**UNIVERSITY COLLEGE OF ENGINEERING PANRUTI**

**CUDDALORE-607106, TAMILNADU**

**IN ELECTRONICS AND COMMUNICATION ENGINEERING**

**IOT-BASED ELECTRIC VEHICLE AUTO PARKING SYSTEM**

2023-2024

SUBMITTED BY

|  |  |  |
| --- | --- | --- |
| 1 | Sowmidra L.R | 422621106036 |

GUIDED BY

**Ms.M.REHANA M.SC (PHYSICS)**



**UNIVERSITY COLLEGE OF ENGINEERING PANRUTI**

**CUDDALORE-607106, TAMILNADU**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**BONAFIDE CERTIFICATE**

Certified that this Business plan report on “**IOT- BASED ELECTRIC VEHICLE AUTO PARRKING SYSTEM REPORT”** is the bonafide record of work done by ----------------- -------------------------------------------------------------practical fulfillment for the award of “DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING ” by Anna University , Chennai.

Submitted for the board examination held on

Internal Guide Head of the department

ii

**ACKNOWLEDGEMENT**

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express the gratitude to the people who have been instrumental in the successful completion of this Business plan.

We express our thanks and gratitude to our beloved dean **Dr.S,Muthukumaran** and our department stafs, for extending his support and permit us to do this business plan.

We are highly indebted to **Mr.D.Illakiyaselvan** for their guidance and constant supervision as well as for providing necessary information regarding the business plan & also for their support in completing the report.

We also express our sincere thanks to our esteemed under guided Naan Muthalvan **Ms.M.REHANA** for his valuable suggestions while doing this report.

We would like to express our gratitude to all staffs of UNIVERSITY COLLEGE OF ENGINEERING PANRUTIfor their kind co-operation and encouragement which help us in completion of this report.

iii

Finally, yet important, we would like to express our heartfelt thanks to our beloved parents for their blessings, our classmates for their help and wishes for the successful completion of this business plan report.

**IOT-BASED ELECTRIC VEHICLE AUTO PARKING SYSTEM**

**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 paper 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. The parking information is sent to the user via notification. Thus, the waiting time for the user in search of parking space is minimized. This project requires components IR sensor, esp8266, servo motor and Adafruit app.

**INTRODUCTION:**

We are building an IoT based car parking system using NodeMCU , five IR sensors , and two servo motors . Two IR sensors are used at entry and exit gate to detect the car while three IR sensor sensors are used to detect the parking slot availability . Servo motors are used to open and close the gates according to the sensor value . Here we are using the Adafruit IO platform to show publish the data on cloud which can be monitored from anywhere in the world.

NodeMCUserves as the central control unit of the system. It's a microcontroller board based on the ESP8266 Wi-Fi module, allowing it to connect to the internet and communicate with other devices.IR sensors Entry and Exit Gates: Two IR sensors are placed at the entry and exit gates to detect the presence of a car approaching or leaving. When a car is detected, the sensor sends a signal to the NodeMCU. Parking slot AvailabilityThree IR sensors are strategically placed within the parking area to monitor individual parking slots. They detect whether a slot is occupied by a vehicle or vacant. Servo Mot**ors** Two servo motors control the entry and exit gates . Based on the signals received from the IR sensors , the NodeMCU commands the servo motors to open or close the gates accordingly , allowing or restricting vehicle access. Adafruit IO Platform This cloud based platform provides a user-friendly interface for monitoring and controlling the parking system remotely from anywhere with an internet connect.The NodeMCU sends data to the Adafruit IO platform , including information about parking slot availability (occupied or vacant )and the status of entry and exit gates (open or close).User can access the Adafruit IO dashboard to view real-time data about parking availability and gate status . They can also receive alerts or notifications regarding any system events or anomalies.

