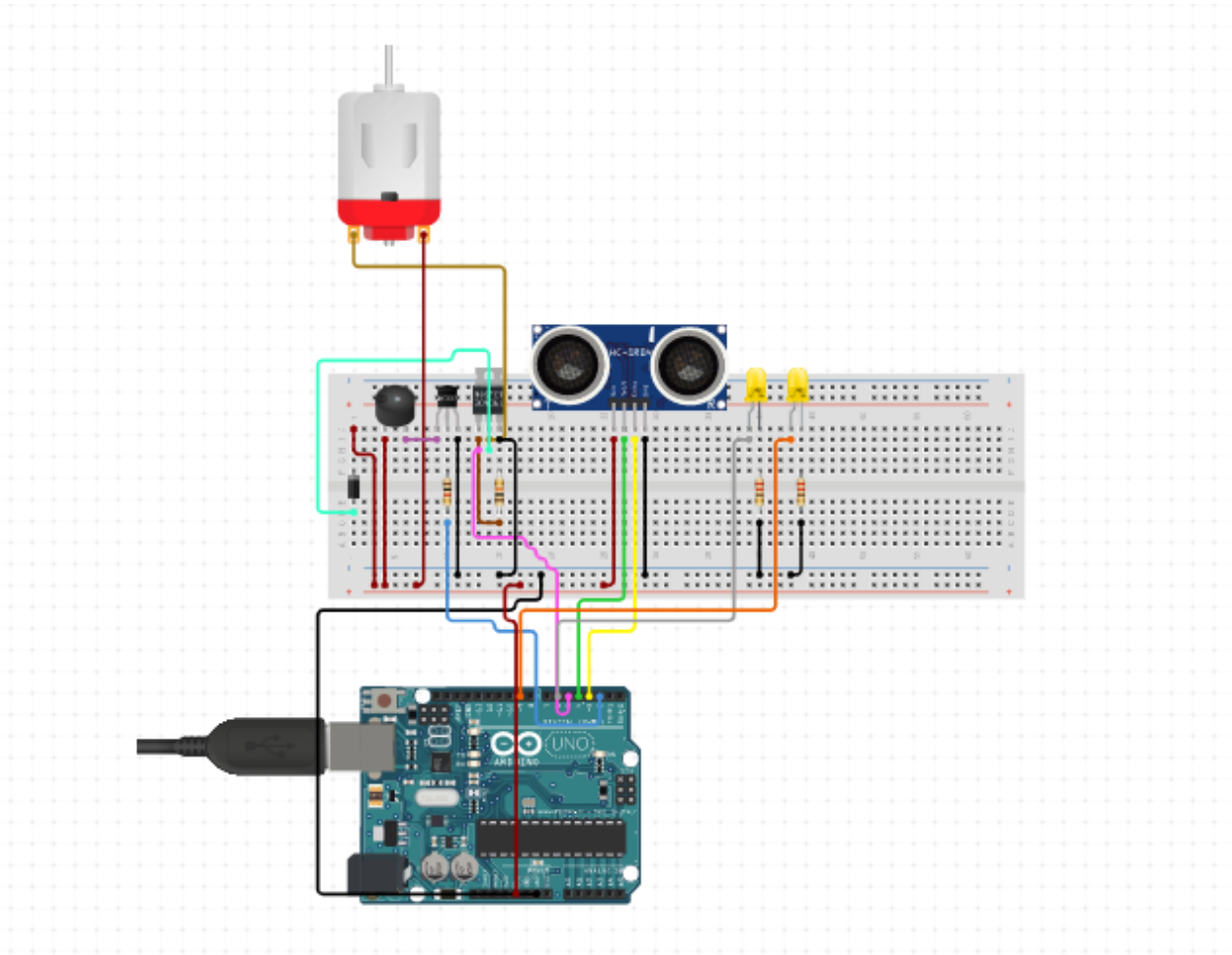
Introduction

The IoT project for this assignment is called Car Braking System. This project is meant to resemble the Automatic Braking system that are common in more newer automobiles today. This project makes use of an ultrasonic sensor to get input of the distance between the vehicle and the object in front of it. Here a dc motor as an actuator is used to simulate the car engine running, which will be controlled by our raspberry pi virtual machine based on the input received from the ultrasonic sensor, turning off the dc motor once the vehicle is too closed to the object to simulate it the vehicle slowing down or stopping. We also make use of a simple active buzzer to simulate as the brakes of the car which will also be controlled by our raspberry pi virtual machine. The raspberry pi virtual machine will enter the input receive from the ultrasonic sensor into the database, check the latest entry in the database to see if the distance is to close or not, and then sends the appropriate signal to the Arduino board in order to control the dc motor and buzzer.

