# SJSU EE138
# Introduction to Embedded Control System Design
# Lab 3: Timer Counter/ Pulse Width Modulation

**Readings:**

Atmel SAMD20 DataSheet

- Section 14: GCLK - Generic Clock Controller
- Section 15: PM - Power Manager
- Section 27: TC - Timer/Counter

**Lab Description:**

In this Lab students will learn how to utilize PWM waveforms for two different functions.

1. Implementing the function of a DAC by using a PWM output and a filter.
2. Control the speed of a motor using PWMs and the Lab 2 code for the ADC to read the position of the POT

**Theory of Operation:**

1. Power Manager
   a. In previous Labs the Power Manager (PM) has either been enabled by default or set in the sample code for you. The PM has Peripheral Clock Masks to allow the user to turn on or off a peripheral clock to conserve power. In this Lab you will need to enable some of the Timer/Counter Peripheral Clocks. Use the following information from lecture to enable the correct TC.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x14 | AHBMASK | 7:0 | | | | NVMCTRL | DSU | HPB2 | HPB1 | HPB0 |
| 0x15 | | 15:8 | | | | | | | | |
| 0x16 | | 23:16 | | | | | | | | |
| 0x17 | | 31:24 | | | | | | | | |
| 0x18 | APBAMASK | 7:0 | | EIC | RTC | WDT | GCLK | SYSCTRL | PM | PAC0 |
| 0x19 | | 15:8 | | | | | | | | |
| 0x1A | | 23:16 | | | | | | | | |
| 0x1B | | 31:24 | | | | | | | | |
| 0x1C | APBBMASK | 7:0 | | | | | PORT | NVMCTRL | DSU | PAC1 |
| 0x1D | | 15:8 | | | | | | | | |
| 0x1E | | 23:16 | | | | | | | | |
| 0x1F | | 31:24 | | | | | | | | |
| 0x20 | APBCMASK | 7:0 | SERCOM5 | SERCOM4 | SERCOM3 | SERCOM2 | SERCOM1 | SERCOM0 | EVSYS | PAC2 |
| 0x21 | | 15:8 | TC7 | TC6 | TC5 | TC4 | TC3 | TC2 | TC1 | TC0 |
| 0x22 | | 23:16 | | | | | PTC | DAC | AC | ADC |
| 0x23 | | 31:24 | | | | | | | | |

The above registers allow for individually enable/disable APB clock goes to the peripherals connected on the APBs. For example, the General Purpose IO is on APBB. The APBB clock to PORT can be enabled by writing a 1 to APBBMASK.PORT.

2. Counter
   a. The counter can be set to count up or down on every clock cycle. When the counter is counting up and the top value is reached, the counter will wrap around to zero on the

next clock cycle. When counting down, the counter will wrap around to the top value when zero is reached. For this lab keep the counter in its default state of counting up.

b. There are three counters to choose from
    i. 8-bit
    ii. 16-bit
    iii. 32-bit

3. Compare and Capture / Period and Frequency
    a. When using the TC with the Compare/Capture Value registers (CCx) configured for compare operation, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or waveform operation.

    b. There are four configurations for the Waveform Generation Operation bit group in the Control A register, two Frequency generation modes and two PWM generation modes.
        i. Normal frequency (NFRQ)
        ii. Match frequency (MFRQ)
        iii. Normal PWM (NPWM)
        iv. Match PWM (MPWM)

    c. Frequency Generation mode is in many ways identical to PWM generation. However, in Frequency Generation mode, what a match is detected, the counter is reset and the output signal toggles. Since the output state changes it value (i.e., toggles) at a fixed time period (determined by the value in the CC register), the output waveform is always a square wave with a fixed 50% duty cycle.

    d. In PWM Generation mode the frequency, for the 8-bit counter, is determined by the Period register. While the Compare/Capture Value registers (CCx) are used to determine the duty cycle of the waveform. This allows for a variable duty cycle and a variable frequency for the 8-bit counter. In a typical application, the content of the CC register is changed to vary the output effective voltage while the frequency of the PWM, once it is set, stays constant. For 16-bit and 32-bit counter operation, see the lecture note and the data sheet.

**Peripheral and Coding:**

| Address | - | SAMD20 Syntax Code | "*por" is pointer variable name |
|---|---|---|---|
| 0x40000400 | - | PM | //definition address for PM functionality |
| offset 0x1C | - | APBCMASK.reg | //used to enable peripheral clocks on the APBC bus |
| 0x42002800 | - | TC2 | //definition address for TC2 functionality |
| 0x42003000 | - | TC4 | //definition address for TC4 functionality |
| 0x42003800 | - | TC6 | //definition address for TC6 functionality |
| offset 0x00 | - | CTRLA.reg | // used to setup and enable TC |
| offset 0x0F | - | STATUS.reg | // used to check synchronization of registers |
| offset 0x10 | - | COUNT.reg | // increments on every clock cycle |
| offset 0x14 | - | PER.reg | //sets the top value for the 8bit mode counter |
| offset 0x18 | - | CC0.reg | //sets the compare and capture for even pin then outputs to WO[0] |
| offset 0x19 | - | CC1.reg | //sets the compare and capture for odd pin then outputs to WO[1] |

***NOTE:*** CC0 and CC1 serve different functions when Waveform Generation Operations other than NPWM is used or when a counter other than 8-bit counter is used.

