WESTERN NEW ENGLAND UNIVERSITY
SPRINGFIELD, MASSACHUSETTS

DEPARTMENT OF MECHANICAL ENGINEERING

ME 455-41 APPLIED MECHATRONIC SYSTEMS
ME 656-41 ADVANCED MECHATRONICS SYSTEMS

Instructor: Dr. Vedang Chauhan

Project 2 Obstacle Detection Using Ultrasonic Sensor

Objective: Building on the previous Smart Mobile Robot work, the objective of this project is to use ultrasonic sensors for obstacle detection, range measurement, and sizing an obstacle.

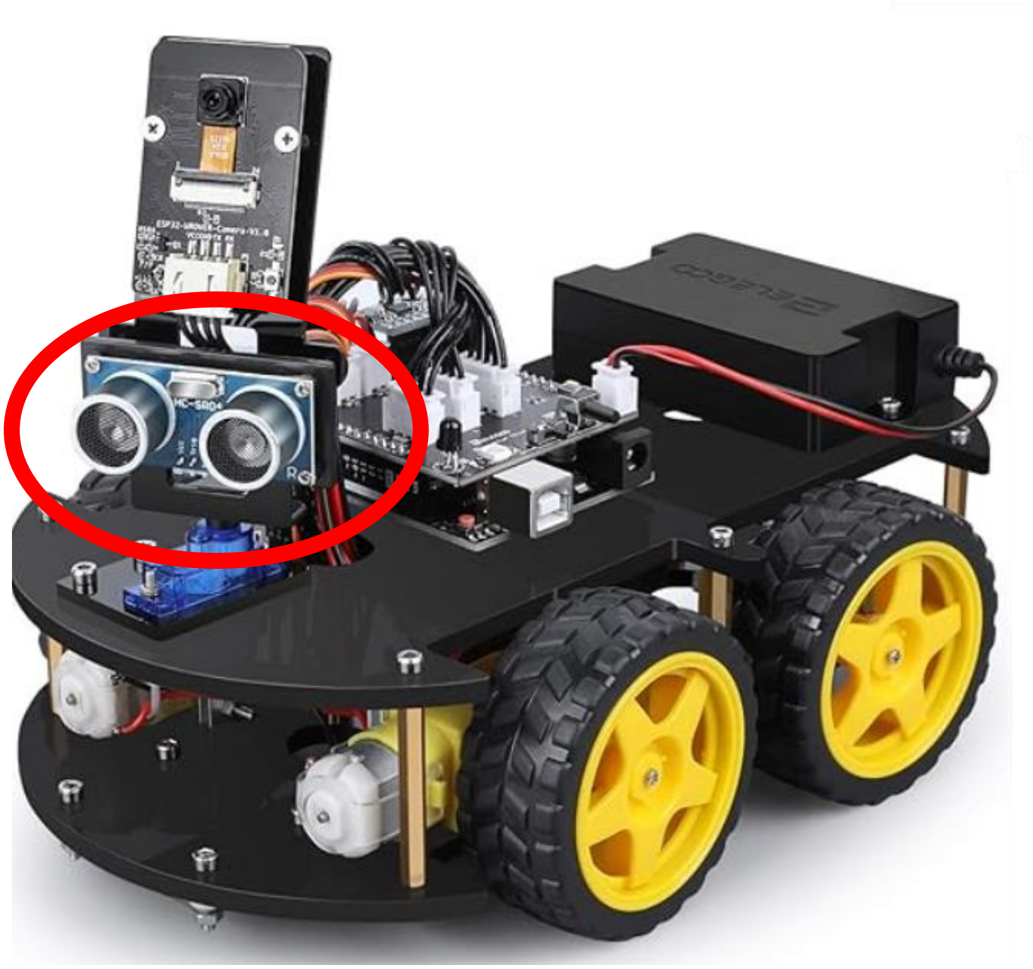


Figure 1 Smart Mobile Robot with an IR Remote and Ultrasonic Sensor (image course: *ELEGOO UNO R3 Smart Robot Car Kit*

Amazon.com)

1: Distance Measurement (without the robot motion)

For this part, you will calibrate and use an ultrasonic sensor to measure distance to an obstacle and print the distance on a serial monitor.

The steps to follow are:

- Place an obstacle in front of the robot at 12 inches.
- Measure and calibrate the ultrasonic sensor reading to accurately output the distance.
- Move the obstacle closer and farther from the robot and record the distance displayed on the serial monitor.
- Add your measurements to Table 1.

A sample circuit diagram for using an ultrasonic sensor is shown in Figure 2.