Conceptual Design

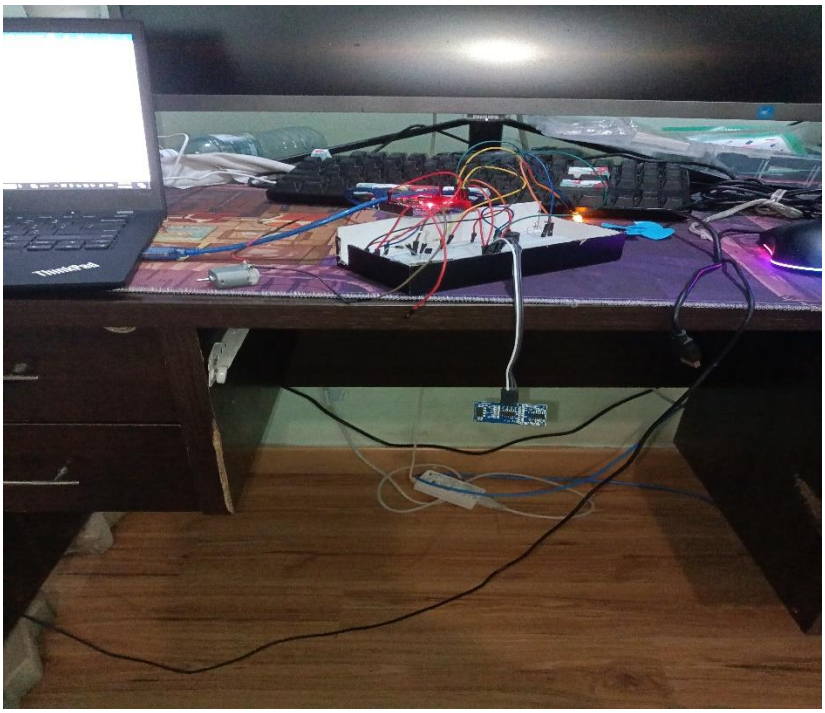
Due to budget a free diagram tool Circuit.io was used. The N Channel Mosfet component was not used in the actual and some of the arrangement is different, but overall the main idea is still present.



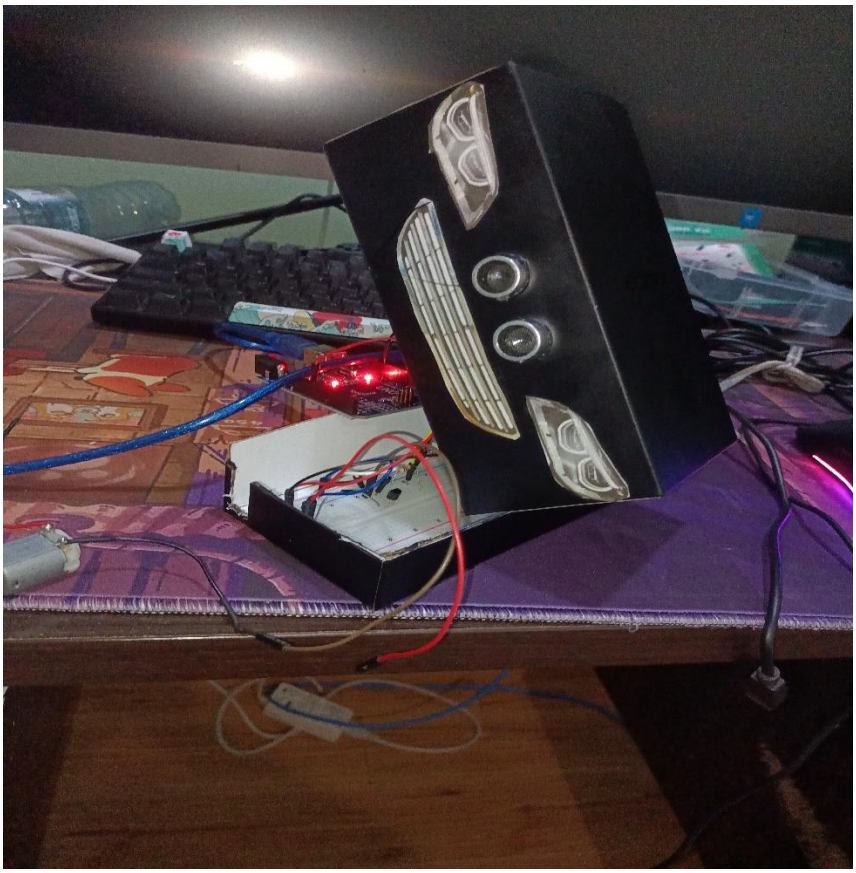
As we can see from the diagram above, the main actuators and sensors will be the ultrasonic sensor, dc motor and buzzer. There are 2 yellow led I have added to simulate the headlight of the car in my model which will be shown later. The ultrasonic sensor will be placed at the front of the model which will represent the front part of the car, as there is where we want to sense our distance from. The echo and trigger pins in our actual project will be connected to pin 2 and pin 3 respectively. The trigger pin will send out the soundwave which will be receive back through the echo pin. Distance is measured this way by calculating how long it took for the soundwave to comeback. The dc motor makes use of a transistor, a 1N4007 diode and a resistor in order for it to work. The motor is connected to pin 4 in the actual project. The fan connected to the motor will be visible on the outside of the model to show that it is properly running. The remaining parts are the buzzer and 2 yellow led, which are connected to pin 5,6 and 7.

