Group 4: Robo Pig

Evan Wetzel, Mia Fox, Dean Allison, Corbin Hoppe, Andres Mont

GEEN 1400: Professor Mah

Final Report

As a whole, our project, the robo pig, was a great learning experience and an overall success. At the beginning of our project, our group had very ambitious ideas as to what our project would be, and how it would function. However, as a group with little to no hands on engineering experience, we had to adjust our project vision greatly to accommodate our groups skills. Throughout the semester, our group learned that in the design process, there are many trade offs that must be made. We constantly ran into issues where we were faced with sacrificing certain aspects of our design. Whether it was moving the ultrasonic sensor and changing the aesthetics of our car, or decreasing the speed of our car for better reaction time, there were many compromises made. This experience taught us a great deal about the design process as a whole, improving all of our engineering skills along the way.

Our clients are preschoolers to young elementary school children. The problem that we chose to address was the need for increased physical activity in children of this age group, which we hoped to accomplish by creating an interactive toy car that avoided obstacles and the child. The child will play with the toy by chasing the car as it avoids objects, and ideally avoids the child as well.

We had very few requirements for this projects when we began. The class requirements were that the project was to be completed within twelve weeks, could not cost more them $375 ($75 per team member) and must not directly copy an existing toy or infringe on current toy patents. The only requirement specific to our project,versus other GEEN 1400 projects, was that the toy must encourage motor skills and running ability. As we did more research on our project we found many inherent requirements based on safety for the children and the goal of our project. The ideal user for our toy is a child between the ages of three and five, children who are working to hone their walking and maneuvering skills. As the engineers of this toy, it was our responsibility to minimize the safety hazards for our user, which became one of our main requirements. This included not having any accessible wires, no sharp points that the child could injure themselves on, no small part they could easily choke on, and since our project is moving, an ability to limit movements that could lead to injury, like falling down stairs. Lastly, we chose to make a toy that was aesthetically pleasing, so children are attracted to it and actually want to play with it.

In the physical design of our toy, there were three things that proved to be the most important: the sensors, the programming and the motors. Conveniently, these things are all connected so if one of them was to fail, it would lead to failure of our entire project. The sensors are what determine is the car sees an object, which tells the program how the car needs to move. If the sensors are not properly programmed, they will not sense things quick enough and will fail to obtain accurate readings, leading to collisions with walls and people. If the motors are not programmed in a way that is correctly affected by the sensors, they would move in ways that would neglect the objective of our entire project, effectively ruining its functionality.

When analyzing the ethics behind our design, we saw only a few potential issues. As addressed earlier, we had to make sure we were creating a safe car. That is why we worked to make the exterior of the car something that would cover anything sharp, or any potential choking hazards. We also chose to make our car specifically focused on being used indoors to prevent potential issues it could encounter outdoors, such as endangering a child in the street. We also considered the environmental effects of our car. Though we were hoping to a have a car made of materials that were more considerate to the environment, the use of the car indoors will lessen the likelihood that children will leave it outside to decay.

Before settling on our final idea, we had four preliminary ideas for what our overall project could be. One idea was a “Velocity Ball”. Like any other ball used to play catch, it would be a fun outdoor activity, but this ball would measure and report the velocity at which it was thrown, either through an app or directly on the ball. It could be used for both learning purposes, in learning about velocity and how it changes through various trajectories, or as a way for young athletes to train, especially baseball or football players. Similar to the velocity ball, we thought of designing an automatic throwing machine. It would be a device that shoots a ball at different speeds and angles chosen by the child. This game would test an athlete’s reflexes and help them train for specific instances. The speeds could be adjusted for any age group as a way to train for sports such as baseball or football. Our third design idea was a remote controlled adventure buddy car. This would be a remote control car that the child controls. It would show various measurements while driving such as velocity, acceleration, and elevation measurements. Similar to our final design, this toy would encourage kids to explore outside and also develop an understanding of extremely basic physics concepts. The project would be geared towards younger children and designed to handle various terrain.

