

BATCH-6

PROJECT NAME:OCCUPANCY DETECTION SYSTEM

ROLL NUMBERS:

21781A0407- A. TEJASWINI

21781A0431- B.SAI PRATHYUSHA

21781A0432- B.KARTHIK

21781A0447- C.BHAVANA

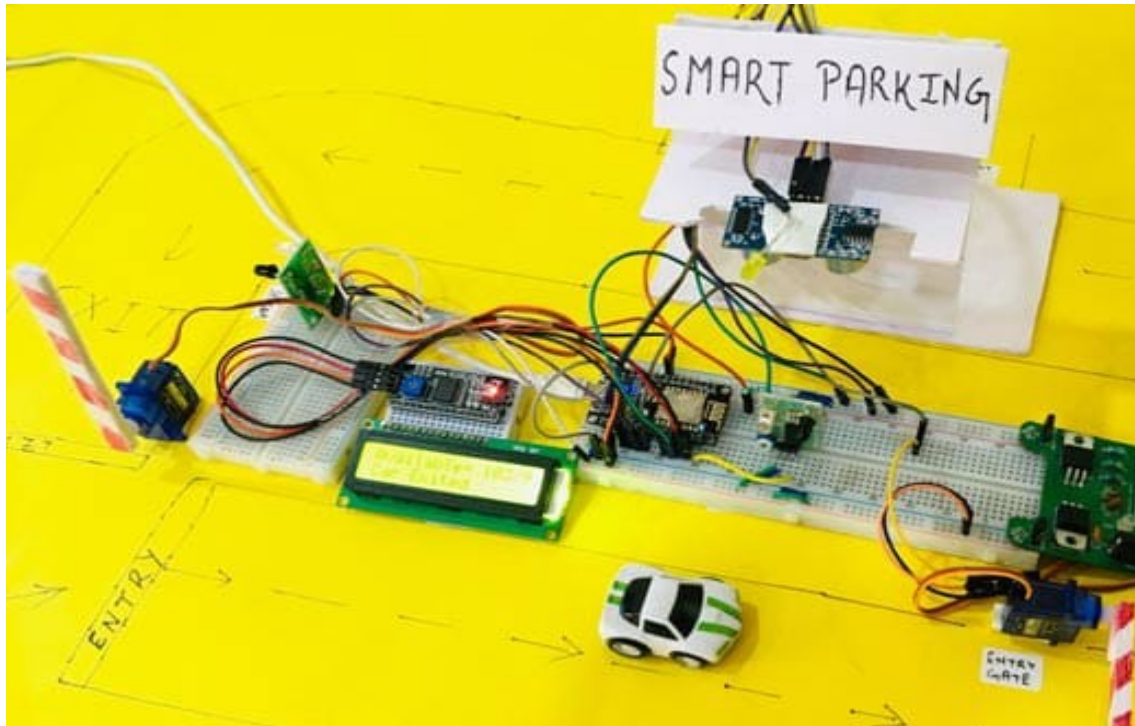
IOT BASED OCCUPANCY DETECTION SYSTEM FOR CAR PARKING USING ESP8266

PROBLEM STATEMENT:

With an increment of cars, the problem of car parking is increasing day by day. Therefore it has now become extremely difficult to find a parking slot. To keep monitoring parking slots under check especially in towns and cities. We have proposed occupancy detection system for car parking using ESP8266 over IOT to achieve this goal.

SCOPE OF SOLUTIONS:

The solution involves setting up sensors in parking spots that detect if a spot is taken or free. It continuously checks and updates the status in real-time. When a spot changes status, like when a car arrives or leaves, it sends an instant notification to alert you. You can easily check parking availability from your phone or computer. It's designed to be affordable, easy to use, and adaptable to different parking sizes. Overall, it's a practical way to manage parking effectively, ensuring you always know where to find an available spot.



REQUIRED COMPONENTS TO DEVELOP SOLUTION:

1. ESP8266 module
2. Passive interfaed sensors(PIR)
3. Web or Mobile Applications
4. Power supply
5. Breadboard
6. Computer with arduino IDE
7. Internet connectivity
8. Basic tools
9. Connecting wires
10. MQTT Broker software
11. Text Editor
12. Serial monitor

DESCRIPTION OF BLOCK DIAGRAM:

Certainly! Here's a simplified description of the block diagram for an occupancy detection system for car parking using ESP8266:

1. *Sensor:*

- Detects if a car is present in a parking space. Common sensors include infrared or ultrasonic sensors.

