# COMPE-375: Embedded Systems Programming Lab

**FALL 2018**

## LAB – 09

## A/D/A Conversion

# Your Checklist to do

- ❑ Demonstrate proper operation of Lab-07 assignment at the beginning of the session
- ❑ For Lab-09 assignment:

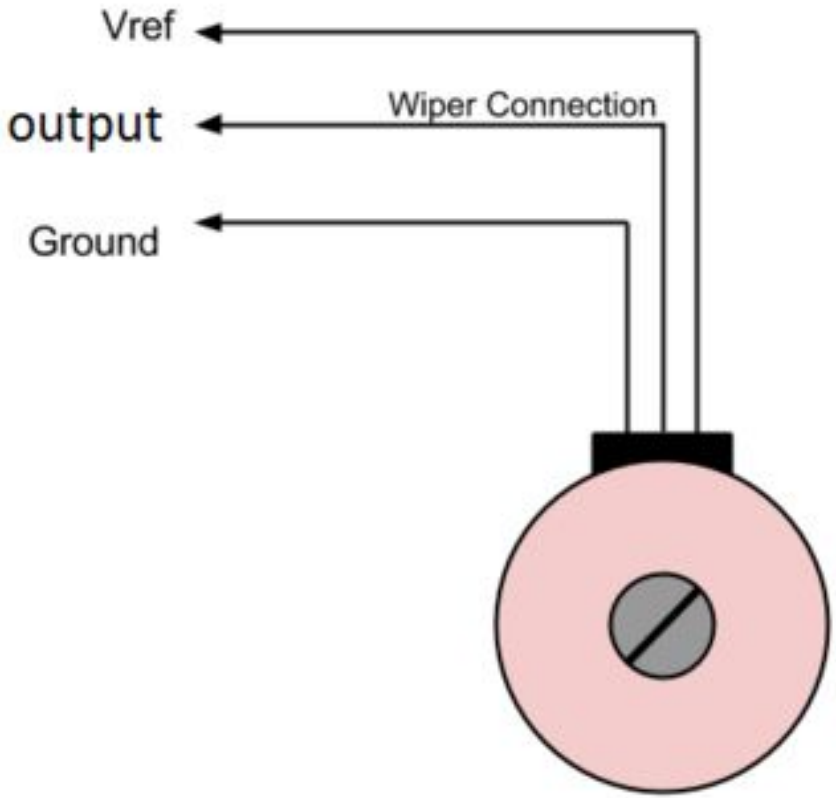
Checkpoint due on: *November 9th(One week later)*

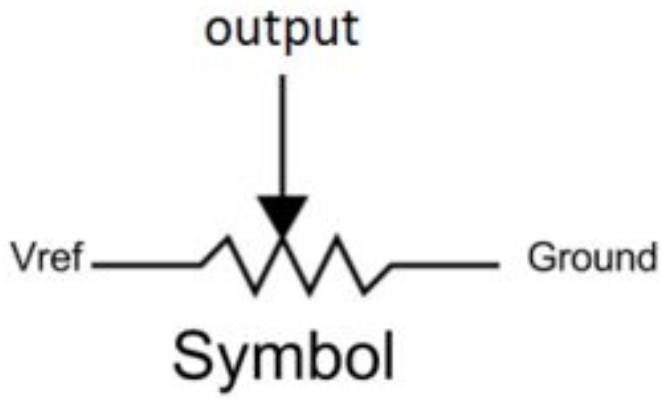
Demo due on: *November 16th(Two weeks later)*

# Task of your code ?

- To control the brightness of your LED0 (PB5) using the potentiometer
- The brightness must be proportional to the position of the wiper in the potentiometer given in your kit

# Variable resistor / potentiometer Connection



# Hints to build your code

- Initialize timer used to generate an interrupt every (X+1) ms *(Refer Lab 09 description file for X, Y, Z)*
- Initialize Timer used for PWM
- Initialize ADC
- Declare Pin as output to check the sample rate (X+1) ms interrupt & Set LED as output too
- Use sei() enable global interrupts
- *In an infinite while loop- give the input to control the timer and generate PWM*

**Timer** (8-bit) **for every *(X+1)* ms to read the ADC:** reads ADC every ***(X+1)*** ms by calling the ISR and hands over the value to a global variable.

$$\mathrm{TOV_{CK}} = \frac{\mathrm{f_{clk}}}{\mathrm{PVal \cdot OCR0A}}$$

$$\mathrm{OCR0A} = \frac{\mathrm{f_{clk}}}{\mathrm{PVal} \cdot \mathrm{TOV_{CK}}}$$

$$\mathrm{OCR0A} = \frac{16000000}{1024 \cdot \frac{1000}{\mathbf{X+1}}}$$

**Timer for PWM based on the given analog input:** varies the duty cycle based on value read from ADC.

$$\mathrm{OCR2A} = \frac{\mathrm{f_{clk}}}{\mathrm{f_{OCR0A}} \cdot 2 \cdot \mathrm{N}} - 1$$

# Sample function for ADC initialization:

```
void adc_init(void)
{
DDRC &= ~(1 << PINCY); // Setting input
ADMUX |= (1 << REFS0) | (1 << MUX1) | (1 << MUX0); //VCC reference
ADCSRA |= (1 << ADEN) | (1 << ADATE) | (1 << ADIE) | (1 << ADPS1) | (1 << ADPS0);
// Enable ADC Auto Trigger & Conversion Complete Interrupt
ADCSRB |= (1 << ADTS1) | (1 << ADTS0); //Compare Match
}
```

```
ISR(ADC_vect)
{
        uint16_t variable = ADC;
        a = variable; //passed to OCR1B
        PINC ^= (1<<Y); //toggle the pin
}
```

# Checkpoint for the Lab

- Send a character on the Serial I/O port corresponding to the value read by the ADC Pin (ReadValue/MaxValue*100) in percentage:
  - If 0% to 10 %, Transmit “0”
  - If 10% to 20 %, Transmit “1”
    :
    :
  - If 90% - 100%, Transmit “9”
- ADC Conversion to be done by busy wait instead of interrupts

**Go ahead and try it yourself now !**