**COMPONENTS:**

1. IR sensor
2. ESP8266
3. Servomotor
4. Connecting wires
5. Adafruit IO
6. Arduino IDE

**IR Sensor:** This is the IR sensor which will be using for the cars

detection. As show in diagram the three male headers are clearly

labeled with the VCC, GND, and OUT. The VCC pin is

connected with the Arduino’s 5 volts. The ground is connected

with the Arduino’s ground. While the OUT pin is connected with

Arduino’s IO pins. which will explain in the circuit diagram.

While the black and white leds are the IR leds “one is the Tx while the other one is the Rx”.

while the other one is the Rx”.

**Nodemcu ESP8266:** This is the Nodemcu ESP8266 wifi module,

with the help of this module we can monitor the car parking slots

from anywhere around the world. As you can see clearly all the

pins are clearly labeled. Never power up the Nodemcu esp8266

wifi module using the Arduino’s 5 volts. If you power up this

module using the arduino’s 5 volt then this wifi module we will

keep reseting. To solve this problem you can design a separate

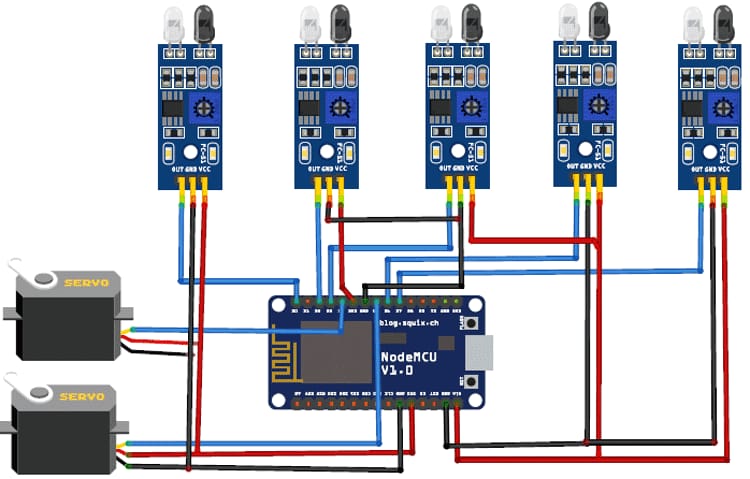
power supply of this module using the LM7805 voltage regulator.

**Adafruit.io :** It is needed to display the IOT project's data online in real-time. It isa cloud server that can be used to connect to IoT devices through wifi and to control these devices through a dashboard by creating qa feeds.Here we use two buttons (Entry &Exit button)and 7 text box (entry time slot1,2, and 3& exit time slot 1,2 &3).

**Servo motor:**

A servo motor, or simply, a servo, is a device that is used to rotate or push parts of a machine to which it Is connected with precision. Unlike DC motors, they generally rotate to a particular angle and then stop.

**CIRCUIT DIAGRAM:**

****

The above circuit diagram represents that the two servo motors are connected to D4 and D5 pin. Then the entry and exit gate sensors are connected to D0 and D2 pins .and the other 3 sensors are connected to the D2, D6 and D7 pins in the Nodemcu. The VCC and GND terminals of all components are connected to the 3V and GND pins in the Nodemcu.

**Arduino code:**

#include <ESP8266WiFi.h>

#include <Servo.h>

#include <NTPClient.h>

#include <WiFiUdp.h>

#include <NTPClient.h>

#include <WiFiUdp.h>

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

const char \*ssid =  "Airtel\_Hemalatha kannan";     // Enter your WiFi Name

const char \*pass =  "Yash@2006"; // Enter your WiFi Password

#define MQTT\_SERV "io.adafruit.com"

#define MQTT\_PORT 1883

#define MQTT\_NAME "L\_R\_Sowmidra"

#define MQTT\_PASS "aio\_pbgx44rYxq53CWFYlsJPXhVjPXoY"

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800,60000);

Servo myservo;                          //servo as gate

Servo myservos;                               //servo as gate

int carEnter = D0;                      // entry sensor

int carExited = D2;                   //exi sensor

int slot3 = D7;

int slot2 = D6;

int slot1  = D3;

int count =0;

int CLOSE\_ANGLE = 80;  // The closing angle of the servo motor arm

int OPEN\_ANGLE = 0;  // The opening angle of the servo motor arm

int  hh, mm, ss;

int pos;

int pos1;

