Smart Parking System

A PROJECT REPORT

For

INTERNET OF THINGS

In

B. Tech-Information Technology and Engineering

By

Mugilan A -19BIT0102
Pranav Sunil Mahalpure -19BIT0109
Kunal Kumar-19BIT0171

Under the Guidance of

Prof.Kavitha B.R

Associate Professor,SITE

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

August,2022

Abstract:

Efficient and smart way to automate the management of the parking system that allocates an efficient parking space using internet of things technology. The IoT provides a wireless access to the system and the user can keep a track of the availability of the parking area. With increase in the population of the vehicles in metropolitan cities, road congestion is the major problem that is being faced. The aim of this project is to resolve this issue. The user usually wastes his time and efforts in search of the availability of the free space in a specified parking area. In this project, we are using IOT to determine the availability of parking space in a parking slot. A software(web portal) is also created by which people can check if a parking slot is available. This project efficiently makes people to identify free parking slots.

Introduction:

Internet of thing (IoT) has the ability to transfer data through network without involving human interactions. IoT allows user to use affordable wireless technology and also helps the user to transfer the data into the cloud. IoT helps the user to maintain transparency. The idea of IoT started with the identity of things for connecting various devices. These devices can be controlled or monitored through computers over internet. IoT contains two prominent words “Internet” and “Things”, where Internet is a vast network for connecting servers with devices.

In recent times, car parking became a major problem in metro cities. People waste a lot of fuel in searching for parking space and this also leads to traffic coagulation. To avoid this problem we propose this project, which uses the concepts of IOT to efficiently determine the free parking slots available. We are using several components for this project namely, Arduino, Node MCU, Object detection sensor(IR Sensor) and wifi module. The IR Sensors will be implemented in every parking slots. So when an car is parked in a slot, the IR sensor will detect this and sends the signal to Aurdino . the aurdino will process the signals and send the results to cloud with the help of Node MCU. This data is accessed by our software by which customers can view the free parking slots. This project can be implemented in shopping malls and big parking areas.

Hardware components:

(i) Arduino

The Arduino UNO is a widely used open-source microcontroller board based on the ATmega328P microcontroller and developed by Arduino.cc. The Arduino is the major control unit to detect or alert when an accident occurs. It collects the data from vibration sensor, GPRS and GSM modules and reflects the output either in display system or through a message. Here vibration sensor plays a major role. This vibration sensor will receive the vibrations of the vehicle which in turn acts as a accident detection module. Arduino gathers the information from all other modules and sends the message to the receiver though GSM module.

(ii) Node MCU
(Node Micro Controller Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266. MCU stands for Micro Controller Unit - which really means it is a computer on a single chip. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals. They are used to automate automobile engine control, implantable medical devices, remote controls, office machines, appliances, power tools, toys etc.

(ii) IR Sensor
IR Sensor An infrared sensor is basically an electronic device which is used to detect the presence of objects. Infrared light is emitted by this device. If this device does not detect any IR light reflected back that means there is no object present. If the light is detected by the sensor there is an object present.

