ENME 461 Lab 3- Actuators and Sensors: Micro Servo, Ultrasonic distance sensor, and Infrared line tracking sensor

1. Introduction

In this lab, you will use the Arduino to command the position of a micro servo and to acquire readings from the ultrasonic distance sensor and a line tracking sensor.

To complete the lab, show all completed projects to your teaching assistant.

2. Objectives

- To position a micro servo using Arduino libraries and a potentiometer input.
- To understand how an ultrasonic distance sensor works and to acquire readings from it.
- To understand how a line tracking sensor works and to acquire readings from it.

3. Project One: Micro Servo Sweep

The objective of this project is to have the servo motor arm perform a half-revolution sweep in discrete steps of 5 degrees and back. The servo arm will start in the position of zero degrees, move in discrete steps of 5 degrees to its 180° position, and dwell for a half second. It will then return from the 180° position to the zero-degree position in discrete steps of 5 degrees each, dwell for another half second, and continue in this manner indefinitely.

3.1. Micro Servo motor

Micro Servo Motor SG90 is a tiny and lightweight server motor with high output power. They are inexpensive, low voltage (about 5 VDC) motors and are available in standard, mini-micro, and quarter scale sizes. The micro servo in the lab is shown in **Figure 1**. Micro servos generally include four components packaged together: a small DC motor, a gear train, a potentiometer, and a control board.

Servos have three leads for power, ground, and control signal. The power signal ranges from 4.8 to 6V and can be conveniently obtained from battery packs. The servo in the lab has a specified motion range of 0 to 180°, but our experiments show that this range may not be obtained on all servos. The control signal is a PWM signal (5V) at a frequency of about 50Hz. This signal can be generated by the Arduino microcontroller. The pulse width of the control signal controls the position of the servo. Based on some experimentation, it was found that this signal ranges from about 0.5 to about 2.5 milliseconds for the servo in the lab. At about 0.5 ms, the servo is at one extreme of its motion range, corresponding to an angle of 0° while at 2.5 ms it is at the other extreme, corresponding to about 180°. At 1.5 ms pulse width, the servo is in the center or mid-position, corresponding to the nominal 90°. The pulse width and corresponding servo position are shown in **Figure 2.** Micro servos operate on the basis of feedback control of position.

Figure 1 – Micro Servo motor

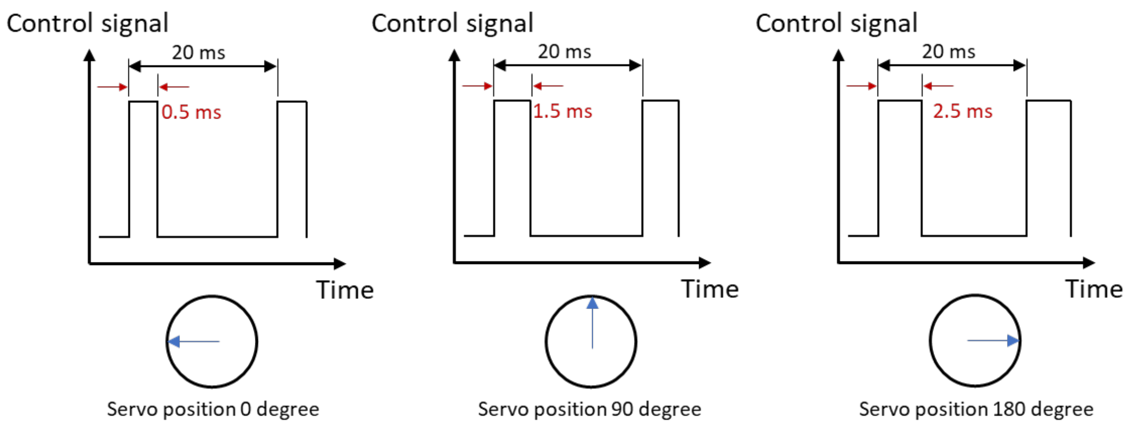


Figure 2 – Pulse sent to servo and corresponding position

3.2. Equipment

The equipment needed for this project are listed below.

- Arduino pack
- Battery pack
- Micro servo with long arm attached
- Wires

3.3. Connection scheme

The servo has three leads colored Brown, Red and Orange. Brown is connected to ground, Red is connected to the positive terminal and Orange is control signal from a PWM-enabled Arduino digital pin. The servo is powered from the battery pack. It may be possible to power the servo using the Arduino 5V supply. When powering the servo from the battery pack, a common ground must be established.

