A PROJECT REPORT

ON

“IoT Based Car Parking Management System”

BY -

SHUBHAM CHAUHAN 180108040

UNDER THE GUIDANCE OF -

PROF. MANAS KHATUA

COURSE – Internet of Things (CS578)
IIT GUWAHATI

IoT Based Car Parking Management System CS578

Problem Statement:

Now days finding parking in busy areas is very hard and there is no system to get the details of parking availability online. Imagine if you can get the parking slot availability information on your phone and you don’t have roaming around to check the availability. This problem can be solved by the IoT based smart parking system. Using the IoT based parking system you can easily access the parking slot availability over the internet. This system can completely automate the car parking system. From your entry to the payment and exit all can be done automatically.

Objective:

We have to sense the occupancy of the parking slot using IR sensors and send the collected data to the server in the Internet for storing and processing. Entrance of a new car is decided based on the slots available.

Hardware Used:

1- Arduino UNO
2 – Node MCU (WiFi Module)
3 – Breadboard
4 – Jumper Wires
5 – LCD display
6 – Android Phone
7 – Servo motor

Description:

In this project, we will learn how to use the Arduino, Nodemcu esp8266 wifi module and Blynk software to build an IOT-based Car Parking Slots monitoring system. The parking slots can be tracked from anywhere in the world using the Nodemcu esp8266 wifi module and Blynk software.
The parking space is split into two parking spaces.

1 - Parking 1

2 - Parking 2

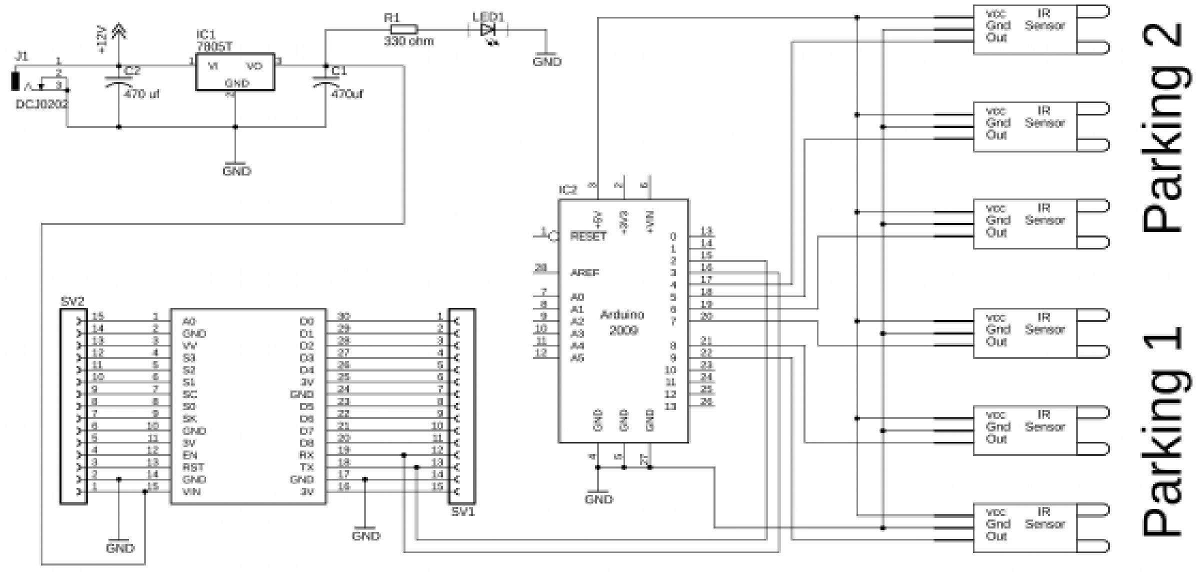
One parking space has 3 slots and the other has 2 slots and each slot has one detector. So we have 5 infrared sensors in all. One sensor is also used to control the servo at the gate. The presence of the car in the slot is detected by each infrared sensor. These sensors are mounted to the Arduino. So when a car is parked in the slot, the infrared sensor is triggered and the command is sent to the wifi module nodemcu esp8226, Nodemcu sends the command to the Blynk request.

Implemented Attributes –

A total of six infrared sensors are used as you can see. All Arduino pins 4 to 9 are attached to sensors. The VCC pins of the infrared sensor and the servo motor are attached to the 5v of the Arduino. Grounds are attached to the ground of the Arduino while pin 4 to 9 connects the out pins of all the infrared sensors. Servo is connected to pin 10 of Arduino.

The Tx and Rx pins of the Nodemcu module are connected to Arduino’s pin2 and pin3. NodeMCU can be powered by the external source as shown below in the diagram. But, I’m going to use two USB cables, one
cable to power the Arduino, and the other USB cable to power the nodemcu.

