Application Note

Wide Range and High Resolution PWM Signal Capture With MSPM0C1104

Joe Ji, Janz Bai

ABSTRACT

PWM signal is used as the speed control or feedback in system design, it is important that the PWM signal is measured accurately in a wide range of period to meet the system control accuracy requirement. Normally speaking the high MCU clock frequency is needed to realize the high resolution PWM signal capture. This application note achieves less than 1% capture error of the input PWM duty cycle and period from 100Hz to 10KHz frequency range with MSPM0C1104 running on 24MHz MCU clock frequency, which highly benefits the system design with low power consumption.

Table of Contents

Trademarks

1 Introduction

1.1 PWM Signal Capture Introduction

A PWM signal is widely used as the speed control or feedback in system design. The relatively high MCU CLK frequency is required to reach a high PWM capture accuracy. A MSPM0C110x series MCU benefits the system design with the variety of peripherals, but the MCU CLK is only 24MHz which can cause errors in PWM capture. This application report proposes the workaround for the PWM capture error with a MSPM0C110x series MCU with only one timer and one GPIO. This workaround can also identify 100% and 0% PWM duty cycle without any additional work.

1.2 MSPM0C110x Introduction

MSPM0C110x microcontrollers (MCUs) are part of the MSP highly-integrated ultra-low-power 32-bit MCU family based on the enhanced Arm ® Cortex®-M0+ core platform operating at up to 24MHz frequency. These MCUs offer high-performance analog peripheral integration.

The MSPM0C110x devices provide up to 16KB embedded flash program memory with 1KB SRAM. This device integrates a variety of high-performance analog peripherals such as one 12-bit 1.5Msps ADC with VDD as the voltage reference, and an on-chip temperature sensor. The device also offers intelligent digital peripherals such as one 16-bit advanced timer, two 16-bit general purpose timer, one windowed watchdog timer, and a variety of communication peripherals including one UART, one SPI, and one I2C. These communication peripherals offer protocol support for LIN, IrDA, DALI, Manchester, smart card, SMBus, and PMBus

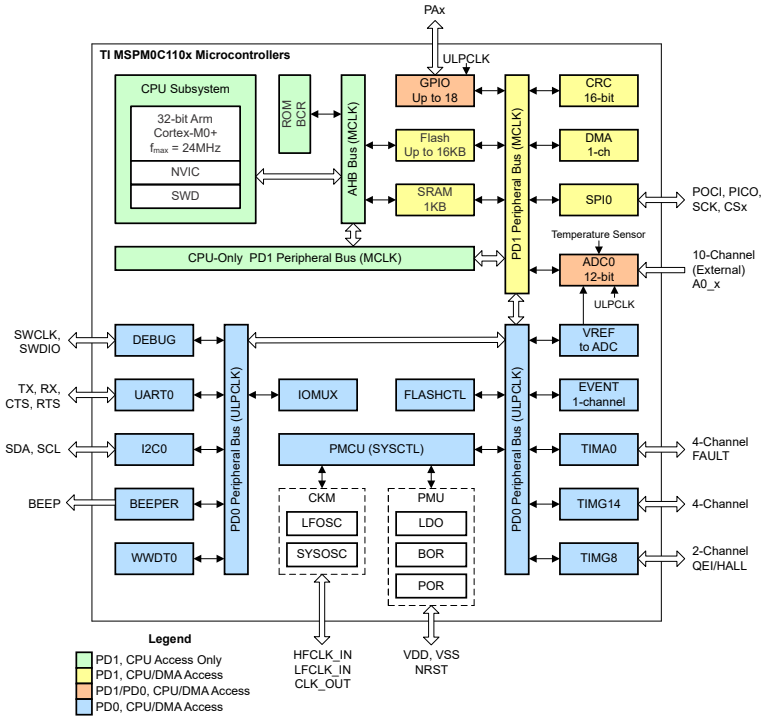

Figure 1-1. MSPM0C110x Functional Block Diagram

2 PWM Signal Capture

2.1 PWM Signal Capture Methods

This section discusses two methods to capture the PWM input signal. The first method is capturing the PWM signal with a TIMx CC block. The second method is capturing the PWM signal with GPIO interrupt.

