BOT-IOT PROJECT

FINAL PROJECT SUBMITTED FOR ELECTRICAL AND ELECTRONICS SOCIETY

SUMMER PROJECT PROGRAMME 2022

BY

ANKITA AYUSHI (BTECH/10360/21)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
BIRLA INSTITUTE OF TECHNOLOGY
MESRA, RANCHI-835215 (INDIA)

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
BIRLA INSTITUTE OF TECHNOLOGY
MESRA, RANCHI-835215 (INDIA)

CERTIFICATE

This is to certify that the contents of the report entitled " BOT-IOT PROJECT " is a bonafide work carried out by Ms. *Ankita Ayushi*. The contents of the report have not been submitted earlier for the award of any other degree or certificates and we hereby commend the work done by her in this connection.

ACKNOWLEDGEMENT

I express deep regards to my project mentor **Aditya Raj** and **Ayushman Dutta** under whose guidance I was able to learn and apply the concepts presented in this project. Their consistent supervision inspired me to study deeply for this project and in carrying out this project work with success.

I am very grateful to **Dr. S Shiva Kumar**, Advisor of Electrical and Electronics Engineering Society (EEESoc), Birla Institute of Technology for providing me with a platform to learn and apply concepts to build something new through **Summer Project Programme** organized by **EEESoc, BIT Mesra.**

I am very grateful to **Dr. Tirthadip Ghose,** Professor and Head, Department of Electrical and Electronics Engineering, Birla Institute of Technology for providing facilities and opportunities for participating in **Summer Project Programme**.

I am also thankful to **Prof. (Dr.) Indranil Manna,** Vice Chancellor, Birla Institute of Technology, Mesra, Ranchi for providing important facilities, required for carrying out this project.

Ankita Ayushi
(BTECH/10360/21)

TABLE OF CONTENTS

TOPICS

ABSTRACT

Internet of things is a system of interrelated devices , machines, objects or animals which are provided with unique identifiers (UIDs) and ability to transfer data over a network without any requirement of human – to – human or human-to machine interaction. It describes objects and devices with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks**.**

IOT is seen as a major revolution in the current trend of technologies , finding application in all spheres . Some examples include smart lighting system , traffic management , home automation , pollution monitoring and reporting , smart parking system etc.

A microcontroller is basically a small computer on single VLSI integrated circuit chip used for controlling other portions of an electronic system, usually via a microprocessor unit (MPU), memory, and some peripherals. It consists of a central processing unit (CPU), non volatile memory, volatile memory, peripherals, and support circuitry.

INTRODUCTION

In this project , a small IOT Based car was built using an open source IOT platform , NodeMCU and Mobile application , Blynk IOT .Wifi is used for communication between the Bot and the mobile application. A motor driver is used to power the four TT Gear motors . Two 3.7 V Li-Ion cells are used to power the motor driver which further supplies power to the NodeMCU.

