IOT BASED HOME AUTOMATION SYSTEM

A MINI PROJECT REPORT

Submitted by

SAUMY BHARDWAJ [RA2211003010248]

SHOVIK BANERJEE [RA2211003010270]

NITIN ANAND [RA2211003010285]

Under the Guidance of

Dr. C. MURALIDHARAN

(Assistant Professor, Department of Computing Technologies)

in partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be **University** *u/s 3 of UGC Act, 1956*

DEPARTMENT OF COMPUTING TECHNOLOGIES,

COLLEGE OF ENGINEERING & TECHNOLOGY,

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY,

KATTANKULATHUR – 603203

APRIL 2024

Department of Computing Technologies
SRM Institute of Science and Technology
Own Work Declaration Form

Degree/Course : B. Tech in Computer Science and Engineering

Student Names : Saumy Bhardwaj, Shovik Banerjee, Nitin Anand

Registration Numbers : RA2211003010248, RA2211003010270, RA2211003010285

Title of Work : IOT BASED HOME AUTOMATION

I/We here by certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data, etc. that are not my own.
- Not made any use of the report(s) or essay(s) of any other student(s)either past or present
- Acknowledged in appropriate places any help that I have received from others. (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course hand book / University website.

  I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except wherever indicated by referring, and that I have followed the good academic practices noted above. **SAUMY BHARDWAJ [RA2211003010248]** **SHOVIK BANERJEE [RA2211003010270]** **NITIN ANAND [RA2211003010285]** **Date:25 April 2024**
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR–603 203

BONAFIDE CERTIFICATE

Certified that 21CSE253T Mini Project Report titled "**IOT BASED HOME AUTOMATION**" is the Bonafide work of **SAUMY BHARDWAJ [RA2211003010248], SHOVIK BANERJEE [RA2211003010270] , NITIN ANAND [RA2211003010285]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation based on which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. C. MURALIDHARAN
SUPERVISOR
Assistant Professor
Department of Computing Technologies

Dr. M. PUSHPALATHA
HEAD OF THE DEPARTMENT
Assistant Professor
Department of Computing Technologies

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, Dr. **T. V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha,** Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

Our inexpressible respect and thanks to our Guide, **Dr. C. Muralidharan**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under his mentorship. He provided us with the freedom and support to explore the research topics of our interest. His passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project.

Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

SAUMY BHARDWAJ [RA2211003010248]

SHOVIK BANERJEE [RA2211003010270]

NITIN ANAND [RA2211003010285]

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 integrity within 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 different transmission mode that will be implemented to communicate the control of the devices by the user through Node MCU to the actual appliance. The main control system implements wireless technology to provide remote access from smart phone. We are using a cloud server-based communication that would 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 be available, 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..

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

BLE Bluetooth Low Energy

DIP Dual In-line Package

GPIO General Purpose Input/Output

GSM Global System for Mobile

HAS Home Automation System

HMI Human Machine Interaction

IIOT Industrial Internet of Things

IOT Internet of Things

LAN Local Area Network

LDR Light Dependent Resistor

LoRaWAN Low Power Wide Area Network

MQTT Message Queue Telemetry Transport

NFC Near Field Communication

NLP Natural Language Processing

NodeMCU Node Micro Controller Unit

PCB Printed Circuit Boardt

PIR Passive Infrared Sensor

RF Comm Radio Frequency Communication

SoC System on a Chip

SSH Secure Socket Shell

SSL Secure Socket Layer

TCP Transmission Control Protocol

UBW Ultra-Wide Band

UID Unique Identifier

USB Universal Serial Bus

Wi-Fi Wireless Fidelity

WSN Wireless Sensor Network

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

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.

1.2 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.

1.3 PROJECT OBJECTIVES

Design of an independent HAS

To formulate the design of an interconnected network of home appliance to be integrated into the HAS. The objective to account for every appliance and its control to be automated and integrated into the network further formulated into the HAS

Wireless control of home appliances (Switch and Voice mode)

To develop the application that would include features of switch and/or voice modes to control the applications.

Monitoring status of appliances

Being able to view the status of home appliances on the application, in order have a better HAS.

Secure connection channels between application and Node MCU

Use of secure protocols over Wi-Fi so that other devices are prevented to achieve control over the HAS. Secure connections are obtained by SSL over TCP, SSH.

Controlled by any device capable of Wi-Fi (Android, iOS, PC)

To achieve flexibility in control of the home appliances, and device capable of Wi-Fi connectivity will be able to obtain a secure control on the HAS.

Extensible platform for future enhancement

With a strong existing possibility of adding and integrating more features and appliances to the system, the designed system needs to be highly extensible in nature.

1.4 SCOPE

The aim is to design a prototype that establishes wireless remote control over a network of home appliances. The application is designed to run on android device providing features like, switch mode control, voice command control and a provision to view the status of the devices on the application itself. Considering its wide range of application, following are the scope of this prototype.

The system can be implemented in homes, small offices and malls as well, being in-charge of control of the electrical appliances.

For remote access of appliances in internet or intranet. The appliances in the above mentioned environment can be controlled in intra-network or can be accessed via internet.

The development of technology friendly environment. The system incorporates the use of technology and making HAS. By the use of day to day gadgets we can utilize them for a different perspective.

1.5 OVERVIEW AND BENEFITS

The benefits of an established wireless remote switching system of home appliances include:

No legal issues

Obtaining access to or traversing properties with hard lines is extremely difficult.

Reduced wiring issues

Considering the increase in price of copper, thus increases the possibility of the wire to be stolen. The use of a wireless remote system to control home appliances means no wire for thieves to steal.

Extended range

As the system establishes control over Wi-Fi, it was a generally considered descent range. That is 150 feet indoors. Outdoors it can be extended to 300 feet, but since the application is of a HAS, an indoor range is considered.

Security

As the connection of the control of the HAS is established over a secure network the system ensures security to the maximum extent.

Integrable and extensive nature

The prototype designed can be integrated to a larger scale. Also it has an extensive nature being able to add or remove the appliances under control according to application.

.

CHAPTER 2

LITERATURE SURVEY

The integration of Internet of Things (IoT) technology with home automation has heralded a new era of residential living, characterized by interconnected smart environments that offer unprecedented levels of convenience, efficiency, and control. This convergence has been facilitated by a confluence of technological advancements spanning sensor technology, communication protocols, and computing infrastructure. Wi-Fi, Bluetooth, Zigbee, Z-Wave, and Thread stand out as the leading communication protocols, enabling seamless connectivity and interoperability among an ever-expanding ecosystem of smart devices within the home.