3.4. Program Concept

Although it is possible to manually program a servo motor, it is much easier to use the servo library included in Arduino IDE. Libraries are pieces of program written by other users to extend and simplify the programming process. The servo library is accessible as "Servo.h". To use, it must be included in the program. This is done either using the command `#include <Servo.h>`at the beginning of the program or choosing "Sketch", "Include Library" and "Servo".

In this library, an object of type "Servo" must be declared at the beginning of program to identify the servo motor. This done using the command `Servo labservo` which defines an object `labservo` as the servo motor being used. In the setup(), the servo object `labservo` must be associated with the digital pin controlling the servo. This is done using the command `labservo.attach(pin)` where pin is the PWM- enabled digital pin where the servo control signal is connected.

`labservo.write(angle)` is the command that will move the servo to the specified angle in degrees.

Optionally, `labservo.read()` reads the value passed to the most recent write() command to indicate the position of servo.

Other commands in this library can be found from the online reference for the Servo library.

The program is then based on loops increasing the angle of the servo from 0 to 180 in steps of 5 for a forward motion and decreasing angles for the return motion.

Note that for smooth motion, delays of appropriate size must be implemented after each rotation step.

4. Project Two: Control the position of the servo arm using a potentiometer

Similar to previous projects, use the multi-turn (big blue) potentiometer (the manufacturer calls it a rotation sensor) to control the position of the servo, where the extreme ends of the servo rotation correspond to the ends of the potentiometer knob turn.

4.1. Equipment

Same as Project One with the addition of the multi-turn potentiometer. This potentiometer is identical in operation to the potentiometers used before. It is larger and has a much wider range. The leads are color coded: red for voltage input, black for ground, green (or blue) for analog output to microcontroller.

5. Project Three: Ultrasonic distance sensing

The objective of this project is to use the ultrasonic distance sensor to measure distance (in cm) from an object and display the distance on the Serial Monitor.

5.1. HC-SR04 Ultrasonic sensor

The HC-SR04 Ultrasonic Sensor, shown in Figure 3, an inexpensive sensor that can measure distance between an object and the sensor. The module includes ultrasonic transmitters, receiver and control circuit.

- An Ultrasonic Transmitter – This transmits the ultrasonic sound pulses, it operates at 40 KHz
- An Ultrasonic Receiver – The receiver listens for the transmitted pulses. If it receives them it produces an output pulse whose width can be used to determine the distance the pulse travelled.

Figure 3 – URM37 Ultrasonic sensor

The HC-SR04 has the following four connections:

- VCC – This is the 5 Volt positive power supply.
- Trig – This is the "Trigger" pin, the one driven to send the ultrasonic pulses.
- Echo – This is the pin that produces a pulse when the reflected signal is received. The length of the pulse is proportional to the time it took for the transmitted signal to be detected.
- GND – This is the Ground pin.

The device operates as follows:

1. A 5 volt pulse of at least 10 uS (10 microseconds) in duration is applied to the Trigger pin.
2. The HC-SR04 responds by transmitting a burst of eight pulses at 40 KHz. This 8-pulse pattern makes the "ultrasonic signature" from the device unique, allowing the receiver to discriminate between the transmitted pattern and the ultrasonic background noise.
3. The eight ultrasonic pulses travel through the air away from the transmitter. Meanwhile the Echo pin goes high to start forming the beginning of the echo-back signal.
4. If the pulse in NOT reflected back then the Echo signal will timeout after 38 mS (38 milliseconds) and return low. This produces a 38 mS pulse that indicates no obstruction within the range of the sensor.
5. If the pulse IS reflected back the Echo pin goes high when the signal is received. This produces a pulse whose width varies between 150 uS to 25 mS, depending upon the time it took for the signal to be received.

6. The width of the received pulse is used to calculate the distance to the reflected object. Remember that the pulse indicates the time it took for the signal to be sent out and reflected back so to get the distance you'll need to divide your result in half.

