ALL INDIA COUNCIL ROBOTICS & AUTOMATION

PROJECT REPORT ON:

"GPS Based Home Appliance"

Submitted by

ROHIT JAMADHAR (3GN18EC036)

RAHUL SHAMBHU (3GN18EC033)

UNDERTAKING

I declare that the work presented in this project titled "**GPS Based Home Appliance**", submitted to the All India council of robotics and Automation, for the award of the Internship in **INTERNET OF THINGS**, is my original work. I have not plagiarized or submitted the same work for the award of any other Internship. In case this undertaking is found incorrect, I accept that my Project may be unconditionally withdrawn.

OCT 2021

Rohit
Rahul

Certificate

Certified that the work contained in the project titled "**GPS Based Home Appliance**", by **Rahul&Rohit**, has been carried out under my supervision and that this work has not been submitted elsewhere for a Internship..

All India Council of

Robotics and Automation

Name of the Internship

Delhi-110020

Acknowledgement

I am deeply indebted to my mentor **SUMIT CHATTERJEE RESEARCH ENGINEER**, **ALAAUDEEN K M** who motivated me along the way
I would like to thank all my teachers who help me in this project I further thank to my friends
I owe my sincere gratitude towards the God

My heartfelt thanks to parents who supported me a lot

Finally, I would like to wind up by paying my heartfelt thanks to AICRA institute who provided me with this great opportunity

Rohit
Rahul

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

COMPANY PROFILE

Name: TIMTS

Headquarter: DELHI, INDIA

Industry: Professional Training and Coaching

Company Type: Sole Proprietorship

Founded: 2012

E-mail: info@timtsindia.com

Phone no: 91 7834999915

CHAPTER 2

TASK PERFORMED

3.1 INTRODUCTION

The Internet of Things (IoT) describes the network of physical objects— "things"— that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. With more than 7 billion connected IoT devices today, experts are expecting this number to grow to 10 billion by 2020 and 22 billion by 2025.

Over the past few years, IoT has become one of the most important technologies of the 21st century. Now that we can connect everyday objects—kitchen appliances, cars, thermostats, baby monitors—to the internet via embedded devices, seamless communication is possible between people, processes, and things.

By means of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyperconnected world, digital systems can record, monitor, and

adjust each interaction between connected things. The physical world meets the digital world—and they cooperate.

3.1.1 INTERNET OF THINGS (IoT): -

Every "thing" in this world can transmit data to the web and communicate with every other "thing" in order to perform a variety of tasks for its owner. The internet of things (IoT), the physical and virtual world are fused.

10 Real World Applications of IoT

1. Smart Home
2. Wearables
3. Connected Cars
4. Industrial Internet

5. Smart Cities
6. IoT in agriculture
7. Smart Retail
8. Energy Engagement
9. IOT in Healthcare
10. IoT in Poultry and Farming

3.1.3 Intel Edison

While cell phones are getting to be greater at each model refresh, there's likewise another flood of progressively little, wearable gadgets hitting the market. The product development has developed in the previous years, with Arduino driving the prototyping equipment to advertise. A year ago, Intel presented a gadget that is engaging for creators yet in addition prepared for IoT and wearable items, the Intel Edison.

Edison gives intriguing chances to anybody with wearables/IoT thoughts who needs to investigate a genuine, item prepared stage. It's a stage rich with highlights that makes it simple to begin as most designers can use a dialect they definitely know. With a cool arrangement of highlights, prepared to utilize libraries and an open programming stage prepared to be hacked, Intel has influenced an incredible activity in their mean to make things to happen.

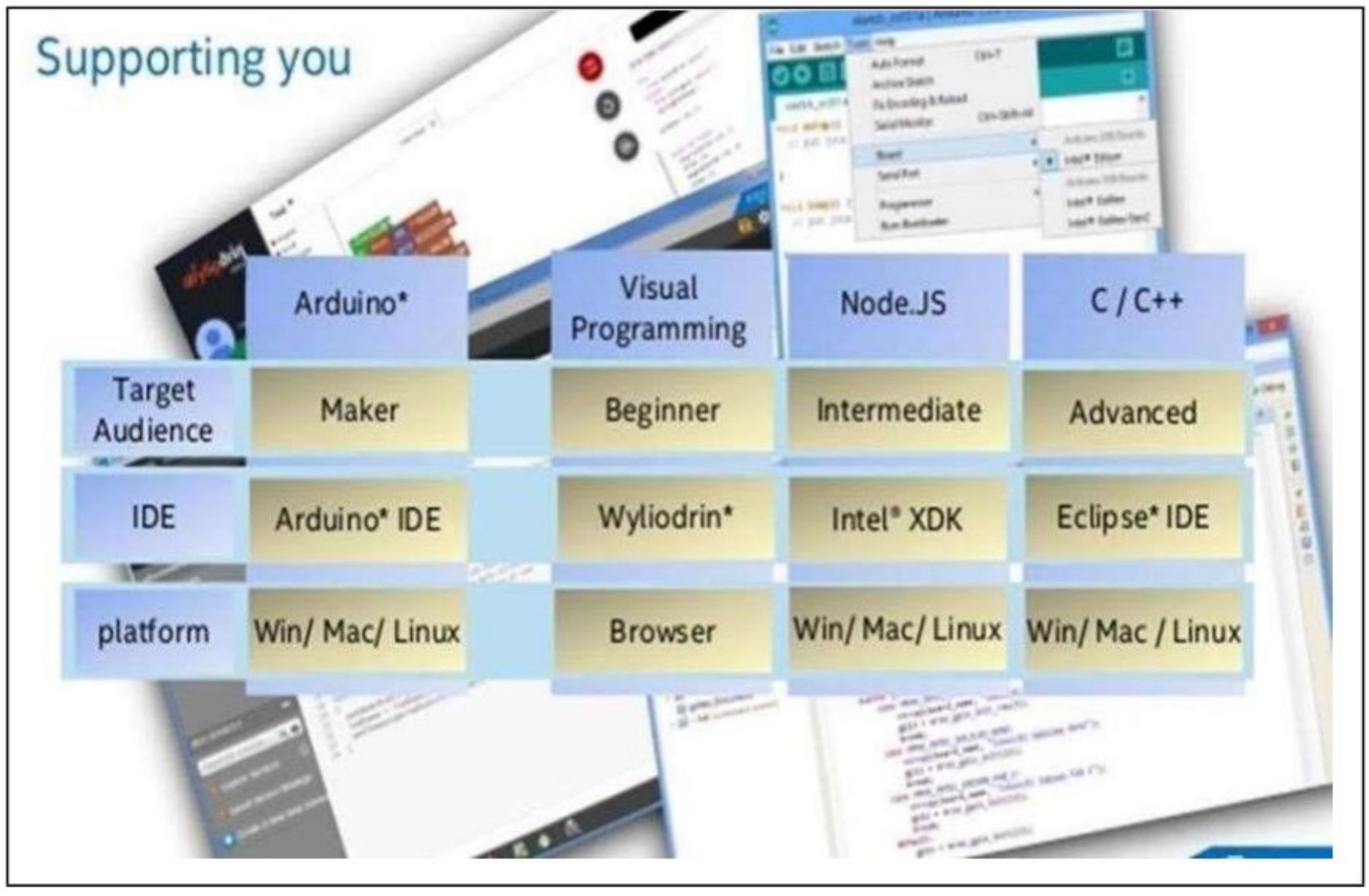


Figure 3.1.3 Intel Edison Supports

3.1.4 INTEL® EDISON MODULE

The Intel® Edison module is a SoC (System on Chip) that includes an

Intel® Atom™ 500MHz dual-core, dual-threaded CPU and an Intel® Quark™

100MHz microcontroller Key

Features:

Integrated Wi-Fi, Bluetooth 4.0 LE;

Support for Yocto Linux, Python, Node.js and Wolfram

Intel® Edison with Kit for Arduino

Fig.3.1.4: Intel® Edison Kit for Arduino

Intel® Edison Kit for Arduino provides the Arduino 1.0 pinout and standard connectors such as a micro-USB connected to a UART, a USB OTG port that can be switched between a second micro-USB device connector, a standard size USB host Type-A connector, an SD card holder, and a DC power jack.

Like an Arduino Uno, the Intel® Edison Kit for Arduino makes possible to have provides 20 digital input/output pins, of which 6 can be used as analog inputs. The Intel® Edison has 4 PWM outputs which can be configured via jumpers to any of the 6 pins supporting PWM on the Arduino Uno (pins 3, 5, 6, 9, 10, or 11).

The Intel® Edison Kit for Arduino is designed to be hardware and software pin-compatible with Arduino shields designed for the Arduino Uno R3. Digital pins 0 to 13 (and the adjacent AREF and GND pins), Analog inputs 0 to 5, the power header, ICSP header, and the UART port pins (0 and 1), are all in the same locations as on the Arduino Uno R3.

The digital IOs and analog pins can be configured to operate at either 5V or 3.3V. The outputs can source or sink 24 mA at 3.3V and 32 mA at 5V.

