

***IOT Based Fire Extinguishing & Toxic Gas detection Robot using Blynk Cloud***

***A Project report submitted in partial fulfillment of the requirements for  
the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

***Submitted by***

Dharmireddi Sadhika (319126512L11)

Maradana Deepika (318126512073)

Kudari Venu (318126512086)

Shaik Shahid (318126512102)

**Under the guidance of**

**Mr. Nitin Kumar**

**Assistant Professor**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(UGC AUTONOMOUS)**

***(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)  
Sangivalasa, bheemili Mandal, Visakhapatnam dist. (A.P)***

2021-2022

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(UGC AUTONOMOUS)**

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC)*

**Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)**



**CERTIFICATE**

This is to certify that the project report entitled “**IOT Based Fire Extinguishing & Toxic Gas detection Robot using Blynk Cloud**” Submitted **Dharmireddi Sadhika (319126512L11)**, **Maradana Deepika (318126512073)**, **KudariVenu (318126512086)**, **Shaik Shahid (318126512102)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

*Project Guide*  
*Nitin Kumar*  
*31/5/22*  
**Mr. Nitin Kumar**

**Assistant Professor**

Department of E.C.E

ANITS

Assistant Professor  
Department of E.C.E.  
Anil Neerukonda

Institute of Technology & Sciences  
Sangivalasa, Visakhapatnam-531 162

*Dr. V. Rajyalakshmi*  
**Head of the Department**

**Dr. V. Rajyalakshmi**

**Professor&HOD**

Department of E.C.E

ANITS

**Head of the Department**  
**Department of E C E**  
**Anil Neerukonda Institute of Technology & Sciences**  
**Sangivalasa - 531 162**

## **ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our project guide Mr. Nitin Kumar Designation, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to Dr. V. Rajyalakshmi, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the Principal and Management, ANITS, Sangivalasa, for their encouragement and cooperation to carry out this work.

We express our thanks to all teaching faculty of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank all non-teaching staff of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

### **PROJECT STUDENTS:**

**Dharmireddi Sadhika (319126512L11)**

**Maradana Deepika (318126512073)**

**Kudari Venu (318126512086)**

**Shaik Shahid (31816512102)**

## **ABSTRACT**

In today's world sometimes when fire accidents occur it is difficult for the fireman to access the place of fire and extinguish it. In the event of a fire breakout, to rescue people and to put out the fire we are forced to use human resources which are not safe. By designing and implementing an autonomous robot capable of detecting and extinguishing flames, disasters can be avoided with minimal risk to human life. These robots have the capability to detect and extinguish fire automatically. These robots are based on IOT and has flame sensor to detect the fire and a sensor to detect toxic gas.

Robotic technology has the potential to be a great relief for emergency response personnel-in their daily work as well as when facing extreme disasters robots can be employed to extinguish fires.

In this project we are using the micro controller (NODE MCU (ESP 32)) to perform the action that is continuously monitors the specified area whether there is any fire and automatically it goes near to the flame and extinguish the fire. Here we are using a MQ7 gas sensor to detect the toxic gas like methane presence in that specified area.

# CONTENTS

## ACKNOWLEDGEMENT

## ABSTRACT

## LIST OF FIGURES

## CHAPTER 1 INTRODUCTION

### 1.1 Introduction

### 1.2 IOT (Internet of Things)

#### 1.2.1 Uses of IOT

#### 1.2.2 How IOT works

#### 1.2.3 Disadvantages of IOT

### 1.3 Introduction to Blynk

#### 1.3.1 Uses of Blynk

### 1.4 Problem Identification

### 1.5 Literature Survey

## CHAPTER 2 HARDWARE REQUIRED

### 2.1 NODE MCU

#### 2.1.1 About NODE MCU

#### 2.1.2 Specifications and Features

### 2.2 Motor Driver (L298D)

#### 2.2.1 About L298D

#### 2.2.2 Features

### 2.3 Servo Motor

#### 2.3.1 About Servo motor

### 2.4 Flame Sensor

#### 2.4.1 About Flame Sensor

#### 2.4.2 Specifications and Features

## 2.5 Dc motor

### 2.5.1 About Dc Motor

### 2.5.2 Principle of Dc Motor

### 2.5.3 Specifications

## 2.6 Dc Motor Pump

### 2.6.1 About Motor pump

### 2.6.2 Specifications

## 2.7 Robot chassis

### 2.7.1 About Robot chassis

### 2.7.2 Features

## 2.8 MQ7 Gas Sensor

### 2.8.1 About MQ7

### 2.8.2 Features and Specifications

## **CHAPTER 3 SOFTWARE REQUIRED**

### 3.1 Arduino IDE

#### 3.1.1 Introduction to Arduino IDE

#### 3.1.2 Arduino IDE Definition

#### 3.1.3 How to get Arduino IDE

#### 3.1.4 Arduino Libraries

#### 3.1.5 Program Structure of Arduino IDE

### 3.2 Introduction to Blynk

#### 3.2.1 Blynk app

## **CHAPTER 4 METHODOLOGY**

### 4.1 Methodology

### 4.2 Block Diagram

4.3 Flow Chart

4.4 Design and Implementation

4.4.1 Flame Detection

4.4.2 Gas Detection

4.4.3 Node MCU

4.4.4 Android Application

4.4.5 Code

**CHAPTER 5 RESULTS**

**CONCLUSION**

**REFERENCES**

## **LIST OF FIGURES**

- Fig 1.1 Internet of Things (IOT)
- Fig 1.2 Blynk app Logo
- Fig 1.3 Blynk Libraries
- Fig 2.1 ESP8266 Module
- Fig 2.2 ESP8266 Module Pin Diagram
- Fig 2.3 Node MCU Module
- Fig 2.4 Node MCU Pin Diagram
- Fig 2.5 Motor Driver IC
- Fig 2.6 L298D Motor Driver Pin Diagram
- Fig 2.7 Servo Motor
- Fig 2.8 Servo Motor Pin Description
- Fig 2.9 Flame Sensor
- Fig 2.10 Dc Motor
- Fig 2.11 Dc Motor pump
- Fig 2.12 Robot Chassis
- Fig 2.13 MQ7 Gas Sensor
- Fig 2.14 MQ7 Gas Sensor Pinout
- Fig 3.1 Arduino IDE
- Fig 3.2 How to get Arduino IDE
- Fig 3.3 Creating new project in Blynk
- Fig 3.4 Selecting Hardware model
- Fig 3.5 Interfacing with Blynk
- Fig 3.6 Selecting gpio pins
- Fig 3.7 Selecting ESP832 in Arduino IDE



# CHAPTER 1

## INTRODUCTION

### 1.1 INTRODUCTION:

In current world scenario communication between everyday objects is done by the use of microcontrollers. Many smart devices assist human being in their day-to-day activities. These devices perform work cost effective and also efficiently. Now comes the use of robots which are used in dangerous places like bomb detection in aeronautical applications, chemical industries, etc., our aim of this paper to design such a robot which would detect and extinguish fire where human beings are not accessible.

The proposed of this robot can detect and extinguish fire on its own. If fire in a place is not detected in its early stage it would spread rapidly. The fireman who are normally involved in this process may lose their lives, Therefore the purpose of firefighting robot is that it would sense the fire using flame sensor, extinguish it using water by means of the water spraying mechanism.

The Fire Fighting Robots are most popularly searched to prevent the fire injuries and to improve the effectiveness of the robot. In order to determine the lead of the fire to the robot, smoke and fire (thermal) reflections can be clued. The project aims to design a robotic fire extinguishing vehicle that can be operated both wirelessly and manually. The main objective of the project is to automatically or manually design and implement a fire fighter robot to extinguish fire. The robot is equipped with sensors that help us detect fire, smoke or any obstacles in its path. The proposed vehicle has a gas spray which, compared to water, is able to extinguish fire at a faster rate. The robot control system presented here can be used for various robotic applications.

The main and only work is to deploy the robot in a fire prone area and the robot will automatically work once it detects a fire breakout. This prototype helps in Rescue operations during fire accidents where the entry of service man is very difficult in the fire prone area. There are several existing types of vehicles for firefighting at home and extinguish forest fires. Our proposed robot is designed to be able to work on its own or be controlled remotely. By using such robots, fire identification and rescue activities can be done with higher security without placing fire fighters at high risk and dangerous conditions. In other words, robots can reduce the need for fire fighters to get into dangerous situations.

When we the field of firefighting has long been a dangerous one, and there have been numerous and devastating losses because of a lack in technological advancement. Additionally, the current methods applied in firefighting are inadequate and inefficient relying heavily on humans who are prone to error, no matter how extensively they have been trained. A recent trend that has become popular is to use robots instead of humans to handle fire hazards. This is mainly because they can be used in situations that are too dangerous for any individual to involve themselves in. Robot consist of the two sensors that are interfaced

in the control circuitry. Sensors are used to detect fire prone area all directions and moves the robot to fire location. When the robot reaches fire zone then a pump extinguisher is attached on the robot comes into action to extinguish the fire.

## 1.2 IOT (INTERNET OF THINGS):

The term IoT is mainly used for devices that wouldn't usually be generally expected to have an internet connection, and that can communicate with the network independently of human action. For this reason, a PC isn't generally considered an IoT device and neither is a smartphone -- even though the latter is crammed with sensors. smartwatch or a fitness band or other wearable device might be counted as an IoT device.

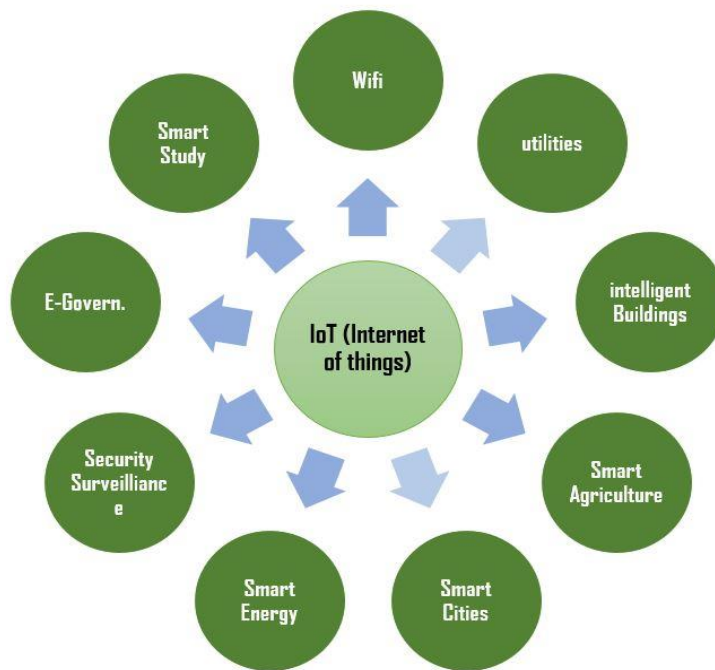


Fig 1.1 Internet of Things

### 1.2.1 Uses of IOT:

#### 1. Efficient resource utilization:

If we know the functionality and the way that how each device works, we definitely increase the efficient resource utilization as well as monitor natural resources.