Architecturally, both centralized and decentralized approaches have been employed in IoT-based home automation systems, each with its unique advantages and considerations. Centralized architectures, leveraging cloud computing resources, offer scalability and centralized management capabilities, while decentralized architectures, including edge computing and fog computing paradigms, provide real-time processing and decision-making capabilities closer to the source of data generation. These architectural frameworks lay the groundwork for a plethora of applications and use cases that extend across various domains of home automation.

Smart lighting systems, for instance, utilize IoT-enabled light bulbs and switches to adjust lighting levels based on occupancy, ambient light levels, or user preferences, thereby optimizing energy usage and enhancing user comfort. Similarly, energy management systems leverage smart meters, thermostats, and appliances to monitor energy consumption patterns, identify inefficiencies, and automate energy-saving measures. Security systems equipped with IoT sensors, cameras, and actuators offer real-time monitoring, intrusion detection, and remote access control, enhancing home security and peace of mind for homeowners.

The advent of voice assistants and smart speakers has further democratized access to home automation, allowing users to interact with and control smart devices using natural language commands. This intuitive interface, coupled with mobile applications and web-based dashboards, empowers users to customize automation routines, monitor device status, and receive alerts and notifications remotely. Moreover, the integration of artificial intelligence (AI) and machine learning (ML) algorithms enhances the intelligence and adaptability of IoT-based home automation systems, enabling personalized experiences and predictive analytics capabilities.

Despite the transformative potential of IoT-based home automation, several challenges persist, hindering widespread adoption and deployment. Security concerns, including data privacy, device authentication, and vulnerability to cyber attacks, remain paramount, necessitating robust security protocols and best practices to safeguard user data and privacy. Interoperability issues, stemming from the fragmentation of standards and protocols, pose challenges for seamless integration and compatibility among disparate smart devices and platforms. Additionally, the high upfront costs associated with deploying IoT infrastructure and devices may act as a barrier to entry for some consumers, necessitating cost-effective solutions and incentivization mechanisms to spur adoption.

Looking ahead, collaborative efforts among researchers, industry stakeholders, and policymakers are essential to address these challenges and unlock the full potential of IoT-based home automation. Standardization efforts, interoperability initiatives, and regulatory frameworks can foster an ecosystem conducive to innovation, competition, and consumer trust. Moreover, investments in research and development, coupled with public awareness campaigns and educational initiatives, can accelerate the adoption and diffusion of IoT technologies in residential settings, driving societal and environmental benefits.

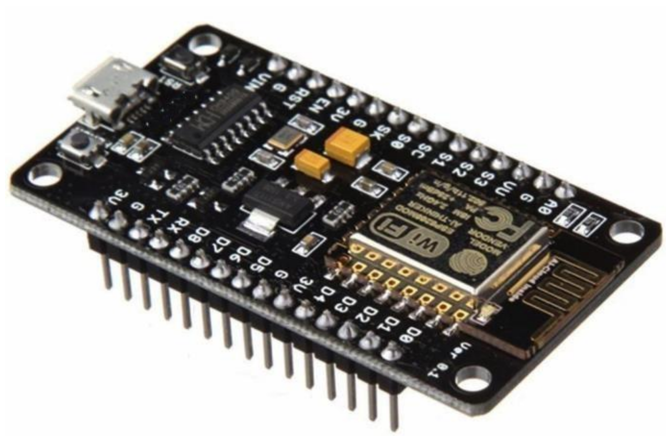
In conclusion, IoT-based home automation represents a transformative force in modern living, offering unparalleled levels of comfort, convenience, and efficiency. By harnessing the power of interconnected devices, data analytics, and intelligent algorithms, smart homes have the potential to enhance the quality of life for individuals and communities worldwide. As we navigate the complexities and opportunities of this burgeoning field, collaboration, innovation, and responsible stewardship will be key to realizing the promise of a connected and automated future.

CHAPTER 3

THEORY

3.1 NODE MCU

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.



3.1 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.

3.1.1 Pin 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.

Table 3.1. Node MCU index ↔ GPIO mapping

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		

[*] 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

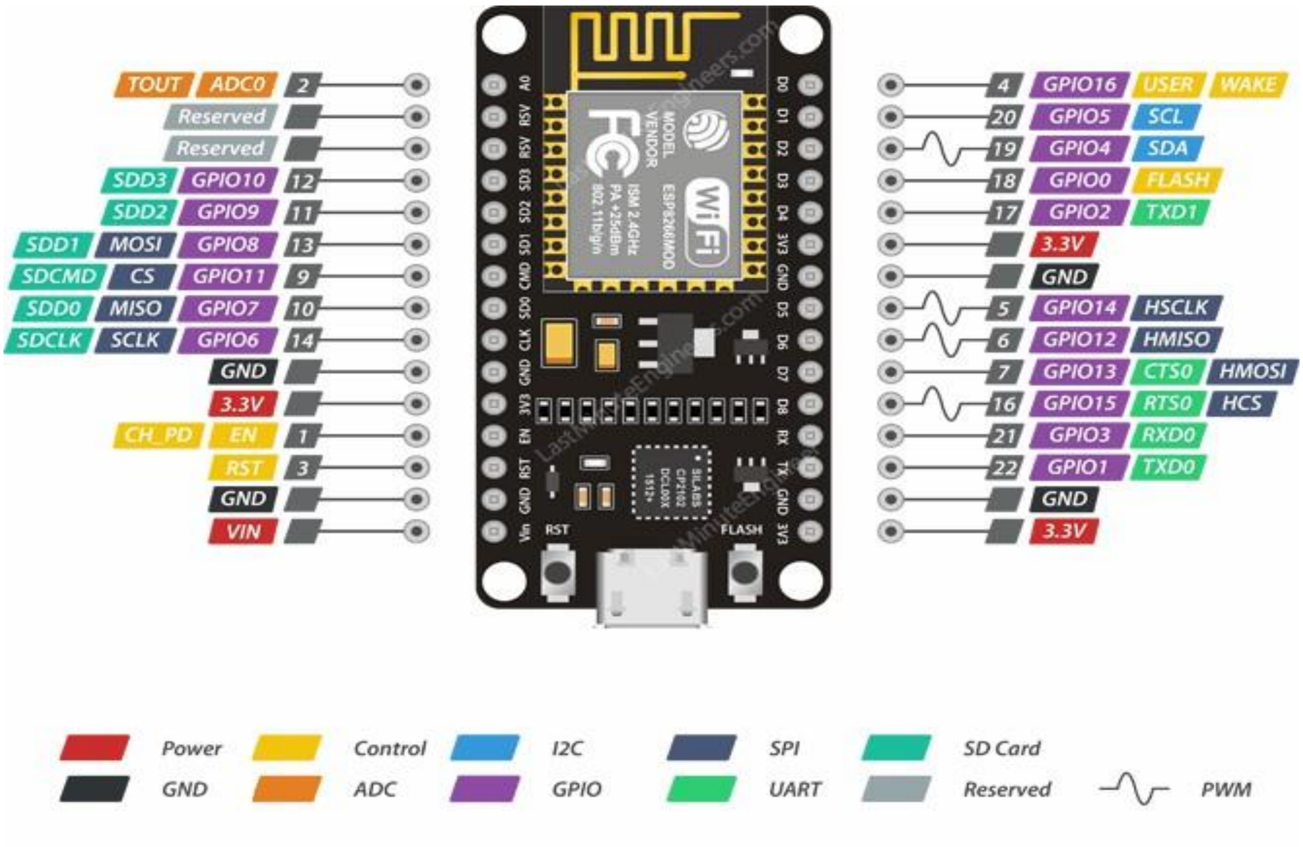
SDIO Pins: ESP8266 features Secure Digital Input/output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

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 3.1.1 ESP8266 Node MCU pinout

3.1.2 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.

