**Smart parking system using IoT**

***Mini*** ***Project*** ***Report*** ***submitted*** ***in*** ***partial*** ***fulfillment.***

***of*** ***the*** ***requirement*** ***for*** ***the*** ***degree*** ***of***

**T.** **E.** **(Information** **Technology)**

Submitted By

* **Sankalp Rane(19101B0064)**
* **Nilabh Waingankar(19101B0021)**
* **Vrushali Gore(19101A0037)**

Under the Guidance of

Prof. **Vinita Bhandiwad**

Department of Information Technology





Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

University of Mumbai

2021-22

CERTIFICATE OF APPROVAL

**For**

**Mini** **Project** **Report**

This is to Certify that

* **Sankalp Rane(19101B0064)**
* **Nilabh Waingankar(19101B0021)**
* **Vrushali Gore(19101A0037)**

Have successfully carried out Mini Project entitled

**Smart parking system using IoT**

In partial fulfillment of degree course in

Information Technology

As laid down by University of Mumbai during the academic year 2021-22

Under the Guidance of

“Prof. **Vinita Bhandiwad** ”

**Dr.Vipul Dalal**

Signature of Guide Head of Department

Examiner 1 Examiner 2 Principal

Dr. S. A. Patekar

**ACKNOWLEGEMENT**

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We express our profound gratitude we give to our **Prof.Vinita Bhandiwad** Ma’am, our respectable project guide, for her gigantic support and guidance. Without her counseling our project would not have seen the light of the day.

We extend our sincere thanks to **Dr. Vipul Dalal**, Head of the Department of

Information Technology for offering valuable advice at every stage of this undertaking.

We would like to thank all the staff members who willingly helped us. We are grateful to VIDYALANKAR INSTITUTE OF TECHNOLOGY for giving us this opportunity.

The days we have spent in the institute will always be remembered and also be

reckoned as guiding in our career.

**Sankalp Rane(19101B0064)**

**Nilabh Waingankar(19101B0021)**

**Vrushali Gore(19101A0037)**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Sr. No. | Topic | Page No. |
| 1 | Introduction |  |
| 2 | Aim & Objectives |  |
| 3 | Problem Definition |  |
| 4 | Proposed System  4.1 Block Diagram  4.2 Flow Chart |  |
| 5 | Components  5.1 Hardware 5.2 Software |  |
| 6 | Project Architecture |  |
| 7 | Code |  |
| 8 | Implementation  8.1 Working  8.2 Circuit Diagram |  |
| 9 | Results |  |
| 10 | Conclusion & Future Scope |  |
|  | IEEE technical paper on project topic |  |
|  | Tinker cad experiment |  |
|  | Technical poster on Project topic |  |
|  | GitHub repository link |  |

**Abstract**

With the development of road infrastructure, there is a significant increase in number of private vehicles which results in traffic congestion, directly effecting the flow of traffic, and life of citizens. Parking becomes a significant problem in the urban areas. The research paper proposes a smart parking system to solve the current parking problem at affordable cost. Previously automatic car parking system were proposed to reduce the space or size required for parking especially in crowded places with few spaces, such as a multi-story car park providing cars with parking on multiple levels stacked vertically to increase the number of parking spaces (Ibrahim, 2017). The proposed system utilizes the latest advancement in the Information and Communication Technologies and consists of four layers: Application, Middleware, Networking, and sensor layer. It offers environmental friendly, reduces harmful emissions during parking, and is a computerized system preprogrammed without human intervention. The research paper highlights the comparison of traditional parking system with smart parking system using IoT. The paper also proposes a framework for smart parking system.

**Introduction**

Smart parking system using IoT to identify vacant positions and occupied positions without the need to waste time in finding a suitable position for the cars. The status of the parking slot detected by the wireless sensor is sent to the database via a gateway and car information is transferred to a database to save and store . The system allows the date and time of entry to the parking area, the date and time of exit from the parking space. Proper and balanced planning based on automatic and intelligent system to organize parking and provide it with sensor system to detect available, occupied, and reserved parking slots. Streamlined entry and exit of parking, no need to lose time and fuel in order to get parking, smart parking feature and the presence of surveillance cameras. To ensure the safety of the vehicle and passengers and the presence of fire extinguishers throughout the parking. Smart parking does not cause traffic jams as the time is not wasted to find the available parking space in the parking area and the driver of the vehicle is aware about the location of available parking slot.

