Phase-4

Project Title : Smart Parking(Distance Measurement and RGB LED Display with ESP32)

Introduction:

This project demonstrates how to use an ultrasonic distance sensor with an ESP32 microcontroller to accurately measure distances and provide real-time feedback. The distance data is displayed on an LCD screen, and an RGB LED changes its color based on the measured distance.

Components:

- ESP32 development board
- Ultrasonic distance sensor (HC-SR04)
- Common cathode RGB LED
- Liquid Crystal Display (LCD) with I2C interface
- Jumper wires

Project Description:

The primary objective of this project is to create a distance measurement system that provides visual feedback through an RGB LED. It utilizes an ultrasonic sensor to measure distances accurately and then visually represents the data on an LCD screen. The RGB LED changes its color to convey distance information.

Functionality:

Ultrasonic Sensor: The HC-SR04 ultrasonic sensor is responsible for measuring distances. It operates on the principle of sending ultrasonic waves and measuring the time it takes for them to bounce back. The distance is calculated based on the time it takes for the wave to return.

LCD Display: The Liquid Crystal Display (LCD) with an I2C interface is used to display real-time distance measurements. It shows the measured distance in centimeters.

RGB LED: A common cathode RGB LED is controlled based on the distance measured by the sensor. The LED changes color according to the following conditions:

- If the distance is greater than or equal to 100 cm, the LED displays green to indicate a safe distance.
- If the distance is less than 100 cm but greater than 1 cm, the LED displays red to signal proximity.
- If the distance is 1 cm or less, the LED displays blue to signify extremely close proximity.
- Serial Communication: The project includes serial communication at a baud rate of 9600. This enables real-time monitoring of distance measurements through a serial monitor, such as the Arduino IDE Serial Monitor.

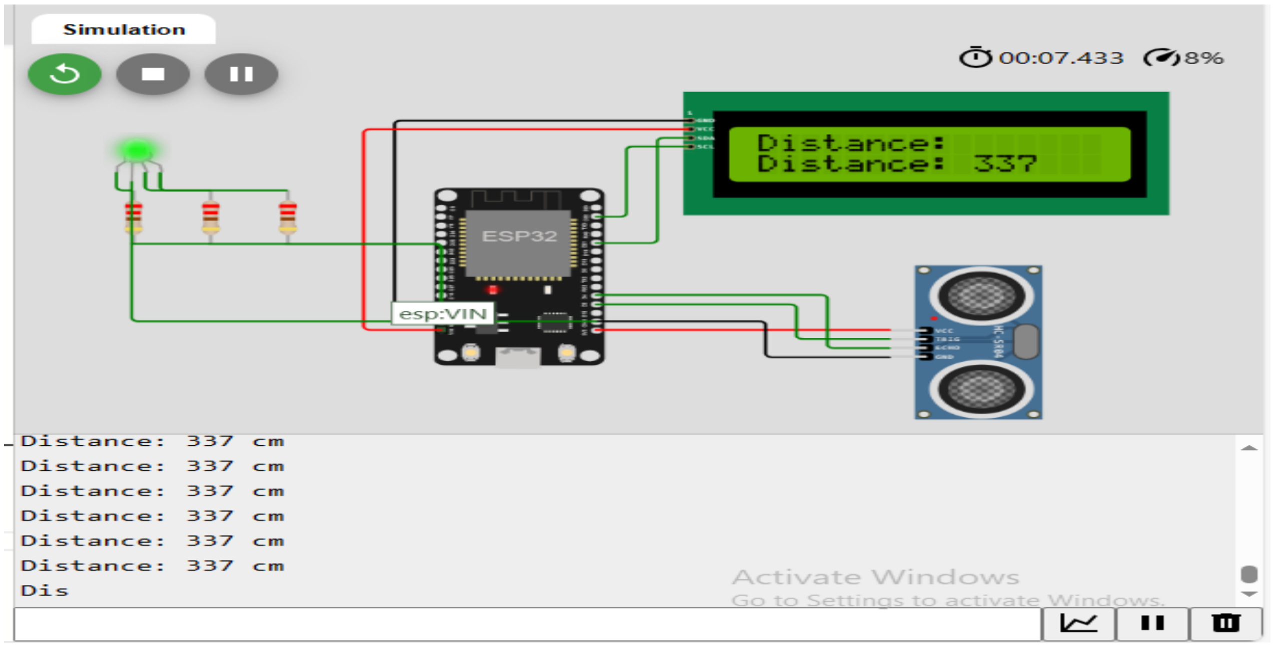
Program:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);  // Set the LCD address to 0x27 for a 16
chars and 2 line display

