

Design and Development of a IOT based Obstacle Avoider Robot for Residential Applications

1Bhavya B. Agrawal, 2Rydam M. Goyal
1Grade X, IGCSE, 2Grade X, IGCSE
J.G. International School , Ahmedabad, India

Abstract - This projects shows the methodology to develop a wireless IOT based robot for residential applications. The proposed robot has four DC brushed motors, an ultrasonic sensor , NODEMCU and LIPO 11.1 volt battery. The proposed system has inbuilt anti collision algorithm along with the speed control options. The robot can be controlled remotely anywhere in the world, due to its IOT technology. In this robot , 3 rd party server BLYNK is used to design user interface of the robot. In a nut shell, many residential applications can be developed using this methodology.

keywords - Internet of Things, NODEMCU, BLYNK Application, Robotics, Obstacle Avoider Robot

I. INTRODUCTION

The Aim of the project is to develop IOT based mobile robot for material handling applications and residential applications.

1.1 Block Diagram of the Project

In figure 1.1 block diagram of the system is shown, where ultrasonic sensor and android BLYNK applications are the inputs to the system. NODEMCU gives output to the DC motors via L298d Motor driver. It should be noted that, pulse width modulation is activated in this robotic system to vary the speed of the motors [1][2]. Table 1 shows the Specifications of the Robot.

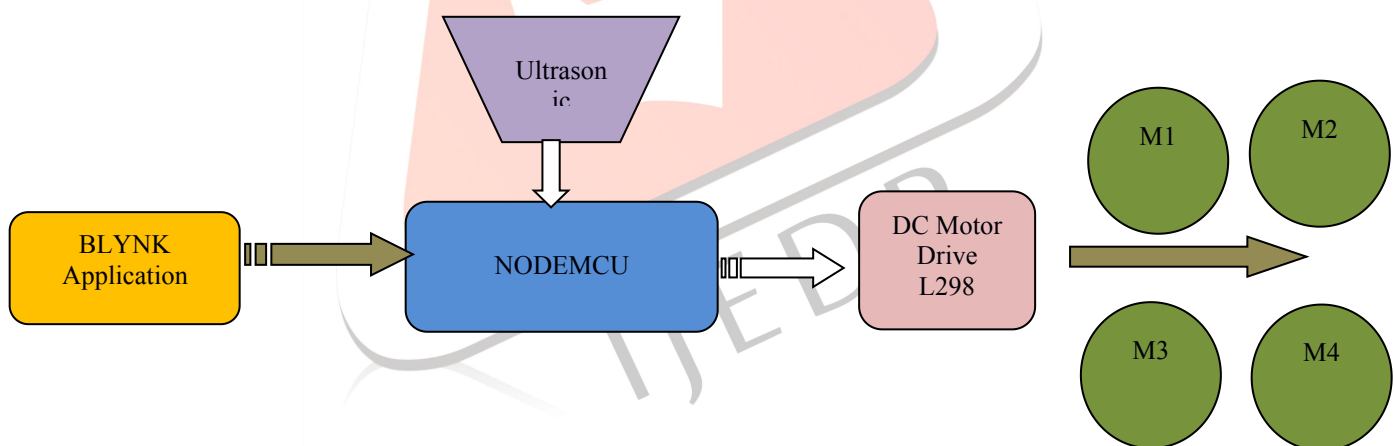


Figure 1.1 Block Diagram of the system

1.2 Objectives

- To make a IOT enabled Robot.
- Multiple Speed Control Option at each motor.
- Obstacle / Anti collision Sensor
- Operation using Android application/ IOS
- Robot should be able to interact with user via Cloud/ server.

1.3 Methodology

- Pulse width Modulation for Speed Control
- Voltage level control for converting logic of 3.3 volts into 5 volts and vice versa [3]
- Robot Chassis is designed using laser cutting machine.
- Programming is done using Arduino IDE.
- Ultrasonic sensor is programmed using Square wave principles (ECHO fro receiving and TRIG for generating square waves.