**AIMS & OBJECTIVES**

Smart Parking is one of the most adopted and fastest growing [Smart City Solutions](https://www.plasmacomp.com/blogs/sneak-peek-into-iot-mart-cities-of-the-future) across the world. Airports, universities, shopping centers and city garages are just a few entities that have begun to realize the significant benefits of automated parking technology. The ability to connect, analyze and automate data gathered from devices, powered by and described as the [Internet of Things](https://www.plasmacomp.com/blogs/what-is-the-internet-of-things), is what makes smart parking possible.

Smart Parking involves the use of low cost sensors, real-time data and applications that allow users to monitor available and unavailable parking spots. The goal is to automate and decrease time spent manually searching for the optimal parking floor, spot and even lot. Some solutions will encompass a complete suite of services such as online payments, parking time notifications and even car searching functionalities for very large lots. A parking solution can greatly benefit both the user and the lot owner. Here are some of the top benefits:

* Optimized parking – Users find the best spot available, saving time, resources and effort. The parking lot fills up efficiently and space can be utilized properly by commercial and corporate entities.
* Reduced traffic – Traffic flow increases as fewer cars are required to drive around in search of an open parking space.
* Reduced pollution – Searching for parking burns around one million barrels of oil a day. An optimal parking solution will significantly decrease driving time, thus lowering the amount of daily vehicle emissions and ultimately reducing the global environmental footprint.
* Enhanced User Experience– A smart parking solution will integrate the entire user experience into a unified action. Driver’s payment, spot identification, location search and time notifications all seamlessly become part of the destination arrival process.
* New Revenue Streams – Many new revenue streams are possible with smart parking technology. For example, lot owners can enable tiered payment options dependent on parking space location. Also, reward programs can be integrated into existing models to encourage repeat users.

**PROBLEM DEFINATION**

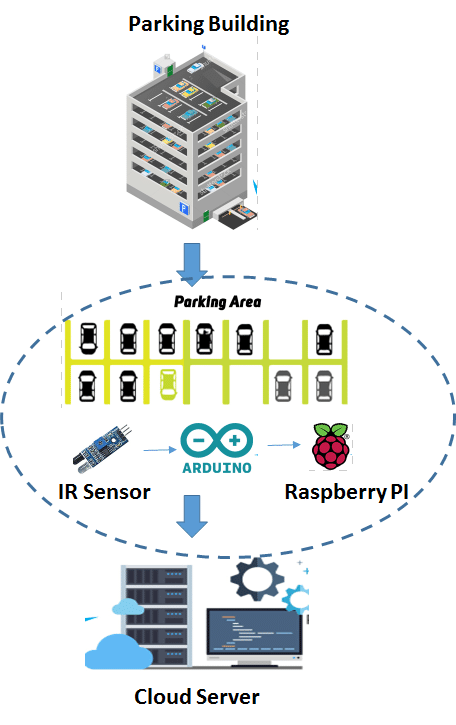
The number of cars is increasing day by day. The main problem is to find a parking slot, whether in the shopping mall or companies or at the airport or in hospitals. An average, people spent 20 minutes to find a suitable parking for the car. Most of the people park their cars in places not designated for parking, and parking in places not reserved for parking slots. This results in the disruption of the traffic and sometimes in the movement of people. Usually, there are reasons to park people in places not reserved for parking such as the area designated for parking is not enough to cater the needs. This is due to the weak planning and not thinking of solutions to keep pace with the large increase in the number of cars, and most people park the car for long times without thinking.

