The University of Arizona
Department of Aerospace and Mechanical Engineering
Mechatronics Laboratory
Instructor: Professor. Eniko T. Enikov, enikov@email.arizona.edu

Laboratory Task Sheet 13

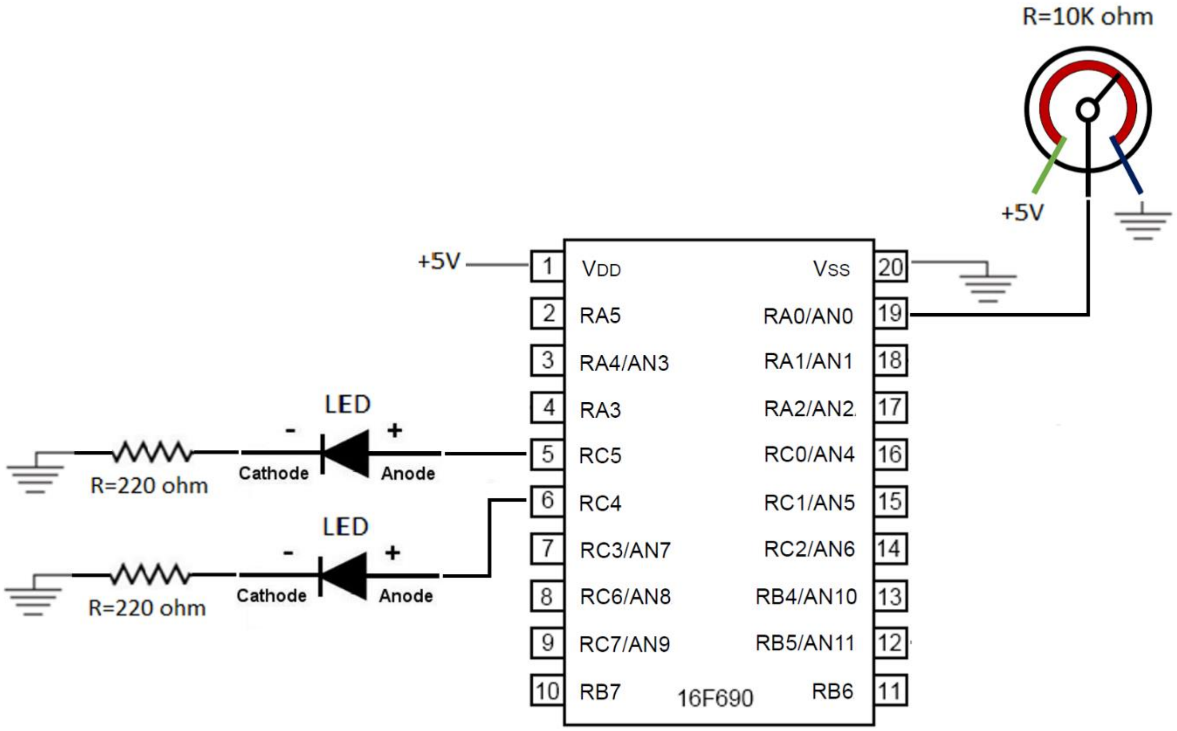
Title: DC Motor Velocity Control

Registers to be learned: CCP1CON & PR2 & SPBRG & T2CON & CCPR1L & PSTRCON & T2CON

Objective: Program the microcontroller and use Pulse Width Modulation (PWM) to control the brightness of the LEDs. Set the PWM on Half Bridge to generate the compliment value of PORTC5 on PORTC4.

Tasks

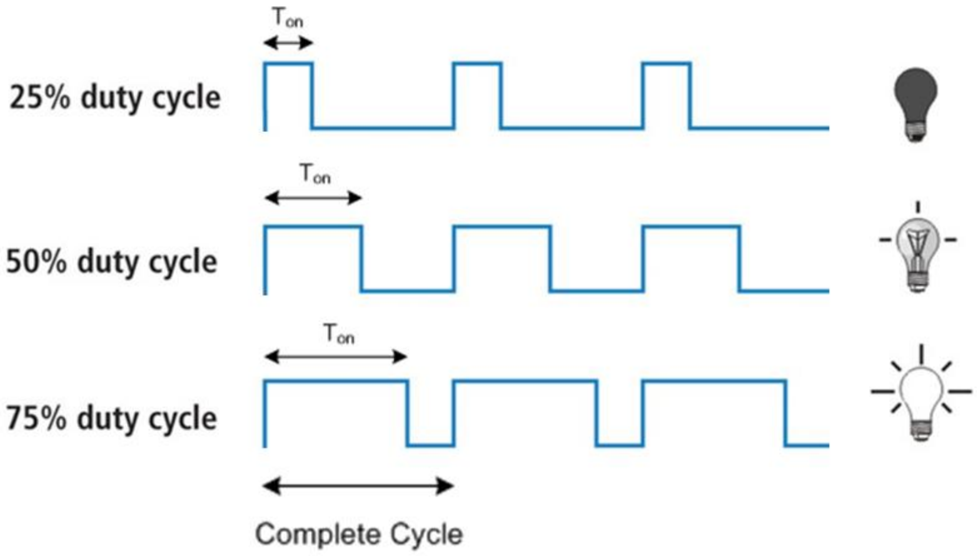
1. Create the circuit below using a potentiometer and two LEDs.

