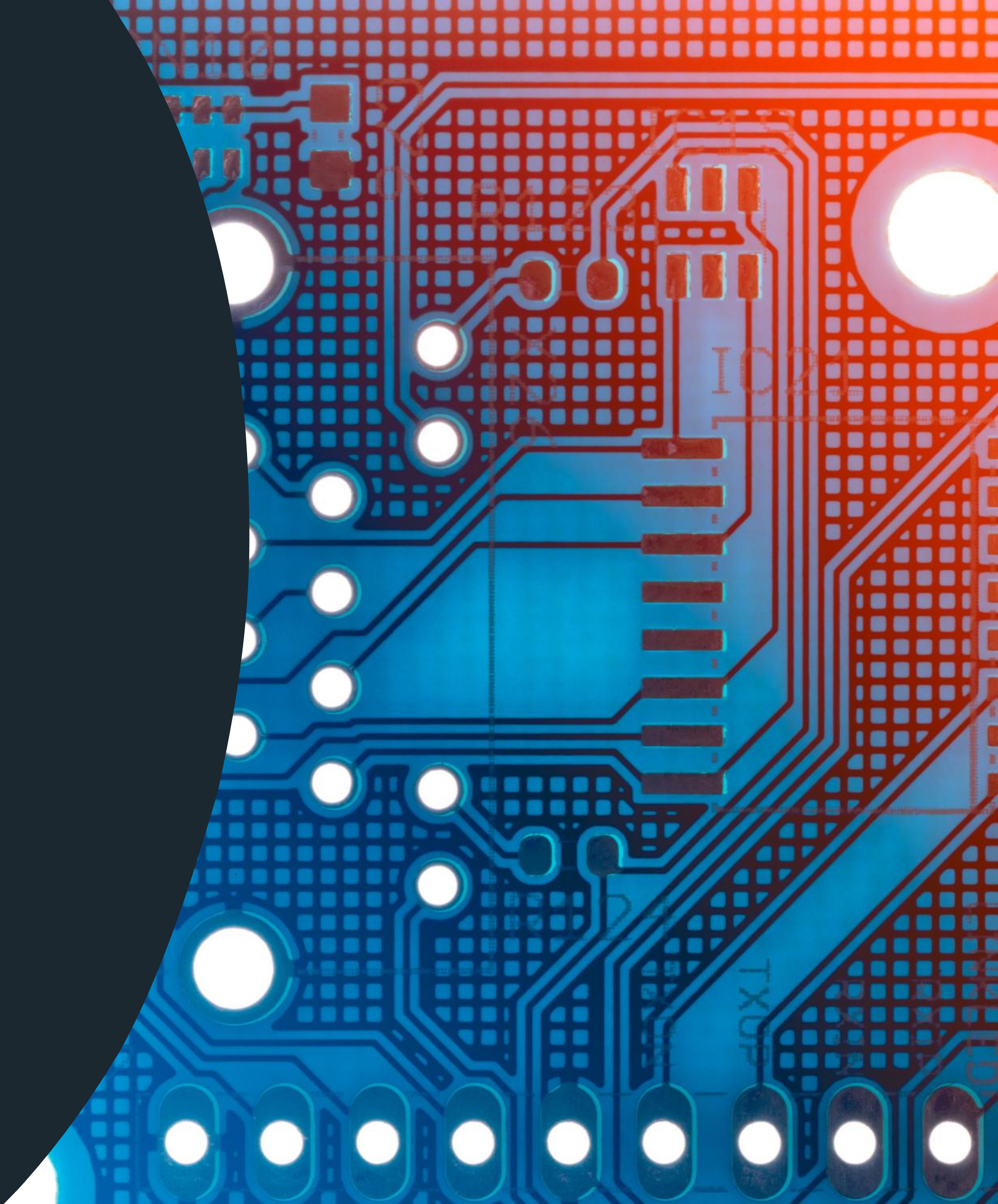
Autonomous Ambulance System

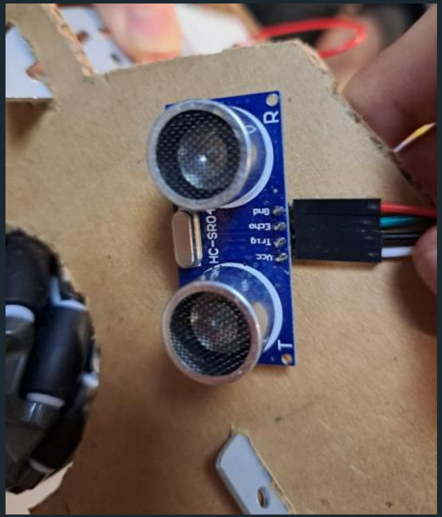
By: Enrique Garcia, Luis Garcia, Yahir Dominguez, Jacob Tenorio, Sam Herrea, Jorge Del Angel

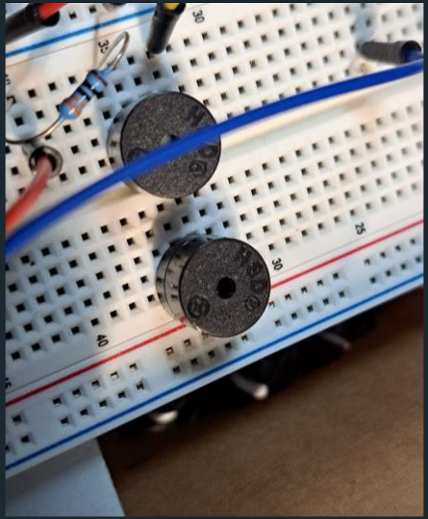


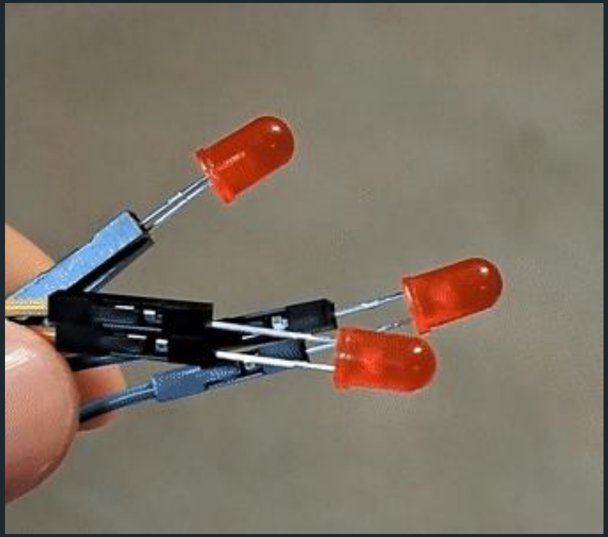
Introduction
Arduino-Powered DC Ambulance

Objective: design, manufacture and create a fully autonomous Arduino powered DC motor ambulance with three main functions:

1. Ability to maneuver and detect obstacles(turning when appropriate)
2. Proper LED system
3. Buzzer for sirens

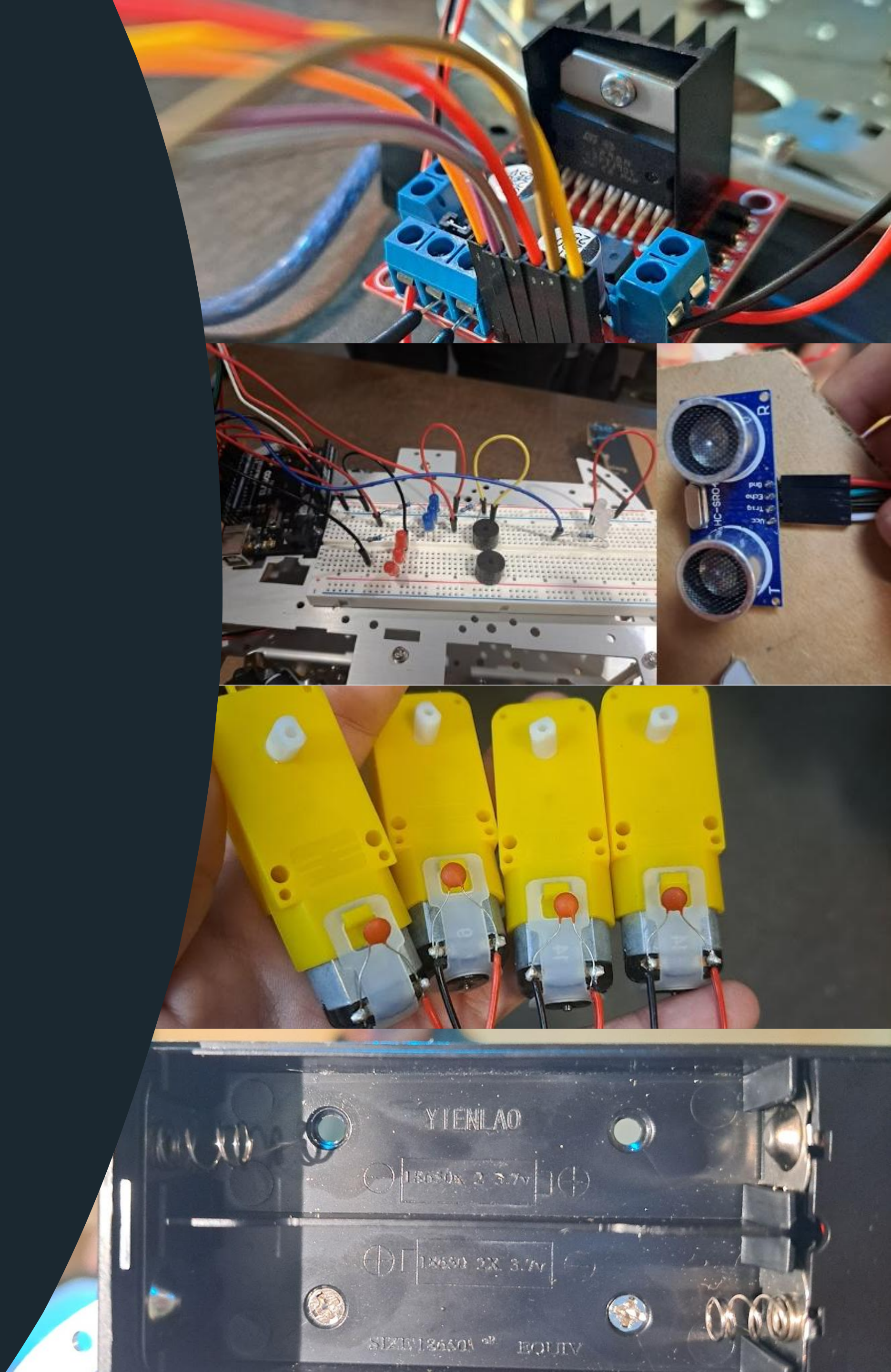






Components of the Ambulance

- Breadboard
- Arduino UNO
- DC motor
- LED lights
- Buzzer
- DC motor drive
- Power supply
- Ultra-Sonic Sensor
- Wires (M-F , M-M,)
- Resistors



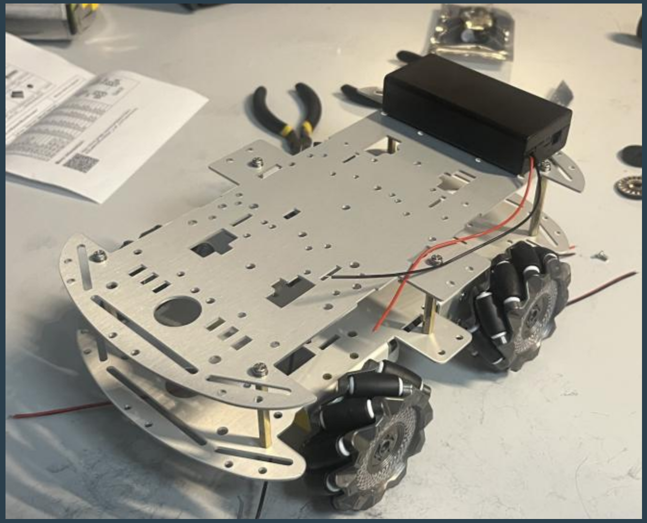
Items and their prices

Item Description	Vendor/Store	Cost (USD)
ELEGOO UNO Project Super Starter Kit	Amazon	$35.99
Car Kit with Metal Chassis and TT Motor Smart Robot 4WD	Amazon	$26.90
Buzzers(Pack of 10)	Amazon	$6.98
3.7V Batteries(2)	Amazon	$9.98
Motor Driver Controller Board	Amazon	$6.99

Initial attempt to making the Body of the Ambulance

Initial Problems

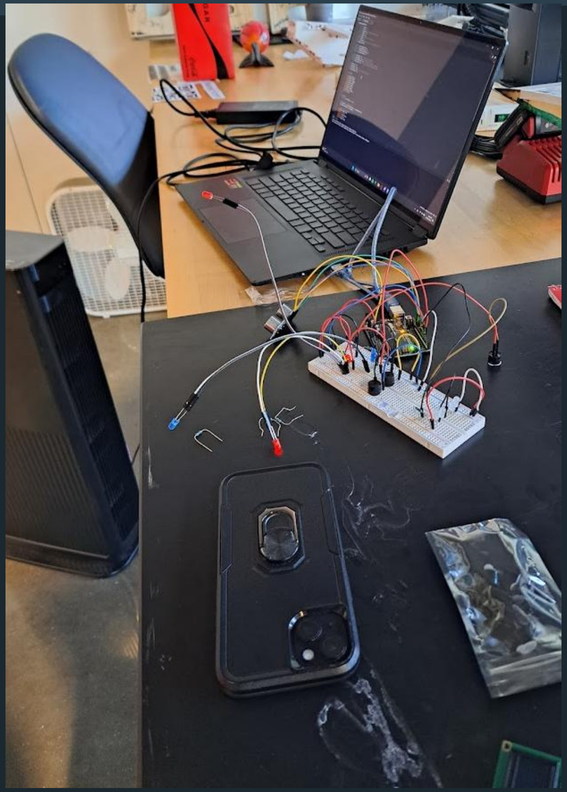
- Motors were not pre-soldered with wires
- No clear instructions provided to assemble
- We didn't know how to make the LED light or buzzers work

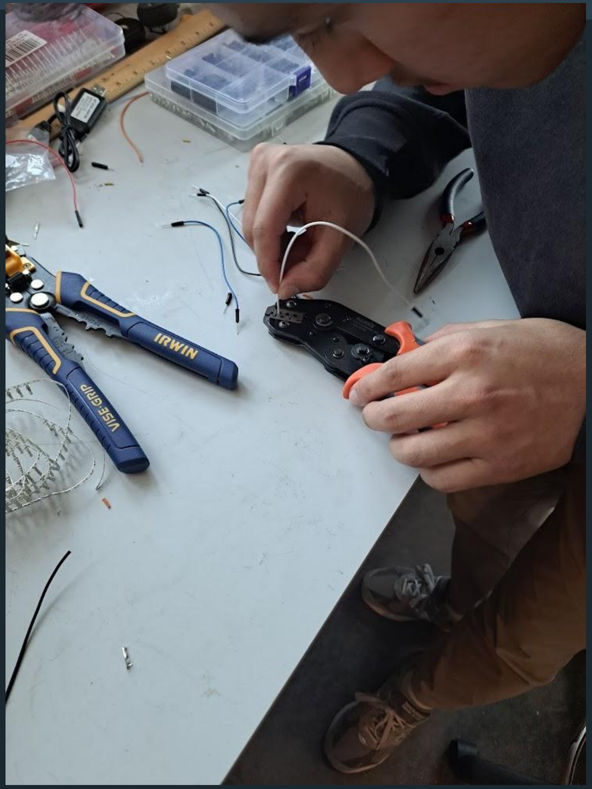


Initial attempt into making the Circuits work

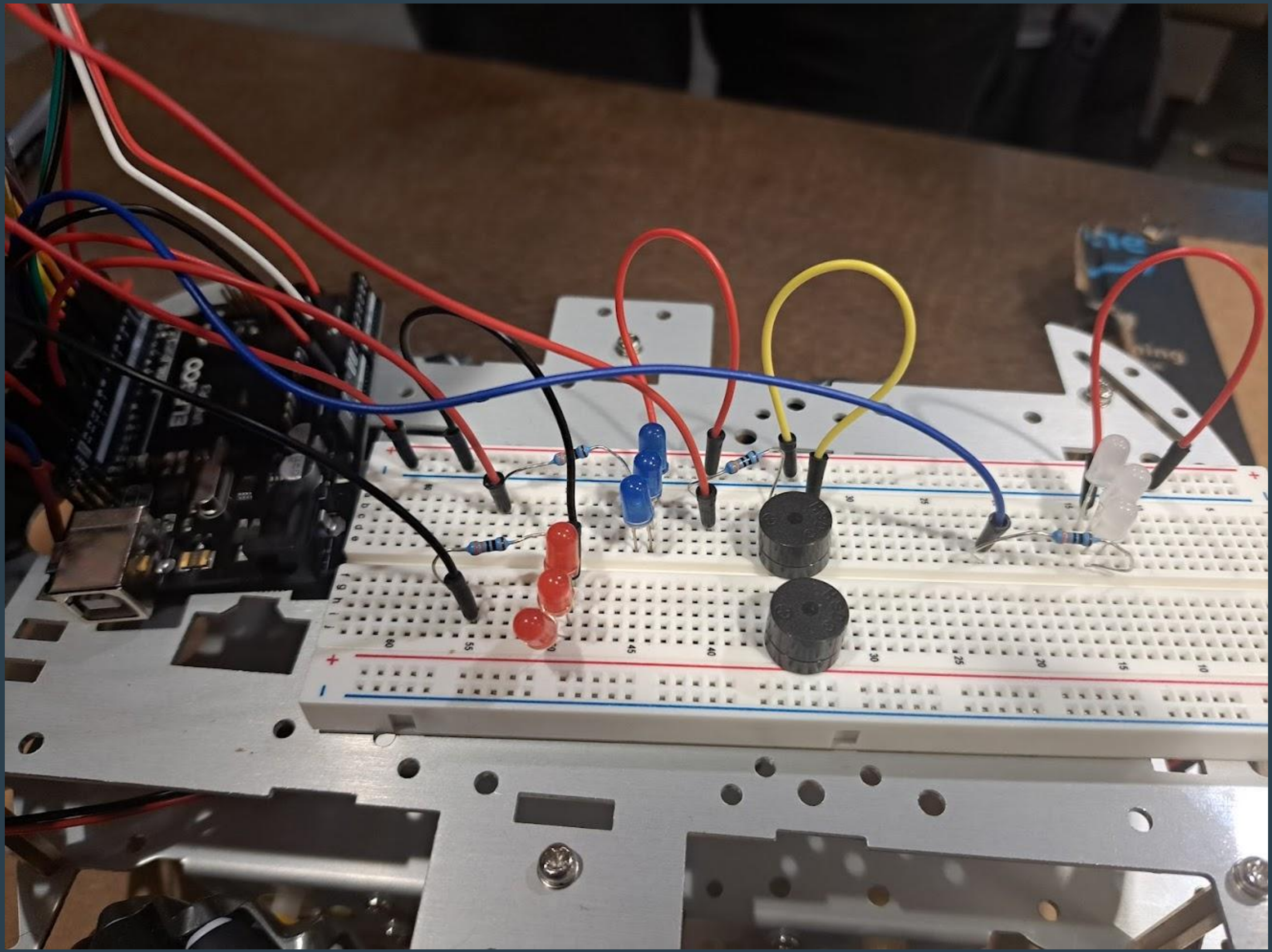
Initial Circuits Problems

- No Motor Drive (without it motors won't run)
- No batteries (no power in the circuit)
- Didn't have enough M-F wires (LED lights needed extension)
- We forgot to use Resistors causing an LED to pop