3.2 BLOCK DIAGRAM

.

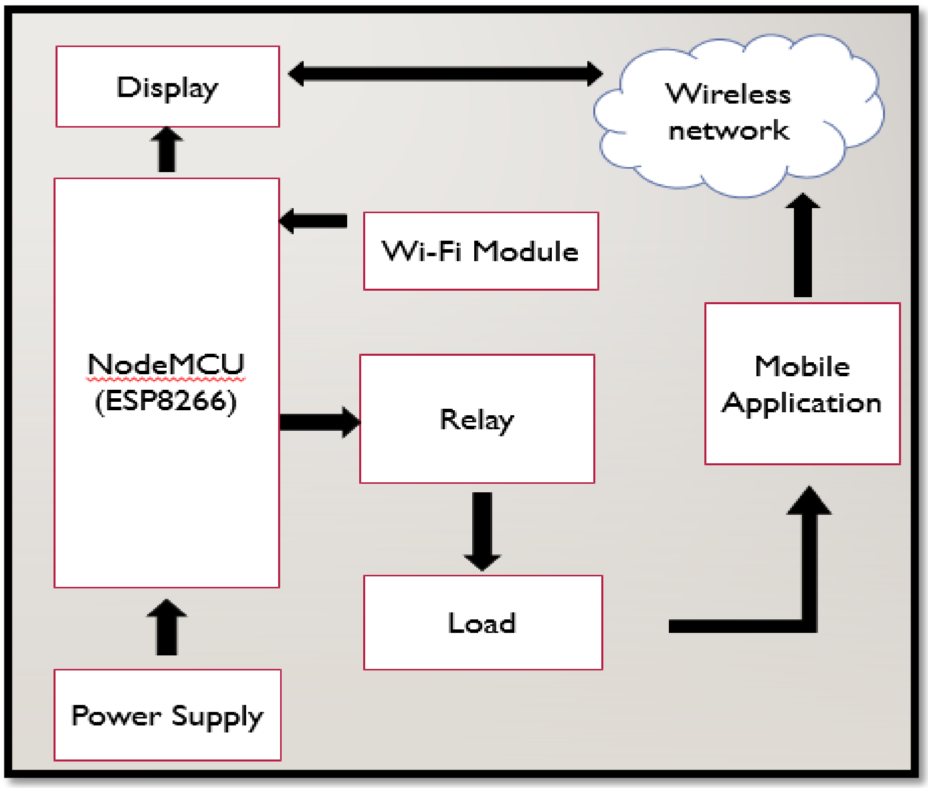


Fig 3.2 Block diagram of proposed system

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.

3.3 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.

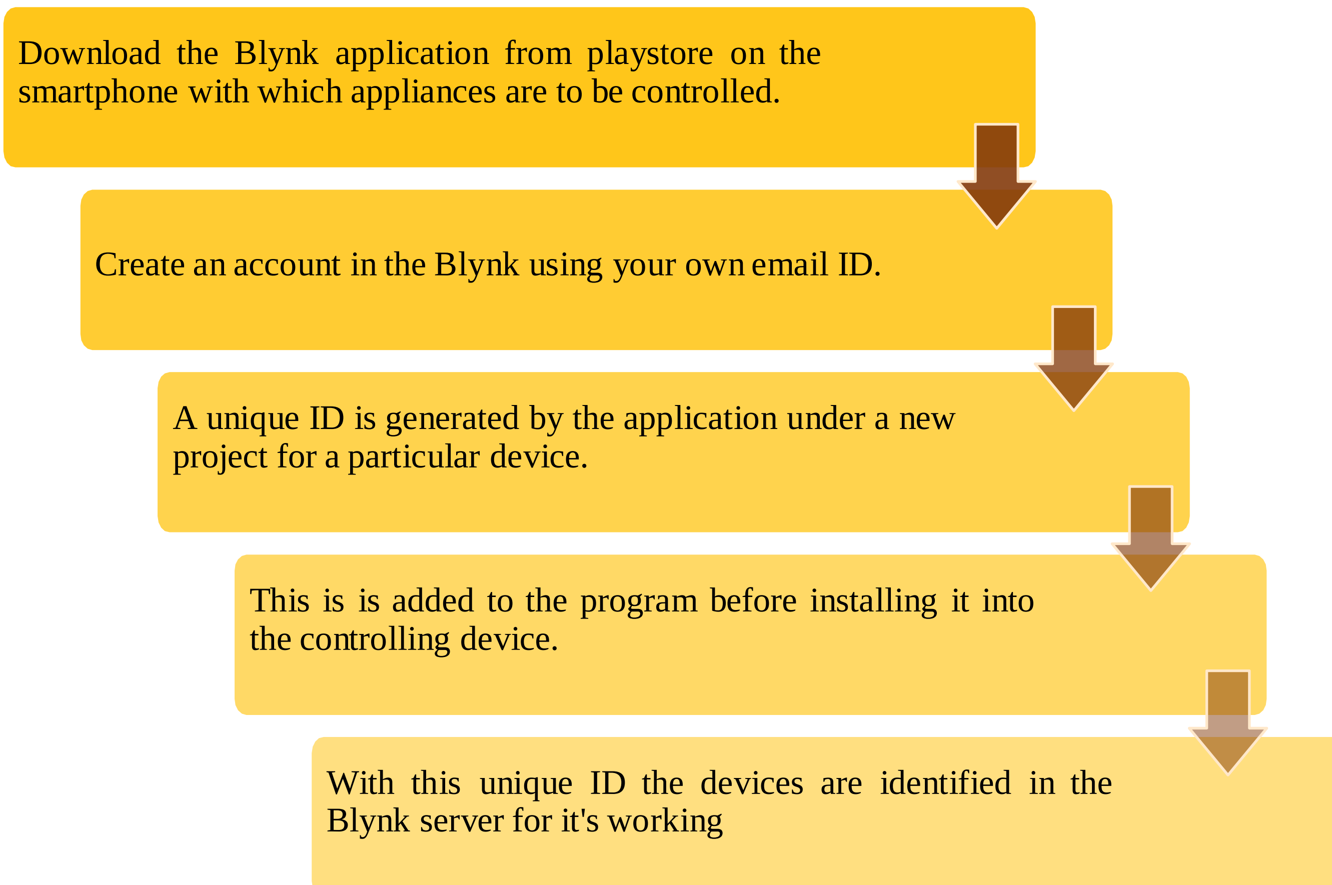


Fig 3.3.1 Creating an account and generating unique ID in Blynk Server

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.

