HOME AUTOMATION USING NODE MCU

Submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering

Degree in Electronics and Telecommunication Engineering

by

RITESH KUMAR VIRANI (REG NO:37250020)

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
JEEPPIAR NAGAR,RAJIV GANDHI SALAI,CHENNAI – 600119

APRIL - 2021

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the Bonafide work of **RITESH KUMAR VIRANI** (**37250020**) who carried out the project entitled in "**Home Automation Using Node MCU"** under our supervision from November 2020 to April 2021.

Internal Guide

Mr. L MEGALAN LEO M.E., (Ph.D.)

Head of the Department

Dr. V. VIJAYA BASKAR M.E., Ph.D.

Submitted for Viva voce Examination held on 19th April 2021________

Internal Examiner **External Examiner**

DECLARATION

I, **RITESH KUMAR VIRANI** (**Reg.No:37250020**) hereby declare that the Project Report entitled on "**HOME AUTOMATION USING NODE MCU"** done by me under the guidance of guide **Mr. L MEGALAN LEO M.E., (Ph.D.)** Dept of ETCE at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Telecommunication Engineering.

Ritesh

DATE: 19th April 2021

PLACE:CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encourage mention doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. NANDITHA Ph.D.** Dean, School of Electronics and **Dr. V.VIJAYA BASKAR M.E., Ph.D.** Head of the Department, Dept. of Electronics and Telecommunication Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mr. L MEGALAN LEO M.E., (Ph.D.)** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Electronics and Telecommunication Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Internet of Things is composed of things that have unique identities and are connected to each other over internet. It is simply connecting and monitoring various devices and sensors through Internet. This paved the way for home automation and monitoring which makes human life more comfortable and secured. This paper describes the overall notion of the IOT based sensing systems and monitoring systems for implementing an automated home. The proposed prototype uses Node MCU board with internet being remotely controlled by Android/iOS OS smart phone. Node MCU is the heart of this system and it can perform as a micro web server and it acts as an interface for the wide range of hardware modules. To control lights, fans and other home appliances which are connected to the relay system, the system offers switching functionalities.

As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, involving remote controlled switches. Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most modern solution with smart phones. In order to achieve this, a microcontroller board with builtin wifi module at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands to the receiver where loads are connected. By touching the specified location on the GUI, the loads can be turned ON/OFF remotely through this technology.

CONTENTS

CHAPTER NO:	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF ABBREVIATIONS	ix
	LIST OF FIGURES	x
1	**INTRODUCTION**	
	1.1 About Internet Technology	1
	1.2 EARLIER HOME AUTOMATION SYSTEM	2
	1.3 Embedded systems	3
	1.3.1 NODE MCU Board	4
	1.3.1.1 Node MCU ESP8266 SPECIFICATIONS	6
	1.3.1.2 APPLICATIONS	6
	1.3.2 Software used	7
	1.3.2.1 Arduino IDE	7
	1.3.2.2 Programming	8
2	**LITERATURE SURVEY**	9
3	**AIM AND SCOPE**	10
	3.1 Aim	10
	3.2 Scope of the Project	10
	3.3 Objective	11
4	**MATERIALS AND METHODS USED**	12
	4.1 Hardware Components used and Specifications	12
	4.1.1 NODE MCU (ESP8266- 12E)	12

4.1.1.1 Features 13

4.1.1.2 Pinout Configuration 14

4.1.1.3 Brief about Node MCU 15

4.1.2 4- Channel Relay Module 16

4.1.2.1 Working 17

4.1.2.2 Construction 18

4.1.2.3 Connectivity 19

4.1.3 Breadboard 20

4.1.3.1 What is Breadboard 20

4.1.3.2 Common Use 20

4.1.3.3 Typical Specifications 21

4.1.4 Jumper Wires 22

4.1.4.1 Types 22

4.1.4.2 Versions of Solid Tips 23

4.1.5 5 V 1 Amps AC To DC Adapter 24

4.1.5.1 Specifications 24

4.2 Software Requirements 25

4.2.1 Arduino IDE 25

4.2.1.1 Node MCU Code VIA IDE 25

4.3 Existing System 27

4.3.1 What they have followed 27

4.4 Proposed System 28

4.4.1 Block Diagram 29

4.4.2 Circuit Diagram 30

4.4.3 Working Process 31

4.4.3.1 Connections 31

4.4.3.2 Code 32

5 **RESULTS AND DISCUSSION** 35

5.1 Light Test Results 35

5.2 System Analysis 36

5.3 Final Circuit Connections 36

6 **SUMMARY AND CONCLUSION** 38

REFERENCES 39

LIST OF ABBREVIATIONS

ABBREVATIONS		EXPANSION
LCD	-	Liquid Crystal Display
IDE	-	Integrated Devlopment Environment
PIR	-	Passive Infrared Sensor
ADC	-	Analog to Digital Converter
SPI	-	Serial Peripheral Interface
GUI	-	Graphical User Interface
IOT	-	Internet Of Things

LIST OF FIGURES

FIGURE NUMBER	FIGURE NAME	PAGE NUMBER
1.3.1	Node MCU Development Board	4
1.3.2	Node MCU ESP8266 Pinout	5
1.3.2.1	Arduino IDE Setup	7
4.1.1	Node MCU	13
4.1.1.3	Briefing Node MCU	15
4.1.2	Relay Module	16
4.1.2.1	Working of Relay	17
4.1.2.3	Connectivity of Relay	19
4.1.3.3	Bus Strips	21
4.1.3.4	Jumper Wires	21
4.1.4.2	Types of Jumper Wires	23
4.1.5	Adapter	24
4.2.1.1	Setting up Node MCU in Arduino IDE	26
4.4.1	Block diagram for proposed system	29
4.4.2	Circuit diagram for proposed system	30
4.4.3	Circuit diagram for Relay	30
4.4.3.2	Code 1	32
4.4.3.3	Code 2	33
4.4.3.4	Serial Monitor	34
5.1.1	Web App Output 1	35
5.1.2	Web App Output 2	35
5.3.1	Output Hardware Connection 1	36
5.3.2	Output Hardware Connection 2	37

CHAPTER 1
INTRODUCTION

1.1 About Internet Of Things

Internet of Things depicts a general concept for the power of network devices to sense and collect data from the planet around us, then share that data across the Internet where it can be processed and applicable for various interesting purposes. Internet of Things is extremely quickly becoming a reality. We can see the results of it around us. Our devices are becoming smarter every day from smart phones to smart TV to smart car to Smart kitchen. Everything is now getting connected to Internet. Internet of Things (IoT) describes a network of physical objects that hook up with one another through the web. Objects or 'things' can transfer data wirelessly without requiring human interaction. A 'thing' are often any object which will be assigned an IP address and given the power to transfer data over a network.

A Thing, within the Internet of Things (IOT), are often an individual with a cardiac monitor implant, a livestock with a biochip transponder, an automobile that has built-in sensors to alert the driving force when tire pressure is low -- or the other natural or man-made object which will be assigned an IP address and given the power to transfer data over a network. These devices collect useful data with the assistance of varied existing technologies then autonomously flow the information between other devices**.** Current market examples which includes smart thermostat systems and washer/dryers that employ Wi-Fi for remote monitoring.

Internet of Things (IoT) is an architecture that comprises of especial hardware boards, Software systems, web APIs, and protocols which together creates a seamless environment that allows smart embedded devices which is to be connected to the web such that sensory data can be accessed and control system can be activated over internet. Also such devices could be connected to internet using various means like WiFi, Ethernet and so on. Furthermore devices might not needed to be connected to internet independently.

1.2 EARLIER HOME AUTOMATION SYSTEM

A home automation system generally connects controlled devices to a main server or "interface". The interface for control of the system uses either wall-mounted terminals, tablet or desktop computers, a mobile application, or an internet interface which will even be accessible off-site through the Internet. While there are many competing sellers, there are gradually increasing efforts towards open source systems (OSS). However, there are issues with the present state of home automation including a lack of authorized security measures and deprecation of older devices without backwards compatibility.

Home automation has high potential for sharing data between relations or trusted individuals for private security and will cause energy saving measures with a positive environmental impact in the future.

Early home automation began with labor-saving machines. Self-contained electric or gas powered home appliances became viable within the 1900s with the introduction of electrical power transmission and led to the introduction of washing machines , water heaters , refrigerators, sewing machines, dishwashers, and garments dryers.

In 1975, the primary general purpose home automation network technology, X10, was developed. It is a communication protocol for electronic devices. It primarily uses electrical power transmission wiring for signalling and control, where the signals involves brief frequency bursts of digital data, and remains the foremost widely available.

1.3 EMBEDDED SYSTEMS

In these present scenario, the world is becoming more and more digital and using advanced technologies as like connected and automated. Embedded system is one of the advanced technology. This system is the one which has computer hardware with software embedded in it. The embedded system is designed to perform a specific task/function. But the task assigned has to be completed in a given time interval and it is an arrangement in which all its units, assembled work together according to set of info provided. When the system performs the given tasks at high speed, then it is used for real-time applications. A set of information or code embedded into the micro controller.

The characteristics of the Embedded Systems are:-

- Specific-Functioned
- Tightly constrained
- Reactive and Real time
- Microprocessors based
- Memory
- HW-SW systems
- Accuracy, reliability and adaptability.

