IOT BASED HOME AUTOMATION SYSTEM

Final year project report submitted to the department of Electrical Engineering in partial fulfilment of the requirements for the degree of Bachelor of Technology in electrical engineering .

By:

ARAV ABHI

REGISTRATION NO-037520 OF 2019-20

UNIVERSITY ROLL NO - 11701619033

ABHISHEK KUMAR

REGISTRATION NO -037596 OF 2019-20

UNIVERSITY ROLL NO- 11701619027

Under the supervision of

Dr. Shilpi Bhattacharya

HOD Department of EE and Associate professor

Department of Electrical Engineering

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Canal South Road, Beliaghata, Kolkata 700 015, West Bengal

Maulana Abul Kalam Azad Institute of Information Technology

(formerly WBUT)

CERTIFICATE

To whom it may concern

The undersigned certify that they have read and recommended to the Department of Electrical Engineering, RCC Institute of Information Technology, a final year project work entitled “IOT based Home Automation System” submitted by Arav Abhi (11701619033) and Abhishek Kumar (117101619027) , has been prepared under our supervision for the partial fulfilment of the requirements for B.tech(EE) degree in MAKAUT ,West Bengal. The report is hereby forwarded.

_____

Dr. Shilpi Bhattacharya

Associate Professor and

HOD Department of EE

RCC Institute of Information Technology

Countersigned by

_____

Dr. Shilpi Bhattacharya

Associate Professor and

HOD Department of EE

RCC Institute of Information Technology, Kolkata -700015

ACKNOWLEDGEMENT

We express our sincere gratitude to Dr. Shilpi Bhattachary, Associate Professor, HOD Department of Electrical Engineering, RCC Institute of Information Technology, as the mentor for our project. It is our great fortune that we have got opportunity to carry out this project work under the supervision. We express our sincere thanks for the encouragement, support and the guidance. We would further like to thank all the faculty member for their cooperation and extended support in undergoing the project work. We also would thank the department of Electrical Engineering to provide the infrastructure and facility in carrying the project like these.

Date: ……2023

_____

Arav Abhi

Reg .No :- 037520 of 2019-20

Roll No:- 11701619033

B.Tech (EE)-8th sem 2023,RCCIIT

_____

Abhishek kumar

Reg. No- 037596 of 2019-20

Roll No- 117101619027

B.Tech (EE)-8th sem 2023,RCCIIT

ABSTRACT

This project presents the overall design of Home Automation System (HAS) with low cost and wireless system. It specifically focuses on the development of an IOT based home automation system that is able to control various components via internet or be automatically programmed to operate from ambient conditions. In this project, we design the development of a firmware for smart control which can successfully be automated minimizing human interaction to preserve the integritywithin whole electrical devices in the home. We used Node MCU, a popular open source IOT platform,to execute the process of automation. Different components of the system will use differenttransmission mode that will be implemented to communicate the control of the devices by the userthrough Node MCU to the actual appliance. The main control system implements wireless technologyto provide remote access from smart phone. We are using a cloud server-based communication thatwould add to the practicality of the project by enabling unrestricted access of the appliances to the user irrespective of the distance factor. We provided a data transmission network to create a stronger automation. The system intended to control electrical appliances and devices in house with relatively low cost design, user-friendly interface and ease of installation. The status of the appliance would beavailable, along with the control on an android platform. This system is designed to assist and provide support in order to 1fulfil the needs of elderly and disabled in home. Also, the smart home concept in the system improves the standard living at home.

LIST OF FIGURES

ABBREVIATION AND ACRONYMS

SL. NO	ACRONYM	EXPANSION
1.	IOT	Internet of Things
2.	RF Comm	Radio Frequency Communication
3.	NodeMCU	Node Micro Controller Unit
4.	Wi-Fi	Wireless Fidelity
5.	SSL	Secure Socket Layer
6.	GPIO	General Purpose Input/Output
7.	NFC	Near Field Communication
8.	LAN	Local Area Network
9.	LoRaWAN	Low Power Wide Area Network
10.	DIP	Dual In-line Package
11.	UBW	Ultra-Wide Band
12.	PIR	Passive Infrared Sensor
13.	UID	Unique Identifier
14.	HAS	Home Automation System
15.	TCP	Transmission Control Protocol
16.	SSH	Secure Socket Shell
17.	IIOT	Industrial Internet of Things
18.	GSM	Global System for Mobile
19.	BLE	Bluetooth Low Energy
20.	SoC	System on a Chip
21.	USB	Universal Serial Bus
22.	LDR	Light Dependent Resistor
23.	HMI	Human Machine Interaction
24.	MQTT	Message Queue Telemetry Transport
25.	WSN	Wireless Sensor Network
26.	NLP	Natural Language Processing
27.	PCB	Printed Circuit Board

Internet of Things (IOT) is a concept where each device is assign to an IP address and through that IP address anyone makes that device identifiable on internet. The mechanical and digital machines are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Basically, it started as the “Internet of Computers.” Research studies have forecast an explosive growth in the number of “things” or devices that will be connected to the Internet. The resulting network is called the “Internet of Things” (IoT). The recent developments in technology which permit the use of wireless controlling environments like, Bluetooth and Wi-Fi that have enabled different devices to have capabilities of connecting with each other. Using a WIFI shield to act as a Micro web server for the Arduino which eliminates the need for wired connections between the Arduino board and computer which reduces cost and enables it to work as a standalone device. The Wi-Fi shield needs connection to the internet from a wireless router or wireless hotspot and this would act as the gateway for the Arduino to communicate with the internet. With this in mind, an internet based home automation system for remote control and observing the status of home appliances is designed.

Due to the advancement of wireless technology, there are several different type of connections are introduced such as GSM, WIFI, and BT. Each of the connection has their own unique specifications and applications. Among the four popular wireless connections that often implemented in HAS project, WIFI is being chosen with its suitable capability. The capabilities of WIFI are more than enough to be implemented in the design. Also, most of the current laptop/notebook or Smartphone come with built-in WIFI adapter. It will indirectly reduce the cost of this system.

1.1 BACKGROUND

The concept of “Home Automation” has been in existence for several years. “Smart Home”, “Intelligent Home” are terms that followed and is been used to introduce the concept of networking appliance within the house. Home Automation Systems (HASs) includes centralized control and distance status monitoring of lighting, security system, and other appliances and systems within a house. HASs enables energy efficiency, improves the security systems, and certainly the comfort and ease of users. In the present emerging market, HASs is gaining popularity and has attracted the interests of many users. HASs comes with its own challenges. Mainly being, in the present day, end users especially elderly and disabled, even though hugely benefited, aren’t seen to accept the system due to the complexity and cost factors.

THEORY

2.IOT (INTERNET OF THINGS)

IOT as a term has evolved long way as a result of convergence of multiple technologies, machine learning, embedded systems and commodity sensors. IOT is a system of interconnected devices assigned a UIDS, enabling data transfer and control of devices over a network. It reduced the necessity of actual interaction in order to control a device. IOT is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

2.1 Features of IOT

2.1.1 Intelligence

IOT comes with the combination of algorithms and computation, software & hardware that makes it smart. Ambient intelligence in IOT enhances its capabilities which facilitate the things to respond in an intelligent way to a particular situation and supports them in carrying out specific tasks. In spite of all the popularity of smart technologies, intelligence in IOT is only concerned as a means of interaction between devices, while user and device interaction are achieved by standard input methods and graphical user interface

2.1.2Connectivity

Connectivity empowers the Internet of Things by bringing together everyday objects. Connectivity of these objects is pivotal because simple object level interactions contribute towards collective intelligence in the IOT network. It enables network accessibility and compatibility in the things. With this connectivity, new market opportunities for the Internet of things can be created by the networking of smart things and applications

2.1.3Dynamic Nature

The primary activity of Internet of Things is to collect data from its environment, this is achieved with the dynamic changes that take place around the devices. The state of these devices change dynamically, example sleeping and waking up, connected and/or disconnected as well as the context of devices including temperature, location and speed. In addition to the state of the device, the number of devices also changes dynamically with a person, place and time.