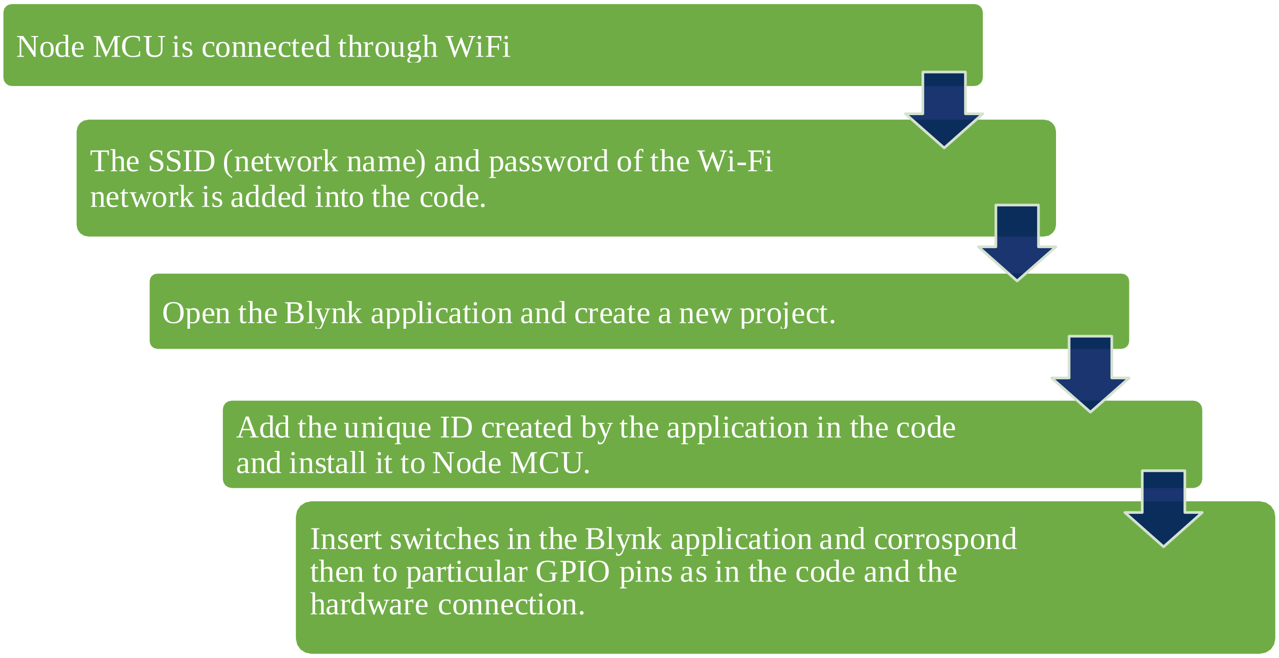


Fig 3.3.2 Setup to control Node MCU from Blynk application

3.4 CIRCUIT DIAGRAM

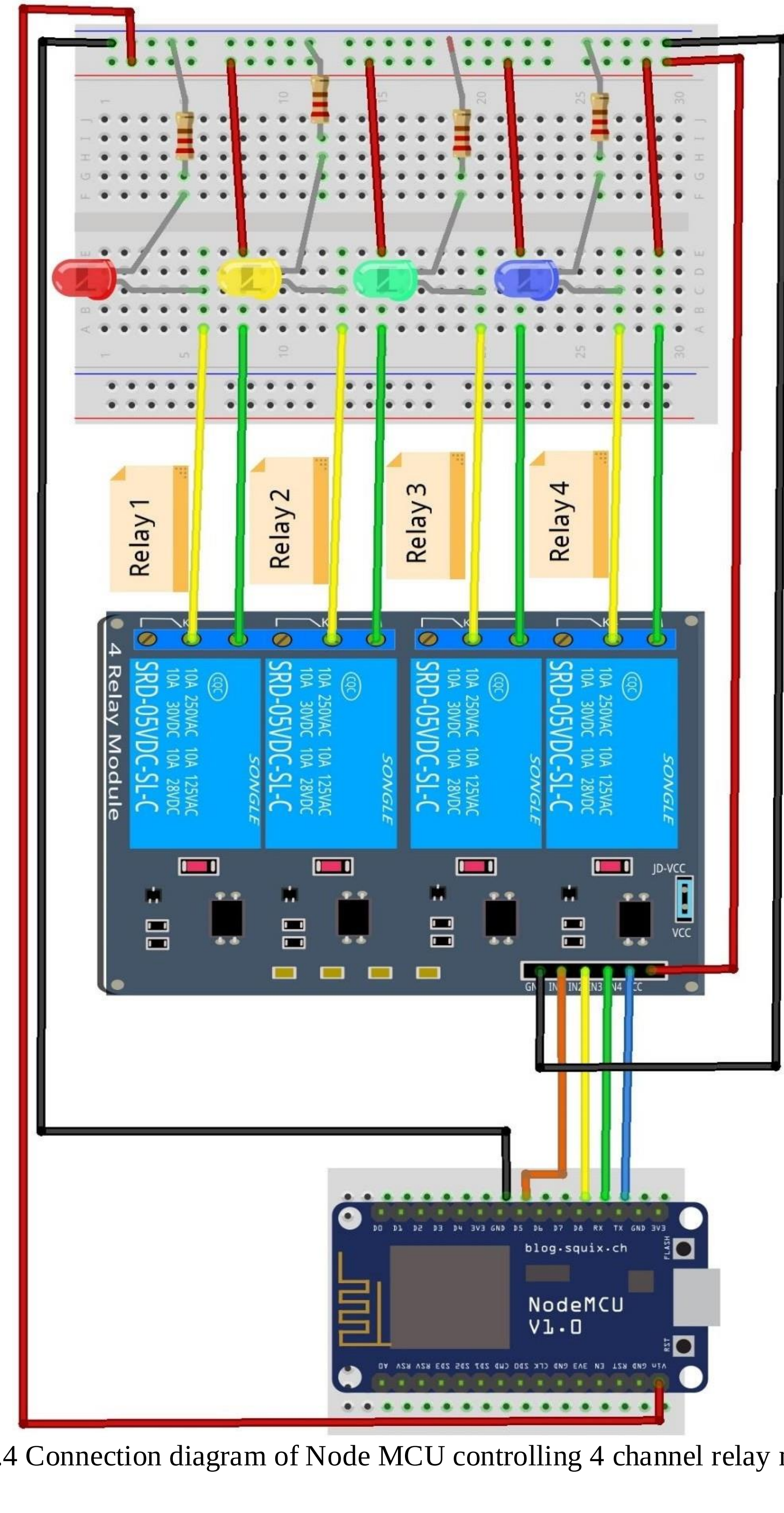


Fig 3.4 Connection diagram of Node MCU controlling 4 channel relay module

CHAPTER 4
METHODOLOGY

4.1 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.

