Department Of Chemical Engineering, INDIAN INSTITUE OF TECHNOLOGY, GUWAHATI

IoT BASED CAR PARKING MANAGEMENT SYSTEM

DIY HARDWARE PROJECT AND ITS APPLICATIONS

BY **SHUBHAM KUMAR GUPTA**

ROLL **180107058**

Email: shubham18f@iitg.ac.in

November 6, 2020

ABSTRACT

In scientific and engineering disciplines, researchers and engineers are most likely to discover solutions to diverse mathematical expressions, in order to solve many real life problems. Among many there is one more problem that is needed to be solved i.e CAR PARKING MANAGEMENT SYSTEM, due to several factors like high traffic and many times if this is not taken care of properly then this may lead to problems and chaos in real life scenario. In this project I am going to demonstrate solution of one such problem that exists today.

KEYWORDS

Optimization, Manipulation, Internet Of Things, Car Parking, IR Sensor, ARDUINO, NODEMCU, I2C ADAPTER, LCD, Ultrasonic Sensor, Servo Motors

1 INTRODUCTION

Car parking is a major issue in modern congested cities of today. There are too many vehicles on the road and not enough parking space. Finding parking space is a problem for the driver. Sometimes it kills a lot of time of driver to find a parking space. This has led to the need for efficient parking management systems. IoT based project smart parking system brings a solution. A major purpose of this project is to avoid unnecessary traveling by a driver for the parking area. Monitoring the whole area at the run time gives the driver an image of the entire parking area, and the user can select that free parking space.

2 IMPLEMENTED ATTRIBUTES

I have implemented all the objectives that were given to us. They are listed as below:

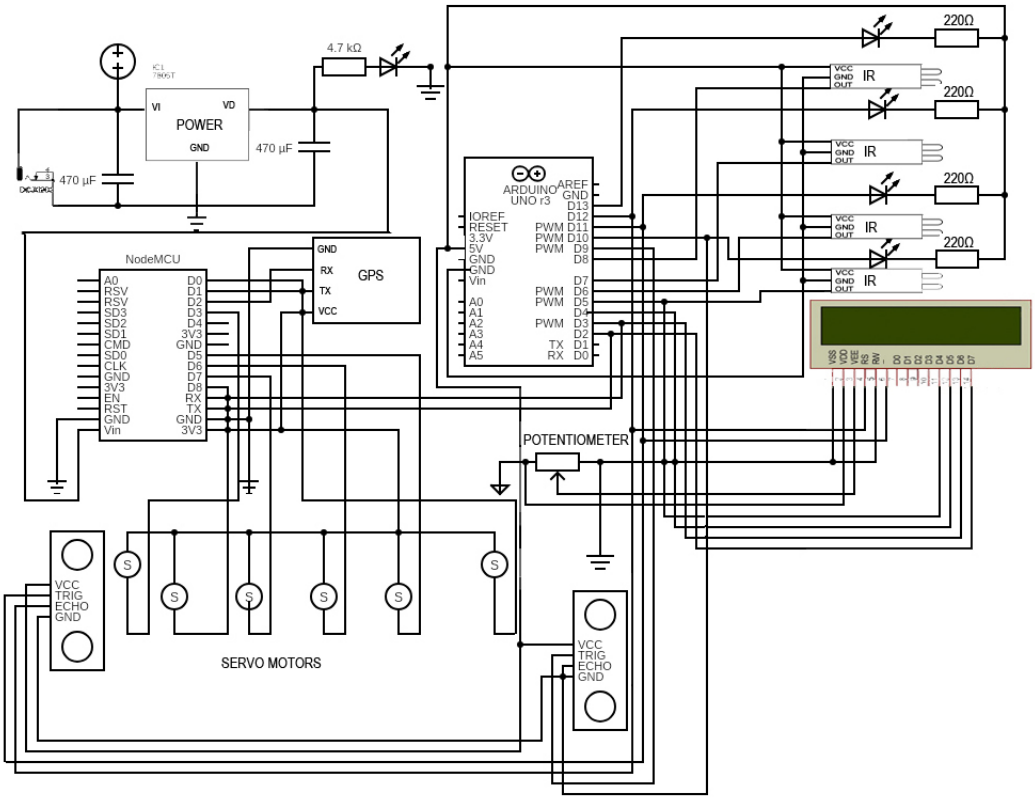
- Sense the parking slot occupancy in a car parking plot of a city
- Location will be tracked by associated GPS sensor
- Collected data will be transmitted to the server in the Internet for storing and processing
- A user/driver can monitor the data using a Mobile App in which data are fed from the server
- A user/driver can inform willingness to reserve a slot in the parking plot
- Inform the user/driver about his booking (time duration and cost) through SMS/Mobile App
- System will sense if a vehicle has arrived on gate for automated gate opening.
- Display the free parking slots in the LCD display placed in every parking plot