Figures 3 and 4 show basic ultrasonic measurement code for getting a direct distance return measurement from the ultrasonic sensor using trigger and echo pins.

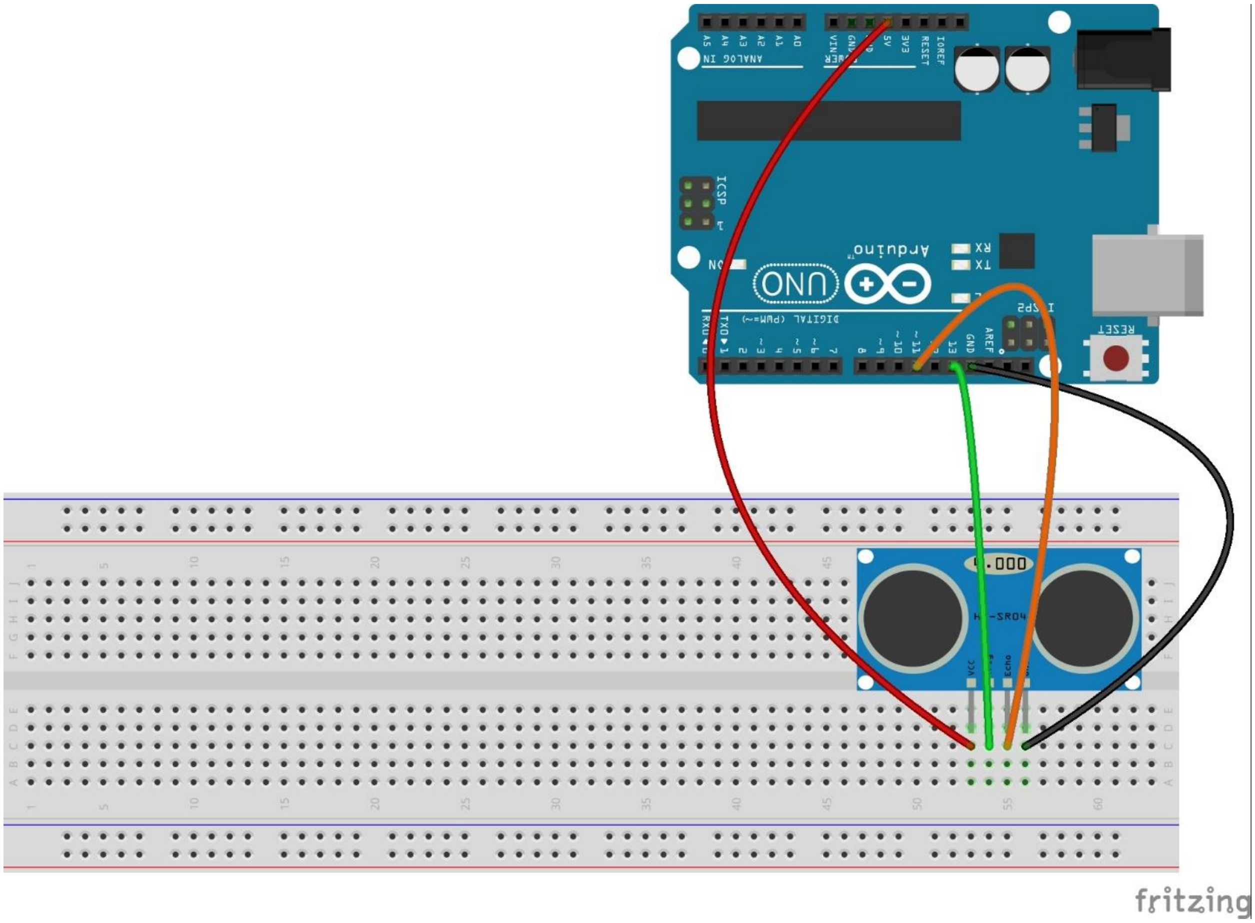
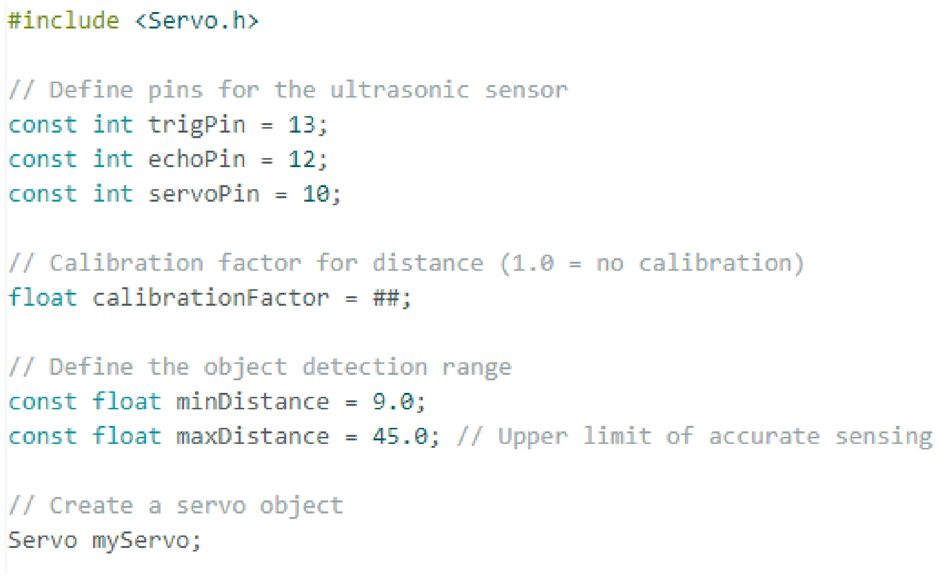
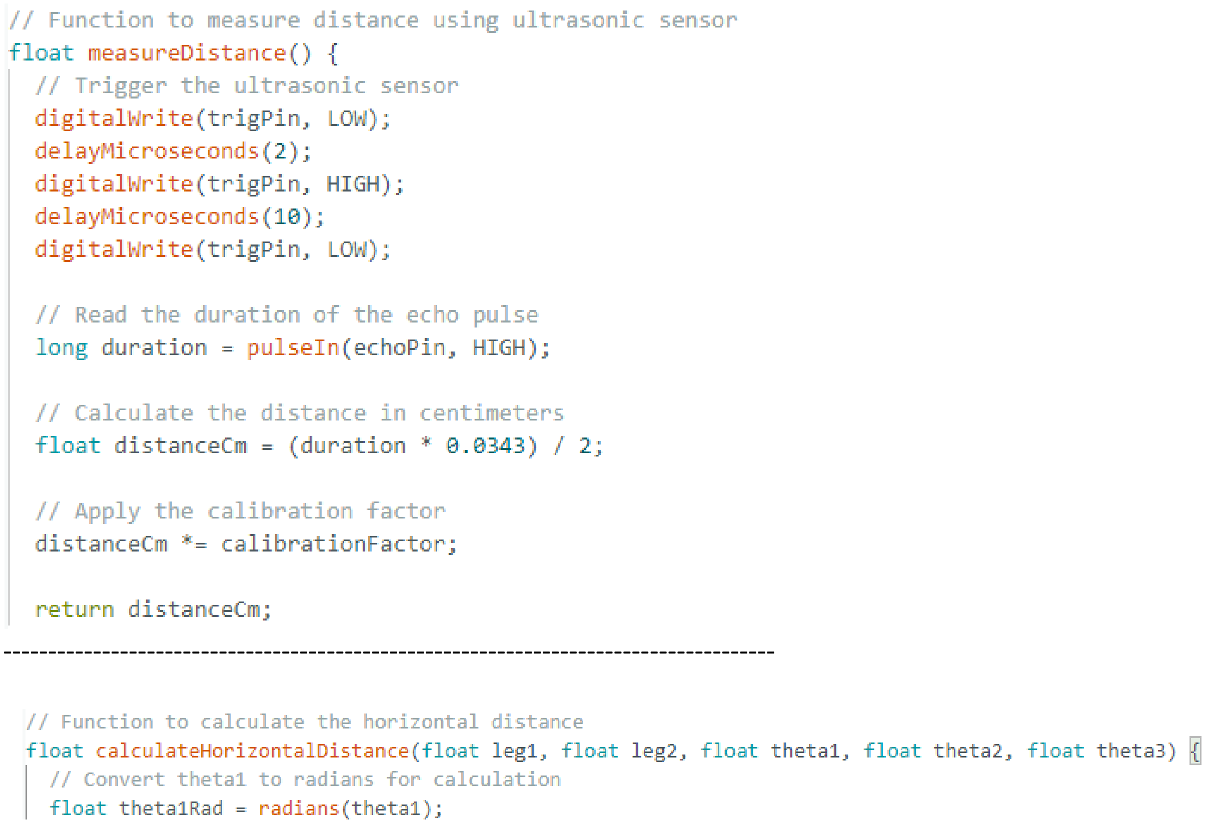