Configuration Diagram:



Car Parking Arduino Programming:

```
#include <SoftwareSerial.h>

#include<Servo.h>

SoftwareSerial nodemcu(2,3);


int entry = 9;

int out_ = 11;

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 cars = 0;

Servo gate;


String sensor1;

String sensor2;

String sensor3;

String sensor4;

String sensor5;



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


void setup()

{

gate.attach(10);

Serial.begin(9600);

nodemcu.begin(9600);
```

```
pinMode(entry, INPUT);
pinMode(out_, INPUT);
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);
gate.write(0);
delay(1000);

}

void loop()
{
int in = digitalRead(entry);
int out = digitalRead(entry);
int s1 = digitalRead(parking1_slot1_ir_s);
int s2 = digitalRead(parking1_slot2_ir_s);
int s3 = digitalRead(parking1_slot3_ir_s);
int s4 = digitalRead(parking2_slot1_ir_s);
int s5 = digitalRead(parking2_slot1_ir_s);
if(cars == 5){
Serial.println(" PARKING FULL! ");
}
else{
Serial.print("Cars in the parking : ");
Serial.print(cars);
Serial.print("  Empty slots := ");
if(s1 == 1){
Serial.print("Slot 1 ");
}
if(s2 == 1){
Serial.print(“Slot 2 ”);
}
if(s3 == 1){
Serial.print(“Slot 3 ”);
}
if(s4 == 1){
Serial.print(“Slot 4 ”);
}
if(s5 == 1){
Serial.print(“Slot 5 ”);
}
if(s1 == 0 && s2 == 0 && s3 == 0 && s4 == 0 && s5 == 0){
Serial.print(“ NOT AVAILABLE! “);
}
Serial.println();
if(in == 0 && cars < 5){
cars = cars + 1;
gate.write(90);
delay(3000);
gate.write(0);
}
if(out == 0 && cars > 0){
cars = cars - 1;
delay(1000);
}

p1slot1();
p1slot2();
p1slot3();
```

```
p2slot1();
p2slot2();
   cdata = cdata + sensor1 +"," + sensor2 + ","+ sensor3 +","+ sensor4 + "," + sensor5 + ",";// + sensor6 +","; comma will be used as a delimiter.
   Serial.println(cdata);
   nodemcu.println(cdata);
   delay(6000); // 6 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);

}



void p1slot1() // parking 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 2 slot1
{
  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 2 slot2
{
  if( digitalRead(parking2_slot2_ir_s) == LOW)
  {
  sensor5 = "255";
  delay(200);
  }
if( digitalRead(parking2_slot2_ir_s) == HIGH)
  {
  sensor5 = "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[] = "entX5XzysOv1SYX6twDrVQNvOOrHn1JH";

// Your WiFi credentials.
char ssid[] = "sam";
char pass[] = "12345678";

SimpleTimer timer;

String myString;  // complete message from arduino, which consists of sensors data
char rdata;       // received characters

int firstVal, secondVal, thirdVal; // sensors
int led1,led2,led3,led4,led5;
// This function sends Arduino's up time every second to Virtual Pin (1).
// 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 than 10 values per second.
  Blynk.virtualWrite(V1, millis() / 1000);

}
```

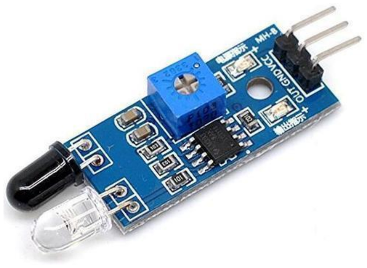
```
void setup()
{

  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);

}

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);

// 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);



// these leds represents the leds used in Blynk application
led1 = l.toInt();
led2 = m.toInt();
led3 = n.toInt();
led4 = o.toInt();
led5 = p.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);

}
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]) : "";
}
```

USER MANUAL -

About the IR sensor Module:

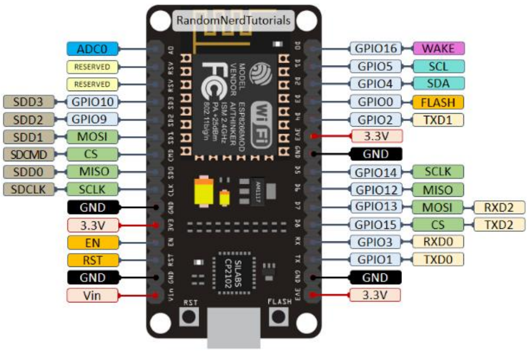
I used IR detector for car detection. As you can see, the VCC, GND, and OUT are clearly labelled with the three male headers. The 5 volt pin of Arduino is attached to the Vcc. The field is connected to the ground of Arduino. While the OUT pin is connected to the IO pins of Arduino. What I'm going to explain in the diagram. In IR there are two black and white LEDs, one for tx and the other for RX.



IR sensor module

About the Nodemcu ESP8266 WIFI Module:

Nodemcu ESP8266 wifi unit, we can track the parking slots from anywhere in the world from this module. As you can see from the pic below, all the pins are clearly labeled. Never use 5 volts of the Arduino to power the Nodemcu esp8266 wifi module. If you use the 5 volt Arduino to power up this module, it will continue to reset this wifi device. To solve this problem, you can use the LM7805 voltage regulator to design a separate power supply for this module. You can also power it using your laptop.



ESP8266 pinout

IoT based Car Parking Blynk Application:

You will need to download the Blynk software from the App Store for the development of the android or apple cell
phone app. Upon installing the Blynk software, you will need to use Facebook or any other email ID to register for free.

REFERENCES –

https://programmingdigest.com/iot-based-car-parking-system-application-designing-and-nodemcu-esp8266-programming/