2.2 PWM Signal Capture with TIMx CC Block

The capture and compare (CC) block is used for capture events or compare events. Using the MSPM0C series MCU, TIMG has up to two identical capture and compare blocks and TIMA has up to four identical capture and compare blocks present to support external or internal signals. Timer capture mode is used to generate capture events and record time intervals, which is useful for speed computation or time measurements. See the *MSPM0 C-Series 24-MHz Microcontrollers Technical Reference Manual* for more information about CC lock settings and key registers for configuring capture mode.

Using two capture registers can combine pulse-width and period capture of a single input waveform. The input signal can be externally connected to channel 0 of the CCP, and the IFCTL_01[1] register can be configured to have the input connected to channel 1 of the CCP internally so capture register 0 (TIMx.CC0) captures the pulse width and capture register 1 (TIMx.CC1) captures the period. The expected internal timing for combined pulse-width and period capture is shown in Figure 2-1.

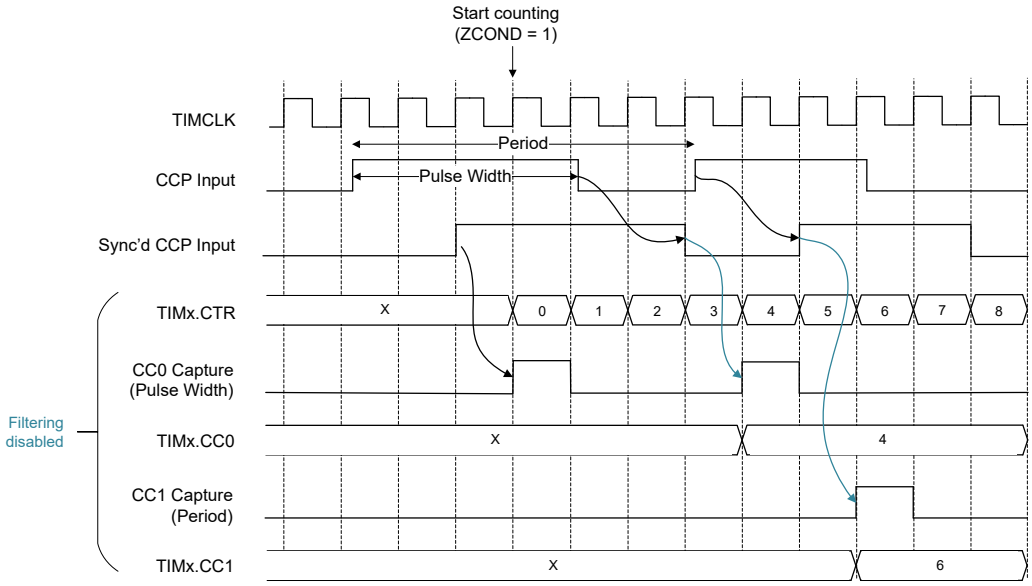

Figure 2-1. Combined Pulse-Width and Period Capture

The duty and period calculation formulas are:

$$Period = T_{Load} - T_{Cc1} \quad (1)$$

$$Duty = \frac{(T_{load} - T_{CC0})}{Period} \quad (2)$$

- T_load is the maximum value loaded in TIMx.CTR
- T_cc0 is the captured value in TIMx.CC0
- T_cc1 is the captured value in TIMx.CC1

The formula functions when TIMx.CTR is set to count down mode.

Figure 2-2 shows the capture time order of this method, since the TIMx.CTR must be reloaded manually after the first input edge signal captured at T1. The time delay from T1 to T2 causes this error.

