IOT BASED SMART PARKING SYSTEM

A project report submitted in partial fulfilment
Of the requirements for the mark of B.Tech in
Information technology

By

A. NAGADEEPA (513221205009)

Under the supervision of

Professor & HOD
Department of Information Technology.

SMART PARKING SYSTEM

Phase 4: Development part 2

Smart Parking: Enhancing Efficiency

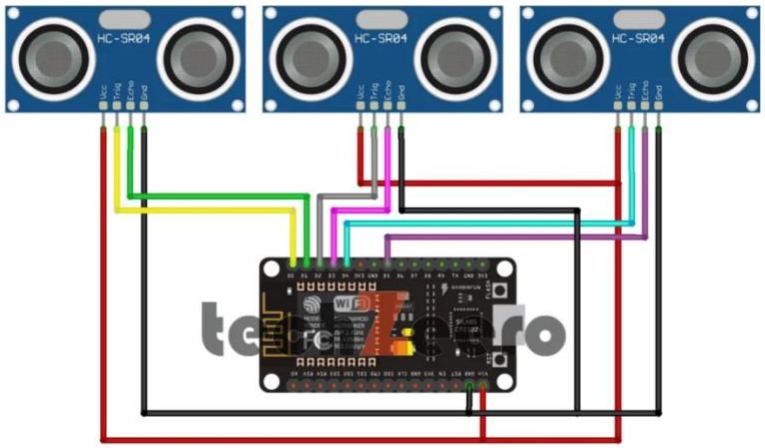
Abstract:

The field of smart parking is growing rapidly and has many potential applications in densely populated urban areas. Internet of Things (IoT) and Arduino-based smart parking systems offer an innovative solution to solve parking problems in real time. Among other things, these systems reduce traffic congestion, improve road safety, reduce greenhouse gas emissions, and improve user experience.

In this paper, we present a smart parking system that uses an innovative architecture and algorithm to optimize vehicle parking. The system also tracks cars, their location, entry and exit details, and account management. An infrared sensor is integrated into the system to monitor the entry and exit of vehicles. This sensor uses beams of light to detect the presence of vehicles in a specific parking area, allowing for efficient use of available parking space.

This project has many benefits, such as optimized space utilization, reduced congestion and pollution, and an improved user experience. The system can be deployed in a variety of environments, including underground parking garages, shopping malls, airports, and train stations.

Circuit Diagram for Smart Parking System



Code for Smart Parking

```
//Code from techzeero.com

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#define FIREBASE_HOST "smart-parking-007.firebaseio.com"  //Your Firebase Project URL goes here withou
#define FIREBASE_AUTH "hp4aQl2MtBZ9aZ8NmQwyRDtjhBCcupJYJmrEiHom"   //Your Firebase Database Secret goe

#define WIFI_SSID "techzeero"                    //your WiFi SSID(Hotspot Name) for which your NodeMCU c
#define WIFI_PASSWORD "techzeero123"             //Password of your wifi network

#define trigPin1 16   //D0 (We Used GPIO Pin Number)
#define echoPin1 5    //D1
#define trigPin2 4    //D2
#define echoPin2 0    //D3
#define trigPin3 2    //D4
#define echoPin3 14   //D5

void setup()
{
  Serial.begin(115200);            // Select the same baud rate if you want to see the datas on Serial
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(trigPin3, OUTPUT);
```

```
  pinMode(echoPin3, INPUT);

  WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status()!=WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected:");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH);
}

void firebasereconnect()
{
  Serial.println("Trying to reconnect");
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop()
{
  if (Firebase.failed())
  {
    Serial.print("setting number failed:");
    Serial.println(Firebase.error());
    firebasereconnect();
    return;
  }

  //Ultrasonic Sensor 1

  digitalWrite(trigPin1, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  long duration1 = pulseIn(echoPin1, HIGH);
  long inch1 = duration1 / 74 / 2;         // In Inch

  if(inch1 <=2)
  {
    Firebase.setInt("U1",1);
    Serial.println("U1 = 1");
  }
  else
  {
     Firebase.setInt("U1",0);
     Serial.println("U1 = 0");
  }
```

```
  //Ultrasonic Sensor 2

  digitalWrite(trigPin2, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);
  long duration2 = pulseIn(echoPin2, HIGH);
  long inch2 = duration2 / 74 / 2;         // In Inch

  if(inch2 <=2)
  {
    Firebase.setInt("U2",1);
    Serial.println("U2 = 1");
  }
  else
  {
     Firebase.setInt("U2",0);
     Serial.println("U2 = 0");
  }

  //Ultrasonic Sensor 3

  digitalWrite(trigPin3, LOW);
  delayMicroseconds(10);
  digitalWrite(trigPin3, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin3, LOW);
  long duration3 = pulseIn(echoPin3, HIGH);
  long inch3 = duration3 / 74 / 2;         // In Inch

  if(inch3 <=2)
  {
    Firebase.setInt("U3",1);
    Serial.println("U3 = 1");
  }
  else
  {
     Firebase.setInt("U3",0);
     Serial.println("U3 = 0");
  }
}
```

2. Materials

2.1 Materials

2.1.1 Arduino UNO Card

The Arduino UNO card is the brain of the smart parking project. It is used to control the whole system, collecting data from the sensors, and sending signals to the entrance gate to open or close the access. The Arduino UNO board is also used to display the parking information in real time on the LCD screen.

2.1.2 Infrared sensor

Infrared sensors are one of the key components of the smart parking project. They are installed at the entrance and exit of the parking lot to detect vehicles and indicate whether a parking space is available or not. The infrared sensors send a signal to the Arduino UNO board every time a vehicle passes by. The Arduino UNO board then uses these signals to update the display on the LCD screen and to control the entrance gate.

2.1.3 Micro servo motor

The micro servo motor is used to control the entrance barrier. It is connected to the Arduino UNO board and opens or closes according to the signals sent by the board. The micro servo motor is an ideal choice for this project because it is small, easy to use and can be controlled precisely.

2.1.4 LCD display

The LCD screen is used to display realtime parking information to drivers. It is connected to the Arduino UNO board and can display the number of available spaces, the number of occupied spaces and other important information. The LCD display is a key part of the project as it allows drivers to quickly know whether it is possible to park or not, which improves traffic flow and reduces congestion.