

Subsystem Design

ADC to PWM

Description

This example demonstrates how to convert an analog signal to a 4kHz PWM output. The analog input signal is sampled using the MSPM0 integrated ADC. The duty cycle of the PWM output gets updated based on the ADC reading. Two timers are required for this example; one to trigger the ADC reading and another to generate the PWM output. Download the code for this example.

Figure 1-1 displays a functional block diagram of the peripherals used in this example.

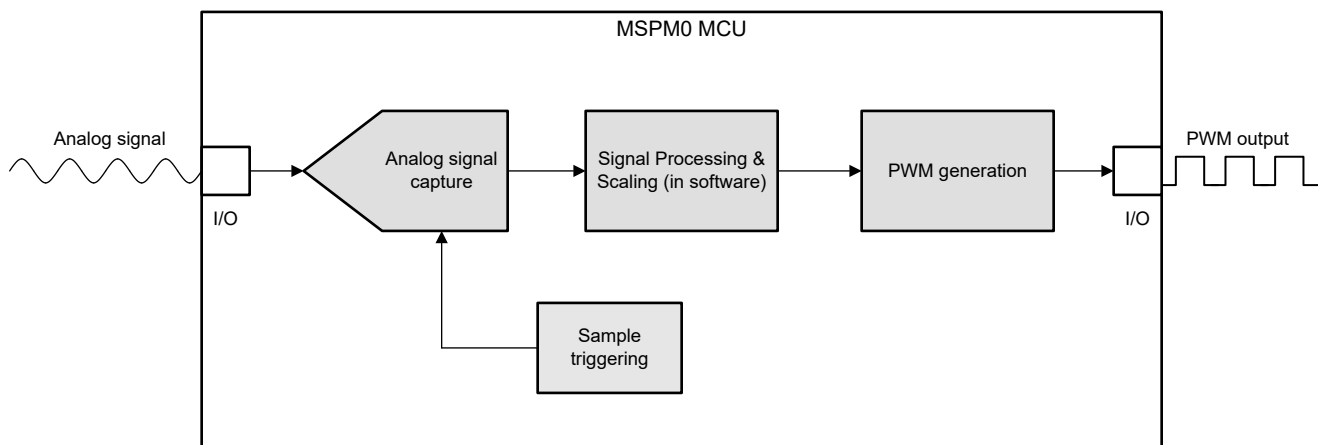


Figure 1-1. Subsystem Functional Block Diagram

Required peripherals

This application requires 2 timers, an integrated ADC, and 2 device pins.

Table 1-1. Peripheral Requirements

Sub-block functionality	Peripheral usage	Notes
Sample triggering	(1x) Timer G	Called <code>TIMER_0_INST</code> in code
PWM generation	(1x) Timer G	Called <code>PWM_0_INST</code> in code
Analog signal capture	1 ADC channel	Called <code>ADC12_0_INST</code> in code
IO	2 pins	(1x) ADC input (1x) PWM output

Compatible devices

Based on the requirements in ,[Table 1-1](#) this example is compatible with the devices in [Table 1-2](#). The corresponding EVM can be used for prototyping.

Table 1-2.

MSPM0Lxxx	EVM
MSPM0Lxxx	LP-MSPM0L1306
MSPM0Lxxx	LP-MSPM0G3507

Design steps

1. Determine the required PWM output frequency and resolution. These two parameters are the starting point when calculating other design parameters. In this example we chose an PWM output frequency of 4kHz and a PWM resolution of 10bits.
2. Calculate the timer clock frequency. The equation $F_{\text{clock}} = F_{\text{pwm}} \times \text{resolution}$ can be used to calculate the timer clock frequency.
3. Determine the ADC sampling rate. The sampling rate is related to the output PWM frequency. In this example, a single ADC sample determines the duty cycle. $F_{\text{adc}} = F_{\text{pwm}}$. However, filtering or averaging may require the application to choose a different sampling rate.
4. Configure peripherals in [SysConfig](#). Select which timer instances will be used. Configure which device pins will be used for ADC input and PWM output. This example uses PA17 for the PWM output (which is connected to Timer G4) and A0.4 for the analog input.
5. Write application code. The remaining piece of this application is to transfer the ADC samples to the PWM timer. This is accomplished in software. See Software Flowchart for an overview of the application or browser the code directly.

Design considerations

1. Max output frequency: Fundamentally, the max PWM output frequency is limited by the IO speed. However, the duty cycle resolution also affect the max output frequency. More resolution requires more timer counts which increases the output period.
2. Clocking: Deciding which clocks to use and what clock division ratios to use is an important design consideration for this application.
 - a. Select a resolution that is a power of 2 so scaling operations can use shifts instead of multiplies and divides b.
 - b. In general, don't divide a slower clock down to a lower frequency. Instead pick a slower clock to reduce power consumption 3.
3. Race conditions on gCheckADC: This application takes care to clear gCheckADC as soon as possible. If the application waits too long to clear gCheckADC it may inadvertently miss new data.
4. Pipelining: The PWM timer selected in this application supports pipelining the timer compare value. Pipelining allows the application to schedule an update to the timer compare value without causing a glitch on the output. Techniques exist to mitigate glitches on timers without support for pipelining. However, this is beyond the scope of this document.