Our final idea, and the one we settled on was a runaway car. Our plan was to create a small toy car that would be programmed to drive away from children, encouraging them to chase it. The car would have different “levels” that would correlate to different speeds and would ideally get children to run around and exercise. In our original plan, it was a car with 7 ideal final features. First it would have a button or sensor to start the car and determine when the car is “caught”. We also wanted an interface that changes the speed/difficulty of the car and a screen that displays high score and the time it takes for the car to be caught. Of course, we wanted an outside cover making the toy car aesthetically pleasing and child friendly, and we always planned to incorporate powerful motors to make the car faster. We also hoped to incorporate large tires to allow the car to maneuver outdoors. The main feature of the car was the use of both ultrasonic and infrared sensors to locate the child and its surroundings.

Through careful consideration and overall testing of both the sensors and the motors, we came to the conclusion that our car should be used strictly indoors. With the range of our sensors and our ethical requirements as limiting factors, decreasing the scope of our cars ability was important. Doing this allowed the motors to perform at quicker speeds. It also removed many risk factors involving safety, as a car that runs away from you has the potential to run into a street or off of something tall. Operating indoors also helped us utilize the car’s sensors, as there are more obstacles, and therefore a greater challenge, as the car would have more directional changes. Since the directional changes work agility, having a car with more sporadic movement would increase the ability of our car to increase a child's physical activity.

Our final project is a robotic car that uses ultrasonic and infrared sensors to detect objects in front of the car and avoid them. The car moves quickly and is designed for a child to chase after it with the objective to catch the car as it avoid objects around a room. As you can see here, the car is grey rather than pink and it's ears and tail are made from pipe cleaners. Ideally they would be made of another plastic, or something sturdier than simply pipe cleaners. Inserted below Figure 1 is Figure 2, showing the internal structure of the car. Figures 5 through 7 show more extensive views of the car as a whole.

Figure 1: This is our pig in it's most final form. It has a shell and encased in the shell is all the hardware.

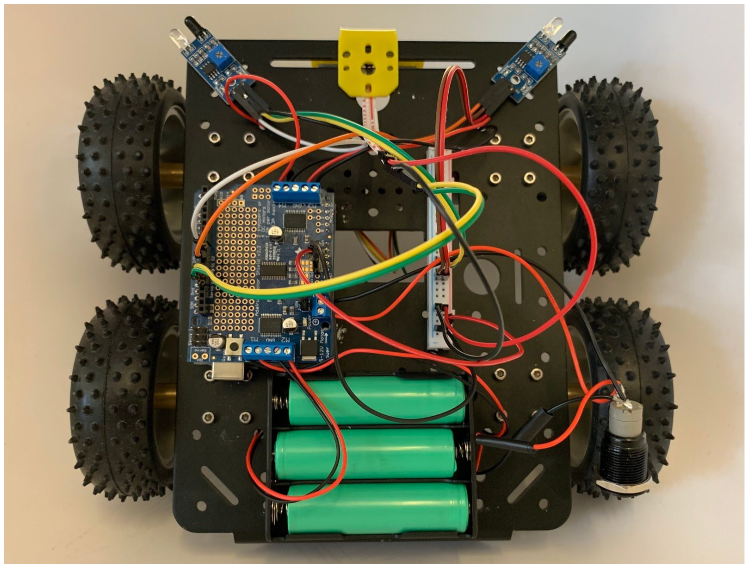

Figure 2: This is a view of the internal structure of our car from above. Here, you can see the two infrared sensors in front, the arduino uno board and the rechargeable batteries that power most of the sensors and motor. Missing from the above photo is our 9V battery that powers the arduino board.

In creating the car, there were three main aspects, the internal structure, the code, and the external shell. There was much more work put into the internal structure, which houses the actual functioning processes of the car. To save time, while making sure that the base of our robot is sturdy, our group decided to buy a chassis, which we placed all of our electronics onto. From there, we had to figure out how to power the arduino board, sensors and motors. We chose rechargeable batteries for the motors so the person using the car doesn’t have to constantly buy new batteries, as the motors take a lot of power and need fully charged batteries. We powered the arduino and sensors with a 9V battery, which is sufficient, as the sensors do not require high amounts of voltage. Through extensive trial and error we were able to write the code of our project. Figure 8, in the appendix, gives a brief overview of our code, and the actual code of our project can be seen in Figure 11. The outer shell of our pig was made through vacuum forming ABS plastic sheets over a clay model of what we wanted the pig to look like.