2.1.4Sensing

IOT wouldn't be possible without sensors that will detect or measure any changes in the environment to generate data that can report on their status or even interact with the environment. Sensing technologies provide the means to create capabilities that reflect a true awareness of the physical world and the people in it. The sensing information is simply the analog input from the physical world, but it can provide a rich understanding of our complex world

2.1.5Heterogeneity

Heterogeneity in Internet of Things as one of the key characteristics. Devices in IOT are based on different hardware platforms and networks and can interact with other devices or service platforms through different networks. IOT architecture should support direct network connectivity between heterogeneous networks. The key design requirements for heterogeneous things and their environments in IOT are scalabilities, modularity, extensibility and interoperability.

2.1.6Security

IOT devices are naturally vulnerable to security threats. As we gain efficiencies, novel experiences, and other benefits from the IOT, it would be a mistake to forget about security concerns associated with it. There is a high level of transparency and privacy issues with IOT. It is important to secure the endpoints, the networks, and the data that is transferred across all of it means creating a security paradigm.

NodeMCU (Node Microcontroller Unit) is a low-cost open source IOT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added.

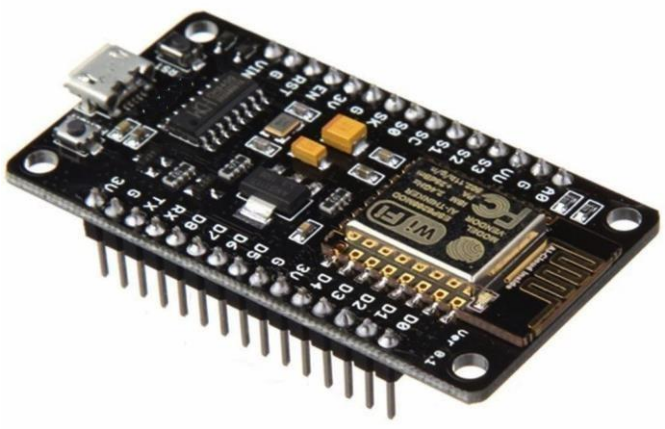


Figure 1. Node MCU Development Board.

NodeMCU is an open source firmware for which open source prototyping board designs are available. The name “NodeMCU” combines “node” and “MCU” (micro-controller unit). The term “NodeMCU” strictly speaking refers to the firmware rather than the associated development kits.

Both the firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua- cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially was based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IOT applications.

2.2.1Pin Configuration of Node MCU Development Board

This module provides an access to the GPIO subsystem. All the access is based on I/O index number of Node MCU kits, not the internal GPIO pins. For example, the D0 pin on the development kit is mapped to GPIO pin 16. Node MCU provides access to the GPIO pins and the following pin mapping table is a part of the API documentation.

PIN NAME ON NODE MCU DEVELOPMENT KIT	ESP8266 INTERNAL GPIO PIN NUMBER	PIN NAME ON NODE MCU DEVELOPMENT KIT	ESP8266 INTERNAL GPIO PIN NUMBER
0 [*]	GPIO16	**7**	GPIO13
1	GPIO5	**8**	GPIO15
2	GPIO4	**9**	GPIO3
3	GPIO0	**10**	GPIO1
4	GPIO2	**11**	GPIO9
5	GPIO14	**12**	GPIO10
6	GPIO12		

Table 1. Node MCU index ↔ GPIO mapping.

[*] D0 (GPIO16) can only be used for GPIO read/write. It does not support open-drain/interrupt/PWM/I²C or 1-Wire.

The ESP8266 Node MCU has total 30 pins that interface it to the outside world. The pins are grouped by their functionality as:

Power pins: There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

GND: is a ground pin of ESP8266 Node MCU development board.

12 IC Pins: are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins: ESP8266 Node MCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel: The Node MCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins: ESP8266 Node MCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins: ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer
- Up to 80 MHz and the divided clocks of 80 MHz
- Up to 64-Byte FIFO

SDIO Pins: ESP8266 features Secure Digital Input/output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins: The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 µs to 10000 µs, i.e., between 100 Hz and 1 kHz.

Control Pins: are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- RST pin – RST pin is used to reset the ESP8266 chip.
- WAKE pin – Wake pin is used to wake the chip from deep-sleep.

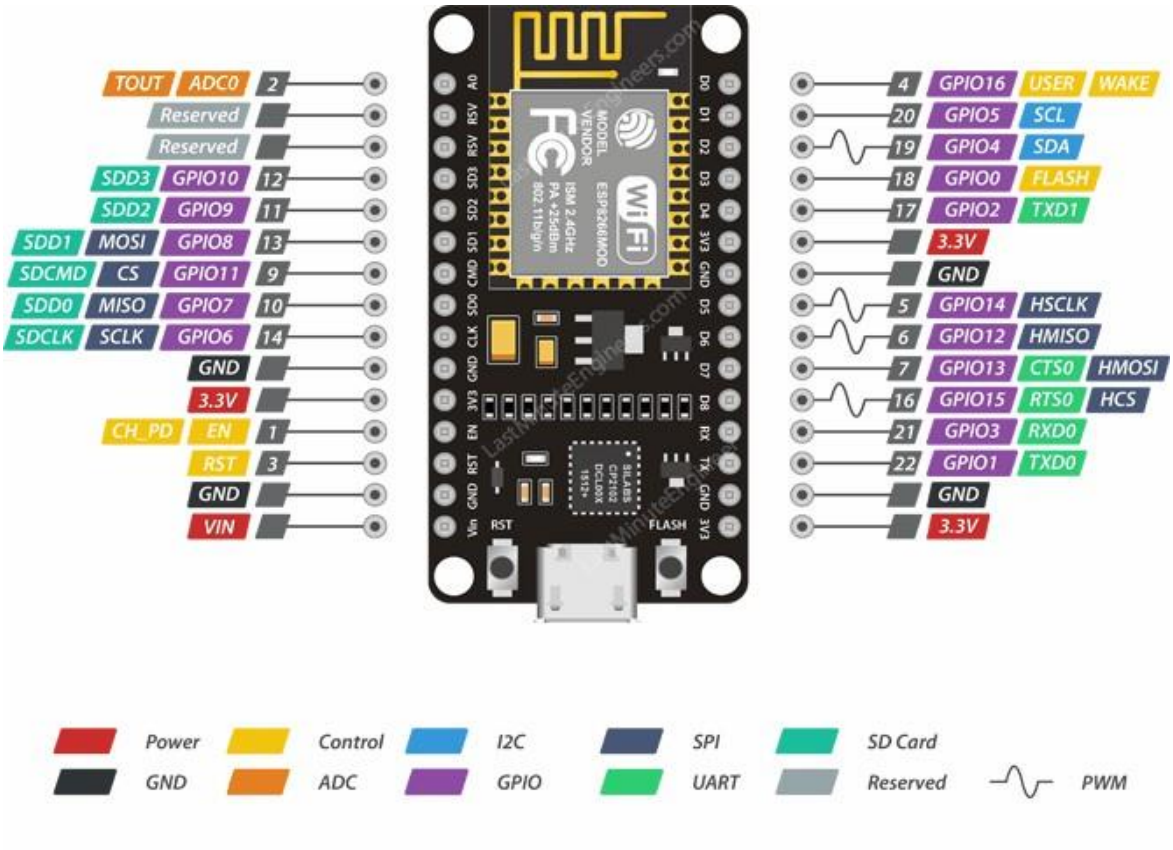
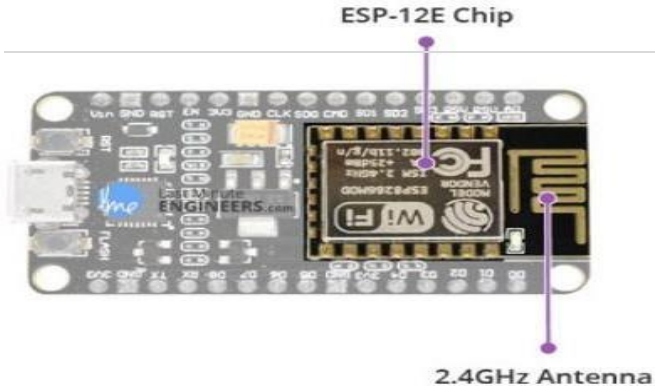