ANALYSIS

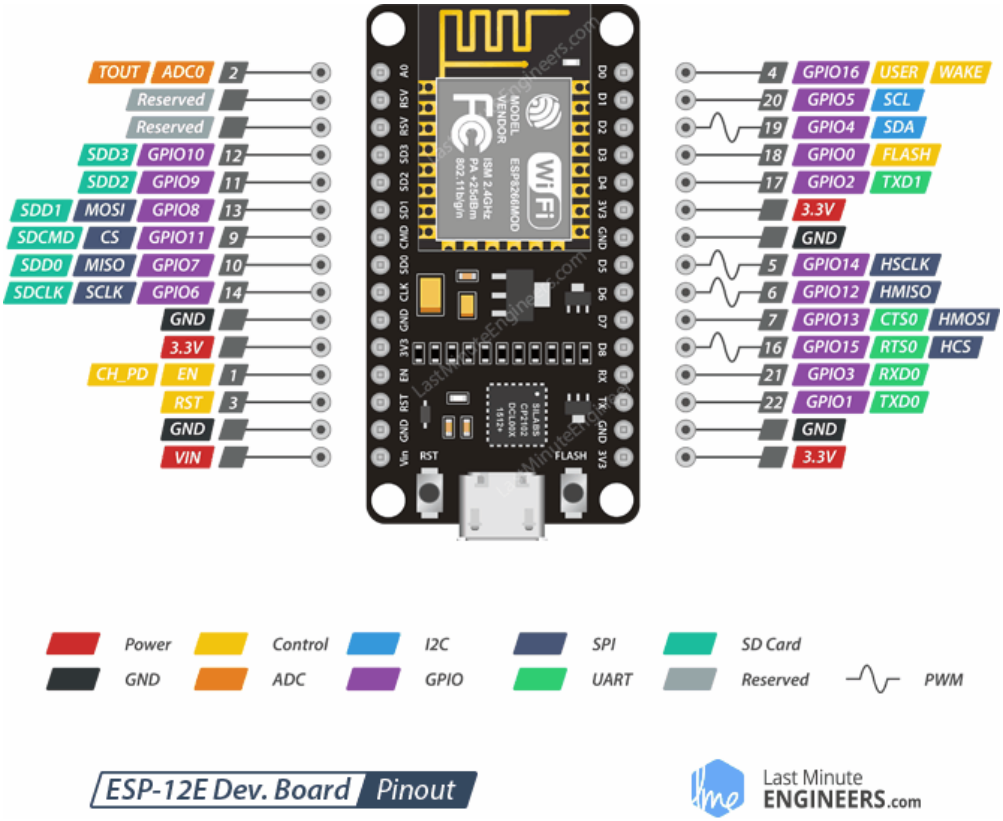
The NodeMCU-ESP8266 Wifi development board was connected to the L298N Motor driver . The motor driver was powered by 2 Li-Ion cells , which further powered the NodeMCU board .The left and right TT Gear motors of the car were attached to the left and right motor sections of motor driver .The code written in Arduino IDE was uploaded in the NodeMCU using USB cable from the laptop . Blynk App was set up for the control of the wifi car .

COMPONENTS

HARDWARE

1) NodeMCU Wifi Development Board – ESP8266

It is an open source firmware and development kit which
can connect objects and let data transfer using the Wi-Fi protocol.
In addition, it provides some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc.It can be easily programmed using Lua programming language or Arduino IDE.



It has 17 GPIO pins(11 are Digital I/O pins), out of which one pin is an analog pin, 4 pins support PWM, 2 pairs are for UART(UART0 and UART1), and supports 1x SPI and 1x I2C protocol, 128Kb of Ram, 4 MB of Flash memory, and a maximum clock speed of 160MHz(80 -160). The operating current is 80mA(average).

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver chip for WiFi connectivity .

The various pins and their functions are mentioned below:

- GPIO4 labelled as D2 often used as SDA (I2C)
- GPIO0 labelled as D3 connected to FLASH button, boot fails if pulled LOW
- GPIO2 labelled as D4 connected to on-board LED, boot fails if pulled LOW – HIGH at Boot time
- GPIO14 labelled as D5 SPI (SCLK)
- GPIO12 labelled as D6 SPI (MISO)
- GPIO13 labelled as D7 SPI (MOSI)
- ADO labelled as AO
- GPIO16 labelled as D0 HIGH at boot used to wake up from deep sleep
- GPIO15 labelled as D8 Pulled to GND : Boot fails if we pulled HIGH

- GPIO3 labelled as RX HIGH at boo
- GPIO1 labelled as TX debug output at boot, boot fails if pulled LOW

- GPIO5 labelled as D1 often used as SCL (I2C)

- RST pin: When we set the RST pin is low, the ESP8266 resets.

2) L298N Motor Driver

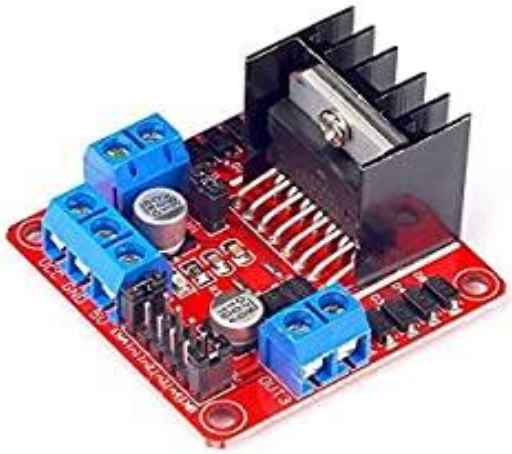
The L298N motor driver can control the speed and spinning direction of the two motors. In addition, it can control a bipolar stepper motor.
Control of the DC motors is achieved in two ways:

1) Controlling the speed using PWM technique
2) Controlling the direction using H Bridge

PWM is a technique in which the average value of the input voltage is adjusted by sending a series of ON-OFF pulses. This average voltage is proportional to the width of the pulses, which is referred to as the Duty Cycle.