HARDWARES USESD

1. Few IR Sensors
2. Two DC Motors and Motor Controlling Board
3. GPS Sensors
4. LCD Display
5. Arduino
6. Node MCU
7. Breadboard / PCB
8. Adapters, Cables, LEDs, Wires
9. Smartphone
10. Google Suite
11. BLYNK APPLICATION

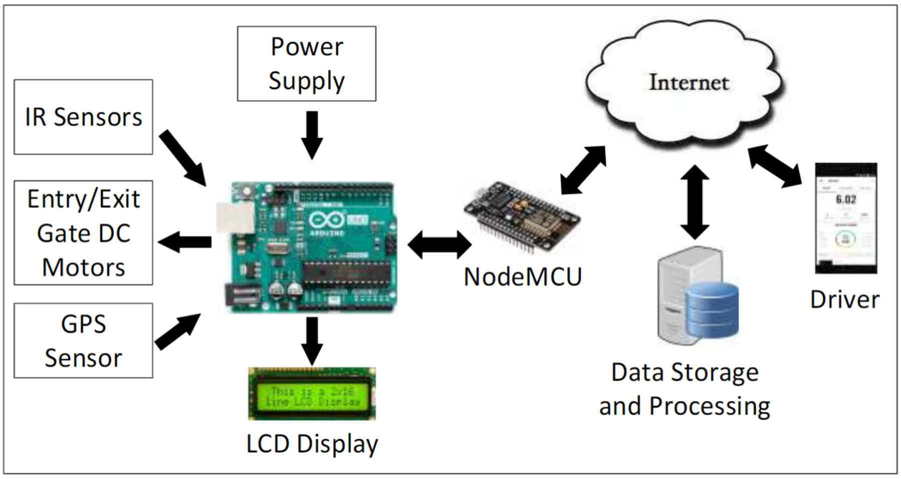
2.1 Configuration Diagram

Below displays the Circuit Diagram of our Circuit



2.2 Block Diagram

Below displays the BLOCK Diagram of our Circuit



IR Sensors
Power Supply
Entry/Exit Gate DC Motors
GPS Sensor
LCD Display
This is a 2x16 line LCD Display
NodeMCU
Internet
Data Storage and Processing
Driver
6.02

2.3 Code

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27,16,2);
SoftwareSerial nodemcu(2,3);

int parking1_slot1_ir_s = 5; // parking slot1 infrared sensor connected with pin number 4 of arduino
int parking1_slot2_ir_s = 6;

int parking2_slot1_ir_s = 7;
int parking2_slot2_ir_s = 8;

const int trig=12;
const int echo =11;

const int trig2=9;
const int echo2 =10;


String sensor1="0";
String sensor2="0";
String sensor3="0";
String sensor4="0";
String last="NULL";

int LED1=A2;
int LED2=A3;
int LED3=A0;
int LED4=A1;


String cdata =""; // complete data, consisting of sensors values
void setup()
{
    lcd.begin();
    lcd.backlight();
    Serial.begin(9600);
    nodemcu.begin(9600);
    pinMode(parking1_slot1_ir_s, INPUT);
    pinMode(parking1_slot2_ir_s, INPUT);

    pinMode(parking2_slot1_ir_s, INPUT);
    pinMode(parking2_slot2_ir_s, INPUT);
    pinMode(trig,OUTPUT);
    pinMode(echo,INPUT);
    pinMode(trig2,OUTPUT);
    pinMode(echo2,INPUT);