Figure 2. ESP8266 Node MCU pinout.

2.2.1Parts of Node MCU Development Board

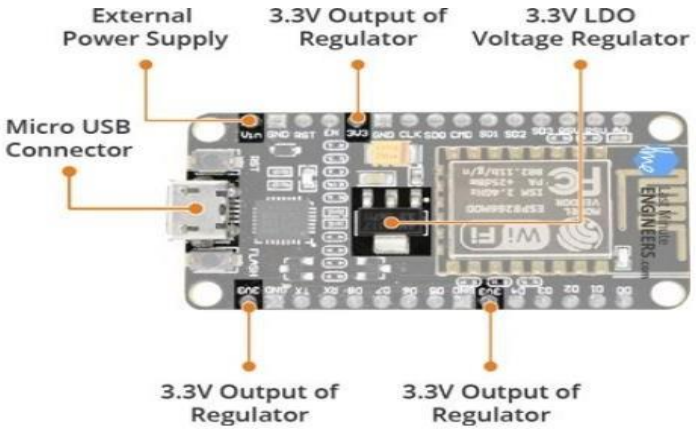
2.2.1.1ESP 12-E Module



- Tensilica Xtensa® 32-bit LX106
- 80 to 160 MHz clock frequency
- 128 kb internal RAM
- 4 MB external flash
- 802.11b/g/n HT40 Wi-Fi transceiver

Figure 3. ESP 12E module in Node MCU Development board.

2.2.1.2Power Requirements



- Operating voltage 2.5V to 3.6V
- On-board 3.6V 600mA regulator
- 80 mA operating current
- 20 µA during sleep mode

Figure 4. Power module on a Node MCU development board.

2.2.1.3Peripheral I/O

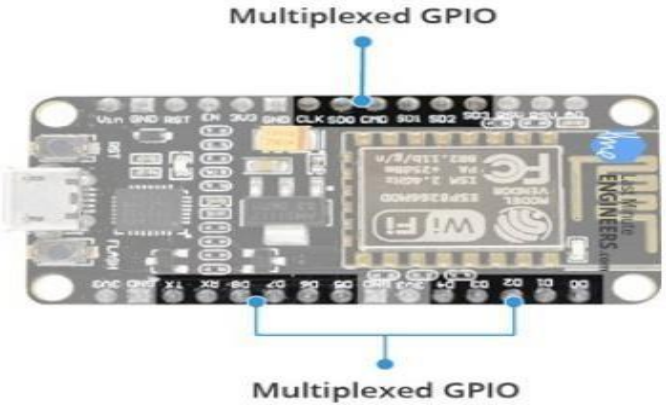


Figure 5. GPIO pins on Node MCU development board.

2.2.1.4 On Board Switches and LED Indicator

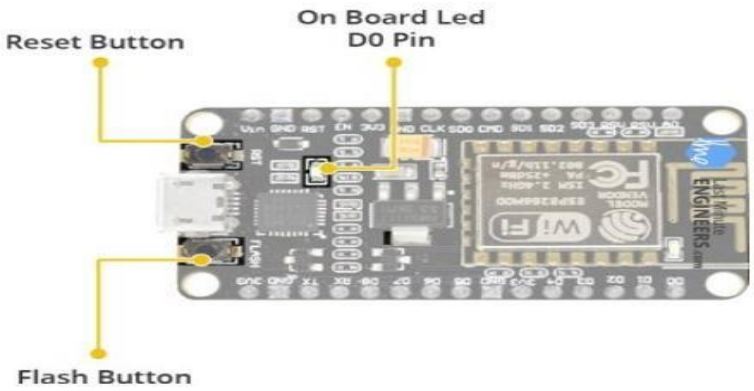
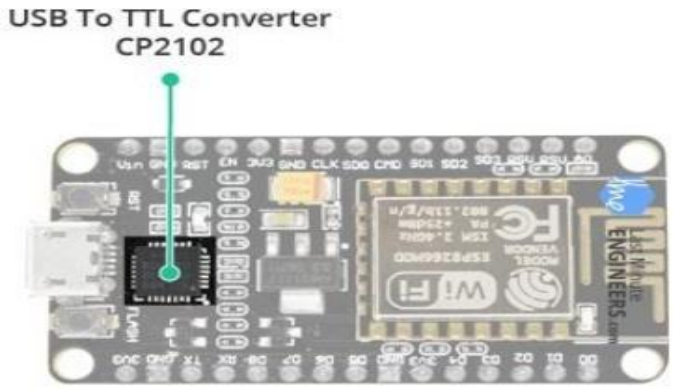


Figure 6. board switches and LED indicators on Node MCU development boar

2.2.1.5 Serial Communication



CP2120 USB-to-UART converter
4.5 Mbps communication speed
Flow control support

Figure 7. CP2120 on Node MCU development board.

2.2.1 Installation of Node MCU

Mostly these days devices download and install drivers on their own, automatically. Windows doesn't know how to talk to the USB driver on the Node MCU so it can't figure out that the board is a Node MCU and proceed normally. Node MCU Amica is an ESP8266 Wi-Fi module based development board. It has got Micro USB slot that can directly be connected to the computer or other USB host devices. Ti has got 15X2 header pins and a Micro USB slot, the headers can be mounted on a breadboard and Micro USB slot is to establish connection to USB host device. It has CP2120 USB to serial converter. In order to install CP2120 (USB to serial converter), user is needed to download the driver for the same. Once user downloads drivers as per its respective operating system, the system establishes connection to Node MCU. The user needs to node down the COM post allotted to newly connected USB device (Node MCU) from device manager of the system. This com port number will be required while using Node MCU Amica. As the CP2120 driver is been installed, the Node MCU can be programmed using Arduino IDE software by coding in embedded C. this requires ESP8266 board installation in Arduino IDE from board manager, and assigning communication port.