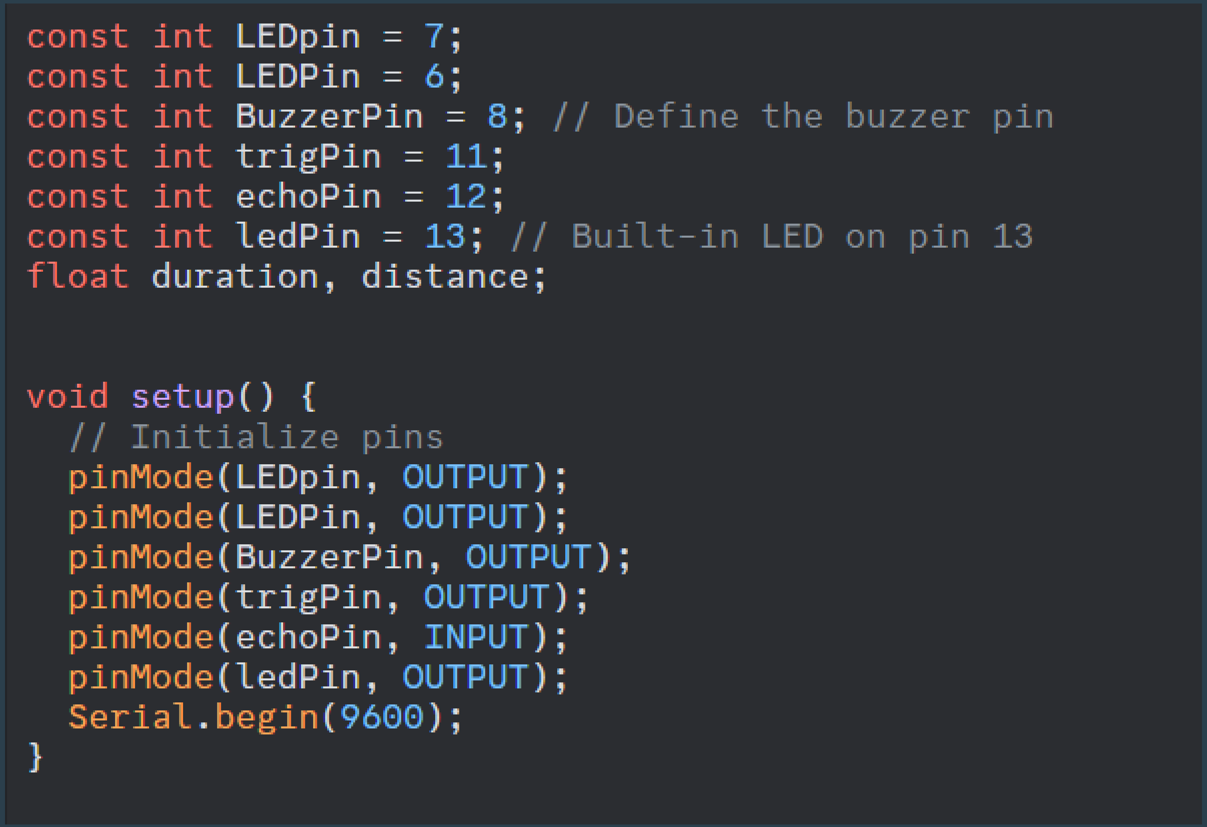




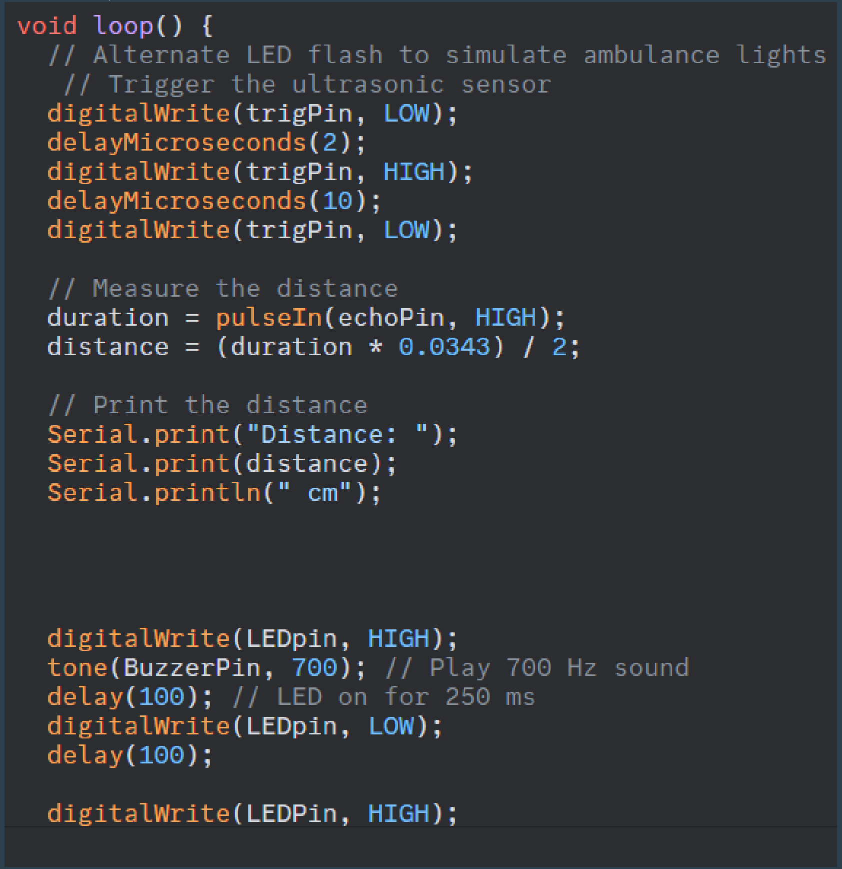
Circuit of the LED light and Buzzer



Initial code for LED lights and Buzzers (sensor too)



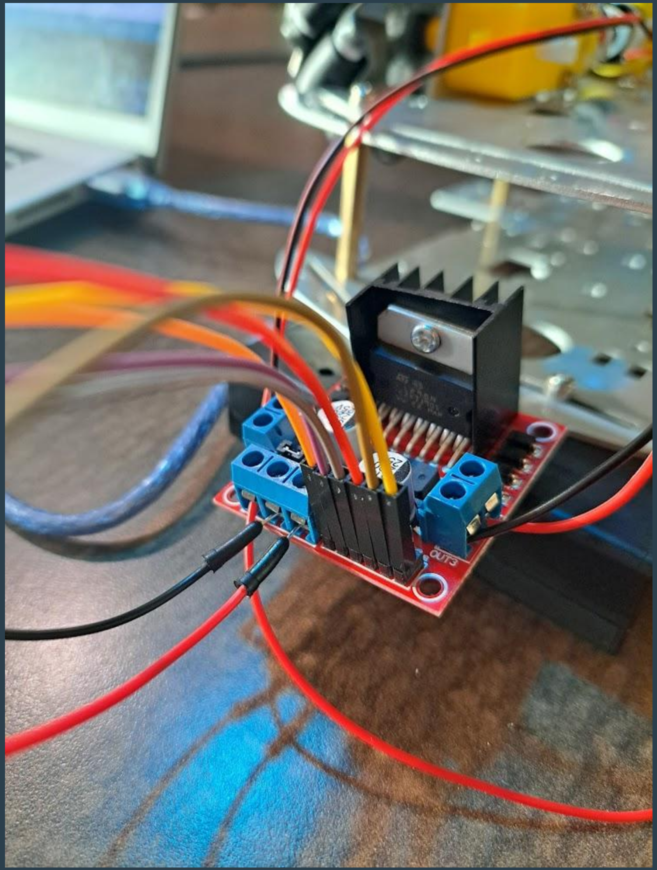
```
const int LEDpin = 7;
const int LEDPin = 6;
const int BuzzerPin = 8; // Define the buzzer pin
const int trigPin = 11;
const int echoPin = 12;
const int ledPin = 13; // Built-in LED on pin 13
float duration, distance;


void setup() {
  // Initialize pins
  pinMode(LEDpin, OUTPUT);
  pinMode(LEDPin, OUTPUT);
  pinMode(BuzzerPin, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}
```