}
String k="a";
String entry="closed",entry2="closed";
void loop()
{
    long duration,distance;
    digitalWrite(trig,HIGH);
    delayMicroseconds(1000);
    digitalWrite(trig,LOW);
    duration=pulseIn(echo,HIGH);
    distance=(duration/2)/29.1;
    if(distance<10 && entry=="closed"){
        entry="opened";
        Serial.print("OP"+String(distance));
    }else if(distance>10 && entry=="opened" ){
        entry="closed";
        Serial.print("CL"+String(distance));
    }
    long duration2,distance2;
    digitalWrite(trig2,HIGH);
    delayMicroseconds(1000);
    digitalWrite(trig2,LOW);
    duration2=pulseIn(echo2,255);
    distance2=(duration2/2)/29.1;
    if(distance2<10 && entry2=="closed"){
        entry2="opened";
    }else if(distance2>10 && entry2=="opened"){
        entry2="closed";
    }
    p1slot1();
    p1slot2();
    p2slot1();
    p2slot2();
    int space=(sensor1=="0"?0:1)+(sensor2=="0"?0:1)+(sensor3=="0"?0:1)+(sensor4=="0"?0:1);
    space=4-space;
    String s=String(space);

    cdata = cdata + sensor1 +"," + sensor2 + ","+ sensor3 +","+ sensor4 + ","+s+" SPACES LEFT!,"+entry+","+entry2;
    lcd.print(s+" LEFT!!");
    for(int i=0;i<2;i++)
    {lcd.print("        ");}

    Serial.println(s+" LEFT!!");

    Serial.println(cdata);
    nodemcu.println(cdata);
    delay(800); // 100 milli seconds
    cdata = "";

    digitalWrite(parking1_slot1_ir_s, HIGH);
    digitalWrite(parking1_slot2_ir_s, HIGH);

    digitalWrite(parking2_slot1_ir_s, HIGH);
    digitalWrite(parking2_slot2_ir_s, HIGH);

}

void p1slot1()
{
    if( digitalRead(parking1_slot1_ir_s) == LOW)
    {
        sensor1 = "255";
        analogWrite(LED1,255);

        delay(100);
    }
    if( digitalRead(parking1_slot1_ir_s) == HIGH)
    {
        sensor1 = "0";
        analogWrite(LED1,0);
        delay(100);
    }
}

void p1slot2() // parking 1 slot2
{
    if( digitalRead(parking1_slot2_ir_s) == LOW)
    {
        sensor2 = "255";
        analogWrite(LED2,255);
        delay(100);
    }
    if( digitalRead (parking1_slot2_ir_s) == HIGH)
    {
        sensor2 = "0";
        analogWrite(LED2,0);
        delay(100);
    }
}
void p2slot1() // parking 1 slot3
{
    if( digitalRead(parking2_slot1_ir_s) == LOW)
    {
        sensor3 = "255";
        analogWrite(LED3,255);

        delay(100);
    }
    if( digitalRead(parking2_slot1_ir_s) == HIGH)
    {
        sensor3 = "0";
        analogWrite(LED3,0);

        delay(100);
    }
}

void p2slot2() // parking 1 slot3
{
    if( digitalRead(parking2_slot2_ir_s) == LOW)
    {
        sensor4 = "255";
        analogWrite(LED4,255);

        delay(100);
    }
    if( digitalRead(parking2_slot2_ir_s) == HIGH)
    {
        sensor4 = "0";
        analogWrite(LED4,0);

        delay(100);
    }
}
```

NODEMCU CODE:

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <SimpleTimer.h>
#include <Servo.h>
#include <TinyGPS++.h>

Servo servo1,servo2,servo3,servo4,servo5,servo6;
static const uint32_t GPSBaud = 9600;
TinyGPSPlus gps;
WidgetTerminal terminal(V3);
const int RXpin=4,TXpin=5;
SoftwareSerial ssgps(RXpin,TXpin); //D1,D2


//char auth[] = "AchgYJbDu5IG_TpbkIbM3C3jcIOY6vau";
char auth[] = "Cj--EuczUexATBFYGngLlTFBh2rdWN0y";
//char auth[] = "7BwXBh4TJs9w0wHyU9My4SE9ARU0mhiz";               // Your Project authentication key

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "shubhamisback";
char pass[] = "hello123";
//char server[]= "192.168.43.22";
SimpleTimer timer;
WidgetMap myMap(V0);

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;
String latv="NULL",lngv="NULL";

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);

}

int selected=0;
String book_status="unbooked";

BLYNK_WRITE(V22)
{
    int Press=param.asInt();
    if(Press==1 && selected!=0){
        opengate(selected);
    }else{
        closegate(selected);
    }
}
BLYNK_WRITE(V21)
{   int Press=param.asInt();
    if(Press==1 && selected!=0){
        //Book
```