1.4 Advantages and Applications

1.6 Specifications of the robot

Table 1 Specification of the robot

Sr.No.	Specifications		
	Particulars	Qty.	Type
1	DC Motors	4	Brushed 200 RPM 100 mA at No load 200 mA at Full load
2	Ultrasonic Sensor	1	30-500 KHz
3	NODEMCU ESP IOT Controller	1	12 E V1.1
4	Robot Chassis	1	Laser Cut , 4 mm Acrylic

II. NODEMCU & BLYNK APPLICATION

2.1 Introduction to NODEMCU

General-purpose input/output (GPIO) is a pin on an IC (Integrated Circuit). It can be either an input pin or output pin, whose behavior can be controlled at the run time [3]. NodeMCU Development kit provides access to these GPIOs of ESP8266. The only thing to take care of is that NodeMCU Devkit pins are numbered differently than internal GPIO notations of ESP8266 as shown in the below figure and table. For example, the D0 pin on the NodeMCU Devkit is mapped to the internal GPIO pin 16 of ESP8266 [4]. Figure 2.1 Shows the Pin diagram of the NODEMCU board.

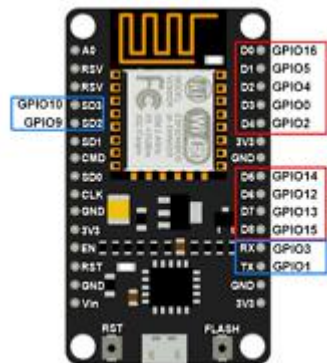


Figure 2.1 NODEMCU Pin Diagram

2.2 Introduction to BLYNK Application

Blynk is a new platform that allows you to quickly build interfaces for controlling and monitoring your hardware projects from your iOS and Android device. After downloading the Blynk app, you can create a project dashboard and arrange buttons, sliders, graphs, and other widgets onto the screen. Using the widgets, you can turn pins on and off or display data from sensors [5]. Example of Blynk application is shown in Figure 2.2.