The majority of our project was experimentally designed. We had three main aspects that we needed to test in order to find the best options. Originally our motors could only go 200RPM. We determined through trial and error that those motors could not go as fast as we would need them to go, considering that our user will be able to move fast. Because of this we upgraded to RPM motors. Once we determined which motors to use we found that we needed a motor shield so that we could run our motors through our power source, which was more voltage than a normal arduino board could handle. Our motor shield can only run 12V so through trial and error we were able to settle on three 3.7V batteries that would make our car run as fast a possible, on a total of 11.1V. Initially, we planned to use whiskers along with our ultrasonic sensors. However, the whiskers would have been a safety hazard for the children playing with the car, considering the sharp points. We then decided to use multiple ultrasonic sensors so that we would have more coverage. Having multiple ultrasonic sensors caused the sensors to interfere with each other, creating false signals. After determining that multiple Ultrasonic sensors wouldn’t work, we resorted to using one ultrasonic sensor along with two infrared sensors on either side to get the most rounded sensor coverage without false signals or health hazards.

Ultrasonic sensors send out sound waves to be reflected back by objects. The sensor can tell the distance from the object by measuring the total time and using the speed of sound to calculate distance. With this equation incorporated into our coding, we are able to tell, to the exact centimeter, how far away an object is in front of the sensor. Then, with some “if else” statements, we are able to tell the car to turn away from the object “if” the distance is less than 15 cm. However, the field of view of the ultrasonic sensors are very one-directional and somewhat narrow, so we encountered the problem of not being able to detect objects to the side of the car. To fix this, we decided to use two IR obstacle avoidance sensors on the corners of the car to detect objects along with the ultrasonic sensor. This improved the accuracy of our readings substantially and the car is now able to effectively avoid objects.

If we were able to acquire more time and money for this project, we would definitely improve the power output and motor capability in order to make the car go faster to run away from children. We would also improve the sensor capability. Ideally the car would sense things farther in front of it and it would more accurately detect when it is about to hit something, as many times it ran into something and then sensed it afterwards. We would also hope to add additional sensors to base the motor control off of where the kid was coming from. We would also use a pink sheet of ABS plastic so the pig was pink, as that is more aesthetically pleasing than a grey sheet. As stated earlier, we would make the ears and tail of something more stable than the pipe cleaners. We also would ensure all the sensors were covered by the casing, as right now the infrared sensors are exposed. Other improvements would be to make a better positioned on and off switch, and an easier way to change and recharge the batteries.

Our final project is a robotic car that uses ultrasonic and infrared sensors to detect and avoid objects in front of it. The electronics of the car are concealed by grey ABS plastic formed in the shape of a pig. We met each of our functional and design requirements. The total cost of our project is $154.12 which is far under our requirement of less then $375. We also successfully made a toy that avoids object in front of it and can be used to encourage physical activity in kids. Our ethical design plan revolved around having a safe toy for children. We met many aspects of this design plan, however we struggled to put all of our electronics inside of the casing of the pig because our sensors wouldn’t work if they were inside the case. Despite this, we completed all of the other ethical requirements by making the toy work inside and making sure that the toy wasn't toxic by choosing not to use paint because it kept chipping.

Our overall team effort was very high. We often met outside of class to work on our project and presentations. Each of our teammates were present in class almost everyday and we were productive in getting our work done because of everyone being there to help. We believe that our team effort over the course of the semester contributed to us having a strong final project, as we didn't have to scramble during the last week because we were on top of our work. Each of our team member did what was required of them so that we had a very well rounded, functioning group.