#### 2. Minimize human effort:

As the devices of IoT interact and communicate with each other and do lot of tasks for us, then they minimize the human effort.

#### 3. Save time:

As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.

4.Enhance Data Collection.

5.Improve security:

Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

### 1.2.2 HOW IOT WORKS?

The entire IoT process starts with the devices themselves like smartphones, smartwatches, electronic appliances like TV, Washing Machine which helps you to communicate with the IoT platform.

**1) Sensors/Devices:** Sensors or devices are a key component that helps you to collect live data from the surrounding environment. All this data may have various levels of complexities. It could be a simple temperature monitoring sensor, or it may be in the form of the video feed.

A device may have various types of sensors which performs multiple tasks **apart** from sensing. Example, A mobile phone is a device which has multiple sensors like GPS, camera but your smartphone is not able to sense these things.

**2) Connectivity:** All the collected data is sent to a cloud infrastructure. The sensors should be connected to the cloud using various mediums of communications. These communication mediums include mobile or satellite networks, Bluetooth, WI-FI, WAN, etc.

**3) Data Processing:** Once that data is collected, and it gets to the cloud, the software performs processing on the gathered data. This process can be just checking the temperature, reading on devices like AC or heaters. However, it can sometimes also be very complex like identifying objects, using computer vision on video.

**4)User Interface:** The information needs to be available to the end-user in some way which can be achieved by triggering alarms on their phones or sending them notification through email or text message. The user sometimes might need an interface which actively checks their IoT system. For example, the user has a camera installed in his home. He wants to access video recording and all the feeds with the help of a web server.

However, it's not always one-way communication. Depending on the IoT application and complexity of the system, the user may also be able to perform an action which may create cascading effects.

For example, if a user detects any changes in the temperature of the refrigerator, with the help of IoT technology the user should be able to adjust the temperature with the help of their mobile phone.

- **Technical Optimization:** IoT technology helps a lot in improving technologies and making them better. Example, with IoT, a manufacturer is able to collect data from

various car sensors. The manufacturer analyses them to improve its design and make them more efficient.

- **Improved Data Collection:** Traditional data collection has its limitations and its design for passive use. IoT facilitates immediate action on data.
- **Reduced Waste:** IoT offers real-time information leading to effective decision making & management of resources. For example, if a manufacturer finds an issue in multiple car engines, he can track the manufacturing plan of those engines and solves this issue with the manufacturing belt.
- **Improved Customer Engagement:** IoT allows you to improve customer experience by detecting problems and improving the process.

### 1.2.3 DISADVANTAGES OF IOT:

- **Security:** IoT technology creates an ecosystem of connected devices. However, during this process, the system may offer little authentication control despite sufficient security measures.
- **Privacy:** The use of IoT, exposes a substantial amount of personal data, in extreme detail, without the user's active participation. This creates lots of privacy issues.
- **Flexibility:** There is a huge concern regarding the flexibility of an IoT system. It is mainly regarding integrating with another system as there are many diverse systems involved in the process.
- **Complexity:** The design of the IoT system is also quite complicated. Moreover, its deployment and maintenance also not very easy.
- **Compliance:** IoT has its own set of rules and regulations. However, because of its complexity, the task of compliance is quite challenging.

## 1.3 Introduction to Blynk:

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other various operations. Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where we can build a graphic interface for any project by simply dragging and dropping widgets. This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address on the available widgets.

There are three major components in the platform:

- **Blynk App**

It allows us to create amazing interfaces for our projects using various widgets we provide. The Blynk innovation is the smartphone app. It allows us to create one or more projects. Each project can contain graphical widgets, like virtual LEDs, buttons, value displays and even a text terminal, and can interact with one or more devices. With the help of the Blynk library, it is possible to control Arduino or ESP32 pins directly from our phone, without

any requirement of writing the code. It is also possible to share a project with friends and even customers so that they can access the connected devices but are unable to modify the project. For example, we can imagine a scenario where we build a smartphone application where we can control lights, window blinds and room temperature from our phone.



Fig:1.2 Blynk App Logo

- **Blynk Server**

It is responsible for all the communications between the smartphone and hardware. We can use the Blynk Cloud or run our private Blynk server locally. It's open-source which can easily handle thousands of devices and can even be launched not only on Node MCU but also on a Raspberry Pi. Unlike IoT platform such as IFTTT, Twilio, and even Adafruit IO, we can host a private instance of the full Blynk server and connect our smartphone Blynk app to it. The Blynk Cloud server is an excellent choice for most projects, as it is always there, ready to use. However, the Cloud Blynk server has imposed limitations. Some limitations are due to the topology of the server depending on our geographical location, the server may be in a different continent, which makes communications between the app, the devices and the server slow due to the amount of time it takes packets to travel across the Internet.

However, a private Blynk server gives you additional benefits:

- Essentially unlimited energy units, so that you can build any Blynk application you can think of.
- Minimal latency, which is useful when your application is used in a limited geographic area and responsiveness is important.
- Total control of your data. You can keep your own backups of your private server, migrate your server to a new host, implement whichever security mechanisms you wish, and finely control your users.

### 1.3.1 Uses of Blynk:

- With Blynk, we can create smartphone applications that allows us to easily interact with microcontrollers or even full computers such as the Raspberry Pi.
- The main focus of the Blynk platform is to make it super-easy to develop the mobile phone application. With Blynk, you can control an LED or a motor from your mobile phone with literally zero programming.
- Blynk is a robust and scalable tool that is used by hobbyists and the industry alike.
- We can use it to monitor the soil humidity of your vegetable garden and turn on the water, or open up our garage door, with our phone.
- We can also use it to control smart furniture that can learn from our routines, or embed IoT and AI to traditional industrial products such as a boiler, or for improving the integrity and safety of oilfields.
- Blynk is free to use for personal use and prototyping. Their business model generates profits by selling subscriptions to businesses that want to publish Blynk-powered apps for their hardware products or services.
- **Blynk Libraries:**

It is used for all the popular hardware platforms; it enables communication with the server and process all the incoming and outgoing commands. The Blynk platform supports range of devices and connectivity types. The support is implemented by means of a Blynk library that targets a device and connectivity type combination. For example, if we want to use our Arduino Uno with an Ethernet shield, we would use the library “BlynkSimpleEthernet” which contains the Blynk firmware plus the required connectivity support. Blynk also supports clients that are not a microcontroller, we can write client code in JavaScript, Python or Lua. This means that we can have a Blynk project that interacts with an Arduino and with Python code running on a Linux virtual machine somewhere on the Cloud.

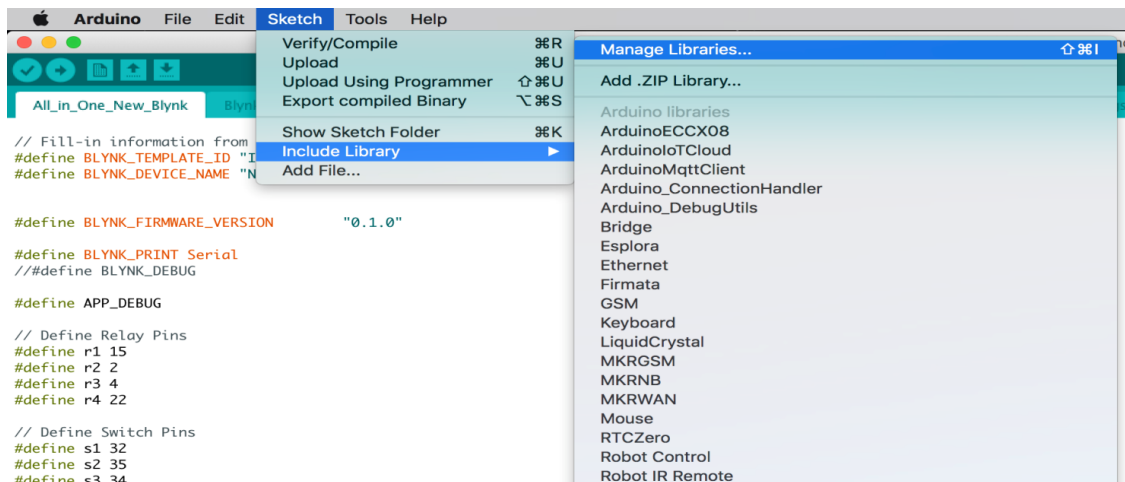


Fig 1.3 Blynk Libraries

#### 1.4 PROBLEM IDENTIFICATION:

Now a days, fire accidents are very common and sometimes it becomes very difficult for a fireman to save someone's life. It is not possible to appoint a person to continuously observe for accidental fire where robot can do that. Therefore, in such cases firefighting robot comes in picture. Robot will detect fire remotely. These robots are mostly useful in industries where probability of accidental fire is more.

Fire incident is a disaster that can potentially cause the loss of life, property damage and permanent disability to the affected victim. Fire fighters are primarily tasked to handle fire incidents, but they are often exposed to higher risks when extinguishing fire, especially in hazardous environments such as in nuclear power plant, petroleum refineries and gas tanks. They are also faced with other difficulties, particularly if fire occurs in narrow and restricted places, as it is necessary to explore the ruins of buildings and obstacles to extinguish the fire and save the victim. A firefighting robot which can be employed to extinguish fires, search for hazardous materials, provide injured with first aid, or get them out of dangerous areas. Firefighters have to cope with a wide range of emergency response scenarios, starting from smaller fire extinction tasks or accidents with very few victims, but leading up to large catastrophes, like terroristic attacks or natural disasters, with possibly hundreds or thousands of casualties.