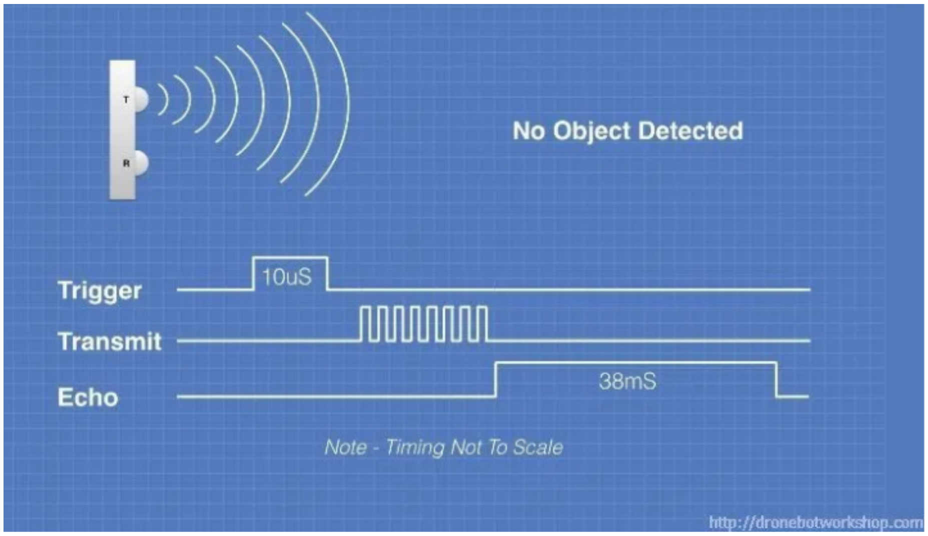


Figure 4– URM37 Ultrasonic sensor signal and no object detected

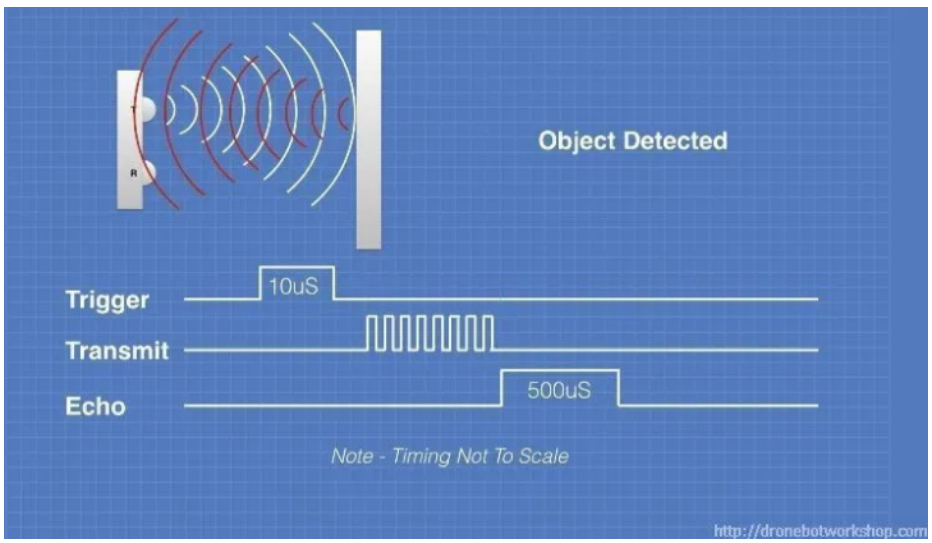


Figure 5– URM37 Ultrasonic sensor signal and object detected

The operating parameters for the ultrasonic sensor can be observed in the following table:

Table 1. Operating parameters

Detecting Range	2 cm- 4 m
Working Voltage	5 V
Working Current	15mA
Resolution	3 mm

5.2. Equipment

The equipment needed for this project are listed below.

- Arduino pack
- HC-SR04 Ultrasonic sensor
- Wires

5.3. Program Concept

The ultrasonic distance measurement mode of operation for this sensor is as follows:

- Turning the TRIG pin from LOW to HIGH will trigger a burst of ultrasonic waves from one of the barrels.
- Upon receipt of the echo of this burst from the object whose distance is being sensed, the sensor will turn the ECHO pin on or off (not clear in the documentation).
- The length of time that the ECHO pin is OFF is used as a measure of distance.

The following steps are required to program this sensor:

- setup():

1- Set the pin that TRIG is connected to as an Output using the pinMode() command. 2- Set this pin to HIGH using the digitalWrite() command.

3- Set the pin that ECHO is connected to as an Input using the pinMode() command.