.

4.2 PROJECT LAYOUT

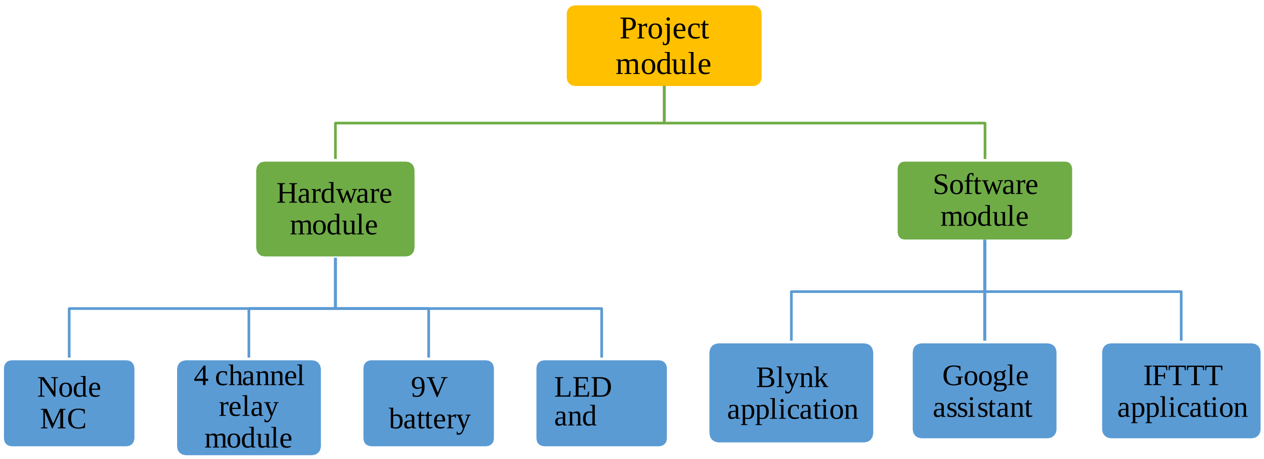


Fig 4.2 Layout of project module

Node MCU is the microcontroller unit in the prototype. It has an in built Wi-Fi module (ESP8266) that establishes 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.

4.3 COMPONENTS REQUIRED

Table 4.3 Component listing

SL. NO	Component	Quantity
1.	Node MCU	1
2.	4 channel relay board	1
3.	9V battery	1
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

4.4 SETTING UP THE SYSTEM

4.4.1 Downloading 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 4 channel relay module, 4 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.

Fig 4.4.1Set up Blynk application

4.4.2 Driver installation for hardware interfacing

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.

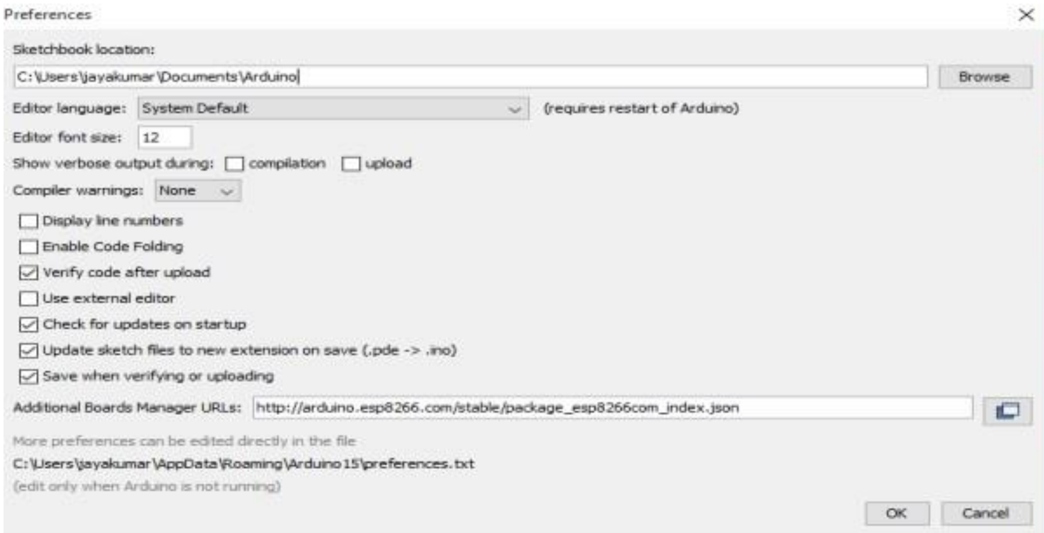
- Once user downloads drivers as per its respective operating system, the system establishes connection to Node MCU.

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 .

4.4.3 Interfacing Node MCU with Arduino IDE

To begin with the latest Arduino IDE version, we'll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window: *http://arduino.esp8266.com/stable/package_esp8266com_index.json*

OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.



Arduino IDE preferences.

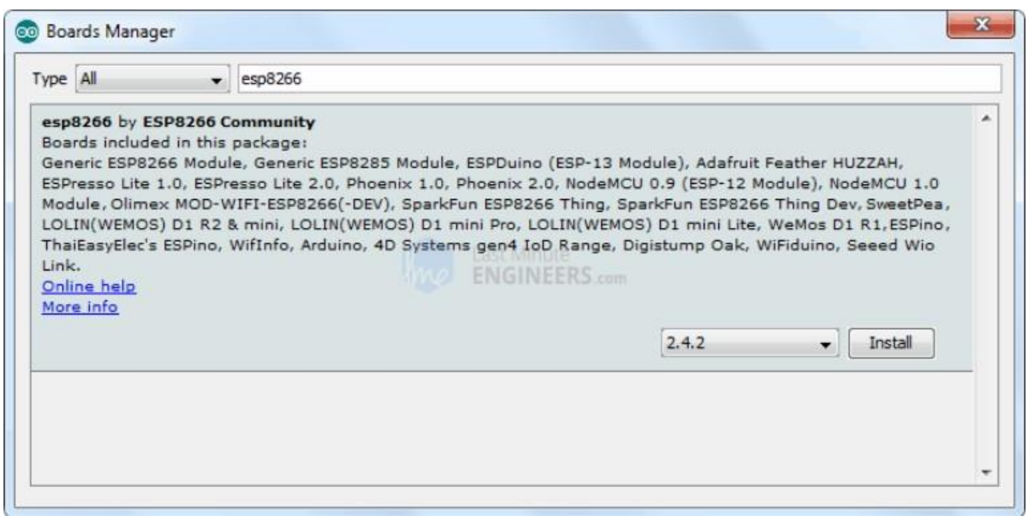


Fig 4.4.3.1 ESP8266 board installation in Arduino IDE

Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select Node MCU 0.9 (ESP-12 Module) option under your Arduino IDE > Tools > Board menu.