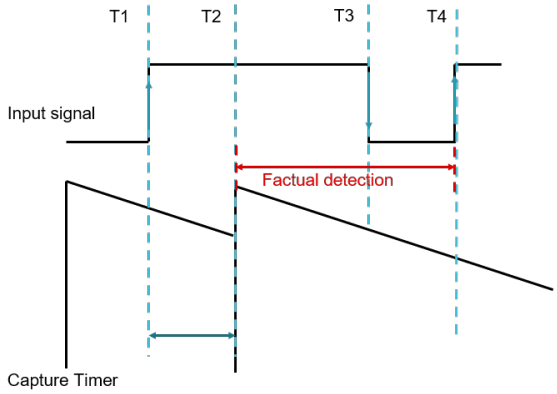

Figure 2-2. Timer Order of Capture PWM Duty and Period with Timx CC Block

- T1: Trigger interrupt to load PRD value into CTR
- T2: Set timer count value back to load value in interrupt handle
- T2 - T1: Time interval of entering interrupt handler to load value
- T3: Count value loaded into CC0
- T4: Count value loaded into CC1 and trigger interrupt

2.3 PWM Signal Capture with GPIO Interrupt

The PWM duty and period capture also can be realized with GPIO interrupt in software. The CPU interrupt can be generated on both GPIO falling and rising edge with GPIO First Event Publisher. The user can control the CPU timer action in the GPIO interrupt to capture PWM duty and period. The capture time order is described in the following sections. The software details are described in following chapters. To improve the capture accuracy, the timer must start or stop as soon as the GPIO interrupt enters to verify the short and equal SW delay between T1, T2, and T3. The time order of this PWM duty and period capture method is shown in Figure 2-3.

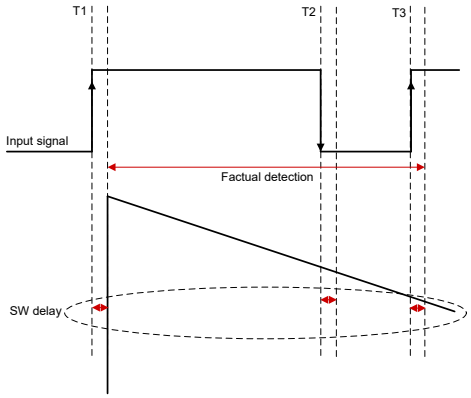

Figure 2-3. Timer Order of Capture PWM Duty and Period with GPIO Interrupt

- T1: Trigger GPIO interrupt in rising edge and start timer.
- T2: Trigger GPIO interrupt in falling edge and record the TIMx.CTR value.
- T3: Trigger GPIO interrupt in rising edge and stop timer.

2.4 Comparison of Different PWM Signal Capture Design

Table 2-1 listed the differences between two PWM capture methods, according to the table PWM signal capture with TIMx CC block realized by MCU hardware. PWM signal capture with GPIO interrupt can reach to higher accuracy and feasibility:

Table 2-1. PWM Signal Capture Methods Comparisons

Items	PWM Signal Capture with TIMx CC Block	PWM Signal Capture with GPIO Interrupt
Capture Delay	Decided by timer capture block	Decided by interrupt and software
Capture error	Decided by timer capture block	Can be managed by SW
Absolute error	Varies according to PWM frequency and duty cycle	Under 1%, simple to compensate the error

3 Software Realization

The demo code can be downloaded through E2E forum, the forum link is attached in the References section, to verify the PWM Signal Capture with GPIO interrupt function, add the "Capture_IO" into CCS project predefined symbols window, to verify the PWM Signal Capture with TIMx CC Block, add the "Capture_CC" into CCS project predefined symbols window.

3.1 Identifying Rising and Falling Edge

GPIO interrupt is triggered on both GPIO rising edge and falling edge. Read the GPIO state to identify whether the GPIO rising edge triggers the interrupt or GPIO falling edge triggers the interrupt according to Table 3-1.