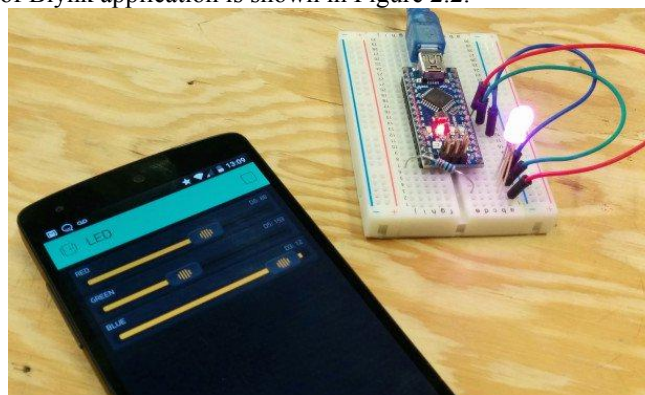


Figure 2.2 Working with BLYNK application to Control LEDs using Arduino Nano

III. CONSTRUCTION OF THE ROBOT

3.1 Chassis Assembly

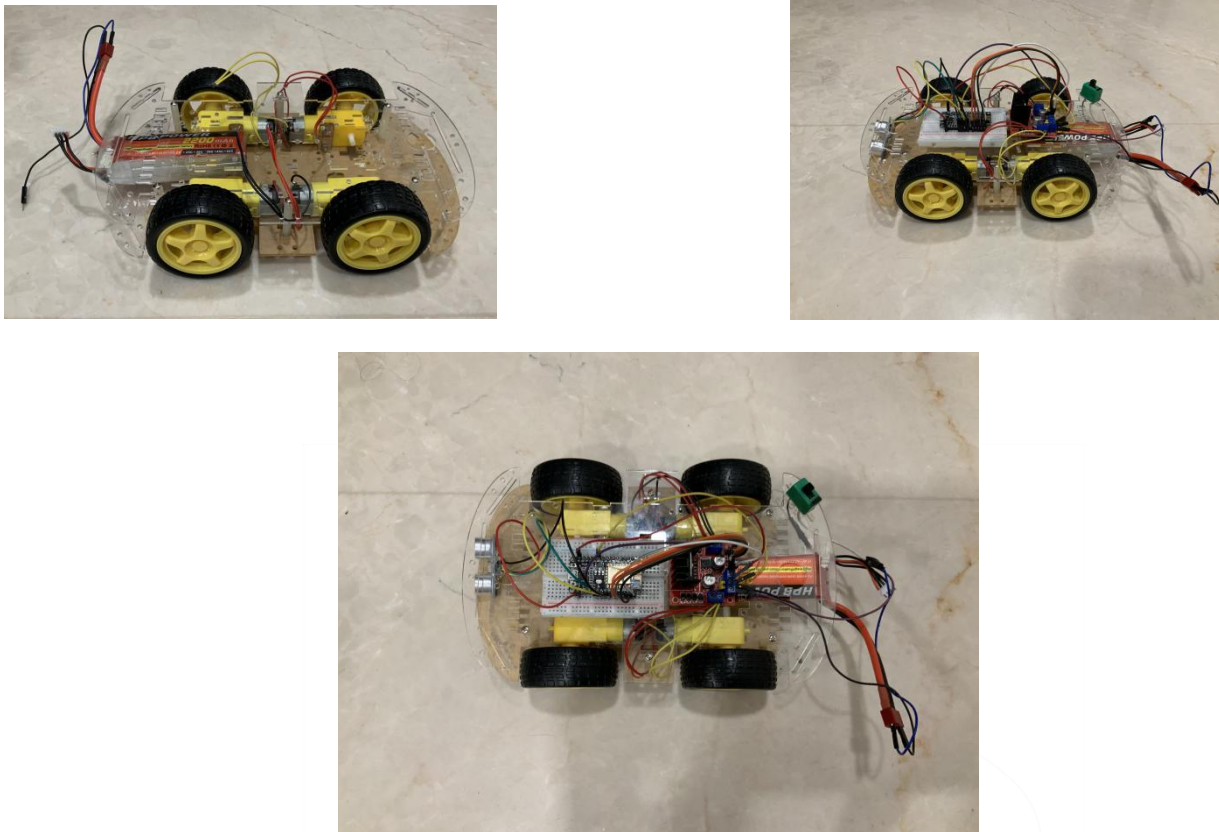


Figure 3.1 Chassis and Assembly of the Robot

Figure 3.1 shows three pictures showing the assembly methods for the mobile robot. In the very first picture, motors are connected in parallel along with LIPO battery. Motors are connected with NODEMCU via L298d motor driver (second picture). Third picture shows the fully assembled robot with battery and switch.

3.2 Circuit Connections and Circuit Diagram

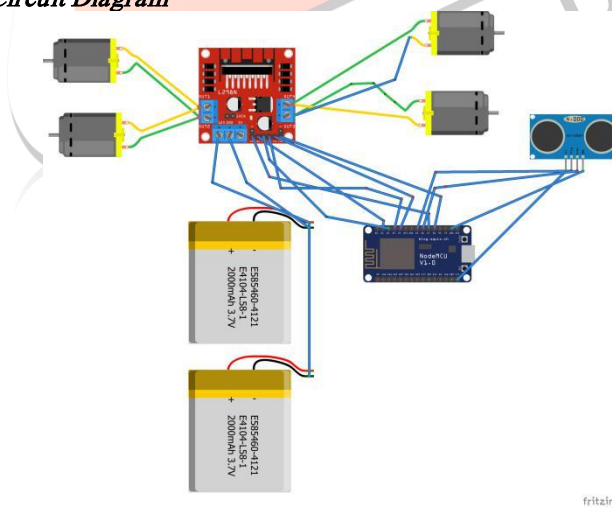


Figure 3.2 Circuit Diagram of the system

In figure 3.2, circuit diagram is shown of the entire robotic system. In the given circuit, NodeMCU is connected with L298d Motor driver along with four DC brushed Motors at pin number D1 to D4. The ultrasonic sensor is also connected with NODEMCU at pin no D5 and D6 (ECHO and TRIGGER PIN). Pin No D1 to D4 are Pulse width Modulation pins. DC Motors work between 6 volts to 11 volts. So in the proposed design, we provided 7.4 volts in simulation (2 LIPO batteries connected in parallel) and provided 11.4 volts in the actual design.