Adavantages

- Easily Customizable
- Less power consumption
- Cheap
- Enhanced performance

Disadvantages

- High development effort
- Larger time to market

1.3.1 NODE MCU Board

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

The Node MCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. Node MCU has 128 KB RAM and 4MB of non-volatile storage to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. Node MCU are often powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

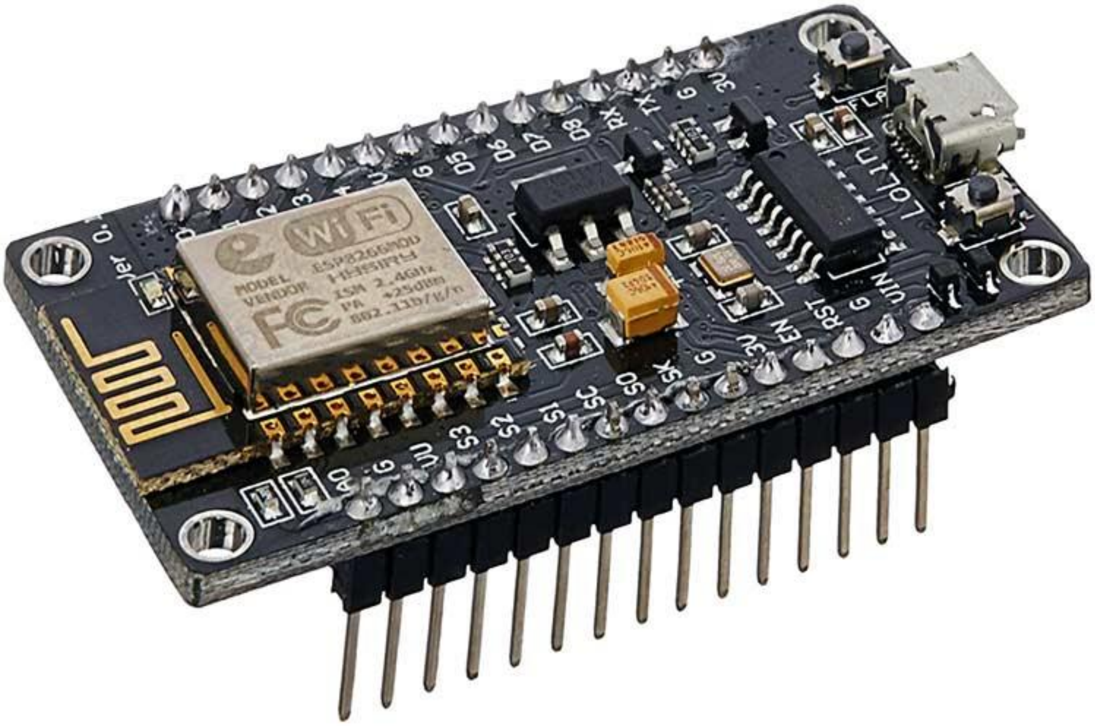


Figure 1.3.1 Node MCU Development Board

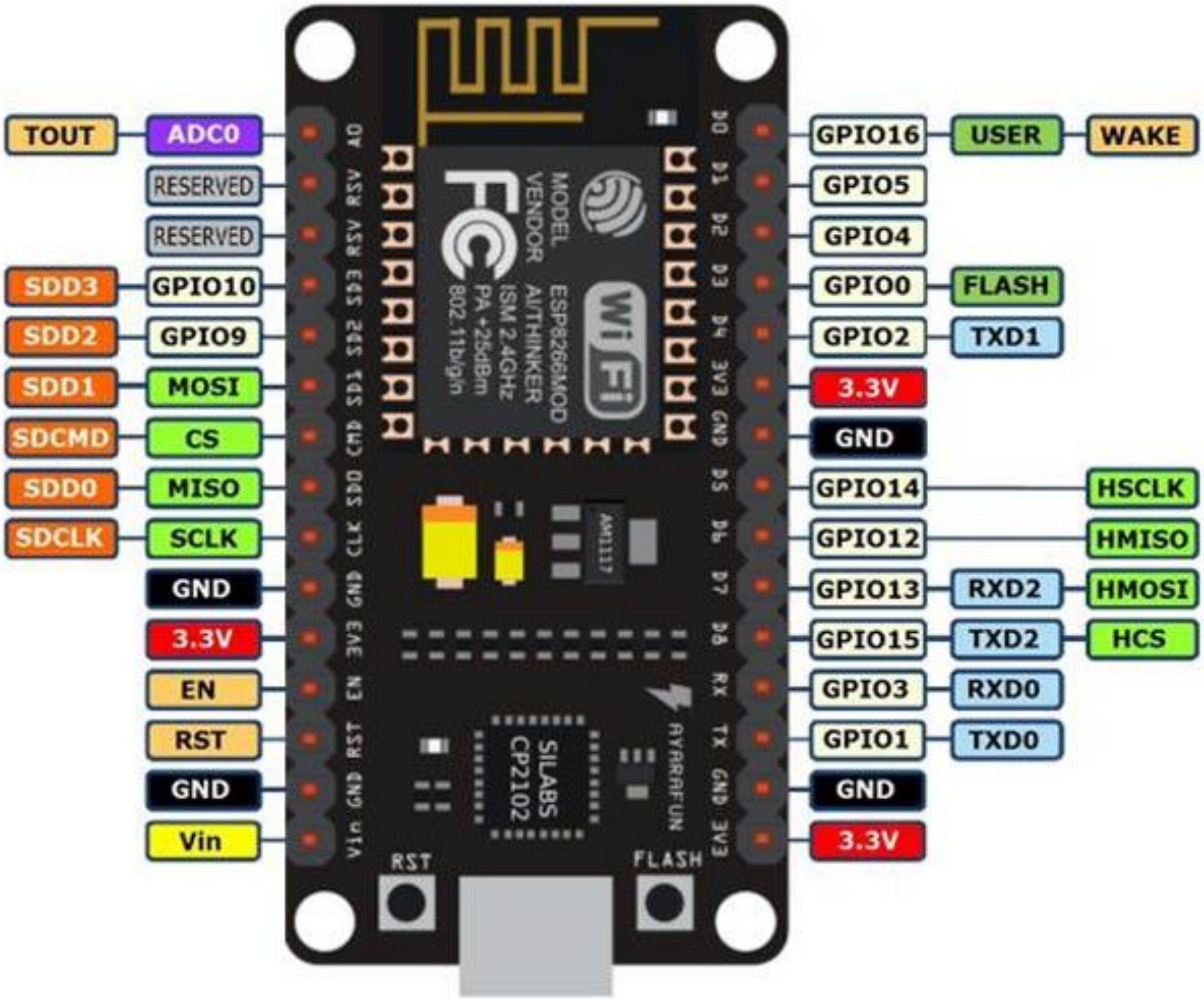


Figure 1.3.2 Node MCU ESP8266 Pinout

1.3.1.1 Node MCU ESP8266 SPECIFICATIONS & FEATURES

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL supported CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to suit smartly inside your IoT projects

1.3.1.2 Applications of Node MCU

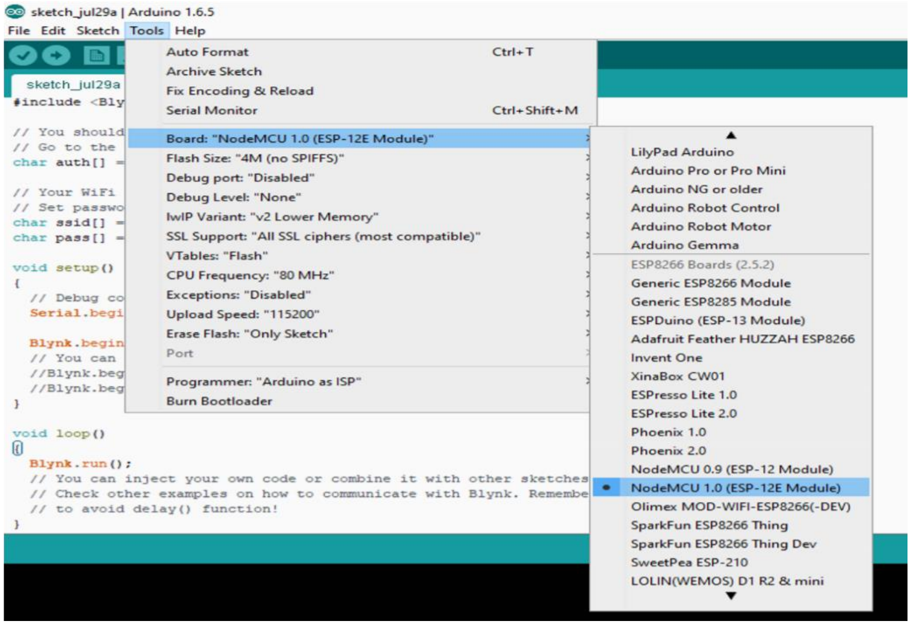
- Prototyping of IoT devices
- Low power battery operated applications
- Network projects
- Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

1.3.2 SOFTWARE USED

1.3.2.1 Arduino IDE

The Arduino IDE (integrated development environment) is a cross platform application which is written in the functions from C, C++ and JAVA. The Arduino IDE is also a derivative of Processing IDE. The Arduino IDE is used for easy to write and upload programs in Arduino boards by using a cable that is connected between board and IDE. The operating system for Arduino software can be Windows, Mac Os and Linux depending upon the user. The IDE has a software library from the wiring projects and to provide a common input and output procedures.

- *setup()*: a function that runs once at the start of a program and that can declare and initialize settings.
- *loop()*: a function that is called repeatedly until the board powers off.



```
#include <Bly

// You should
// Go to the
char auth[] =

// Your WiFi
// Set passwo
char ssid[] =
char pass[] =

void setup()
{
  // Debug co
  Serial.begi

  Blynk.begin
  // You can
  //Blynk.beg
  //Blynk.beg
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches
  // Check other examples on how to communicate with Blynk. Remembe
  // to avoid delay() function!
}
```

Board: "NodeMCU 1.0 (ESP-12E Module)"
Flash Size: "4M (no SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
SSL Support: "All SSL ciphers (most compatible)"
VTables: "Flash"
CPU Frequency: "80 MHz"
Exceptions: "Disabled"
Upload Speed: "115200"
Erase Flash: "Only Sketch"
Programmer: "Arduino as ISP"

Figure 1.3.2.1 Arduino IDE Setup

1.3.2.2 Programming Node MCU ESP8266 with Arduino IDE

The Node MCU Development Board can be easily programmed with Arduino IDE since it has an easy access. Programming Node MCU with the Arduino IDE may hardly take 5-10 minutes or depending upon the task that is to be performed. All you need is the Arduino IDE, a USB cable and the Node MCU board.