Figure 2: Connecting Ultrasonic Sensor to Arduino



```
#include <Servo.h>

// Define pins for the ultrasonic sensor
const int trigPin = 13;
const int echoPin = 12;
const int servoPin = 10;

// Calibration factor for distance (1.0 = no calibration)
float calibrationFactor = ##;

// Define the object detection range
const float minDistance = 9.0;
const float maxDistance = 45.0; // Upper limit of accurate sensing

// Create a servo object
Servo myServo;
```

Figure 3: Variables needed for set up



```
// Function to measure distance using ultrasonic sensor
float measureDistance() {
  // Trigger the ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the duration of the echo pulse
  long duration = pulseIn(echoPin, HIGH);

  // Calculate the distance in centimeters
  float distanceCm = (duration * 0.0343) / 2;

  // Apply the calibration factor
  distanceCm *= calibrationFactor;

  return distanceCm;
```

```
// Function to calculate the horizontal distance
float calculateHorizontalDistance(float leg1, float leg2, float theta1, float theta2, float theta3) {
  // Convert theta1 to radians for calculation
  float theta1Rad = radians(theta1);
```

Figure 4:SOME OF THE Variables needed for calculation

Table 1 Distance Measurement Using an Ultrasonic Sensor

#	Actual Distance	Measured Distance with Ultrasonic sensor
1	8 in	
2	10 in	
3	12 in	
4	16 in	
5	20 in	

2: Sizing an obstacle (without the Robot Motion)

You will use an ultrasonic sensor and a servo motor to rotate the sensor and measure the distance to an obstacle. The distances and angles of the servo motor will be used to measure the width of the obstacle as follows. Refer to Figure 5 for a schematic diagram.

- Write a program that can sweep servo motor from 0° to 180° and measures distance to an obstacle using the ultrasonic sensor.
- Your program should record the servo angle and distance when the first side edge of the obstacle is detected.
- Continue sweeping the servo and record the angle and distance when the last edge of the obstacle is detected.
- Using the measured distances and the angle between the edges of the obstacle, calculate the size of the obstacle using the Law of cosines (Figure 6) and print the size on the serial monitor.
- Use different objects as obstacles and record their sizes using the program. Add your values to Table 2.

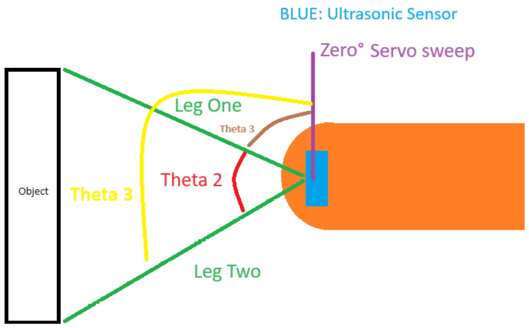
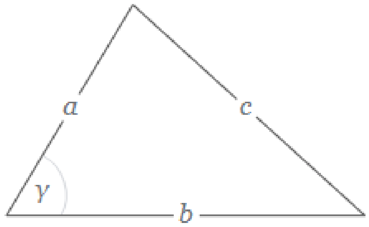
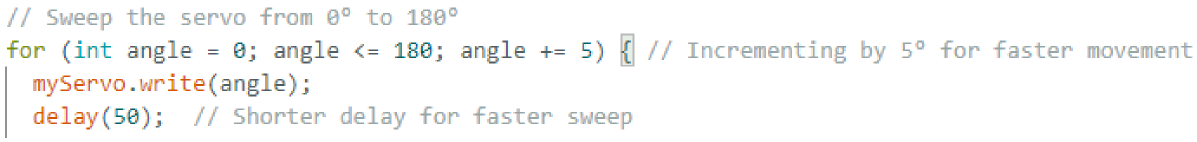


Figure 5: Object width detection



$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma}$$

Figure 6 Law of cosines



```
// Sweep the servo from 0° to 180°
for (int angle = 0; angle <= 180; angle += 5) { // Incrementing by 5° for faster movement
  myServo.write(angle);
  delay(50);  // Shorter delay for faster sweep
```

Table 2 Sizing an obstacle

#	Actual Width	Measured Width
1		
2		
3		
4		
5		

Deliverables and assessment rubric:

Submit onto Kodiak your final PDF report (file name: Project3_YourGroupNumber.PDF”) that contains • Photos of your task 1 and task 2 set ups. • Table 1 and Table 2. • Arduino Program • A video link of task 2.	**80**
A paragraph on your learning experience and challenges you faced.	**10**
Quality and professionalism of the report.	**10**

***