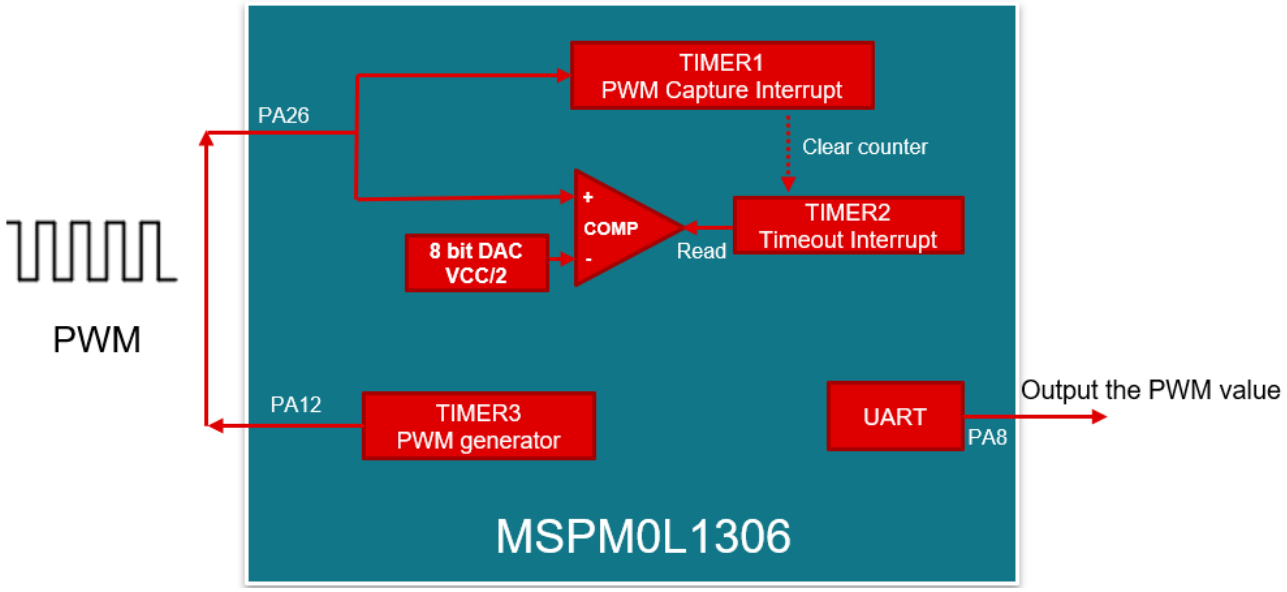

Figure 2-4. COMP Implementation Interconnection in the Demo

This implementation does not use additional pins for detection; the COMP shares the same input pin with TIMER1 capture input. The implementation needs the PWM capture pin to share the function of the capture and COMP input of the timer.

2.2.2 Use ADC Detect High and Low

All of MSPM0 has ADC and multi-input channels, and can be used for 0% or 100% duty cycle high or low detection. The internal connection can be connected as shown in the demo code.