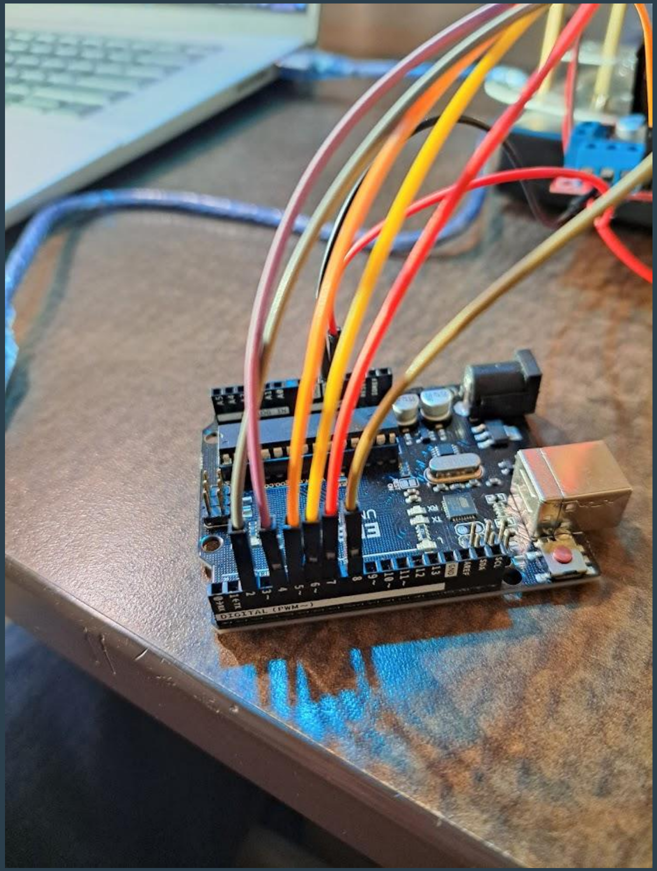


```
void loop() {
  // Alternate LED flash to simulate ambulance lights
   // Trigger the ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the distance
  duration = pulseIn(echoPin, HIGH);
  distance = (duration * 0.0343) / 2;

  // Print the distance
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");




  digitalWrite(LEDpin, HIGH);
  tone(BuzzerPin, 700); // Play 700 Hz sound
  delay(100); // LED on for 250 ms
  digitalWrite(LEDpin, LOW);
  delay(100);

  digitalWrite(LEDPin, HIGH);
```

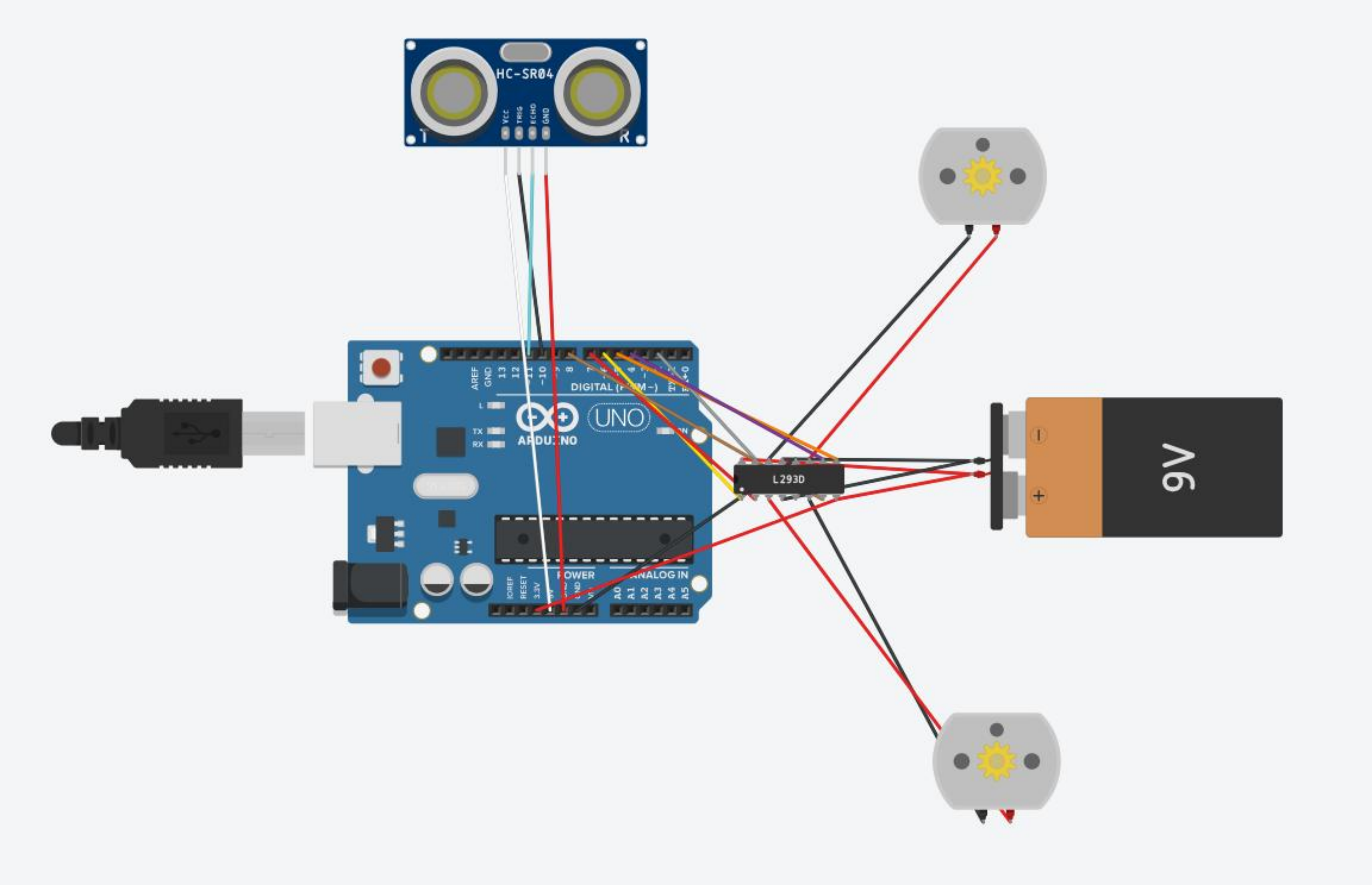
Initial attempt to test the motors

- We accidentally didn't connect the wires correctly which lead to one of the motor drivers popping
- Initial code wouldn't work

