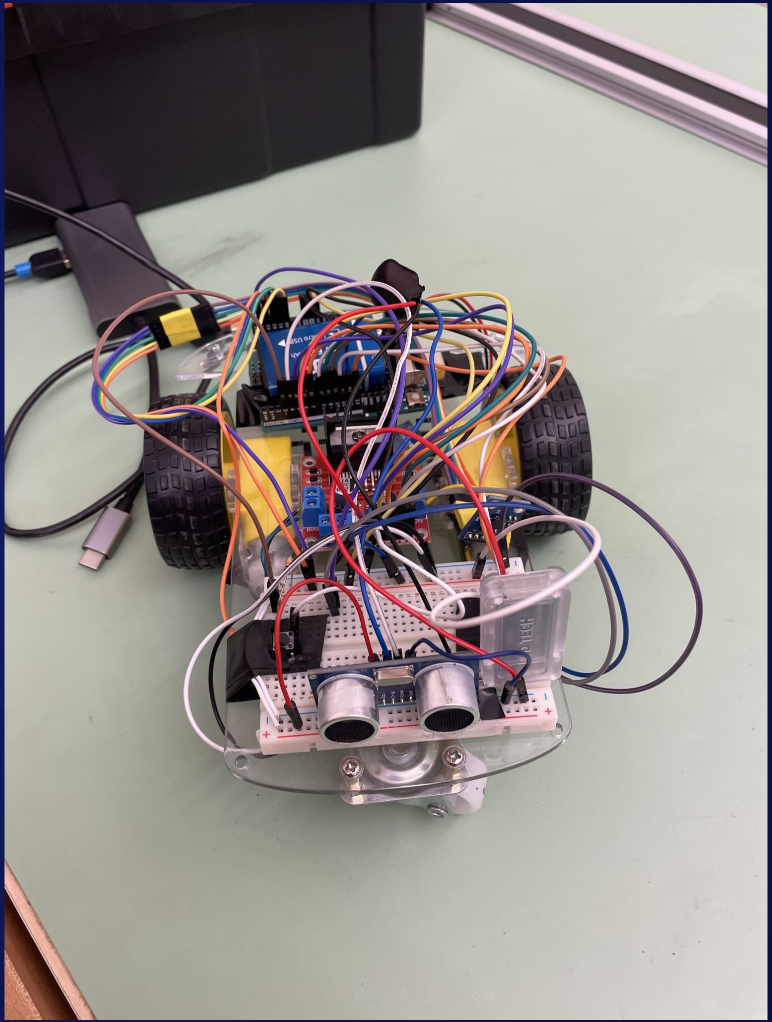
Automated Vehicle

By Elliot Kang

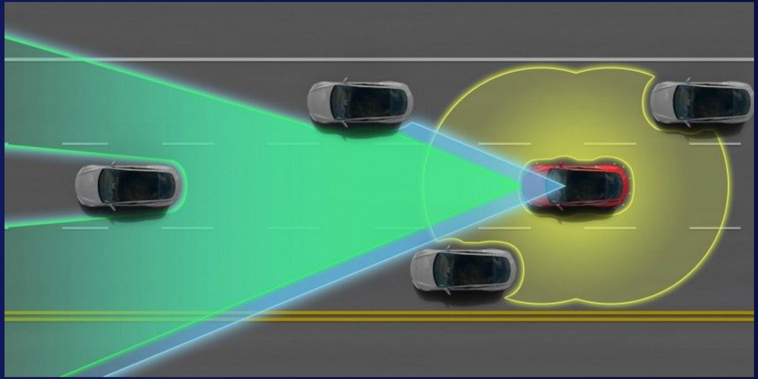


Contents

Introduction	Project Introduction
Project Goal	Drive with bluetooth joystick and Automated with Line sensor, Store Accelerometer to EERPOM
Functions	Functions & Features
Data/Demo	Data received from the EEPROM
Conclusion	Next Steps

Project Goal

Create a vehicle that can drive around using a Bluetooth joystick, Ultrasonic sensor for crash avoidance, record data to EEPROM via accelerometer data.