**PROPOSED SYSTEM**

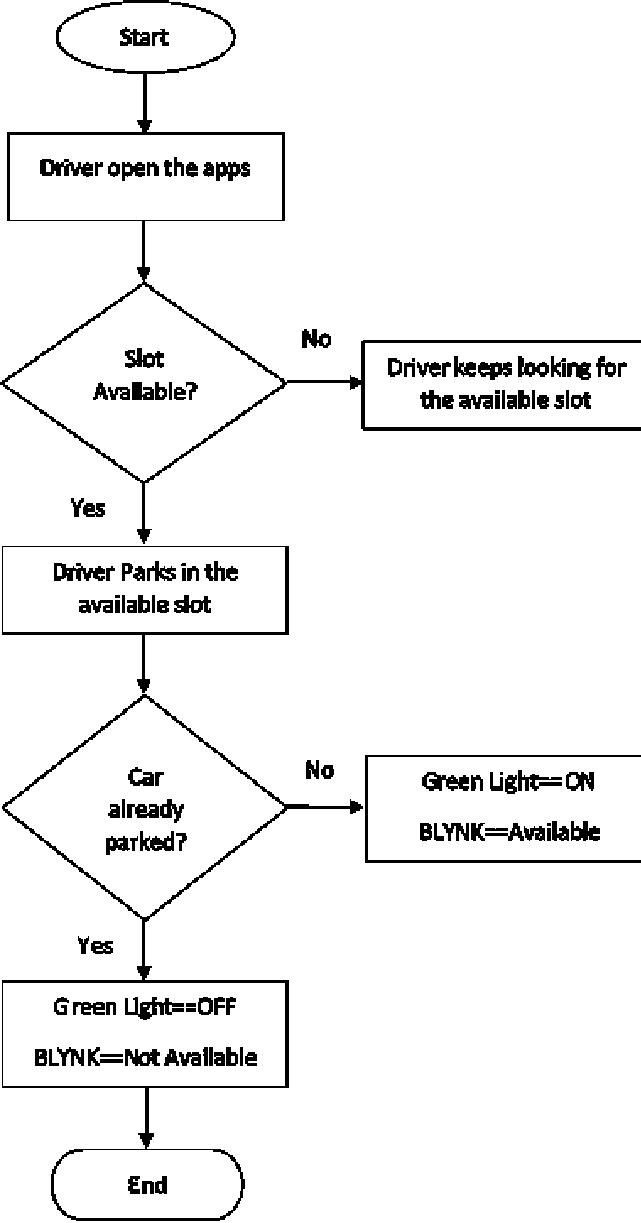
Smart parking system

The traditional parking has been developed to a parking system that helps the driver to know the occupied and available positions through a display that contains the number of available parking spaces and where they are located. Vacancy in a car cannot take this position, and in the case of the car out of the position goes light green and gives an update of the screen the presence of a vacant position can any car can stand in that position , this system is used in many places and solve the problem of random parking and not to stand in places not available to stand up .Smart parking consists of two networks, an external network and an internal network, the external network is that the user connects to the Internet and enter the application of smart parking to reserve a position and be booking anywhere available in a data network, and the user enters the server of smart parking to be able to reserve the desired position without the need to access parking Through the application the user can know any vacant positions and available and places parking , The internal network of parking is the process of connecting all devices smart parking When the server is sending a signal to the cloud and the role of the cloud send this signal to the display screen and from the display screen to the sensor and the sensor to the top of the position and vice versa and this communication is done internally without the intervention of any employee or user.