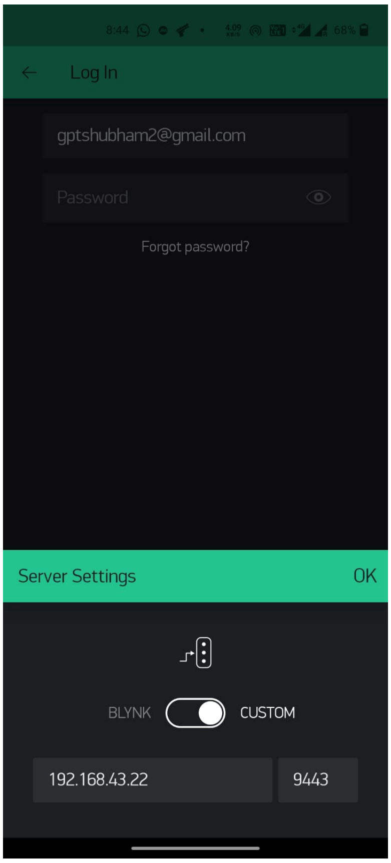
```
      book_status="booked";
    }else{
        //UNBOOK
        book_status="unbooked";
    }
}
BLYNK_WRITE(V20)
{
    selected=param.asInt();
    switch (param.asInt())
    {
        case 1: { // Item 1
            Serial.println("Item 1 selected");
            break;
        }
        case 2: { // Item 2
            Serial.println("Item 2 selected");
            break;
        }
        case 3: { // Item 3
            Serial.println("Item 2 selected");
            break;
        }
        case 4: { // Item 4
            Serial.println("Item 2 selected");
            break;
        }
    }
}
String space="0";
void setup()
{
    Serial.begin(9600);
    Serial.println(F("Working"));
    ssgps.begin(GPSBaud);
    Blynk.begin(auth, ssid, pass,IPAddress(192,168,43,22), 8080);
    timer.setInterval(1000L,sensorvalue1);
    timer.setInterval(1000L,sensorvalue2);
    timer.setInterval(1000L,sensorvalue3);
    timer.setInterval(1000L,sensorvalue4);
    timer.setInterval(1000L,setlat);
    timer.setInterval(1000L,setspace);
    timer.setInterval(1000L,showterminal);

    servo1.attach(15); //D8
    servo2.attach(13); //D7
    servo3.attach(12); //D6
    servo4.attach(14); //D5
    servo5.attach(0); //D3
    servo6.attach(2); //D4
}
String latval="NULL",lngval="NULL";
String entry="closed",entry2="closed";
BLYNK_WRITE(V23){
    if(ssgps.available()>0){
        if(gps.encode(ssgps.read())){
            if(gps.location.isValid()){
                latval=String(gps.location.lat()); lngval=String(gps.location.lng());
                Serial.println("Location:"+latval);
            }else{
                Serial.println("Location: INVALID");
            }

        }
    }
}
void loop()
{

    while(ssgps.available()>0){
        if(gps.encode(ssgps.read())){
            if(gps.location.isValid()){
                latval=String(gps.location.lat()); lngval=String(gps.location.lng());
                Serial.println("Location:"+latval);
            }else{
                Serial.println("Location: INVALID");
            }

        }
    }
    if (Serial.available() == 0 )
    {
        Blynk.run();
        timer.run(); // Initiates BlynkTimer
    }

    if (Serial.available() > 0 )
    {
        rdata = Serial.read();
        myString = myString+ rdata;
        if( rdata == '\n')
        {
            Serial.println(myString);
            String l = getValue(myString, ',', 0);
            String m = getValue(myString, ',', 1);
            String n = getValue(myString, ',', 2);
            String o = getValue(myString, ',', 3);
            space=getValue(myString, ',', 4);
            if(latval!="NULL"){
                latv = latval;
                lngv = lngval;
            }
            if(String(getValue(myString, ',', 5))=="opened" && entry=="closed"){
                openEntryGate();
                entry="opened";
                Serial.print("OPENING!");
            }else if(String(getValue(myString, ',', 5))=="closed" && entry=="opened"){
                closeEntryGate();
                entry="closed";
                Serial.print("CLOSING!");
            }
            if(String(getValue(myString, ',', 6))=="opened" && entry2=="closed"){
                openExitGate();
                entry2="opened";
            }else if(String(getValue(myString, ',', 6))=="closed" && entry2=="opened"){
                closeExitGate();
                entry2="closed";
            }
            led1 = l.toInt();
            led2 = m.toInt();
            led3 = n.toInt();
            led4 = o.toInt();

            int stotal=(led1+led2+led3+led4)/255;
            showterminal();
            myString = "";
            // end new code
        }
    }

}
void showterminal()
{
    if(myString!="")
    {Blynk.virtualWrite(V3, myString);}
}

void openEntryGate(){
    servo5.write(90);
}
void openExitGate(){
    servo6.write(90);
}

void closeEntryGate(){
    servo5.write(0);
}
void closeExitGate(){
    servo6.write(0);
}

void opengate(int n){
    switch(n){
        case 1:servo1.write(90);break;
        case 2:servo2.write(90);break;
        case 3:servo3.write(90);break;
        case 4:servo4.write(90);break;
    }

}
void closegate(int n){
    switch(n){
        case 1:servo1.write(00);break;
        case 2:servo2.write(00);break;
        case 3:servo3.write(00);break;
        case 4:servo4.write(00);break;
    }
}


void setlat(){
    if(latv!="NULL"){Blynk.virtualWrite(V1, String(latv));}
    if(lngv!="NULL"){Blynk.virtualWrite(V2, String(lngv));}
    myMap.location(1, latv.toFloat(), lngv.toFloat(), "GPS_Location");
}
void setspace(){
    Blynk.virtualWrite(V16, String(space));
}
void sensorvalue1()
{
    int sdata = led1;
    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;
    Blynk.virtualWrite(V12, sdata);

}

void sensorvalue4()
{
    int sdata = led4;
    Blynk.virtualWrite(V13, 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]) : "";
}
```