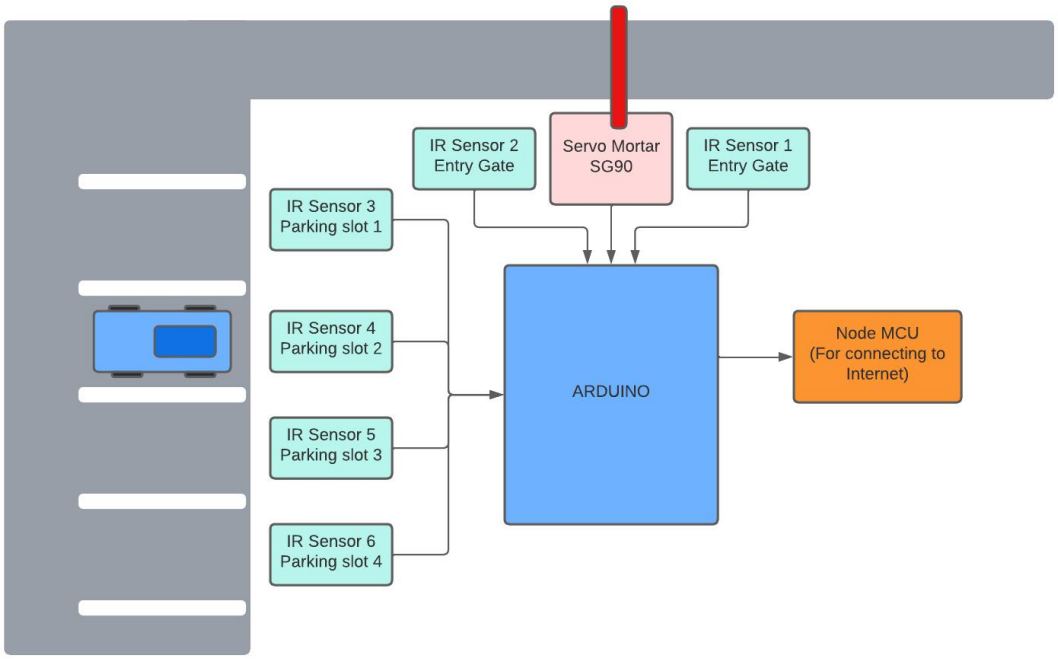
(iv) Wifi Component
It is used to send data from embedded system to the internet using URL by HTTP POST method using TCP/IP protocol. It is developed by espressif systems. It is a 32 bit microcontroller with 80kb user data. It contains 16 gpio pins.

(v) LED Lights

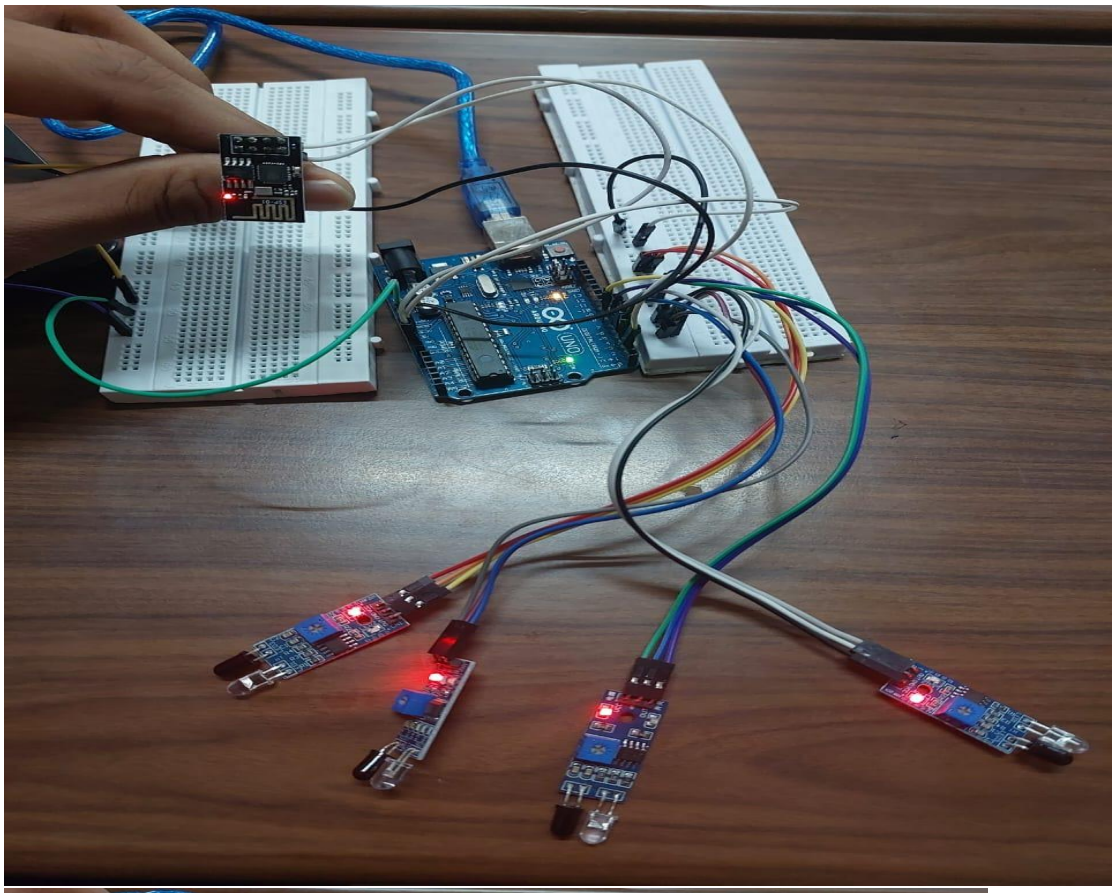
A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. LED Lights are used to indicate if a parking slot is occupied. When it is occupied the IR Sensor detects it and sends signal to LEDs. The LED will then light up for indication.