**BLOCK DIAGRAM**



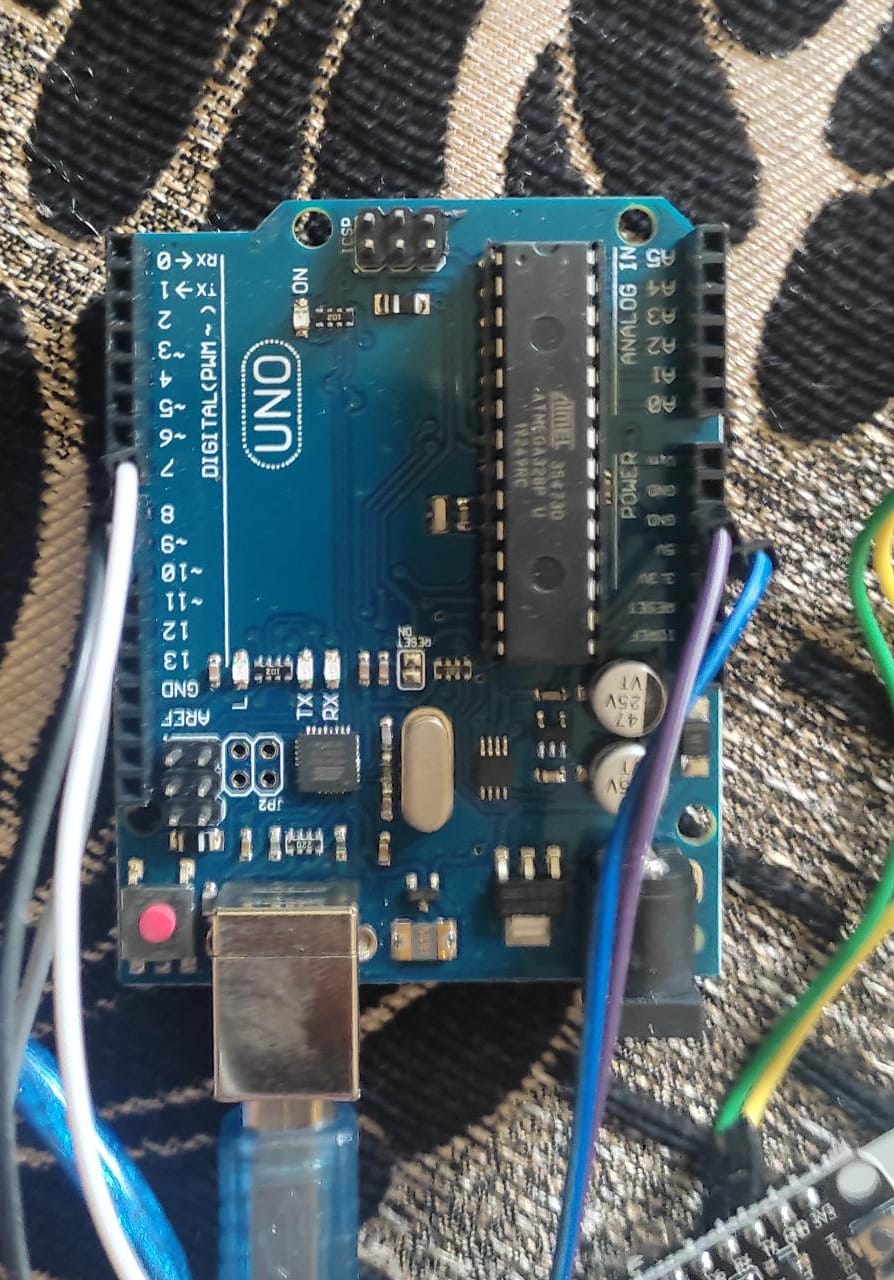