Program:

```
Void setup () {

//set up routine runs once;

}

Void loop () {

//loop routine runs over and over again forever;

}
```

3.1.5Hardware Interfacing of Various Sensors:

Sensor: A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.

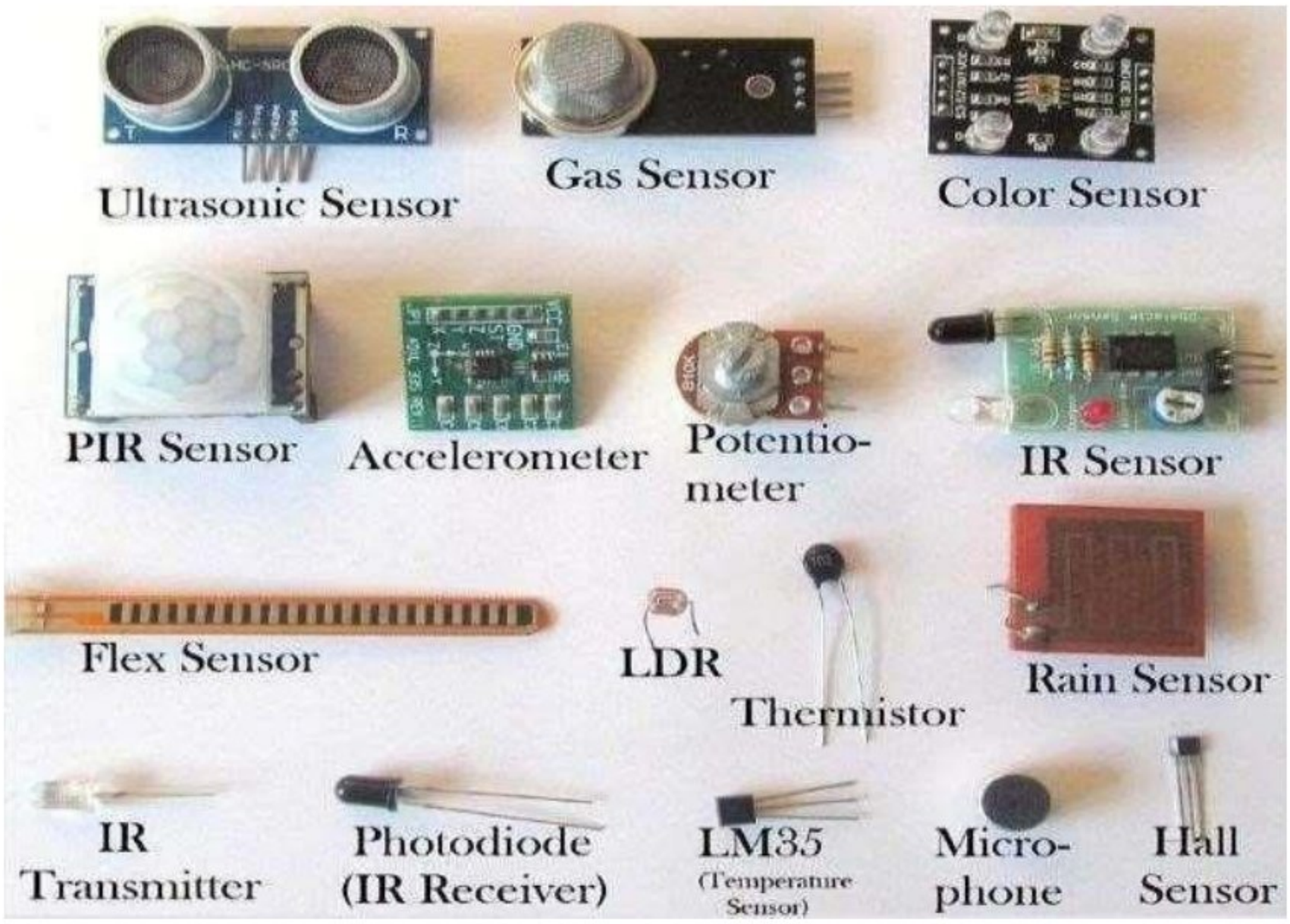


Fig.3.1.5 Types of Sensors

Analog Sensor:

Analog sensors produce a continuous output signal or voltage which is generally proportional to the quantity being measured. Physical quantities such as temperature, speed, pressure etc. all are analog quantities as they tend to be continuous in nature

Digital Sensor:

A digital sensor is an electronic sensor where data conversion and data transmission are done digitally. Sensors are often used for analytical measurements, e.g., the measurement of chemical and physical properties of liquids. Typically, measured parameters are pH value, conductivity, oxygen, redox potentials, and others. Such

measurements are used in the industrialized world and give vital input for process control.

3.2 Hands-On- ESP8266

3.2.1 ESP8266

ESP8266 is a low-cost Wi-Fi chip. ESP8266 chip is shown in Figure 1.1. This chip consists of a microcontroller and a full TCP/IP stack. ESP8266 is made by Es press if Systems, a Shanghai-based Chinese fabless semiconductor company. ESP8266 is a very popular chip for developing IoT devices. The ESP8266's specifications are listed as the following [1]:

Fig 3.2.1 ESP8266 wi-fi chip

- L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz

- Memory: – 32 KiB instruction RAM – 32 KiB instruction cache RAM – 80 KiB user-data RAM – 16 KiB ETS system-data RAM

- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)

- IEEE 802.11 b/g/n Wi-Fi – Integrated transmit/receive (TR) switch, balun, LNA, power amplifier, and matching network – WEP or WPA/WPA2 authentication, or open networks

- 16 GPIO pins (GPIO6–11 are used for communication with on board flash memory) , SPI , I 2C (software implementation) , I 2S interfaces with DMA (sharing pins with

GPIO) , UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2 , 10-bit ADC (successive approximation ADC) .

3.2.2 ESP-12E

ESP-12E is a Wi-Fi module that uses the ESP8266. ESP-12E is shown in Figure 1.2. ESP-12E is made by Ai-Thinker, a third-party manufacturer. There are other ESP modules made by this manufacturer. They are referred to as “ESP-xx modules”. The disadvantage of ESP-12E is that it is not breadboard friendly.

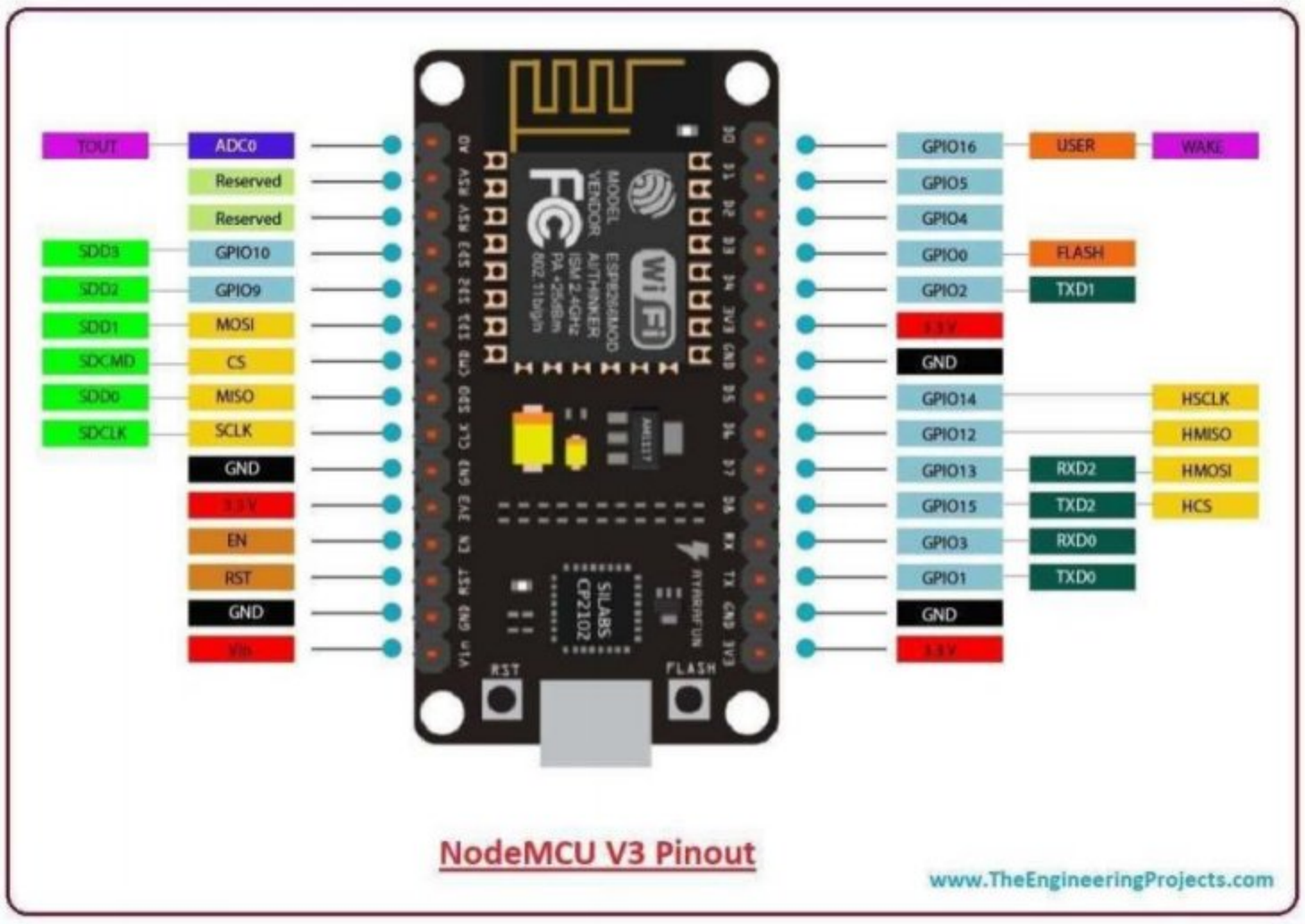
Figure 3.2.2 (a) ESP-12E

ESP-12E has an RF shield (metal enclosure) that covers the ESP8266 chip as shown in Figure 1.3. This shield is used for compliance with Federal Communications Commission (FCC) emissions rules. This shield can minimize interference with other devices

