

Name:Prasanna Golda.J

Rollno: 950421104037

Subjecttopics: InternetofThings

Date:26/10/2023

Projectsubmission:Phase4development part2

SMARTPARKING:

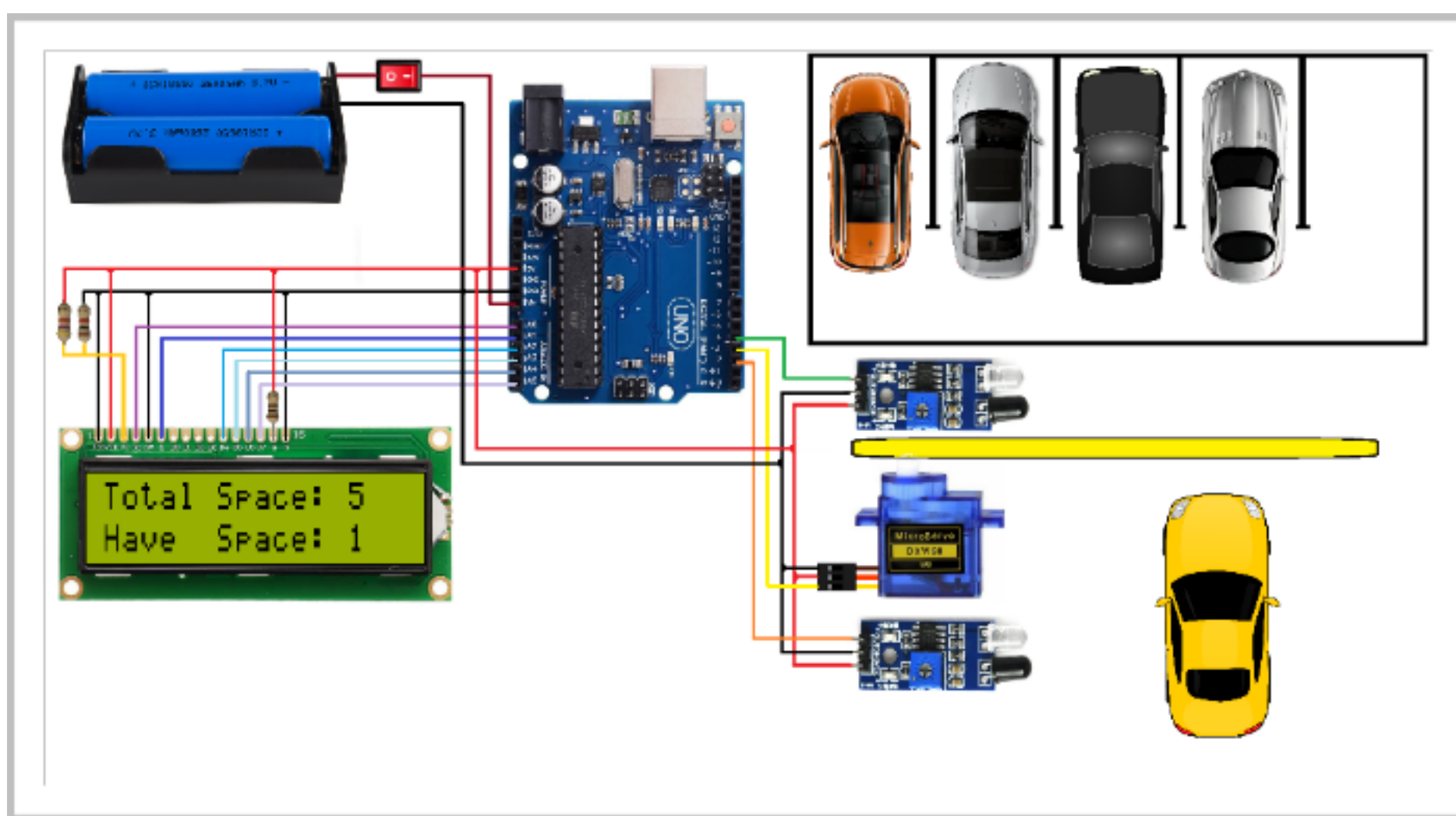
Introduction:

Hardware Required

- [Arduino Uno](#)
- [1602 LCD Display](#) (optional)
- [Power supply](#)
- [IR sensor](#)
- [Micro Servo motor](#)
- [Jumper cables](#)

Software Required

- Arduino IDE and WOKWI project simulation

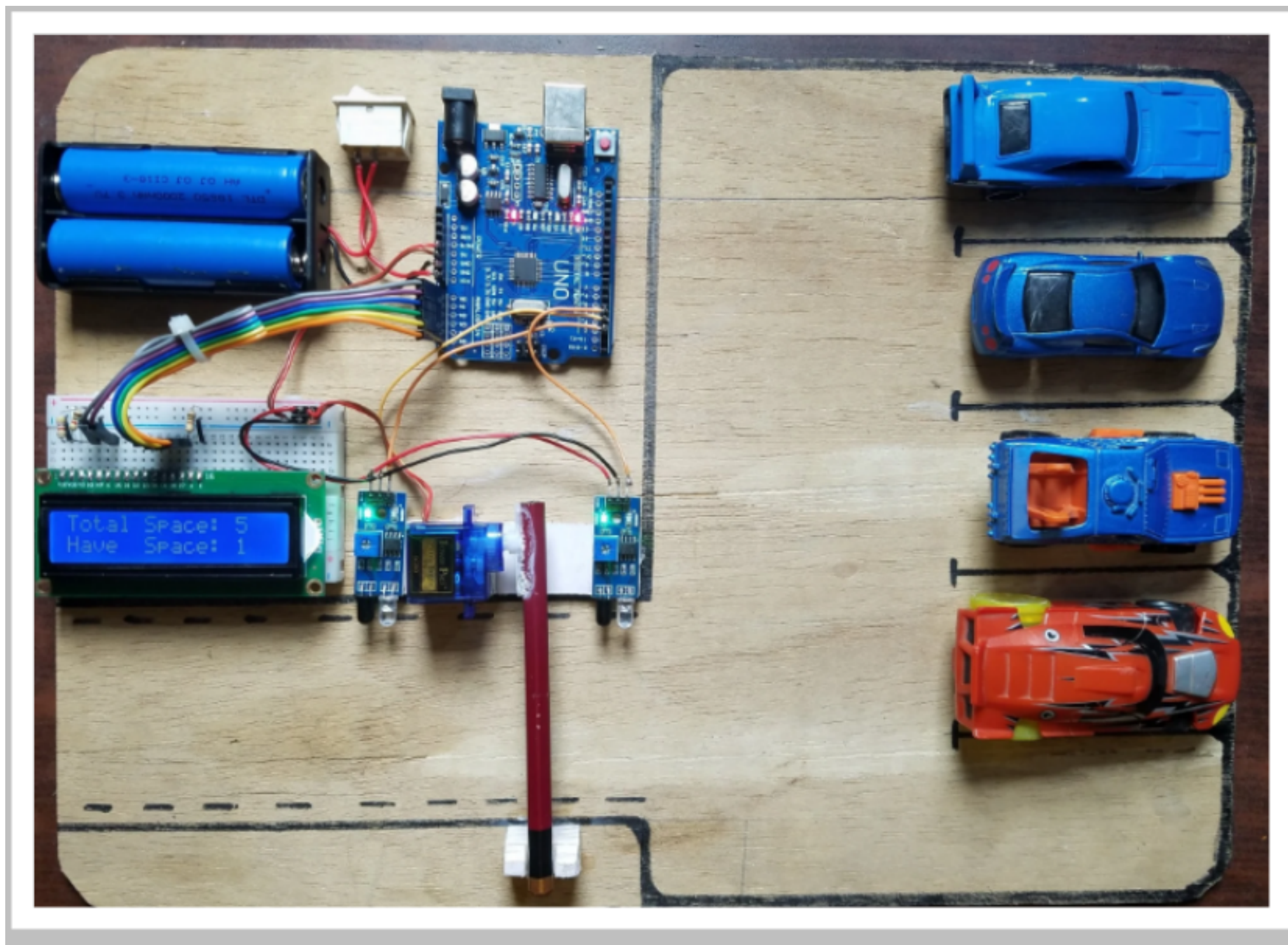


The Signal pins of both IR Sensors are connected to the Arduino Digital Pins 2 and 4. The Signal Pin of the Servo is connected to the digital pin 3.