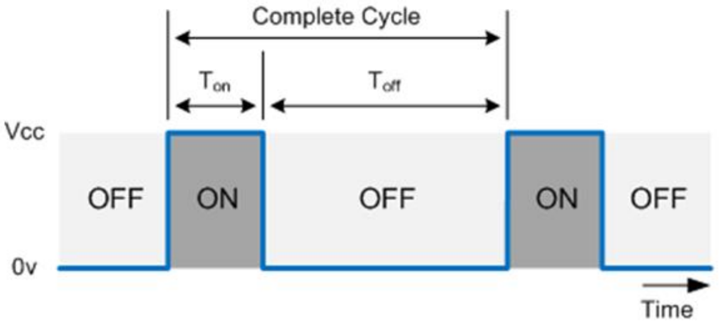


Pulse Width Modulation (PWM)

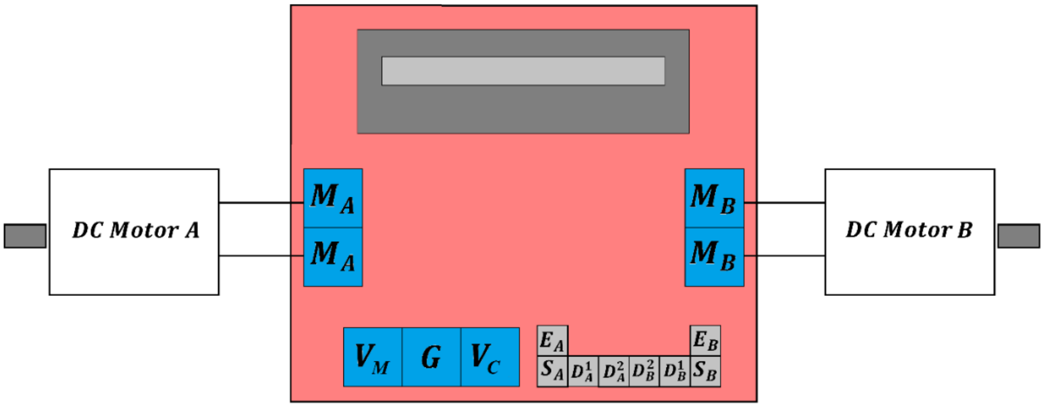
The Pulse Width Modulation creates an output averaged voltage by controlling the Pulse's Width over total period. PWM is widely used for speed control of DC Motor.

The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on. When a digital signal is on half of the time and off the other half of the time, the digital signal has a duty cycle of 50% and resembles a "square" wave. When a digital signal spends more time in the on state than the off state, it has a duty cycle of >50%. When a digital signal spends more time in the off state than the on state, it has a duty cycle of <50%.





DC Motor Driver (H-Bridge)



V_M	G	V_C
Voltage Source for Motor A and B (5-12 volts)	**Common Ground**	**Voltage Source for Motor A and B (0-5 volts)**

Motor	**Speed**	**Direction of rotation**	**Enable**
M_A	S_A	D_A	E_A
M_B	S_B	D_B	E_B

D	**Rotating CW**	**Rotating CCW**
D_A^1	**0**	**1**
D_A^2	**1**	**0**

D	**Rotating CW**	**Rotating CCW**
D_B^1	**0**	**1**
D_B^2	**1**	**0**

2. Make a copy of the P16f690_Template file and name it TASK13Group00. Open the file in MPLAB Software and use the table below to construct the code.

Suggested Code Structure
Define all the necessary Memory Bytes
Start
Call Initialization **Go to Main**
Main Call delay Use ADCON0 Register to initiate the conversion Wait until the conversion is done Use ADRESH Register to move result of the conversion to work Register Move Work Register to CCPR1L Register Go to Main
Delay Make a delay for 5 µ second Return
Initialization Bank3 Use PSTRCON Registers to enable PA and PB pins PWM waveform Use PSTRCON Registers to assign PC and PD to port pins Use PSTRCON Registers to make steering update happens at the beginning of the next PWM period Bank2 Use ANSEL and ANSELH Registers to define all the ports as digital Use ANSEL and ANSELH Registers to define PORTA0 as analog Bank1 Use OSCCON Register to set the oscillator on 8 MHz Use TRISA Register to define PORTA0 as Input Use TRISC Register to define PORTC4 and PORTC5 as output Use ADCON1 to set the ADC (Analog to Digital Convertor) clock on FOSC/16 Bank0 Use ADCON0 Register to enable ADC Use ADCON0 Register to stop the conversion Use ADCON0 Register to set PORTA0 as input channel of the convertor Use ADCON0 Register to make the convertor Left Justified Use CCP1CON Register to make all P1A, P1C, P1B, P1D active-high Use CCP1CON Register to clear both PWM Duty Cycle Least Significant Bits Use CCP1CON Register to set PWM on Half-Bridge mode Move decimal 255 to PR2 Register Clear CCPR1L Register Use T2CON Register to enable Timer2 Use T2CON Register to set both pre-scaler and post-scaler on 16 Return

```
end
```

3. Program the microcontroller and test it on the circuit.
4. Demonstrate the result to the instructor.
5. Upload the code on D2L and save it for yourself.