String h, m,EntryTimeSlot1,ExitTimeSlot1, EntryTimeSlot2,ExitTimeSlot2, EntryTimeSlot3,ExitTimeSlot3;

boolean entrysensor, exitsensor,s1,s2,s3;

boolean s1\_occupied = false;

boolean s2\_occupied = false;

boolean s3\_occupied = false;

WiFiClient client;

Adafruit\_MQTT\_Client mqtt(&client, MQTT\_SERV, MQTT\_PORT, MQTT\_NAME, MQTT\_PASS);

//Set up the feed you're subscribing to

Adafruit\_MQTT\_Subscribe EntryGate = Adafruit\_MQTT\_Subscribe(&mqtt, MQTT\_NAME "/f/EntryGate");

Adafruit\_MQTT\_Subscribe ExitGate = Adafruit\_MQTT\_Subscribe(&mqtt, MQTT\_NAME "/f/ExitGate");

//Set up the feed you're publishing to

Adafruit\_MQTT\_Publish CarsParked = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/CarsParked");

Adafruit\_MQTT\_Publish EntrySlot1 = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/EntrySlot1");

Adafruit\_MQTT\_Publish ExitSlot1 = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/ExitSlot1");

Adafruit\_MQTT\_Publish EntrySlot2 = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/EntrySlot2");

Adafruit\_MQTT\_Publish ExitSlot2 = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/ExitSlot2");

Adafruit\_MQTT\_Publish EntrySlot3 = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/EntrySlot3");

Adafruit\_MQTT\_Publish ExitSlot3 = Adafruit\_MQTT\_Publish(&mqtt,MQTT\_NAME "/f/ExitSlot3");

void setup() {

  delay(1000);

  Serial.begin (9600);

  mqtt.subscribe(&EntryGate);

  mqtt.subscribe(&ExitGate);

  timeClient.begin();

  myservo.attach(D4);      // servo pin to D4

  myservos.attach(D5);       // servo pin to D5

  pinMode(carExited, INPUT);    // ir as input

  pinMode(carEnter, INPUT);     // ir as input

  pinMode(slot1, INPUT);

  pinMode(slot2, INPUT);

  pinMode(slot3, INPUT);

  WiFi.begin(ssid, pass);                                     //try to connect with wifi

  Serial.print("Connecting to ");

  Serial.print(ssid);                          // display ssid

  while (WiFi.status() != WL\_CONNECTED) {

    Serial.print(".");                          // if not connected print this

    delay(500);

  }

  Serial.println();

  Serial.print("Connected to ");

  Serial.println(ssid);

  Serial.print("IP Address is : ");

  Serial.println(WiFi.localIP());                                            //print local IP address

}

void loop() {

 MQTT\_connect();

 timeClient.update();

 hh = timeClient.getHours();

 mm = timeClient.getMinutes();

 ss = timeClient.getSeconds();

 h= String(hh);

 m= String(mm);

 h +" :" + m;

 entrysensor= !digitalRead(carEnter);

 exitsensor = !digitalRead(carExited);

 s1 = digitalRead(slot1);

 s2 = digitalRead(slot2);

 s3 = digitalRead(slot3);

   if (entrysensor == 1) {                     // if high then count and send data

   count=  count+1;                             //increment count

   myservos.write(OPEN\_ANGLE);

   delay(3000);

   myservos.write(CLOSE\_ANGLE);

   }

   if (exitsensor == 1) {                            //if high then count and send

   count= count-1;                                    //decrement count

   myservo.write(OPEN\_ANGLE);

   delay(3000);

   myservo.write(CLOSE\_ANGLE);

  }

  if (! CarsParked.publish(count)) {}

  if (s1 == 1 && s1\_occupied == false) {

        Serial.println("Occupied1 ");

        EntryTimeSlot1 =  h +" :" + m;

        //Serial.print("EntryTimeSlot1");

