FINAL CODE FOR CAR REVERSE

// Define LED pins

const int redLED = 11;

const int yellowLED = 10;

const int greenLED = 9;

// Define buzzer pin

const int buzzerPin = 3;

// Ultrasonic Sensor 1 pins

const int trigPin1 = 12;

const int echoPin1 = 13;

// Ultrasonic Sensor 2 pins

const int trigPin2 = 6;

const int echoPin2 = 7;

void setup() {

  // Initialize serial communication

  Serial.begin(9600);

  // Set LED pins as OUTPUT

  pinMode(redLED, OUTPUT);

  pinMode(yellowLED, OUTPUT);

  pinMode(greenLED, OUTPUT);

  // Set buzzer pin as OUTPUT

  pinMode(buzzerPin, OUTPUT);

  // Set ultrasonic sensor pins

  pinMode(trigPin1, OUTPUT);

  pinMode(echoPin1, INPUT);

  pinMode(trigPin2, OUTPUT);

  pinMode(echoPin2, INPUT);

}

void loop() {

  // Measure distances from both sensors

  float distance1 = measureDistance(trigPin1, echoPin1);

  float distance2 = measureDistance(trigPin2, echoPin2);

  // Take the closest distance for safety

  float closestDistance = min(distance1, distance2);

  // Show distances on Serial Monitor

  Serial.print("Sensor 1 Distance: ");

  Serial.print(distance1);

  Serial.print(" cm\t");

  Serial.print("Sensor 2 Distance: ");

  Serial.print(distance2);

  Serial.print(" cm\t");

  Serial.print("Closest Distance: ");

  Serial.print(closestDistance);

  Serial.println(" cm");

  // Call the function to activate LEDs and buzzer based on distance

  reverseParkingSystem(closestDistance);

  delay(300); // Delay to prevent too much serial spam & stabilize readings

}

// Function to measure distance using ultrasonic sensor

float measureDistance(int trigPin, int echoPin) {

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  long duration = pulseIn(echoPin, HIGH, 30000); // 30ms timeout (prevents hanging)

  // If no echo received, return a large number (no object)

  if (duration == 0) {

    return 400.0; // 400 cm (out of range)

  }

  float distance = (duration \* 0.0343) / 2; // cm

  return distance;

}

// Function to handle LEDs and buzzer based on distance

void reverseParkingSystem(float distance) {

  // First, turn off all LEDs

  digitalWrite(redLED, LOW);

  digitalWrite(yellowLED, LOW);

  digitalWrite(greenLED, LOW);

  // Decision based on distance

  if (distance > 30.0) {

    // Safe distance

    digitalWrite(greenLED, HIGH);

    noTone(buzzerPin); // No sound

  }

  else if (distance <= 30.0 && distance > 15.0) {

    // Warning zone

    digitalWrite(yellowLED, HIGH);

    // Short beep every loop (loop already has delay(300))

    tone(buzzerPin, 1000); // 1000Hz frequency

    delay(500);            // beep duration

    noTone(buzzerPin);     // silence for the rest of loop delay

  }

  else if (distance <= 15.0) {

    // Danger zone

    digitalWrite(redLED, HIGH);

    // Continuous loud beep

    tone(buzzerPin, 1000); // 2000Hz frequency continuous tone

  }

}