const int triggerPin = 2;
const int echoPin = 4;
const int redPin = 12;    // Red anode connected to GPIO 12
const int greenPin = 13;  // Green anode connected to GPIO 13
const int bluePin = 14;   // Blue anode connected to GPIO 14

void setup() {
  Serial.begin(9600);
  pinMode(triggerPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(redPin, OUTPUT);   // Set the Red LED pin as an output
  pinMode(greenPin, OUTPUT); // Set the Green LED pin as an output
  pinMode(bluePin, OUTPUT);  // Set the Blue LED pin as an output

  lcd.init();                      // Initialize the LCD
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Distance:");
}
```

```
void loop() {
  long duration, distance;
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;

  lcd.setCursor(0, 1);
  lcd.print("            "); // Clear the line
  lcd.setCursor(0, 1);
  lcd.print("Distance: ");
  lcd.print(distance);

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

  if (distance >= 100) {
    // Set the RGB LED to green
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);
  } else if (distance < 100 && distance > 1) {
    // Set the RGB LED to red
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);
  } else {
    // Set the RGB LED to blue
    analogWrite(redPin, 0);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 255);
  }
}
```

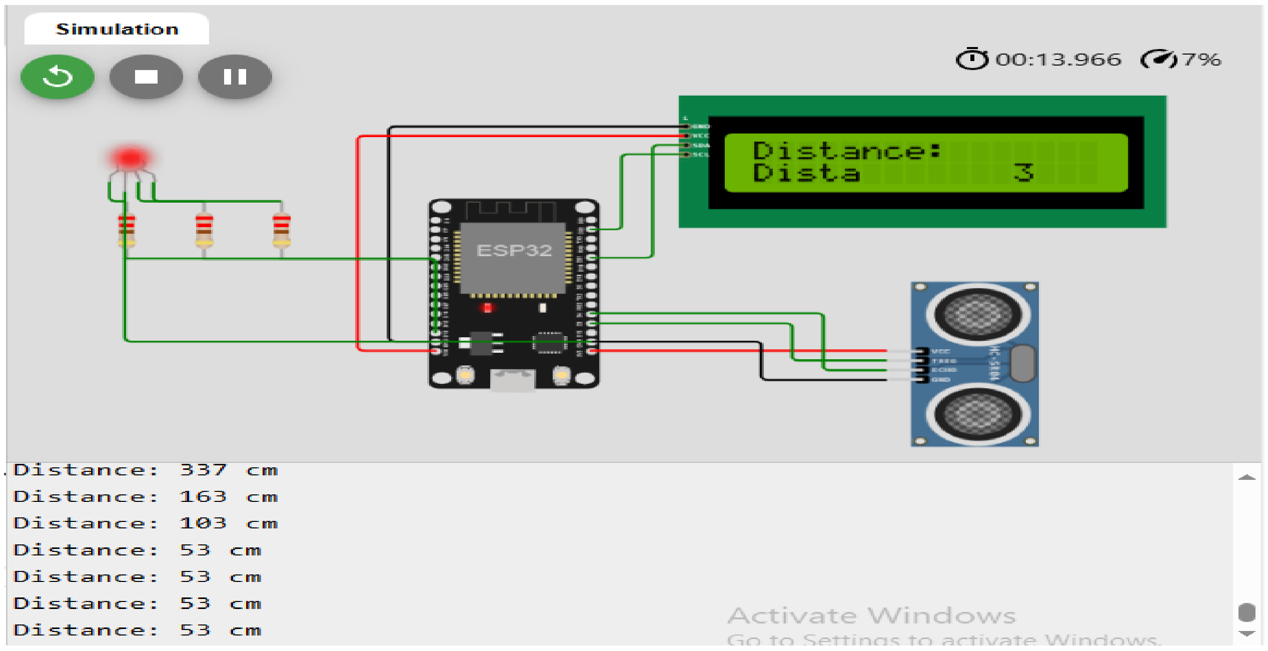
Output:

Distance >100



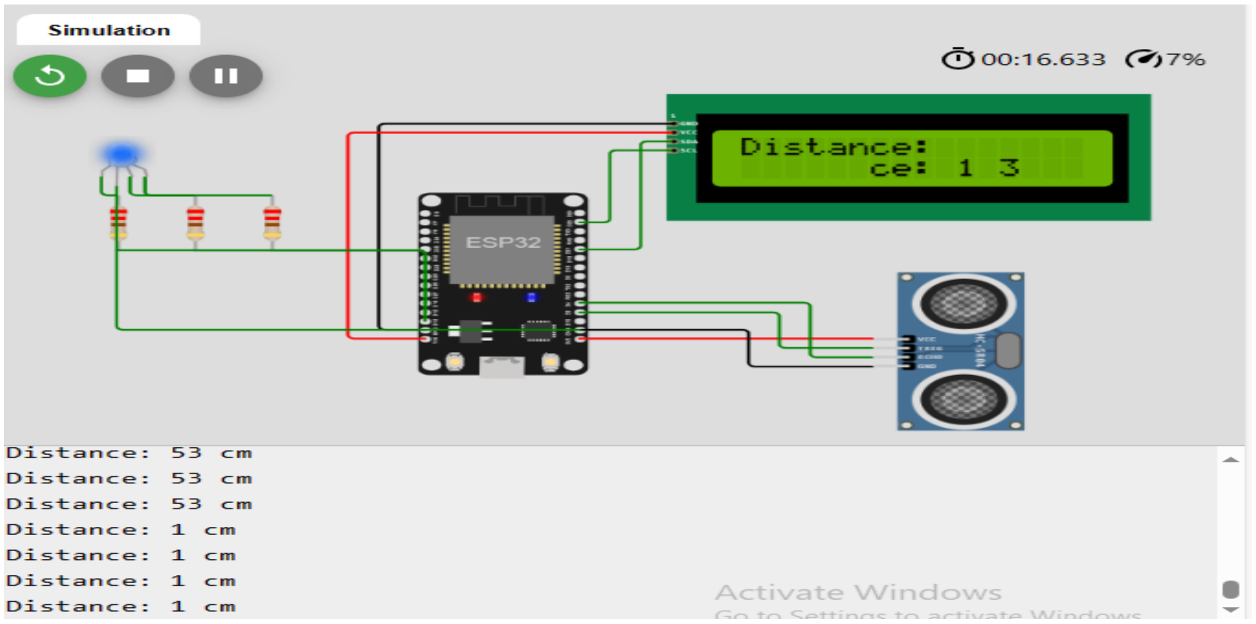
```
Distance: 337 cm
Distance: 337 cm
Distance: 337 cm
Distance: 337 cm
Distance: 337 cm
Distance: 337 cm
Dis
```

Distance <100



```
Distance: 337 cm
Distance: 163 cm
Distance: 103 cm
Distance: 53 cm
Distance: 53 cm
Distance: 53 cm
Distance: 53 cm
```

Stop car



```
Distance: 53 cm
Distance: 53 cm
Distance: 53 cm
Distance: 1 cm
Distance: 1 cm
Distance: 1 cm
Distance: 1 cm
```

More Details:

https://wokwi.com/projects/379372859103666177