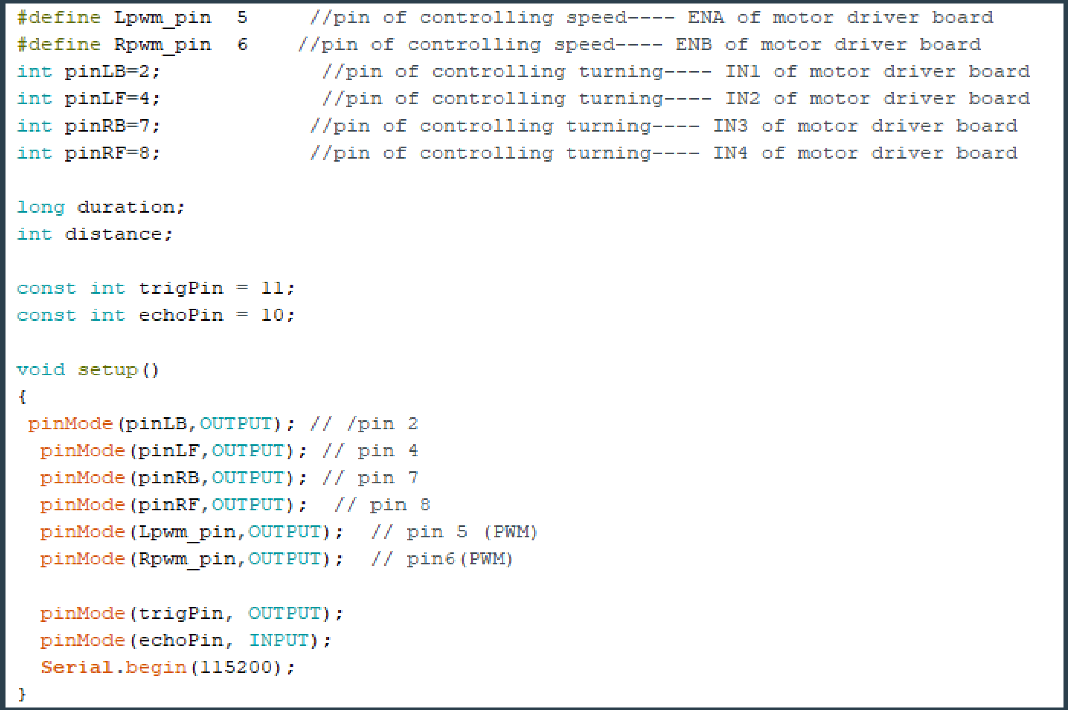




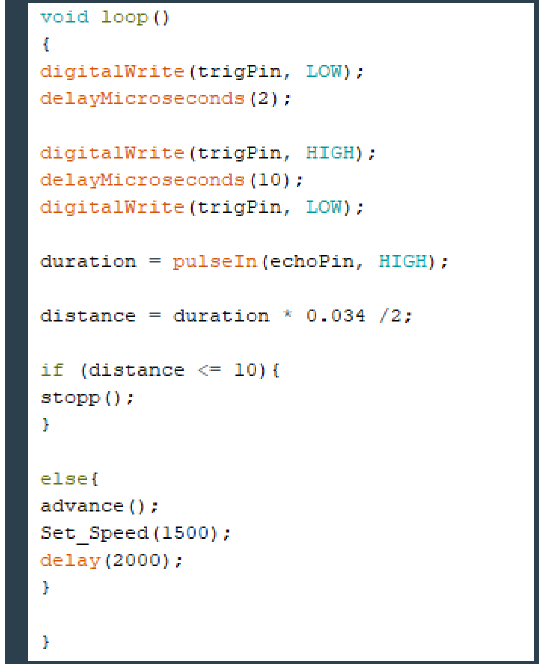
Initial Motor + Sensor Circuit



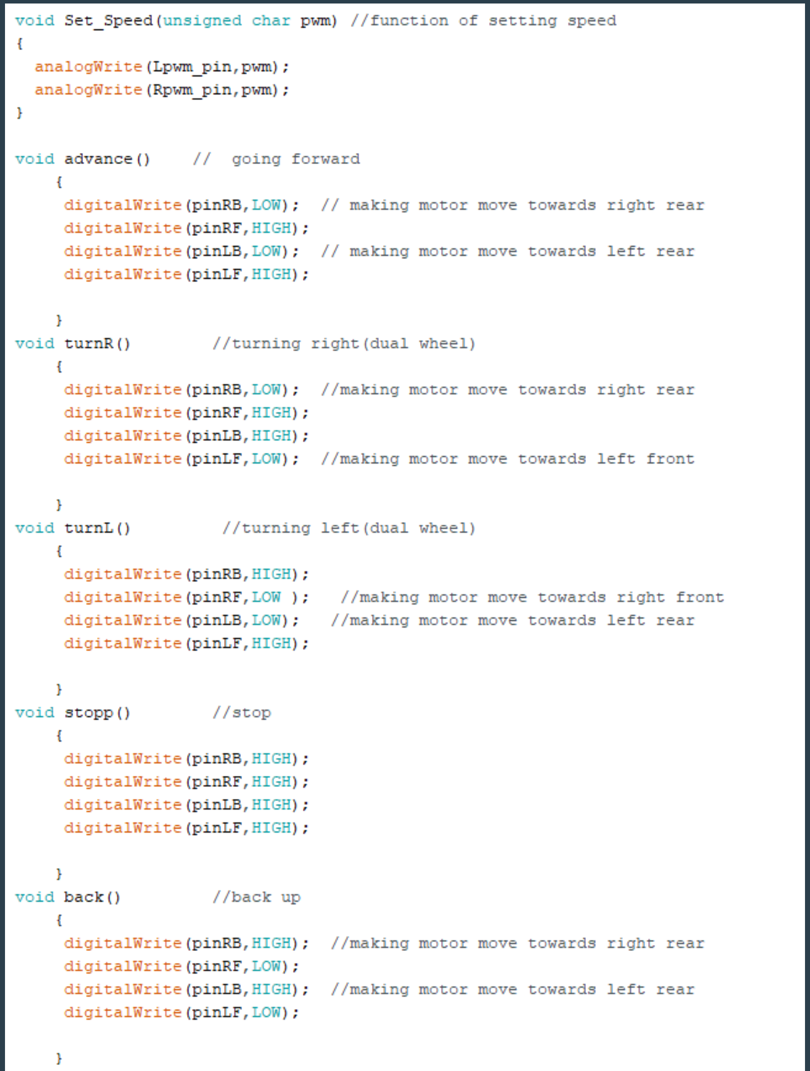
Code to test motors



```
#define Lpwm_pin  5     //pin of controlling speed---- ENA of motor driver board
#define Rpwm_pin  6    //pin of controlling speed---- ENB of motor driver board
int pinLB=2;             //pin of controlling turning---- IN1 of motor driver board
int pinLF=4;             //pin of controlling turning---- IN2 of motor driver board
int pinRB=7;            //pin of controlling turning---- IN3 of motor driver board
int pinRF=8;            //pin of controlling turning---- IN4 of motor driver board

long duration;
int distance;

const int trigPin = 11;
const int echoPin = 10;

void setup()
{
 pinMode(pinLB,OUTPUT); // /pin 2
  pinMode(pinLF,OUTPUT); // pin 4
  pinMode(pinRB,OUTPUT); // pin 7
  pinMode(pinRF,OUTPUT);  // pin 8
  pinMode(Lpwm_pin,OUTPUT);  // pin 5 (PWM)
  pinMode(Rpwm_pin,OUTPUT);  // pin6(PWM)

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(115200);
}
```

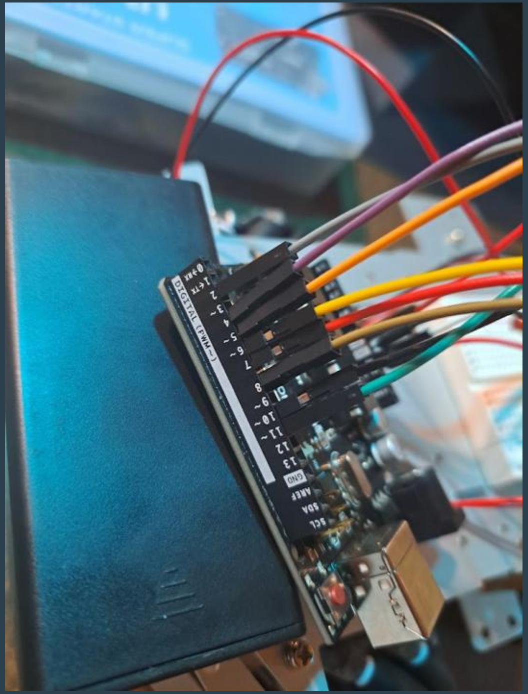


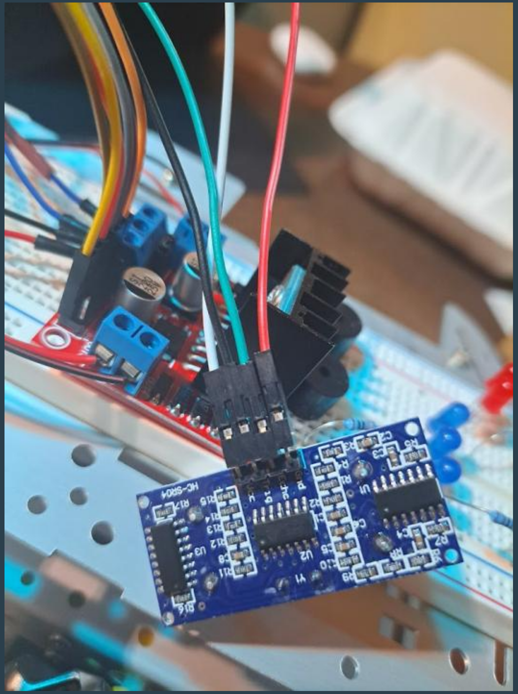
```
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration * 0.034 /2;

if (distance <= 10){
stopp();
}

else{
advance();
Set_Speed(1500);
delay(2000);
}

}
```