The higher the duty cycle, the higher the average voltage applied to the DC motor, resulting in an increase in motor speed. The shorter the duty cycle, the lower the average voltage applied to the DC motor, resulting in a decrease in motor speed.

The spinning direction of a DC motor can be controlled by changing the polarity of its input voltage. An H-bridge circuit is made up of four switches arranged in a H shape, with the motor in the center. Closing two specific switches at the same time reverses the polarity of the voltage applied to the motor. This causes a change in the spinning direction of the motor.



3) JUMPER WIRES AND BREADBOARD

Jumper wires are used for connections between all the components and breadboard is used for solderless connection.

4) 3.7 V Li ion cells

The cells were used for power supply

5) TT Gear Motor and Wheels

Four TT Gear motors were used for rotation of the wheels and mobilisation of the car .

SOFTWARE

1) ARDUINO IDE

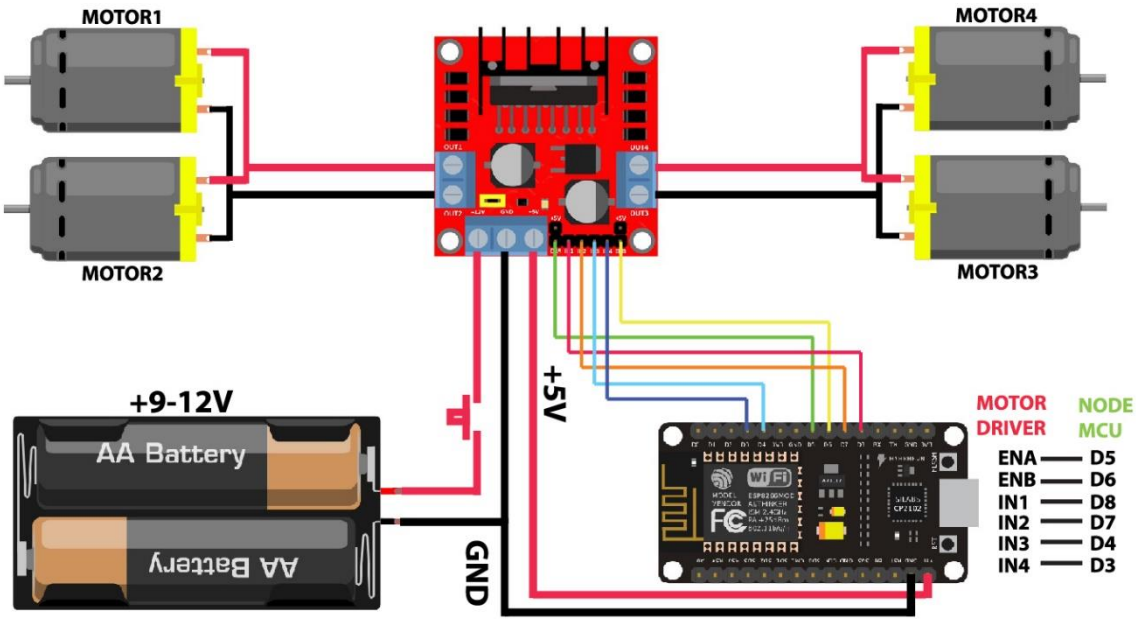
The open-source **Arduino** Software (**IDE**) was used to write code and upload it to the NodeMCU board. It is basically an open-source electronic prototyping platform which enables users to create interactive electronic objects.

2) BLYNK APP

It is an IOT Based platform , an android based app , which helps in interacting with the hardware devices.

METHODOLOGY AND IMPLEMENTATION

CIRCUIT CONNECTIONS



CODE :

Following code was used to program the NodeMCU board.

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod... | Debug - Debugging is not supported by 'NodeMCU 1.0 (ESP-12E Module)'

Final_code.ino

```
// Get the joystick values
BLYNK_WRITE(V1) {
  x = param[0].asInt();
}
// Get the joystick values
BLYNK_WRITE(V0) {
  y = param[0].asInt();
}
// Check these values using the IF condition
void WIFI_car() {
  if (x ==1) {
    carForward();
    Serial.println("carForward");
  } else if (x ==-1) {
    carBackward();
    Serial.println("carBackward");
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod...

Final_code.ino

```
  } else if (y ==-1) {
    carLeft();
    Serial.println("carLeft");
  } else if (y ==1) {
    carRight();
    Serial.println("carRight");
  } else if (x ==0 && y==0) {
    carStop();
    Serial.println("carstop");
  }
  else if (x==1 && y==1){
    carForwardRight();
    Serial.println("car forward right");
  }
   else if (y==-1 && x==1){
    carForwardLeft();
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod...