Table 3-1. GPIO Falling and Rising Edge Interrupt

GPIO State	GPIO Edge to Trigger the Interrupt
GPIO input = low level	Falling edge
GPIO input = high levl	Rising edge

3.2 Time Order Classification

This demo takes the GPIO falling edge as the start of one PWM pulse. The correct time order must be: the first falling edge is detected, the rising edge is detected and the second falling edge is detected. In some corner cases, the PWM duty cycle is less than 0.2% or the PWM duty cycle is larger than 99.7%. The time order can be inaccurate due to the noise. The gflag and gflag_2 in the demo are set to identify the correct time order. The corresponding actions in different situations are listed in Table 3-2

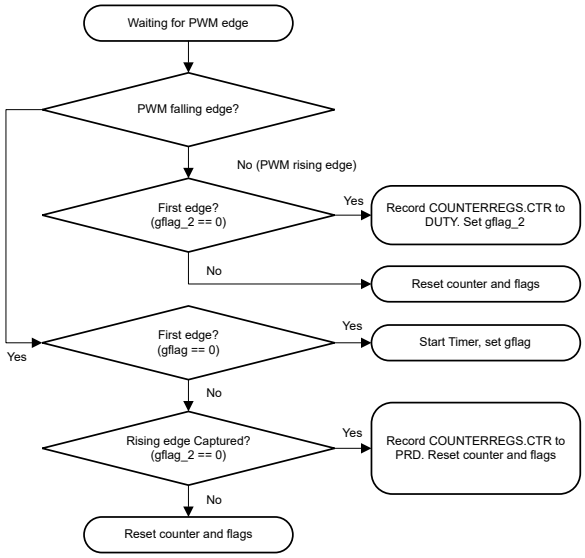

Figure 3-1. Software Diagram of PWM Duty and Period Capture with GPIO Interrupt

Table 3-2. Actions in GPIO Interrupt Handler in Different Conditions

GPIO Edge	gflag	gflag_2	Actions
Rising edge	0	0	Do nothing
	0	1	Does not occur
	1	0	Record COUNTERREGS.CTR to DUTY, Set gflag_2
	1	1	Reset counter and flags
Falling edge	0	0	Start Timer, set gflag
	0	1	Does not occur
	1	0	Reset counter and flags
	1	1	Record COUNTERREGS.CTR to PRD, Reset flags

3.3 Signal Filter and Result Calculation

To avoid the effects of input jitter and noise, the demo code sets a filer window for the captured duty cycle and period, which can be changed by the macro definition *Filterwindow*. The signal filter and result calculation flow is described in Figure 3-2 .

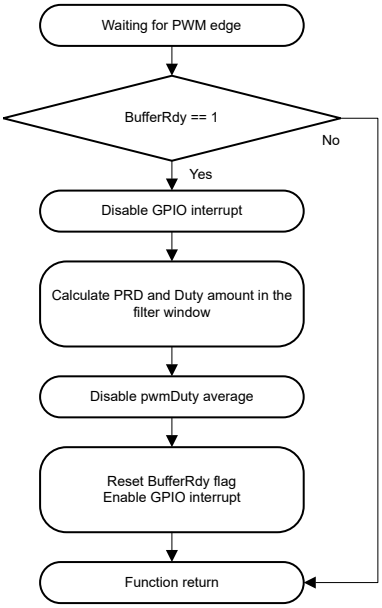

Figure 3-2. Software Diagram of Signal Filer and Result Calculation

4 System Test

This section discusses the test setup and test results of the PWM capture methods.

4.1 Test Setup

The waveform generator is required to obtain the reference PWM for testing. The waveform generator can also be replaced by another MCU or DSP which can generate the accurate PWM. Connect the PWM signal to PA23 and GND to LP-MSPM0C1104. Connect the LP-MSPM0C1104 to a PC through a micro-USB cable, as shown in Figure 4-1.