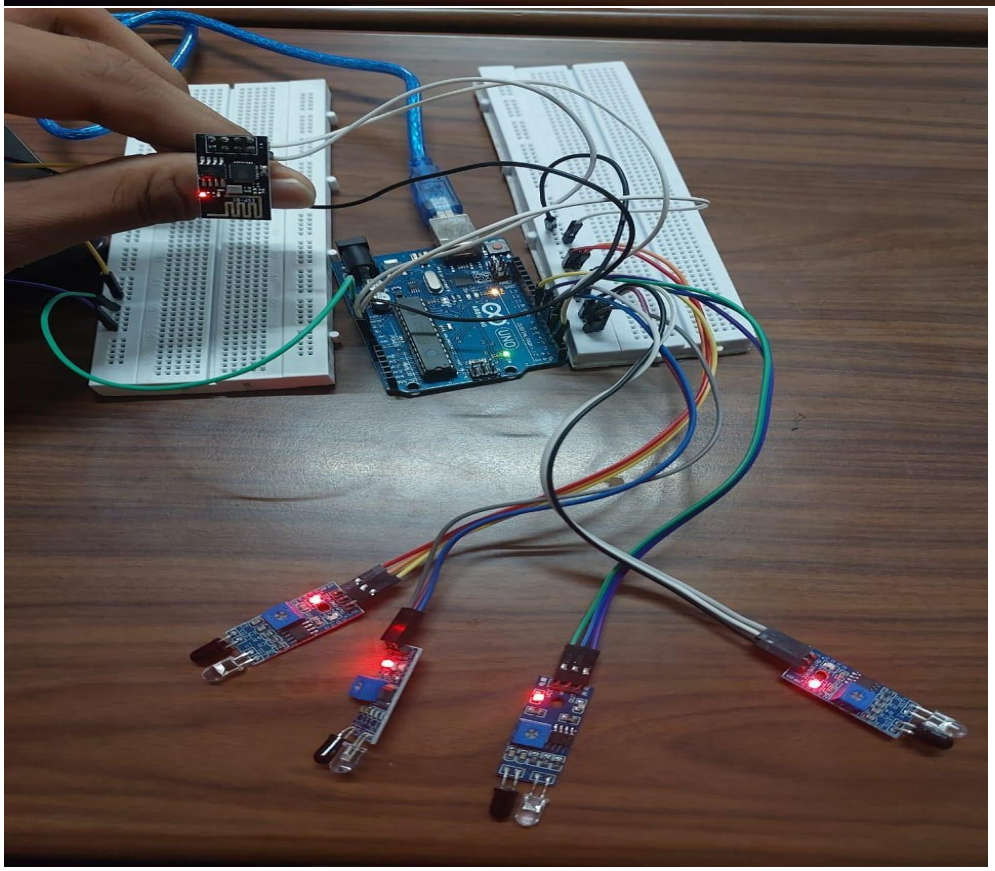
Working
In our project, we will have two IR sensors for detecting a car and a IR sensor for each parking slot. When a car parks in a parking slot, the IR Sensor will get activated and it sends the signal to the arduino board. This signal is processed by aurduino and it calculated the free parking slots available and the position of the free slot. This information is relayed to the incoming customers. If the parking slot is full, the entry gate will be closed so that new vehicles can’t cause traffic inside the parking slots. The entry and exit sensor will detect on incoming car (in case there is free slot) and open the entry gate for them.