Final_code.ino

```
    Serial.println("car forward left");
  }
   else if (y==-1 && x==-1){
    carBackwardLeft();
    Serial.println("car backward left");
  }
   else if (y==1 && x==-1){
    carBackwardRight();
    Serial.println("car backward right");
  }
}
void loop() {
  Blynk.run();// Run the blynk function
  WIFI_car();// Call the main function

}
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod...

Final_code.ino

```

/*****Motor movement functions******/
void carForward() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}
void carBackward() {
  analogWrite(ENA, Speed);
  analogWrite(ENB, Speed);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod...

Final_code.ino

```
116   digitalWrite(IN4, HIGH);
117  }
118  void carLeft() {
119   analogWrite(ENA, Speed);
120   analogWrite(ENB, Speed);
121   digitalWrite(IN1, HIGH);
122   digitalWrite(IN2, LOW);
123   digitalWrite(IN3, HIGH);
124   digitalWrite(IN4, LOW);
125  }
126  void carRight() {
127   analogWrite(ENA, Speed);
128   analogWrite(ENB, Speed);
129   digitalWrite(IN1, LOW);
130   digitalWrite(IN2, HIGH);
131   digitalWrite(IN3, LOW);
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod... Upload

Final_code.ino

```
132   digitalWrite(IN4, HIGH);
133  }
134
135  void carStop() {
136   digitalWrite(IN1, LOW);
137   digitalWrite(IN2, LOW);
138   digitalWrite(IN3, LOW);
139   digitalWrite(IN4, LOW);
140  }
141   void carForwardRight(){
142    analogWrite(ENA, Speed);
143    analogWrite(ENB, Speed/2);
144    digitalWrite(IN1, LOW);
145    digitalWrite(IN2, HIGH);
146    digitalWrite(IN3, HIGH);
147    digitalWrite(IN4, LOW);
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod... Verify

Final_code.ino

```
149   }
150   void carForwardLeft(){
151    analogWrite(ENA, Speed/4);
152    analogWrite(ENB, Speed);
153    digitalWrite(IN1, LOW);
154    digitalWrite(IN2, HIGH);
155    digitalWrite(IN3, HIGH);
156    digitalWrite(IN4, LOW);
157
158   }
159  void carBackwardRight() {
160   analogWrite(ENA, Speed);
161   analogWrite(ENB, Speed/4);
162   digitalWrite(IN1, HIGH);
163   digitalWrite(IN2, LOW);
164   digitalWrite(IN3, LOW);
```

Final_code | Arduino IDE 2.0.0

File Edit Sketch Tools Help

NodeMCU 1.0 (ESP-12E Mod...

Final_code.ino

```
159  void carBackwardRight() {
160   analogWrite(ENA, Speed);
161   analogWrite(ENB, Speed/4);
162   digitalWrite(IN1, HIGH);
163   digitalWrite(IN2, LOW);
164   digitalWrite(IN3, LOW);
165   digitalWrite(IN4, HIGH);
166  }
167  void carBackwardLeft() {
168   analogWrite(ENA, Speed/4);
169   analogWrite(ENB, Speed);
170   digitalWrite(IN1, HIGH);
171   digitalWrite(IN2, LOW);
172   digitalWrite(IN3, LOW);
173   digitalWrite(IN4, HIGH);
174  }
```

CONCLUSION

A Wifi controlled bot was built successfully using IOT technology via. Hardware components NodeMCU development board and controlled using Blynk IOT Platform .Its movements were successfully recorded.

FUTURE SCOPE

1) Robotic Arms can be attached to the Bot for further handling of packages and items in warehouse based places.
2) A camera can be attached to the bot for security and detection measures.
3) RFID and Barcode/QR Code scanningTechnology can be implemented for detection and recognition of packages in warehouses , supply chain management system .
4) The simple wifi based bot can further be modified to a voice controlled bot.
5) The bot can further be converted in a bomb diffuser bot with the use of robotic arms and camera .

REFERENCES

- https://www.arduino.cc/en/Tutorial/HomePage
- https://www.youtube.com/watch?v=zJnDbdefeCA
- https://blynk.io/en/getting-started
- https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/
- https://lastminuteengineers.com/