The demo code can be downloaded through E2E forum, the forum link is attached in the References section. Add predefined symbols *Capture_CC* then click *Build* to debug the PWM signal capture with TIMx CC block, or add predefined symbols *Capture_IO* then click *Build* to debug the PWM signal capture with GPIO interrupt. The *Timer Clock Prescaler* is set to 5 and *Desired Timer Period* is set to 12ms in the project to capture the 100Hz frequency PWM.

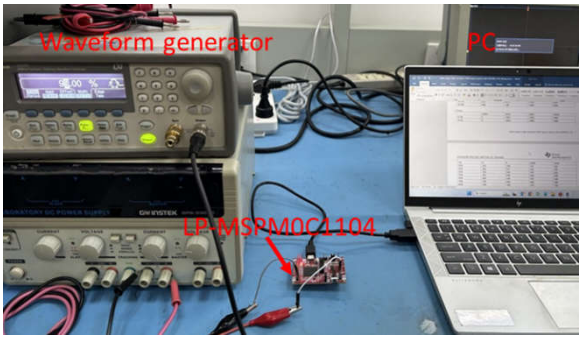

Figure 4-1. Hardware Setup for Test

4.2 Variable Monitor

The monitor variable in both projects is *pwmDuty*. Compare the value with the real PWM duty set in the waveform generator to obtain the capture error.

4.3 PWM Signal Capture Resolution Test and Comparison

To compare the two methods performance in full range and corner cases, 100Hz/ 2KHz/ 5KHz/ 10KHz PWM frequencies are used in this test, PWM duty cycle covers from 0.2% to 99.8%. Figure 4-2 shows the comparison error with GPIO interrupt (Blue line) and TIMx CC block (Orange line), the x-axis represents the PWM duty cycle, to show the corner case properly, the number is not proportional, the y-axis represents the capture error in ‰. Table 4-1 and Table 4-2 shows the capture result with GPIO interrupt (Blue line) and TIMx CC block (orange line).

In all the PWM duty range, the GPIO interrupt method keeps the capture error below 1%, it also has the robust performance in the corner case, when PWM duty <1% and >99%, which avoid the abnormal data. The TIMx CC block method shows high resolution when PWM duty is large especially >99%, but the capture value must be corrected manually when PWM duty <1% to avoid the >100% wrong results.

