MEMORANDUM

TO: Dr. Christopher Peters
FROM: Yelnur Abilakim
DATE: 4 Nov 2021
SUBJECT: ECE-303 Lab 6: Motors

Summary
In this project, a collision avoidance system is simulated using the ultrasonic sensor and LED, both of which are included in the kit. The LED is off until an object passes a threshold distance, at which point LED turns on.

Introduction
Collision avoidance is used in modern automobiles. It uses sensors (light, sound, etc.) to determine if there is something in front of the car. If the car needs to slow down, signals will control servos to reduce fuel into the engine, and autonomously control the brakes.

Methods
Experimental apparatus used in this lab:

- Breadboard.
- Arduino MEGA 2560 connected as follows:
  - To laptop via USB cable.
  - To outlet via power cable.
- One 220Ω resistor.
- One LED.
- One ultrasonic sensor.
- Six jump wires.

The circuit schematic used for this lab can be seen in Figure 1.

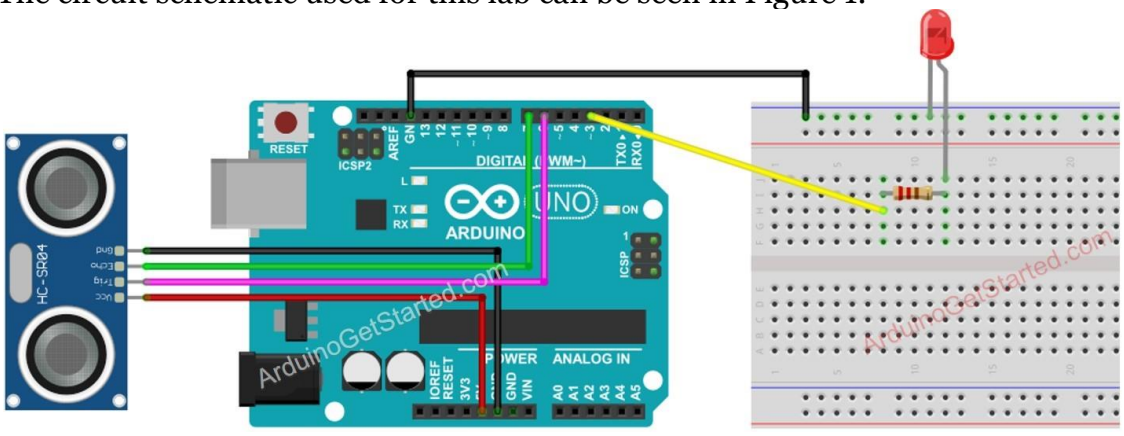

Figure 1. Circuit schematic

Results
The physical circuit built according to the schematic is shown in Figure 2.

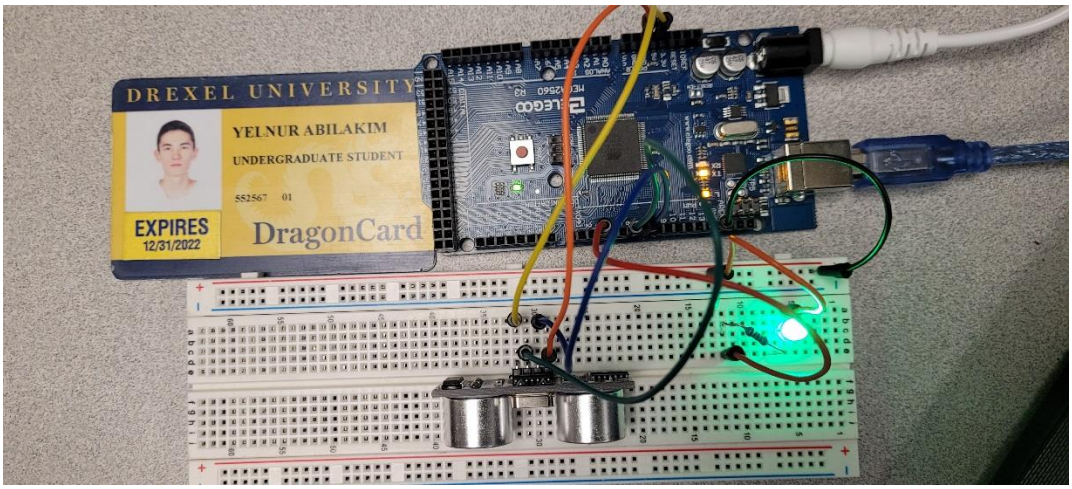

Figure 2. Physical circuit

The LED lights up when an object is within 10cm from the ultrasonic sensor. The Arduino code written for this experiment is appended at the end.

Conclusions/Recommendations

Ultrasonic sensor is a useful tool for analyzing environment for obstacles. It can be implemented with Arduino. LED light acts as a clear indicator of the sensor's findings.

Appendices

Figures 3 and 4 demonstrate the Arduino code for the experiment.

```
const int TRIG_PIN = 6; // Arduino pin connected to Ultrasonic Sensor's TRIG pin
const int ECHO_PIN = 7; // Arduino pin connected to Ultrasonic Sensor's ECHO pin
const int LED_PIN  = 3; // Arduino pin connected to LED's pin
const int DISTANCE_THRESHOLD = 5; // centimeters

// variables will change:
float duration_us, distance_cm;

void setup() {
  Serial.begin (9600);       // initialize serial port
  pinMode(TRIG_PIN, OUTPUT); // set arduino pin to output mode
  pinMode(ECHO_PIN, INPUT);  // set arduino pin to input mode
  pinMode(LED_PIN, OUTPUT);  // set arduino pin to output mode
}

void loop() {
  // generate 10-microsecond pulse to TRIG pin
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  // measure duration of pulse from ECHO pin
  duration_us = pulseIn(ECHO_PIN, HIGH);
  // calculate the distance
  distance_cm = 0.017 * duration_us;

  if(distance_cm < DISTANCE_THRESHOLD)
    digitalWrite(LED_PIN, HIGH); // turn on LED
  else
```

Figure 3. Arduino code (part 1)

```
  Serial.begin (9600);       // initialize serial port
  pinMode(TRIG_PIN, OUTPUT); // set arduino pin to output mode
  pinMode(ECHO_PIN, INPUT);  // set arduino pin to input mode
  pinMode(LED_PIN, OUTPUT);  // set arduino pin to output mode
}

void loop() {
  // generate 10-microsecond pulse to TRIG pin
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  // measure duration of pulse from ECHO pin
  duration_us = pulseIn(ECHO_PIN, HIGH);
  // calculate the distance
  distance_cm = 0.017 * duration_us;

  if(distance_cm < DISTANCE_THRESHOLD)
    digitalWrite(LED_PIN, HIGH); // turn on LED
  else
    digitalWrite(LED_PIN, LOW);  // turn off LED

  // print the value to Serial Monitor
  Serial.print("distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");

  delay(500);
}
```

Figure 4. Arduino code (part 2)