**CSCE 236 Embedded Systems, Spring 2019**
**Lab 3**

Due: Tuesday, Feb 26th, 2019 (end of class)

**Name: _____**

## 1. Instructions

This Lab is an individual assignment, collaboration is not allowed. Show your work and describe your reasoning to get partial credit if your solution is incorrect. Complete all of the sections below and make sure to get the instructor or TA to sign off where required. You should keep your own notes on what you complete since parts of future homework will build on this lab.

**Include a printout of the code with your assignment and turn in a copy of your code on Canvas!** Failing to electronically turn in your code will result in up to a 20 point penalty on this assignment. Points may also be deducted for coding errors, poor style, or poor commenting. You will build a different functions for each portion of this lab and comment out the function call when not being used.

## 2. Robot Assembly

Assemble your Robot kit. Mount your power switch, motors, Arduino Uno, servo, ultrasonic sensor, battery pack, and L298N Motor Driver onto the robot chassis. Once you mount the parts on the chassis begin wiring them up to the Arduino.

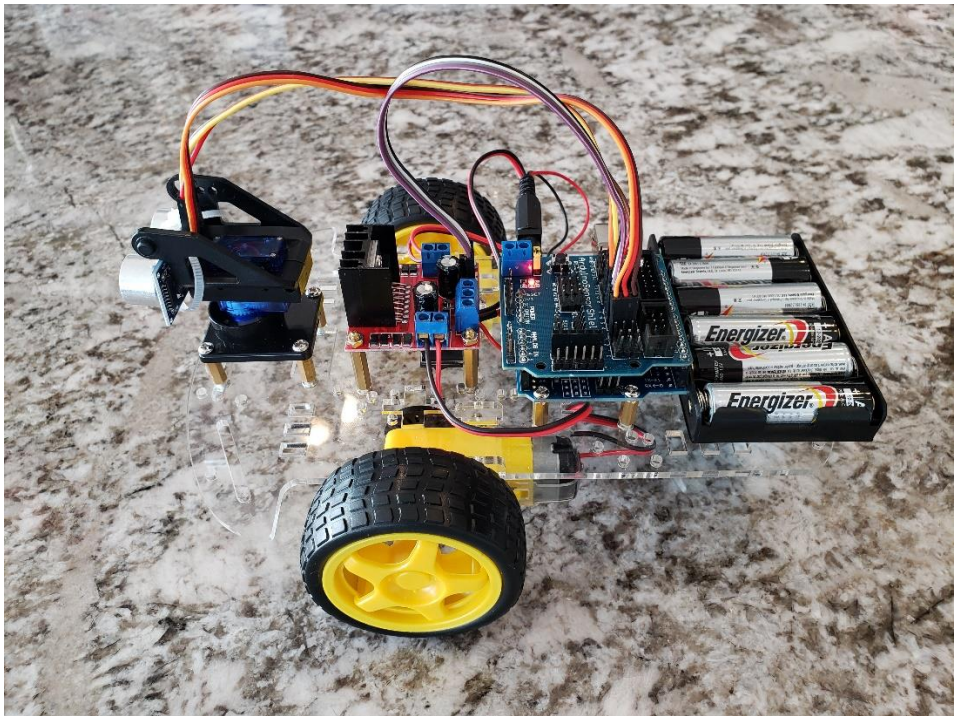


Figure 1: Assembled Robot

**Checkoff:** *After you mount the parts on the robot chassis. Initially you only need to wire your robot up to power up the Arduino, servo, and ultrasonic sensor.*

Instructor sign off: _____

## 3. "Robot Sensing"

Overview:

This lab is designed to assist you in learning the concepts associated with the input capture features for your Arduino. A single ultrasonic rangefinder is attached to a servo motor that can rotate. You will program your Arduino to use the rangefinder to determine whether your mobile robot is approaching a wall in front or on one of its sides. The skills you will learn from this lab will come in handy in the future as you start interfacing multiple systems.

Each robot has one rangefinder/servo pair. By using the headers available to you on the top of the Arduino Sensor Shield, you can read a pulsewidth that is proportional to the distance between an object and the sensor. You will need to send the servo a pulse of a particular length in order to rotate the sensor to each side and back to the center position.

Servo:

This servo enables angular control of the output shaft (e.g. between -90 degrees to +90 degrees). We will use PWM to control the angular position of the output shaft. Most servos use PWM signals with a period of approximately 20 milliseconds. By varying the pulse length between 1ms to 2ms the servo will go from -90 degrees to 0 degrees to +90 degrees as shown below in Figures 4 and 5.