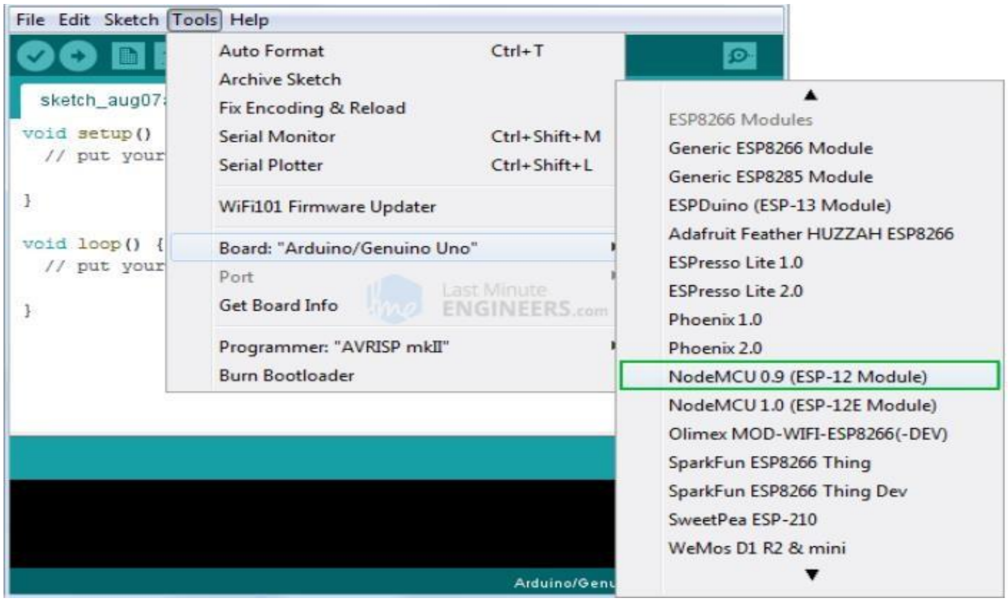


Fig 4.4.3.2 Arduino IDE board manager installation.

Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed: 115200

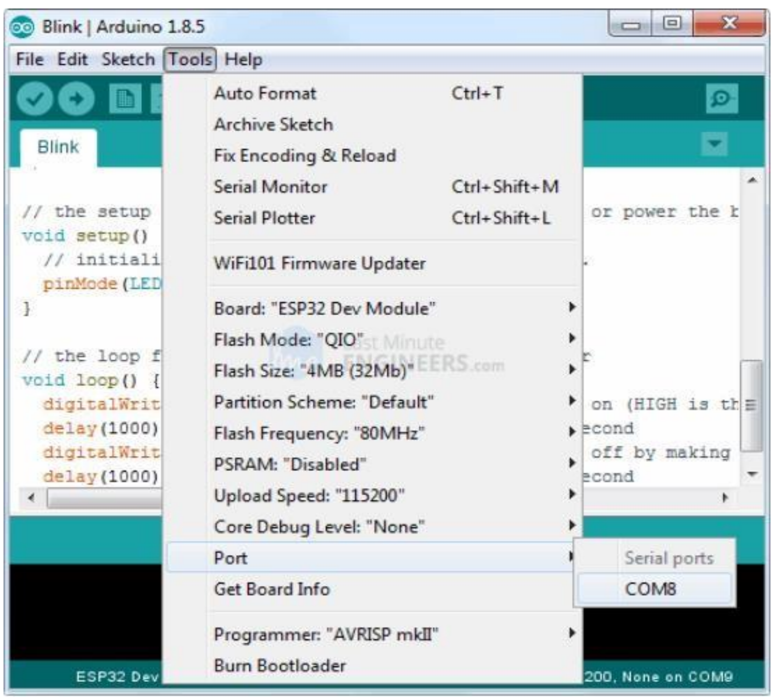


Fig 4.4.3.3 Assigning communication port on Arduino IDE.

4.4.4 Uploading code to Node MCU

- NodeMCU is connected to PC using a USB cable.
- Now, we'll set up the Arduino IDE by changing some settings. So, open up the Arduino IDE. Select Tools > Board and select 'NodeMCU 1.0 (ESP-12E Module)' as the board. And that's allthe settings we need to change. So now we begin writing the code.
- Select Files > Examples > Blynk > Boards_WIFI > ESP8266_Standalone. A new file with someprewritten code opens. The following changes to the code are made.

  1. The line which says 'char auth[] = "YourAuthToken", replace YourAuthToken part with your Blynk's authentication token that was generated by the Blynk server

  2. The line which says char ssid[] = "YourNetworkName", replace YourNetworkName part with the name of Wi-Fi network that the Node MCU must connect to.

  3. The line where it says char pass[] = "YourPassword" and replace the YourPassword part with the password of the Wi-Fi network.

- The code is ready to be uploaded to the hardware. On clicking upload button, the code is uploaded to Node MCU and the next time it's powered on, it automatically connects to the assigned Wi-Fi network.

4.4.5 Installation and setup of IFTTT

- To configure IFTTT we visit their website https://ifttt.com and sign up using google account.
- After signing in, we select My Applets from header, and select New. Search for Google assistant and connect. Allow IFTTT for permission to use Google account to add voice commands to it.
- Configure the application to work as desired, and Create Trigger.

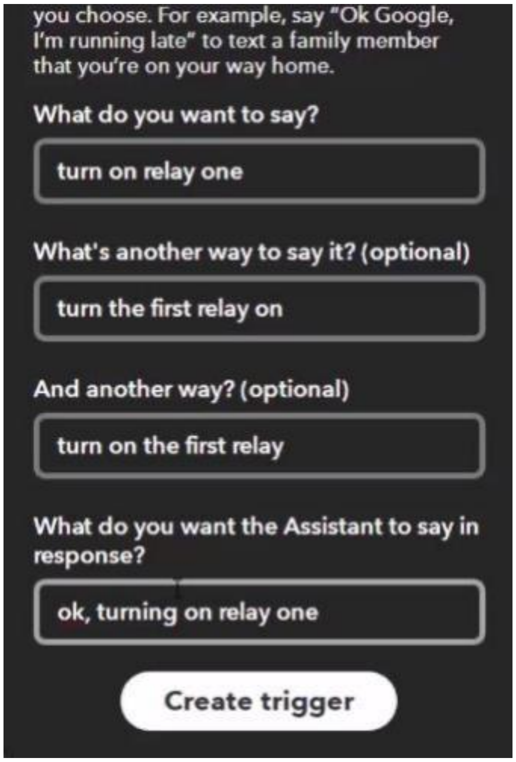


Fig 4.4.5.1 IFTTT configured with actions and commands.