2.3 BLOCK DIAGRAM

2.3.1 Block diagram of the proposed system

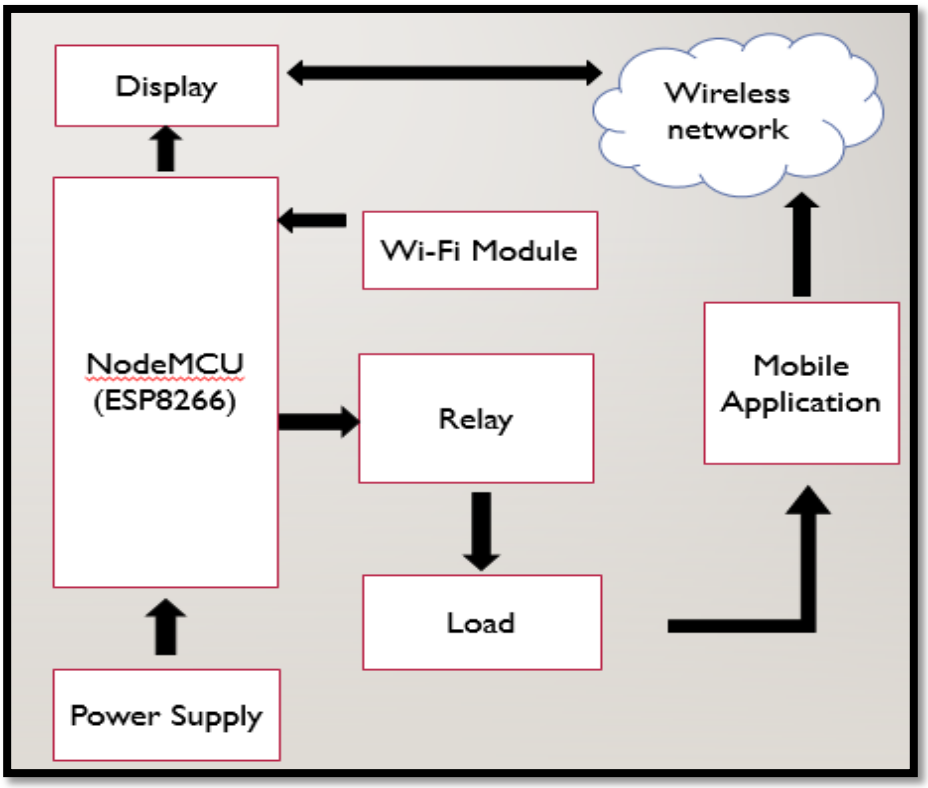


Figure 11. Block diagram of proposed system.

The block diagram gives the functionality of the overall project. The Node MCU unit is the microcontroller or the main controlling unit of the system. The user uses the mobile application in setting commands for functioning of the appliances. The mobile application interprets the command form in user in voice or switch mode and sends signal to the Node MCU unit, over a wireless network established by Wi-Fi communication. Hence the Wi-Fi module (actually inbuilt into Node MCU), helps the microcontroller establish Wi-Fi communication with a device and take commands from an application over wireless network. The Node MCU on further receiving the signal then turns on/off the appliance with the help of relay. The Node MCU, relay and the final appliances are physically connected. There is a power supply unit that powers the microcontroller, the relay as well as the final appliances. There is also a display unit that displays the status of the application.

2.3.2 Proposed system

The android OS provides the flexibility of using the open source. The inbuilt sensors can be accessed easily. The application used to control the system has the following features. Android Phone acts as a client and data are sent via sockets programming. The application takes command from user in two different modes.

- **Switch mode:** Switch mode uses the radio buttons that are used to control the home appliances. The radio button sends the status of the switch.
- **Voice mode:** Voice Mode is used to control the home appliances using voice command. Using the inbuilt microphone of Smartphone, the application creates an intent that fetches the speech data to the Google server which responds with a string data. The string data are further analysed and then processed.

More detailed discussion about the modes of control and how they actually control the system is discussed if coming chapters.

2.4 OVERVIEW OF PROJECT

The following describes the process of creating an account in Blynk application and generating unique ID against a particular device. This ID acts as an identifier for the particular device on the Blynk server.

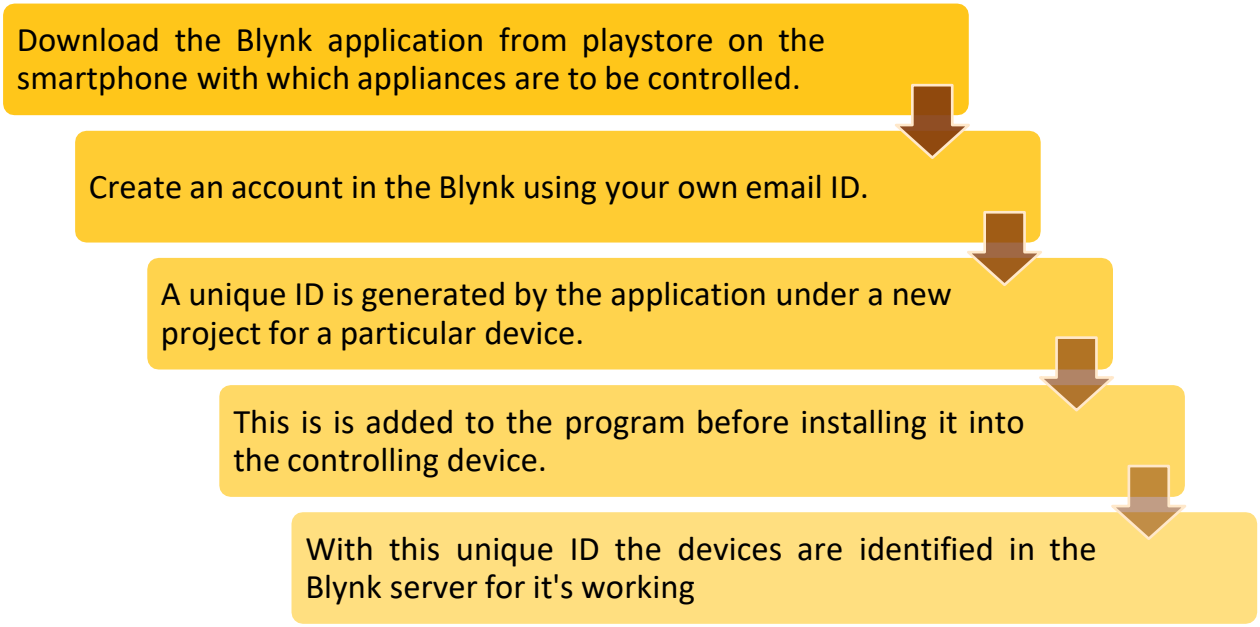


Figure 12. Creating an account and generating unique ID in Blynk Server.

Once the unique Id is generated the next step would be to include this key into the coding written in embedded C to establish communication between Node MCU and Blynk Server. The following describes this process.