Appendix

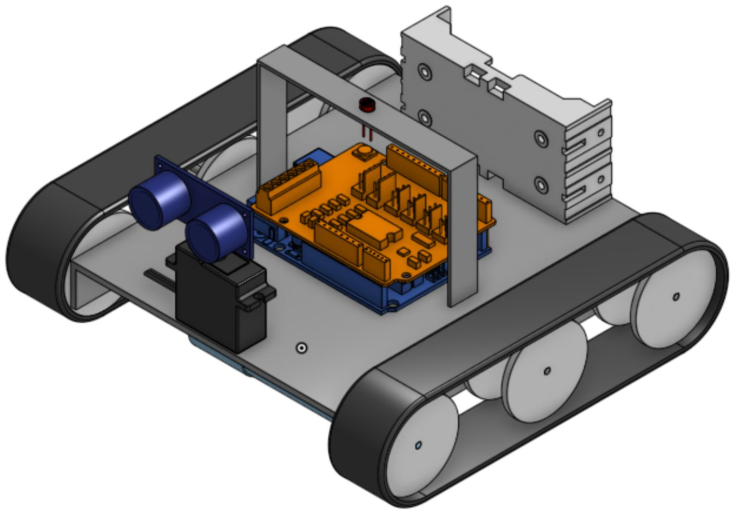

Figure 3: This is our original CAD design, showing the inside of the car, where the hardware would be housed inside the outer shell.

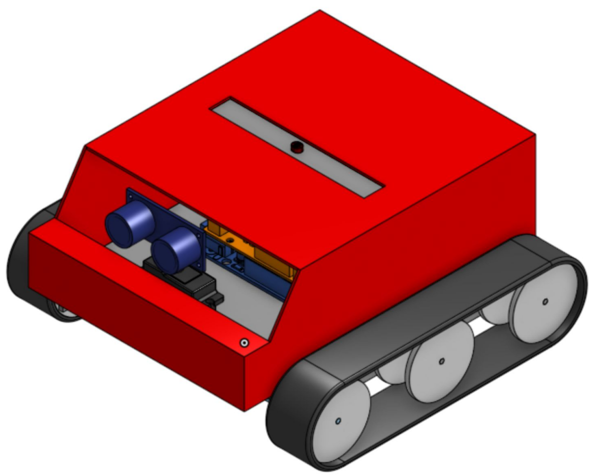

Figure 4: This is our original CAD showing the outside of our car. Since then, we changed our shell to be pig shaped. It was originally supposed to utilize the ultrasonic sensors to be the pig nose of our sensor.

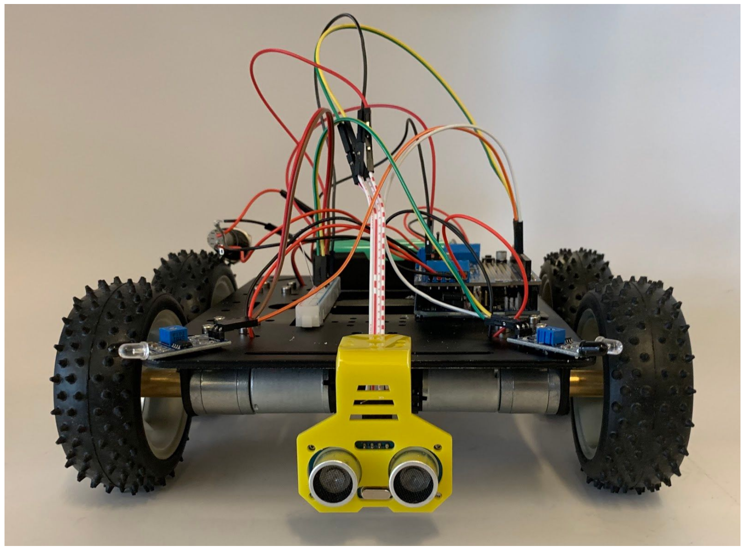

Figure 5: This is very close to our final design of our car, with the wires and sensors on the front.

