Drexel University

To: Dr. Peters

From: Tyler Ostinato

Date: Nov 1, 2021

Re: ECE 303-Lab 6-Motors

Purpose

Create a collision avoidance system by utilizing an ultrasonic sensor and DC motor. The goal of this lab will be to create a system that slows down a DC motor as an object gets closer to a super sonic sensor.

Discussion

Ultrasonic Sensor | 1.2

To get the ultrasonic sensor working on the Arduino Mega 2560 I referred to create.arduino.cc. This website gave me a basic diagram to connect the ultra-sonic sensor to the board as well as outputting distance to the serial terminal (Figure 1).

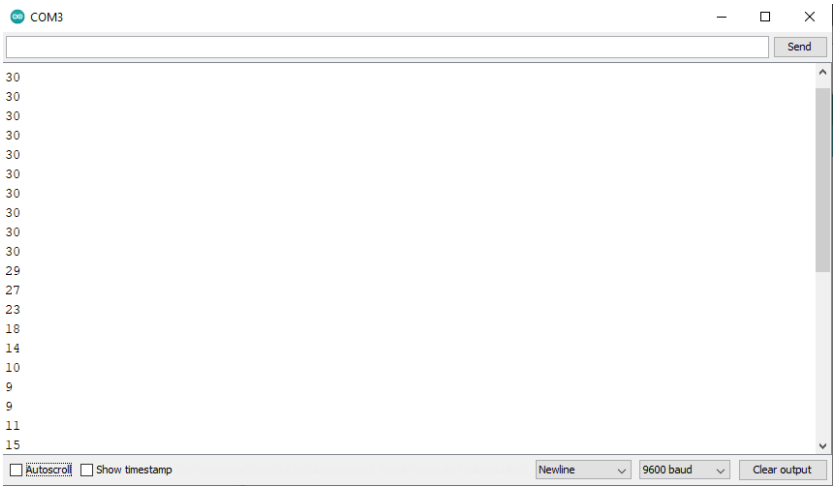

Figure 1: Serial output as object gets closer to ultra-sonic sensor.

Angle of Acceptance | 1.3

According to the datasheet of the ultrasonic sensor provide in our kit the effective angle of acceptance is 30 degrees. Using my ultra-sonic sensor I was able to measure an object 30cm away aim 5-degree increments. After about 15 degrees left and right the sensor can no longer see the object (values represented as zeros in this case) which is exactly what we would expect according to the datasheet (Figure 2).

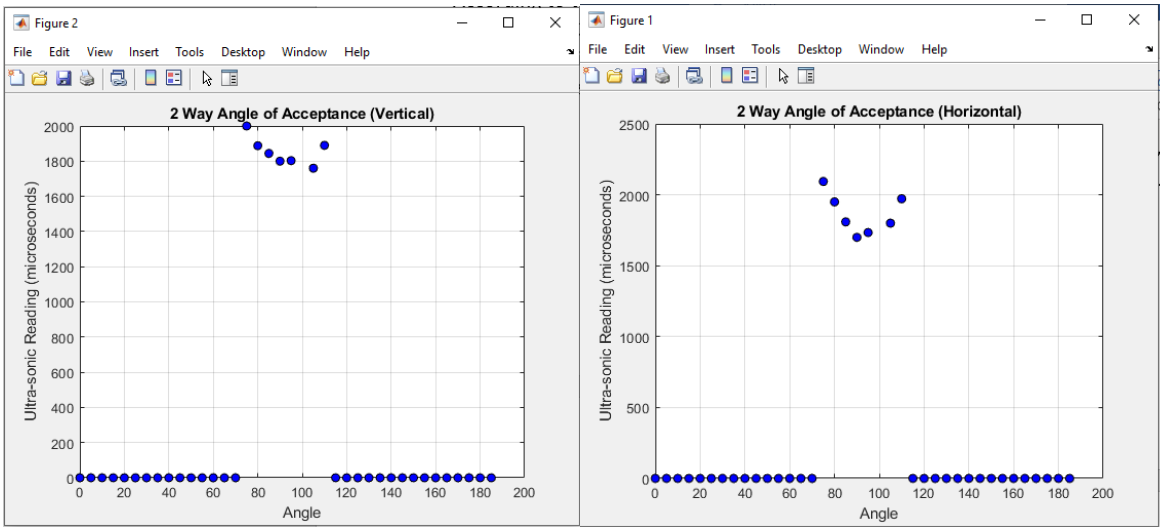

Figure 2: Ultra-sonic sensor angle of acceptance.