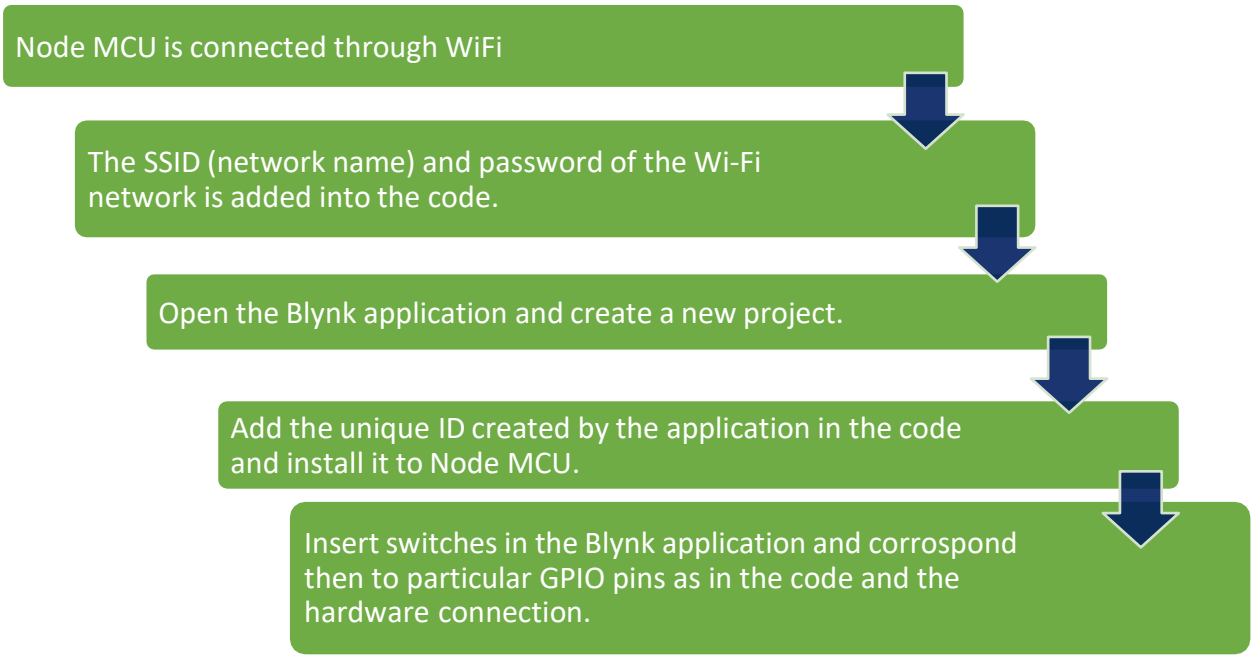


Figure 13. Setup to control Node MCU from Blynk application

2.5 CIRCUIT DIAGRAM

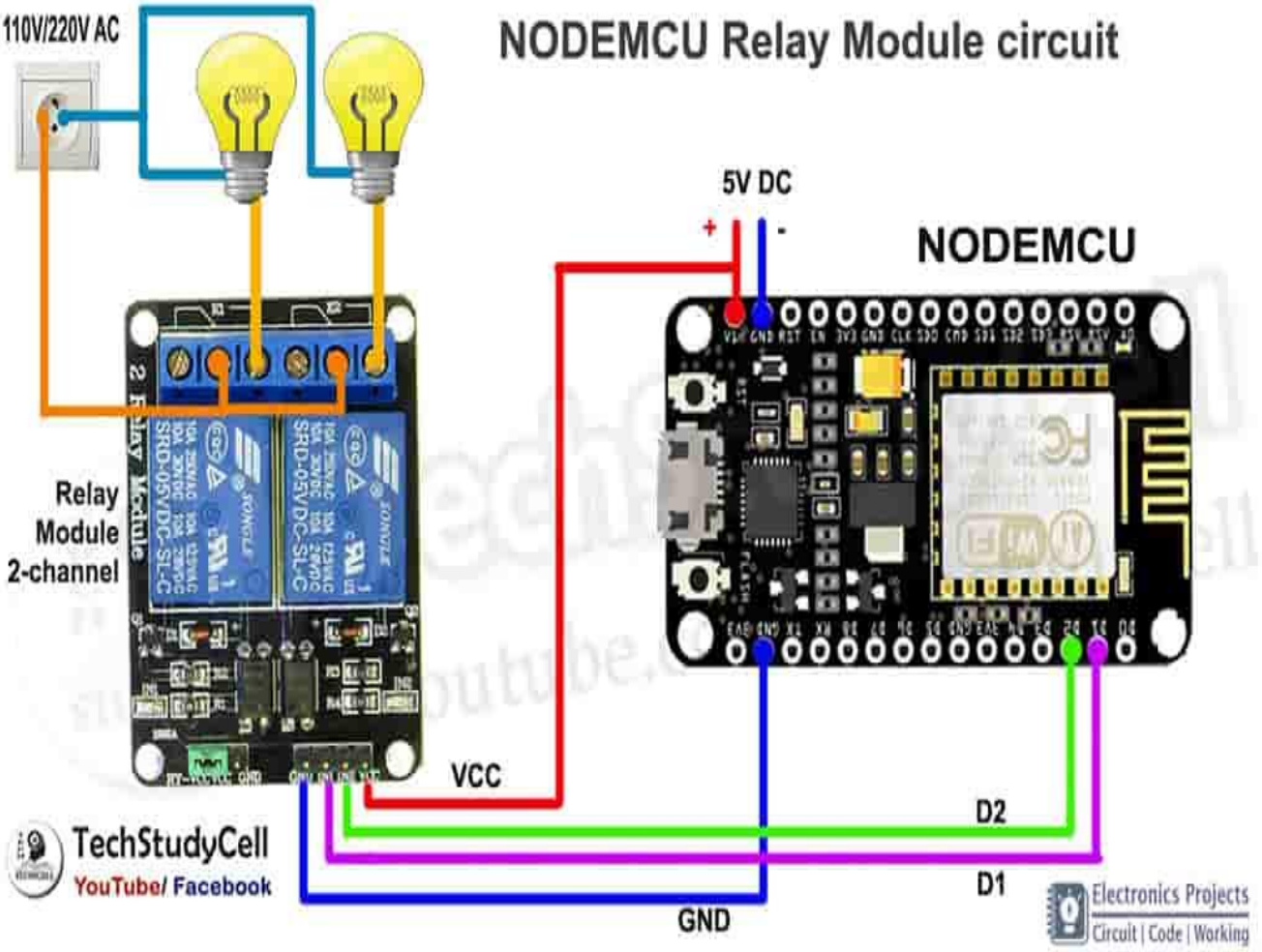


Figure 15. Connection diagram of Node MCU controlling 2 channel relay module.

HARDWARE MODELLING AND SETUP

3.1 MAIN FEATURES OF THE PROTOTYPE

The features of the developed prototype are:

- The prototype establishes a wireless remote switching system of home appliances.
- The prototype uses Wi-Fi to establish wireless control, which gives an indoor range to about 150 feet.
- The command to switch on and off an appliance can be given from radio buttons on the application from one's smartphone.
- There is also a provision developed to use voice commands on smartphone to remotely switch home appliances
- Any device capable of Wi-Fi connectivity can be used to control the prototype.
- The control over home appliances is obtained over secure connections, by SSL over TCP, SSH.
- Simple design easy to integrate into a verity of appliances and extend on further range.
- Displays the status of each appliances on the application in smartphone
- Cost effective.