- Connect the positive terminal of the power supply to VIN on the Arduino and the negative terminal to GND.
- This completes the circuit diagram for the car parking system.

Working Principle

Working Principle

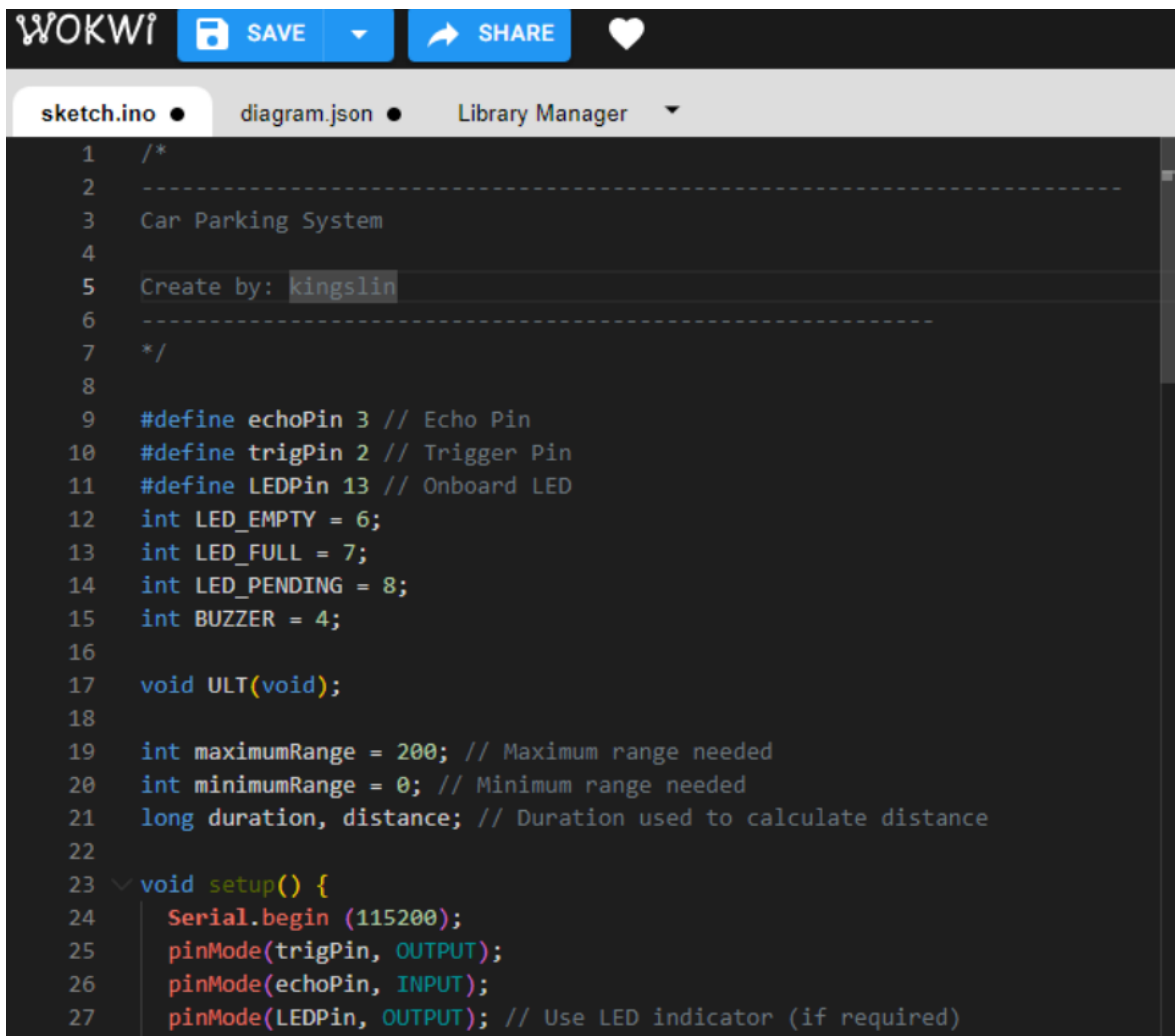


- The working concept of this involves 4 components: IR Sensor, Arduino board, Servomotors, and the LCD Display.
- The IR sensors are continuously scanning both sides of the crossing for cars so they can give an alert when the car is either coming or leaving.
 - As soon as the car approaches a crossing from either side, the command is sent to the Arduino board. The Arduino board upon receiving the command gives out the

signal to the servo to open the crossing.

- The Arduino then gives out the command to the LCD Display to either increase or decrease the number of empty spaces.
- The whole process gets started again. This completes the working concept of this project.

Arduino Code & Wokwi testing code



The screenshot shows the Wokwi IDE interface. At the top, there's a dark header with the 'WOKWI' logo, a 'SAVE' button, a 'SHARE' button, and a heart icon. Below the header, there's a tab bar with 'sketch.ino' (selected), 'diagram.json', and 'Library Manager'. The main area displays the following code:

```
1  /*
2  -----
3  Car Parking System
4
5  Create by: kingslin
6  -----
7  */
8