All these scenarios have in common that it is inherently dangerous for human personnel to work in such situations

## **1.5 LITERATURE SURVEY:**

In olden days, there is no automatic extinguishing of fire. The fire brigade should be called to the fire prone area which is operated by human whose lives are at greater risk. It takes time to arrive at the fire place and mostly water is used to put off fire. Later with the evolution of technology, fire extinguishers came into existence which is filled with CO<sub>2</sub> limitedly. In recent years around 2016, manually controlled fire extinguishing robot came into existence which is followed by pre trained robot for particular environment like smart homes fire detection and avoidance. It also sends alert to the user through Bluetooth which is capable for communication over some meters. The proposed robot uses iot for streaming live pictures of the fire prone area and puts off all types of fire by fitting it with the corresponding fire extinguishing substances depending on the fire prone area.

A fire extinguisher robot designed with the look and function of a robot assembled with the intention to extinguish the fire by using a water pump as actuators. The movement of the fire extinguisher robot is controlled using Android smart phones via Wi-Fi networks utilizing Wi-Fi module contained in the robot. User commands are sent to the microcontroller on the robot and then translated into robotic movement. The robot was equipped with cameras and ultrasonic sensors. The camera played role in giving feedback to user and in finding the source of fire. Feedback provided by camera on the robot is displayed on a screen of the smart phone.

The requirement of this proposed work is to create a robot that could fully governing itself. When the robot is kicked off by the user, the robot could maneuver itself, poke into, and quench the fire on its own without any backing. The designed robot indicates the fire through alarm. It is preprogrammed with the environment. The robot can only be used in a pre-trained area like home, and not for general purpose.

The robot has multiple modes of operation; the first mode of operation is to regulate the movement wirelessly using a Node MCU module, paired with a smart phone application. The second mode of operation is consummated by linking the interfacing IR sensors with Node MCU for obstacle detection and temperature sensing using temperature sensing LM35 IC. The quenching mechanism used in the different modes of operation is a water pump. The robot proposed is trained to do the functionality of the fire brigade.

The goal of the proposed work is achieved by training the robot as a fire fighter which is controlled automatically. The robot has a water tanker unit and a pump unit. Both the units are self – restrained by wireless communication along with a microcontroller, the flames of the fire are quenched by spraying the water. The transmitter end, send commands to the receiver which helps in the robotic movement in all the directions. The decoder decodes in prior feeding it to another microcontroller to drive DC motors via motor driver IC to make the robot in firefighting.



## CHAPTER 2

### HARDWARE REQUIRED

#### 2.1 NODE\_MCU:

##### 2.1.1 About Node\_MCU:

Node MCU is an open-source Lua based firmware and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.



Fig 2.3: NODE MCU module

The Node MCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Ten silica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency.

NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. Node MCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

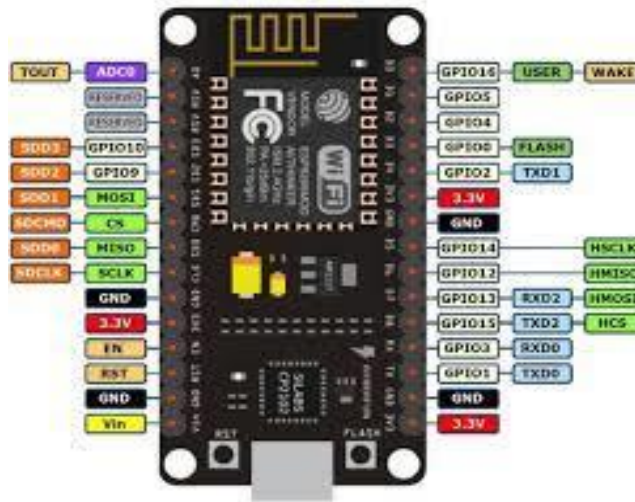


Fig 2.4: NODE MCU Pin Diagram

The ESP8266 is the low-cost Wi-Fi microchip this small module allows a microcontroller to connect to the Wi-Fi network. Node\_MCU is an open-source firmware for which we can program it using Arduino IDE so we have uploaded a program for motion control of our robot. Node\_MCU can be powered using Micro USB jack and VIN pin External Supply Pin. It supports UART, SPI, and I2C interface.

Node\_MCU Development Board can programmed with an Arduino IDE since it is easy to use. Programming Node MCU with the Arduino IDE is much easier. All you need is the Arduino IDE, a USB cable and the Node MCU board.

### 2.1.2 Node MCU ESP8266 Specifications & Features:

1. Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
2. Operating Voltage: 3.3V
3. Input Voltage: 7-12V
4. Digital I/O Pins (DIO): 16
5. Analog Input Pins (ADC): 1
6. UARTs: 1
7. SPIs: 1
8. I2Cs: 1
9. Flash Memory: 4 MB
10. SRAM: 64 KB
11. Clock Speed: 80 MHz
12. USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
13. PCB Antenna

14. Small Sized module to fit smartly inside your IoT projects.

## 2.2 Motor Driver:

### 2.2.1 About L298D:

The L298D is a popular 15-Pin Motor Driver IC. As the name suggests it is mainly used to drive motors. A single L298D IC is capable of running four DC Motors at the same time; also has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 Timers, digital gates or even Micron rollers like Arduino, PIC, ARM etc..

L298D is designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current and high-voltage loads. All inputs are TTL compatible and tolerant up to 7 V. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full H (or bridge) reversible drive suitable for solenoid or motor applications.

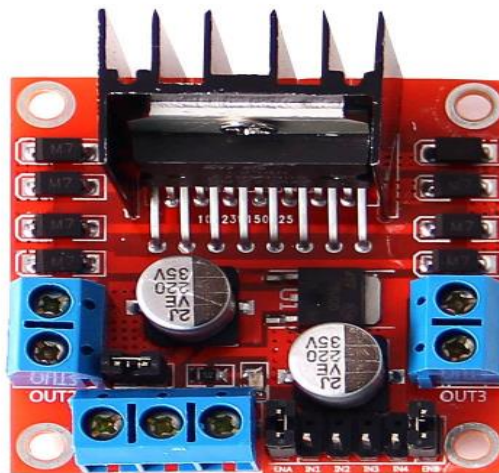


Fig 2.5: Motor Driver IC

The external high-speed output clamp diodes of the L298D are integrated to reduce system complexity and overall system size. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L298D is characterized for operation from 0°C to 70°C.

There are 4 input pins for direction control in L298d. Pin 5,7 (1A and 2A) on the left side and pin 13,10 (3A and 4A) on the right of the IC. The left side input pins regulate the rotation of the motor connected across the left end and the right-side input pins regulate the motor on the rightside. The motors are rotated based on the inputs provided across the input pins as HIGH or LOW signals.

L298D motor driver L298D is a typical motor driver IC that allows DC motor to drive in either direction. L298D is a 15-pin IC that can control a set of two DC motors simultaneously in any direction. It means you can control four DC motors with a single L298D IC

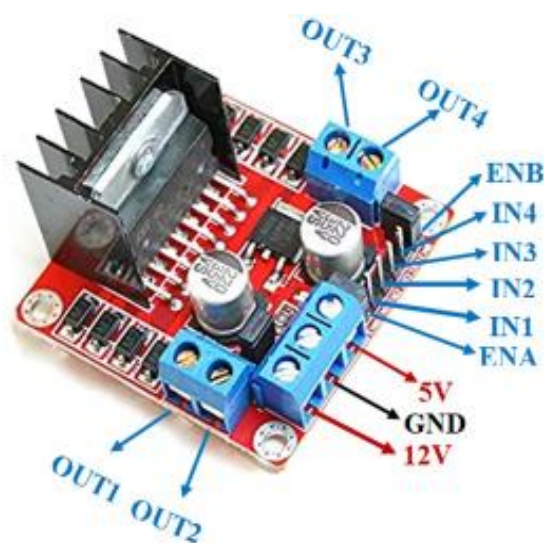


Fig 2.6: L298D motor driver Pin Diagram

### 2.2.2 L298D Features:

1. Wide Supply-Voltage Range: 4.5 V to 36 V
2. Separate Input-Logic Supply
3. Internal ESD Protection
4. High-Noise-Immunity Inputs
5. Output Current 1 A Per Channel (600 mA for L298D)
6. Peak Output Current 2 A Per Channel (1.2 A for L298D)
7. Output Clamp Diodes for Inductive Transient Suppression (L298D)

## 2.3 Servo motor:

### 2.3.1 About Servo motor:

Servo motor is a rotatory actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with the servo motors. It is not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed loop control system.

Servo motor is a rotatory actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with the servo motors. It is not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed loop control system.



Fig 2.7: Servo Motor

SG90 Servo Motor Pinouts	
Terminal Color	Brief Description
Red	VCC Connected to 3.5V to 5V
Brown	Ground
Orange/Yellow	Control Signal or PWM Signal to be applied on this terminal

Fig 2.8: Servo Motor Pin Description

A servo motor is a motor with a built-in “servomechanism”. The servomechanism uses a sensor to monitor the motor shaft position and a controller to control the motor. It is fed a signal that indicates the position that the shaft should be set to. It then moves the motor into the required position. In the analog servo motors we will be working with the control signal which is a PWM signal whose pulse width determines the angle the motor shaft is to be positioned at. The motor itself is a simple DC motor with a lot of gearing to slow down its speed and to increase its torque. In order to function properly the servo motor needs a sensor that can accurately measure its shaft position. On some industrial and high-end hobby servos this is done using an optical interrupter disc, but in most standard hobby servo motors the sensor is a potentiometer. This works well as these servos typically travel 180 to 270 degrees, well within the range of a potentiometer.

## 2.4 Flame sensor:

### 2.4.1 About flame sensor:

A flame-sensor is one kind of detector which is mainly designed for detecting as well as responding to the occurrence of a fire or flame. The flame detection response can depend on its fitting. It includes an alarm system, a natural gas line, propane & a fire suppression system. This sensor is used in industrial boilers. The main function of this is to give authentication whether the boiler is properly working or not. The response of these sensors is faster as well as more accurate compare with a heat/smoke detector because of its mechanism while detecting the flame.

This sensor/detector can be built with an electronic circuit using a receiver like electromagnetic radiation. This sensor uses the infrared flame flash method, which allows the sensor to work through a coating of oil, dust, water vapor, otherwise ice.

The pin configuration of this sensor is shown below. It includes four pins which include the following. When this module works with a microcontroller unit then the pins are



Fig 2.9: Flame sensor

Pin1 (VCC pin): Voltage supply ranges from 3.3V to 5.3V

Pin2 (GND): This is a ground pin

Pin3 (AOUT): This is an analog output pin (MCU.IO)

Pin4 (DOUT): This is a digital output pin (MCU.IO)

## **2.4.2: Features & Specifications**

The features of this sensor include the following.