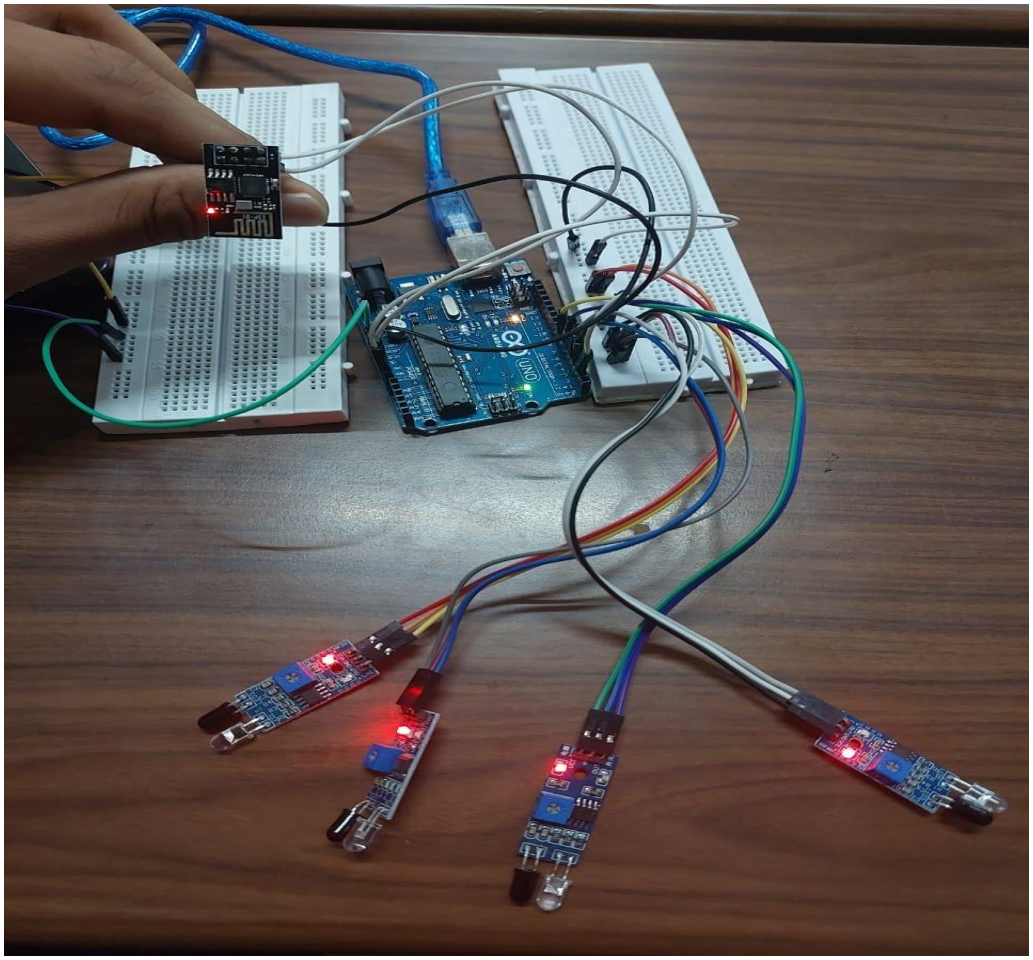
Model Diagram

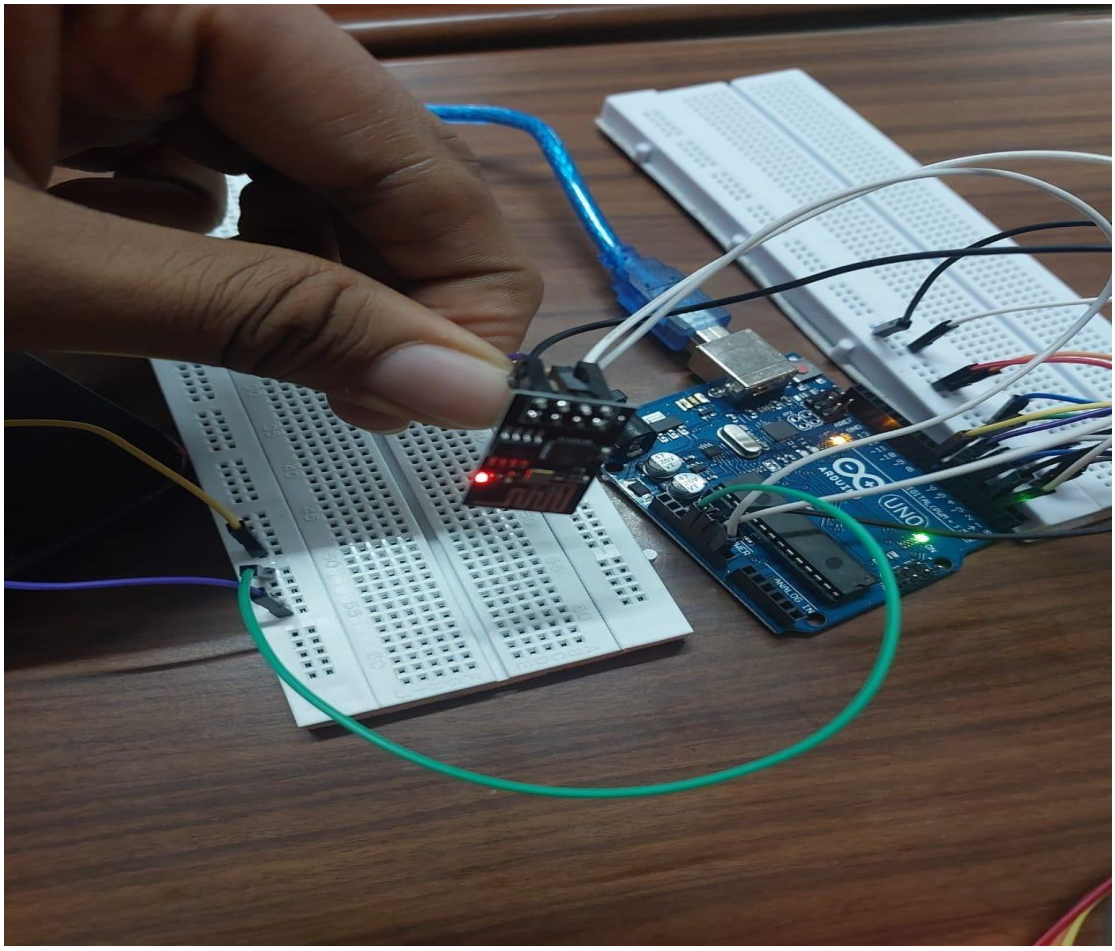


SETUP and OUTPUT









Arduino Code

```
#include <SoftwareSerial.h>
#define RX 2
#define TX 3
String AP = "POCO M2";
String PASS = "mugil123";
String API = "GN689L1DPR58292X";
String HOST = "api.thingspeak.com";
String PORT = "80";
String field = "field1";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
String valSensor = "40000";
SoftwareSerial esp8266(RX,TX);

 #define ir_car1 5
#define ir_car2 6
#define ir_car3 7
#define ir_car4 8
String p1;
String p2;
```

```
String p3;
String p4;
int S1=0, S2=0, S3=0, S4=0 ;
int flag1=0, flag2=0;
int avail= 4;
int slot = 4;
void setup() {
  Serial.begin(9600);
  esp8266.begin(115200);
  sendCommand("AT",5,"OK");
  sendCommand("AT+CWMODE=1",5,"OK");
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
}

void loop() {
 valSensor = getSensorData();
 String getData = "GET /update?api_key="+ API +"&"+ field +"="+String(valSensor);
sendCommand("AT+CIPMUX=1",5,"OK");
 sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK");
 sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");
 esp8266.println(getData);delay(1500);countTrueCommand++;
 sendCommand("AT+CIPCLOSE=0",5,"OK");
}
int getParkingslot1(){
  Read_Sensor();
  int ans=0;
  if(S1==1)
   {
    ans=1;
   }
  else
   {
    ans=0;
   }
   return ans;
}
int getParkingslot2(){
  Read_Sensor();
  int ans=0;
  if(S2==1)
   {
    ans=1;
   }
  else
   {
    ans=0;
   }
   return ans;
}
int getParkingslot3(){
  Read_Sensor();
```

```
 int ans=0;
 if(S3==1)
  {
   ans=1;
  }
 else
  {
   ans=0;
  }
  return ans;
}
int getParkingslot4(){
 Read_Sensor();
 int ans=0;
 if(S4==1)
  {
   ans=1;
  }
 else
  {
   ans=0;
  }
  return ans;
}
String getSensorData(){
  Read_Sensor();
  String p1="";
  String p2="";
  String p3="";
  String p4="";
 if(S1==1)
  {
   //Serial.print("  S1:Fill ");
   p1="1";
  }
 else
  {
   //Serial.print("  S1:Empty");
   p1="0";
  }

 if(S2==1)
  {
   //Serial.print("  S2:Fill ");
   p2="1";
   }
 else
  {
   //Serial.print("  S2:Empty");
   p2="0";
   }
```

```
    if(S3==1)
   {
    //Serial.print("  S3:Fill ");
    p3="1";
    }
   else
    {