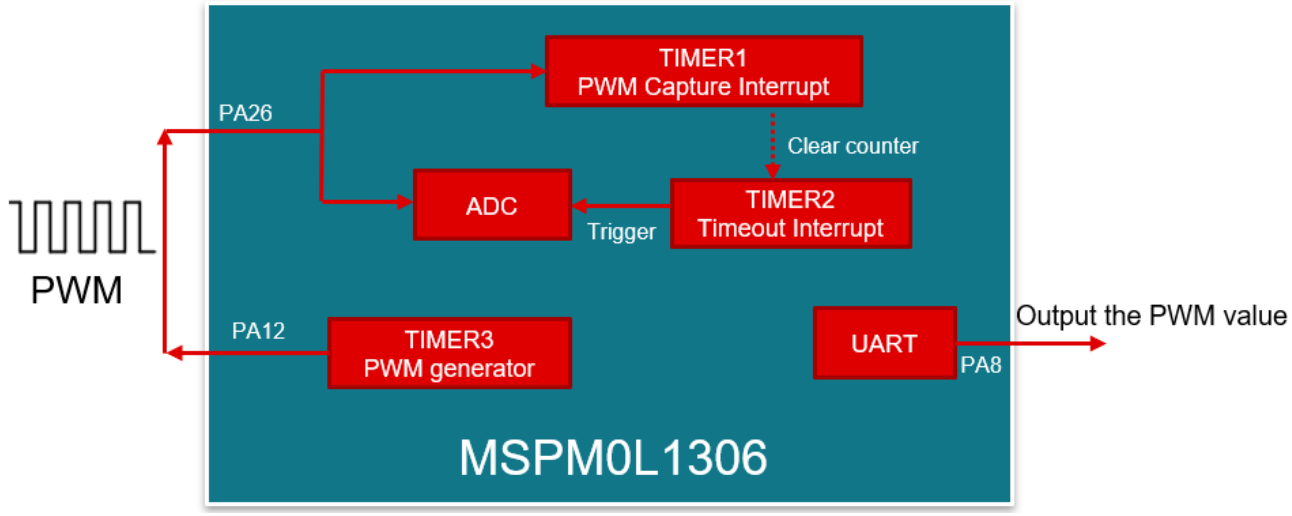

Figure 2-5. ADC Implementation Interconnection in the Demo

Do not use additional pins for detection in the implementation. The ADC channel shares the same input pin with TIMER1 capture input. This needs the PWM capture pin to share the function of the capture and ADC input of the timer.

2.2.3 Use GPIO Detect High or Low

If the implementations above cannot be used in the application, then use GPIO to detect for 0% or 100% duty cycle high or low. This implementation needs an additional GPIO to detect PWM. The internal connection can be connected as shown in the demo code.