1. Photosensitivity is high
2. Response time is fast
3. Simple to use
4. Sensitivity is adjustable
5. Detection angle is 600,
6. It is responsive to the flame range.
7. Accuracy can be adjustable
8. Operating voltage of this sensor is 3.3V to 5V
9. Analog voltage o/ps and digital switch o/ps
10. The PCB size is 3cm X 1.6cm
11. Power indicator & digital switch o/p indicator
12. If the flame intensity is lighter within 0.8m then the flame test can be activated, if the flame intensity is high, then the detection of distance will be improved.

## **2.5 Dc motor**

### **2.5.1 About Dc motor:**

A DC motor or direct current motor is an electrical machine that transforms electrical energy into mechanical energy by creating a magnetic field that is powered by direct current. When a DC motor is powered, a magnetic field is created in its stator. The field attracts and repels magnets on the rotor; this causes the rotor to rotate. To keep the rotor continually rotating, the commutator that is attached to brushes connected to the power source supply current to the motors wire windings.



Fig 2.10: Dc Motor

### **2.5.2: Principle of DC Motor:**

When a current-carrying conductor is placed in a magnetic field, it experiences a torque and has a tendency to move.

In other words, when a magnetic field and an electric field interact, a mechanical force is produced. The DC motor or direct current motor works on that principle. This is known as motoring action.

The direction of rotation of this motor is given by Fleming's left hand rule, which states that if the index finger, middle finger, and thumb of your left hand are extended mutually perpendicular to each other and if the index finger represents the direction of the magnetic field, middle finger indicates the direction of the current, then the thumb represents the direction in which force is experienced by the shaft of the DC motor.

### **2.5.3 specifications & Features:**

1. Standard 130 Type DC motor
2. Operating Voltage: 4.5V to 9V
3. Recommended/Rated Voltage: 6V
4. Current at No load: 70mA (max)
5. No-load Speed: 9000 rpm
6. Loaded current: 250mA (approx)
7. Rated Load: 10g\*cm
8. Motor Size: 27.5mm x 20mm x 15mm
9. Weight: 17 grams



## 2.6: Dc motor pump:

### 2.6.1 About Dc motor:

The pump moves water in a cylinder through a piston or plunger. It is the pump that produces a force so that the water creates a flow. Unlike a motor engine which uses the movement of the piston to turn the crankshaft, the pump relies on the crankshaft movement to drive the pistons. This movement allows the water to flow in and out of the pump.

There's a common misconception that a pump produces the pressure. However, what the pump does is actually move the water. Instead, it is the force exerted from the water as it flows through the pump that creates the pressure. So, when talking about the pressure from a high-pressure pump, you're actually looking at is the speed at which the pump moves the water throughout the pump at a particular gallon per minutes measurement.



Fig 2.11: DC Motor Pump

There are two check valves in the cylinder of the pump that controls which direction the water flows through the pump. You could call them the entrance and exit to the pump that you can only go through one way. The valve that opens inward is the inlet check valve, which allows the water to flow into the pump. The outlet check valve does the opposite. It lets water escape from the pump.

### 2.6.2: Specifications:

1. Input Voltage: DC 3V-5V.
2. Flow Rate: 1.2-1.6 L/min.
3. Operation Temperature: 80 Deg. C.
4. Operating Current: 0.1-0.2A.
5. Suction Distance: 0.8 meter (Max)
6. Outside diameter of water outlet: 7.5mm.
7. Inside diameter of water outlet: 5.0 mm.
8. Diameter of water Inlet : 5.0 mm.

## 2.7 Robot chassis:

### 2.7.1 About robot chassis:

This Transparent Robot Smart Car Chassis comes with a quality product. The kit is easy to assemble/disassemble. Now it has become very easy to design robotic smart cars.

The motor power supply is 3V ~ 6V. Rotating speed of the motor is 125RPM. All the parameter above is tested without load. The car Chassis is transparent, you need to tear out a protective paper. The colour of a protective paper may be different.

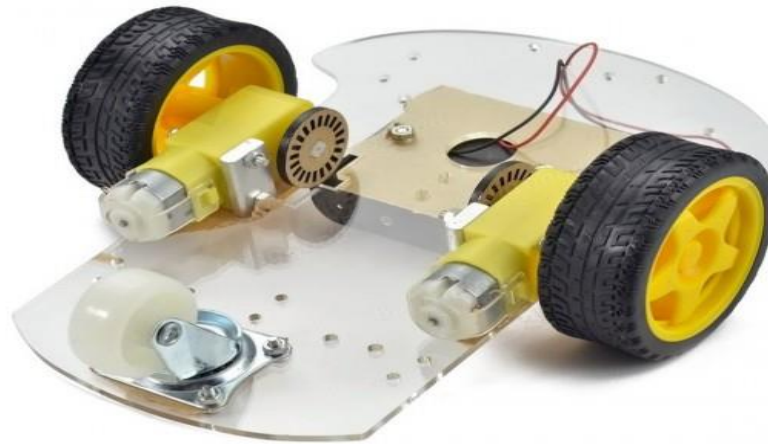


Fig 2.12:Robot chassis

### 2.7.2:Features:

1. Smart car chassis tracing car Robot car chassis with code disk, Ideal for DIY.
2. This car is the tachometer encoder.
3. With a 4 AA battery box (batteries not included).
4. It can be used for distance measurement, velocity.
5. Can use with other devices to realize the function of tracing, obstacle avoidance, distance testing, speed testing, Wireless remote control.

## 2.8 MQ7 Gas Sensor:

### 2.8.1 About MQ7 gas sensor:

With the increasing use of gases, gas leakage problems are also arising. We have seen so many accidents because of the gas leakage issue and therefore we cannot neglect this issue,

either in industries or at homes. So, there are sensors like MQ7 that detect gas leakage and warn the people around. So, if you are searching for any gas sensor, this sensor may help you.

MQ series sensor uses a small heater inside with an electrochemical sensor in order to measure a different kinds of gas combinations. They can be calibrated, but, in order to do that, a known concentration of measured gas or gases is needed. Gas sensor is an indoor module and is used at room temperature. The MQ-7 sensor has high sensitivity to Carbon Monoxide and is able to detect the concentration of carbon monoxide in the air. The measurable concentration range of this sensor is between **20ppm to 2000ppm**.

MQ7 is a sensitive gas sensor that can detect or sense liquefied gas, propane, butane, natural, and other combustible gases in the environment. And, hence used in gas leakage detectors, and other industrial gas detectors, or monitoring devices. The output voltage of this MQ7 Gas sensor increases when the concentration of gases or a gas increase. We can adjust the sensitivity by adjusting the potentiometer.



Fig 2.13: MQ7 gas sensor

The sensor contains a micro  $AL_2O_3$  ceramic tube, a sensitive layer of tin dioxide ( $SnO_2$ ), measuring electrode, and heater, fixed into a crust made by plastic and stainless-steel net.

The sensor includes a sensitive filament formed of tin dioxide. In clean air, this filament has lower electrical conductivity. When there is a combustible gas like LPG in the surrounding, the conductivity of filament rises, and the amount of change in its resistance can show the equivalent gas concentration.

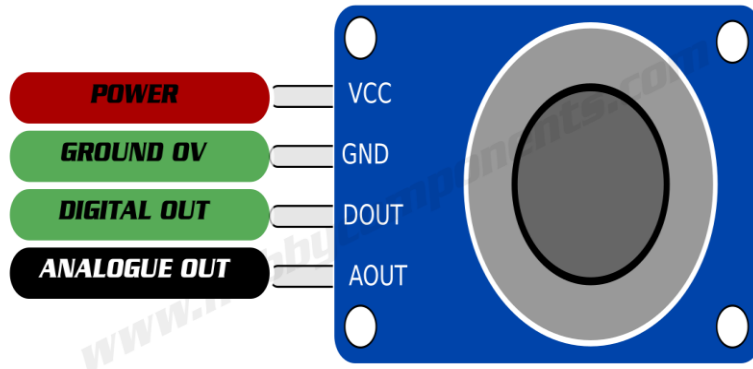


Fig 2.14:MQ7 Gas Sensor Pin Out

### 2.8.2 Features:

- Small sensitivity to alcohol, smoke.
- Fast response
- Stable and long life
- Simple drive circuit
- Indication of the signal output.
- Dual signal output (analog output, and TTL level output)
- TTL output signal is low. (When low output signal light, and can be connected directly to the microcontroller)
- Liquefied petroleum gas, natural gas, city gas, better sensitivity.
- Has a long life and reliable stability

### 2.8.3 Specifications:

- Working Voltage: 4.9V-5.1V
- Heater Resistance:  $31 \pm 10\%$
- Adjustable Resistance
- Sensing Resistance Range: 10 – 60K
- Sensor Heat Consumption: 0.5-800mW
- Detecting Concentration: 200-10000ppm

## **CHAPTER 3**

### **SOFTWARE REQUIRED**

#### **3.1 Arduino IDE**

##### **3.1.1 Introduction to Arduino IDE:**

IDE stands for “Integrated Development Environment”: it is an official software introduced by Arduino.cc, that is mainly used for editing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go. In this article, we will introduce the Software, how we can install it, and make it ready for developing applications using Arduino modules.



Fig:3.1 Arduino IDE

##### **3.1.2 Arduino IDE Definition:**

1. Arduino IDE is an open-source software that is mainly used for writing and compiling the code into the Arduino Module.
2. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
3. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
4. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
5. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
6. The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.

7. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

8. This environment supports both C and C++ languages.

### 3.1.3 How to get Arduino IDE:

we can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MacOS, we select to download the correct software version that is easily compatible with our operating system.

