Final Project Report

By Aditya Singla (asingla4) and Siddhant Singh (ss59)

Introduction

1. **Problem Description:** We got the idea of our final project from Lab 9 where we worked on making our car 'autonomous' by incorporating snap action switches. In order for the snap action switches to work, there need to be contact between the barrier and the car, which we felt defeated the purpose of an autonomous car. There shouldn't be contact between the car and an obstacle for the car to adjust its path. In order to fix this problem, we set out to make an 'autonomous' car which can change its path by sensing the obstacle around it and changing its course accordingly.

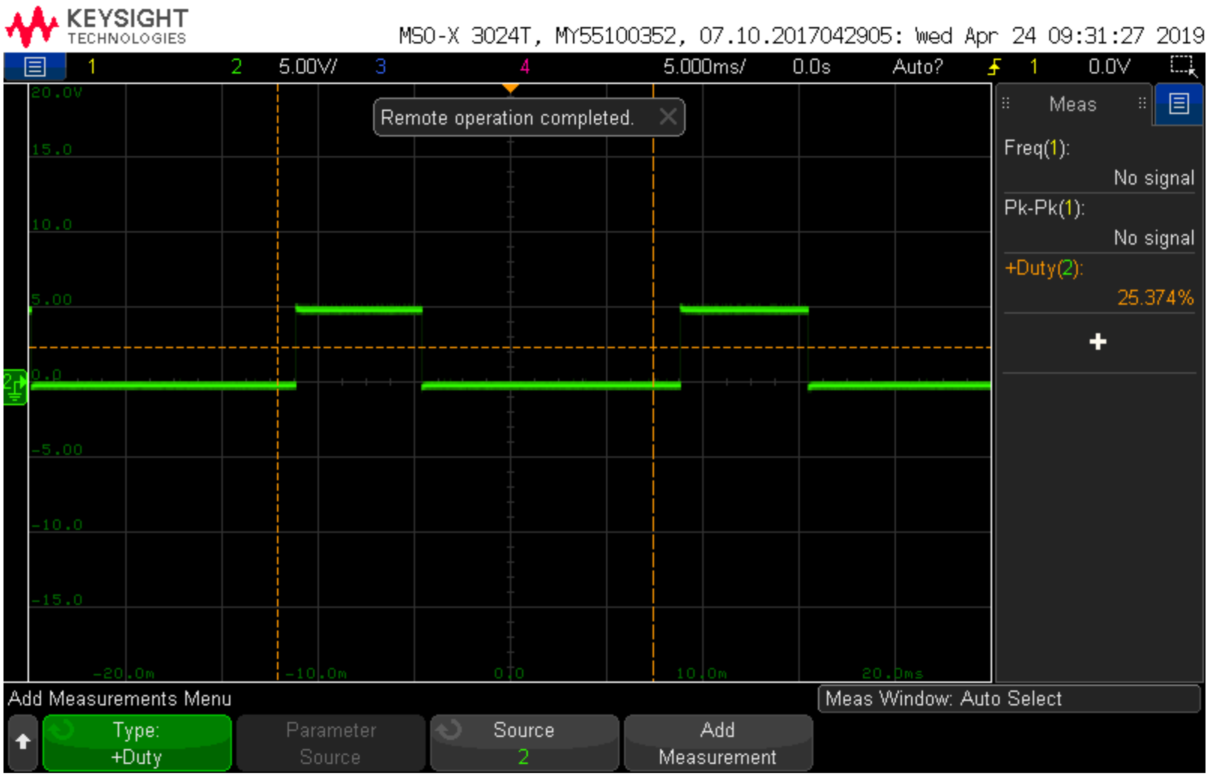
2. **Design Concept:** Our design used the chassis of the car that we used throughout the ECE110 lab. However, we decided against using the existing circuits and snap actions switches and incorporated a new circuit. We decided that ultrasonic sensors were the best route forward to detect an obstacle and change the path of the car in order to avoid any sort of collision. When the ultrasonic sensor detected an obstacle (when the obstacle was a pre-set distance away), we used the H-bridge to reverse the polarity of the motors so it can change its path. Moreover, in order to make it clear to the spectator which mode the car is in currently (even though it is apparent just by looking at the motion of the car), we added LED cues. A solid green LED is on when the car is under regular motion, that is it is moving forward in a straight line. However, when the car is turning, there is a blinking red LED. The changes that were made in the motion of the car was through an Arduino which was powered by a set of batteries in order to provide enough voltage for the motors to run well. What we have created is a prototype of our vision. Due to the high complexity in the coding aspect of the project, we decided to show what our vision is by implementing only one ultrasonic sensor and focusing more on the hardware aspect of the project.