9  #define echoPin 3 // Echo Pin
10 #define trigPin 2 // Trigger Pin
11 #define LEDPin 13 // Onboard LED
12 int LED_EMPTY = 6;
13 int LED_FULL = 7;
14 int LED_PENDING = 8;
15 int BUZZER = 4;
16
17 void ULT(void);
18
19 int maximumRange = 200; // Maximum range needed
20 int minimumRange = 0; // Minimum range needed
21 long duration, distance; // Duration used to calculate distance
22
23 void setup() {
24     Serial.begin (115200);
25     pinMode(trigPin, OUTPUT);
26     pinMode(echoPin, INPUT);
27     pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
```

```

28     pinMode(LED_EMPTY, OUTPUT);
29     pinMode(LED_FULL, OUTPUT);
30     pinMode(LED_PENDING, OUTPUT);
31     pinMode(BUZZER, OUTPUT);
32 }
33
34 void loop() {
35     ULT();
36     Serial.println(distance); //show distance
37
38     /*vacant, out green light*/
39     if(distance >= 200){
40         digitalWrite(LED_EMPTY,1);
41         digitalWrite(LED_PENDING,0);
42         digitalWrite(LED_FULL,0);
43         Serial.println("Empty Space.");
44     }
45
46     /*someone is parking, out yellow light*/
47     else if(distance < 200 && distance >= 50){
48         digitalWrite(LED_EMPTY,0);
49         digitalWrite(LED_PENDING,1);
50         digitalWrite(LED_FULL,0);
51         tone(BUZZER, 800);
52         delay(100);
53         digitalWrite(LED_EMPTY,0);
54         digitalWrite(LED_PENDING,0);