DC Motor | 1.4

Next I set up the DC motor on my board. The DC motor needs more voltage than the Arduino can supply so we must use a separate power source to power the motor. After attaching this to the board we need to wire the motor to an H bridge and connect the H bridge to the Arduino and the power supply (Figure 3).

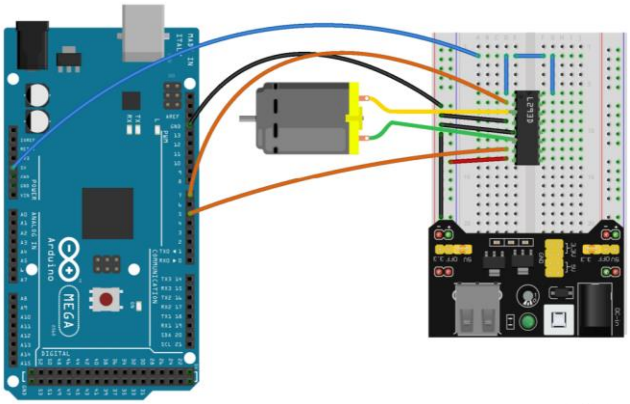

Figure 3: Schematic DC motor, Power Supply, and H bridge connected to Arduino Mega 2560.

LEDs | 1.5

Lastly we need to at a few LEDs that will serve as an indicator of how close the object is to the ultra-sonic sensor. I used 3 LEDs each wired in series to separate pins on the Arduino.

Bring it all together | 1.6

Now that all the components are set up, we nee them to talk to each other. First we can control the speed of the motor by mapping it to the output of the ultra-sonic sensor using Arduino's map() command. This will allow the motor to spin at a proportional rate to the distance of the object. When the object is very close the motor should stop and if it is greater than 30cm it should spin at full speed (Figure 4).

```
// Change motor speed based on 0-30cm distance
motor_speed=map(distance,0,30,0,255);
analogWrite(forward,motor_speed);
```

Figure 4: Mapping ultrasonic-sensor distance to DC motor speed.

Next we need to add the LED functionality. To do this I set a few if else statements to turn on the LEDs at varying distances. For example when the object is greater than 30cm away all 3 LEDs will be set to LOW. If the object is less than 10cm away all 3 LEDs will be set to HIGH (Figure 5).

```
// Set LEDs to change as ditance gets closer
if(distance < 10){
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, HIGH);
}
else if(distance > 10 && distance < 16){
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, LOW);
}
else if(distance > 16 && distance < 24){
  digitalWrite(led1, HIGH);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
}
else if (distance > 24){
  digitalWrite(led1, LOW);
  digitalWrite(led2, LOW);
  digitalWrite(led3, LOW);
}
```

Figure 5: Set LED outputs based on object distance.

Now that everything is wired correctly the system will now change the motor speed of an object and give an indicator of how close the object is automatically!

Conclusion

In this lab we successfully wired many components together to create a collision detection system. We used various components of previous labs plus new concepts of DC motors and ultrasonic sensors. This lab will also provide context for the final project in the class which involves a collision detection system.

Appendix

Arduino Sketch

```
// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement
int motor_speed=0;
int led1=8;
int led2=9;
int led3=10;

const int forward=5;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);

  pinMode(forward,OUTPUT);
```

```
}
void loop() {
 // Clears the trigPin condition
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
 digitalWrite(trigPin, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 // Reads the echoPin, returns the sound wave travel time in microseconds
 duration = pulseIn(echoPin, HIGH);
 // Calculating the distance
 distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
 // Displays the distance on the Serial Monitor
 Serial.print("Distance: ");
 Serial.print(distance);
 Serial.println(" cm");
 delay(250);
 if(distance > 30){
  distance = 30;
 }
 // Change motor speed based on 0-30cm distance
 motor_speed=map(distance,0,30,0,255);
 analogWrite(forward,motor_speed);

 // Set LEDs to change as ditance gets closer
 if(distance < 10){
  digitalWrite(led1, HIGH);
  digitalWrite(led2, HIGH);
  digitalWrite(led3, HIGH);
 }
```

```
  else if(distance > 10 && distance < 16){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, LOW);
  }
  else if(distance > 16 && distance < 24){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
  }
  else if (distance > 24){
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
  }

}
```