    //Serial.print("  S3:Empty");
    p3="0";
    }
     if(S4==1)
   {
    //Serial.print("  S4:Fill ");
    p4="1";
    }
   else
    {
    //Serial.print("  S4:Empty");
    p4="0";
    }
   int total = S1+S2+S3+S4;
  slot = avail-total;
  String ans="1"+p1+p2+p3+p4;
  return ans;
}
void Read_Sensor()
{
 S1=0, S2=0, S3=0, S4=0;
 if(digitalRead(ir_car1) == 0){S1=1;}
   if(digitalRead(ir_car2) == 0){S2=1;}
 if(digitalRead(ir_car3) == 0){S3=1;}
 if(digitalRead(ir_car4) == 0){S4=1;}

}
void sendCommand(String command, int maxTime, char readReplay[]) {
  Serial.print(countTrueCommand);
  Serial.print(". at command => ");
  Serial.print(command);
  Serial.print(" ");
  while(countTimeCommand < (maxTime*1))
  {
    esp8266.println(command);//at+cipsend
    if(esp8266.find(readReplay))//ok
    {
      found = true;
      break;
    }

    countTimeCommand++;
  }
```

```
 if(found == true)
 {
  Serial.println("OYI");
  countTrueCommand++;
  countTimeCommand = 0;
 }

 if(found == false)
 {
  Serial.println("Fail");
  countTrueCommand = 0;
  countTimeCommand = 0;
 }

 found = false;
}
```