        //Serial.print(EntryTimeSlot1);

        s1\_occupied = true;

        if (! EntrySlot1.publish((char\*) EntryTimeSlot1.c\_str())){}

    }

  if(s1 == 0 && s1\_occupied == true) {

       Serial.println("Available1 ");

       ExitTimeSlot1 =  h +" :" + m;

       //Serial.print("ExitTimeSlot1");

       //Serial.print(ExitTimeSlot1);

       s1\_occupied = false;

        if (! ExitSlot1.publish((char\*) ExitTimeSlot1.c\_str())){}

}

  if (s2 == 1&& s2\_occupied == false) {

      Serial.println("Occupied2 ");

      EntryTimeSlot2 =  h +" :" + m;

      //Serial.print("EntryTimeSlot2");

      //Serial.print(EntryTimeSlot2);

      s2\_occupied = true;

      if (! EntrySlot2.publish((char\*) EntryTimeSlot2.c\_str())){}

    }

  if(s2 == 0 && s2\_occupied == true) {

       Serial.println("Available2 ");

       ExitTimeSlot2 =  h +" :" + m;

       //Serial.print("ExitTimeSlot2");

       //Serial.print(ExitTimeSlot2);

       s2\_occupied = false;

        if (! ExitSlot2.publish((char\*) ExitTimeSlot2.c\_str())){}

  }

  if (s3 == 1&& s3\_occupied == false) {

      Serial.println("Occupied3 ");

      EntryTimeSlot3 =  h +" :" + m;

     //Serial.print("EntryTimeSlot3: ");

      //Serial.print(EntryTimeSlot3);

      s3\_occupied = true;

       if (! EntrySlot3.publish((char\*) EntryTimeSlot3.c\_str())){}

    }

  if(s3 == 0 && s3\_occupied == true) {

       Serial.println("Available3 ");

       ExitTimeSlot3 =  h +" :" + m;

       //Serial.print("ExitTimeSlot3: ");

       //Serial.print(ExitTimeSlot3);

       s3\_occupied = false;

        if (! ExitSlot3.publish((char\*) ExitTimeSlot3.c\_str())){ }

  }

  Adafruit\_MQTT\_Subscribe \* subscription;

  while ((subscription = mqtt.readSubscription(5000)))

     {

   if (subscription == &EntryGate)

     {

      //Print the new value to the serial monitor

      Serial.println((char\*) EntryGate.lastread);

   if (!strcmp((char\*) EntryGate.lastread, "ON"))

      {

       myservos.write(OPEN\_ANGLE);

       delay(3000);

       myservos.write(CLOSE\_ANGLE);

    }

}

  if (subscription == &ExitGate)

     {

      //Print the new value to the serial monitor

      Serial.println((char\*) EntryGate.lastread);

   if (!strcmp((char\*) ExitGate.lastread, "ON"))

      {

       myservo.write(OPEN\_ANGLE);

       delay(3000);

       myservo.write(CLOSE\_ANGLE);

    }

}

}

}

void MQTT\_connect()

{

  int8\_t ret;

  // Stop if already connected.

  if (mqtt.connected())

  {

    return;

  }

  uint8\_t retries = 3;

  while ((ret = mqtt.connect()) != 0) // connect will return 0 for connected

  {

       mqtt.disconnect();

       delay(5000);  // wait 5 seconds

       retries--;

       if (retries == 0)