Details on IDE:

The IDE environment is mainly distributed into three sections

- 1. Menu Bar
- 2. Text Editor
- 3. Output Pane

As we download and open the IDE software, it will appear like an image below

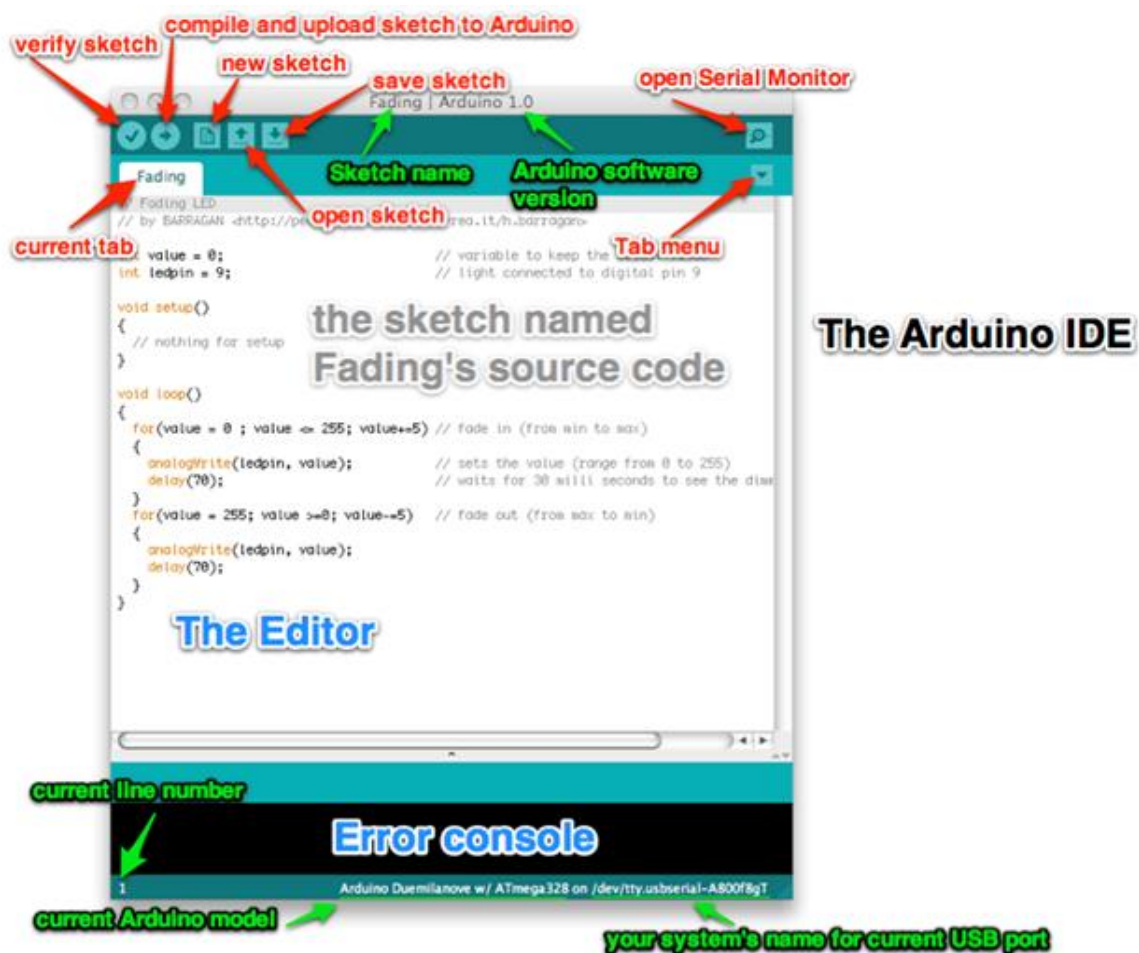


Fig 3.2 How to Get Arduino IDE

The bar appearing on the top is called Menu Bar that comes with five different options as follow

- File – You can open a new window for writing the code or open an existing one.
- Edit – Used for copying and pasting the code with further modification for font
- Sketch – For compiling and programming
- Tools – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.
- Help – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

### 3.1.4 Libraries:

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.

As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, Temperature sensors DHT11/22, LCD or I2C library it will appear on the text editor as

```
#include<EEPROM>.
```

```
#include<dht.h>
```

```
#include<I2Cdev.h>
```

```
#include<SoftwareSerial.h>
```

Most of the libraries are preinstalled and come with the Arduino software. However, we can also download them from the external sources.

### Making Pins as Input or Output

The digital Read and digital Write commands are used for addressing and making the Arduino pins as an input and output respectively. These commands are text sensitive i.e., you need to write them down the exact way they are given like digital Write starting with small “d” and write with capital “W”. Writing it down with Digital write or digital write won’t be calling or addressing any function.

**Example:** if we want to use Pin D13 as output, the code will be; pin Mode (13, OUTPUT); followed by digital Write (13, HIGH)

If we want to use Pin D13 as input, the code will be: `pin Mode (13, INPUT);`  
followed by `x=digital Read (13);`

### **3.1.5 Program Structure of Arduino IDE:**

Declarations

#### **Variables:**

Whenever you're using Arduino, you need to declare global variables and instances to be used later on. In a nutshell, a variable allows you to name and store a value to be used in the future. For example, you would store data acquired from a sensor in order to use it later. To declare a variable, you simply define its type, name and initial value.

It's worth mentioning that declaring global variables isn't an absolute necessity. However, it's advisable that you declare your variables to make it easy to utilize your values further down the line.

Instances

In software programming, a class is a collection of functions and variables that are kept together in one place. Each class has a special function known as a constructor, which is used to create an instance of the class. In order to use the functions of the class, we need to declare an instance for it.

#### **Setup ():**

Every Arduino sketch must have a setup function. This function defines the initial state of the Arduino upon boot and runs only once.

Here we'll define the following:

1. Pin functionality using the pin Mode function
2. Initial state of pins
3. Initialize classes
4. Initialize variables
5. Code logic

#### **Loop ():**

The loop function is also a must for every Arduino sketch and executes once setup () is complete. It is the main function and as its name hints, it runs in a loop over and over again. The loop describes the main logic of your circuit.



### 3.2 Introduction about blynk:

Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. It's really simple to set everything up and you'll start tinkering in less than 5 mins. Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP32 chip, Blynk will get you online and ready for the Internet Of Your Things.

#### 3.2.1 Blynk App :

Blynk application can be found from the following links 1. Android Blynk App 2. IOS Blynk App After downloading the app, create an account and log in. (If possible than log in with your real mail id for better connectivity later.)

Create a Blynk Project

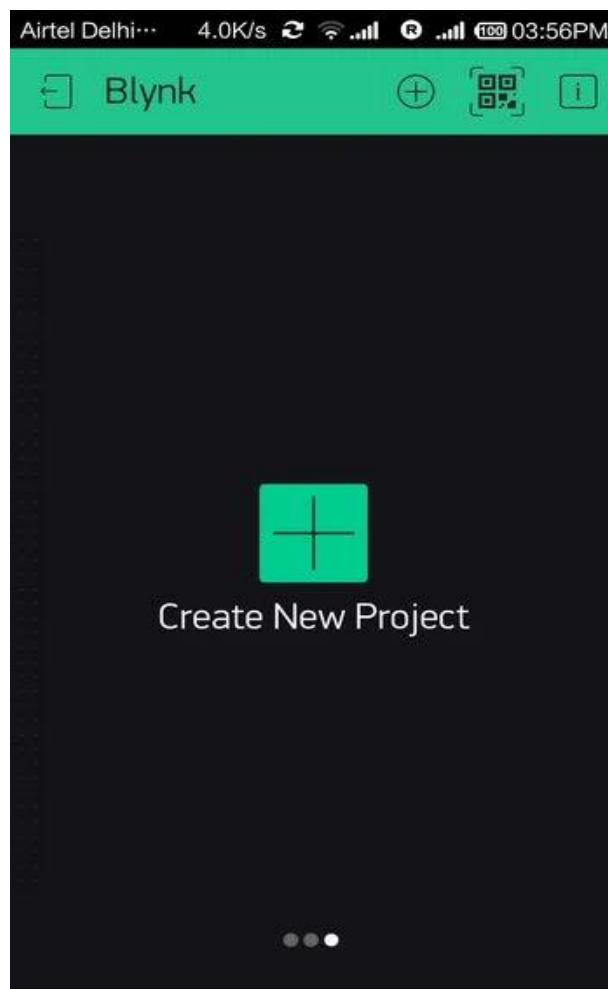


Fig3.3:Creating new project in Blynk

- Click the “Create New Project” in the app to create a new Blynk app. Give it any name. Blynk works with hundreds of hardware models and connection types. Select

the Hardware type. After this select the connection type. In this project we used Wi-Fi connectivity.

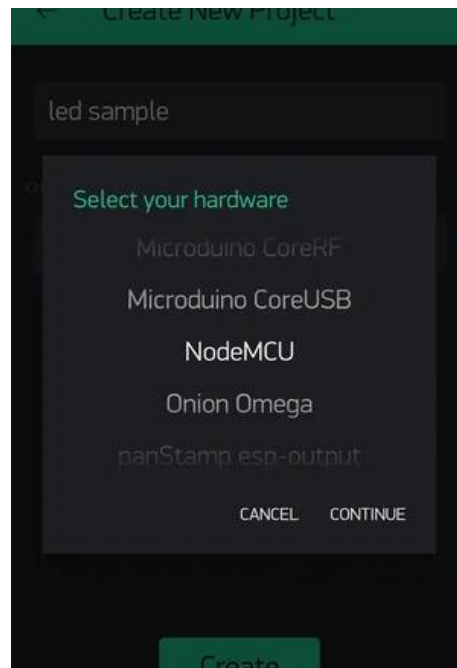


Fig 3.4:Selecting hardware model

- Name your project and select NodeMcu from hardware and connection type Wi-Fi
- Then click on + icon on your upper right side of your main screen to add button to control led.