3.2.3 Node-MCU

Node-MCU is a development board that uses the ESP-12E. Node-MCU is shown in Figure 1.4. This is Node-MCU V3 which is used in this book. At the time of writing, it is the latest Node-MCU generation. Compared to the ESP-12E, NodeMCU is breadboard friendly and includes USB to serial interface. Node-MCU can be programmed using Lua scripting, Arduino, or ESP8266 SDK. In this book, we will program Node-MCU using Arduino, which is the easiest and most popular method.

Figure 3.2.3 Node-MCU IOT ESP8266 Development kit



3.3 UART COMMUNICATION

In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are

needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART.

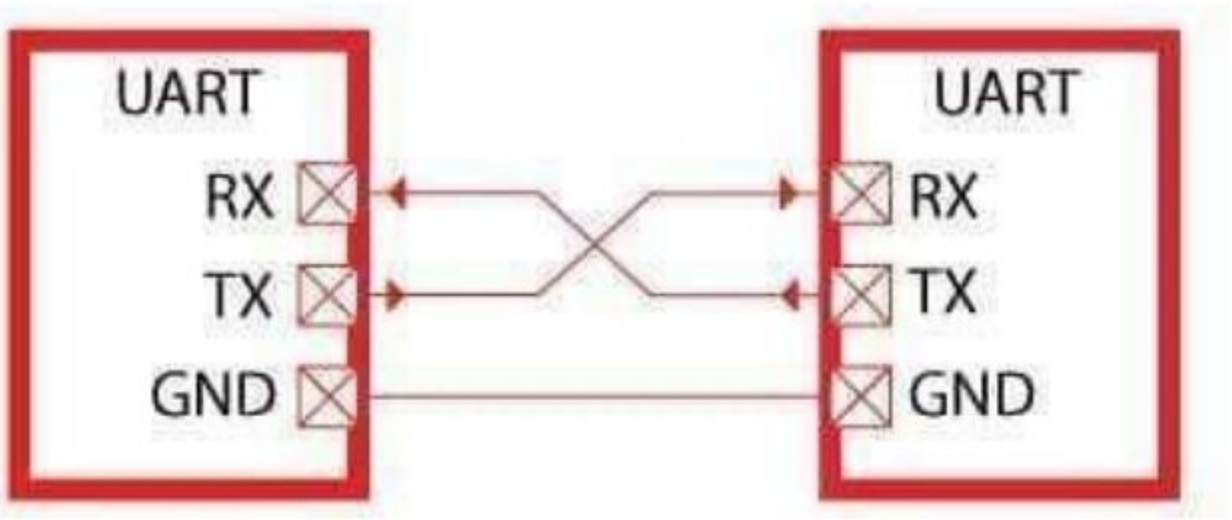


Figure 3.3.1 Connection Between Two UART

UARTs transmit data asynchronously, which means there is no clock signal to synchronize the output of bits from the transmitting UART to the sampling of bits by the receiving UART. Instead of a clock signal, the transmitting UART adds start and stop bits to the data packet being transferred. These bits define the beginning and end of the data packet so the receiving UART knows when to start reading the bits.

The UART that is going to transmit data receives the data from a data bus. The data bus is used to send data to the UART by another device like a CPU, memory, or microcontroller. Data is transferred from the data bus to the transmitting UART in parallel form. After the transmitting UART gets the parallel data from the data bus, it adds a start bit, a parity bit, and a stop bit, creating the data packet. Next, the data packet is output serially, bit by bit at the Tx pin. The receiving UART reads the data packet bit by bit at its Rx pin. The receiving UART then converts the data back into parallel form and removes the start bit, parity bit, and stop bits.

UART transmitted data is organized into packets. Each packet contains 1 start bit, 5 to 9 data bits (depending on the UART), an optional parity bit, and 1 or 2 stop bits:

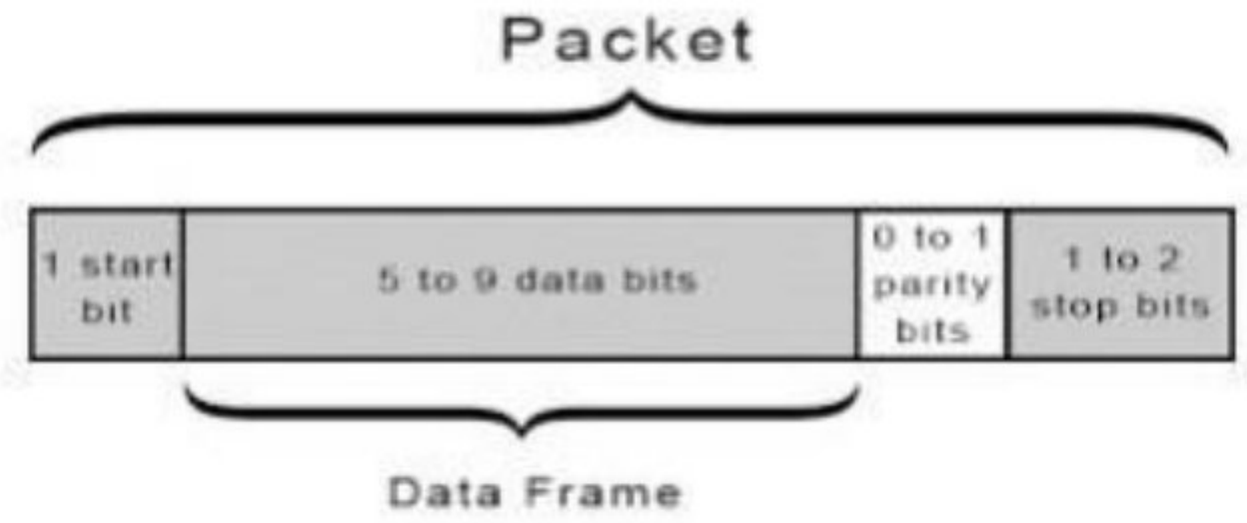


Figure 3.3.2 Basics of UART Communication

START BIT

The UART data transmission line is normally held at a high voltage level when it's not transmitting data. To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

DATA FRAME

The data frame contains the actual data being transferred. It can be 5 bits up to 8 bits long if a parity bit is used. If no parity bit is used, the data frame can be 9 bits long. In most cases, the data is sent with the least significant

bit first

PARITY

Parity describes the evenness or oddness of a number. The parity bit is a way for the receiving UART to tell if any data has changed during transmission. Bits can be changed by electromagnetic radiation, mismatched baud rates, or long-distance data transfers. After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and checks if the total is an even or odd number. If the parity bit is a 0 (even parity), the 1-bits in the data frame should total to an even number. If the parity bit is a 1 (odd parity), the 1-bits in the data frame should total to an odd number.

STOP BIT

To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for at least two bit durations.

STEPS OF UART TRANSMISSION

1. The transmitting UART receives data in parallel from the data bus.

2. The transmitting UART adds the start bit, parity bit, and the stop bit(s) to the data frame.
3. The entire packet is sent serially from the transmitting UART to the receiving UART. The receiving UART samples the data line at the preconfigured baud rate.
4. The receiving UART discards the start bit, parity bit, and stop bit from the data frame.

5. The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end.

ADVANTAGES :

1. Only uses two wires.
2. No clock signal is necessary.
3. Has a parity bit to allow for error checking.
4. The structure of the data packet can be changed as long as both sides are set up for it .
5. Well documented and widely used metho.

DISADVANTAGES :

1. The size of the data frame is limited to a maximum of 9 bits.
2. Doesn't support multiple slave or multiple master systems.
3. The baud rates of each UART must be within 10% of each other.