Matlab Code

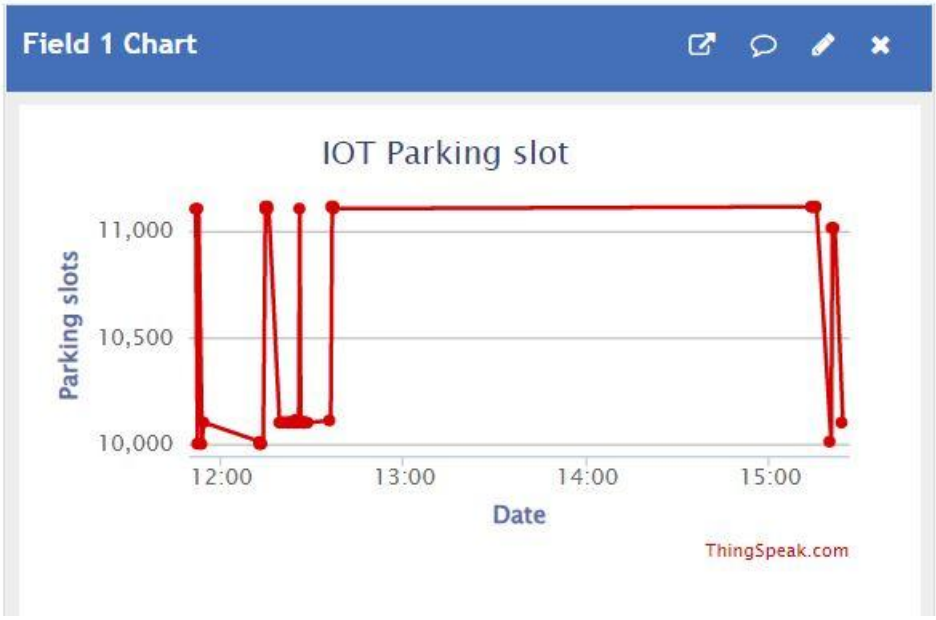
```
% side pane of this page.

% TODO - Replace the [] with channel ID to read data from:
readChannelID = 1842167;
% TODO - Enter the Read API Key between the '' below:
readAPIKey = 'AIL3JZ8VMKAQJQYX';

% TODO - Replace the [] with channel ID to write data to:
writeChannelID = 1842840;
% TODO - Enter the Write API Key between the '' below:
writeAPIKey = 'MQNNBHP6V5W0S3RA';

%% Read Data %%
data = thingSpeakRead(readChannelID, 'Fields',1, 'ReadKey',readAPIKey);
park = dec2base(data,10) - '0';
p1=park(2);
p2=park(3);
p3=park(4);
p4=park(5);

%% Analyze Data %%
% Add code in this section to analyze data and store the result in the
% 'analyzedData' variable.
analyzedData = data;

%% Write Data %%
thingSpeakWrite(writeChannelID, [p1,p2,p3,p4], 'Fields',[1,2,3,4], 'WriteKey', writeAPIKey);
%%thingSpeakWrite(writeChannelID, p2, 'Fields',2, 'WriteKey',writeAPIKey);
%%thingSpeakWrite(writeChannelID, p3, 'Fields',3,'WriteKey', writeAPIKey);
%%thingSpeakWrite(writeChannelID, p4, 'Fields',4,'WriteKey', writeAPIKey);
```