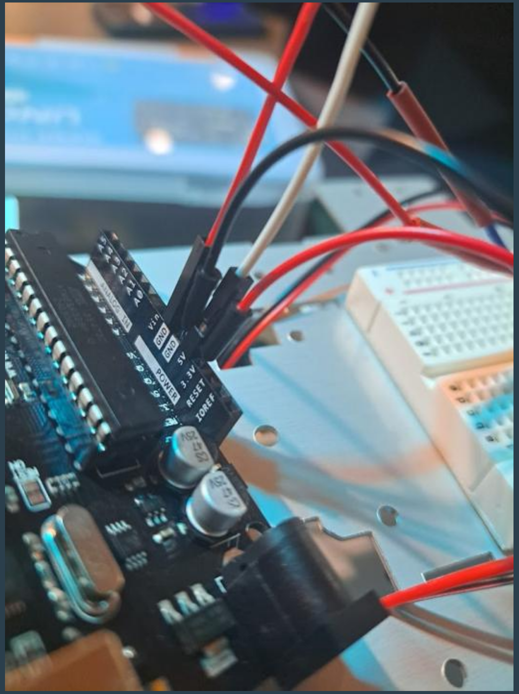


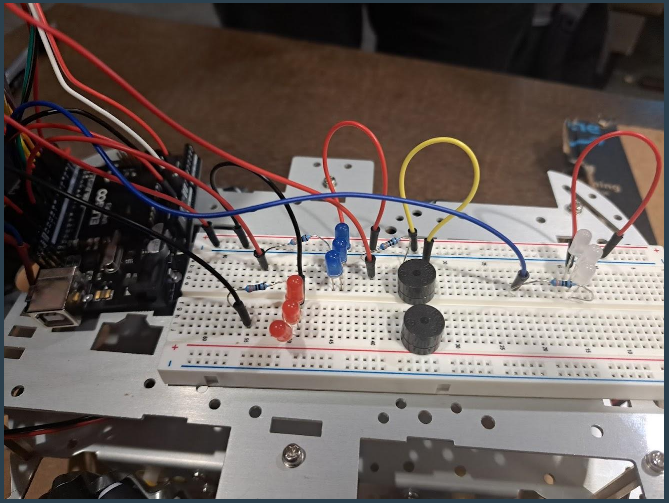
```
void Set_Speed(unsigned char pwm) //function of setting speed
{
  analogWrite(Lpwm_pin,pwm);
  analogWrite(Rpwm_pin,pwm);
}

void advance()    //  going forward
    {
     digitalWrite(pinRB,LOW);  // making motor move towards right rear
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,LOW);  // making motor move towards left rear
     digitalWrite(pinLF,HIGH);

    }
void turnR()        //turning right(dual wheel)
    {
     digitalWrite(pinRB,LOW);  //making motor move towards right rear
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,LOW);  //making motor move towards left front

    }
void turnL()         //turning left(dual wheel)
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,LOW );   //making motor move towards right front
     digitalWrite(pinLB,LOW);   //making motor move towards left rear
     digitalWrite(pinLF,HIGH);

    }
void stopp()        //stop
    {
     digitalWrite(pinRB,HIGH);
     digitalWrite(pinRF,HIGH);
     digitalWrite(pinLB,HIGH);
     digitalWrite(pinLF,HIGH);

    }
void back()         //back up
    {
     digitalWrite(pinRB,HIGH);  //making motor move towards right rear
     digitalWrite(pinRF,LOW);
     digitalWrite(pinLB,HIGH);  //making motor move towards left rear
     digitalWrite(pinLF,LOW);

    }
```