Analysis of Components

1. Characterization of each sensor

Our circuit uses an ultrasonic sensor to sense any obstacles in the car's way. As we initially set out to make a completely analog circuit (without the use of an Arduino), we believed that it was necessary for us to fully understand how the ultrasonic sensor worked. Using the knowledge gained in module 7D, we used a waveform generator to generate a waveform with f = 50Hz, A = 5V, and duty cycle = 50%.

We connected the ultrasonic sensor to the waveform generator by connecting the trigger (which was the input) to the waveform generator. And then set up the oscilloscope across the echo signal (which was the output) and ground to obtain the following waveforms:



KEYSIGHT TECHNOLOGIES MSO-X 3024T, MY55100352, 07.10.2017042905: Wed Apr 24 09:31:27 2019

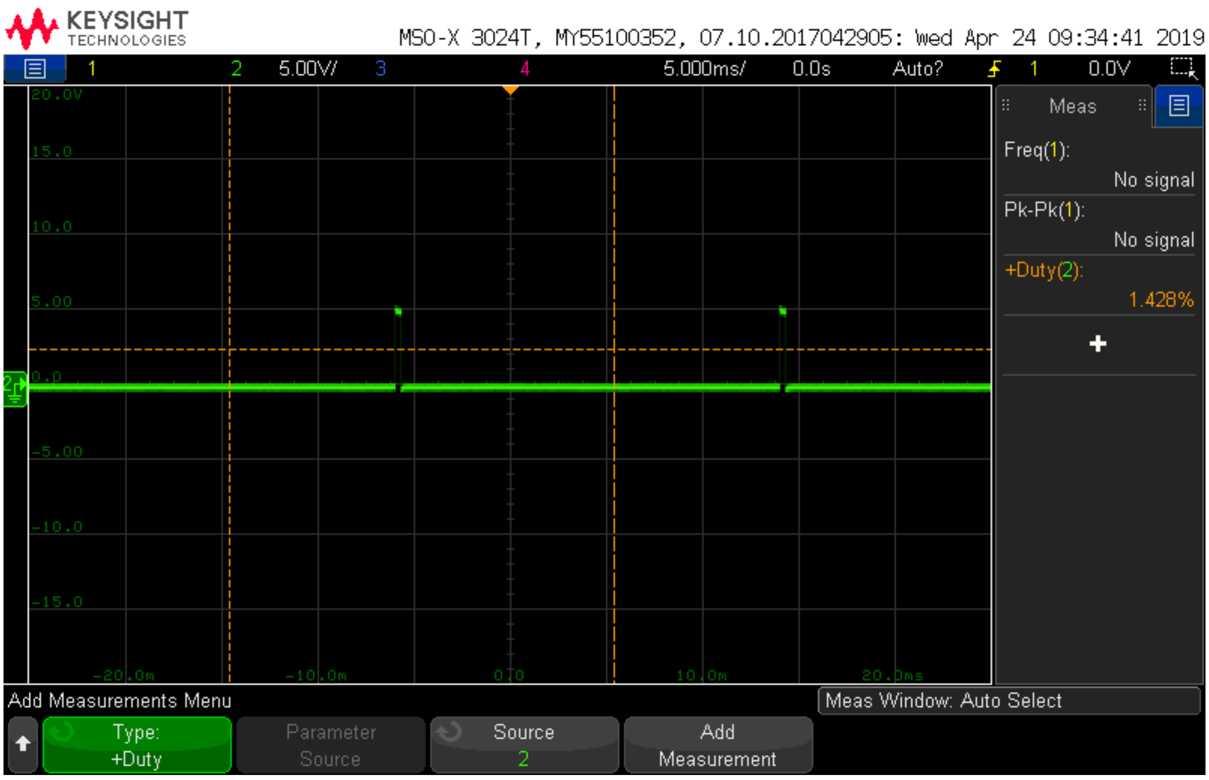
Remote operation completed.

Freq(1): No signal

Pk-Pk(1): No signal

+Duty(2): 25.374%

When the object is placed at a large distance, resulting in duty cycle: 25.374%



When the object is placed near the sensor, resulting in an almost zero duty cycle.