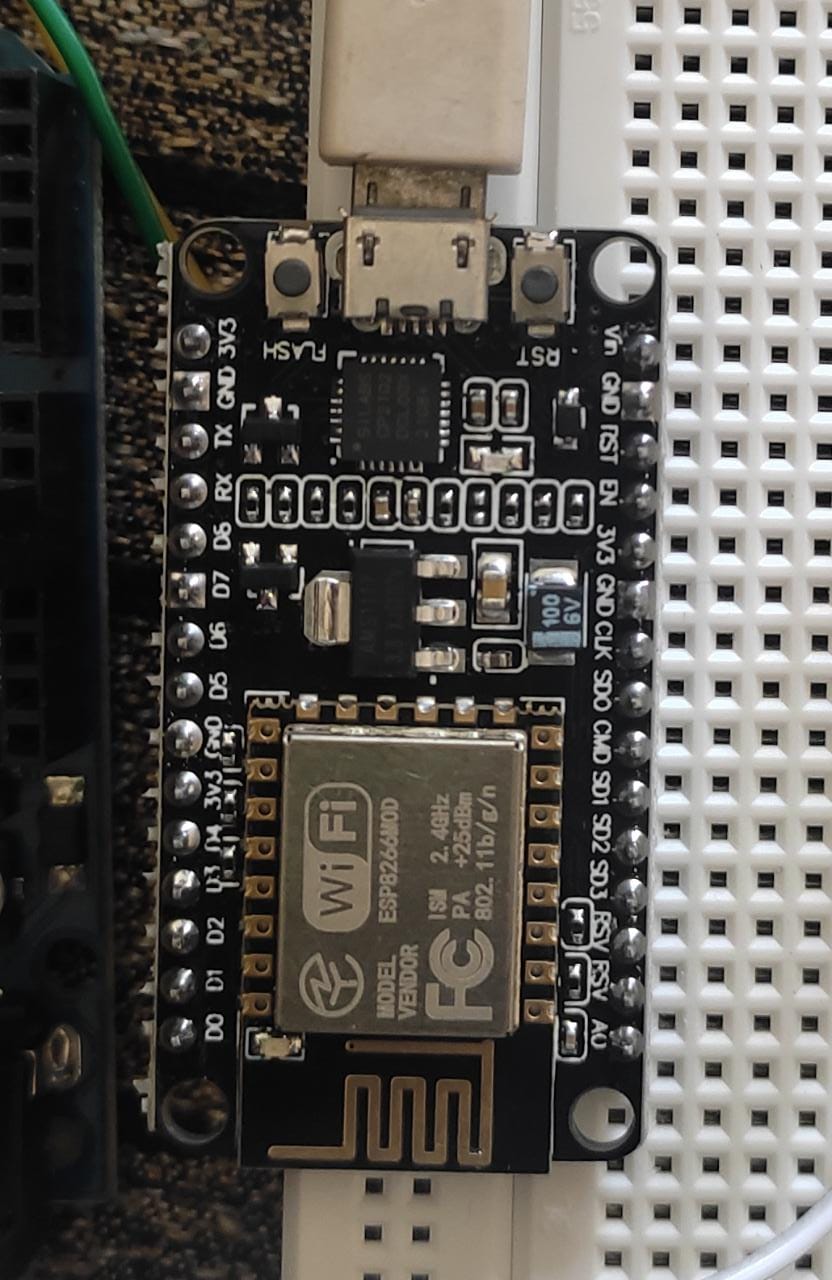
**FLOWCHART**