3.2PROJECT LAYOUT

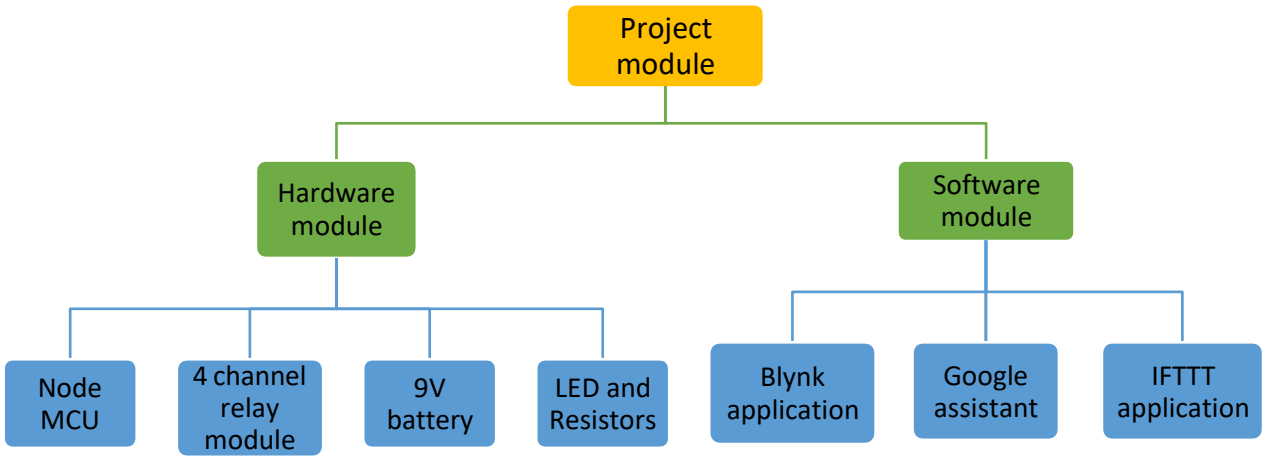


Figure 16. Layout of project modu

Node MCU is the microcontroller unit in the prototype. It has an in built Wi-Fi module (ESP8266) thatestablishes wireless remote switching of home appliances.

Four channel relay module consists 4 individual relays physically connected between Node MCU and the home appliances. It takes signals form GPIO pins of Node MCU and accordingly connects or disconnects home appliances from the supply. They act as the switching device.

LED and resistors are used in this prototype to replace real appliances. They indicate power being turned on and off to the appliances. In real time operation they would be replaced by actual home appliances.

Blynk application was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it, etc. the prototype primarily uses Blynk application to sense commands from user to the hardware over wireless network.

Google assistant is a system software present on the android phone. It interprets the voice commands by the user to turn on or off an appliances.

IFTTT application the voice commands interpreted by the google assistant isn't understandable by Blynk application thus unable to send to the hardware. IFTTT is an intermediate application that interprets commands from Google assistant and sends on and off signal to Blynk application Via Blynk server.

3.3COMPONENTS REQIRED

SL. NO	Component	Quantity
1.	Node MCU	1
2.	2 channel relay board	1
3.	BLUB	2
4.	LED	4
5.	2.2K Ω Resistor	4
6.	Blank PCB (KS100)	1
7.	Male pin header	1
8.	Female pin header	1
9.	Jumper wires	8
10.	USB Cable	1

Table 2. Component listing.

3.4SETTING UP THE SYSTEM

3.4.1Downloading and installing and Blynk application on smartphone

- Blynk application is downloaded and installed from the Play Store.
- Once the application is installed, a new account is created and logged in to it.
- After logging in, a new project is created. The project is named, hardware is selected as Node MCU and the connection type is selected as Wi-Fi, and created.
- At this point Blynk will sends an authentication token to email id. This authentication token will be used to identify the hardware in the Blynk server.
- As the prototype uses 2 channel relay module, 2 buttons are added to the screen from the side bar.
- All the 4 buttons are then customised by adding a name and selecting the digital pin it will correspond to. This section will actually affect the hardware connection as the relays will be physically connected to the digital pins corresponded here.
- The setup of Blynk application is now complete.

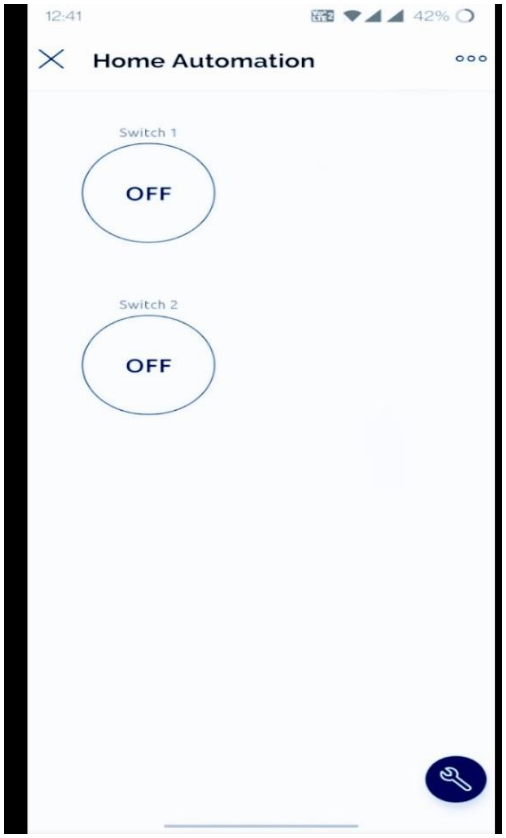


Figure 17. Set up Blynk application

3.5 HADWARE ASSEBLY

Hardware assembly mainly includes connecting specific digital pins of NodeMCU to the 2 relays on the relay module, including the connection of supply and ground pins. The main functional assemble in this prototype is simple. The further 4 relays are fit to be connected to any appliance desired to be controlled.

The vital part in hardware assembly is taking into account the digital pin that corresponds to which relay. This connection is done as per the setup of Blynk application. The radio buttons on Blynk application are set up to switch a particular digital pin in Node MCU. It is made sure that the relay connection are physically made according to this set up. For example, we have assigned the radio button on Blynk application corresponding to relay 1 to work with D3. Then physical connection of relay 1 is made with D3 of Node MCU.

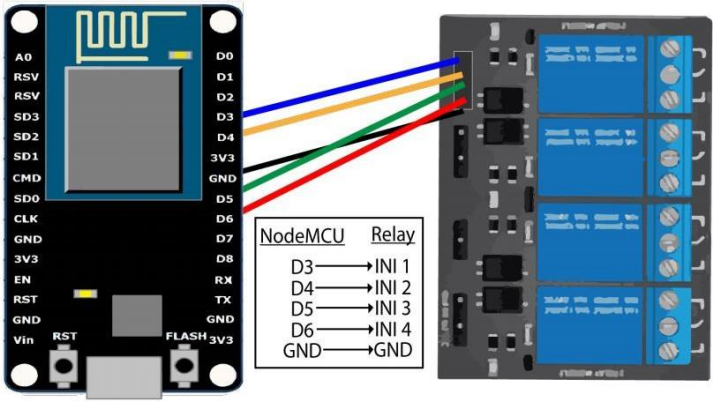


Figure 18. Node MCU & 2 channel relay connection.