3.4 ARDUINO

Figure 3.4.1 ARduino Uno

Arduino is an open-source gadgets stage in light of simple to utilize equipment and programming. Arduino sheets can read inputs - light on a sensor, a finger on a catch, or a Twitter message - and transform it into a yield - initiating an engine, turning on a Drove, distributing something on the web. You can guide your board by sending an arrangement of guidelines to the microcontroller on the board. To do as such you utilize the Arduino programming dialect (in light of Wiring), and the Arduino Programming (IDE), in view of Handling.

A portion of the computerized I/O pin have unique capacities which are depict underneath.

1. Pins 0 and 1 are utilized for serial correspondence. They are utilized to get and transmit serial information which can be utilized as a part of a few different ways like programming the Arduino board and speaking with the client through serial screen.

2.Pins 2 and 3 are utilized for outer interferes. An outside occasion can be activated utilizing these pins by identifying low esteem, change in esteem or falling or rising edge on a flag.

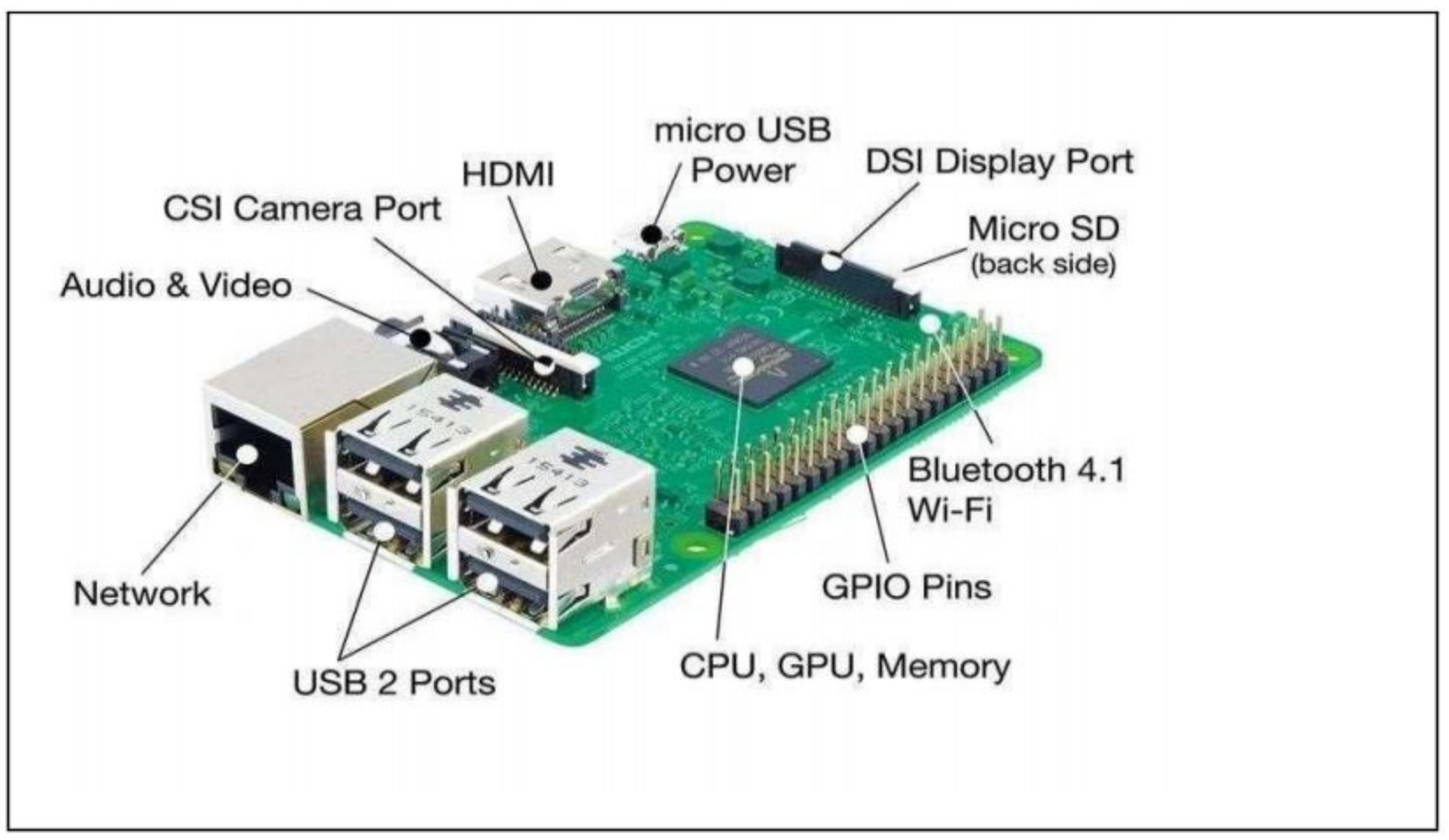
3.As specified before, 6 of the 14 advanced I/O Pins i.e., 3, 5, 6, 9, 10, and 11 can give 8bit PWM yield.

4.Pins 10, 11, 12 and 13 (SS, MOSI, MISO AND SCK individually) are utilized for SPI correspondence.

5.Pin 13 has a worked in Driven associated with it. At the point when the stick is HIGH, the Drove is turned on and when the stick is LOW, it is killed.

3.5 RASPBERRY PI

The Raspberry Pi is a minimal effort, charge card estimated PC that fittings into a PC screen or television, and utilizations a standard console and mouse. It is a fit little gadget that empowers individuals of any age to investigate figuring, and to figu¹e out how to program in dialects like Scratch and Python.



The raspberry pi comes in two models, they are model A and model B. the main difference between model A and model B is USB port. Model A board will consume less power and that does not include an Ethernet port. But the model B board includes an Ethernet port and designed in chin. The raspberry pi comes with a set of open-source technologies, i.e communication and multimedia web technologies. In the year 2014, the foundation of the raspberry pi board launched the computer module, that packages a model B raspberry pi board into module for use as a part of embedded systems, to encourage their use.

Raspberry Pi 3 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I²C)		DC Power 5v	04
05	GPIO03 (SCL1 , I²C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)		(I²C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Figure 3.5.2 Raspberry Pi Pin

We will talk about the programming of PI:

1.Take the 16GB small scale SD card and devote it particularly for PI OS.

2.Choose and Download OS programming.

[https://www.raspberrypi.org/downloads/]

3.Format the SD card and introduce OS on to the SD memory card utilizing advantageous strategies.

4.Take the SD card after OS establishment and embed it in PI board.

5.Connect screen, console and mouse

6.Power the board with small scale USB connector

7.Once the power is tuned ON the PI will keep running on the OS introduced in the memory card and will begin from boot.

8.Once all drivers are checked the PI will request approval, this is set as a matter of course and can be changed.

9.After approval you will achieve work area where all application program improvement begins.

Applications

1. Hobby projects.
2. Low-cost PC/tablet/laptop
3. IoT applications
4. Media centre
5. Robotics
6. Industrial/Home automation
7. Server/cloud server
8. Print server
9. Security monitoring
10. Web camera Gaming
11. Wireless access point
12. Environmental sensing/monitoring (e.g., WEATHER STATION)