ADVANTAGES

- Huge documentation and support
- Large library collection
- Open source software
- Easily portable
- Less power consumption
- Highly customizable.

APPLICATIONS

- Home Automations
- ISP Programming
- Entry level circuit designing

CHAPTER 2
LITERATURE SURVEY

The expeditiously growing internet has opened new horizons for development in various fields. The home automation industry has seen a rapid growth in the previous years. It has become a topic of interest for many people around the world. Vishwateja Mudiam Reddy & Naresh Vinay in their paper “Internet of Things Enabled Smart Switch” designed a system which integrates the cloud and web app.

With the assistance of flip-flops, logic gates and a processor, the switches might be controlled. The proposed model was intended for reducing the value of those systems which was the most barrier within the wide adaptation of this technology. Khusvinder Gill & Shuang-Hua Yang created a common home gateway for ZigBee and Wi-Fi. This enables a remote control using a simple graphical user interface. The system was cost effective and had a good security inside the house. Salma and Dr. Radcliffe with a goal of increasing the popularity and reach of home automation designed a system that used the Novel Network Protocol. It gave the choice of controlling the commercial devices through a mobile phone or laptop. An additional network device had been used for remote access in place of a microcontroller.

A reliable and simple system with a power to integrate with very lesser efforts for off the shelf products was created by Carelin and I. Jacob Raglend. The system uses ZigBee for home automation and GSM for remote access. It didn’t provide any GUI and also it had been susceptible to security threats as anyone could access the system. Rozita Teymourzadeh, Salah Addin Ahmed designed a GSM based system for home automation. Using the GSM protocol, it became possible to access the system by using the Short Message System (SMS). The system also gave feedback to the user about the present state of any desired object.

CHAPTER 3
AIM AND SCOPE

3.1 PROJECT AIM

The aim of the project is to design and construct a home automation system which will remotely switch on or off any household appliances connected to it, using a microcontroller with a built-in WiFi for WIFI based Web Application.

3.2 PROJECT SCOPE

3.2.1 SMART HOMES

In the coming years, fully automated smart homes will surely become a reality because the home automation is growing rapidly in this tech world. Due to good user convenience and accessibility, smart homes are appealing for a wide range of people all over the world. The User can check for the electricity usage, the condition of their devices and receive notifications accordingly.

3.2.2 SMART CITIES

With increasing automation and IOT, devices can share info with each other. This will help in building modern and hi-tech cities. Cities that might be free form all types of pollutions, traffic accidents, etc. problems.

3.2.3 AGRICULTURE

The proposed system might be used in Agricultural as well. The various devices utilized in fields are often operated from any remote location.

3.3 PROJECT OBJECTIVE

The objective of this project is to implement a very low cost, reliable and scalable home automation system which will be used for remotely switching on or off any home appliance, using a microcontroller to achieve hardware simplicity low cost built-in wifi module at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands at the receiver end where loads are connected.

As technology is advancing so houses also are getting smarter. Modern houses are gradually shifting from conventional switches to centralized system, involving remote controlled switches. Presently, conventional wall switches located in several parts of the house makes it difficult for the user to travel near them to work. Even more it becomes harder for the elderly or physically handicapped people to try to do so. Remote controlled home automation system provides a latest solution with smart phones.

CHAPTER 4
METHODS AND MATERIAL USED

4.1 HARDWARE COMPONENTS USED AND SPECIFICATION

In this project some hardware components have been used to control the home appliances. Such as:

- Node MCU (ESP8266-12E)
- 4-Channel Relay Module
- Breadboard
- Jumper Wire
- 5 V 1Amp AC To DC Adapter

4.1.1 NODE MCU (ESP8266- 12E)

The heart of project is the WiFi enabled board that needs no introduction; the ESP8266 based Node MCU development board. Node MCU is an open-source Lua based software and development board specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is predicated on the ESP-12 module.

Node MCU was born out of the desire to overcome the limitations associated with the first versions of the ESP8266 module which was not compatible with the breadboards, it was difficult to power and had more difficulty in programming. The Node MCU board is easy to use at a very low cost and that quickly endeared it to the heart of makers and it is one of the most popular boards today.

Both the firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware is predicated on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, like lua-cjson and SPIFFS. Thanks to resource constraints, users got 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 generally 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 allowed a easier way of prototyping on breadboards. The design was initially supported on the ESP-12 module of the ESP8266, which may be a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, that is widely utilized in IoT applications.

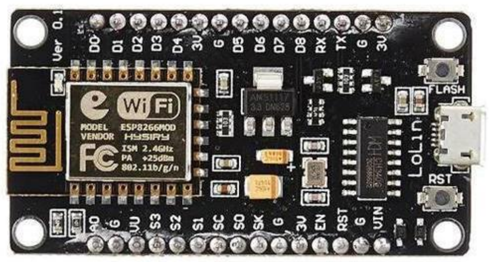

Ver 0.1
D0 D1 D2 D3 D4 3V G D5 D6 D7 D8 RX TX G 3V
Wi Fi
MODEL ESP8266MOD
VENDOR HYSIRY
ISM 2.4GHz
PA +25dBm
802.11b/g/n
FLASH
LoLin
RST
A0 G VU S3 S2 S1 SC S0 SK G 3V EN RST G VIN

Fig. 4.1.1 Node MCU

The sole reason for using the Node MCU over Raspberry Pi is that the Node MCU has inbuilt Wi-Fi. This reduces the cost and hence the Node MCU is cheaper than the other devices available in the market. Inbuilt Wi-Fi helps in remote access. The system is accessible from any remote location round the world provided an online connection. Once given an input, the device will still operate albeit there’s no internet access. The device can be physically handled as well.

4.1.1.1 FEATURES

- Open-source
- Interactive
- Programmable
- Less cost
- Simple
- Smart
- WI-FI enabled
- Arduino-like hardware IO
- Advanced API for hardware IO, which can dramatically reduce the repeated work for configuring and manipulating hardware. Code like arduino, but interactively in the Lua script.
- Nodejs style network API

4.1.1.2 Node MCU Development Board Pinout Configuration

Pin Category	**Name**	**Description**
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** Node MCU can be powered through the USB port **3.3V:** Regulated power supply of 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	Node MCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	Node MCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	Node MCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		Node MCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

4.1.1.3 BRIEF About NODE MCU

The Node MCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and function at 80MHz to 160 MHz adjustable clock frequency. Node MCU has 128 KB RAM and 4MB of non-volatile storage to store data and programs. Its higher processing power with the in-built Wi-Fi / Bluetooth and Deep Sleep Operating features makes it best for IoT projects.

Node MCU are often powered using the Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

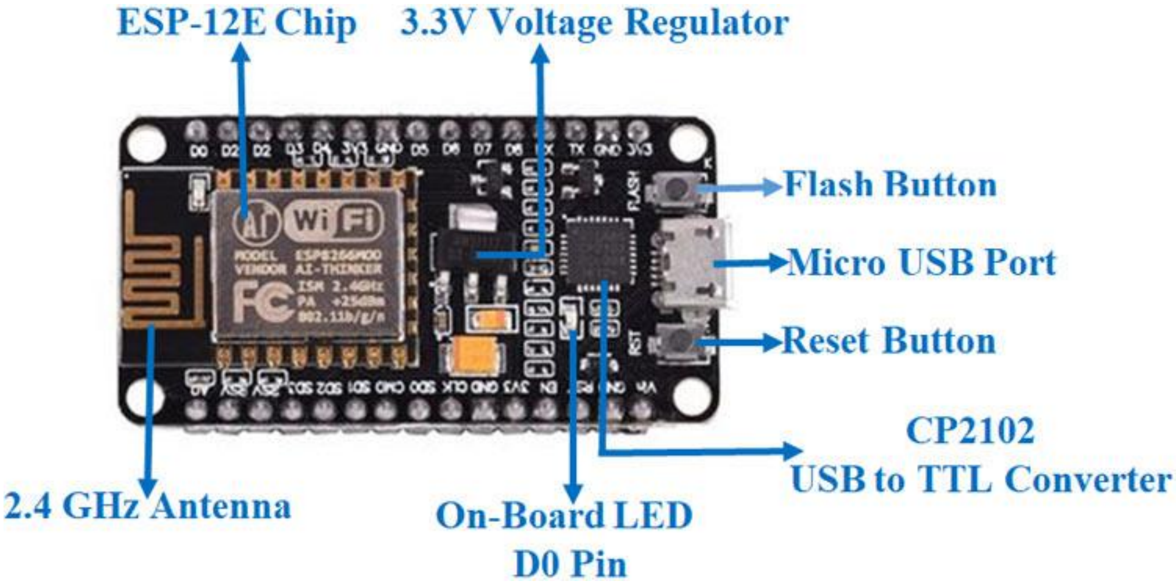


Figure 4.1.1.3 Briefing Node MCU

4.1.2 4- CHANNEL RELAY MODULE

The relay is that device that open or closes the contacts to cause the operation of the opposite electric control. It detects the intolerable or unwanted condition with an assigned area and gives the commands to the circuit breaker to disconnect the affected area. Thus isolates the system from damage.

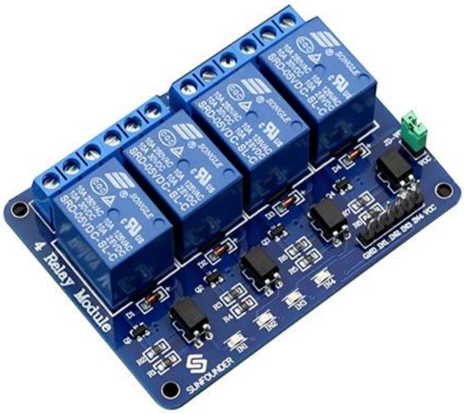


Fig. 4.1.2 Relay Module

The 4 Channel Relay Module may be a convenient board which may be used to control high voltage, high current load like motor, solenoid valves, lamps and AC load. It is designed to transmit and receive data with microcontroller such as Arduino, PIC and etc. The relay's terminals (COM, NO and NC) are being brought out with screw terminal. It also comes with a LED indicator to indicate the status of relay.

