

Smart Parking Assistant

Introduction

This project is a collision avoidance system placed in the garage to help people park their cars safely without hitting their garage walls. For this project, I am using an ultrasonic sensor to calculate the car's distance from the garage wall and displaying them with green, yellow, and red LEDs to indicate the presence of the car, cautioning the car, letting the car know it is in a danger zone, respectively.

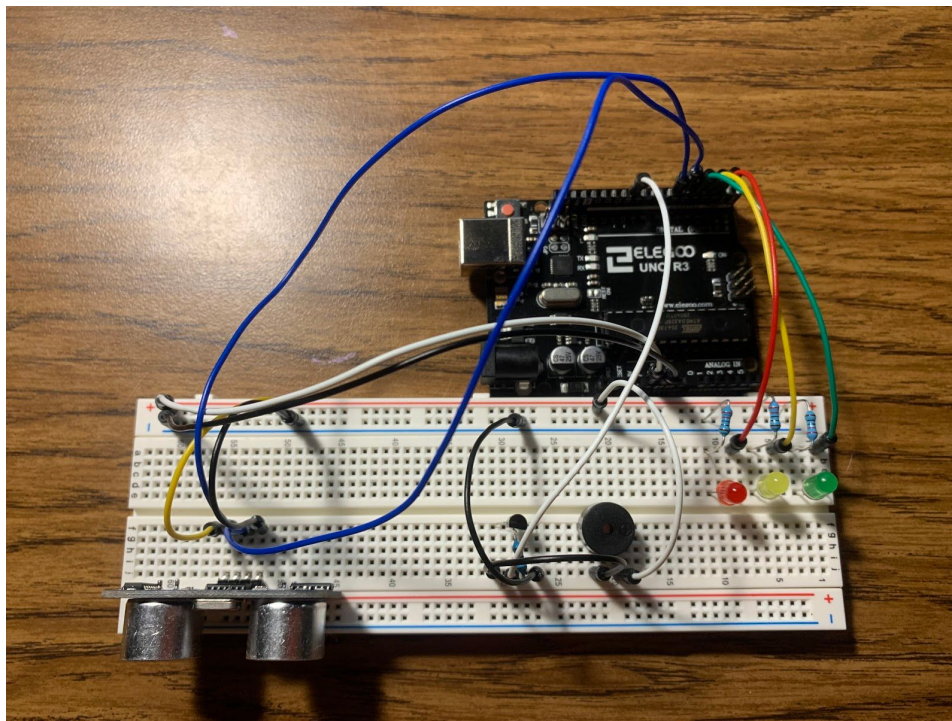
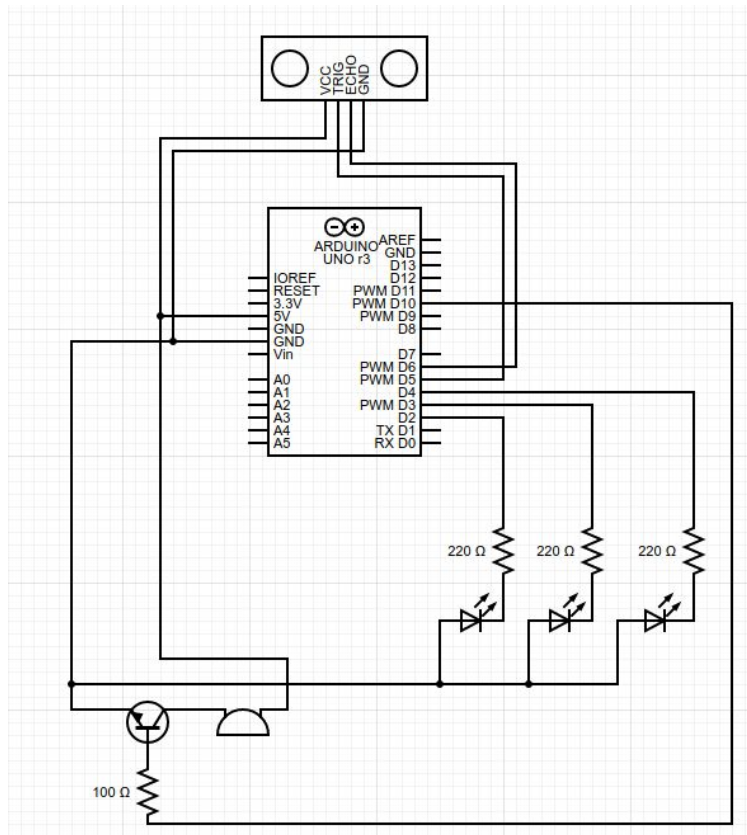
Logic

There are three phases to the project; (1) wait for the car, (2) no car in the garage, and (3) the car has parked and stopped moving. Phase 1 involves the ultrasonic sensor to look for a moving car within its proximity. If an object is detected, then one of the LEDs will turn on based on how far the object is. If the object is way too close, the buzzer will sound along with the red LED. Phase 2 is when there is no car in the garage, no object within the sensor proximity, the LEDs will turn off. Phase 3 is when the car has stopped moving, but is still within the proximity, then the sensor will wait for 15 CPU cycles and then turn off all the LEDs as well as the buzzer if it is on.