Implementation

This was the first initial design for the project. As we can see the main actuators and sensors are present and testing is being done.



Here I started building the model and trying to fit the breadboard and Arduino into the model. Extensions were made to the initial design such as female-to-male jumper wires to let the ultrasonic sensor be able to sit on the front of the model, as well as additions of yellow led to simulate headlight for the front of the model.



Here we are starting to finalise the model. The ultrasonic sensor will sit on the front of the model while the rest of the Arduino and breadboard will be inside the model. There are 2 small slits on the left of the model one for the Arduino board connection to the computer and another for the dc motor fan later on.

Finalized model for the car braking system. Do note the sensor is a bit fiddley so the data receive from it isn’t the most accurate. The motor is placed on the back and the buzzer inside the model.

Resources

Tutorialspoint.com. 2022. *Arduino - DC Motor*. [online] Available at: <https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm> [Accessed 4 June 2022].

Jabbaar, A., 2022. *Ultrasonic Sensor HC-SR04 with Arduino Tutorial*. [online] Arduino Project Hub. Available at: <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6> [Accessed 4 June 2022].

Tutorialspoint.com. 2022. *Arduino - Ultrasonic Sensor*. [online] Available at: <https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm> [Accessed 4 June 2022].

Appendix for Code

Arduino

```
const int pingPin = 3; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 2; // Echo Pin of Ultrasonic Sensor
const int yellow1= 6;
const int yellow2=7;
int motorPin = 4;
int buzzer=5;
long duration, inches, cm;
bool fan=true;

void setup() {
  Serial.begin(9600); // Starting Serial Terminal
  pinMode(motorPin,OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(pingPin, OUTPUT);
  pinMode(echoPin, INPUT);
 pinMode(yellow1,OUTPUT);
  pinMode(yellow2, OUTPUT);
}

void loop() {

digitalWrite(yellow1,HIGH);
digitalWrite(yellow2,HIGH);

   char state = Serial.read();
if (state == 'a')
```

```
    {
      fan=true;
    }
if (state == 'b')
    {
      fan=false;
    }
if (fan==true){
  digitalWrite(motorPin, HIGH);
   noTone(buzzer);
      delay(1);
  }

  if(fan==false) {
    digitalWrite(motorPin, LOW);

    tone(buzzer,349);
      delay(1);
    }
   digitalWrite(pingPin, LOW);
   delayMicroseconds(2);
   digitalWrite(pingPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(pingPin, LOW);

   duration = pulseIn(echoPin, HIGH);
   cm = microsecondsToCentimeters(duration);
   Serial.println(cm);
```

```
  delay(250);


}



long microsecondsToCentimeters(long microseconds) {
  return microseconds / 29 / 2;
}
```

Raspberry Pi

```
import serial
import pymysql
import time
import functools

serialPort=serial.Serial(port='/dev/ttyS0',baudrate=9600,timeout=.1)

while True:
        data=serialPort.readline()[:-2]
        if data.isdigit():
                dbConn=pymysql.connect("localhost","pi","","ultrasonic_db")
                with dbConn:
                        cursor=dbConn.cursor()
                        cursor.execute("SELECT distanceCM from distanceLog order by tempId
DESC LIMIT 1")
                        dbvalue=cursor.fetchone()
```

```
                    if int(data)<dbvalue[0]-2 or int(data)>dbvalue[0]+2:
                        cursor.execute("INSERT INTO distanceLog (distanceCM)
VALUES (%s)",(data))
                        dbConn.commit()
                        cursor.execute("SELECT distanceCM from distanceLog order by
tempId DESC LIMIT 1")
                        dbvalue=cursor.fetchone()
                        print(dbvalue[0])
                        if dbvalue[0]<5:
            serialPort.write(b"b\n")
        if dbvalue[0]>20:
            serialPort.write(b"a\n")
```