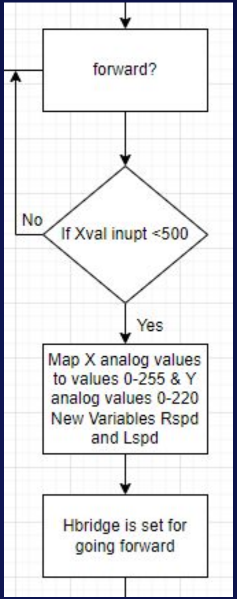


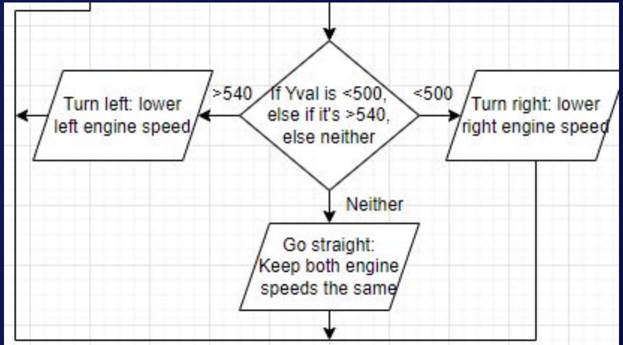
01

Flow Diagram

Forward!

Flow Diagram of Function

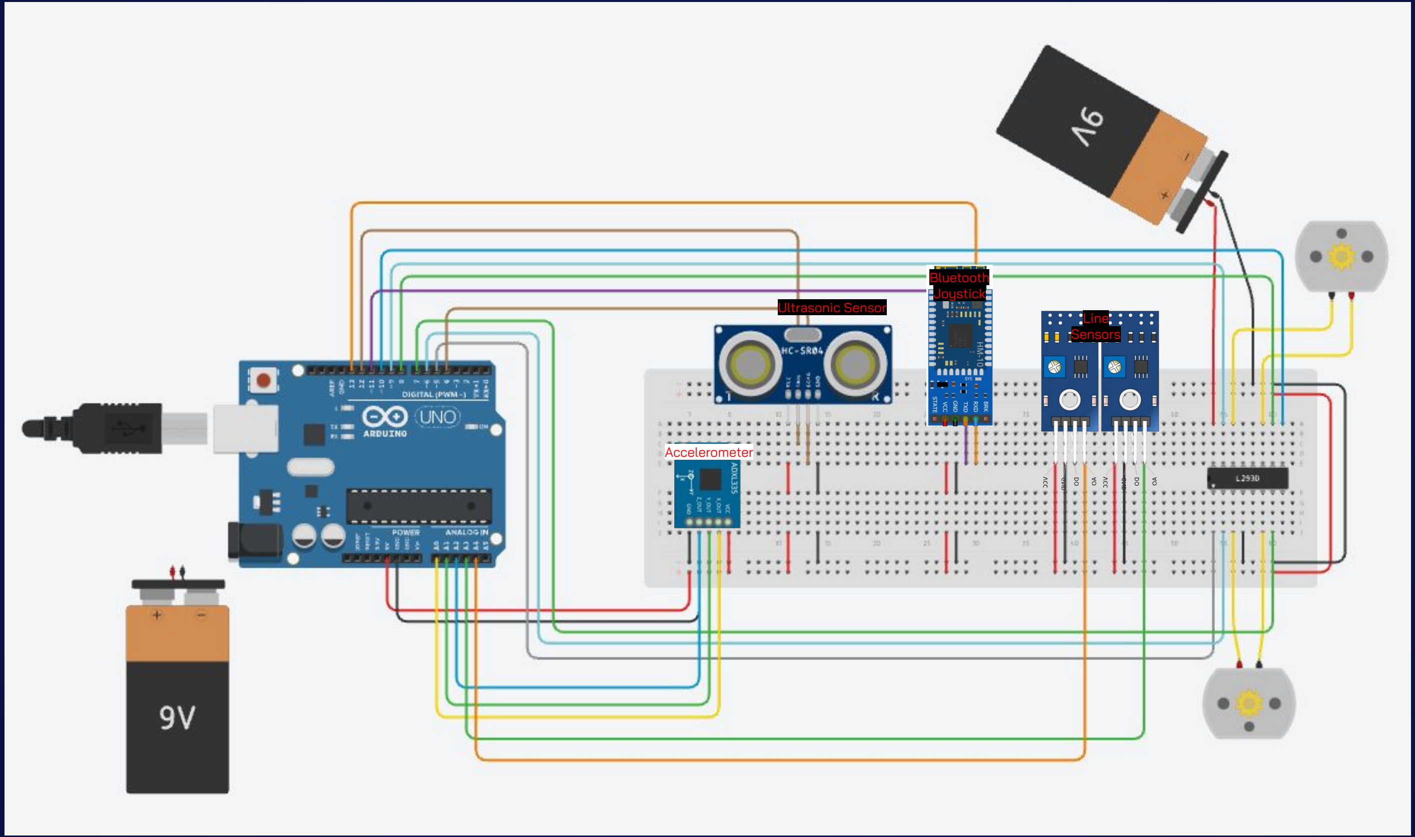

forward?
No
If Xval inupt <500
Yes
Map X analog values to values 0-255 & Y analog values 0-220 New Variables Rspd and Lspd
Hbridge is set for going forward