Hardware requirements

- Arduino UNO R3 Controller
- Ultrasonic Sensor
- Passive Buzzer
- 3 x Light Emitting Diode (Red, Green, and Yellow)
- 3 x 220Ω resistor for the LEDs
- NPN transistor
- 1 x 100Ω resistor
- Breadboard jumper wires

Assembly



Assembly Explanation

Red, yellow, and green LEDs' positive ends are connected to digital pins 2, 3, and 4, respectively, and 220 Ω resistors are placed between the LEDs' negative end and the arduino's ground pin. The trig pin of the sensor is connected to digital pin 5, and the echo pin is connected to digital pin 6. The buzzer is connected to digital pin 10 via an NPN transistor with the 100 Ω resistor. All the positive pins are connected to the 5V pin of the arduino, and all the remaining negative pins are connected to the ground pin of the arduino.

Code

```
int redLED = 2;    // Red LED connected to pin 2
int yellowLED = 3; // Yellow LED connected to pin 3
int greenLED = 4;  // Green LED connected to pin 4
int trigPin = 5;   // Sensor Trig pin connected to pin 5
int echoPin = 6;   // Sensor Echo pin connected to pin 6
int buzzer = 10;   // Buzzer connected to pin 10
long TempDistance = 0; // Variable to store the temporary distance
int count = 0;     // count variable to check if the object has stopped moving

void setup() {
    Serial.begin(9600);
    // Ultrasonic sensor
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    // LEDs
    pinMode(redLED, OUTPUT);
    pinMode(greenLED, OUTPUT);
    pinMode(yellowLED, OUTPUT);

    // Buzzer
    pinMode(buzzer, OUTPUT);
}

void loop() {
    // Calculate distance in inches by reading values from sensor
    long duration, Distance;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    Distance = (duration / 2) / 74; // distance in inches
```

```

    if(count < 20) {
        if(Distance > 200) {
            turnAlloff();
        }
        if((Distance > 55) && (Distance <= 200)) {
            digitalWrite(redLED, LOW);
            digitalWrite(yellowLED, LOW);
            digitalWrite(greenLED, HIGH);
            analogWrite(buzzer, 0);
        }
        if((Distance > 15) && (Distance <= 55)) {
            digitalWrite(redLED, LOW);
            digitalWrite(yellowLED, HIGH);
            digitalWrite(greenLED, LOW);
            analogWrite(buzzer, 0);
        }
        if(Distance <= 15) {
            digitalWrite(redLED, HIGH);
            digitalWrite(yellowLED, LOW);
            digitalWrite(greenLED, LOW);
            analogWrite(buzzer, 0);
        }
        if(Distance < 8) {
            analogWrite(buzzer, 500);
        }
    }

    if((Distance == TempDistance) || ((Distance + 1) == TempDistance) ||
((Distance - 1) == TempDistance)) {
        if(count >= 20) { // turn off after 20 cycles
            Serial.println("No movement detected.");
            turnAlloff();
        }else {
            count++;
            Serial.println(count);
        }
    }else {
        count = 0;
    }
    TempDistance = Distance;
    Serial.print(Distance);
    Serial.println(" inches");
    Serial.print("Count: ");
    Serial.println(count);
    delay(500);
}

```

```
// Function to turn off the LEDs and the buzzer
void turnAllOff() {
    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, LOW);
    digitalWrite(yellowLED, LOW);
    noTone(buzzer);
}
```

Code Explanation

We start by defining our constants and the global variables which will be used throughout the code. We then define each pin mode in the setup section, we then create a function `turnAllOff()`, which will be responsible for turning off all the LEDs and the buzzer.

Two things are happening inside the `loop()`, first we want to calculate the distance between the car and the sensor, in inches, by reading the values from the sensor. Then, we start checking the values of each distances; if the distance is greater than 200 inches, there's nothing in the garage, if the distance between 200 inches and 55 inches, we turn on the green LED, if the distance is between 55 and 15 inches, we turn on the yellow LED, if the distance is less than or equal 15 inches, we turn on the red LED, and when the distance is less than 8 inches, we turn on the buzzer along with red LED.

After that, we want to check to see if the car is still moving or not. And we do this by setting a counter whose values are based on the car's movement inside the garage, this will in turn decide when to turn off the LEDs. It will compare the value of "Distance" with a "TempDistance" and if the values are the same, this indicates that the car has stopped moving, therefore we increment the counter. If the car moves at any-time, we reset the counter to 0. Then finally, we set the "TempDistance" equal to the value of "Distance". We also set the counter's limit to 20 CPU cycles.

Demo