**Required Tasks:** ***please comment your code**

**Task 1: PWM generated sinusoidal from PA13 using the 8-bit counter**

- Sine wave should about 500Hz (+/- 50Hz)
    - Screen capture the waveform before and after the filter for the Lab Report.

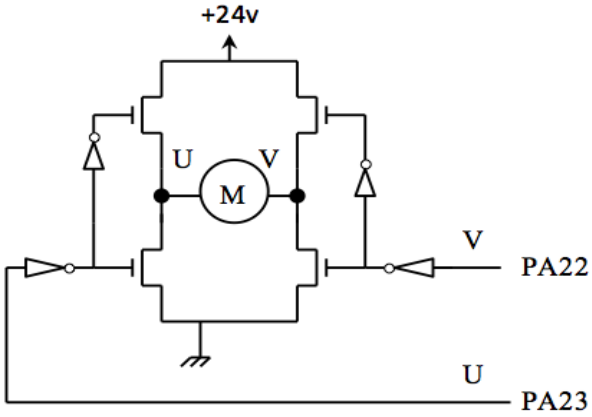
**Task 2: Open loop motor controller (Read the motor control section of Chapter 5 of the lecture note)**

- PA22 and PA23 are the outputs for the PWM to the motor.
- The POT will control the speed and direction of motor.

**Schematics for ADC and PWM:**

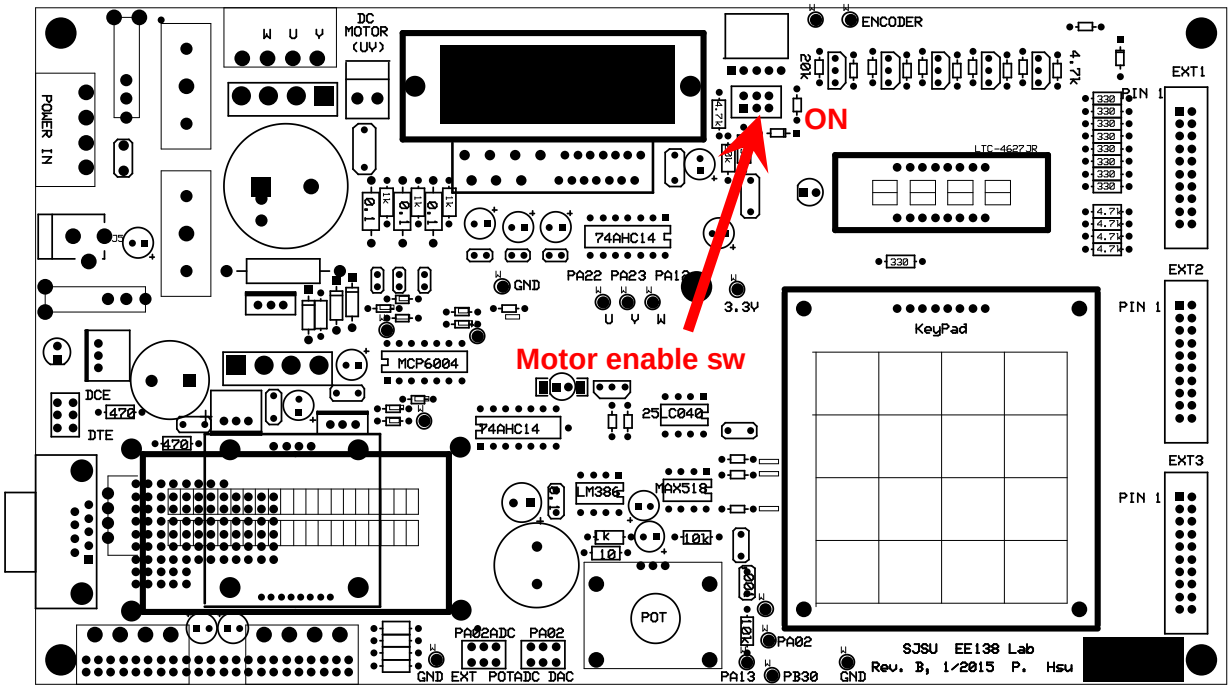
See the schematics in Lab 2. Notice the RC filter formed by the 0.1uF and 10k resistor from PA13.

The following figure shows the logic of the motor drive power circuit. Note that the PWM waveform should be output to PA22 and PA23.




Lab 3 Motor driver circuit for task 2

**The motor enable switch must be at the ON position (see figure below).**.




Location of the motor enable switch.