****

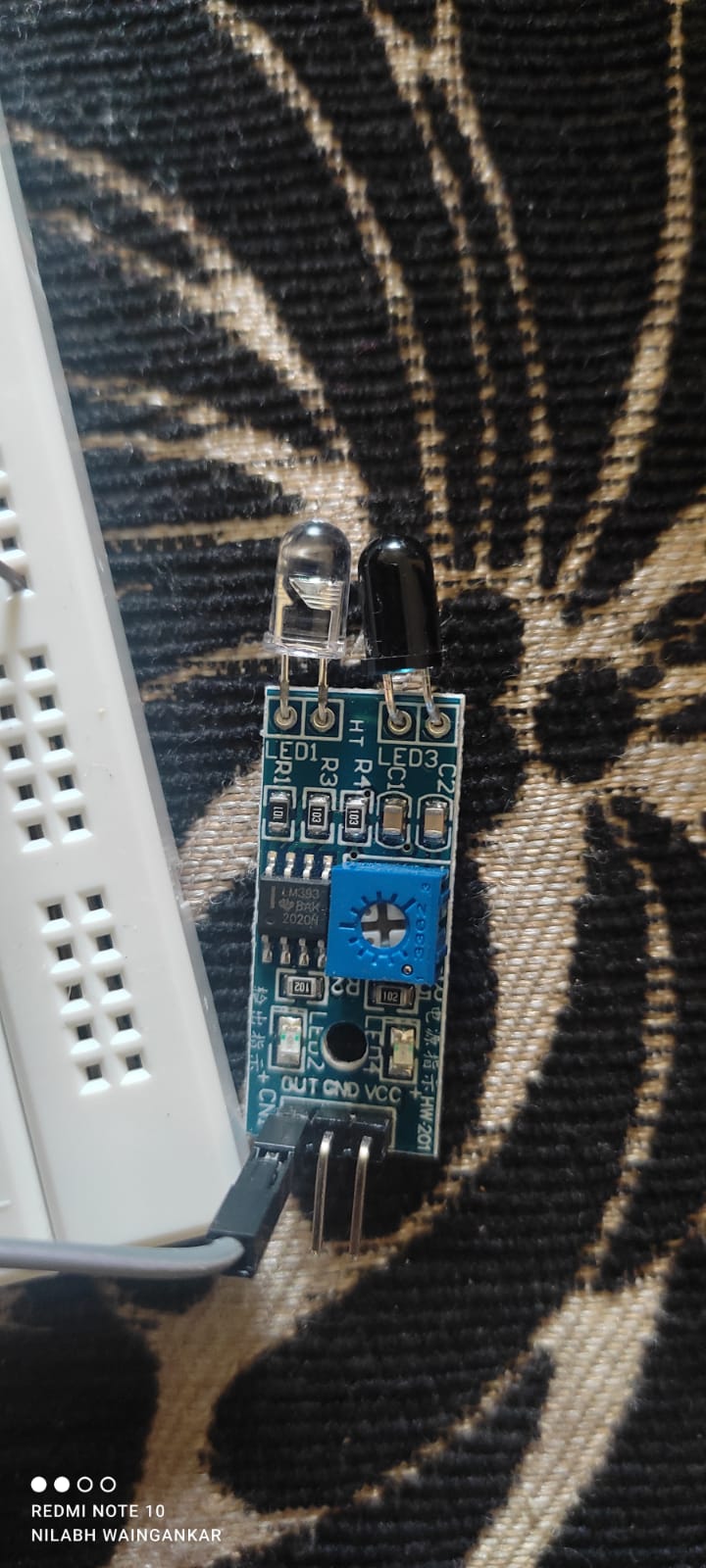
**COMPONENTS**

* **HARDWARE**

**ARDUINO NODE MCU**

** **

**IR SENSORS**

****

* **SOFTWARE**

**Blynk Application of IoT based car parking:**

For the android or apple cell phone application designing you will need to download the Blynk application from the App Store. After you download the Blynk application then you will need to register yourself for free using Facebook or any other email id. The cell phone application making is explained in the video.

**PROJECT ARCHITECTURE**

IOT based Car Parking Slots monitoring system using Arduino, Nodemcu esp8266 wifi module, and Blynk application. With the help of the Nodemcu esp8266 wifi module and Blynk application, the parking slots can be monitored from anywhere around the world. In this Tutorial, you will also learn how to use the tabs and led widgets in the Blynk application.

The Parking Area is divided into two Parkings.

Parking 1

Parking 2

Each Parking has 3 Slots and every slot has one infrared sensor. So we have a total of 6 infrared sensors. Each sensor is used to detect the presence of Car in the Slot. These infrared sensors are connected with the Arduino. So when a car is parked in the slot, the Arduino sends a command to the Nodemcu esp8266 wifi module, then Nodemcu then sends the command to the Blynk application.

**CODE**

**Car Parking Arduino Programming:**

SoftwareSerial nodemcu(2,3);

int parking1\_slot1\_ir\_s = 4; // parking slot1 infrared sensor connected with pin number 4 of arduino

int parking1\_slot2\_ir\_s = 5;

int parking1\_slot3\_ir\_s = 6;

int parking2\_slot1\_ir\_s = 7;

int parking2\_slot2\_ir\_s = 8;

int parking2\_slot3\_ir\_s = 9;

String sensor1;

String sensor2;

String sensor3;

String sensor4;

String sensor5;

String sensor6;

String cdata =""; // complete data, consisting of sensors values

void setup()

{

Serial.begin(9600);

nodemcu.begin(9600);

pinMode(parking1\_slot1\_ir\_s, INPUT);

pinMode(parking1\_slot2\_ir\_s, INPUT);

pinMode(parking1\_slot3\_ir\_s, INPUT);

pinMode(parking2\_slot1\_ir\_s, INPUT);

pinMode(parking2\_slot2\_ir\_s, INPUT);

pinMode(parking2\_slot3\_ir\_s, INPUT);

}

void loop()

{

p1slot1();

p1slot2();

p1slot3();

p2slot1();

p2slot2();

p2slot3();

   cdata = cdata + sensor1 +"," + sensor2 + ","+ sensor3 +","+ sensor4 + "," + sensor5 + "," + sensor6 +","; // comma will be used a delimeter