A relay module is an electrical switch that is operated by an electromagnet. The electromagnet gets activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an circuit.

A simple relay consists of wire coil wrapped around a soft iron core, or solenoid, an iron yoke that delivers a less reluctance path for the magnetic lines of forces, a movable iron armature and one or more sets of contacts. The movable armature is then hinged to the yoke and linked to one or more set of the moving contacts. Held in situ by a spring, the armature leaves a gap within the magnetic circuit when the relay is de-energized. While in this position, one of the two sets of contacts is closed while the other set remains open.

When electrical current is passed through a coil, it generates a magnetic flux that successively activates the armature. This movement of the movable contacts makes or breaks a reference to the fixed contact. When the relay is de-energized, the sets of contacts that were closed, open and breaks the connection and vice versa if the contacts were open. When switching off the current to the coil, the armature is returned, by force, to its relaxed position.

4.1.2.1 Working Principles of RELAY

It works on the principle of an electromagnetic attraction. When the circuit of the relay senses the fault current, it energies the electromagnetic field which produces the temporary magnetic field.

This magnetic field moves the relay armature for opening or closing the connections. The small power relay has only one contacts, and the high power relay has two contacts for opening the switch.

The inner section of the relay is shown in the figure below. It has an iron core which is wound by a control coil. The power supply is given to the coil through the contacts of the load and the control switch. The current flows through the coil produces the magnetic field around it.

Due to this magnetic field, the upper arm of the magnet attracts the lower arm. Hence close the circuit, which makes the current flow through the load. If the contact is already closed, then it moves oppositely and hence open the contacts.

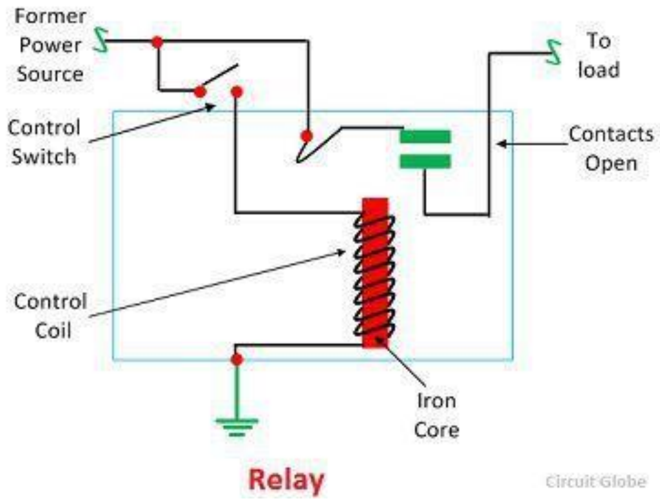


Figure 4.1.2.1 Working of Relay

4.1.2.2 CONSTRUCTION Of RELAY

The relay operates both electrically and mechanically. It consists electromagnetic and sets of contacts which perform the operation of the switching. The construction of relay is mainly classified into four groups. They are the contacts, bearings, electromechanical design, terminations and housing.

- **Contacts** – The contacts are the most important part of the relay that affects the reliability. The good contact gives limited contact resistance and reduced contact wear. The selection of the contact material depends upon the several factors like nature of the current to be interrupted, the magnitude of the current to be interrupted, frequency and voltage of operation.

- **Bearing** – The bearing may be a single ball, multi-ball, pivot-ball and jewel bearing. The single ball bearing is used for high sensitivity and low friction. The multi-ball bearing provides low friction and greater resistance to shock.

- **Electromechanical design** – The electromechanical design includes the design of the magnetic circuit and the mechanical attachment of core, yoke and armature. The reluctance of the magnetic path is kept minimum for making the circuit more efficient. The electromagnet is made up of soft iron, and the coil current is usually restricted to 5A and the coil voltage to 220V.

- **Terminations and Housing** – The assembly of an armature with the magnet and the base is made with the help of spring. The spring is insulated from the armature by moulded blocks which provide dimensional stability. The fixed contacts are usually spot welded on the terminal link.

4.1.2.3 Connectivity Of RELAY

Relay module is being connected directly to digital circuits including microcontroller kits easily to control big loads by a microcontroller. The inputs IN1, IN2, IN3 and IN4 operate four relays with voltage between 3-5 volts DC. Input and output circuits are separated by Optocouplers to protect digital circuits in case connection mistakes happened or short circuits.

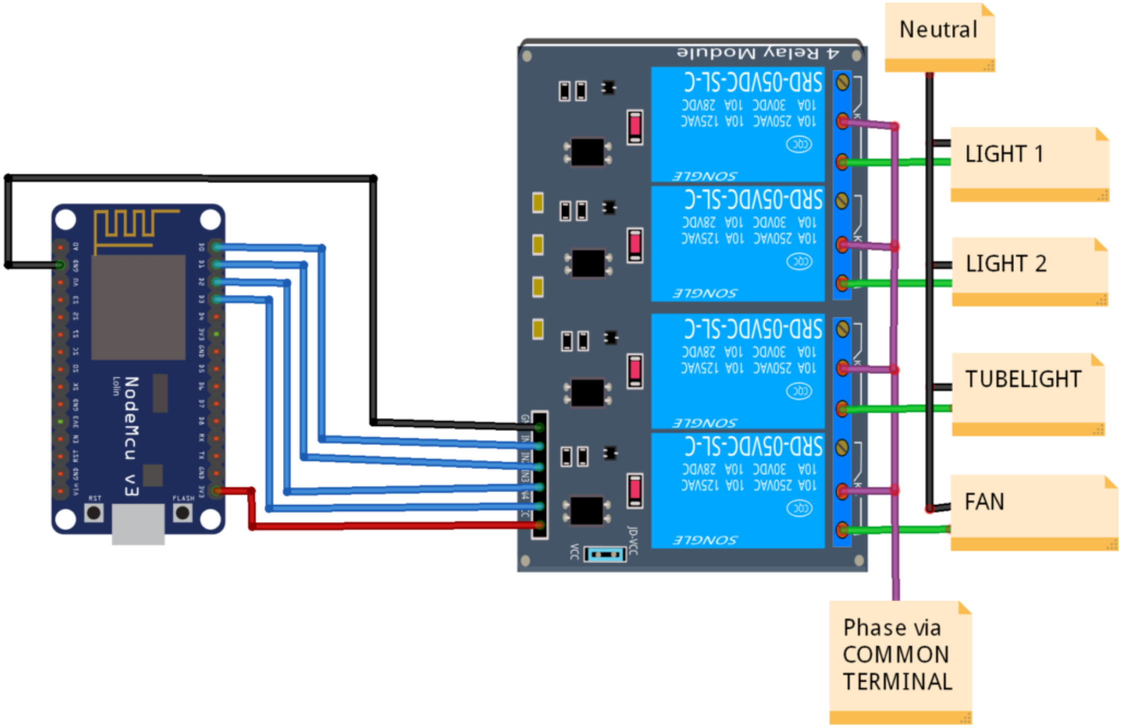


Figure 4.1.2.3 Connectivity of Relay

4.1.3 BREADBOARD

4.1.3.1 What is BREADBOARD?

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.

Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. For this reason, solderless breadboards are also popular with students and in technological education. Older breadboard types did not have this property. A stripboard (Veroboard) and similar prototyping printed circuit boards, which are used to build semi-permanent soldered prototypes or one-offs, cannot easily be reused. A variety of electronic systems may be prototyped by using breadboards, from small analog and digital circuits to complete central processing units (CPUs).

Compared to more permanent circuit connection methods, modern breadboards have high parasitic capacitance, relatively high resistance, and less reliable connections, which are subject to jostle and physical degradation. Signaling is limited to about 10 MHz, and not everything works properly even well below that frequency.

4.1.3.2 COMMON USE

A common use in the system on a chip (SoC) era is to obtain an microcontroller (MCU) on a pre-assembled printed circuit board (PCB) which exposes an array of input/output (IO) pins in a header suitable to plug into a breadboard, and then to prototype a circuit which exploits one or more of the MCU's peripherals, such as general-purpose input/output (GPIO), UART/USART serial transceivers, analog-to-digital converter (ADC), digital-to-analog converter (DAC), pulse-width modulation (PWM; used in motor control), Serial Peripheral Interface (SPI), or I²C.

4.1.3.3 TYPICAL SPECIFICATIONS:

- **TERMINAL STRIPS**

  The main areas, to hold most of the electronic components.

  In the middle of a terminal strip of a breadboard, one typically finds a notch running in parallel to the long side. The notch is to mark the centerline of the terminal strip and provides limited airflow (cooling) to DIP ICs straddling the centerline[citation needed]. The clips on the right and left of the notch are each connected in a radial way; typically five clips (i.e., beneath five holes) in a row on each side of the notch are electrically connected. The five columns on the left of the notch are often marked as A, B, C, D, and E, while the ones on the right are marked F, G, H, I and J. When a "skinny" dual in-line pin package (DIP) integrated circuit (such as a typical DIP-14 or DIP-16, which have a 0.3-inch (7.6 mm) separation between the pin rows) is plugged into a breadboard, the pins of one side of the chip are supposed to go into column E while the pins of the other side go into column F on the other side of the notch.

- **BUS STRIPS**

  To provide power to the electronic components.

  A bus strip usually contains two columns: one for ground and one for a supply voltage. However, some breadboards only provide a single-column power distribution bus strip on each long side. Typically the row intended for a supply voltage is marked in red, while the row for ground is marked in blue or black. Some manufacturers connect all terminals in a column. Others just connect groups of, for example, 25 consecutive terminals in a column.