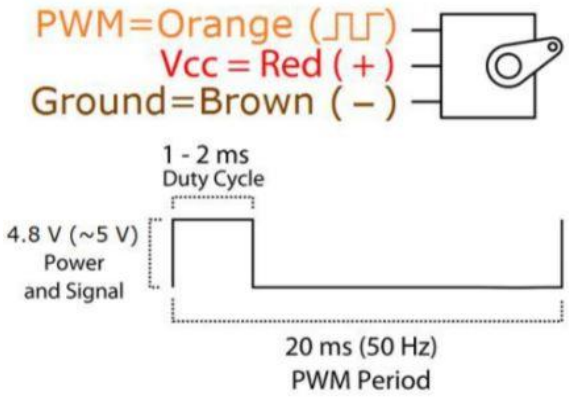


Figure 2: Servo Duty cycle to PWM

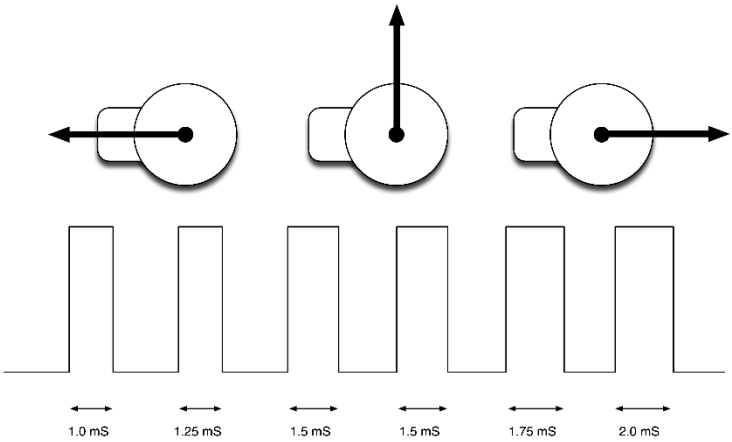


Figure 3: Servo angular position based on pulse length.

We will now generate the required 50 Hz PWM signal using a timer on our Arduino Uno to generate the required output duty cycle to move the servo between

**Checkoff:** *Write code that generates a PWM signal to move the servo between -90 degrees to 0 degrees to +90 degrees and back. If you have trouble with this please ask for help!*

Instructor sign off: _____

Ultrasonic Sensor Boards:

**Read the datasheets and answer the following questions:**

1. How fast does the signal pulse from the sensors travel?

2. If the distance away from the sensor is 1 in, how long does it take for the sensor pulse to return to the sensor? 1 cm?

3. What is the range and accuracy of the sensor?

4. What is the minimum recommended delay cycle (in ms) for using the sensor? How does this compare to the "working frequency"?

**Below are some things you should think about as you design your interfaces:**

- **Consider if/how you will configure the input capture subsystem** for your ultrasonic sensor. What are the registers you will need to use? Which bits in those registers are important? What is the initialization sequence you will need? Should you put a limit on how long to sense? If so, how long makes sense given the limitations of the sensor (or the requirements of the project)?

- **Consider the hardware interface.** Which signals will you use? Which pins correspond to those signals? How will you send a particular pulse width to the servo?

- **Consider the software interface you will create to your sensor.** Will you block or use interrupts? Will you stop moving while you sense?

- Will the ultrasonic sensor be ready to sense immediately after the servo changes position? How do you know?

- How long should you keep sending PWM pulses? Keep in mind that you may have to send many PWM pulses before your servo is in the correct position. Even after that, can you stop sending PWM pulses?

- **Consider how to make your code extensible.** It will be easier to achieve the bonus functionality of creating an ultrasonic library if you design it right from the beginning. You should also consider how you will make the sensor and servo work together.

**Required Functionality**

Learn to interface with the ultrasonic sensor and servo using your onboard timers. You will use one of the 8-bit timers to send create PWM signal to drive the position of the Servo without using the Arduino servo library. Additionally, will use the Timer 1 input capture pin to measure the echo

pulse width from the Ultrasonic sensor. You may use the delayMicroseconds() function to send out the 10 microsecond trigger on one of your digital output pins.

You will use the Timer 1 subsystem to light LEDs based on the presence of a wall/object. The presence of a wall on the left side of the robot should light LED1 on your Launchpad board. The presence of a wall next to the right side should light LED2 on a breadboard. A wall in front should light both LEDs. Demonstrate that the LEDs do not light until the sensor comes into close proximity (i.e. within 12 inches) with a wall.

***Checkoff:*** *The appropriate LED lights based on the proximity of the wall on left (LED1), right (LED2), for Front (both LEDs)?*

Instructor sign off: _____

***B Functionality***

*You need to fully characterize the sensor for your robot. Create a table and graphical plot with at least three data points that shows the rangefinder pulse lengths for a variety of distances from a wall to the front side of your robot. This table/graph should be generated for only* ***one servo position****. You can output the calculated distance measurements in inches as an output on the serial port. Use these values to determine how your sensor works so you can properly use it with the servo to figure out distances so that you could effectively navigate your robot through a maze.*

***Checkoff:*** *Show the instructor the output distance displayed on your serial port?*

Instructor sign off: _____

***A Functionality***

*Create a standalone library that includes a header and implementation file for your ultrasonic/servo code and upload it on Canvas. You can call them ultrasonic.h (header) and ultrasonic.c (implementation). It should have a thoughtful interface and be capable of being reused in the robot maze project.*