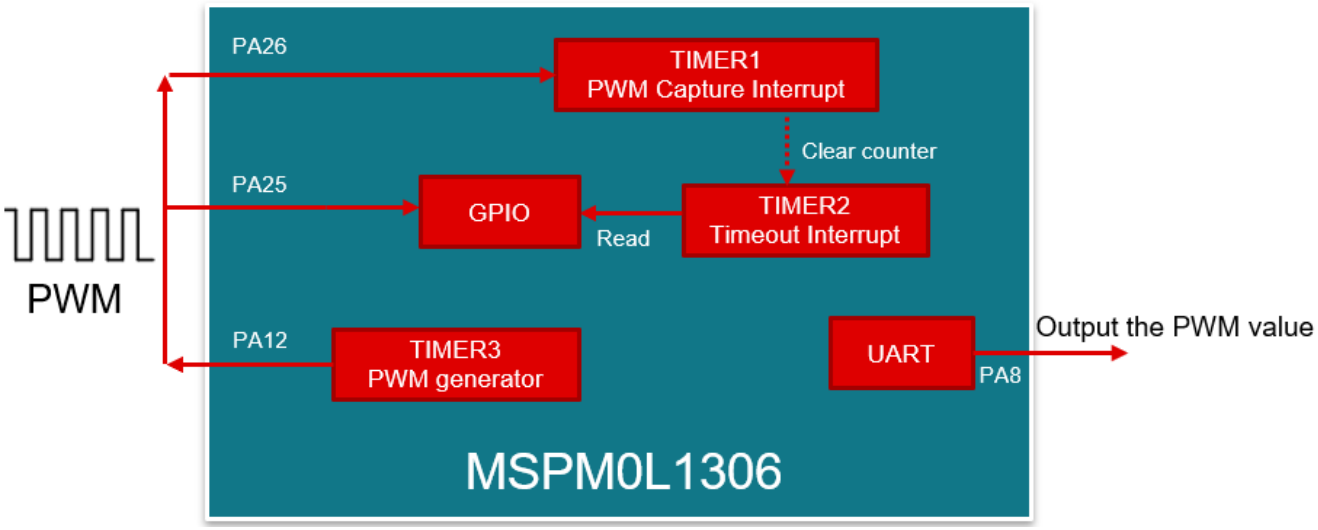

Figure 2-6. GPIO Implementation Connection in the Demo

Although IOMUX has the input mux module, TI recommends to use one GPIO as the capture input of the timer and the GPIO input for 0% or 100% duty cycle high or low. There can be one missing PWM cycle if there is a non-0% or non-100% duty cycle PWM when the IO mux stays in GPIO mode, and can also have some glitches when changing back to the capture mode of the timer. This implementation of using two GPIOs can avoid these issues.

3 Demo Code Running

This application note provides three different demo codes that using COMP, ADC and GPIO for 0% or 100% duty cycle high or low detection. The demo can be tested with LP-MSPM0L1306. The signals assignment is shown in Table 3-1.

Table 3-1. Signals Assignment Test with LP-MSPM0L1306

Signals	LP-MSPM0L1306	Description
PWM	PA12	Generate PWM running at frequency of 125Hz; the duty cycle changes between 0%, 50% and 100%
Capture	PA26	Capture PWM duty cycles
Capture GPIO	PA25	Used for 0% or 100% duty cycle high or low detection (only the GPIO implementation is needed)
UART	PA8	Print every PWM duty cycles with 9600 baud rate
Debug GPIOs	PA0, PA1, PA3	For debug (currently the PA0 is used to monitor all TIMER1 interrupts, PA1 is used to monitor PWM edge capture interrupts and PA3 is used to monitor TIMER1 zero event interrupts)

The software can be downloaded at software demo.

The COMP demo code is used as an example to show how to run the demo below.

1. Import the demo code into CCS and build
2. Use a jumper wire to connect PA12 and PA26
3. Use logic analyzer like Seleae to monitor the signals (PWM, UART and debug GPIOs)
4. Connect LP-MSPM0L1306 to a PC and download the code into the board
5. Turn on the logic analyzer to start the capture and run the code
6. The capture results are shown below

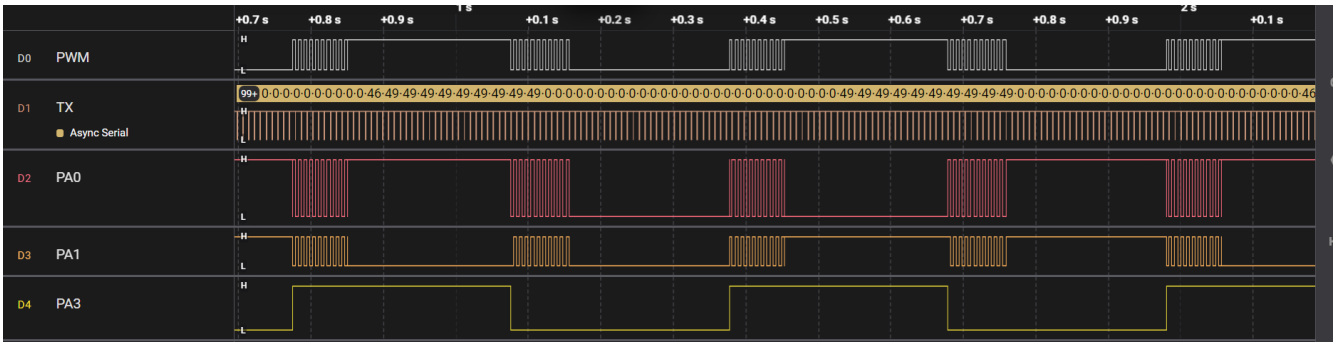

Figure 3-1. Test Results Signals Captured by Seleae

4 Summary

This application note provides implementations for designers to detect a PWM signal with a duty cycle ranging from 0% to 100% with an MSPM0 MCU. One limitation is the PWM frequency can't be changed.

5 References

- Texas Instruments, MSPM0 L-Series 32MHz Microcontrollers, technical reference manual

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265