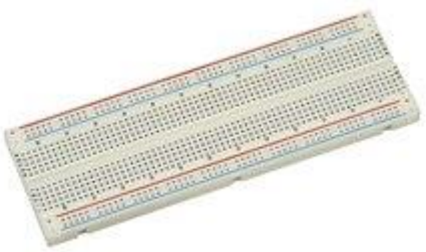


Fig. 4.1.3.3 Bus Strips

4.1.4 JUMPER WIRES

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire or cable) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Fig. 4.1.3.4 Jumper Wires

4.1.4.1 TYPES

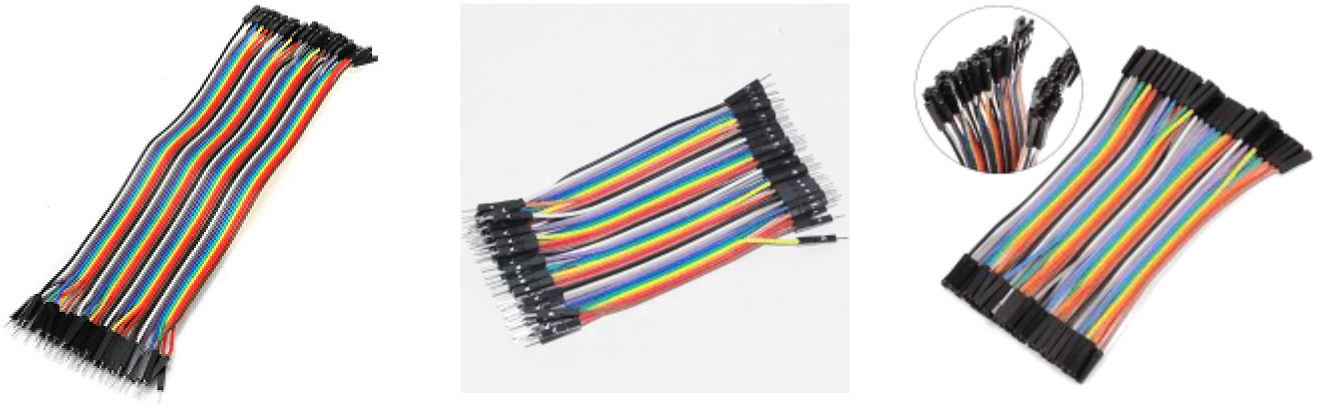
There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

- **Solid tips** – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
- **Crocodile clips** – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
- **Banana connectors** – are commonly used on test equipment for DC and low-frequency AC signals.
- **Registered jack (RJnn)** – are commonly used in telephone (RJ11) and computer networking (RJ45).
- **RCA connectors** – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.

- **RF connectors** – are used to carry radio frequency signals between circuits, test equipment, and antennas.
- **RF jumper cables** - Jumper cables is a smaller and more bendable corrugated cable which is used to connect antennas and other components to network cabling. Jumpers are also used in base stations to connect antennas to radio units. Usually the most bendable jumper cable diameter is 1/2".

4.1.4.2 VERSIONS OF SOLID TIPS JUMPER WIRES

Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you'll need.



MALE TO FEMALE JUMPER WIRES

MALE TO MALE JUMPER WIRES

FEMALE TO FEMALE JUMPER WIRES

Figure 4.1.4.2 Types of Jumper Wires

4.1.5 5V 1 AMPS AC To DC ADAPTER

This AC to DC power supply will do 5V at 1A! They're switch mode power supplies which means the output is regulated to 5V (no more 14V outputs!). These have a standard USB 'A' connector for the output so you can power your Arduino, Raspberry Pi, etc. through a USB cable.

Figure 4.1.5 Adapter

4.1.5.1 SPECIFICATIONS

- Input Voltage (V): 100 ~ 280 VAC @50 ~ 60Hz.
- Input current (mA): 100.
- Output Power: 5V 1A.
- Input Plug: 2-Pin EU type.
- Output Plug: 5.5mm DC plug.

4.2 SOFTWARE REQUIREMENTS

4.2.1 ARDUINO IDE

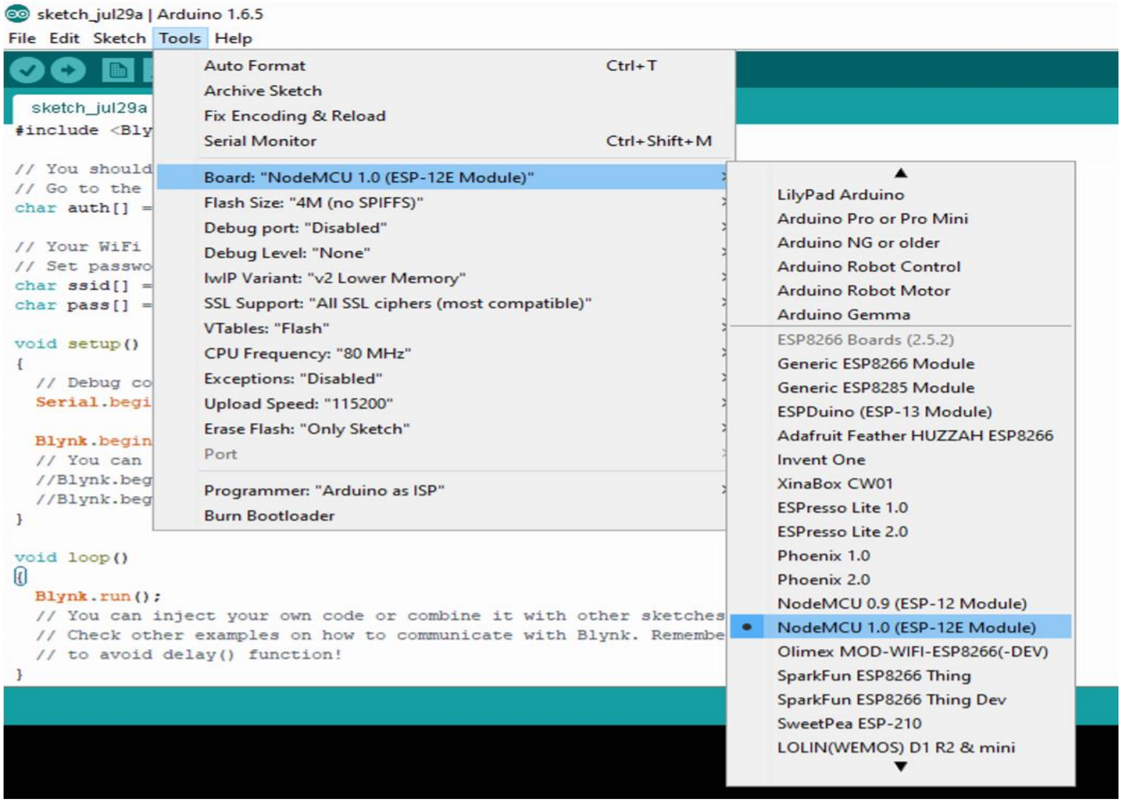
The Arduino IDE (integrated development environment) is an cross platform application that is written in the functions from C, C++ and JAVA. The Arduino IDE is an derivative of Processing IDE. The Arduino IDE is used for easy to write and upload programs in Arduino boards by using an cable that was connected between board and IDE. The operating system of Arduino software is Windows, Mac Os and Linux. The IDE supplies a software library from the wiring projectsand to provide a common input and output procedures.

- *setup()*: a function that runs once at the start of a program and that can initialize settings.
- *loop()*: a function called repeatedly until the board powers off.

The system is controlled using a Web Page which is based on IoT platform. The intended device can be powered ON or OFF using the above application. Also, it provides the facility to set the intensities of different appliances. The system becomes platform independent due to the use of a web application. It can be operated from any location by just opening the IoT platform web application. The web application also serves as a platform for managing the devices and the data.

4.2.1.1 NODE MCU CODE VIA ARDUINO IDE

To code Node MCU via Arduino IDE, the Node MCU needs to be added to Arduino IDE library first by adding this address to Arduino IDE preferences. After this reference is added to Arduino IDE, download node MCU to boards manager and then select Node MCU 1.0 (ESP12E Module). After node MCU is added to Arduino IDE library, upload this code with changing hotspot name and password.


sketch_jul29a | Arduino 1.6.5

Board: "NodeMCU 1.0 (ESP-12E Module)"
Flash Size: "4M (no SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
IwIP Variant: "v2 Lower Memory"
SSL Support: "All SSL ciphers (most compatible)"
VTables: "Flash"
CPU Frequency: "80 MHz"
Exceptions: "Disabled"
Upload Speed: "115200"
Erase Flash: "Only Sketch"
Programmer: "Arduino as ISP"

```
#include <Bly

// You should
// Go to the
char auth[] =

// Your WiFi
// Set passwo
char ssid[] =
char pass[] =

void setup()
{
  // Debug co
  Serial.begi

  Blynk.begin
  // You can
  //Blynk.beg
  //Blynk.beg
}

void loop()
{
  Blynk.run();
  // You can inject your own code or combine it with other sketches
  // Check other examples on how to communicate with Blynk. Remembe
  // to avoid delay() function!
}
```

Figure 4.2.1.1 SETTING UP NODE MCU IN ARDUINO IDE

The code includes the hotspot name and password match with the android. The code does not need to identify the relay input, as it is included in [Arduino IDE]. When access point is given by the microcontroller sent as password and SSID is the name of smart phone hotspot.

4.3 EXSTING SYSTEM

A home automation system is an automating the bulk of electronic and electrical tasks within a home. It uses a combination of hardware and software to enable control and management over appliances and devices within a home.

Home automation not only refers to reduce human efforts but also energy efficiency and time saving. Here home automation is done by using ESP8266 – WiFi SoC and Blynk Play store App.

4.3.1 WHAT THEY HAVE FOLLOWED IN EXSTING SYSTEMS

In this existing system, home automation is done by using the Blynk app which sends the command by it's Blynk Server. Home Automation mean controlling lighting, climate, entertainment systems, and appliances without a manual switch.

Blynk is a mobile application which has its own server to process user requests. It is an open source application and anybody can use it in their Home Automation to control devices, monitor sensor data and get a notification by some trigger actions.

Blynk app has been used to read data from sensors located in home environment and user controls home appliances based on these data. Being busy in hectic schedule of daily life user may not be able to read sensor data continuously to take some action through app.