   Serial.println(cdata);

   nodemcu.println(cdata);

   delay(6000); // 100 milli seconds

   cdata = "";

digitalWrite(parking1\_slot1\_ir\_s, HIGH);

digitalWrite(parking1\_slot2\_ir\_s, HIGH);

digitalWrite(parking1\_slot3\_ir\_s, HIGH);

digitalWrite(parking2\_slot1\_ir\_s, HIGH);

digitalWrite(parking2\_slot2\_ir\_s, HIGH);

digitalWrite(parking2\_slot3\_ir\_s, HIGH);

}

void p1slot1() // parkng 1 slot1

{

  if( digitalRead(parking1\_slot1\_ir\_s) == LOW)

  {

  sensor1 = "255";

delay(200);

  }

if( digitalRead(parking1\_slot1\_ir\_s) == HIGH)

{

  sensor1 = "0";

delay(200);

}

}

void p1slot2() // parking 1 slot2

{

  if( digitalRead(parking1\_slot2\_ir\_s) == LOW)

  {

  sensor2 = "255";

  delay(200);

  }

if( digitalRead(parking1\_slot2\_ir\_s) == HIGH)

  {

  sensor2 = "0";

delay(200);

  }

}

void p1slot3() // parking 1 slot3

{

  if( digitalRead(parking1\_slot3\_ir\_s) == LOW)

  {

  sensor3 = "255";

  delay(200);

  }

if( digitalRead(parking1\_slot3\_ir\_s) == HIGH)

  {

  sensor3 = "0";

delay(200);

  }

}

// now for parking 2

void p2slot1() // parking 1 slot3

{

  if( digitalRead(parking2\_slot1\_ir\_s) == LOW)

  {

  sensor4 = "255";

  delay(200);

  }

if( digitalRead(parking2\_slot1\_ir\_s) == HIGH)

  {

  sensor4 = "0";

delay(200);

  }

}

void p2slot2() // parking 1 slot3

{

  if( digitalRead(parking2\_slot2\_ir\_s) == LOW)

  {

  sensor5 = "255";

  delay(200);

  }

if( digitalRead(parking2\_slot2\_ir\_s) == HIGH)

  {

  sensor5 = "0";

delay(200);

  }

}

void p2slot3() // parking 1 slot3

{

  if( digitalRead(parking2\_slot3\_ir\_s) == LOW)

  {

  sensor6 = "255";

  delay(200);

  }

if( digitalRead(parking2\_slot3\_ir\_s) == HIGH)

  {

  sensor6 = "0";

delay(200);

  }

}

**Nodemcu esp8266 wifi module Programming of IoT based car parking**:

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <SoftwareSerial.h>

#include <SimpleTimer.h>

char auth[] = "ac173b0527c94a91a6cde0dcdfe6bdef";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "ZONG MBB-E8231-6E63";

char pass[] = "08659650";

SimpleTimer timer;

String myString; // complete message from arduino, which consistors of snesors data

char rdata; // received charactors

int firstVal, secondVal,thirdVal; // sensors

int led1,led2,led3,led4,led5,led6;

// This function sends Arduino's up time every second to Virtual Pin (1).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void myTimerEvent()

{

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V1, millis() / 1000);

}

void setup()

{

  // Debug console

  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

    timer.setInterval(1000L,sensorvalue1);

    timer.setInterval(1000L,sensorvalue2);

    timer.setInterval(1000L,sensorvalue3);

    timer.setInterval(1000L,sensorvalue4);

    timer.setInterval(1000L,sensorvalue5);

    timer.setInterval(1000L,sensorvalue6);

}

void loop()

{

   if (Serial.available() == 0 )

   {

  Blynk.run();

  timer.run(); // Initiates BlynkTimer

   }

  if (Serial.available() > 0 )

  {

    rdata = Serial.read();

    myString = myString+ rdata;

   // Serial.print(rdata);

    if( rdata == '\n')

    {

     Serial.println(myString);

  // Serial.println("fahad");

// new code

String l = getValue(myString, ',', 0);

String m = getValue(myString, ',', 1);

String n = getValue(myString, ',', 2);

String o = getValue(myString, ',', 3);

String p = getValue(myString, ',', 4);

String q = getValue(myString, ',', 5);

// these leds represents the leds used in Blynk application

led1 = l.toInt();

led2 = m.toInt();

led3 = n.toInt();

led4 = o.toInt();

led5 = p.toInt();

led6 = q.toInt();

  myString = "";

// end new code

    }

  }

}

void sensorvalue1()

{

int sdata = led1;

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V10, sdata);

}

void sensorvalue2()

{

int sdata = led2;

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V11, sdata);

}

void sensorvalue3()

{

int sdata = led3;

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V12, sdata);

}

void sensorvalue4()

{

int sdata = led4;

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V13, sdata);

}

void sensorvalue5()

{

int sdata = led5;

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V14, sdata);

}

void sensorvalue6()

{

int sdata = led6;

  // You can send any value at any time.