- loop():
  1- Set the TRIG pin to LOW.
  2- Set the TRIG pin to HIGH. This triggers the ultrasonic burst.
  3- Measure the time the ECHO pin is HIGH using "duration=pulseIn(pin,HIGH);" command. This command waits for the "pin" to go LOW, starts timing until the pin goes HIGH. It stores the length of time in **microseconds** in **"duration"** variable. The variable **"duration"** must be of type **long,** not **int**, since the length of time in microseconds can overflow memory dedicated to int.
  4- According to the manufacturer, every 50 microseconds corresponds to 1cm. Therefore, dividing **duration** by **50** gives the distance in centimeters**.**

6. Project Four: Infrared (IR) Line Tracking Sensing

The objective of this project is to use the IR line tracking sensor to track a line made of black tape.

6.1. TCRT5000 in the IR line tracking module

The TCRT5000 is an IR sensor unit. It has both a Photodiode and a Phototransistor coupled in its package. The photo diode has two pins (Anode and Cathode) which can be used to generate an IR signal. Similarly the Photo transistor also has two pins (Collector and Emitter) which can be used to read the IR signal that is reflected back. This sensor can be used to detect the presence of object or any other reflective surface in front it, also with some level of programming it can also calculate the distance of the object in front it. But the distance can only be calculated for short range objects (measuring distance range from 1mm to 8mm) and the central point is about 2.5mm. and is also subjected to environmental disturbance. So if you look for a sensor to measure distance of an object in front of it you should try the HC-SR04 Ultrasonic sensor.
Other than that this sensor is perfect for detecting the proximity of an object in front of it. It can also easily distinguish between black and white colour hence widely used in line following and maze solving robots.

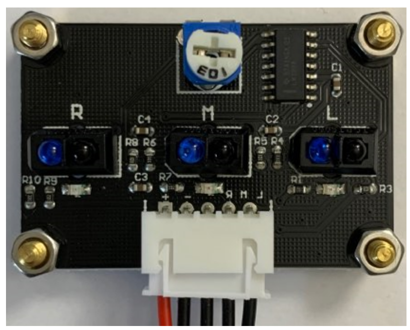


Figure 6 – IR line tracking module

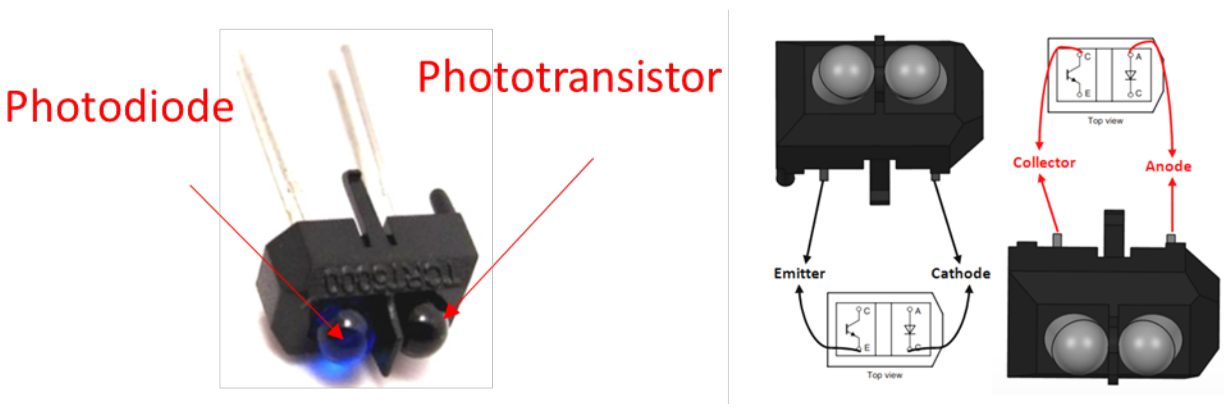


Figure 7 – Reflective Optical Sensor with Transistor Output (TCRT5000)

- Collector – The collector of the phototransistor is connected to +5V.
- Emitter– The emitter of the phototransistor is grounded through a resistor.
- Anode – The Anode of photodiode is connected to +5V.
- Cathode – The Cathode of photodiode is grounded through a resistor.