3.6 Connecting via wifi code

```
//email: fdnqqxdi@drope.ml
//pass : qwertyui

// use gpio pin2,3,4,5
#define BLYNK_PRINT Serial
/* Fill-in your Template ID (only if using Blynk.Cloud) */
#define BLYNK_TEMPLATE_ID "TMPL3zsAA3-hk"
#define BLYNK_TEMPLATE_NAME "Home Automation"
#define BLYNK_AUTH_TOKEN "3PR1n2JzGhW3gXbmrdwOwC5pa1LKaFh_"

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

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

// Your WiFi credentials.
// Set password to "" for open networks.
//char ssid[] = "SRROBOTICS2G";
//char pass[] = "SATISH@0744";
char ssid[] = "vivo Y33s";
char pass[] = "rahul@8318";


void setup()
{
  // Debug console
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}
void loop()
{
  Blynk.run();
```

FLOW CHART

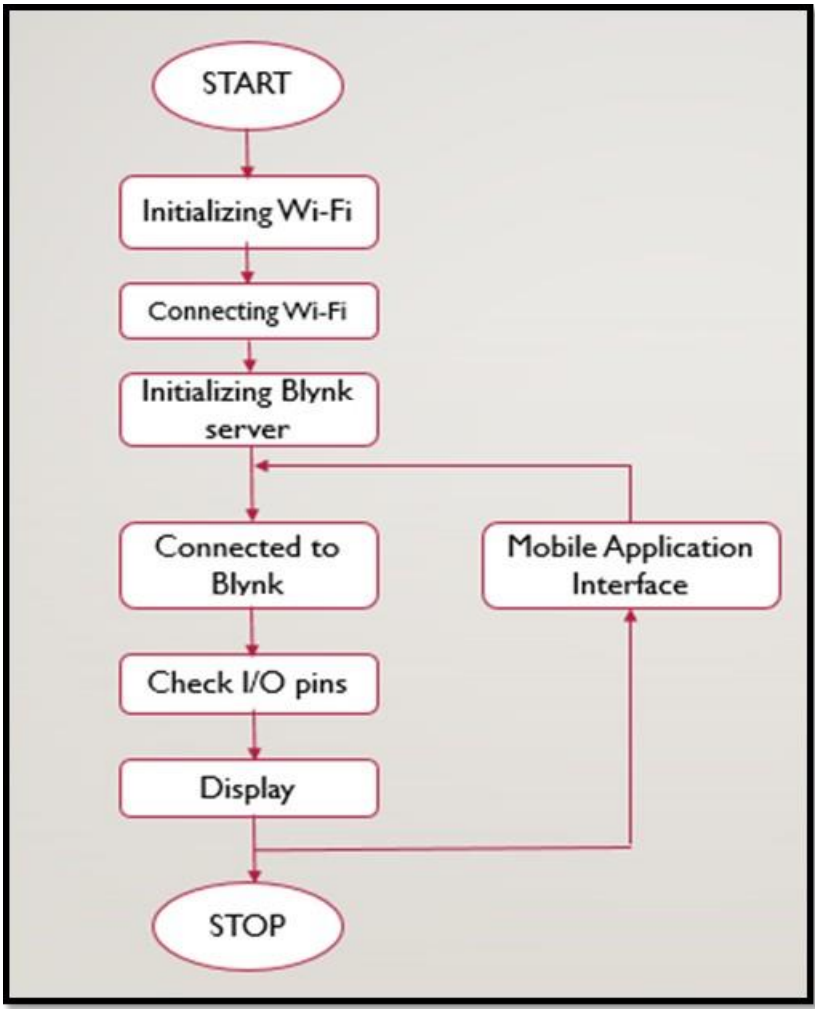


Figure 19. Flow chart of prototype function.

This flow chart shows the working of the project. The process starts be initializing the Wi-Fi, the network name and password are written in the code and uploaded to Node MCU. The android device is connected to Node MCU over Wi-Fi. The Blynk server is set up and connection is made, the devices is identified in the Blynk server using the generated authentication token. The command for controlling the load is given to the application, and this command, over Wi-Fi network is sent to the Node MCU.

HARDWARE MODLE

NODE MCU

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as luacjson and SPIFFS.

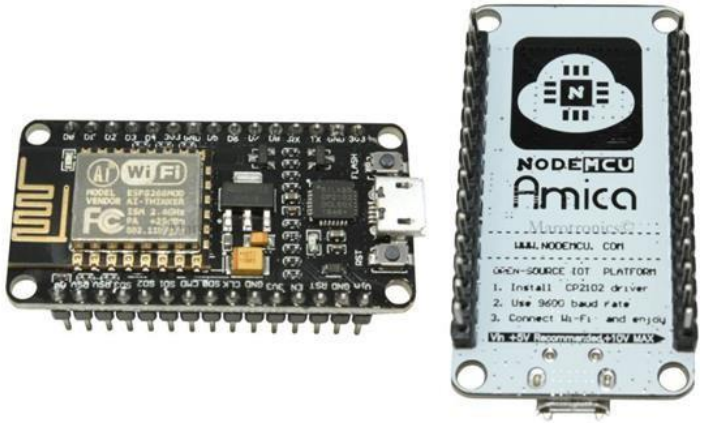


Figure 20. Node MCU module.

RESISTOR

Figure 21. Resistor.

Resistance is the opposition of a material to the current. It is measured in Ohms Ω. All conductors represent a certain amount of resistance, since no conductor is 100% efficient. To control the electron flow (current) in a predictable manner, we use resistors. Electronic circuits use calibrated lumped resistance to control the flow of current. Broadly speaking, resistor can be divided into two groups viz. fixed & adjustable (variable) resistors. In fixed resistors, the value is fixed & cannot be varied. In variable resistors, the resistance value can be varied by an adjuster knob.

RELAY

2 CHANNEL 5V RELAY MODULE

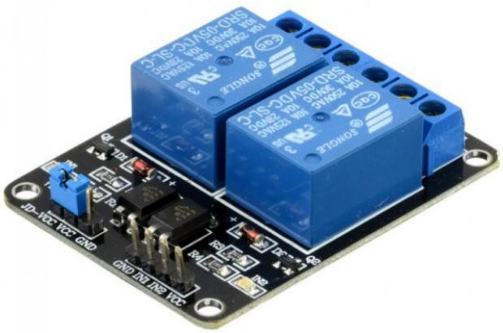


Figure 22. 2 Channel 5V Relay Module

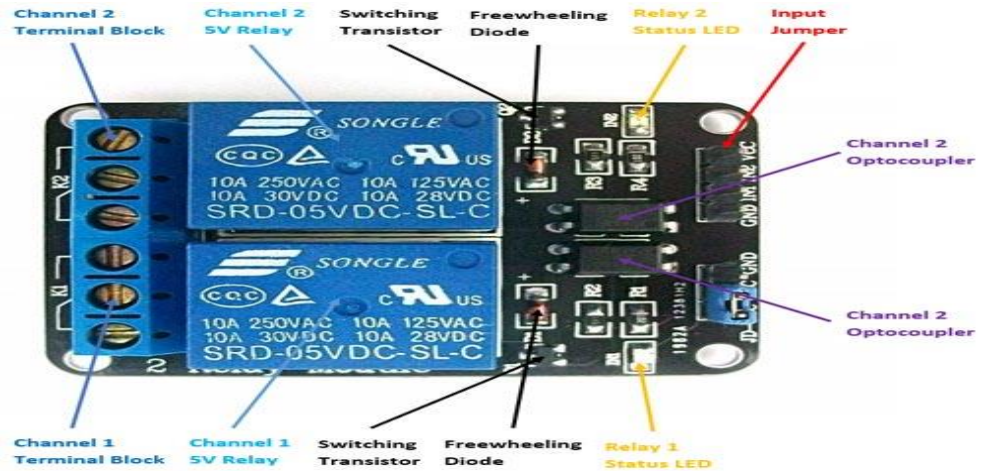


Figure 23. Relay Module

The dual-channel relay module contains switching relays and the associated drive circuitry to make it easy to integrate relays into a project powered by a microcontroller. On the left are two terminal blocks, which are used to connect mains wires to the module without soldering.