3.6 Getting started with Blynk

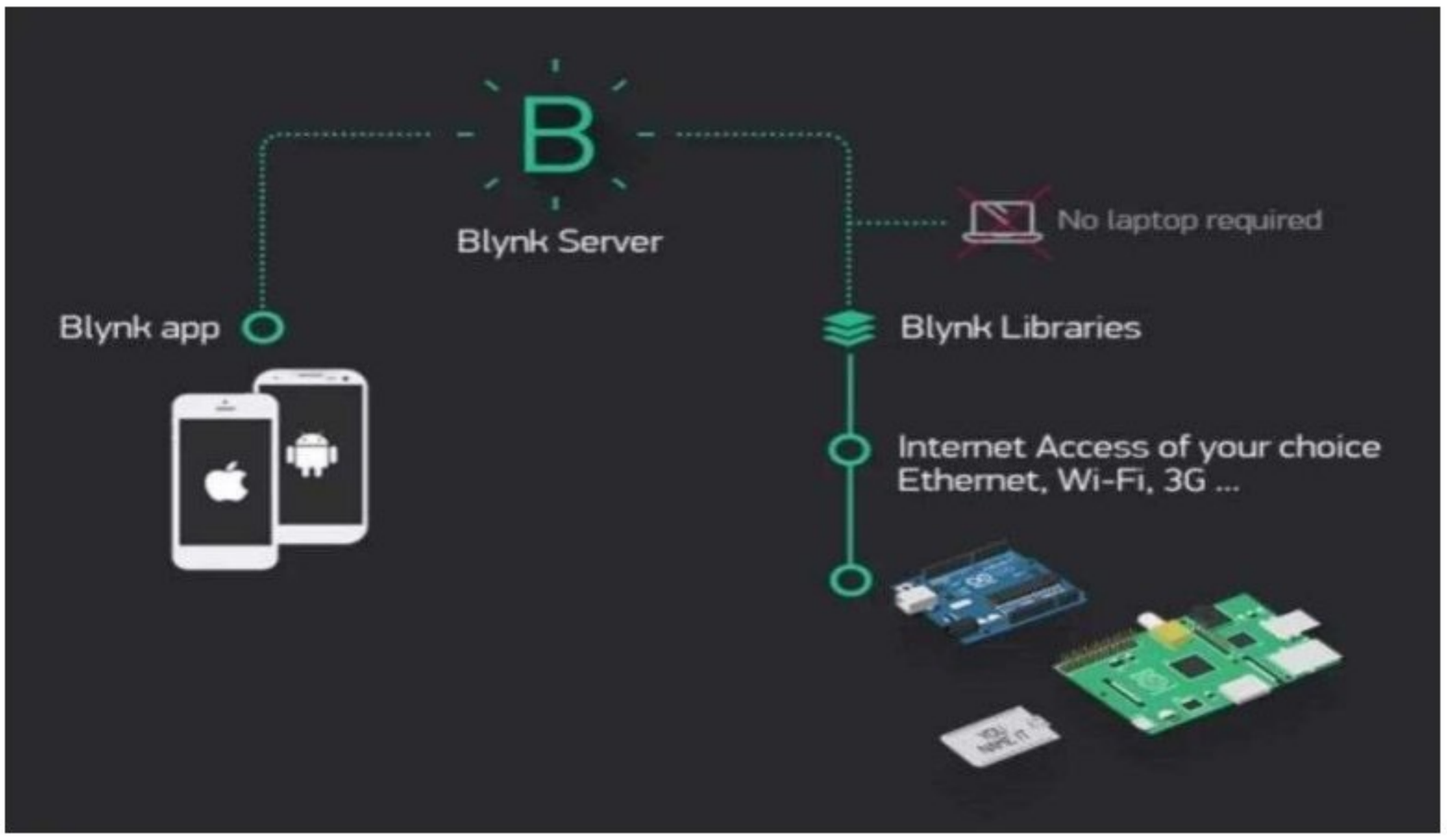


Fig.3.8.1:Downloading Blynk Application for Android

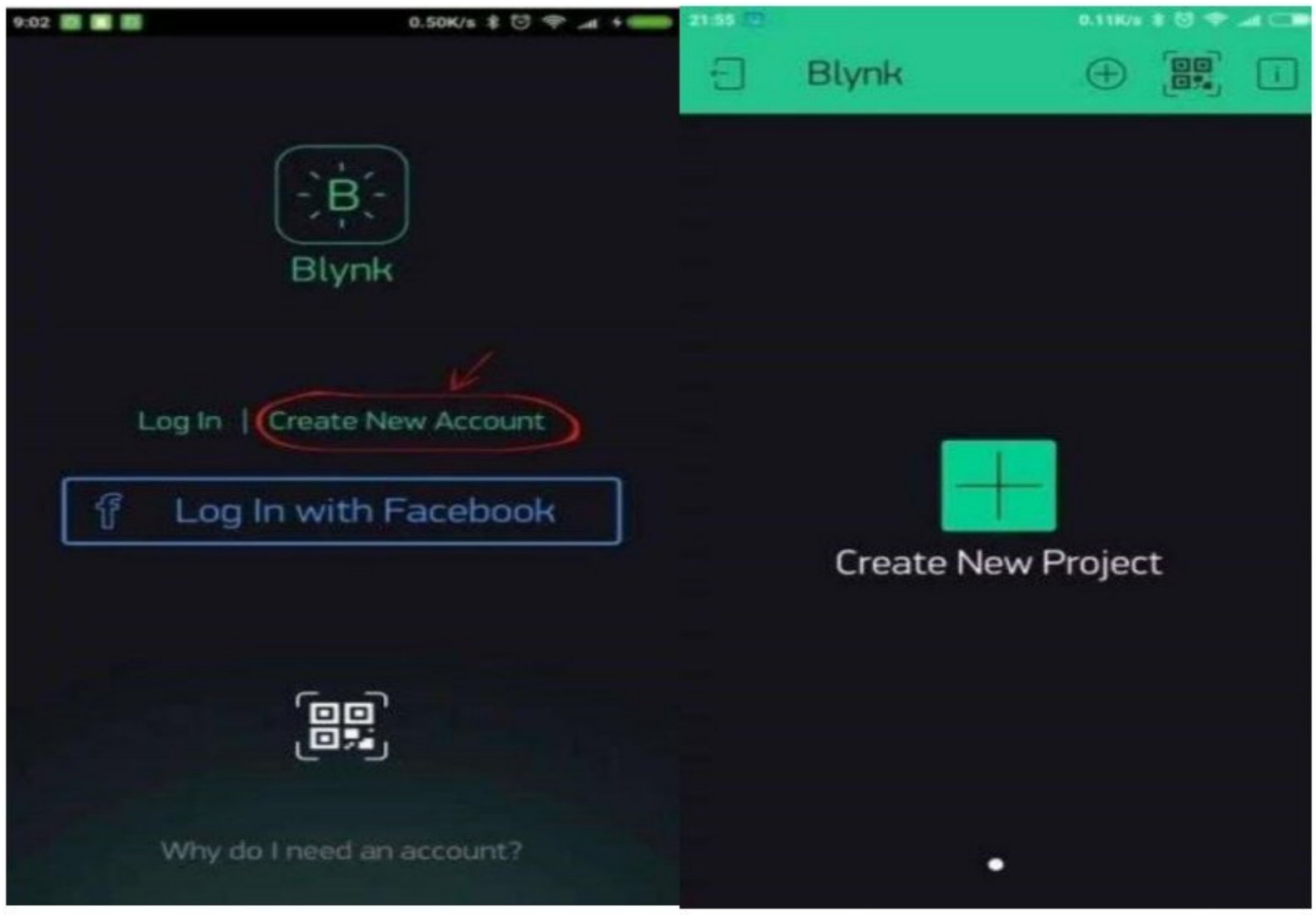


Fig.3.8.2:Creating New Account and Fig.3.8.3:Creating New Project

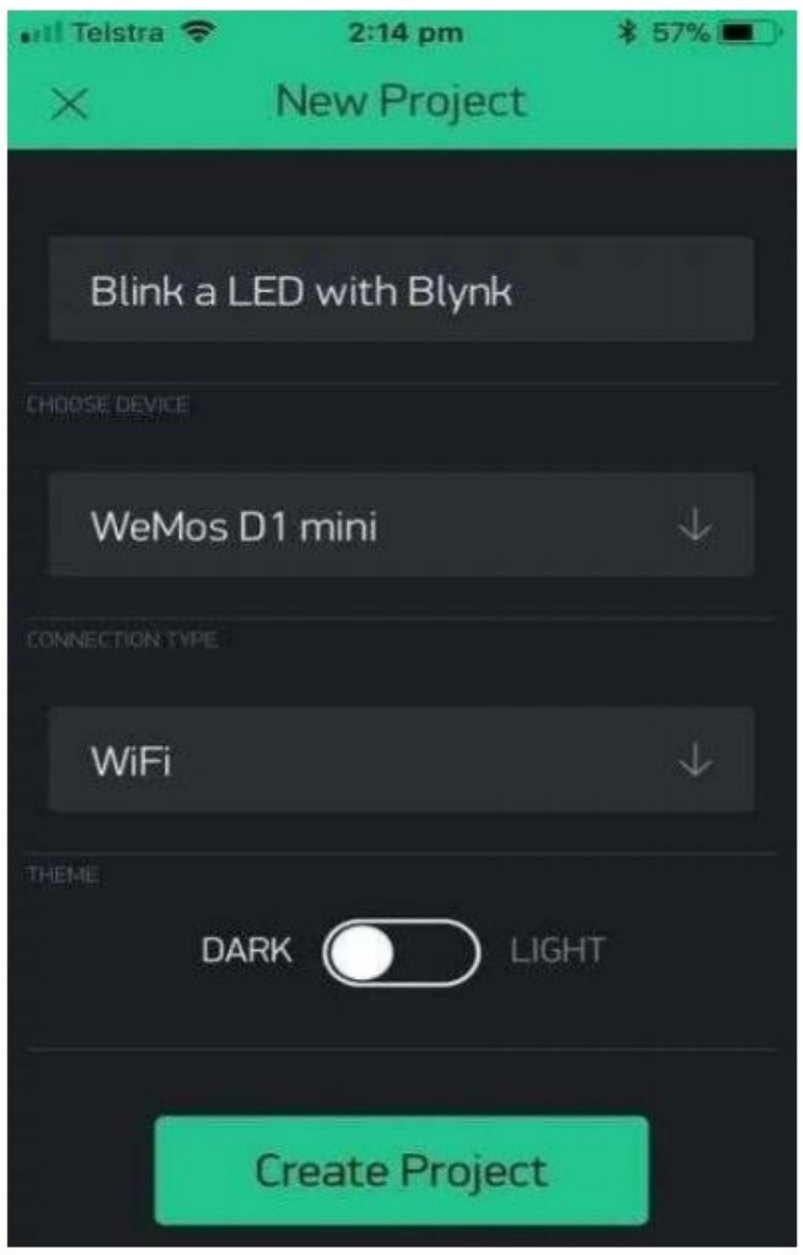


Fig.3.8.4:App Setting- Name and Board Selection

1. GET THE AUTH TOKEN

a. In request to interface Blynk Application and your equipment, you require an Auth Token.

b. Create another record in Blynk Application.

c. Create Another Undertaking. At that point pick the board and association you will utilize.

d. After the task was made, we will send you Auth Token over email.

