Smart Vehicle Parking System with Arduino and Ultrasonic Sensor

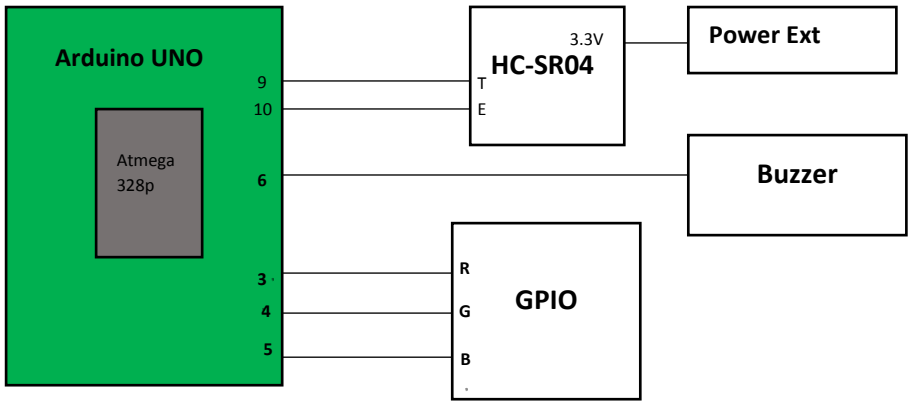
1. Aim

To design and implement an automated parking system using an ultrasonic sensor and an Arduino Uno board to detect vehicles and indicate the availability of parking spaces.

2. Components Required

- Arduino Uno (Microcontroller)
- Ultrasonic Sensor (HC-SR04) (For distance measurement)
- LEDs (Red & Green) (For parking status indication)
- Buzzer (For alerting when space is occupied)
- Resistors & Jumper Wires (For connections)
- Breadboard (For prototyping)

3. Circuit Diagram & Working Principle



3.1 Circuit Connections

- VCC of HC-SR04 → 5V on Power Extension Module
- GND of HC-SR04 → GND on Arduino
- Trigger Pin of HC-SR04 → Digital Pin 9 on Arduino
- Echo Pin of HC-SR04 → Digital Pin 10 on Arduino
- Red LED → Digital Pin 3 (Indicates parking occupied)
- Green LED → Digital Pin 4 (Indicates parking available)
- Blue LED → Digital Pin 5 (Indicates parking available)
- Buzzer → Digital Pin 6

3.2 Working

The ultrasonic sensor continuously measures the distance between the sensor and any approaching vehicle. If a vehicle is detected (distance <10 cm), the red LED turns ON, and the buzzer sounds

an alert, indicating the parking space is occupied. If no vehicle is detected (distance > 10 cm), the green LED turns ON, indicating the parking space is available.

4. Arduino Code

```
#define TRIG_PIN 9
#define ECHO_PIN 10
#define BUZZER_PIN 6
#define GREEN_LED 4
#define BLUE_LED 5
#define RED_LED 3

void setup() {
  Serial.begin(9600);
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
pinMode(BUZZER_PIN, OUTPUT);
pinMode(GREEN_LED, OUTPUT);
pinMode(BLUE_LED, OUTPUT);
pinMode(RED_LED, OUTPUT);
}

void loop() {
  long duration;
  float distance;

  // Trigger the ultrasonic pulse
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

  // Read the Echo pin
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;  // Convert time to distance (cm)

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

  // Distance-based alert system
  if (distance > 10) {  // Safe Distance (No alert)
digitalWrite(GREEN_LED, HIGH);
digitalWrite(BLUE_LED, LOW);
digitalWrite(RED_LED, LOW);
noTone(BUZZER_PIN);  //  Stop the buzzer completely
  }
  else if (distance <= 10 && distance > 5) {  // Caution
digitalWrite(GREEN_LED, LOW);
digitalWrite(BLUE_LED, HIGH);
digitalWrite(RED_LED, LOW);
```

```
tone(BUZZER_PIN, 1000, 200);  // Short beep
delay(500);
noTone(BUZZER_PIN);  // Ensure buzzer turns off after the beep
  }
  else if (distance <= 10) {  // Danger - Stop
digitalWrite(GREEN_LED, LOW);
digitalWrite(BLUE_LED, LOW);
digitalWrite(RED_LED, HIGH);
tone(BUZZER_PIN, 1000);  // Continuous beep
  }

delay(200);
}
```

5. Observations

Test Condition	Ultrasonic Sensor Reading	LED Status	Buzzer Status
Safe Distance	Distance > 10 cm	Green LED ON	OFF
Caution	Distance<=10 && >5	Blue LED ON	BEEP
Danger-Stop	Distance < 10 cm	Red LED ON	ON

- The ultrasonic sensor accurately detects the presence of a vehicle in the parking space.
- The red LED and buzzer activate when a vehicle is present, and the green LED turns ON when the space is available.
- The system effectively updates in real-time based on sensor readings.

9. Conclusion

The Arduino-based parking system using an ultrasonic sensor successfully detects vehicles and indicates parking space availability through LEDs and a buzzer. This project can be expanded for multiple parking slots and integrated with IoT-based cloud monitoring for smart city applications.