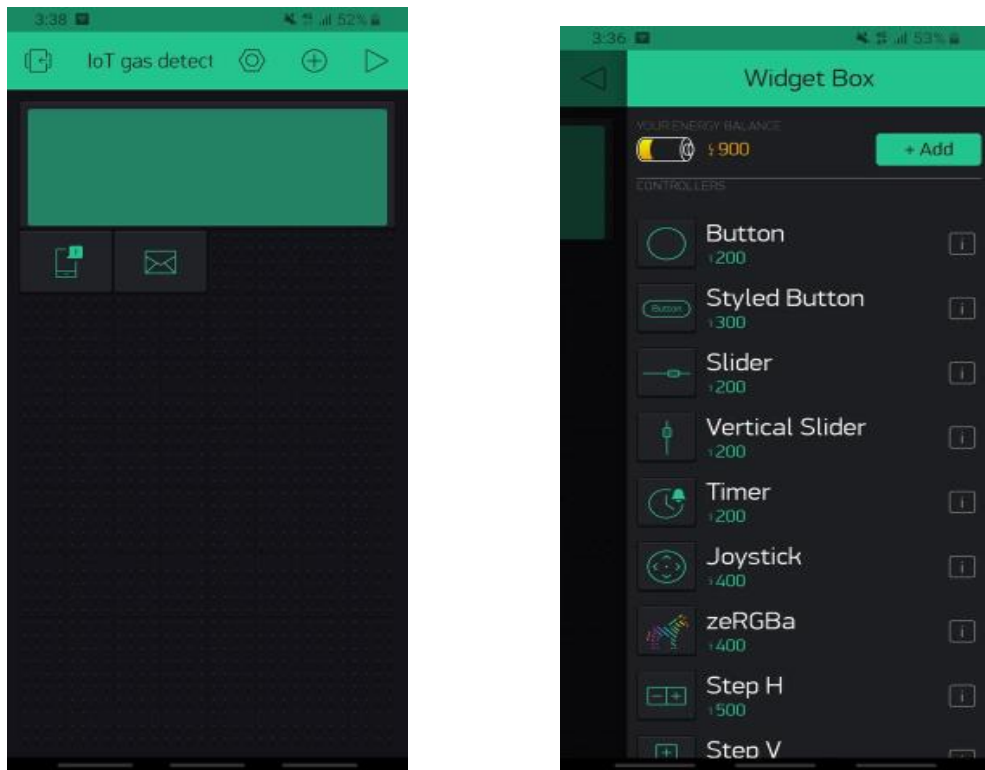


Fig3.5:Interfacing with Blynk

- Drag button icon on the main screen and then click on that button icon

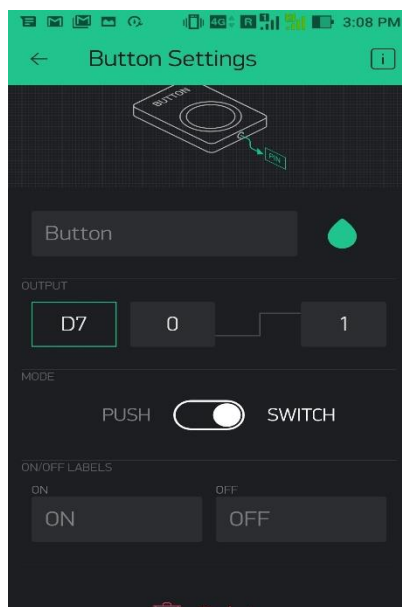


Fig 3.6:Selecting the desired gpio pins

- Change Pin to whatever pin you want to connect your led in output section.
- Then go back to main screen
- Check your mail box. Blynk send you Authorization token. Now we need to install Blynk libraries for Arduino.

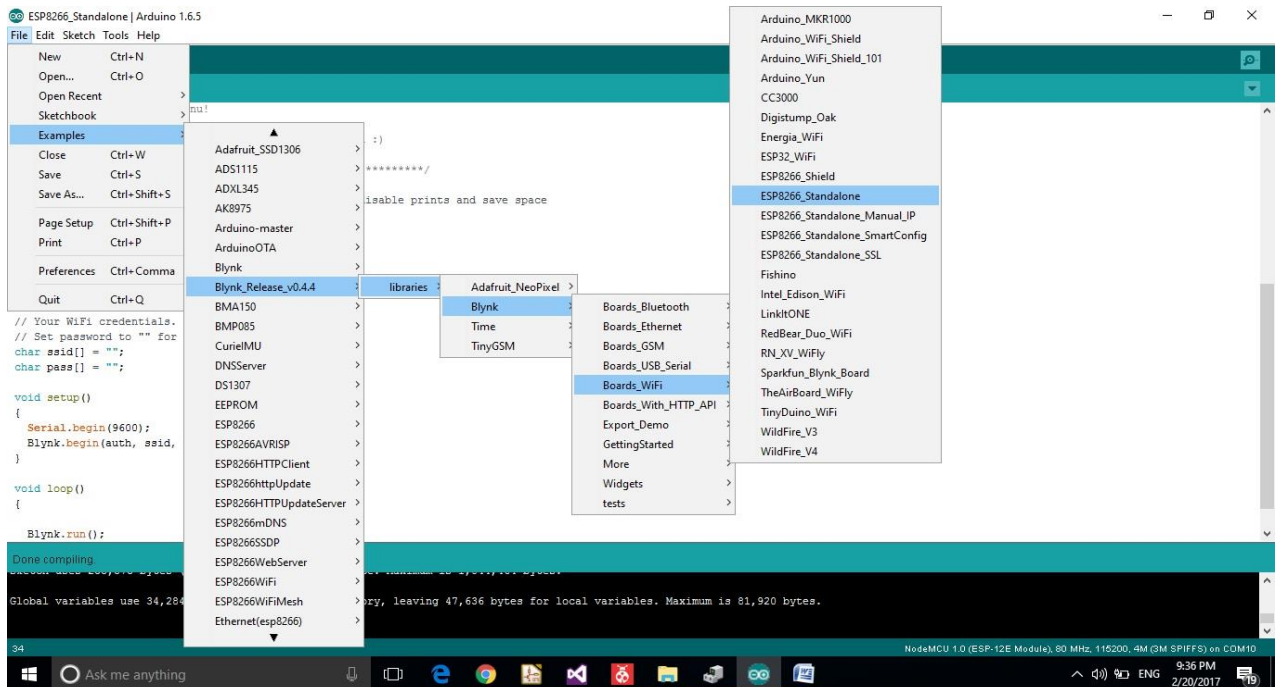


Fig 3.7 Selecting ESP8266 in Arduino IDE

### Blynk library should be installed manually

1. Download **Blynk\_Release\_v0.4.4.zip** (scroll down to the Downloads section)
2. Unzip it. This archive contains several libraries.
3. Copy them to **sketchbook folder** of Arduino IDE.  
To find exact sketchbook folder location, go to **File -> Preferences** top menu in Arduino IDE.

The structure of your sketchbook folder should look like this, along with your saved sketches:

sketchbook/libraries/Blynk

sketchbook/libraries/BlynkESP8266\_Lib

...

sketchbook/tools/BlynkUpdater

sketchbook/tools/BlynkUsbScript

...

In this release

- **New hardware:**
  - Energia BLE
  - TI Launchpad MSP432
- **New features:**
  - Sketch Builder
  - Support overriding BLYNK\_RUN\_YIELD
  - New RTC mechanism  
(**things changed, please update your existing code accordingly to new examples**)
  - Preparation for OTA support
- **Bugs fixed:**
  - Fix Photon hanging, improve Particle cloud responsiveness
  - Less workarounds all over the code
  - Better usb-ser script
  - LED color example fixed
  - Updated IDE tools
- Download all three files
- Then install libraries. In case you don't know how to install Arduino libraries – check here. Follow "Importing a .zip Library" guide.
- Open Blynk ESP32 standalone example from examples. Add auth token which you get on your mail in that

```
char auth [] = "";
```

- Also don't forget to change your Wi-Fi ssid and password
- Compile and upload the code

## CHAPTER 4

### METHODOLOGY

#### 4.1 METHODOLOGY

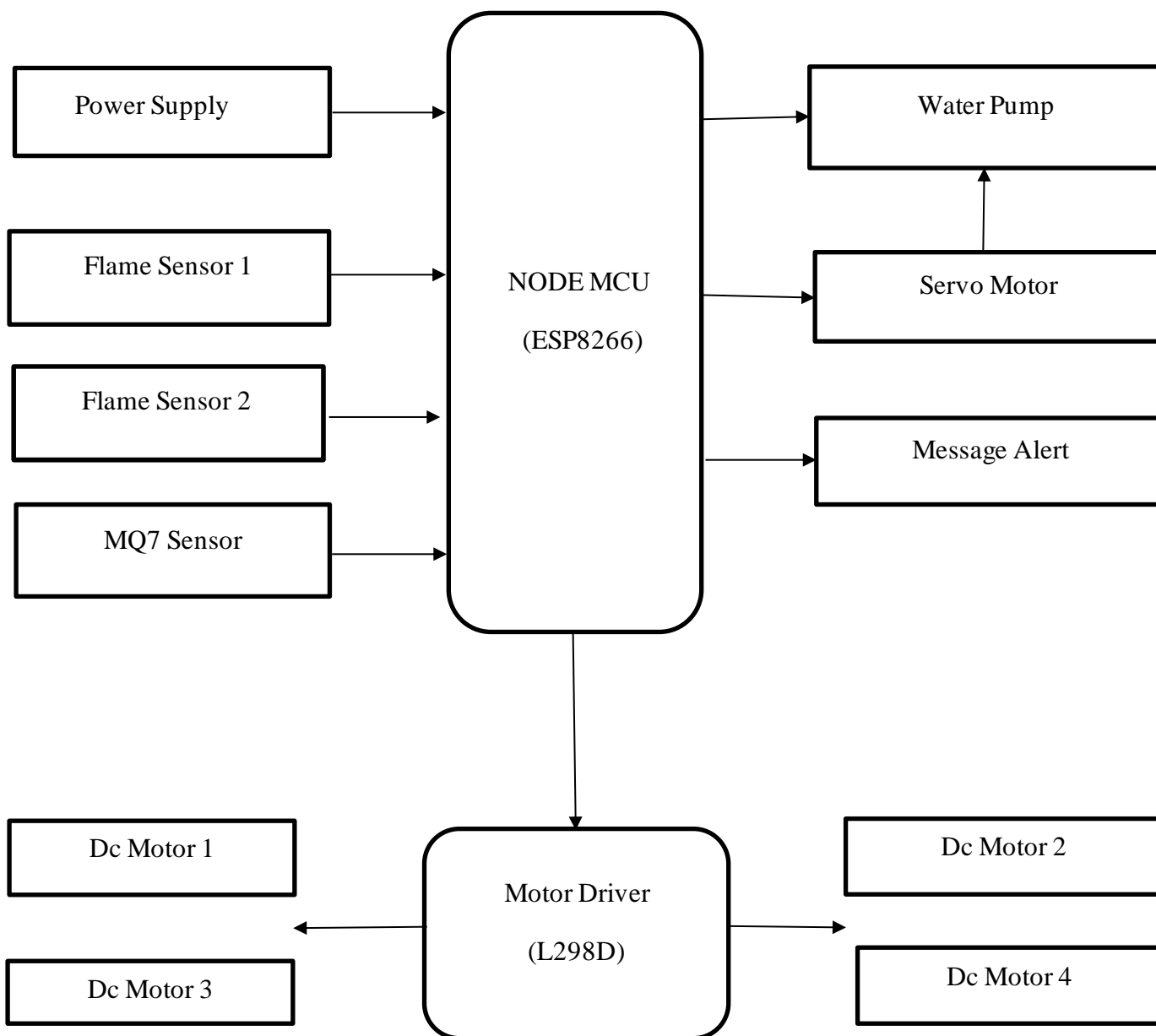
In the indoor of an industrial organization, the robot is placed at the place where fire accidents can happen. A micro-controller and sensors are used to operate this robot. If there exist any fire signal, the sensors will send the information to the central co-ordinator unit. The co-ordinator unit is the brain of the robot. It makes decisions based upon the sensor signals.