- Select webhooks that will allow to send commands to Blynk server. Add *http://188.166.206.43/ YourAuthTokenHere / update / DigitalPinToBeUpdateHere* to the URL field.

  YourAuthTokenHere is replaced by the authentication token generated by Blynk server. DigitalPinToBoUpdatedHere is replaced by the digital pin of Arduino that corresponds to the Node MCU rather than the one of Node MCU itself.

  Following details are added to program the applet. Here '0' means to turn on, so we are basically saying Blynk to turn on relay that is connected to pin D3, which in our case is relay one.

  Click on Create Action and finish.

Fig 4.4.5.2 Configuration of applet to switch relay with voice commands.

- Similarly, another applet is created to turn off the relay, repeating all the steps above except the following changes: instead of writing "Turn on relay one", written "Turn off relay one"and instead of ["0"], written ["1"]. Two triggers are created to turn on and off one Relay.

- Similarly, we create triggers for remaining 3 relays by change the phrase and Digital pin for each Relay. All the other steps will remain the same.

In the end for 4 relays, we have 8 triggers to turn each of them on or off. After all this is done, voice commands to Google Assistant can switch relay.

4.5 HARDWARE ASSEMBLY

Hardware assembly mainly includes connecting specific digital pins of NodeMCU to the 4 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.cientific research, especially in the computational domain, heavily relies on the strategic incorporation of specialized libraries to enhance efficiency and precision. In the domain of chest radiography analysis using deep learning methodologies, several Python-based libraries emerge as essential to achieve the desired outcomes.

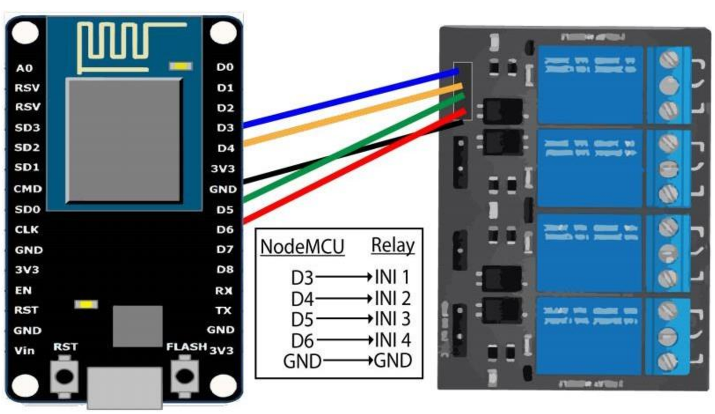


Fig 4.5Node MCU & 4 channel relay connection.

CHAPTER 5

LOGIC AND OPERATION

5.1 FLOW CHART

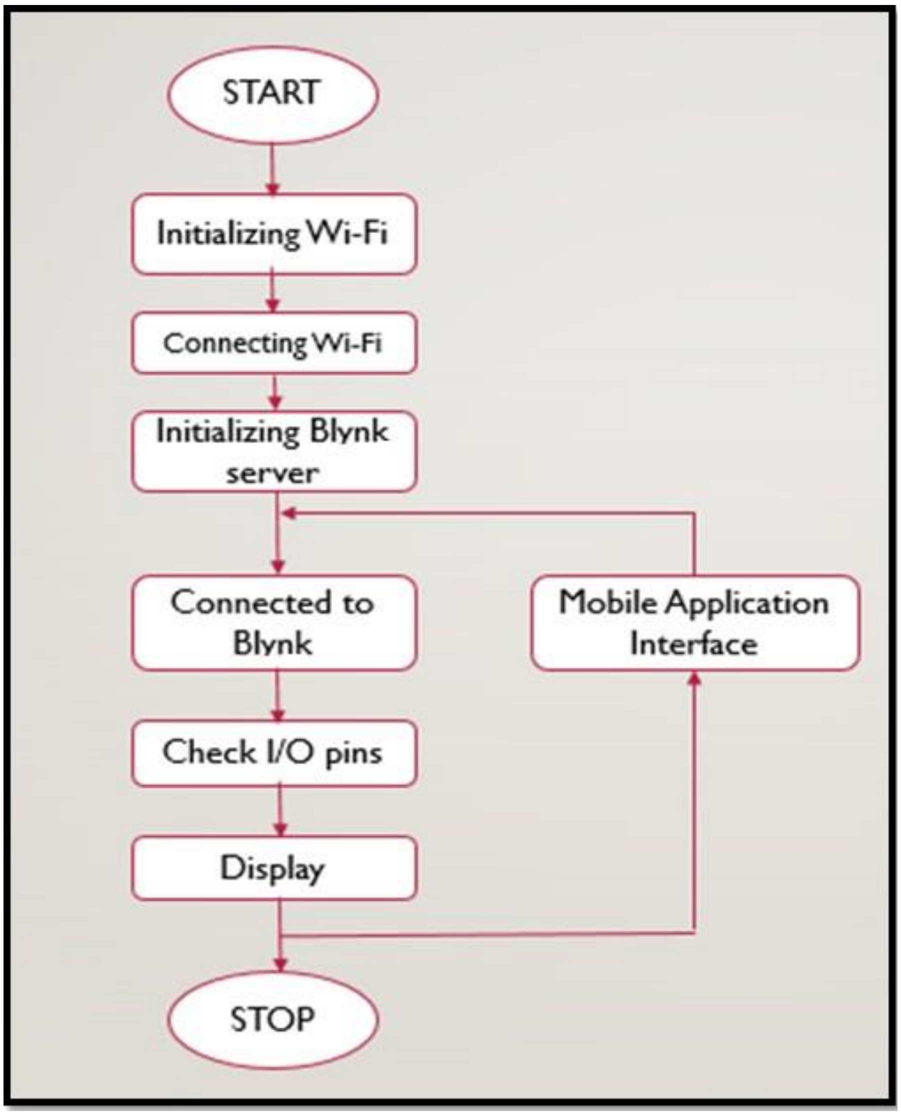


Fig 5.1 Flow chart of prototype function.

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.

5.2 PRINCIPLE AND OPERATION

Node MCU 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 “Node MCU” 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 lua-cjson, and spiffs.