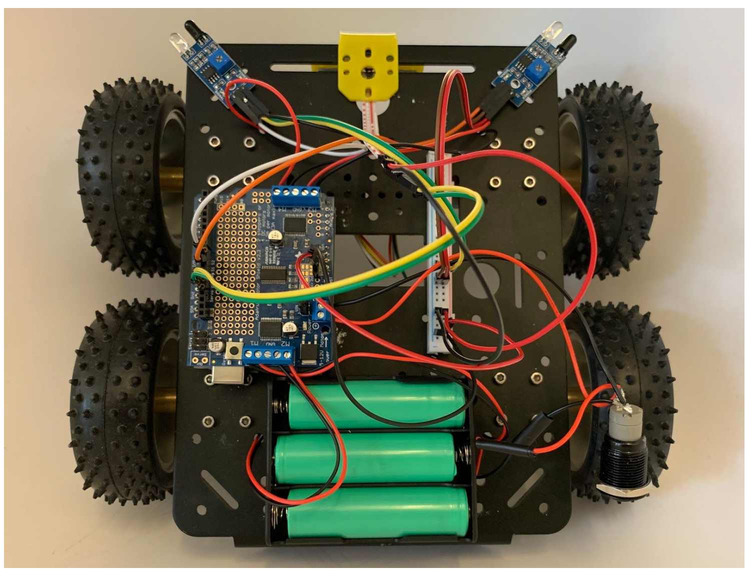

Figure 2: This is a view of the internal structure of our car from above. Here, you can see the two infrared sensors in front, the arduino uno board and the rechargeable batteries that power most of the sensors and motor. Missing from the above photo is our 9V battery that powers the arduino board.

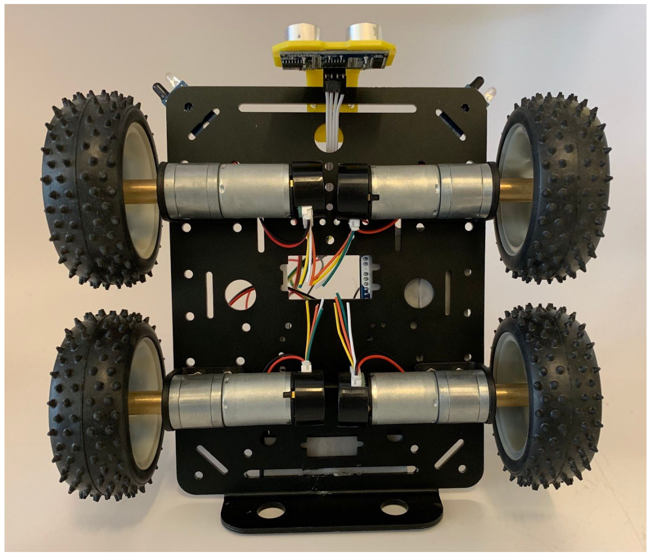

Figure 6: This shows the underside of our car, particularly the 4 motors that power the car.

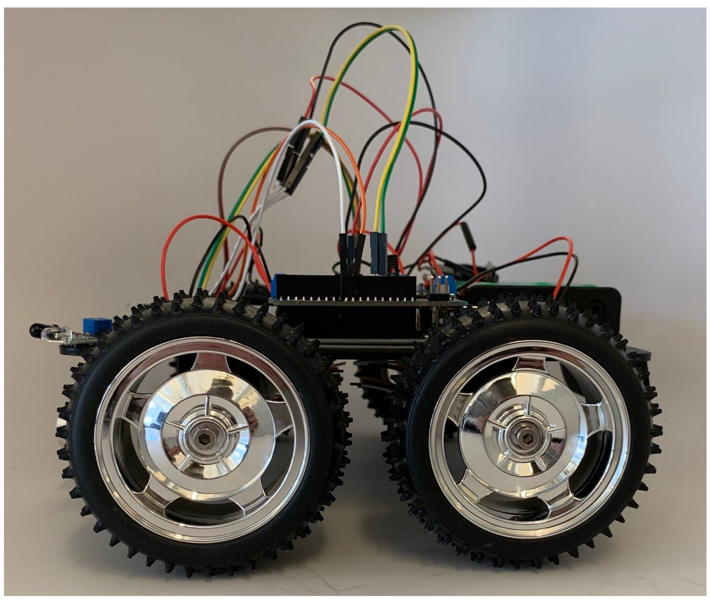

Figure 7: This is a side view of our car so you can see about how big the wheels are in comparison to the car itself. Each wheel is 8cm in diameter.