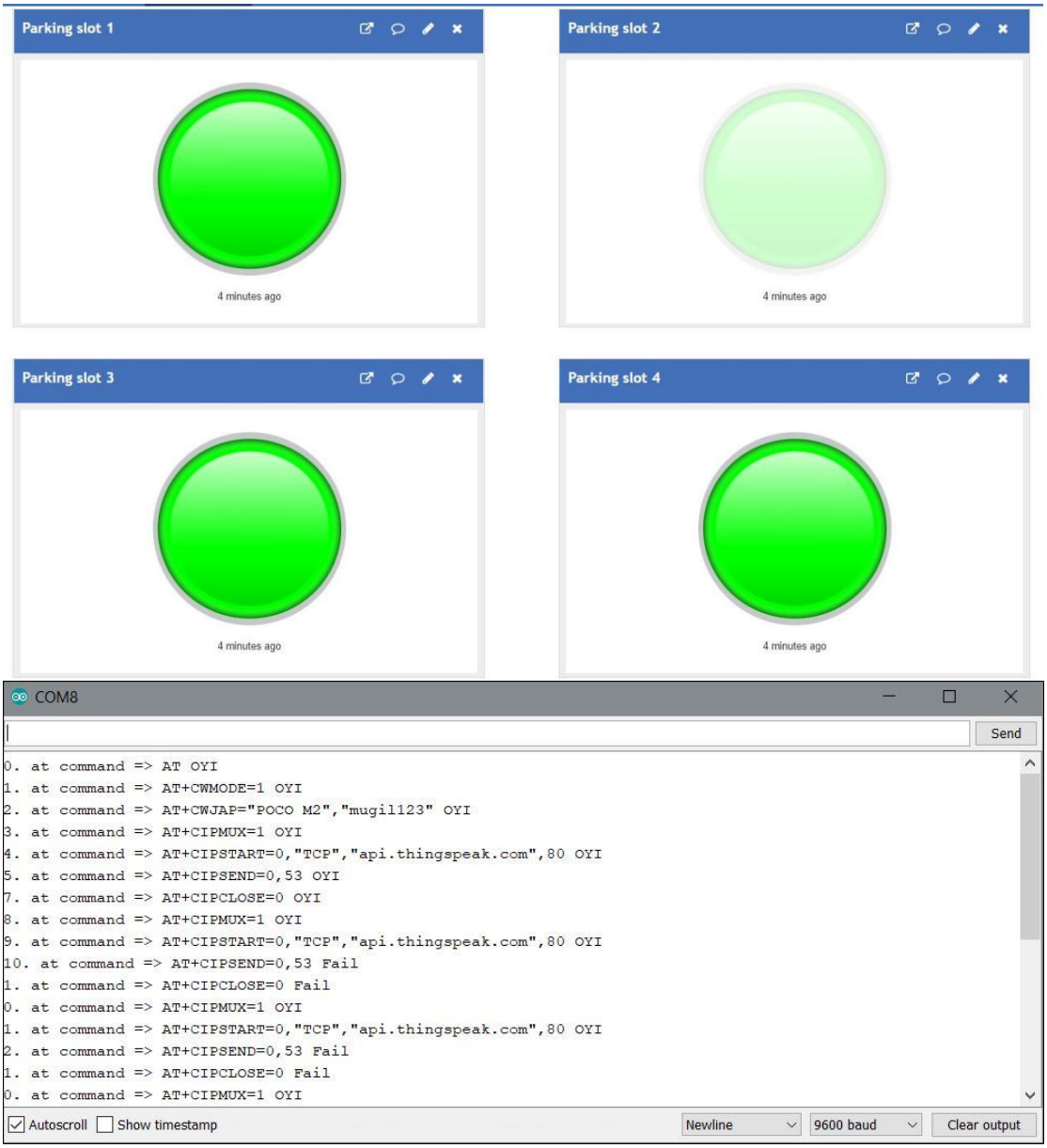

Field 1 Chart
Parking slots detection
Parking slots
1
0.5
0
13:00
14:00
15:00
Date
ThingSpeak.com


Field 2 Chart
Parking slots detection
Parking slot 1
1
0.5
0
13:00
14:00
15:00
Date
ThingSpeak.com


Field 3 Chart
Parking slots detection
Parking slot 2
1
0.5
0
13:00
14:00
15:00
Date
ThingSpeak.com


Field 4 Chart
Parking slots detection
Parking slot 3
1
0.5
0
13:00
14:00
15:00
Date
ThingSpeak.com


Field 1 Chart
IOT Parking slot
Parking slots
11,000
10,500
10,000
12:00
13:00
14:00
15:00
Date
ThingSpeak.com



Conclusion

The concept of Smart Cities have always been a dream for humanity. Since the past couple of years large advancements have been made in making smart cities a reality. The growth of Internet of Things and Cloud technologies have give rise to new possibilities in terms of smart cities. Smart parking facilities and traffic management systems have always been at the core of constructing smart cities. In this paper, we address the issue of parking and present an IOT based Cloud integrated smart parking system. The system that we propose provides real time information regarding availability of parking slots in a parking area. Users from remote locations could book a parking slot for them by the use of our mobile application. The efforts made in this paper are indented to improve the parking facilities of a city and thereby aiming to enhance the quality of life of its people.

References:

1.https://the constructor.org/transportation/vehicle-traffic volume-counting-using-arduino-uno/20528/

2.https://www.element14.com/community/thread/27501/l/vehicle-counter-using-arduino?displayFullThread=true

3.https://www.hackster.io/team-uno/car-counter-using-arduino processing-php-adaee2

4.https://store.arduino.cc/usa/arduino-uno-rev3

5.https://www.elecrow.com/inductive-loop-vehicle-detector v21-pcb-board-p-1458.html