Check your email inbox and discover the Auth Token.

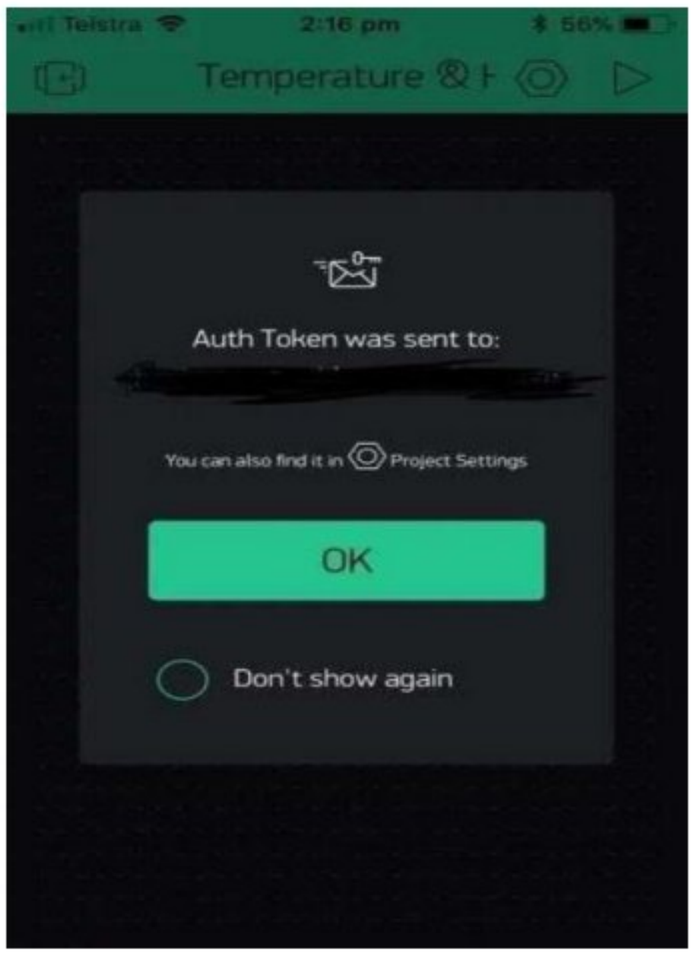


Fig.3.8.5:Getting of Auth Token

2. INSTALL BLYNK LIBRARY

$ sudo apt-get update

$ sudo apt-get install git-core

$ git clone git://git.drogon.net/wiringPi

$ ls

$ cd wiringPi

$ ./build

$ git clone http://dithub.com/blynkkk/blynk-library
$ cd blynk-library/Linux

$ make clean all target=raspberrypi

$ sudo ./blynk – token=token num["53e4da8793764b6197fc44a673ce4e21"]

You should see something like this

```
___ __          __
  / _ )/ /_ _____ / /__
 / _  / / // / _ \/ '_/
/____/_/\_, /_//_/_/\_\
       /___/ v0.4.4
```

[1240] Connecting to Your WIFI

[1240] Connected to your WIFI

[1240] My IP: 192.168.10.172

[1240] Blynk v0.4.4

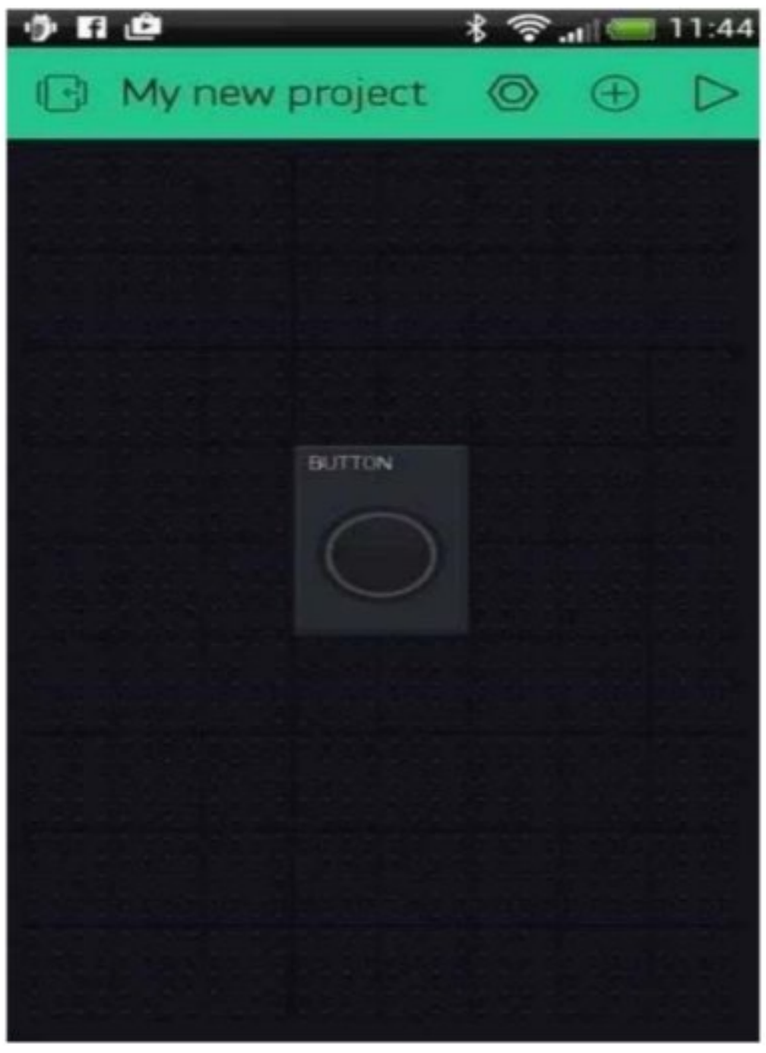
[5001] Connecting to blynk-cloud.com:80

[5329] Ready (ping: 1ms)

Congrats! You are connected!

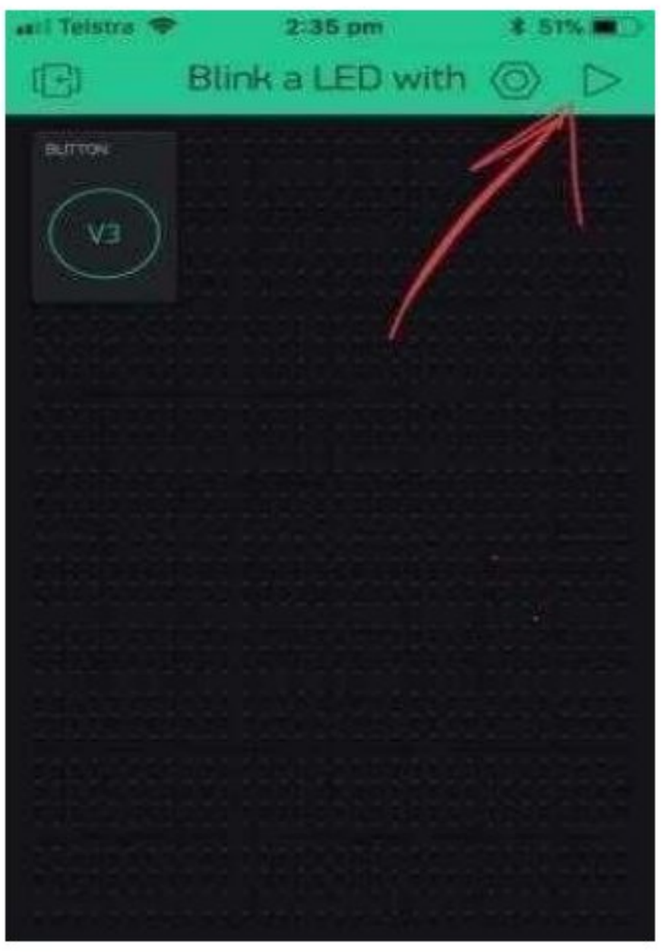
3. Add Widget

4. Drag and Drop



5. Widget settings

6. Run the project



3.7 Project

3.7.1 GPS Based Home Appliance Control

Introduction

Home automation is a topic which gaining popularity day by day, because of large advantages. One can achieve home automation by simply connecting home appliance electrical devices to the internet or cloud storage. the reason for this surge demand of network enabled home automation is reaching the zenith in recent days for its simplicity and comparable affordability. Platforms based on cloud computing help to connect to the thing's surroundings everyone so that one can find it easy to access anything and everything at any time and place in user friendly manner using custom defined portals. Hence, cloud act as a front end to access IOT. Here we are assuming a system which can control devices through wireless based network or cloud-based approach. In project we use IOT based home automation system which goal is to develop a home automation system that gives the user complete control over all remotely controllable aspects of his or her home. The automation system will have ability to be controlled from a central host PC, the internet, and also remotely accessed via a packet PC with a windows mobile based application.