3 USER MANUAL

1. Connect your PC with WIFI Hotspot and insert your credentials in NODEMCU code
2. Upload these codes to ARDUINO UNO and NODEMCU respectiviely
3. Connect all hardwares as showin in cricuit diagram

4. Install BLYNK Application on your smartphone
5. Sign and Register by using YOUR IP ADDRESS and PORT 9443



Log In

gptshubham2@gmail.com

Password

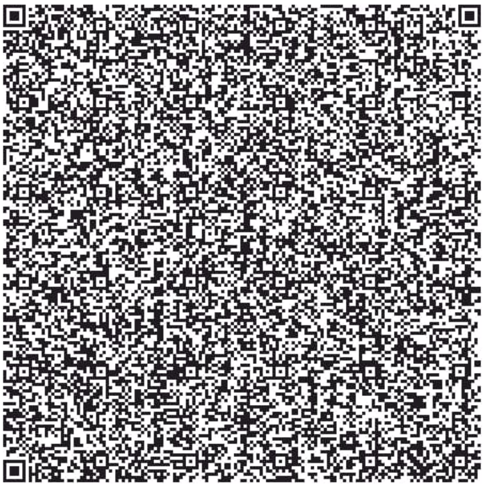
Forgot password?

Server Settings OK

BLYNK CUSTOM

192.168.43.22 9443

6. Select New Project and scan below QR to get this project on BLYNK



7. Run a Blynk Local Server or Upload script to Google Cloud VM Instance
8. BLYNK SERVER CODE can bee seen from here https://github.com/blynkkk/blynk-server
9. Open Serial Monitor to check everything working correctly
10. Create New VM instance in Google Cloud Console and upload github files and run the server
11. Finally Projcet is complete

4 DISCUSSION

Here we demonstrate the use of IOT based parking management system that allows for efficient parking space utilization using IOT technology. To demonstrate the concept, we use IR sensors for sensing parking slot occupancy along with a DC motors to simulate as gate opener motors. We now use a WiFI modem for internet connectivity and an AVR microcontroller for operating the system. The system detects if parking slots are occupied using IR sensors. Also it uses IR technology to sense if a vehicle has arrived on gate for automated gate opening. The system reads the number of parking slots available and updates data with the cloud server to allow for checking parking slot availability online. This allows users/drivers to check for available parking spaces online from anywhere and avail hassle free parking. Thus the system solves the parking issue for cities and get users/drivers an efficient IOT based parking management system.

CONCLUSION

Finally using all those and configuring my hardware I solve this problem and completed this. This shows how a better management can be done easily using IOT and advance modeern systems which can help us in many ways and solving traffic and chaos situations
FOR OUTPUT AND RESULT PLEASE REFER TO THE VIDEO LINK:
https://youtu.be/aX-rREl-w_w

REFERENCES

1. https://www.researchgate.net/publication/326461095_Connected_Car_IoT_Overview
2. https://www.parking-net.com/about-parking/car-parking-techniques
3. https://en.wikipedia.org/wiki/Arduino_Uno
4. https://en.wikipedia.org/wiki/NodeMCU
5. https://en.wikipedia.org/wiki/Internet_of_things
6. https://en.wikipedia.org/wiki/Car_parking_system