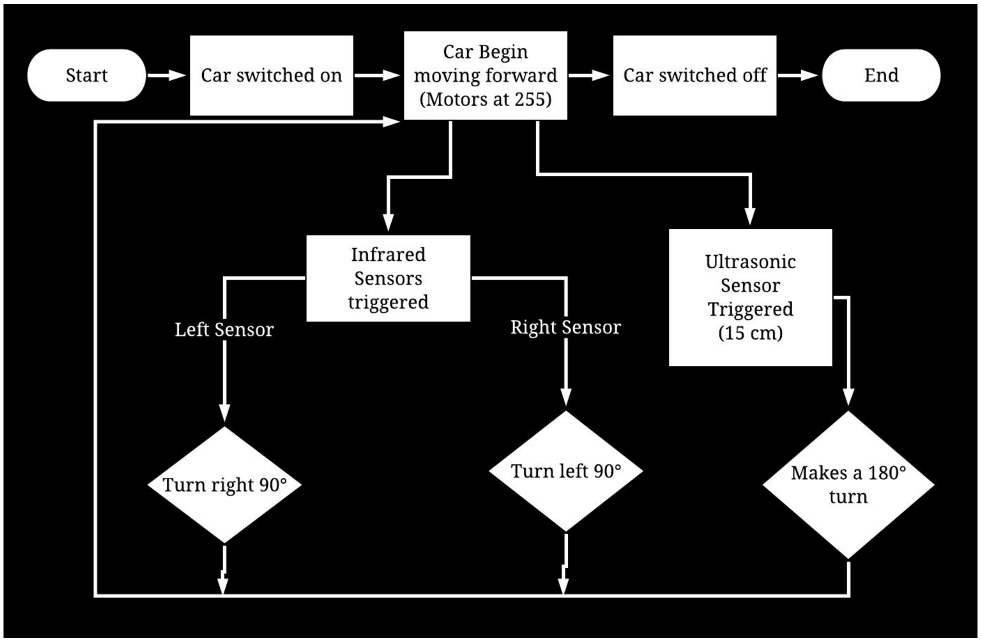

Figure 8: The above is our programming flowchart, showing a very simple, outline of what our code runs through.

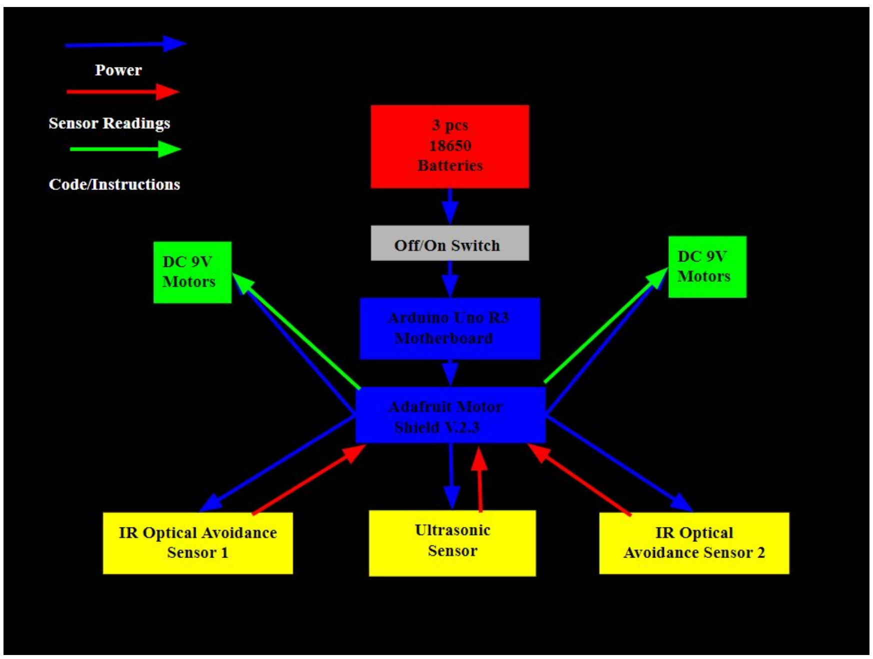

Figure 9: Shown here is the electronic flow diagram, showing what is powered by what and what connects to what in the simplest way.