Turn left: lower left engine speed
>540
If Yval is <500, else if it's >540, else neither
<500
Turn right: lower right engine speed
Neither
Go straight: Keep both engine speeds the same

02

Circuitry

Tinkercad Circuitry

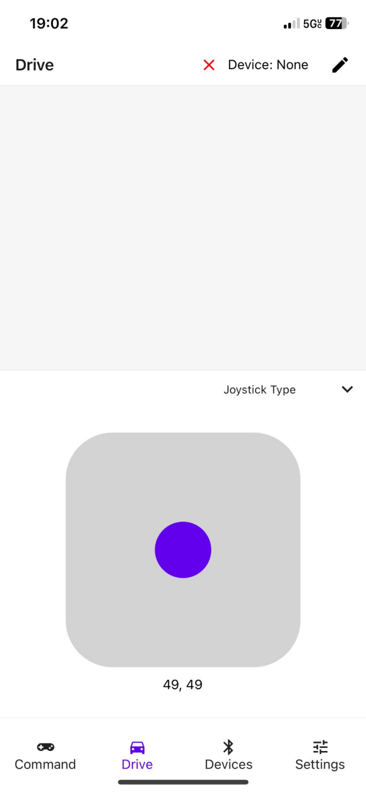


03

Code/Functions

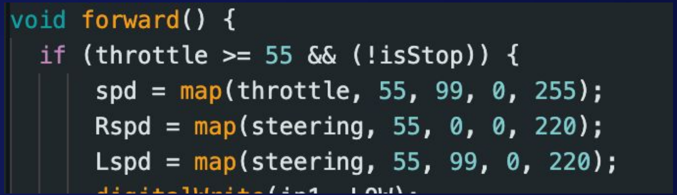
Bluetooth Joystick

- Using Arduino Blue app you can connect phone to Arduino
- Switch modes to line sensor mode and

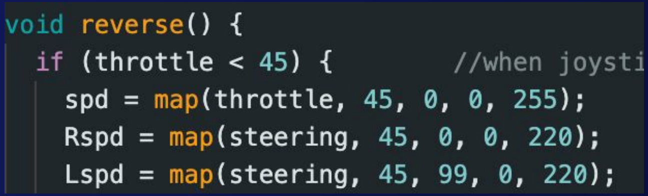


Motor Functions Forward/Backward

- in1 and in2 is right motor
- in3 and in4 is left motor
- Swap inputs to swap direction
- Can control using Bluetooth Joystick



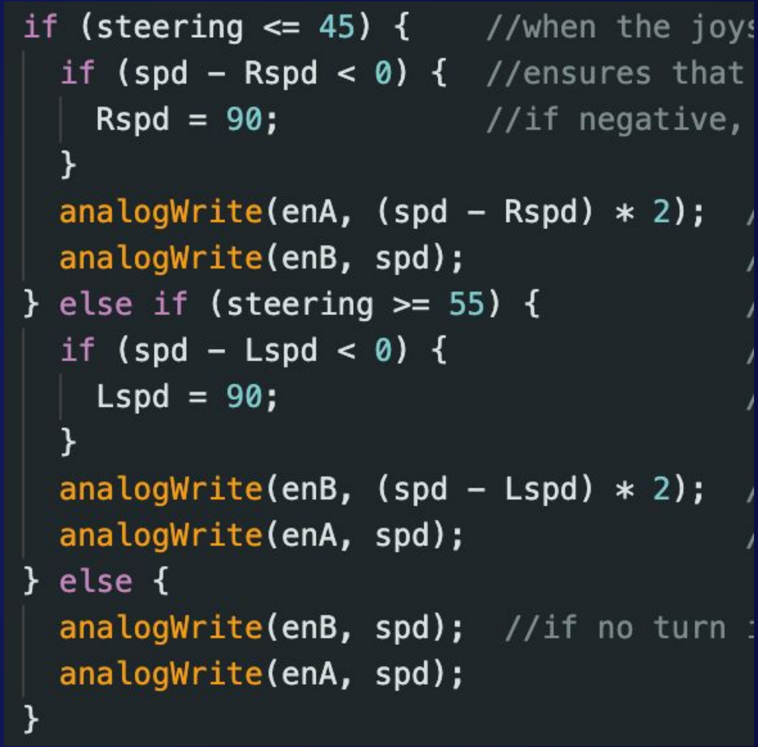
```
void forward() {
  if (throttle >= 55 && (!isStop)) {
    spd = map(throttle, 55, 99, 0, 255);
    Rspd = map(steering, 55, 0, 0, 220);
    Lspd = map(steering, 55, 99, 0, 220);
    digitalWrite(in1, LOW);
```



```
void reverse() {
  if (throttle < 45) {          //when joysti
    spd = map(throttle, 45, 0, 0, 255);
    Rspd = map(steering, 45, 0, 0, 220);
    Lspd = map(steering, 45, 99, 0, 220);
```