The whole system includes a sensors, a Microcon-troller unit, and motors. The sensors' data is sent continuously to the NodeMCU. NodeMCU Collects the data from sensors and then process the data depending upon the program. If the data are analog value, then the data are converted to a digital value using ADC conversion and sent to the server and the android app. The flame sensor value is used to decide to switch on the water pump automatically. By using flame sensors findthe exact position of the fire. The module must be connected with a Wi-Fi router or mobile hotspot to transfer the feedback data from the robot to the server and Android App.

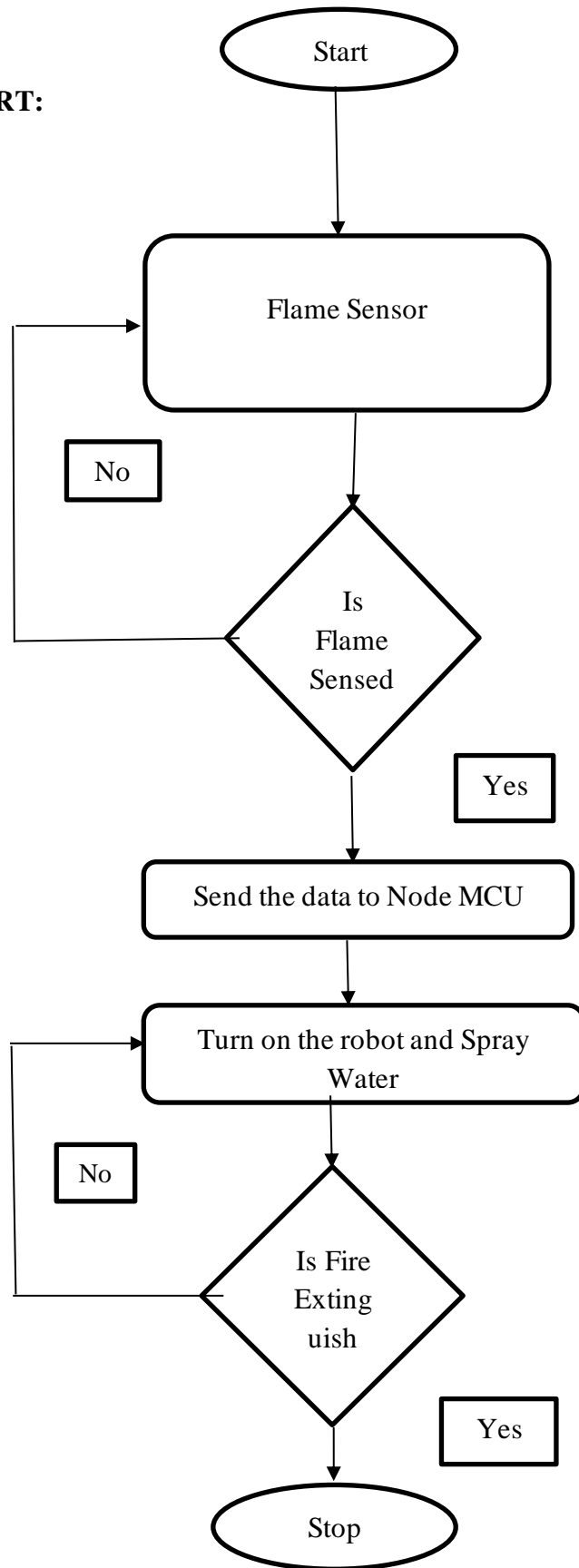
When the robot goes into autonomous mode it started searching for fire. The robot moves round in a circular way from right to left. This movement is the default movement for the autonomous part. There are 3 flame sensors. Flame sensors cover up to 0 to 60 degrees searching for the higher wavelength flame. We put 2 flame sensors in the front of our robot to cover up a significant amount of range to search fire. The 2 flame sensors on the lower part search for fire, if the left flame sensor sense fire, then the robot moves left, if the right flame sensor sense fire, then the robot moves right, and if the middle flame sensor sense fire, then the robot goes forward. The robot moves forward in that direction only when middle sensor senses fire. When it moves forward it follows a straight line towards the fire so it takes minimal time to reach for the fire. Then the pump starts and the fire extinguisher is ready to spray.

A servo motor was attached to the fire extinguisher spray. The servo that is attached to the fire extinguisher spray creates a TENSION on the cap of the spray to release water from the water pump. Another Sensor called MCQ 7 is placed in this project which is used to detect the methane and some harmful gases. When the MCQ 7 sensor senses any harmful gas, it provides information to the NODEMCU which is connected to the Blynk app. Then the NODEMCU sends an alert message to our android regarding presence of harmful gas.

**4.2 BLOCK DIAGRAM:**

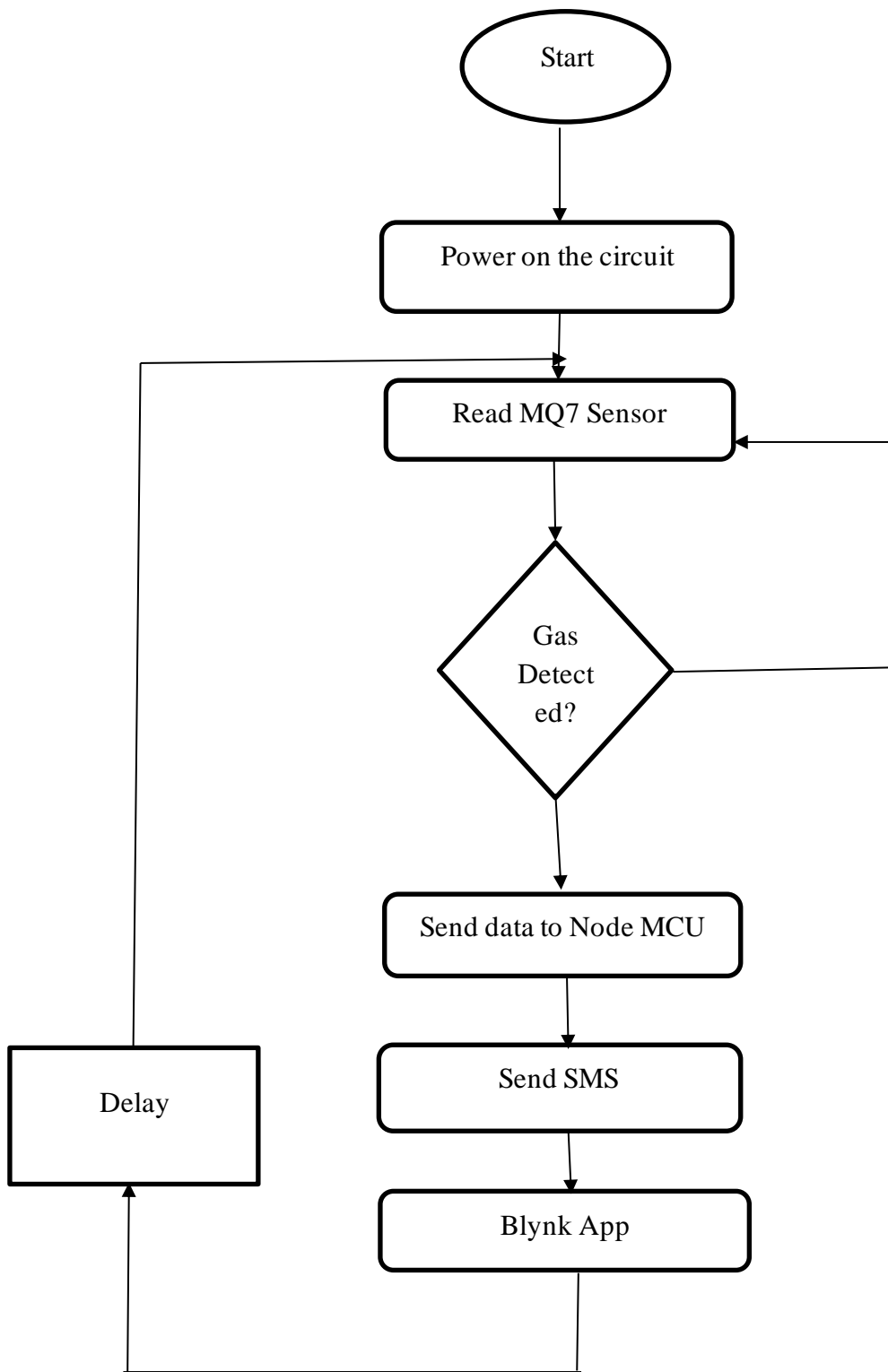


**4.3 FLOW CHART:**



Flow Chart for Fire Extinguishing





Flow chart for Detecting Methane Gas

#### **4.4 DESIGN AND IMPLEMENTATION:**

The control circuit is installed within the walled compartment of the structure. A total of 6-Volt and 10 AH battery has been used for powering the control circuit. The rechargeable Li-ion batteries are connected in series to drive two motors. Each motor drives only one wheel so that sufficient torque is available for the robot to climb steep paths. The IR- flame sensor is located above the structure of the robot to detect the flame . A pipe is put in front of the robot to extinguish the fire.

##### **4.4.1 Flame detection:**

A large amount of infrared radiation is emitted along with a sparse amount of UV rays and visible rays during a fire. The human eye captures the radiation as visible yellow-red flames. However, some fire occurrences, such as hot charcoal, may not emit visible light, but it does emit infrared. Thus, a flame sensor is used in this robot that detects infrared rays and sends a signal upon the fire detection to assure the occurrence of fire.

##### **4.4.2 Gas Detection:**

Gas sensor module MQ7 is used in this robot which detects the presence of combustible gases detecting LPG, natural gas, town gas, avoid the noise of alcohol and cooking fumes and cigarette smoke. MQ-7 is used in gas leakage detecting equipment in consumer and industry applications, this sensor is suitable for detecting LPG, natural gas, coal gas. Avoid the noise of alcohol, cooking fumes, and cigarette smoke. The sensitivity can be adjusted by the potentiometer. The sensor is highly sensitive and provides a speedy response. It provides additional safety on the occurrence of fire to detect the places where combustible gases may congregate due to leakage from a gas pipe or any other sources. The gas sensor sends a signal to the developed app based on its detection of gases. If combustible gases are found, the app displays “GAS: HIGH” notation and “GAS: LOW” notation when no combustible gases are detected.