As the object is placed closer to the sensor, the duty cycle of the output decreases. We believed that we could use this fact to get the car to come to a stop (by gradually decreasing its speed as it approaches an obstacle), without using an Arduino.

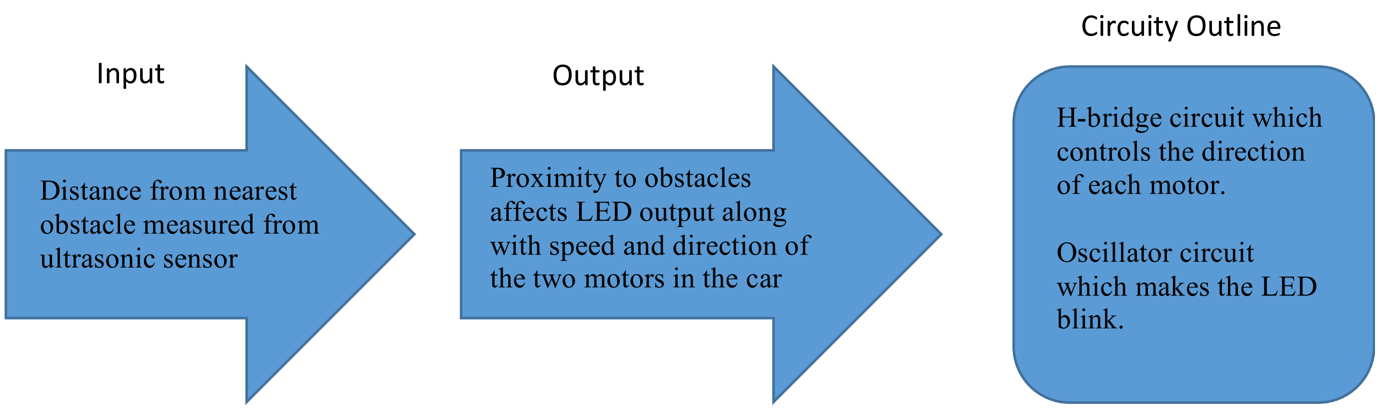
2. **Design Considerations:** Since our project working revolved mainly around the ultrasonic sensor, we had to make sure that the sensor will calibrated properly and the set distance was neither too much, nor too little so that the project could work according to plan. Due to the field of view of the ultrasonic sensor, the location where we place the ultrasonic sensor was very sensitive. At first we tried mounting the ultrasonic sensor on top of the chassis of the car, but that meant that not obstacles could be detected properly by the sensor and the car was not moving as planned. Hence, we decided to mount the sensor at the base of the chassis, which meant that the functioning of the car was improved drastically.

   We also had to include a set of batteries to power the motors, and in order to reduce the center of gravity of the car, we decided that the best place to mount the batteries was at the base of the car and since the regular battery set could not fit into the restricted space between the motors, we opted for a smaller battery set and lowered the center of gravity of the car so the car could function properly.

   Moreover, in order to avoid any jerky motion of the car while changing the path, we set the speed of the motors very carefully.