2. *ESP8266 (Microcontroller):*

- Acts as the brain of the system.
- Collects information from the sensor.
- Makes decisions based on whether a parking space is occupied or vacant.

3. *Wi-Fi Connection:*

- Allows the ESP8266 to communicate with the outside world.
- Enables sending data to a server or cloud platform.

4. *Server/Cloud:*

- Receives information from ESP8266.
- Processes and stores data about parking space occupancy.

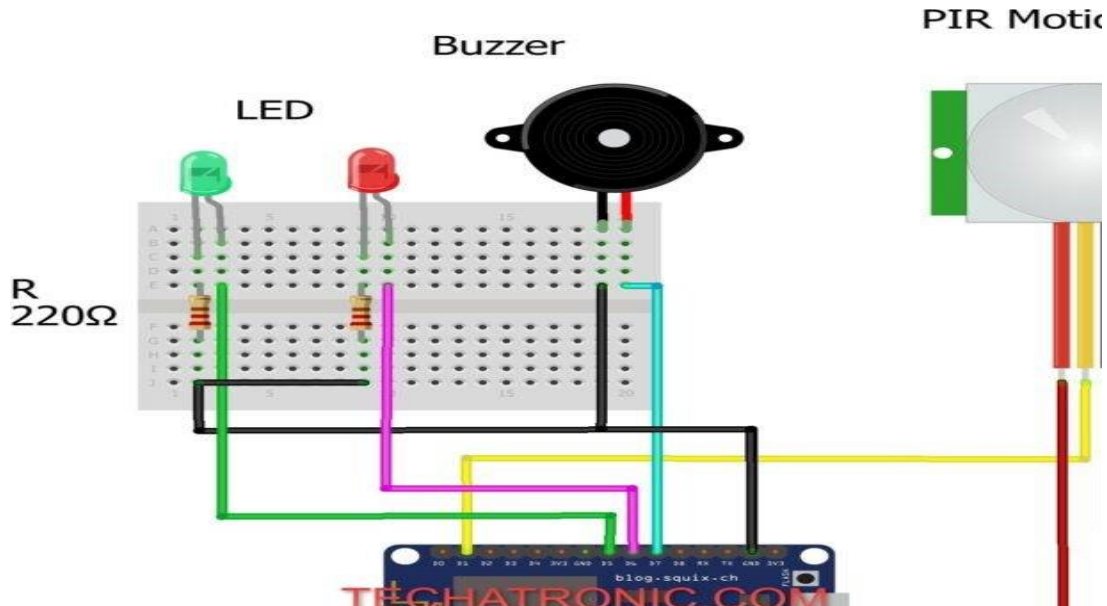
5. *User Interface:*

- A web or mobile app that users can access.
- Displays real-time parking space availability information.

6. *LED Indicator:*

- Provides on-site visual feedback.
- Indicates whether a parking space is occupied or vacant.

This system allows users to check parking availability remotely through an app while providing on-site feedback through LED indicators. The ESP8266 is the central controller ensuring smooth communication between the sensor, cloud, and user interface.



PROGRAM:

```
#include <ESP8266WiFi.h>
```

```
const char* ssid = "yourSSID";
```

```
const char* password = "yourPASSWORD";
```

```
const int sensorPin = D1; // Replace with the actual GPIO pin connected to the sensor
```

```
const int ledPin = D2; // Replace with the actual GPIO pin connected to the LED indicator
```

```
void setup() {
```

```
    pinMode(sensorPin, INPUT);
```

```
    pinMode(ledPin, OUTPUT);
```

```
    Serial.begin(115200);
```

```
    WiFi.begin(ssid, password);
```

```

while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
}

void loop() {
    int sensorValue = digitalRead(sensorPin);
    if (sensorValue == HIGH) {
        // Parking space occupied
        digitalWrite(ledPin, HIGH);
        sendDataToServer("occupied");
    } else {
        // Parking space vacant
        digitalWrite(ledPin, LOW);
        sendDataToServer("vacant");
    }

    delay(1000); // Adjust the delay based on your project requirements
}

void sendDataToServer(String status) {
    // Implement code to send data to your server or cloud platform
    // Use HTTP requests or MQTT, depending on your setup
    // You may need to include additional libraries based on your communication protocol
}

```

CONCLUSION:

We conclude that the ESP8266-based parking occupancy system efficiently manages parking by providing real-time updates on availability and sending alerts when spots change status. With remote access via web or mobile apps, users can check parking status from anywhere. It's secure, cost-effective, and easy to set up. This solution simplifies parking management, ensuring a smoother experience for drivers and attendants.