```

```

56         noTone(BUZZER);
57         delay(500);
58         Serial.println("Car is going to park here or going out.");
59     }
60
61     /*occupied, out red light*/
62     else{
63         digitalWrite(LED_EMPTY,0);
64         digitalWrite(LED_PENDING,0);
65         digitalWrite(LED_FULL,1);
66         Serial.println("Car is Parking.");
67     }
68 }
69
70 void ULT(){
71     digitalWrite(trigPin, LOW);
72     delayMicroseconds(2);
73     digitalWrite(trigPin, HIGH);
74     delayMicroseconds(10);
75     digitalWrite(trigPin, LOW);
76     duration = pulseIn(echoPin, HIGH);
77
78     //Calculate the distance (in cm) based on the speed of sound.
79     distance = duration / 58.2;
80 }

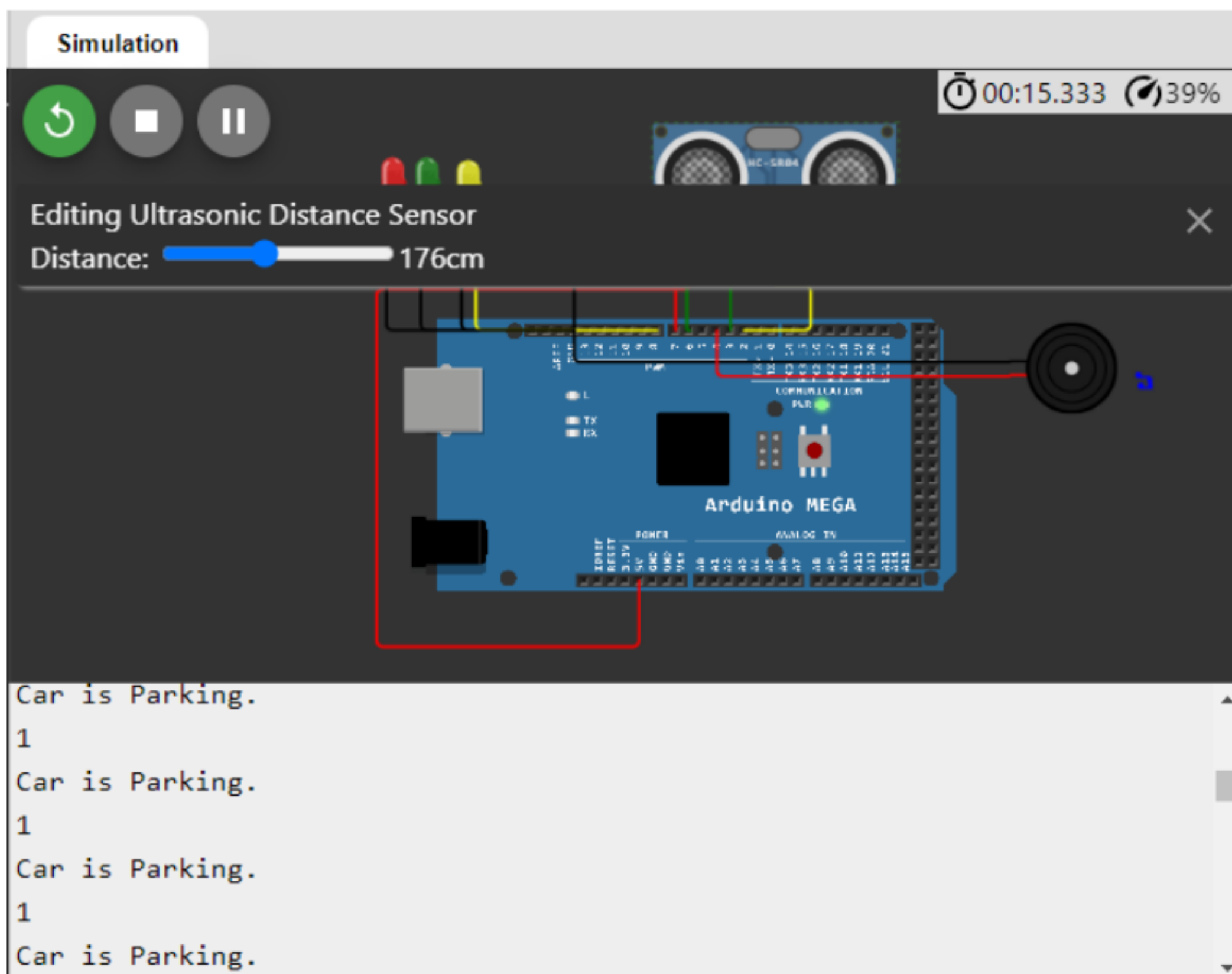
```