3.3 Full Mock up model/ Working Model

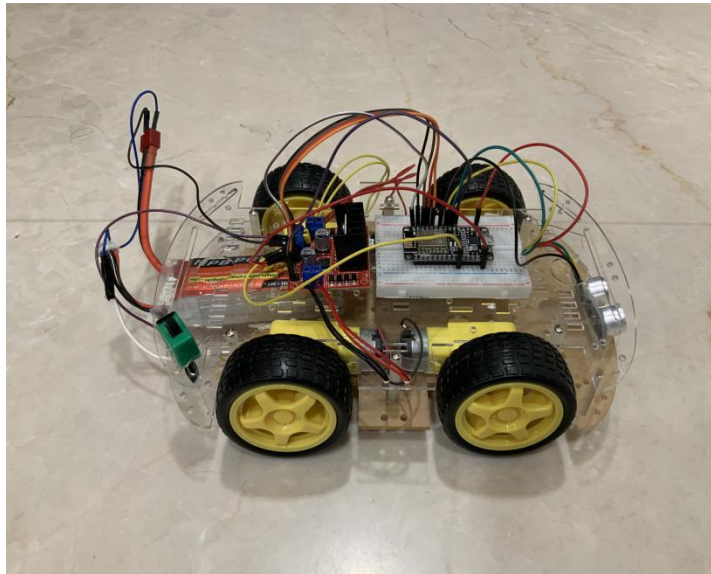


Figure 3.3 Fully functional Robot with NODEMCU and Ultrasonic Sensor

IV. RESULTS AND TESTING THE ROBOT
4.1 BLYNK Application User Interface



Figure 4.1 BLYNK Application User Interface to Control the Robot

As per figure 4.1, we can control the Robot robot using Blynk Application via NODEMCU board. Pin No V6 is used to collect the data from the ultrasonic sensor. A red virtual led is connected at virtual pin V5. Pin No V7, V8, V9 and V10 are used to control the direction of the robot. A slider is connected to virtual pin V4 to control the speed of Motors. Figure 4.2 and 4.3 show real time working of the robot.

4.2 Working robot picture

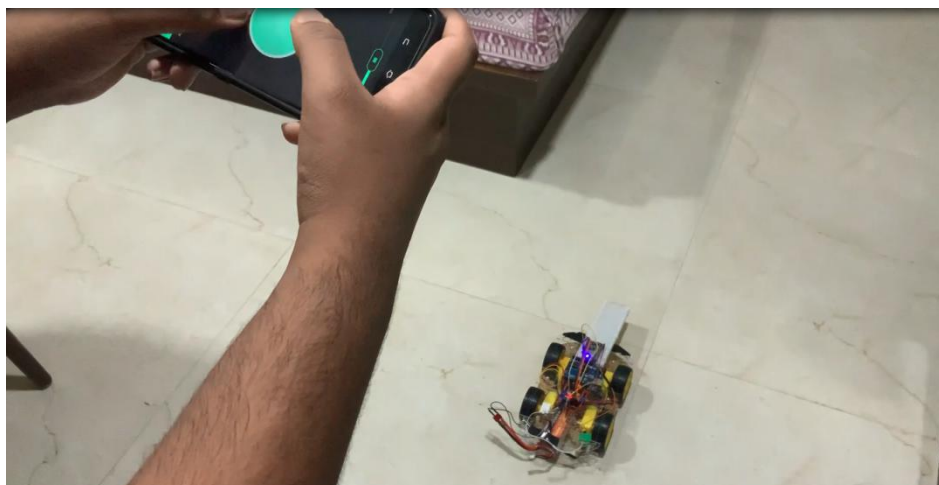


Figure 4.2 Robot Controlling using BLYNK Application



Figure 4.3 Robot Controlling using BLYNK Application

V. Programming

//THIS IS THE RESEARCH PAPER CODE (THE FINAL ONE, WITH THE CODE FOR BUTTONS INSTEAD OF JOYSTICK)

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

```
char auth[] = "bioIbO4VkVhoXJdmg00htUBZabY1ePeH";
char ssid[] = "rydham";
char pass[] = "rydham123";
int echo = D5;
int trig = D6;
unsigned long duration = 0;
long distance = 0;
int a = D1, b = D2, d = D3, c = D4;
int EN_A = D0, EN_B = D7;
int x, y;
int spd = 0;
WidgetLED led1(V5);
int s1 = 0;
int s2 = 0;
boolean st1 = LOW;
boolean st2 = LOW;
boolean st3 = LOW;
boolean st4 = LOW;
int fow = 0;
int lef = 0;
```

```

int rig = 0;
int bac = 0;
int count = 0;
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  pinMode(a,OUTPUT);
  pinMode(b,OUTPUT);
  pinMode(c,OUTPUT);
  pinMode(d,OUTPUT);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
}

void loop()
{
  Blynk.run();
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duration = pulseIn(echo, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  Blynk.virtualWrite(V6, distance);
  if(fow == 1 && distance > 15){
    Blynk.setProperty(V5, "color", "#D3435C");
    led1.on();
    forward();
  }
  else if(lef == 1 && distance > 15){
    Blynk.setProperty(V5, "color", "#23C48E");
    led1.on();
    left();
  }
  else if(bac == 1 && distance > 15){
    Blynk.setProperty(V5, "color", "#23C48E");
    led1.on();
    backward();
  }
  else if(rig == 1 && distance > 15){
    Blynk.setProperty(V5, "color", "#23C48E");
    led1.on();
    right();
  }
  else{
    Blynk.setProperty(V5, "color", "#D3435C");
    Blynk.virtualWrite(V6, 1001);
    led1.on();
    stopp();
  }
  analogWrite(EN_A, s1);
  analogWrite(EN_B, s2);
  digitalWrite(a, st1);
  digitalWrite(b, st2);
  digitalWrite(c, st3);
  digitalWrite(d, st4);
}

BLYNK_CONNECTED(){