Blynk has a nice GUI but one may sometimes face a problem due to it's busy server and get a late response or zero response.

4.4 PROPOSED SYSTEM

This project uses the Cloud and a Web Browser to control the manually operated switches. A cloud server is created for the environment where the switches are mounted. The switches are interfaced with Node MCU which has an inbuilt Wi-Fi. It can use this to enable or disable the switches. The user communicates with the processor through the Web Browser. The processor then controls the switches based on the commands received from the user and also updates the user about the status of the switches after the control operation is performed to the cloud.

Hardware Requirements

- Node MCU (ESP8266-12E) DEVELOPMENT BOARD
- 4-Channel Relay Module
- Breadboard
- Jumper Wire
- 5 V 1Amp AC To DC Adapter

Software Requirements

Arduino IDE – Compiler

Language – Embedded C

4.4.1 BLOCK DIAGRAM

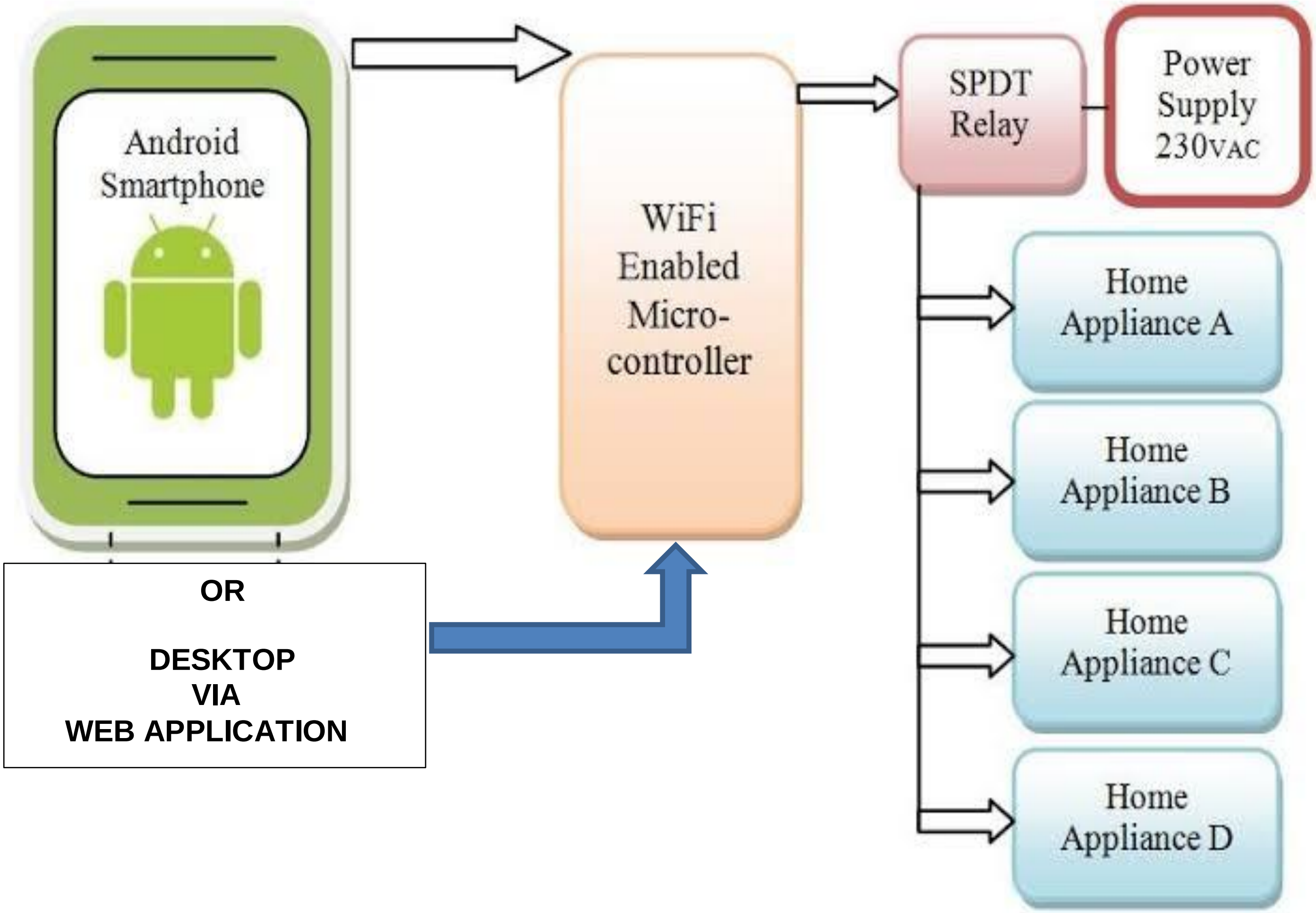


Figure 4.4.1 Block diagram for proposed system.

4.4.2 CIRCUIT DIAGRAM

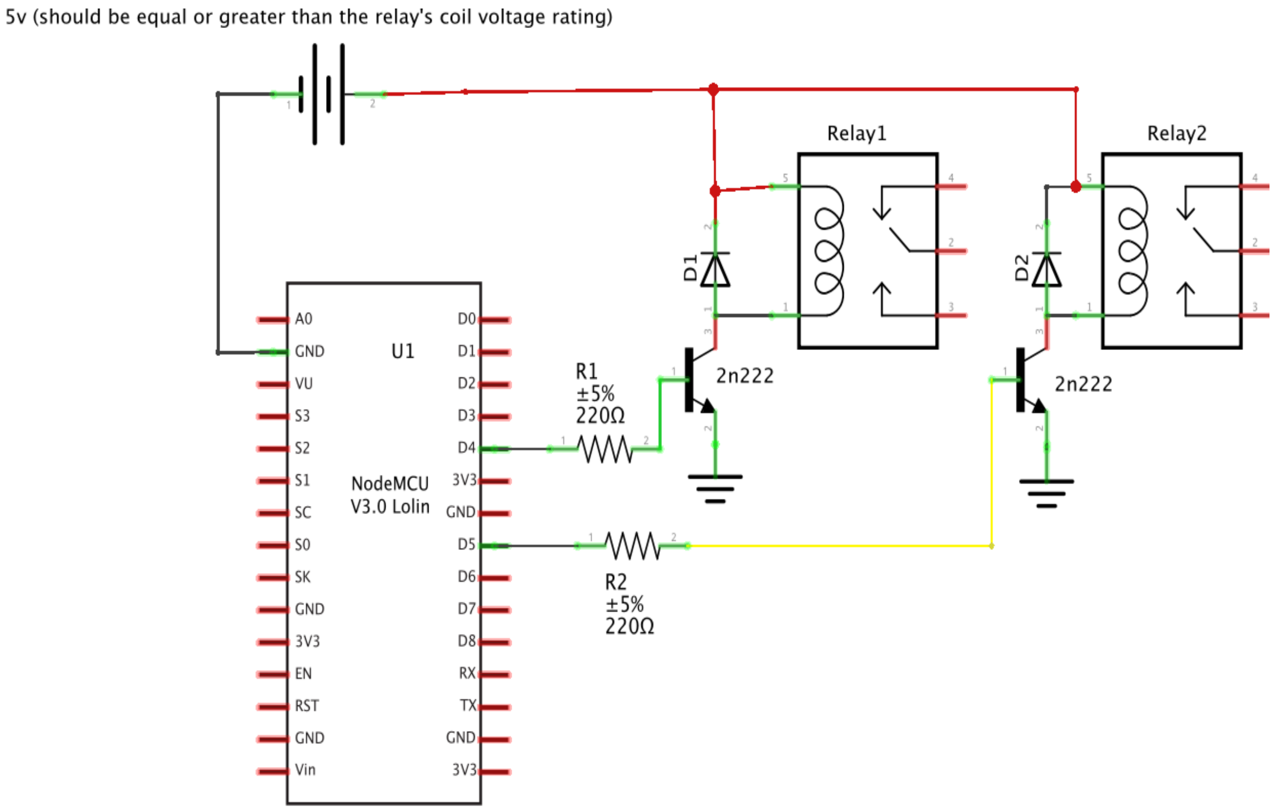


Figure 4.4.2 Circuit diagram for proposed system.

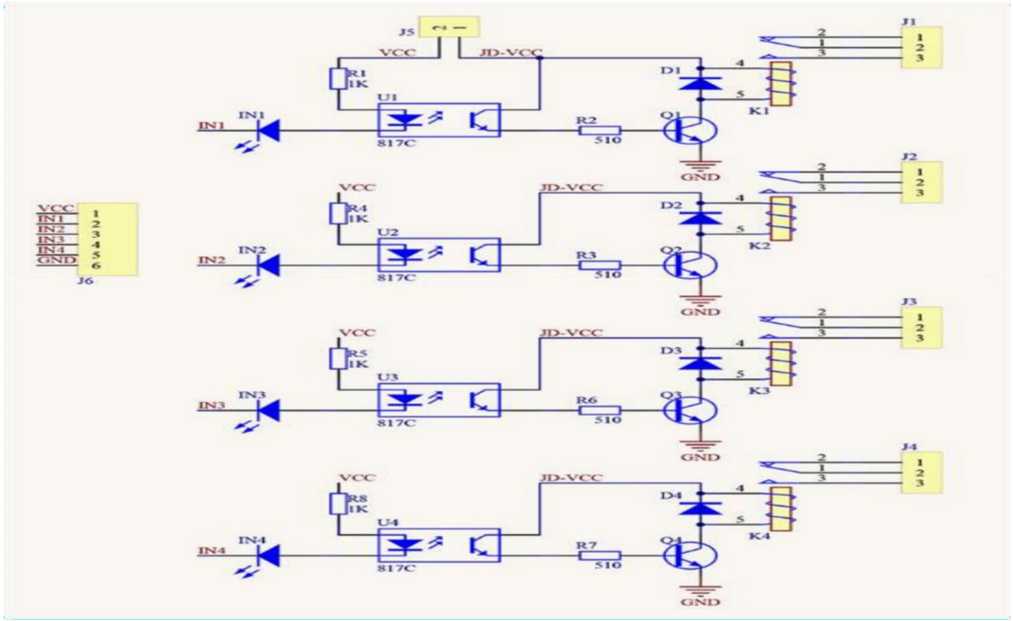


Figure 4.4.3 Circuit diagram of RELAY MODULE

4.4.3 Working process

The heart of this project is the WiFi enabled board that needs no introduction; the ESP8266 based Node MCU development board. It is an open source platform for developing WiFi based embedded systems and it is based on the popular ESP8266 WiFi Module, running the Lua based Node MCU firmware. Node MCU was born out of the desire to overcome the limitations associated with the first versions of the ESP8266 module which was not compatible with breadboards, it was difficult to power and even more difficult to program. The Node MCU board is easy to use, low cost and that quickly endeared it to the heart of makers and it is one of the most popular boards today.