Next, come to the two relays. As marked on the body of the relay, the relay coil is rated for 5VDC, and the contacts are rated for 10A at 250VAC or 30VDC, or 125VAC or 28VDC.

The switching transistors amplify the signal from the inputs enough to drive the relay. The freewheeling diodes prevent voltage spikes across the switching transistors.

The status LEDs turn on when the relay is active and indicate switching. The optocouplers are used to provide additional isolation between the input and the relays. The isolation can be selected using the VCC/JDVCC jumper. The input jumper has two input and two power pins and can be easily used to connect to jumper wires and other microcontrollers.

BLANK PCB

A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer (outer and inner layers)..

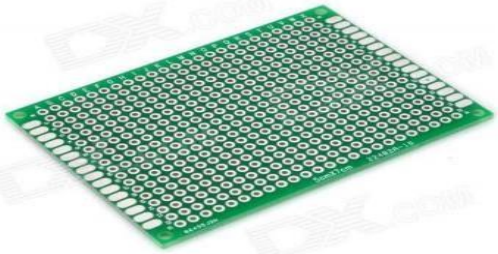


Fig 24. Blank PCB

TRANSFORMER

A transformer is a passive component that transfers electrical energy from one electrical circuit to another circuit, or multiple circuits.

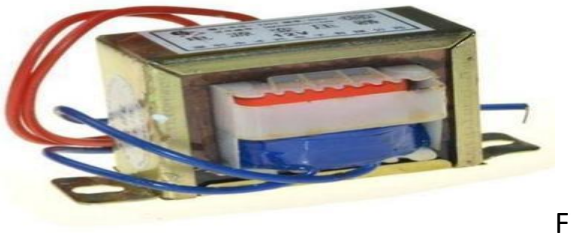


Fig.25 Transformer

CAPACITOR 100MF:

It is used to store energy.

Fig.26 Capacitor

RESULT ANALYSIS

The experimental model was made according to the circuit diagram and the results were as expected. The home appliances could be remotely switched over Wi-Fi network. Both the switch mode and the voice mode control methodologies were successfully achieved. The Blynk application was also successful in displaying the status of every application.

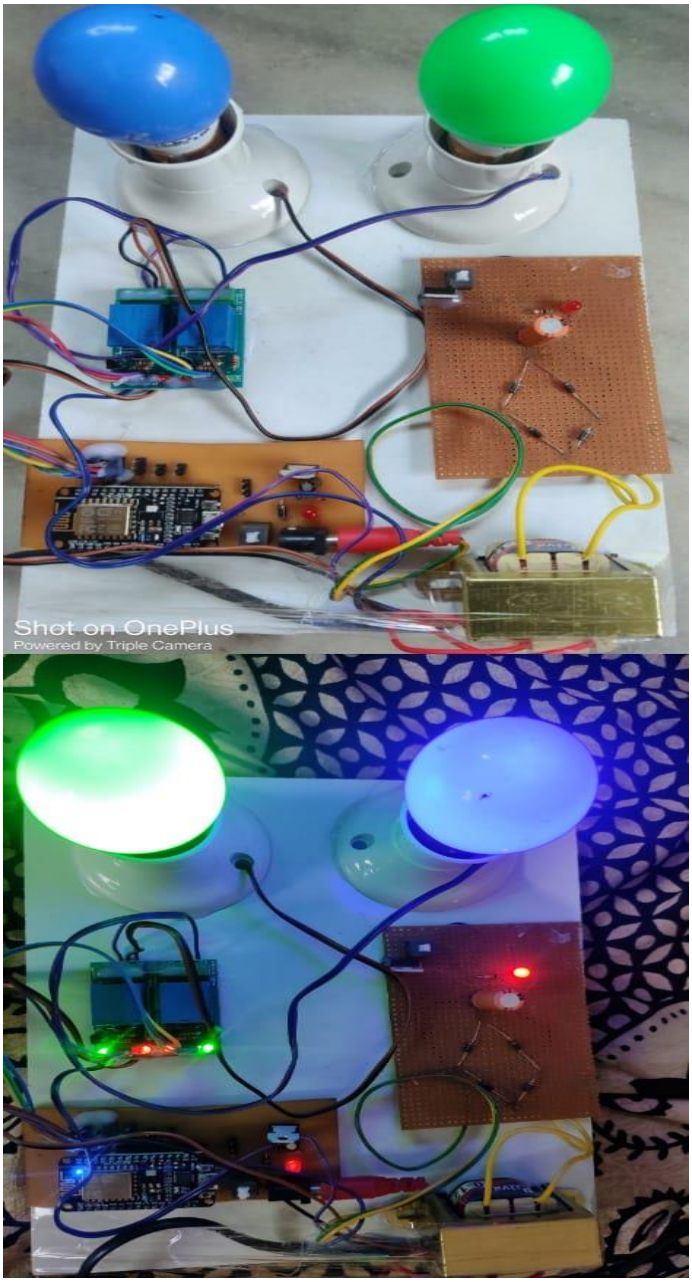


Fig.27 Result

CONCLUSION

It is evident from this project work that an individual control home automation system can be cheaply made from low-cost locally available components and can be used to control multifarious home appliances ranging from the security lamps, the television to the air conditioning system and even the entire house lighting system. And better still, the components required are so small and few that they can be packaged into a small inconspicuous container. The designed home automation system was tested a number of times and certified to control different home appliances used in the lighting system, air conditioning system, home entertainment system and many more . Hence, this system is scalable and flexible. Android devices having lower API version than 16 requires internet access to convert the speech datato string data. Currently, the application is made for Android Smart Phones; other OS platform doesn't support our application. During voice mode, external noises (voice) may affect our result. Thespeech instruction that we command in our voice mode may not give exact result as expected. Therehence lies an ambiguity in result.

REFERENCE

1. *"Smart Energy Efficient Home Automation System using IOT", by* Satyendra K. Vishwakarma, Prashant Upadhyaya, Babita Kumari, Arun Kumar Mishra.

2. *"IOT Based Smart Security and Home Automation", by* Shardha Somani, Parikshit Solunke, Shaunak Oke, Parth Medhi, Prof. P. P. Laturkar.

3. *"A Dynamic Distributed Energy Management Algorithm of Home Sensor Network for Home Automation System", by* Tui-Yi Yang, Chu-Sing Yang, Tien-Wen Sung; in 2016 Third International Conference on Computing Measurement Control and Sensor Network.

4. *"Enhance Smart Home Automation System based on Internet of Things", by* Tushar Churasia and Prashant Kumar Jain; in Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2019) IEEE Xplore Part Number:CFP19OSV-ART; ISBN:978-1-7281-4365-1

5. *"Visual Machine Intelligence for Home Automation", by* Suraj, Ish Kool, Dharmendra Kumar, Shovan Barman.

6. *"A Low Cost Home Automation System Using Wi-Fi based Wireless Sensor Network Incorporating internet of Things", by* Vikram.N, Harish.K.S, Nihaal.M.S, Raksha Umesh, Shetty Aashik Ashok Kumar; in 2017 IEEE 7th International Advance Computing Conference.

7. *"Voice Controlled Home Automation System using Natural Language Processing and Internet of Things", by* Mrs. Paul Jasmin Rani, Jason Bakthakumar, Praveen Kumaar.B, Praveen Kumaar.U, Santhosh Kumar; in 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM)

8. Wikipedia(2009). HomeAutomation. From https://en.wikipedia.org/wiki/Home_automation

9. Theory of IOT from :https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

10. About Node MCU from: https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/