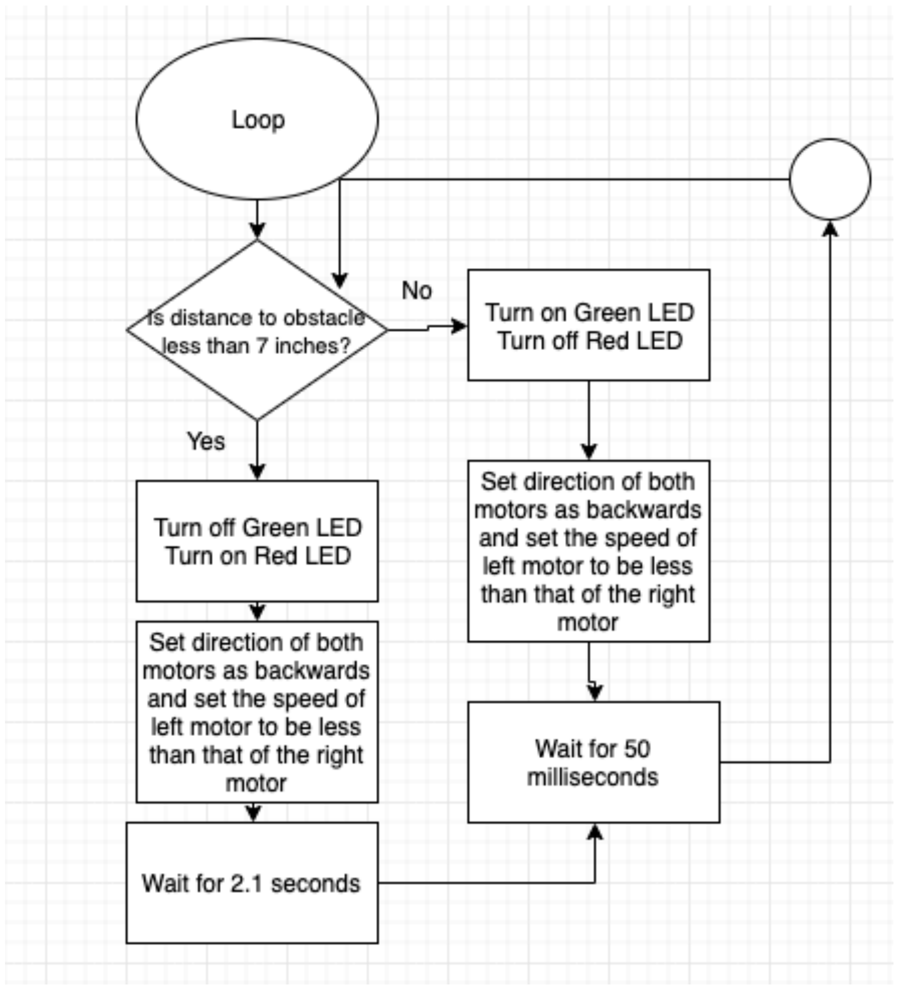
Design Description

1. Block Diagram


Input
Distance from nearest obstacle measured from ultrasonic sensor
Output
Proximity to obstacles affects LED output along with speed and direction of the two motors in the car
Circuity Outline
H-bridge circuit which controls the direction of each motor.
Oscillator circuit which makes the LED blink.

2. Arduino

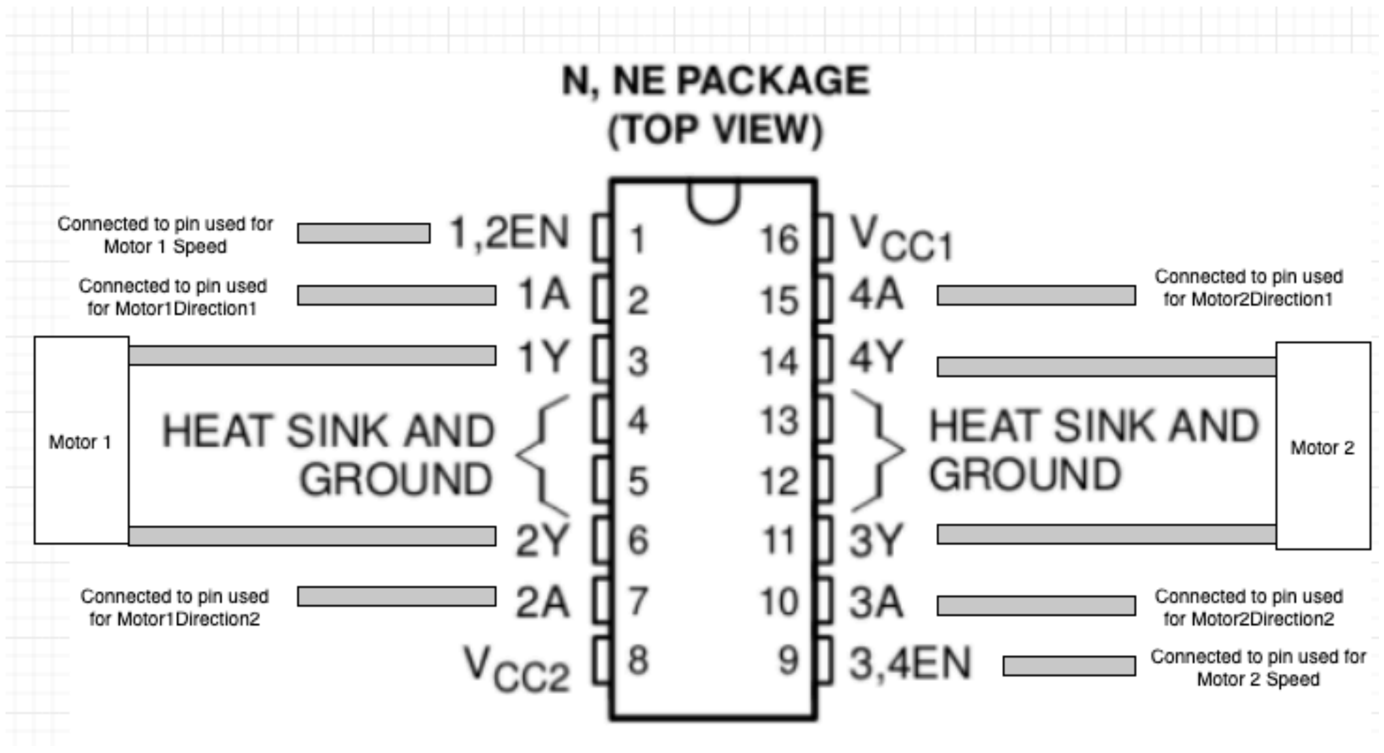
The following flowchart represents the code we uploaded onto our Arduino