##### **4.4.3 NodeMCU:**

Module Data between the robot and the app is communicated through the NodeMCU-ESP8266(Node Micro-controller Unit) module. Node MCU is an open-source Lua based firmware. Adoptable prototyping is possible using the NodeMCU-ESP8266 module, which is implemented in C programming. This module has both the Dual-Mode Wi-Fi and Bluetooth. The system communicates via Wi-Fi to remotely monitoring data.

#### 4.4.4 Android Application:

The robot transfer all of the information to the android app named Blynk App. Blynk app is the easiest way to build a mobile app that supports hardware platforms such as Arduino, Raspberry Pi, and a similar micro-controller, which can be used to create required project. This app can control the robot via Wi-Fi and it indicates the presence of gas leakage and sends an alert message to the respective mobile phone.

#### 4.5 Code:

```
#define BLYNK_TEMPLATE_ID "TMPLeVFKD9-Q"
#define BLYNK_DEVICE_NAME "fire extinguisher robot"
#define BLYNK_AUTH_TOKEN "CmvTwwEsrNDWsRHWtg4NXVtc0RjKIg-b"

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

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "moto g5";
char pass[] = "motog5_1234";

Servo servo;
BlynkTimer timer;
//v1--forward,v2--backward,v3--left,v4-right,v5--gas sensing,v6-flame,v7-flame
#define in1 D1
#define in2 D2
#define in3 D3
#define in4 D4
#define flamePin1 D5
#define flamePin2 D6
#define servoPin D7
#define pumpPin D0
```

```

int x, y, z, w;
int flame1Value;
int flame2Value;
void sensorData () {
  int gasValue=analogRead(A0);
  flame1Value=digitalRead(flamePin1);
  Serial.println("flamesensor:"+String(flame1Value));
  flame2Value=digitalRead(flamePin2);
  Blynk.virtualWrite(V5, gasValue);
  Blynk.virtualWrite(V6, flame1Value);
  Blynk.virtualWrite(V7, flame2Value);
  }
BLYNK_WRITE(V4)
{
  x=param.asInt();

  Serial.println(x);
  if(x==1)
  {
    Serial.println("Right Direction");
    digitalWrite(in1, HIGH);
    digitalWrite(in3, HIGH);
  }
  Else
  {
    Serial.println("No Right Direction");
    digitalWrite(in1, LOW);
    digitalWrite(in3, LOW);
  }
}
BLYNK_WRITE(V3)
{
  y=param.asInt();

```

```

Serial.println(y);
if(y==1)
{
  Serial.println("Left Direction");
  digitalWrite(in2,HIGH);
  digitalWrite(in4,HIGH);
  }
Else
{
  Serial.println("No Left Direction");
  digitalWrite(in2,LOW);
  digitalWrite(in4,LOW);
  }
}
BLYNK_WRITE(V1)
{
  z=param.asInt();

  Serial.println(z);
  if(z==1)
  {
    Serial.println("forward direction");
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);
    }
  Else
  {
    Serial.println("No forward direction");
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
  }
}

```

```

BLYNK_WRITE(V2)
{
  w=param.asInt();
  Serial.println(w);
  if(w==1)
  {
    Serial.println("backward direction");
    digitalWrite(in1,HIGH);
    digitalWrite(in4,HIGH);
  }
  Else
  {
    Serial.println("No backward direction");
    digitalWrite(in1,LOW);
    digitalWrite(in4,LOW);
  }
}
void setup()
{
  Serial.begin(9600);
  servo.attach(servoPin);
  Blynk.begin(auth, ssid, pass);
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  pinMode(flamePin1,INPUT);
  pinMode(flamePin2,INPUT);
  pinMode(pumpPin,OUTPUT);
  timer.setInterval(1000,sensorData);
}
void loop()
{

```

```

if(digitalRead(flamePin2) ==1 &&digitalRead(flamePin1) ==0)
{
  digitalWrite(pumpPin,HIGH);
  servo.write(0);
}

else if(digitalRead(flamePin2) ==0&&digitalRead(flamePin1) ==1)
{
  digitalWrite(pumpPin,HIGH);
  servo.write(160);
}
else if(digitalRead(flamePin2) ==1 &&digitalRead(flamePin1) ==1)
{
  digitalWrite(pumpPin,HIGH);
  servo.write(90);
}
Else
{
  digitalWrite(pumpPin,LOW);
  servo.write(90);
}
Blynk.run();
timer.run();
}

```

## CHAPTER 5

### RESULTS

Final prototype of the robot:



Fig 5.1 Robot when Detects fire



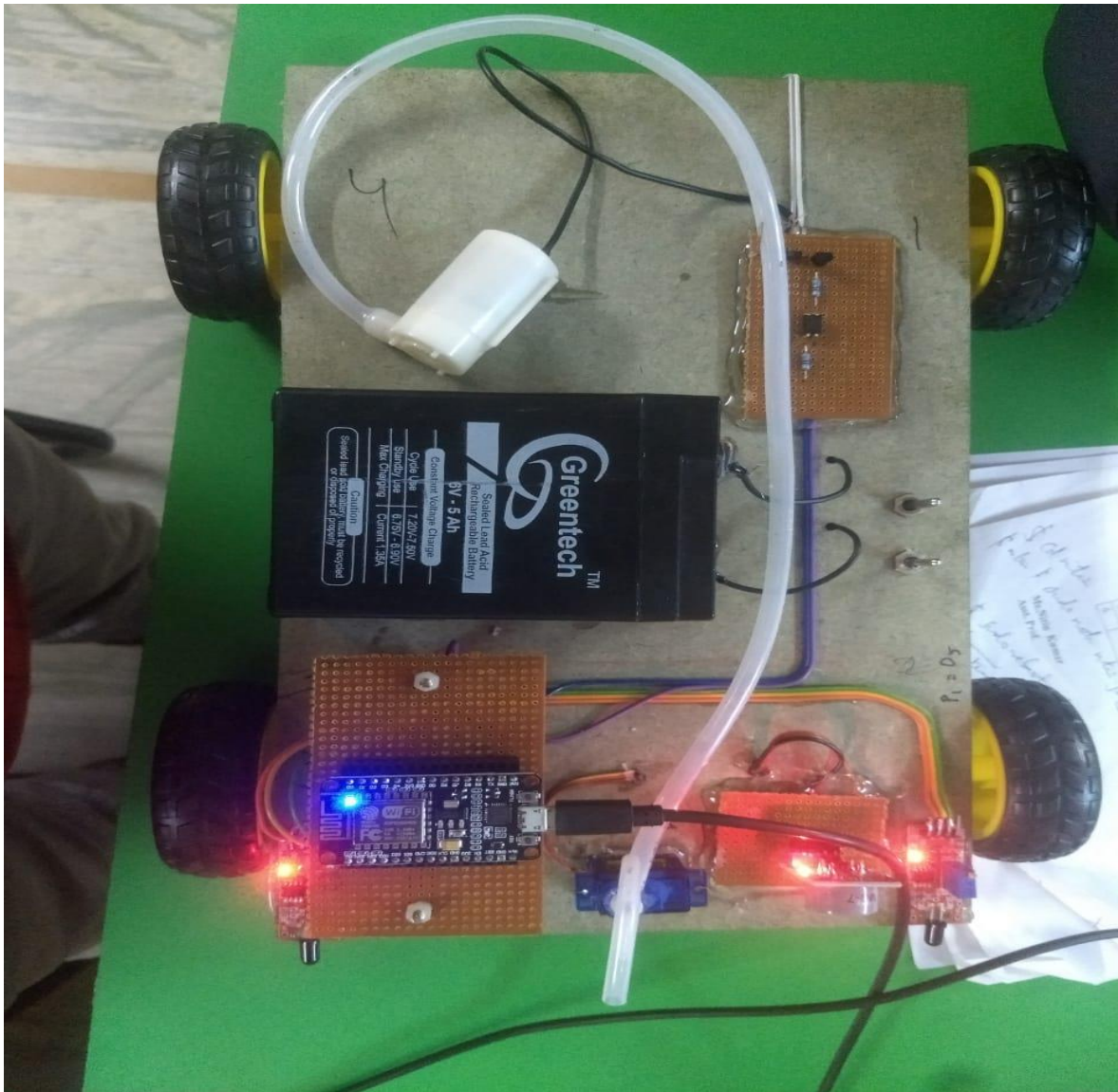


Fig 5.2 Robot under Normal condition

## CONCLUSION

This fire extinguishing robot using IoT is introduced to help fire men to avoid the risk situations. This proposed work presents an outline about the fire extinguishing robot using IoT and also gives objectives and advantages of the proposed system. It provides better features that make it a more practically usable robot in real time. It has advantageous features such as ability to detect location of fire automatically and poisonous gases like methane, carbon monoxide besides having a compact body and lightweight structure.

From the beginning till the end, from the scratch of building this robot to the final stage of this robot, we keep in mind that we want to make our robot real-life implementable. We make every step perfectly before moving to other steps we test every step before implementing it on the larger chassis. Firstly, we made a demo version of our robot to check the Arduino codes. After the completion and test of every component, we make the bigger chassis-based robot. The robot can be used at a place that has a small entrance or in small spaces because it has a compact structure. The system can potentially be useful to accompany fire fighters and prevent an outbreak. By introducing this system in industries and other factories, fire accident rates can be decreased.

## REFERENCES

- Diwanji, M., Hisvankar, S., & Khandelwal, C. (2019). Autonomous Fire Detecting and Extinguishing Robot. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT).
- Kirti Kadam, Ayush Bidkar, Vaishnavi Pimpale, Dhanashri Doke, Rachana Patil, “Fire Fighting Robot”, International Journal of Engineering and Computer Science , 6 January 2018
- D. A. Klein, B. Illing, B. Gaspers, D. Schulz and A. B. Cremers, “Hierarchical Salient Object Detection for Assisted Grasping,” to appear in the Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017.
- P. BhosaleKiran Uttam, Galande Abhijeet Baspusaheb, Jadhav Pappu Shivaji, “Industrial Automation using IoT,” International Journal of Computer Applications, 2017.
- <https://nevonprojects.com/arduino-based-autonomous-fire-fighting-robot/>
- [https://en.wikipedia.org/wiki/Internet\\_of\\_things](https://en.wikipedia.org/wiki/Internet_of_things)