Bringing the LEDs, buzzer, sensors, and motors all together

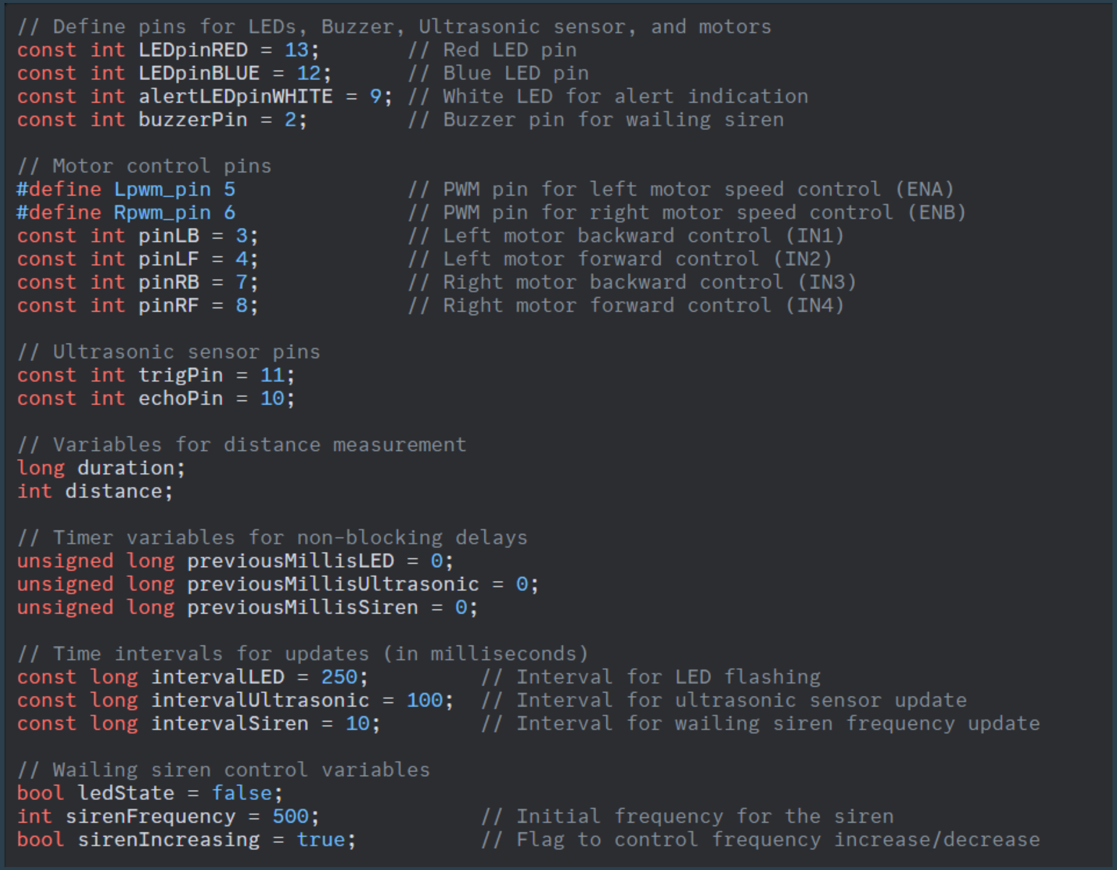




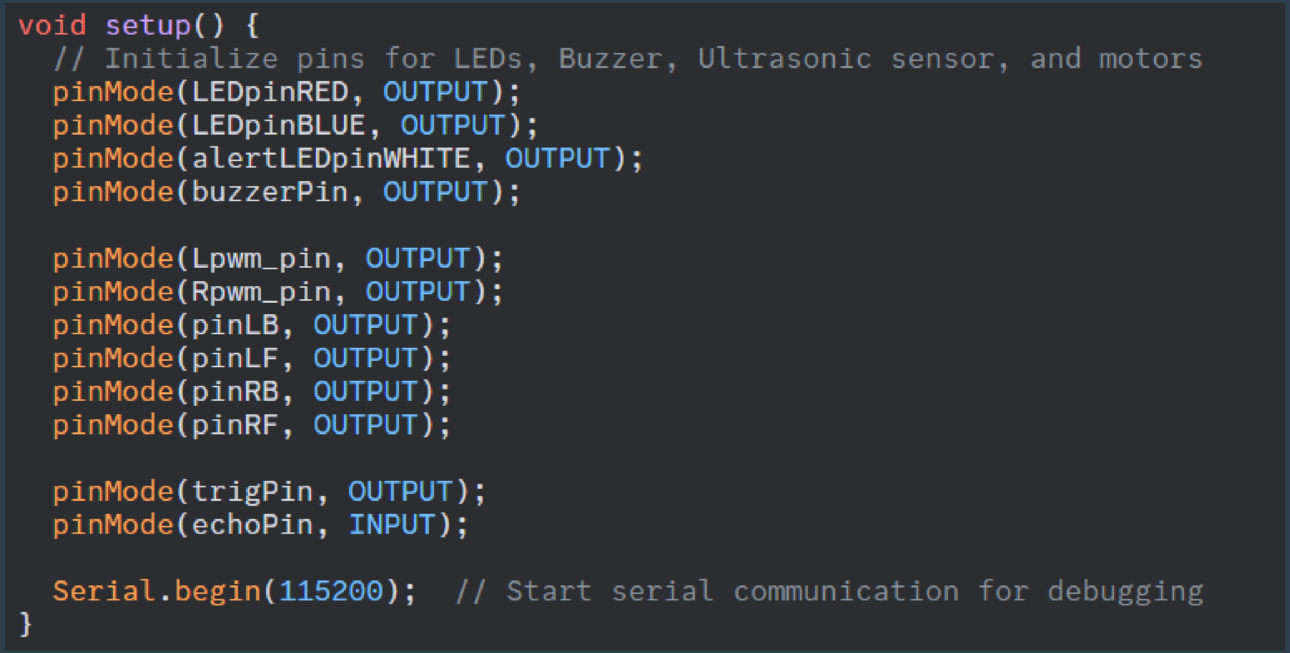




Final Code – Initial Part of Code

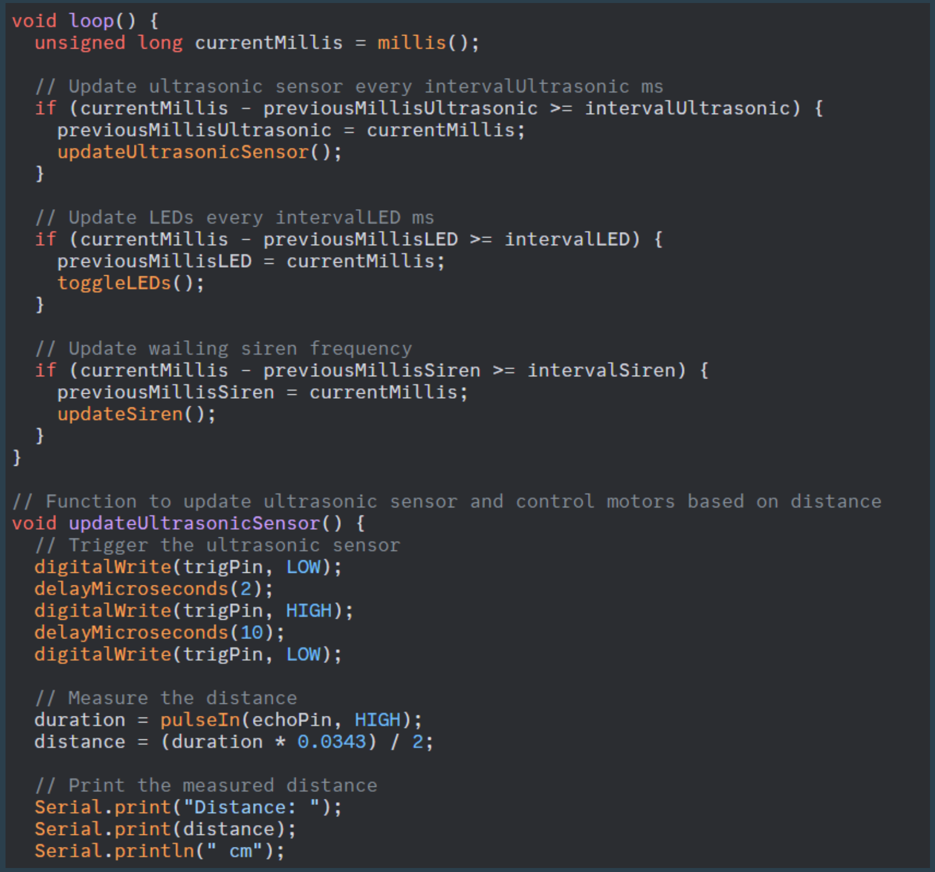


```
// Define pins for LEDs, Buzzer, Ultrasonic sensor, and motors
const int LEDpinRED = 13;       // Red LED pin
const int LEDpinBLUE = 12;      // Blue LED pin
const int alertLEDpinWHITE = 9; // White LED for alert indication
const int buzzerPin = 2;        // Buzzer pin for wailing siren

// Motor control pins
#define Lpwm_pin 5              // PWM pin for left motor speed control (ENA)
#define Rpwm_pin 6              // PWM pin for right motor speed control (ENB)
const int pinLB = 3;            // Left motor backward control (IN1)
const int pinLF = 4;            // Left motor forward control (IN2)
const int pinRB = 7;            // Right motor backward control (IN3)
const int pinRF = 8;            // Right motor forward control (IN4)

// Ultrasonic sensor pins
const int trigPin = 11;
const int echoPin = 10;

// Variables for distance measurement
long duration;
int distance;

// Timer variables for non-blocking delays
unsigned long previousMillisLED = 0;
unsigned long previousMillisUltrasonic = 0;
unsigned long previousMillisSiren = 0;

// Time intervals for updates (in milliseconds)
const long intervalLED = 250;         // Interval for LED flashing
const long intervalUltrasonic = 100;  // Interval for ultrasonic sensor update
const long intervalSiren = 10;        // Interval for wailing siren frequency update

// Wailing siren control variables
bool ledState = false;
int sirenFrequency = 500;             // Initial frequency for the siren
bool sirenIncreasing = true;          // Flag to control frequency increase/decrease
```

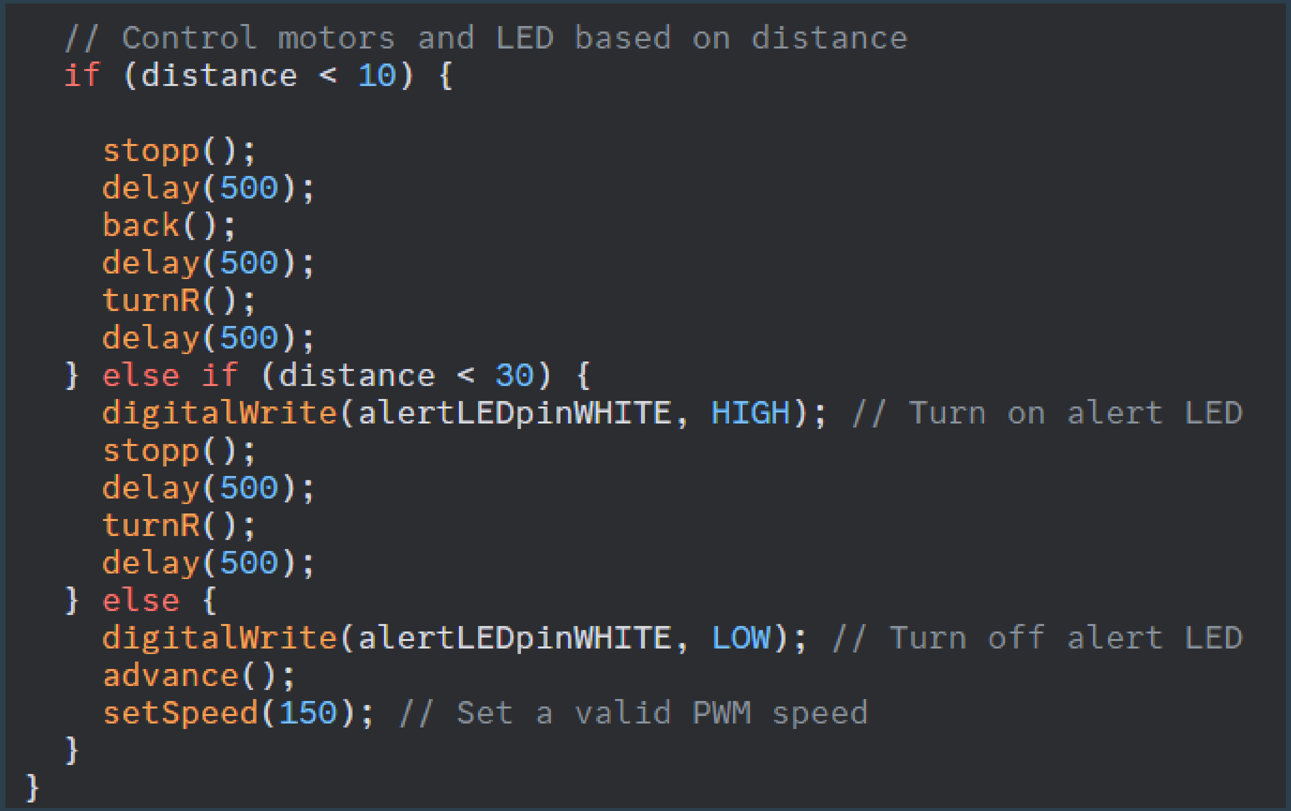


```
void setup() {
  // Initialize pins for LEDs, Buzzer, Ultrasonic sensor, and motors
  pinMode(LEDpinRED, OUTPUT);
  pinMode(LEDpinBLUE, OUTPUT);
  pinMode(alertLEDpinWHITE, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  pinMode(Lpwm_pin, OUTPUT);
  pinMode(Rpwm_pin, OUTPUT);
  pinMode(pinLB, OUTPUT);
  pinMode(pinLF, OUTPUT);
  pinMode(pinRB, OUTPUT);
  pinMode(pinRF, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(115200);  // Start serial communication for debugging
}
```

Final Code – The Loop

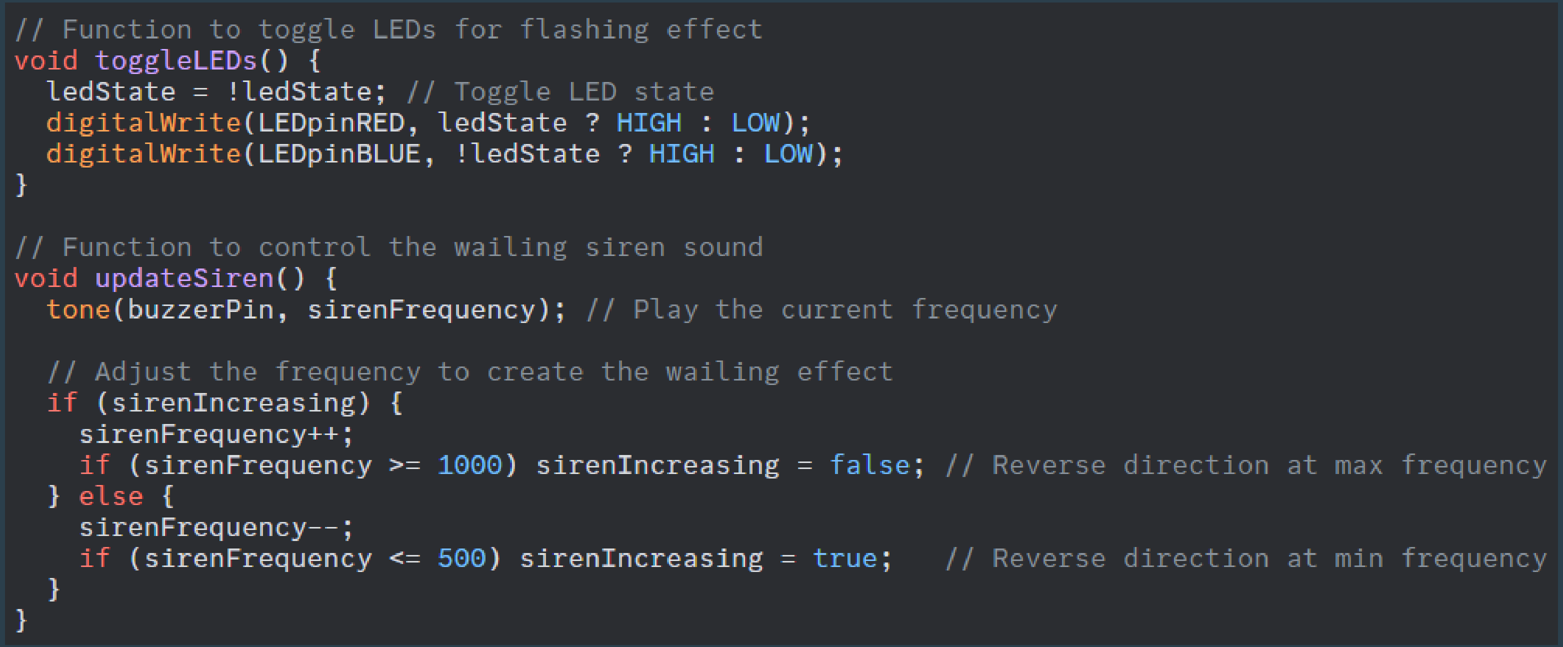


```
void loop() {
  unsigned long currentMillis = millis();

  // Update ultrasonic sensor every intervalUltrasonic ms
  if (currentMillis - previousMillisUltrasonic >= intervalUltrasonic) {
    previousMillisUltrasonic = currentMillis;
    updateUltrasonicSensor();
  }

  // Update LEDs every intervalLED ms
  if (currentMillis - previousMillisLED >= intervalLED) {
    previousMillisLED = currentMillis;
    toggleLEDs();
  }

  // Update wailing siren frequency
  if (currentMillis - previousMillisSiren >= intervalSiren) {
    previousMillisSiren = currentMillis;
    updateSiren();
  }
}

// Function to update ultrasonic sensor and control motors based on distance
void updateUltrasonicSensor() {
  // Trigger the ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Measure the distance
  duration = pulseIn(echoPin, HIGH);
  distance = (duration * 0.0343) / 2;

  // Print the measured distance
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
```