Loop
Is distance to obstacle less than 7 inches?
No
Yes
Turn on Green LED Turn off Red LED
Set direction of both motors as backwards and set the speed of left motor to be less than that of the right motor
Wait for 50 milliseconds
Turn off Green LED Turn on Red LED
Set direction of both motors as backwards and set the speed of left motor to be less than that of the right motor
Wait for 2.1 seconds

The actual code we used is also included for reference:

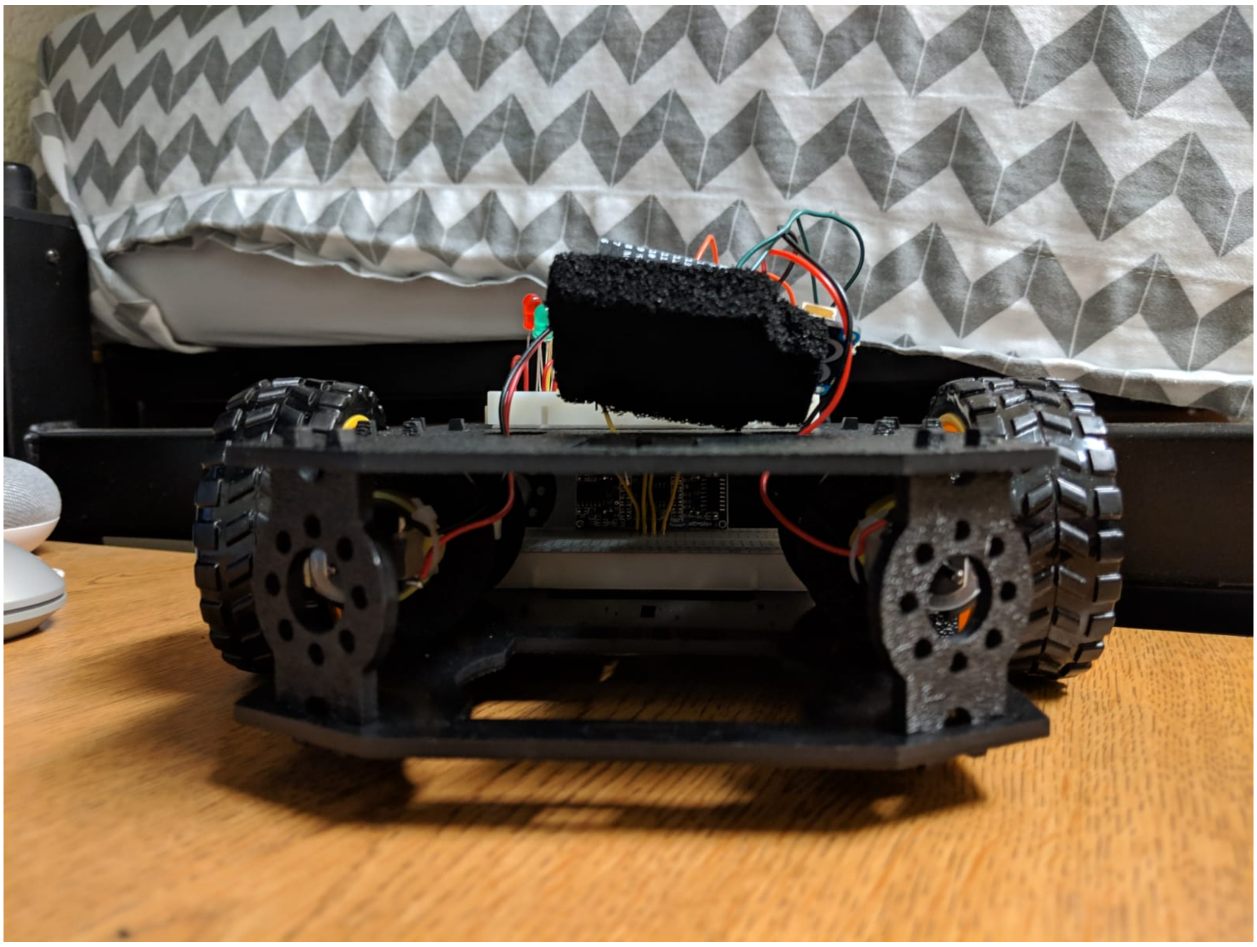
```
const int trigPin = 9;        //connects to the trigger pin on the distance sensor

const int echoPin = 10;    //connects to the echo pin on the distance sensor

const int redPin = 13;       //pin to control the red LED inside the RGB LED
const int greenPin = 11;    //pin to control the green LED inside the RGB LED
float distance = 0;            //stores the distance measured by the distance sensor
const int Motor1Direction1=7;
const int Motor1Direction2=8;
const int Motor1Speed=6;
const int Motor2Direction1=3;
const int Motor2Direction2=4;
const int Motor2Speed=5;

void setup() {
  //Ultrasonic code starts
  Serial.begin (9600);        //set up a serial connection with the computer
  pinMode(trigPin, OUTPUT);   //the trigger pin will output pulses of electricity

  pinMode(echoPin, INPUT);    //the echo pin will measure the duration of pulses coming back from the distance sensor
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);

  //Motor code starts
  pinMode(Motor1Direction1, OUTPUT);
  pinMode(Motor1Direction2, OUTPUT);
  pinMode(Motor1Speed, OUTPUT);
  pinMode(Motor2Direction1, OUTPUT);
  pinMode(Motor2Direction2, OUTPUT);
  pinMode(Motor2Speed, OUTPUT);
}

void loop() {