Simulation

00:15.333 39%

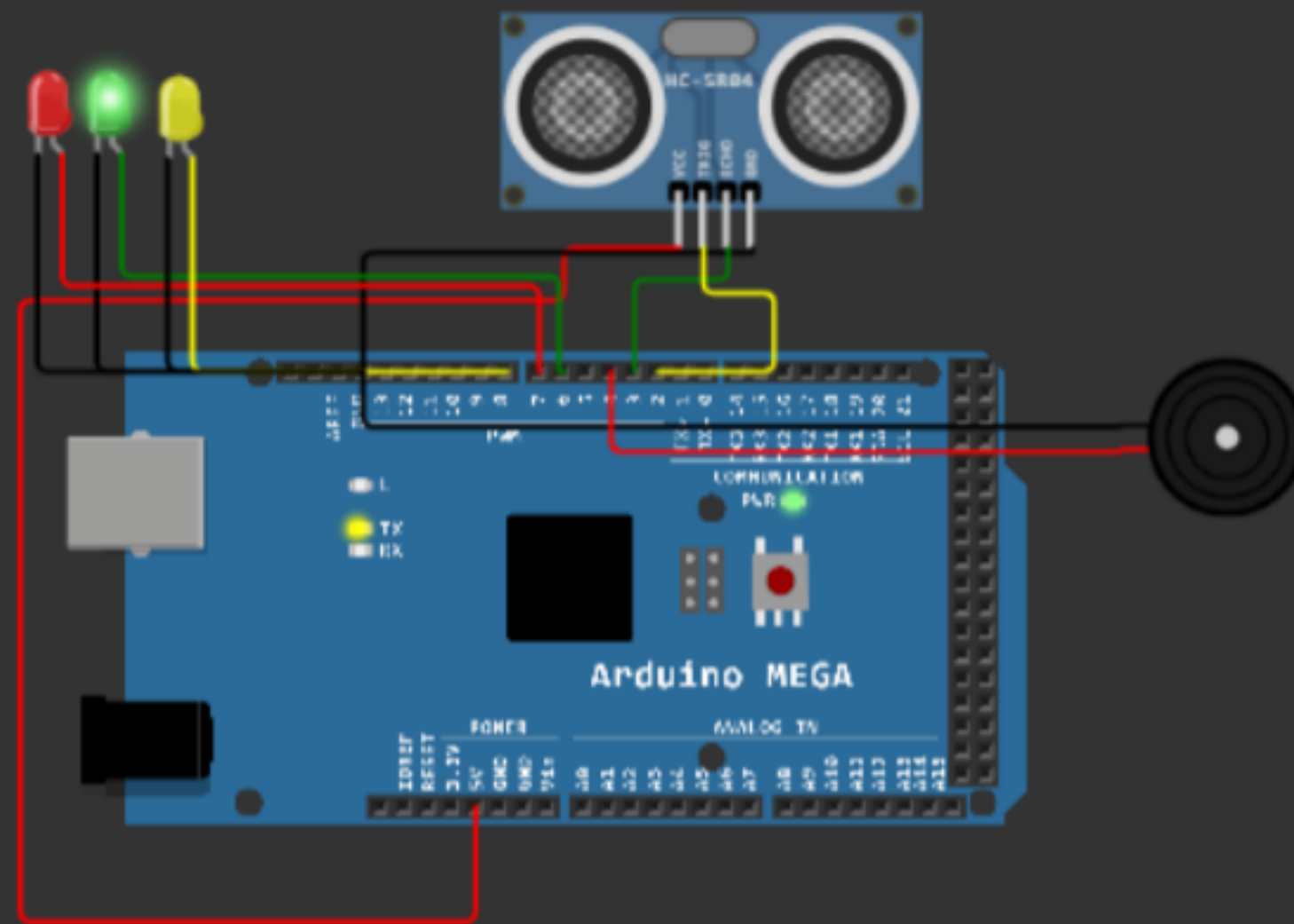
Editing Ultrasonic Distance Sensor

Distance: 176cm



Car is Parking.
1
Car is Parking.
1
Car is Parking.
1
Car is Parking.





Empty Space.

214

Empty Space.

214

Empty Space.

214

Empty Space.





The feature Benefits of Smart Parking Technology

- Optimized parking.

- Reduced traffic.
- Reduced pollution.
- Enhanced User Experience.
- Integrated Payments and POS.
- Increased Safety.
- Real-Time Data and Trend Insight.
- Decreased Management Costs.

Thephase4developmentpart2

forloTpart2(SmartParking)succesfullyvisualized&
completed.