For this project, we will add a 4-channel relay module to the ESP8266 board. The project flow involves the control of Node MCU's GPIOs from a webpage on any device connected on the same network as the board. The status of the GPIOs control the coils of the relays and that causes the relay to alternate between normally open (NO) and normally closed (NC) condition depending on the state of the GPIO, thus, effectively turning the connected appliance "ON" or "OFF".

4.4.3.1 CONNECTIONS

To make the connections easy to follow, here is a pin map of the connection between the NODE MCU and the Relay Module:

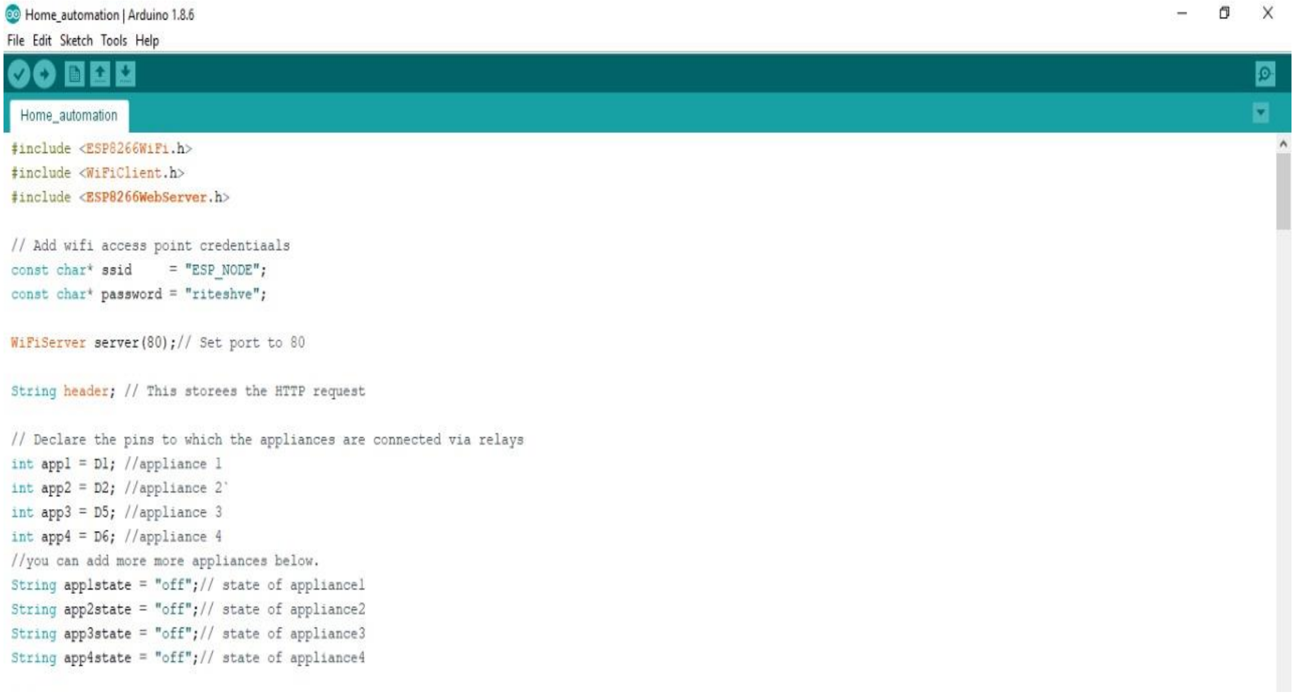
Node MCU	**–**	**Relay Module**
3.3V	-	VCC
GND	-	GND
D1	-	IN1
D2	-	IN2
D5	-	IN3
D6	-	IN4

4.4.3.2 CODE

One of the easiest way to program Node MCU is via the Arduino IDE. This, however, requires setting up the Arduino IDE by installing the board support file for Node MCU. The code is based on the ESP8266WiFi.h library which allows the easy use of WiFi functionalities of the board. It contains all we need to create or join a WiFi access point and also create a server and client which are all important for this project. The library comes attached with the Node MCU board files for the Arduino, so there is no need to install it once the board files have been installed.

The code for this project will enable us to control appliances connected to the GPIOs (via relays) of the Node MCU board remotely. To start with, we include the library that we will use for the project, which in this case, is the **ESP8266WiFi.h** library.

Next, we add the credentials of the WiFi access point to which the Node MCU will be connected. Ensuring that the username and password are between the double quotes. We also specify the port through which the system will communicate and create a variable to hold requests. As shown in the below fig.



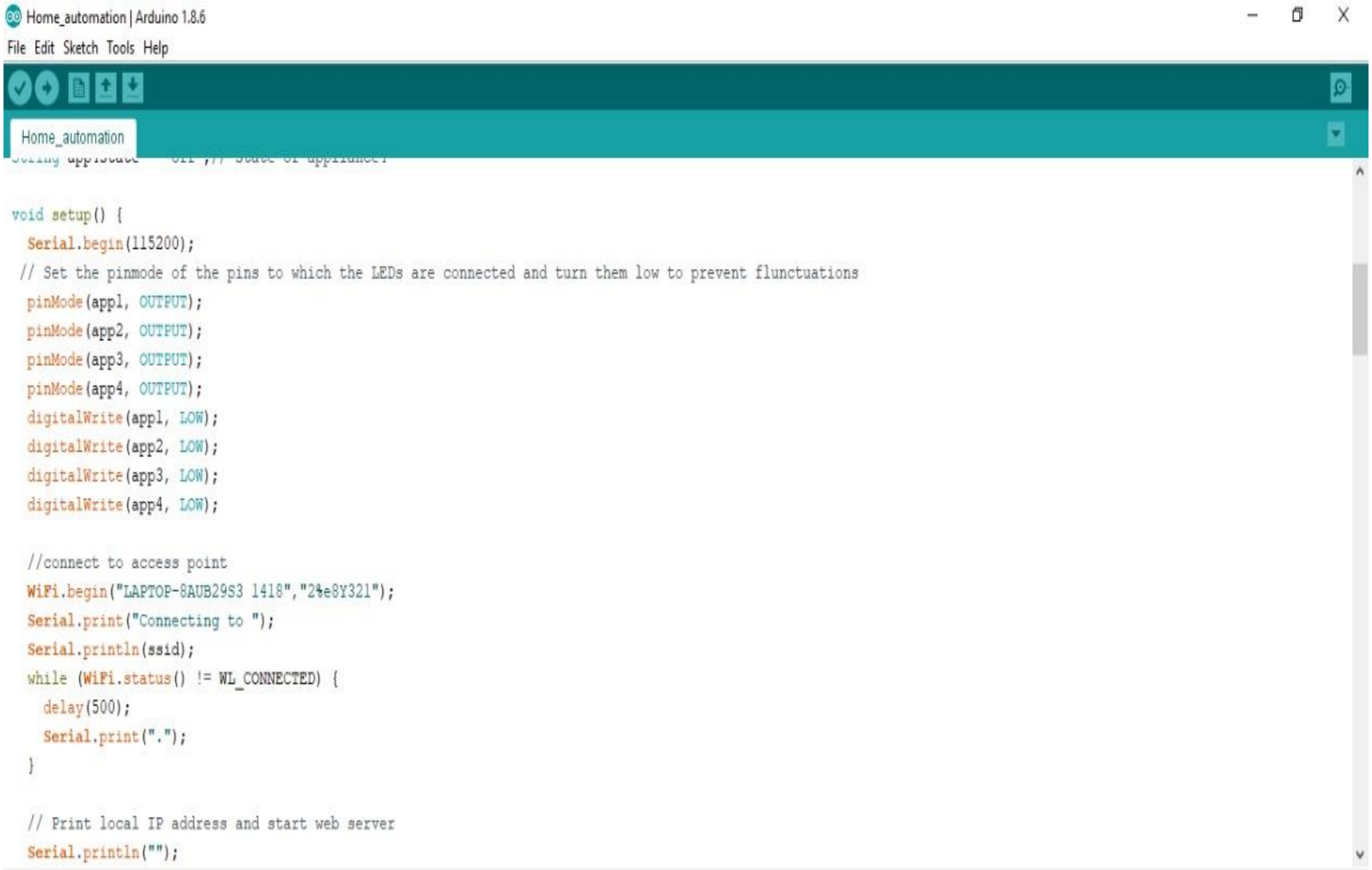
```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

// Add wifi access point credentiaals
const char* ssid     = "ESP_NODE";
const char* password = "riteshve";

WiFiServer server(80);// Set port to 80

String header; // This storees the HTTP request

// Declare the pins to which the appliances are connected via relays
int app1 = D1; //appliance 1
int app2 = D2; //appliance 2`
int app3 = D5; //appliance 3
int app4 = D6; //appliance 4
//you can add more more appliances below.
String app1state = "off";// state of appliance1
String app2state = "off";// state of appliance2
String app3state = "off";// state of appliance3
String app4state = "off";// state of appliance4
```

Figure 4.4.3.2 Code 1

Next, is the void setup() function. We start by initializing the serial monitor (as it will be used for debugging later on) and setting the pin Modes of the pins to which the relays are connected as output. We then set the pins “LOW” to ensure the system starts at OFF state.