  distance = getDistance();
  Serial.print(distance);  //print the distance that was measured
  Serial.println(" in");     //print units after the distance
  if(distance <= 7){                           //if the object is close
    //make the RGB LED red
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);

    digitalWrite(Motor1Direction1, HIGH);
    digitalWrite(Motor1Direction2, LOW);
    analogWrite(Motor1Speed,130);

    digitalWrite(Motor2Direction1, HIGH);
    digitalWrite(Motor2Direction2, LOW);
    analogWrite(Motor2Speed,180);
    delay(2100);
  }
  else{           //if the object is far away
    //make the RGB LED green
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    digitalWrite(Motor1Direction1, LOW);
    digitalWrite(Motor1Direction2, HIGH);
    analogWrite(Motor1Speed,176);

    digitalWrite(Motor2Direction1, LOW);
    digitalWrite(Motor2Direction2, HIGH);
    analogWrite(Motor2Speed,170);
  }
  delay(50);       //delay 50ms between each reading
}
float getDistance()
{
  float echoTime;
       //variable to store the time it takes for a ping to bounce off an object
  float calculatedDistance;         //variable to store the distance
  //send out an ultrasonic pulse that's 10ms long
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  echoTime = pulseIn(echoPin, HIGH);      //use the pulsein command to see how long
 it takes for the
                                          //pulse to bounce back to the sensor
  calculatedDistance = echoTime / 148.0;
 //calculate the distance of the object that reflected the pulse (half the bounce
 //time multiplied by the speed of sound)

  return calculatedDistance;           //send back the distance that was calculated

}
```