5.2.1 Advantages of Node MCU

- Low cost, the Node MCU is less costly compared to any other IOT based device.
- Node MCU has Arduino Like hardware I/O. It is becoming very popular in these days that Arduino IDE has extended their software to work in the field of ESP 8266 Field module version.
- Node MCU has easily configurable network API.
- Integrated support for Wi-Fi network: ESP 8266 is incorporated in Node MCU, which is an easily accessible Wi-Fi module.
- Reduced size of board.
- Low power consumption.

5.2.2 Disadvantages of Node MCU

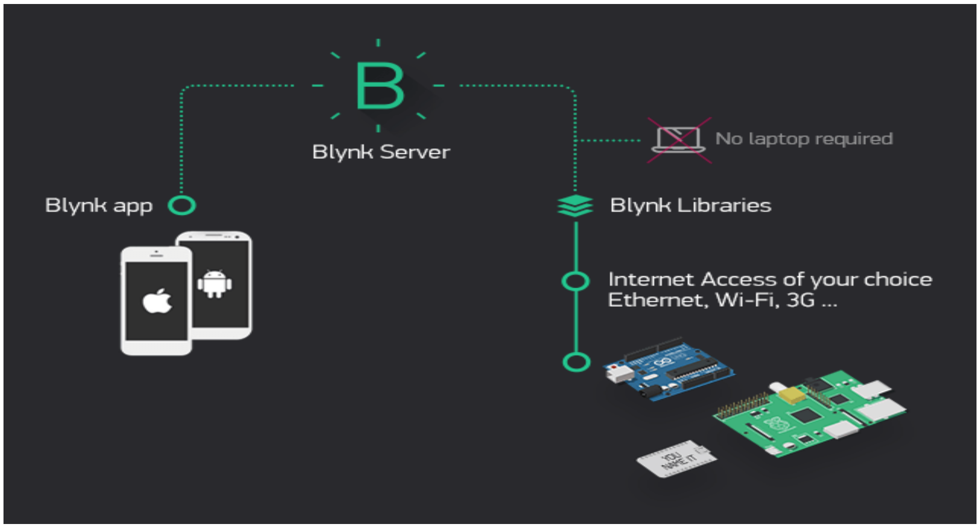
- The operation of the circuit depends on the working internet connection. If the working internet connection is not available then it will not run.
- Node MCU also depends on the free server provided by the third party, if the free server is not working then it will not run.
- Node MCU has less resources of official documentation
- Need to learn a new language and IDE
- Reduced pinout
- Scarce documentation

5.3 BLYNK APPLICATION

The Blynk application was designed for the primary purpose of Internet of Things. 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 graphic interface for a prototype can be built by simply dragging and dropping widgets. It can control hardware remotely, it can display sensor data, can store and visualize data and possessed a lot more functionality. There are three major components in the platform:

- **Blynk Application:** 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 an 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 outgoing commands.

Every time a radio button is accessed in the Blynk application, the message travels to the Blynk Cloud, where it finds the specific hardware by the unique generated authentication token. It works in the same way for the opposite direction



5.4 WIRELESS COMMUNICATION NETWORK

The prototype aims to wireless control over home appliances with the technology of IOT. As discussed earlier, IOT supports various wireless communication protocols, like Bluetooth, Z-Wave, Zigbee etc. this prototype uses Wi-Fi as wireless communication network to establish remote access over home appliances. This is because Wi-Fi has its own advantages over other wireless communication protocols.

Advantages of Wi-Fi over other wireless technologies like Bluetooth and ZigBee

Bluetooth is generally used for point to point networks and Bluetooth operates at a much slower rate of around 720 Kbps which is very small for video transfer or moving large amount of data like the image captured from a camera, whereas the bandwidth of Wi-Fi can be up to 150Mbps and very ideal for video transmission.

Wi-Fi is very much secure means of communication than Bluetooth.

Wi-Fi connection to send video, audio, and telemetry operation, while accepting remote control commands from an operator who can be located virtually anywhere in the world.

Robots are already being eyed for obvious tasks like conducting search-and rescue missions during emergencies or hauling gear for soldiers in the jungle or woods. The mechanics of the robot uses the concept that has been developed to ensure robust navigation, search and transportation in rough terrain.

Table 5.4. Comparison chart of Wi-Fi with other wireless communication protocols

STANDARD	BLUETOOTH	UBW	ZIGBEE	WI-FI
IEEE specification	802.15.1	802.15.3a	802.15.4	802.11a/g/b
Frequency band	2.4 GHz	3.1-10.6 GHz	868/915 MHz; 2.4 GHz	2.4 GHz; 5 GHz
Maximum signal rate	1 Mb/s	110 Mb/s	250 Kb/s	54 Mb/s
Nominal range	10 m	10 m	10-100 m	100 m
Nominal TX power	0-10 dBm	-41.3 dBm/MHz	(-25) -0 dBm	10-20 dBm

RF channels	79	1-15	1/10; 16	14 (2.4 GHz)
Channel bandwidth	1 MHz	500 MHz- 7.5 GHz	0.3/0.6 GHz; 2 MHz	22 MHz
Modulation type	GFSK	BPSK, QPSK	BPSK (+ASK), O- QPSK	BPSK, QPSK, COFDM, CCK, M-QAM
Spreading	FHSS	DS-UBW, MB-OFDM	DSSS	DSSS, CCK, OFDM
Co-existence mechanism	Adaptive frequenc y hopping	Adaptive frequency hopping	Dynamic frequency selection	Dynamic frequency selection, transmit power control
Basic cell	Piconet	Picomet	Star	BSS
Extension of basic cell	Scatteme t	Peer-to-peer	Cluster tree, Mesh	ESS
Maximum cell nodes	8	8	>65000	2007
Encryption	E0 Stream chipper	AES block cipher (CTR, counter mode)	AES block cipher (CTR, counter mode)	RC4 stream cipher (WEP), AES block cipher
Authentication	Shared secret	CBC-MAC (CCM)	CBC-MAC (extention of CCM)	WPA2 (802.11i)
Data protection	16-bit CRC	32-bit CRC	16-bit CRC	32-bit CRC

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

6.1 RESULT

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.

6.2 FURTHER ENHANCEMENT AND FUTURE SCOPE

Looking at the current situation we can build cross platform system that can be deployed on various platforms like iOS, Windows. Limitation to control only several devices can be removed by extending automation of all other home appliances. The prototype can include sensors to implement automatic control of the home appliances like; an LDR that can sense daylight and switch lamp accordingly, a PIR to detect motion and be used for security purposes making an alarm buzz, or a DHT11 sensor that's senses ambient temperature and humidity of atmosphere and switch fan/air conditioner accordingly. Scope of this project can be expanded to many areas by not restricting to only home, but to small offices.

Android devices having lower API version than 16 requires internet access to convert the speech data to 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. The speech instruction that we command in our voice mode may not give exact result as expected. There hence lies an ambiguity in result.

6.3 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..

REFERENCES

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/