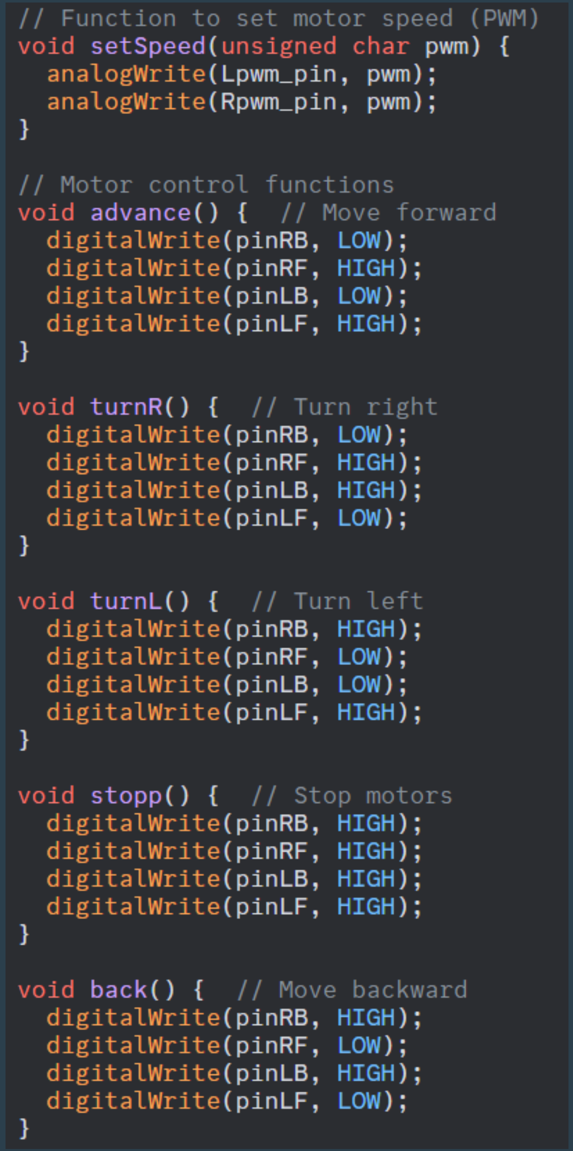


```
  // Control motors and LED based on distance
  if (distance < 10) {

    stopp();
    delay(500);
    back();
    delay(500);
    turnR();
    delay(500);
  } else if (distance < 30) {
    digitalWrite(alertLEDpinWHITE, HIGH); // Turn on alert LED
    stopp();
    delay(500);
    turnR();
    delay(500);
  } else {
    digitalWrite(alertLEDpinWHITE, LOW); // Turn off alert LED
    advance();
    setSpeed(150); // Set a valid PWM speed
  }
}
```

Final Code – Functions for the Ambulance



```
// Function to toggle LEDs for flashing effect
void toggleLEDs() {
  ledState = !ledState; // Toggle LED state
  digitalWrite(LEDpinRED, ledState ? HIGH : LOW);
  digitalWrite(LEDpinBLUE, !ledState ? HIGH : LOW);
}

// Function to control the wailing siren sound
void updateSiren() {
  tone(buzzerPin, sirenFrequency); // Play the current frequency

  // Adjust the frequency to create the wailing effect
  if (sirenIncreasing) {
    sirenFrequency++;
    if (sirenFrequency >= 1000) sirenIncreasing = false; // Reverse direction at max frequency
  } else {
    sirenFrequency--;
    if (sirenFrequency <= 500) sirenIncreasing = true;   // Reverse direction at min frequency
  }
}
```

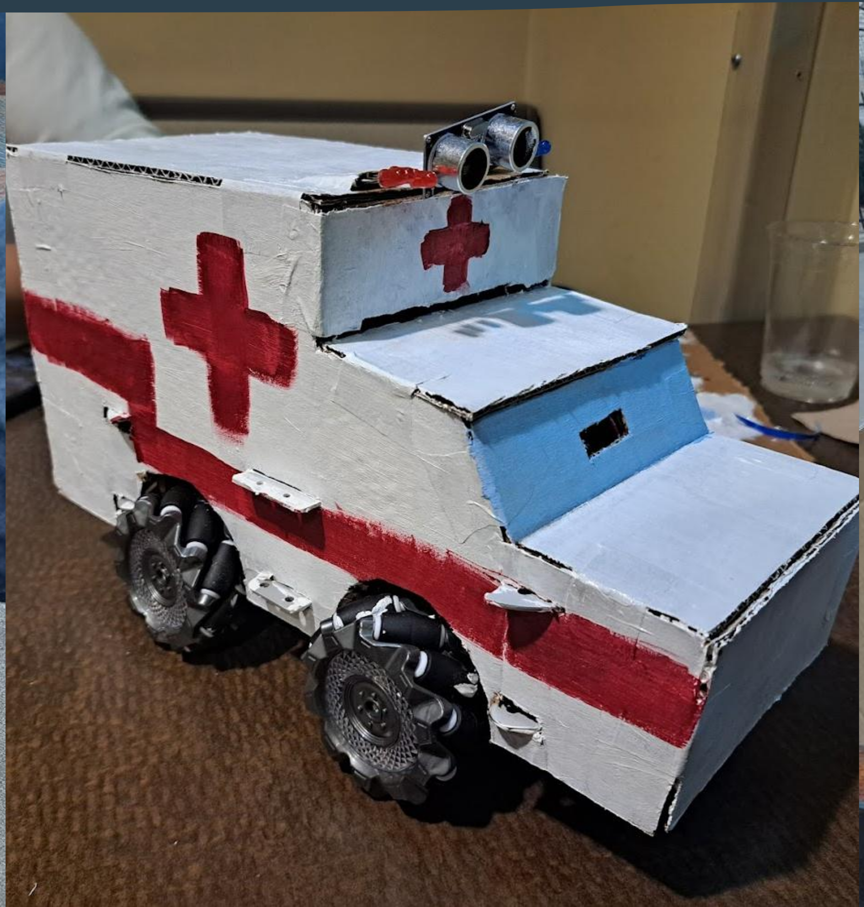


```
// Function to set motor speed (PWM)
void setSpeed(unsigned char pwm) {
  analogWrite(Lpwm_pin, pwm);
  analogWrite(Rpwm_pin, pwm);
}

// Motor control functions
void advance() {  // Move forward
  digitalWrite(pinRB, LOW);
  digitalWrite(pinRF, HIGH);
  digitalWrite(pinLB, LOW);
  digitalWrite(pinLF, HIGH);
}

void turnR() {  // Turn right
  digitalWrite(pinRB, LOW);
  digitalWrite(pinRF, HIGH);
  digitalWrite(pinLB, HIGH);
  digitalWrite(pinLF, LOW);
}

void turnL() {  // Turn left
  digitalWrite(pinRB, HIGH);
  digitalWrite(pinRF, LOW);
  digitalWrite(pinLB, LOW);
  digitalWrite(pinLF, HIGH);
}

void stopp() {  // Stop motors
  digitalWrite(pinRB, HIGH);
  digitalWrite(pinRF, HIGH);
  digitalWrite(pinLB, HIGH);
  digitalWrite(pinLF, HIGH);
}

void back() {  // Move backward
  digitalWrite(pinRB, HIGH);
  digitalWrite(pinRF, LOW);
  digitalWrite(pinLB, HIGH);
  digitalWrite(pinLF, LOW);
}
```

Overall Rough Perspective of the Ambulance

Building the Ambulance

- Jubel 225 was the working space
- Budget concerns in terms of CAD and a 2-week queue



Skills List

HARD SKILLS

- Coding
- Crimping
- Soldering
- Wiring (Breadboard)

SOFT SKILLS

- Time management
- Teamwork
- Communication