Materials:	Cost:
Chassis	$70
HC SR04 Ultrasonic Distance Sensor	$1.95
Sensor Holder	$1.09
Infrared Sensors	$2.60
Adafruit Motor Shield v2.3	$19.65
Arduino Uno R3	$16.98
3 18650-25R 20A 2500mAh Rechargeable 3.7V Battery	$21.00
Battery Case	$8.00
Push Button Switch	$8.49
ABS Plastic Sheet	$4.00
Outside Decorations	$0.39

Figure 10: The above is the overall cost of materials for our entire project. Calculated here are just the materials we used in our project.

```
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include
"utility/Adafruit_MS_PWMServoDriver.h"
Adafruit_MotorShield AFMS =
Adafruit_MotorShield();
Adafruit_DCMotor *M1 = AFMS.getMotor(1);
Adafruit_DCMotor *M2 = AFMS.getMotor(2);
Adafruit_DCMotor *M3 = AFMS.getMotor(3);
Adafruit_DCMotor *M4 = AFMS.getMotor(4);

const int trigPin = 12;
const int echoPin = 13;
float duration, distance;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(8, INPUT);
  pinMode(9, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  AFMS.begin();
}
void loop() {
  // put your main code here, to run
repeatedly:
  int Rightbump = digitalRead(9);
  int Leftbump = digitalRead(8);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration*0.034/2;

  Serial.println(distance);
  if(distance <=15){
  backward();
  turn180();
  }
  else if(Rightbump == 0){
  backward();
  turnLeft90();
  }
```

```
void stopCar(){
   M1->setSpeed(0);
   M2->setSpeed(0);
   M3->setSpeed(0);
   M4->setSpeed(0);
   M1->run(FORWARD);
   M2->run(FORWARD);
   M3->run(FORWARD);
   M4->run(FORWARD);
   delay(300);
}
void backward(){
   M1->setSpeed(180);
   M2->setSpeed(255);
   M3->setSpeed(255);
   M4->setSpeed(180);
   M1->run(BACKWARD);
   M2->run(BACKWARD);
   M3->run(BACKWARD);
   M4->run(BACKWARD);
   delay(500);
}
void turnLeft90(){
   M1->setSpeed(180);
   M2->setSpeed(255);
   M3->setSpeed(255);
   M4->setSpeed(180);
   M1->run(FORWARD);
   M2->run(BACKWARD);
   M3->run(FORWARD);
   M4->run(BACKWARD);
   delay(400);
}
void turnRight90(){
   M1->setSpeed(180);
   M2->setSpeed(255);
   M3->setSpeed(255);
   M4->setSpeed(180);
   M1->run(BACKWARD);
   M2->run(FORWARD);
   M3->run(BACKWARD);
   M4->run(FORWARD);
   delay(400);
}
void turn180(){
   M1->setSpeed(180);
```

```
  else if(Leftbump == 0){
  backward();
  turnRight90();
  }
  else{
  straight();
  }
}
void straight(){
   M1->setSpeed(180);
   M2->setSpeed(255);
   M3->setSpeed(255);
   M4->setSpeed(180);
   M1->run(FORWARD);
   M2->run(FORWARD);
   M3->run(FORWARD);
   M4->run(FORWARD);
   delay(50);
}
```

```
   M2->setSpeed(255);
   M3->setSpeed(255);
   M4->setSpeed(180);
   M1->run(BACKWARD);
   M2->run(FORWARD);
   M3->run(BACKWARD);
   M4->run(FORWARD);
   delay(760);
}
```

Figure 11: This is the code of our project. It reads the entire first column and then to the top of the second column.