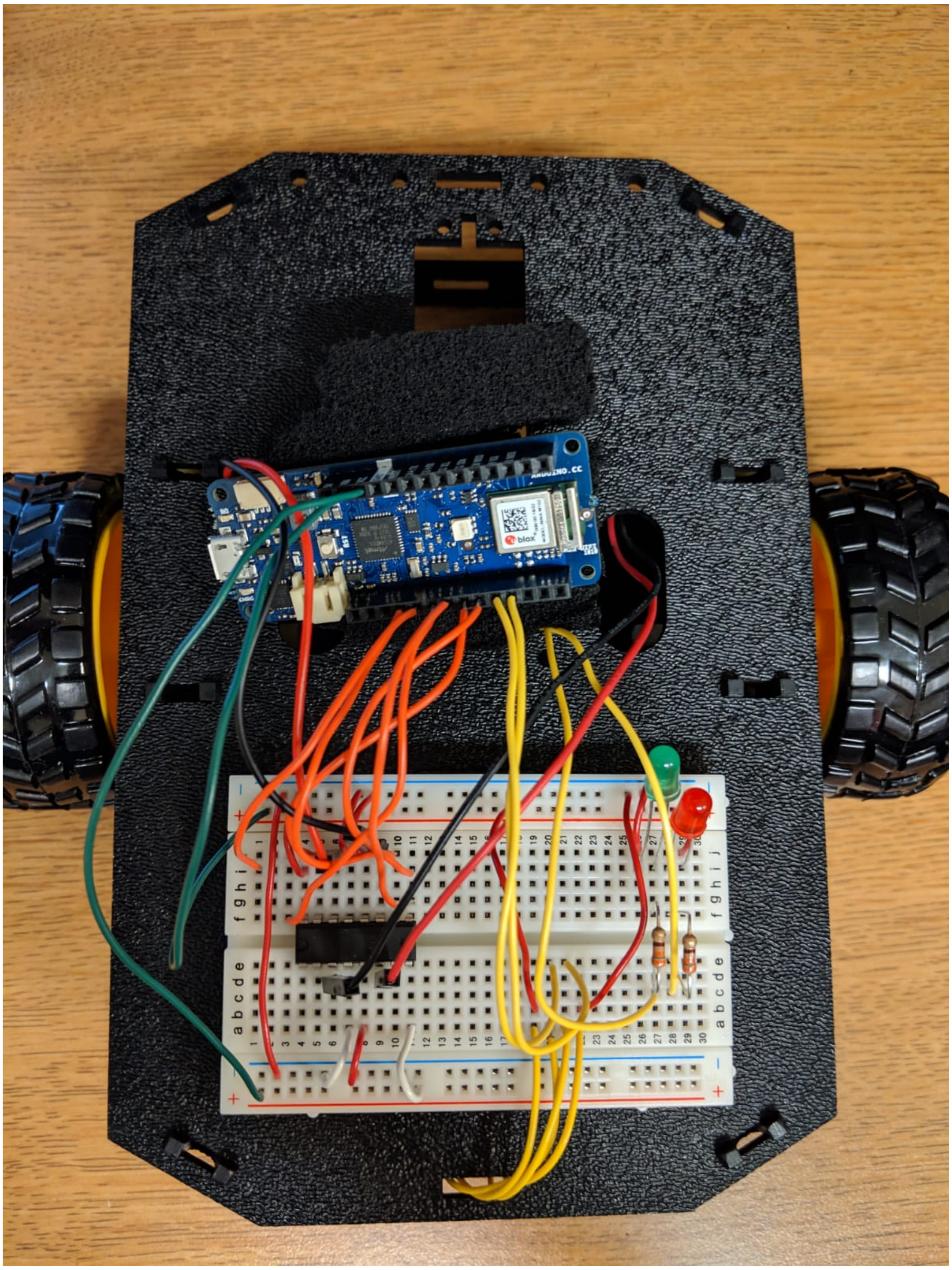
3. Circuit Schematic:

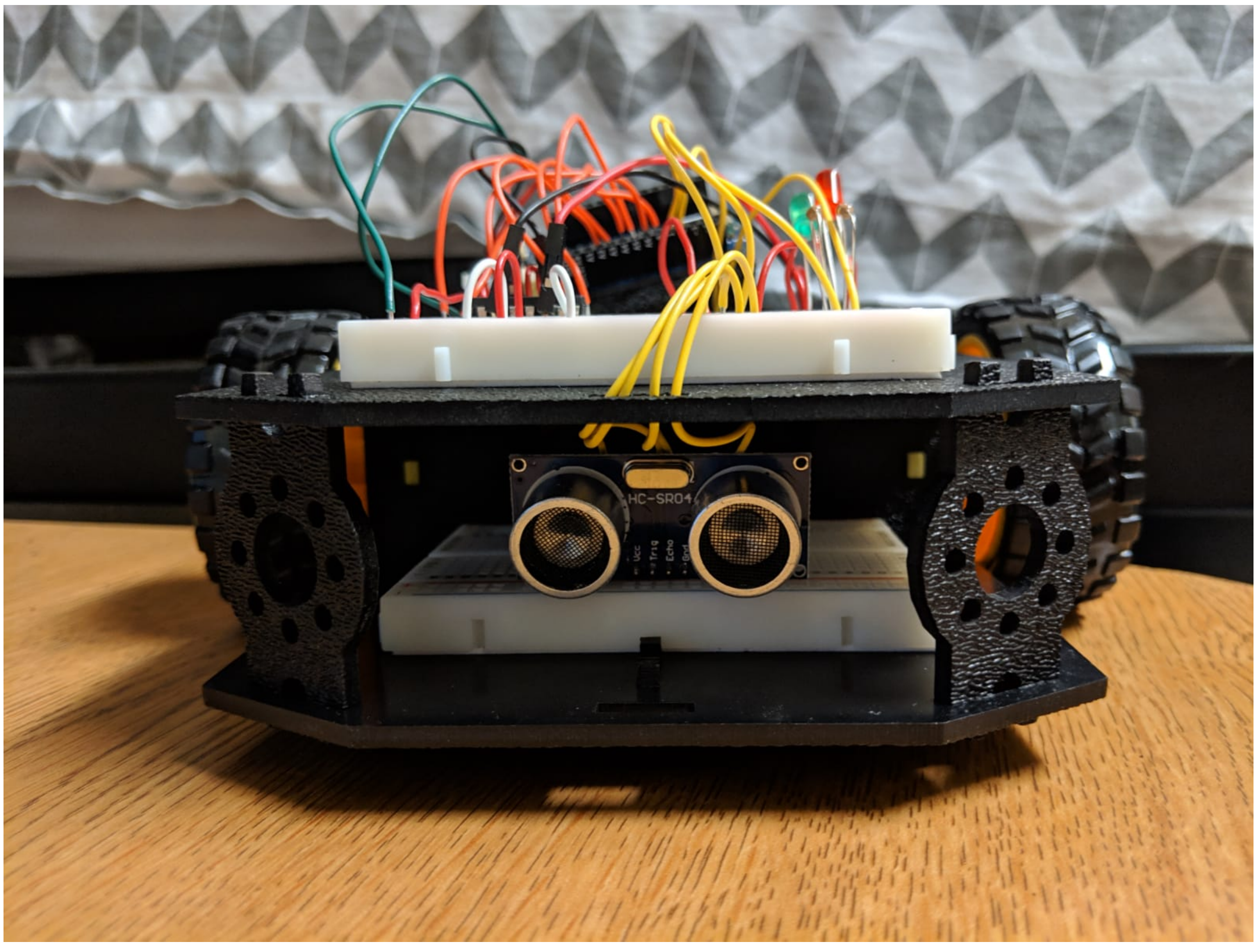


N, NE PACKAGE
(TOP VIEW)
Connected to pin used for Motor 1 Speed
Connected to pin used for Motor1Direction1
Motor 1
HEAT SINK AND GROUND
Connected to pin used for Motor1Direction2
1,2EN
1A
1Y
2Y
2A
VCC2
1
2
3
4
5
6
7
8
16
15
14
13
12
11
10
9
VCC1
4A
4Y
3Y
3A
3,4EN
Connected to pin used for Motor2Direction1
HEAT SINK AND GROUND
Motor 2
Connected to pin used for Motor2Direction2
Connected to pin used for Motor 2 Speed

4.Physical and Mechanical Construction







The Ultrasonic sensor and the H-bridge are connected to the Arduino which is powered by the battery which is not present in this picture right now. The ultrasonic sensor is placed at the bottom of the chassis of the car in order to make sure that the sensor can pick up all the objects.

Conclusion

1. **Lessons learned:** We first decided against using an Arduino and operate the ultrasonic sensors and the H-bridge via analog signals. We decided to use a clock and create multiple states for the working of the ultrasonic sensor and the H-bridge by using the knowledge that we gained in ECE120. However, this method did not yield the necessary results and hence we decided to use a microcontroller (an Arduino) and use digital signals instead. What we learned was that using digital signals instead of analog signals is much easier and the flexibility provided by using a microcontroller is very useful. However, in making of the project, we enjoyed understanding the applications of H-bridge in changing the polarities of the motor and hence, changing the path of the car.
2. **Self-Assessment**: We felt that we did a good job and did justice to our concept. Our project worked well and achieved all the targets that we had set as a group. However, we felt like if we had calibrated more ultrasonic sensors, we could have presented a project that would virtually never crash into an obstacle. Overall, our project satisfied all our goals and was a good testament to our vision.