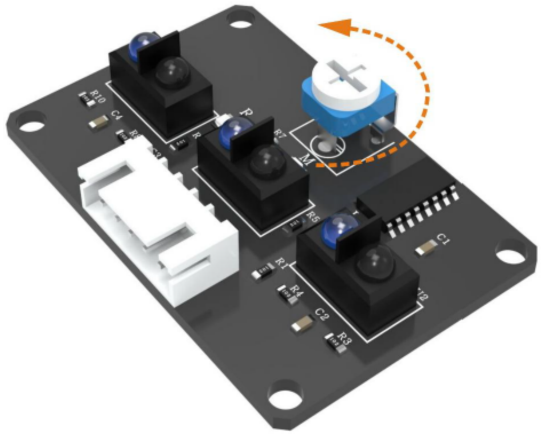


Figure 8– Potentiometer to adjust the sensitivity by changing its resistance value

6.2. Equipment

The equipment needed for this project are listed below.

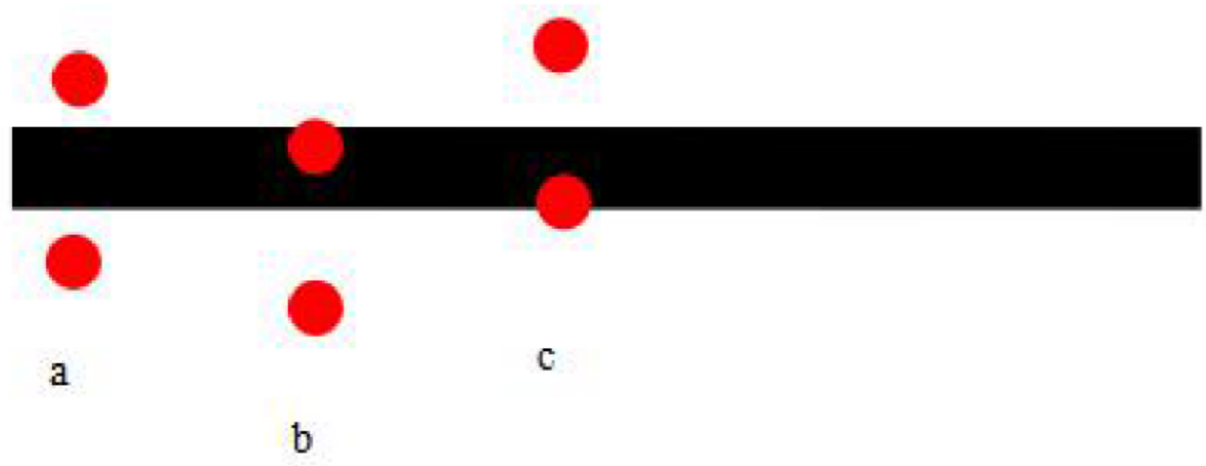
- Arduino pack
- IR line tracking module
- Black tape
- Wires
- LED

6.3. Connection scheme

The IR line tracking module has five pins; Left sensor output, Middle sensor output, Right sensor output, Ground(-), and VCC(+5V). It is possible to power the sensor using the Arduino 5V supply. This sensor can either be used as an analog sensor or as a digital sensor. However, we will only use the digital signal returned by this IR sensor. The line tracking sensor consists of an infrared transmitter tube and an infrared receiver tube. The former is a LED that can transmit the infrared ray, while the latter is a photoresistor which is only responsible for receiving the infrared light. When an infrared receiver sensed infrared light, "HIGH" status shows on the corresponding pin.; otherwise, the status of the pin remains "LOW". Light reflectance for the black surface is different from that for the white surface. Hence, whenever the sensor is placed too far from an objective, or against a black surface, almost no infrared ray can be sensed by the infrared receiver.

6.4. Program Concept

The objective of this project is using IR sensor to track a segment of black tape, and using two external LEDs to show the tracking status.



As shown in the figure#, two red pot represents left and right sensor, respectively. The complete project should show following functions:

- **a.** As the middle sensor detects the black tape, the status of tracking is good, so no external LED should be turned on.
- **b.** As the left sensor detects the black tape, the tracking deviates to left, so the LED representing “left sensor detects” should be turned on.
- **c.** As the right sensor detects the black tape, the tracking deviates to right, so the LED representing “right sensor detects” should be turned on.

7. Reference

- https://dronebotworkshop.com/hc-sr04-ultrasonic-distance-sensor-arduino/