In this we are implementing the above project with the help of simple led blink using blynk app

Fig 3.7.1 GPS Based Home Appliance

Components Used

1. Node-MCU
2. Bread board
3. Led lights
4. Blynk app
5. Connecting Wires
6. USB

Circuit Diagram

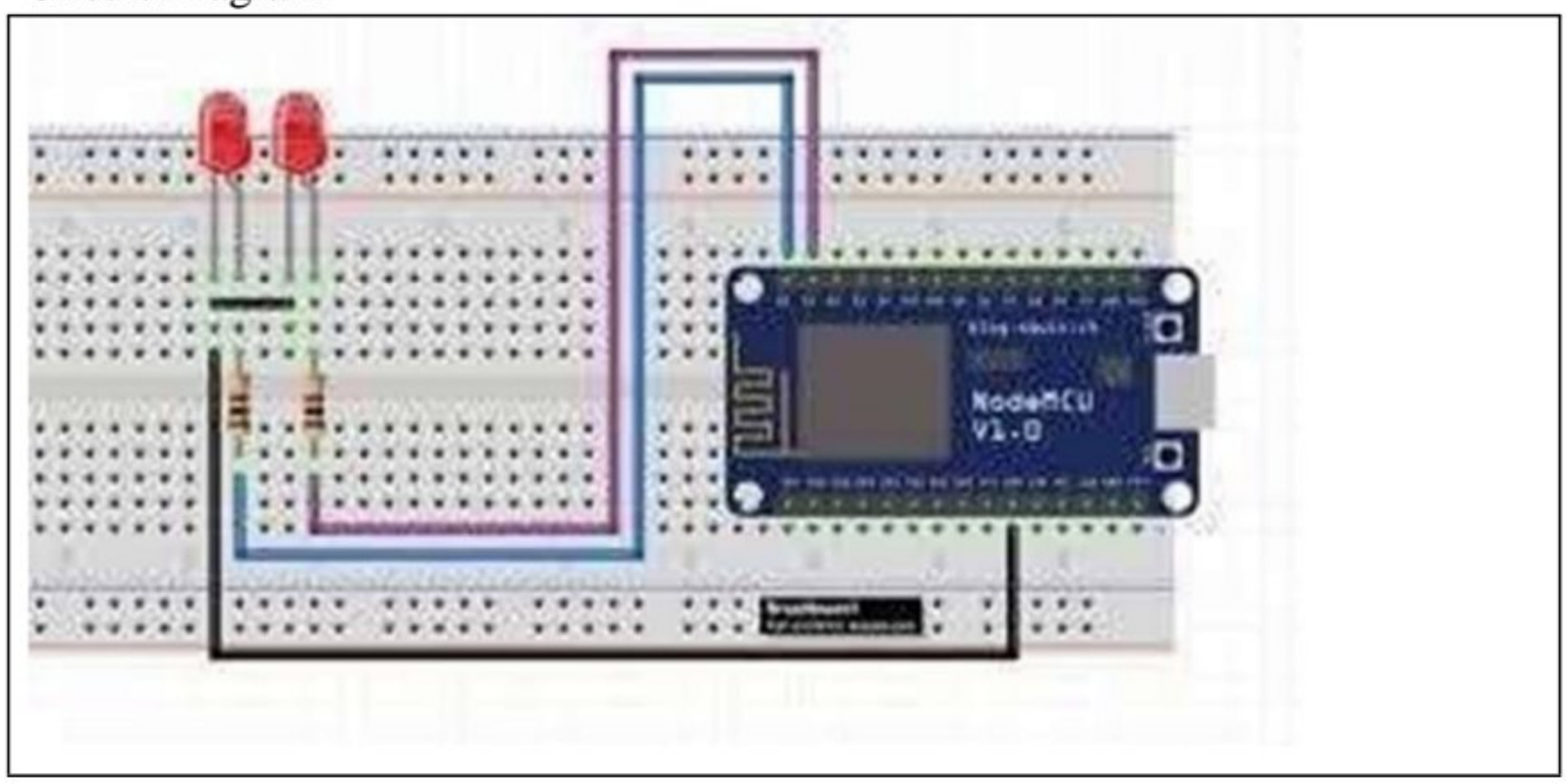


Fig.3.7.2 Circuit dia gram

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 Expressive Systems, and hardware which is based on the ESP-12 module.

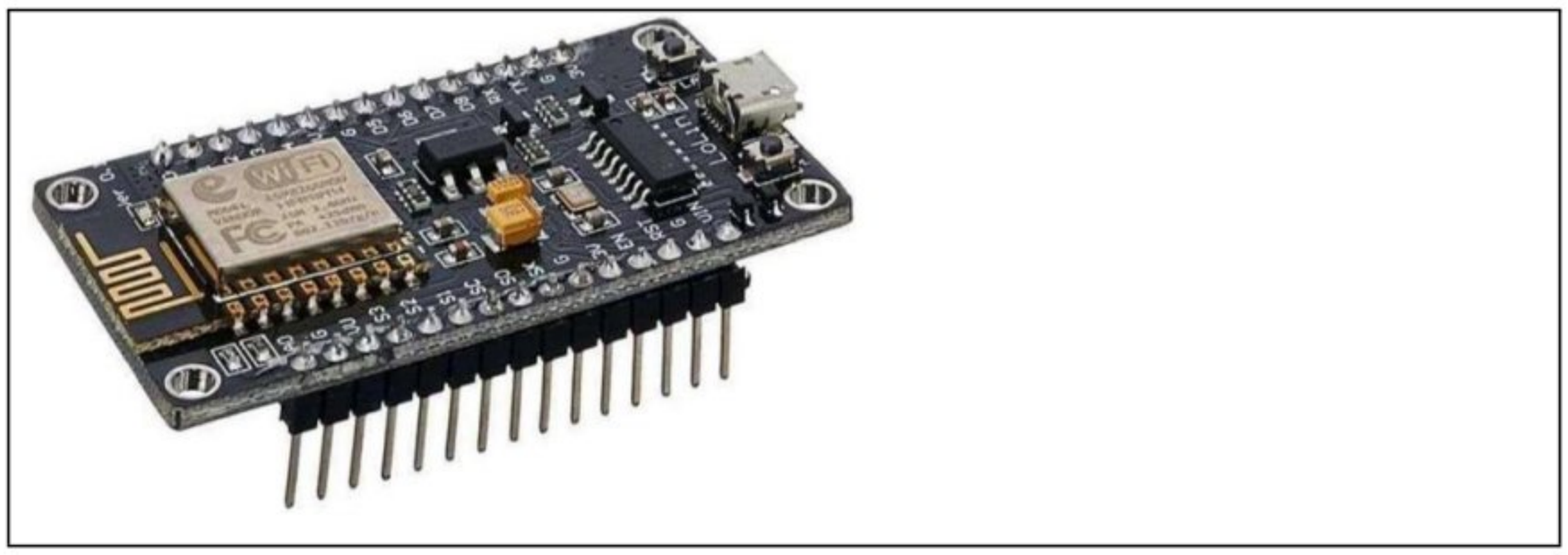


Fig 3.7.3 Node-MCU

How does it work? There are three major components in the platform:

Blynk App - allows to you create amazing interfaces for your projects using various widgets we provide.

Blynk Server - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Its features:

*Supports majority of development boards like Arduino, RPI, esp8266

* Easy to use

* Awesome widgets like LCD, push buttons, labelled value, graphs

* Not restricted to local Wi-Fi network

*Direct pin manipulation with no code writing

*Easy to integrate and add new functionality using virtual pins

Setting Up Blynk

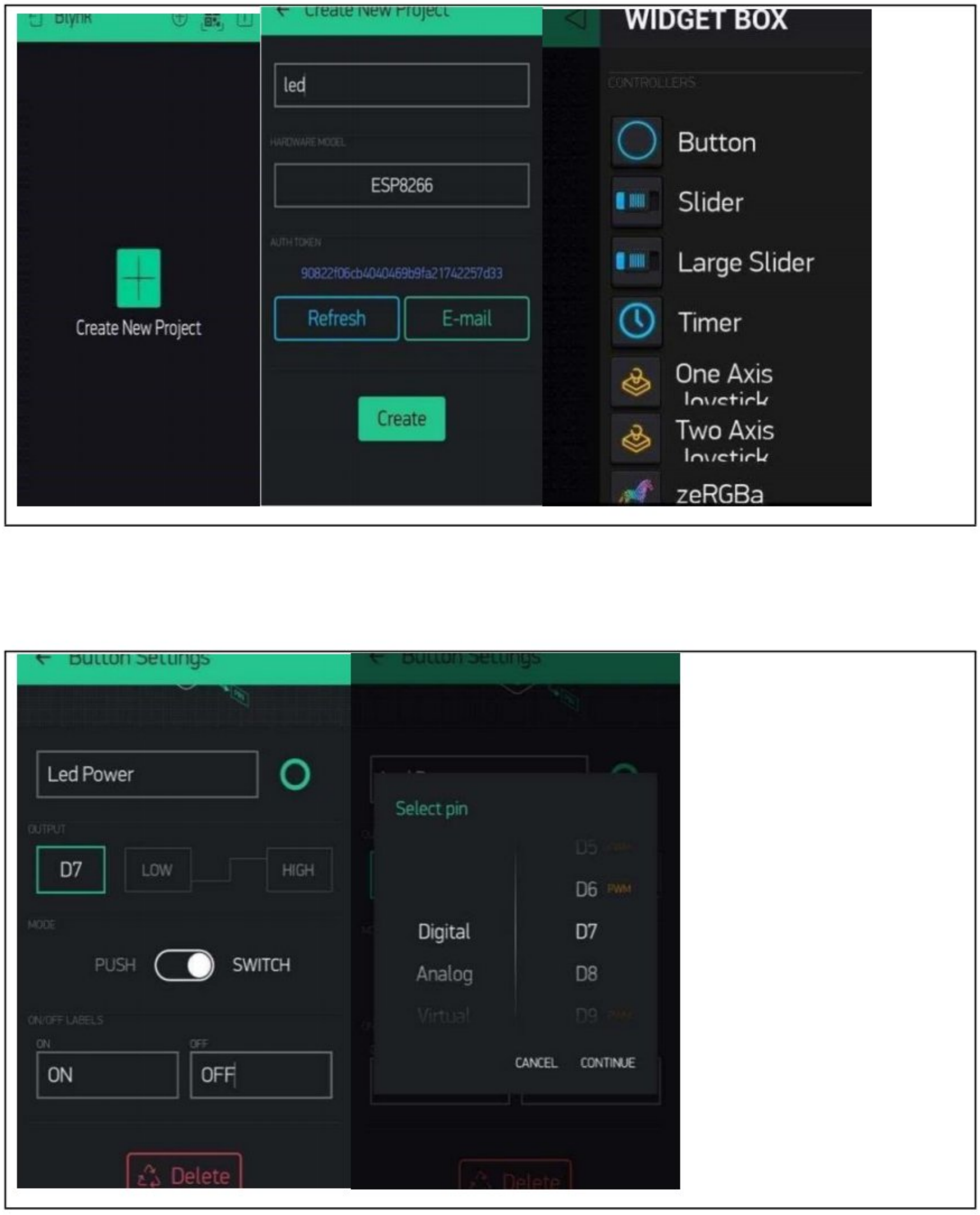


Fig.3.7.4 Setting up of blynk

Final Execution

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "YourAuthToken";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8442);
}

void loop()
{
  Blynk.run();
}
```

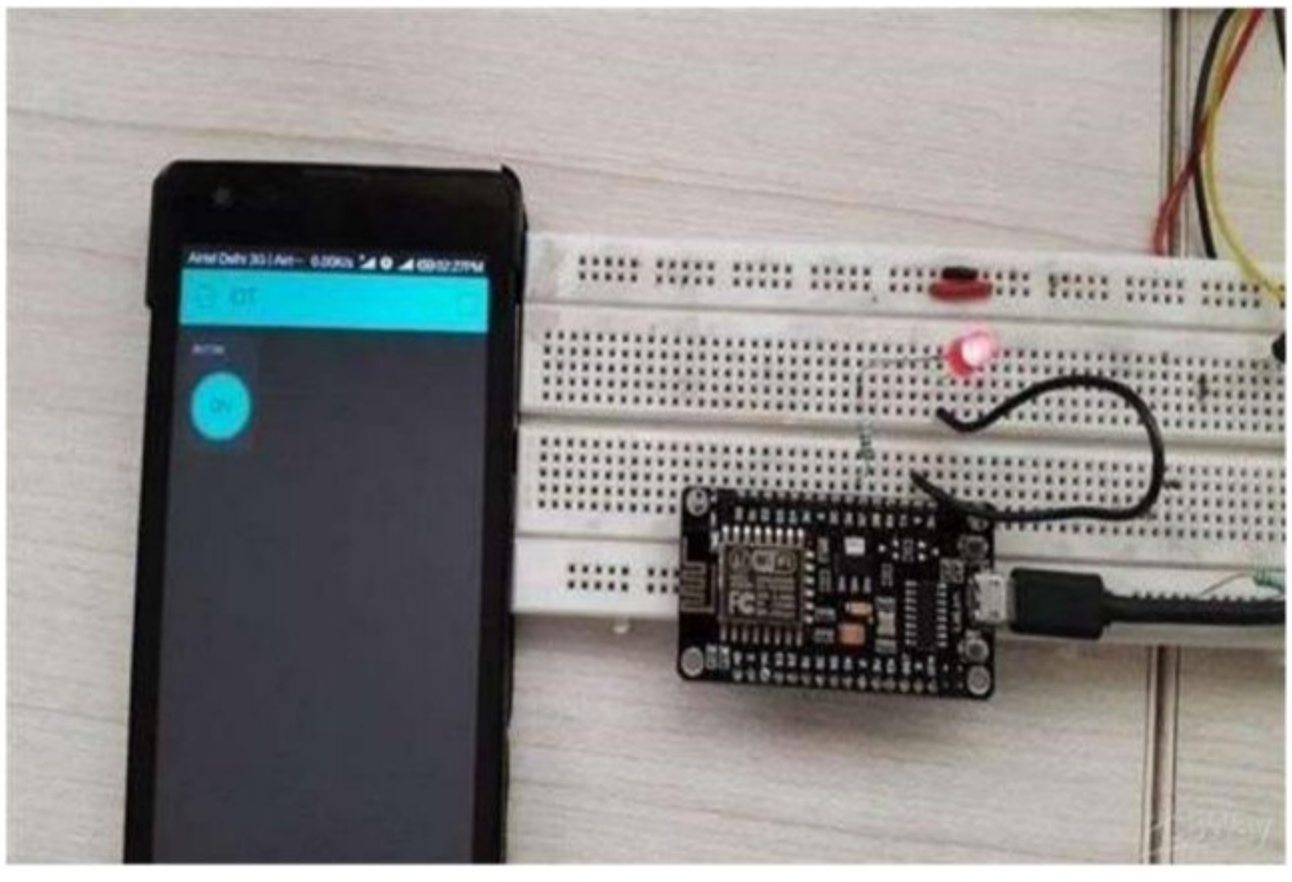


Fig.3.7.5 Final Execution

CHAPTER4

REFLECTION

The technical skills that I learned or improved at the training, other than those gained at university

- Hardware's: Intel Edison Arduino kit, Arduino Uno, Raspberry Pi
- Programming language: Python, Arduino Programming, Shell, SQL
- Linux – general operation
- Applying distinctive kinds of information procurement with Intel edison equipment and intel Arduino programming effectively out of the blue.
- Learning how to manage any new sensor, chip or any electronic component by referring to its datasheets.
- Increasing my understanding of how a successful design process should flow in a step by-step sequence with troubleshooting arising problems and reading more at each step. This was gained by both practicing and monitoring work.
- Increasing my ability to work as a member in a team. Actually, this had been an issue of mine; I did not believe much in teamwork. Now, after the training, I can say that my belief in the benefits of teamwork, when the team is concerted, has increased a lot.
- Acquiring increased ability in dealing with hardware and programming.

CHAPTER5

CONCLUSION

In conclusion, I am well satisfied with my training. I have learned numerous new specialized subjects, acquired a number of new technical skills and improved another group of existing skills, other than those picked up at college research facilities. I enjoyed most about my preparation that it is firmly identified with scholastic materials and research centres we considered in the college. This negates the normal saying that next to no of the materials educated in college building courses are utilized by engineers working in the work advertise. However, this does not imply that I have adapted minimal new things in my training.

REFERENCES

- An IoT based Health Monitoring System using Arduino Uno, International Journal of Engineering Research & Technology Richa, Anwesha Das, Ajeet Kumar Kushwaha, Prof. Mini Sreejith (IJERT) ISSN: 2278-0181 Vol. 10 Issue 03, March-(2021).

- Smart Home Automation System using Arduino and IOT V. Sudharani , D. Siva, M. Vijaya Raju, International Journal of Science and Research (IJSR), ISSN: 2319-7064, (2017).

- Home Automation And Security By Using Arduino And IoT, S. Magesh Kumar International Journal of Advanced Science and Technology, (2020).

- Internet of Things (IoT): BLYNK Framework for Smart Home Ermi Media's, Syufrijal, and Muhammad Rif'an 3rd UNJ International Conference on Technical and Vocational Education and Training,Volume (2019).

- Controlling Arduino board with smartphone and Blynk via internet, M. Todica , Researchgate publication November, (2016).