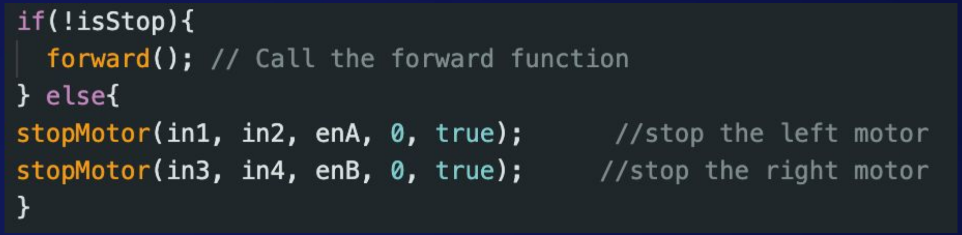
Motor Functions Turning

- Turns the car via subtracting the speed of one wheel by the turn amount on the joystick.
- Turning is in the horizontal direction.
- Same turning function works both forwards and backwards



```
if (steering <= 45) {    //when the joys
  if (spd - Rspd < 0) {  //ensures that
    Rspd = 90;           //if negative,
  }
  analogWrite(enA, (spd - Rspd) * 2);  /
  analogWrite(enB, spd);               /
} else if (steering >= 55) {           /
  if (spd - Lspd < 0) {                /
    Lspd = 90;                         /
  }
  analogWrite(enB, (spd - Lspd) * 2);  /
  analogWrite(enA, spd);               /
} else {
  analogWrite(enB, spd);  //if no turn
  analogWrite(enA, spd);
}
```

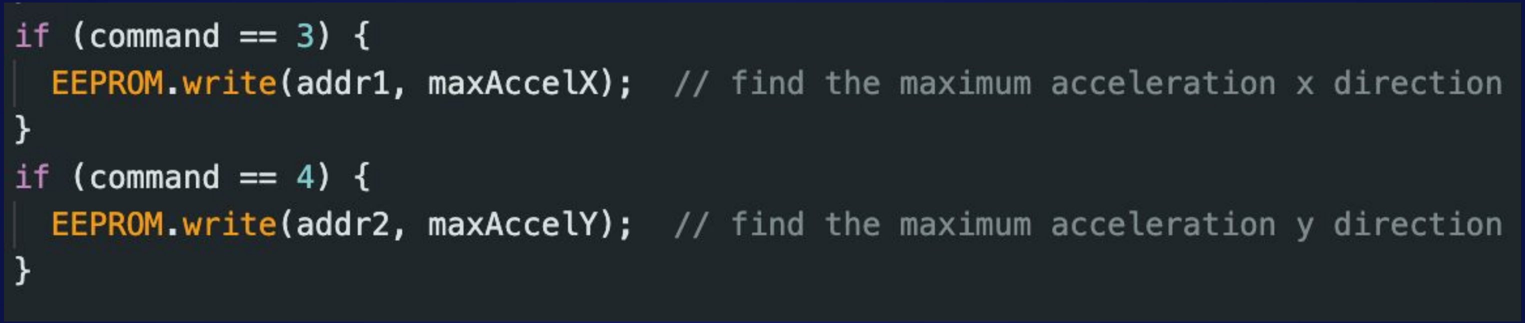
Ultrasonic Sensor



```
if(!isStop){
  forward(); // Call the forward function
} else{
stopMotor(in1, in2, enA, 0, true);      //stop the left motor
stopMotor(in3, in4, enB, 0, true);     //stop the right motor
}
```

Gauges distances using sound by converting electrical signals into sound waves. If the ultrasonic detects something in the way then the motors will stop and will not let you go forward. However, you may back up to get out of your position.