Software Flowchart

Figure 1-2 shows the operations performed by application to convert an ADC reading into a PWM output.

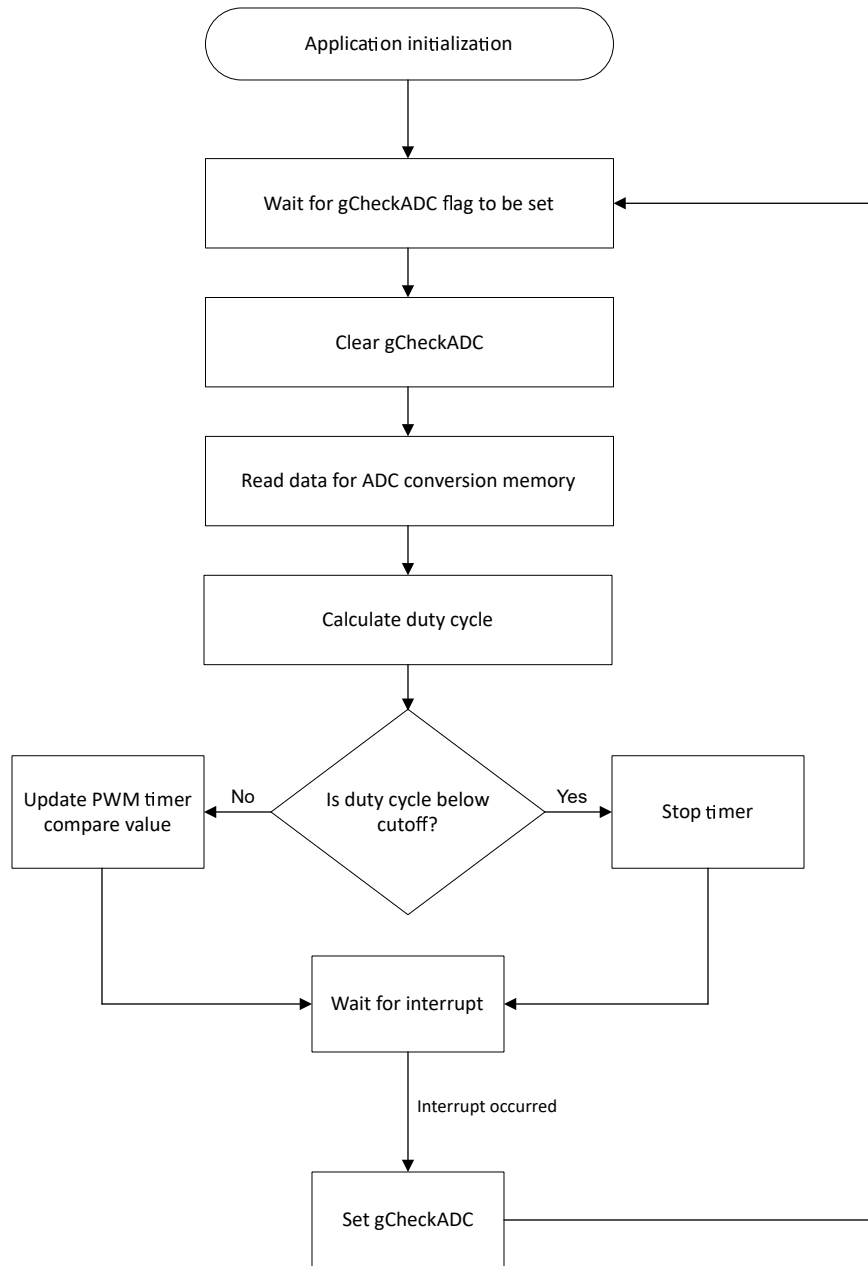


Figure 1-2. Application Software Flowchart

Application code

This application's PWM output has 10bits of resolution. However, the ADC samples are 12bits, so we must convert the 12 bit ADC readings into 10 bit values that can be used to set the compare value of the PWM timer. Depending on an applications requirements different scaling can be required.