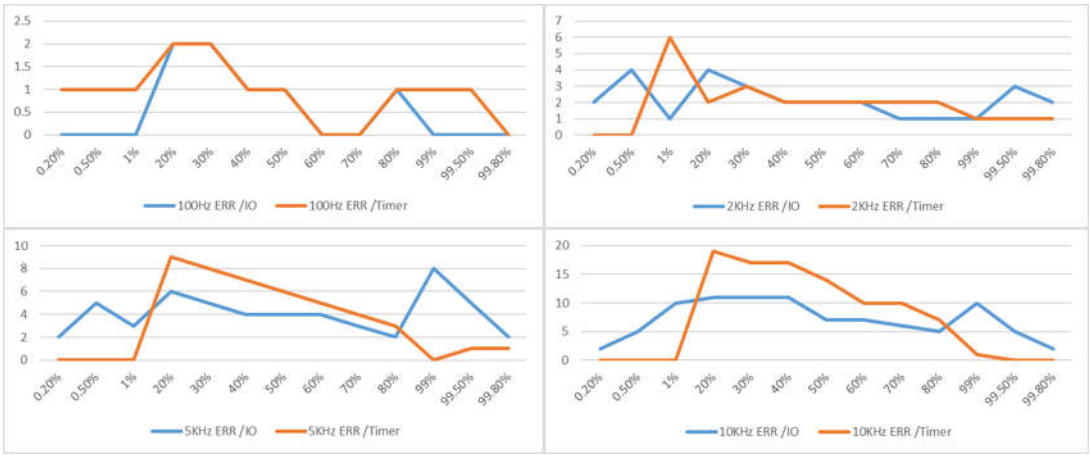

Figure 4-2. PWM Duty Capture Error Comparison

Table 4-1. PWM Duty Cycle Capture Error with GPIO Interrupt

Input Signal	100Hz		2KHz	5KHz	10KHz			
	Calculation result	Error	Calculation result	Error	Calculated result	Error	Calculation result	Error
0.2%	0.2%	0	0.0%	0.2%	0.0%	0.2%	0.0%	0.2%
0.5%	0.5%	0	0.9%	0.4%	0.0%	0.5%	0.0%	0.5%
1%	1.0%	0	1.1%	0.1%	2.3%	0.3%	0.0%	1.0%
20%	20.2%	0.2%	20.4%	0.4%	20.6%	0.6%	21.0%	1.0%
30%	30.2%	0.2%	30.3%	0.3%	30.5%	0.5%	31.0%	1.0%
40%	40.1%	0.1%	40.2%	0.2%	40.4%	0.4%	41.0%	1.0%
50%	50.1%	0.1%	50.2%	0.2%	50.4%	0.4%	50.7%	0.7%
60%	60.0%	0	60.2%	0.2%	60.4%	0.4%	60.7%	0.7%
70%	70.0%	0	70.1%	0.1%	70.3%	0.3%	70.6%	0.6%
80%	79.9%	0.1%	80.1%	0.1%	80.2%	0.2%	80.5%	0.5%
99%	99.0%	0	99.1%	0.1%	98.2%	0.8%	100.0%	1.0%
99.5%	99.5%	0	99.2%	0.3%	100.0%	0.5%	100.0%	0.5%
99.8%	99.8%	0	100.0%	0.2%	100.0%	0.2%	100.0%	0.2%

Table 4-2. PWM Duty Cycle Capture Error with TIMx CC Block

Input Signal	100Hz		2KHz		5KHz		10KHz	
	Calculation result	Error	Calculation result	Error	Calculation result	Error	Calculation result	Error
0.2%	0.1%	0.1%	0%	0.2%	0%	0.2%	0%	0.2%
0.5%	0.4%	0.1%	0%	0.5%	0%	0.5%	0%	0.5%
1%	0.9%	0.1%	0.4%	0.6%	0%	1.0%	0%	1.0%
20%	20.2%	0.2%	19.8%	0.2%	19.1%	0.9%	18.1%	1.9%
30%	30.2%	0.2%	29.7%	0.3%	29.2%	0.8%	28.3%	1.7%
40%	40.1%	0.1%	39.8%	0.2%	39.3%	0.7%	38.3%	1.7%
50%	50.1%	0.1%	49.8%	0.2%	49.4%	0.6%	48.6%	1.4%
60%	60.0%	0	59.8%	0.2%	59.5%	0.5%	59.0%	1.0%
70%	70.0%	0	69.8%	0.2%	69.6%	0.4%	69.0%	1.0%
80%	79.9%	0.1%	79.8%	0.2%	79.7%	0.3%	79.3%	0.7%
99%	98.9%	0.1%	98.9%	0.1%	99.0%	0	98.9%	1.0%
99.5%	99.4%	0.1%	99.4%	0.1%	99.4%	0.1%	99.5%	0
99.8%	99.8%	0	99.7%	0.1%	99.7%	0.1%	99.8%	0

5 Summary

This application note proposes the workaround for the PWM capture error with MSPM0C110x series MCU with only one timer and one GPIO. This workaround can also identify 100% and 0% PWM duty cycles without any additional work. The test results show that the method can keep the capture error below 1% in 100Hz to 10KHz PWM frequency range. The test results also show the robustness in PWM duty <1% and >99% corner cases.

6 References

- Texas Instruments, *MSPM0 C-Series 24-MHz Microcontrollers Technical Reference Manual* , technical reference manual.
- Texas Instruments, *MSPM0C110x, MSPS003 Mixed-Signal Microcontrollers*, data sheet.
- Texas Instruments, E2E Forum, E2E forum.

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