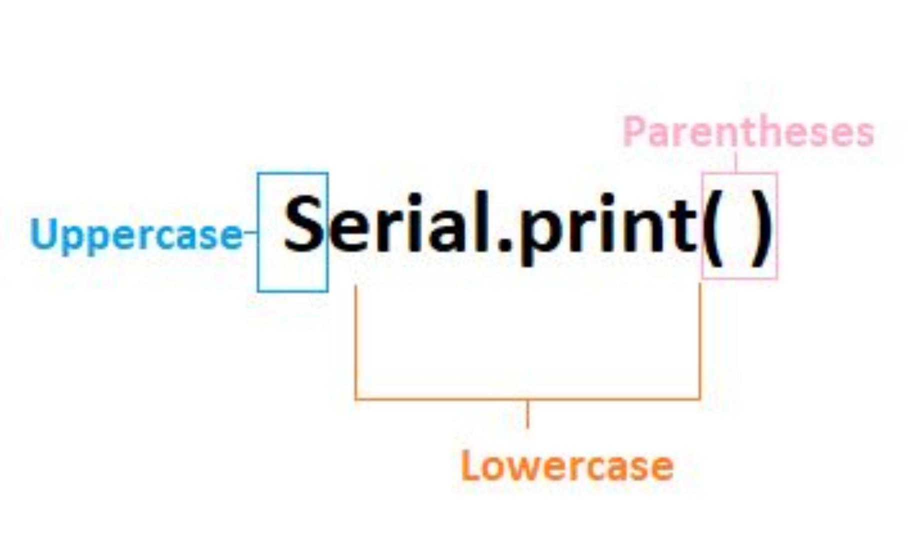
EEPROM



```
if (command == 3) {
  EEPROM.write(addr1, maxAccelX);  // find the maximum acceleration x direction
}
if (command == 4) {
  EEPROM.write(addr2, maxAccelY);  // find the maximum acceleration y direction
}
```

If you send a 3 to Arduino, then the EEPROM will take in the max acceleration of X-axis. If you send a 4 to Arduino, then the EEPROM will take in the max acceleration of Y-axis.

EEPROM (continued.)

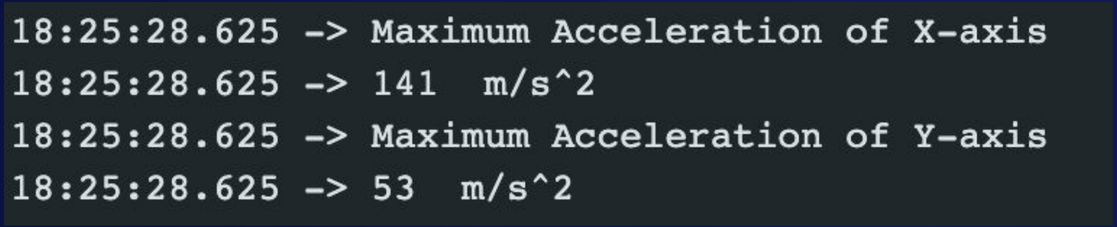


Used a second program to print EEPROM data

04

Data

EEPROM Data



```
18:25:28.625 -> Maximum Acceleration of X-axis
18:25:28.625 -> 141  m/s^2
18:25:28.625 -> Maximum Acceleration of Y-axis
18:25:28.625 -> 53  m/s^2
```

What Does this Mean?

The data we got was the maximum acceleration in the x and y axis.

Crash Avoidance Video

04

Final Thoughts

Collaboration

From Classmates

- Nico and Kenneth helped me with my Bluetooth joystick for forward, reverse, and stop functions.
- Roderick helped me with wiring
- Matt helped me with line sensor and Bluetooth command communication

To Classmates

- Helped Matt with EEPROM
- Helped Kenneth and Nico with crash avoidance via ultrasonic sensor

Improvements

Line Sensor

Still unsure why my line sensor doesn’t work, but in theory it works

Communication

Using button for better EEPROM recording

Expansions

- Stop rear light (red)
- Backup light (white)
- Potentiometer and speakers for stereo/radio speaker
- LCD display to display things

Conclusion

Input: Accelerometer, Line Sensor

Output: Motors, Ultrasonic Sensor, EEPROM

Libraries: EEPROM, stdbool, ArduinoBlue, SoftwareSerial

Functions: forward(), reverse(), stop(), debounceCalibrate(), ultrasonicSensor(), ReadAxis(), Calibrate(), AutoCalibrate(), lineSensor(), stateChange()

Thanks!

Do you have any questions?