       {

         // basically die and wait for WDT to reset me

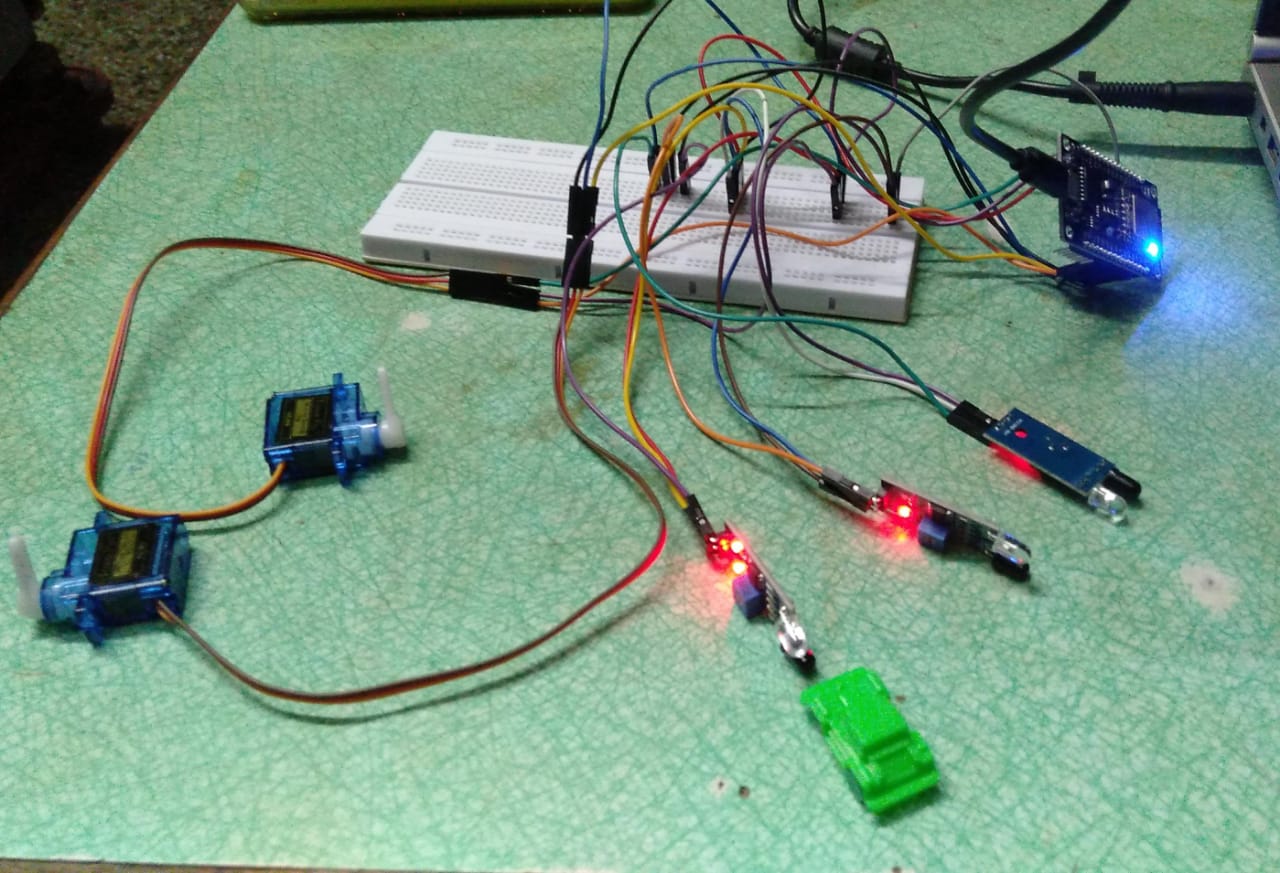
         while (1);

       }

  }

}

**RESULT:**



**CONCLUSION:**

In conclusion, the IoT-based car parking system we are developing integrates NodeMCU, five IR sensors, two servo motors, and the Adafruit IO platform to create a comprehensive solution for efficient parking management.

The system utilizes IR sensors placed at entry and exit gates to detect incoming and outgoing vehicles, as well as within the parking area to monitor parking slot availability. This information is relayed to the NodeMCU, which controls servo motors to open and close gates accordingly.

By leveraging the Adafruit IO platform, data collected by the system is published to the cloud, enabling users to remotely monitor parking availability and gate status from anywhere in the world. This enhances convenience for both drivers and parking administrators, facilitating better utilization of parking spaces and smoother traffic flow.

Overall, the IoT-based car parking system offers a scalable and adaptable solution for modern parking management needs, combining hardware components with cloud-based monitoring and control capabilities for seamless operation and user convenience.