```



```

Blynk.syncVirtual(V4);
Blynk.syncVirtual(V5);
Blynk.syncVirtual(V6);
Blynk.syncVirtual(V7);
Blynk.syncVirtual(V8);
Blynk.syncVirtual(V9);
Blynk.syncVirtual(V10);
}

BLYNK_WRITE(V4){
  spd = param.asInt();
}
BLYNK_WRITE(V7){
  fow = param.asInt();
}
BLYNK_WRITE(V8){
  rig = param.asInt();
}
BLYNK_WRITE(V9){
  bac = param.asInt();
}
BLYNK_WRITE(V10){
  lef = param.asInt();
}

void stopp(){
  s1 = 0;
  s2 = 0;
  st4 = LOW;
  st3 = LOW;
  st2 = LOW;
  st1 = LOW;
}

void backward(){
  s1 = spd;
  s2 = s1;
  st1 = HIGH;
  st3 = HIGH;
  st2 = LOW;
  st4 = LOW;
}

void forward(){
  s1 = spd;
  s2 = s1;
  st2 = HIGH;
  st4 = HIGH;
  st1 = LOW;
  st3 = LOW;
}

void right(){
  s1 =spd;
  s2 = s1;
  st1 = HIGH;
  st2 = LOW;
  st3 = LOW;
  st4 = HIGH;
}

void left(){
  s2 = spd;
  s1 = s2;

```



```
st1 = LOW;  
st2 = HIGH;  
st3 = HIGH;  
st4 = LOW;
```

```
}
```

CONCLUSION

At the end of this project it can be concluded that, the obstacle avoider robot can be controlled using Internet of things and 3rd party BLYNK server. In addition with that, data of Ultrasonic sensor can also be monitored via cloud on Blynk Application. Using this methodology and technology, one can use this robot for the house hold applications by adding few more actuators and sensors.

REFERENCES

- [1] Parikh P.A., Sheth S., Patel T. (2016) Positional Analysis of a DC Brushed Encoder Motor Using Ziegler-Nichols Algorithm. In: Mandal D.K., Syan C.S. (eds) CAD/CAM, Robotics and Factories of the Future. Lecture Notes in Mechanical Engineering. Springer, New Delhi. https://doi.org/10.1007/978-81-322-2740-3_62
- [2] Parikh P.A., Joshi K.D., Trivedi R. (2021) Vision-Based Trajectory Planning for a Five Degree of Freedom Assistive Feeding Robotic Arm Using Linear Segments with Parabolic Blend and Cycloid Functions. In: Billingsley J., Brett P. (eds) Mechatronics and Machine Vision in Practice 4. Springer, Cham. https://doi.org/10.1007/978-3-030-43703-9_16
- [3] Priyam A. Parikh, Reena Trivedi, and Jatin Dave, "Trajectory Planning for the Five Degree of Freedom Feeding Robot Using Septic and Nonic Functions," International Journal of Mechanical Engineering and Robotics Research, Vol. 9, No. 7, pp. 1043-1050, July 2020. DOI: 10.18178/ijmerr.9.7.1043-1050
- [4] Parikh, P., Sheth, S., Vasani, R., & Gohil, J. K. (2018). Implementing Fuzzy Logic Controller and PID Controller to a DC Encoder Motor – “A case of an Automated Guided Vehicle.” *Procedia Manufacturing*, 20, 219–226. doi:10.1016/j.promfg.2018.02.032
- [5] Anil, Kanoria Shubham, Pandya Jeet Madhusudan and P. Parikh. “Design and Development of Universal Motor Control Unit using MATLAB and Arduino.” *International Journal for Scientific Research and Development* (2017): 89-93.