Additionally, more advanced signal processing of the incoming data can be required. For example limiting, averaging or other filtering may be important in different scenarios. These type of operations can be performed in the function below.

```
void updatePWMfromADCvalue(uint16_t adcValue) {
    // Check to see if the adc value is above our minimum threshold
    if (adcValue > PWM_DEADBAND)
    {
        // Convert 12bit adcValue into 10bit value by right
        // shifting by 2 because the PWM resolution is 10bit
        uint16_t adcvalue_10bit = adcValue >> 2;
        // PWM timer is configured as a down counter (i.e it
        // starts counting down from PWM_LOAD_VAL) and its
        // initial state is high therefore we must perform
        // the following operation so that small values of
        // adcvalue_10bit result in small duty cycles
        uint16_t ccv = PWM_LOAD_VAL - adcvalue_10bit;
        // write the new ccv value into the corresponding timer
        // register
        DL_TimerG_setCaptureComparevalue(PWM_0_INST,
                                         ccv,
                                         DL_TIMER_CC_0_INDEX);

        // Start the timer if it is not already running
        if ( !DL_TimerG_isRunning(PWM_0_INST) ) {
            DL_TimerG_startCounter(PWM_0_INST);
        }
    }
    else {
        // If adcResult is not above deadband value then disable timer
        DL_TimerG_stopCounter(PWM_0_INST);
    }
}
```

Results

When the input voltage is below a preset deadband value the output is disabled, as shown in figure 1-3.

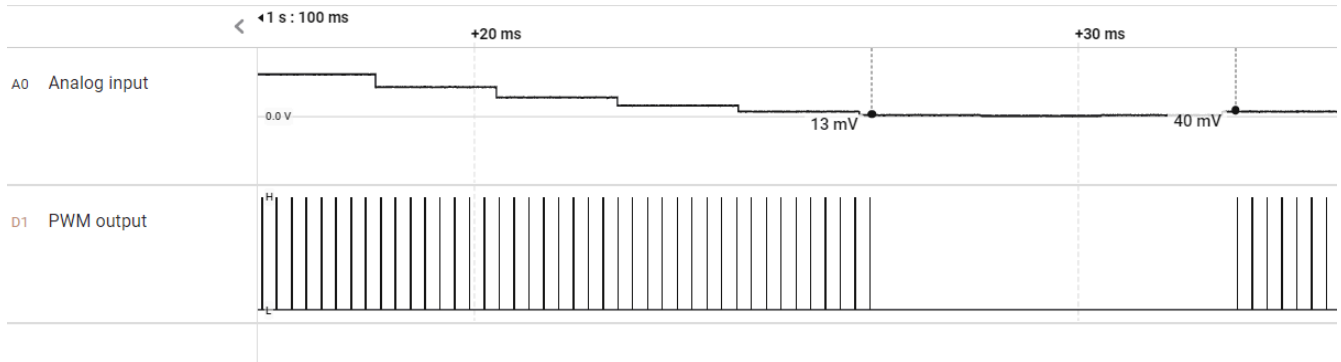


Figure 1-3. PWM output is disabled when ADC input is below deadband

In figure 1-4 the input voltage is 2.26V. The measured duty cycle is 67.93%. A quick calculation confirms that the expected duty cycle is 68.4%.

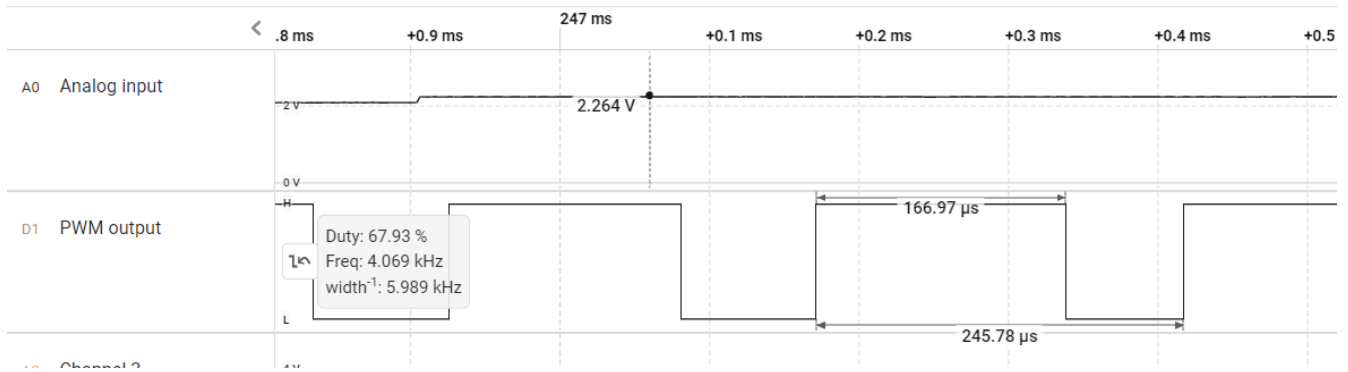


Figure 1-4. PWM output duty cycle corresponds to input voltage

Additional Resources

- [Download the MSPM0 SDK](#)
- [Learn more about SysConfig](#)
- [MSPM0L LaunchPad](#)
- [MSPM0G LaunchPad](#)
- [MSPM0 Timer academy](#)
- [MSPM0 ADC academy](#)

Revision History

DATE	REVISION	NOTES
August 2024	*	Initial Release

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](https://www.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
Copyright © 2024, Texas Instruments Incorporated