  // Please don't send more that 10 values per second.

  Blynk.virtualWrite(V15, sdata);

}

String getValue(String data, char separator, int index)

{

    int found = 0;

    int strIndex[] = { 0, -1 };

    int maxIndex = data.length() - 1;

    for (int i = 0; i <= maxIndex && found <= index; i++) {

        if (data.charAt(i) == separator || i == maxIndex) {

            found++;

            strIndex[0] = strIndex[1] + 1;

            strIndex[1] = (i == maxIndex) ? i+1 : i;

        }

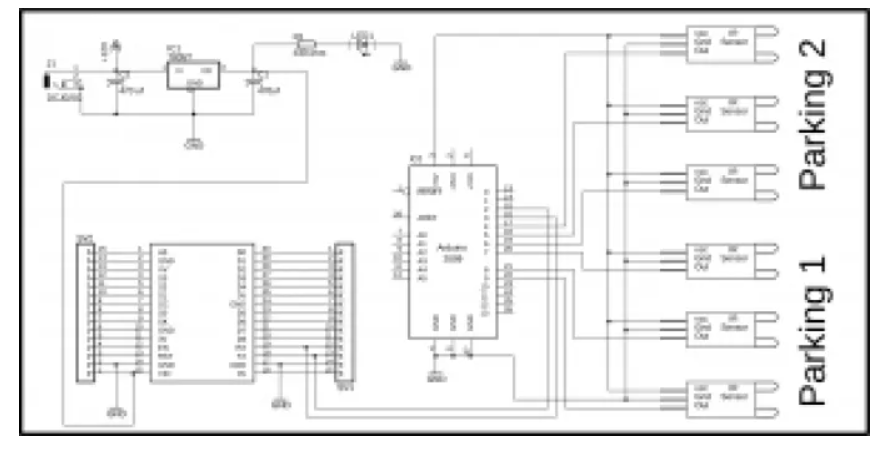
    }

    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

**IMPLEMENTATION**

**CIRCUIT DIAGRAM:**

****

As you can see six infrared sensors are connected with the Arduino pins 4 to 9. The infrared sensor VCC pins are connected with the Arduino’s 5v. Grounds are connected with the Arduino’s Ground while the out pins of all the infrared sensors are connected with pin 4 to 9.

The Nodemcu module tx and Rx pins are connected with pin2 and pin3 of the Arduino. while the Vin pin of the Nodemcu module is connected with the output of the voltage regulator. This is a regulated 5v power supply based on the lm7805 voltage regulator.

Two 470uf capacitors are connected at the input and output side of the regulator.  A 330-ohm resistor is connected in series with a 2.5v led. This is a current limiting resistor. While J1 is a dc female socket, over here you can connect a 12v adaptor or battery. But you can also power up the Nodemcu module using your laptop USB port. I will be using two USB cables, I cable will be used for powering up the Arduino and the other USB cable will be used to power up the Nodemcu.

**CONCLUSION & FUTURE SCOPE**

IoT based Smart parking system has been proposed in this paper to avoid traffic congestion, random parking, and obstruction of traffic in the parking area as well as to search and wait for a parking space. The proposed system described in this paper is built with four layers: Application, Middleware, Networking, and sensor layer. The research paper highlights the comparison of traditional parking system with smart parking system using IoT. The paper also proposes a framework for smart parking system

In future works, this system can be improved by adding other applications such as online booking by using gsm. The driver or user can book their parking lot at home or on the way to the shopping mall. This can reduce the time of the user to searching the vacant parking lot. As a further study, different sensor systems can be added to improve this system to detect the object and guide the driver or users fastest. We will try to reduce the mechanical structure and try to make it ecofriendly.