Now, we connect to the access point using the credentials provided as arguments to the WiFi.begin() function and we use the WiFi.status() function to check if connection was successful. The system will keep trying until the connection is successful.

As shown in the below Figure.

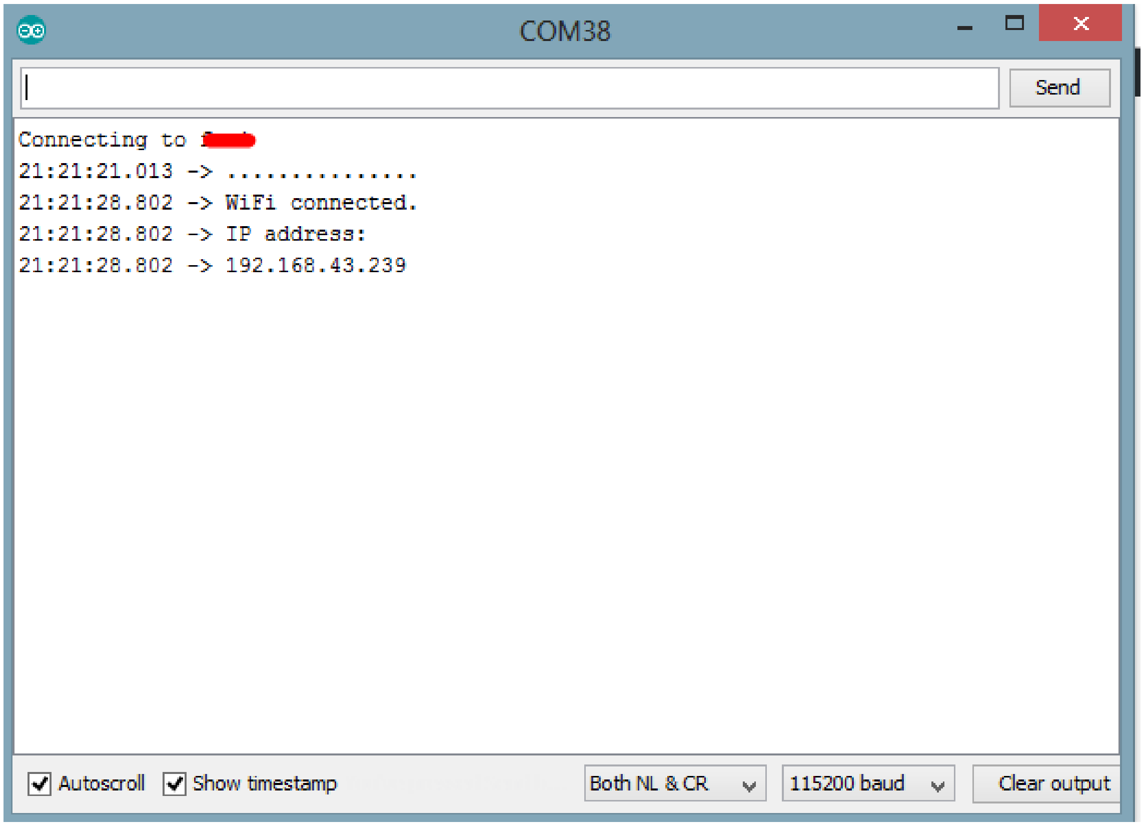


```
String appState = "off";// State of appliance1

void setup() {
  Serial.begin(115200);
 // Set the pinmode of the pins to which the LEDs are connected and turn them low to prevent flunctuations
  pinMode(app1, OUTPUT);
  pinMode(app2, OUTPUT);
  pinMode(app3, OUTPUT);
  pinMode(app4, OUTPUT);
  digitalWrite(app1, LOW);
  digitalWrite(app2, LOW);
  digitalWrite(app3, LOW);
  digitalWrite(app4, LOW);

  //connect to access point
  WiFi.begin("LAPTOP-8AUB29S3 1418","2%e8Y321");
  Serial.print("Connecting to ");
  Serial.println(ssid);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  // Print local IP address and start web server
  Serial.println("");
```

Figure 4.4.3.3 Code 2

If the connection is successful, a text is printed on the serial monitor to indicate this, along with the IP address of the Node MCU. This IP address becomes the web address for the server and should be entered on any web browser on the same network as the server so we are able to access it.

Upload the code to the Node MCU. Ensure everything is connected as described under the schematics section. After uploading the code, the IP address of your web server displayed in the serial monitor as shown below.



```
Connecting to [redacted]
21:21:21.013 -> ...............
21:21:28.802 -> WiFi connected.
21:21:28.802 -> IP address:
21:21:28.802 -> 192.168.43.239
```

☑ Autoscroll ☑ Show timestamp Both NL & CR 115200 baud

Figure 4.4.3.4 Serial Monitor

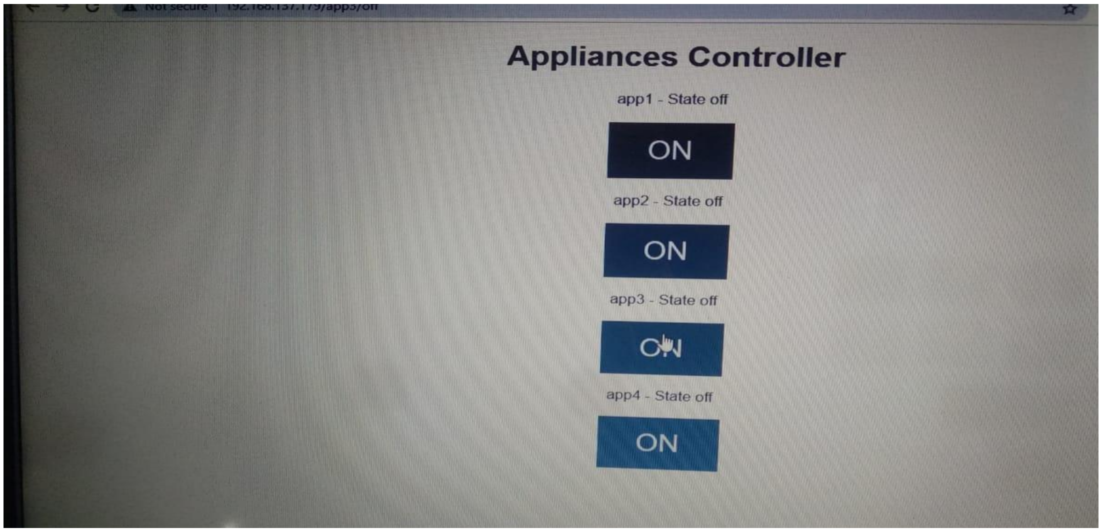
PRECAUTIONS

It’s important to note that, connecting the appliances to the relay involves interaction with AC voltages which could be dangerous. Ensuring to have experience interacting with AC voltages and do so in a safe manner.

CHAPTER 5
RESULTS AND DISCUSSION

5.1 Light Control Test Results

The Light Control Test is done by pressing the ON / OFF button widget on the Web application on the respective Android smart phone/Pc for lights and fans. This is done after the system is turned on and connected to a Wi-Fi internet connection. If at any time the internet connection is lost or bad signal, then it also affects system performance.



Appliances Controller

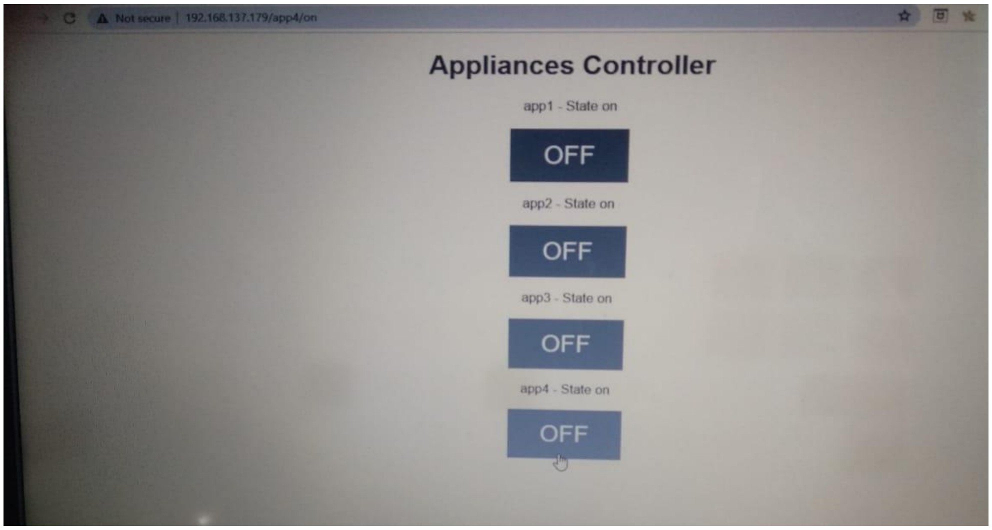
app1 - State off

app2 - State off

app3 - State off

app4 - State off

Figure 5.1.1 Web App Output 1



Appliances Controller

app1 - State on

app2 - State on

app3 - State on

app4 - State on

Figure 5.1.2 Web App Output 2

5.2 SYSTEM ANALYSIS

From testing the entire system above, the smart home works according to the purpose of this project. Comparison of this research with previous studies, namely this study uses control buttons, thus increasing the diversity of the smart home system itself. Also, used a microcontroller that is different from previous studies that is the Node MCU ESP8266 module which has advantages compared to other microcontrollers. The smart home has been successfully built with hardware arranged in such a way that it can achieve results that are as expected. In this case the hardware plays a very important role as the main device is the Node MCU ESP8266 module. The advantages of using the Node MCU ESP8266 are more practical than buying various components and then assembling them by yourself.

5.3 The Final Hardware Circuit Connections

Using components and materials mentioned above. Figure below shows the project that's used as an (IoT) system controlled by Web application. Loads used in this project are bulbs, they can be changed with other devices by changing bulbs with AC plugs to connect home-use devices or equipment.

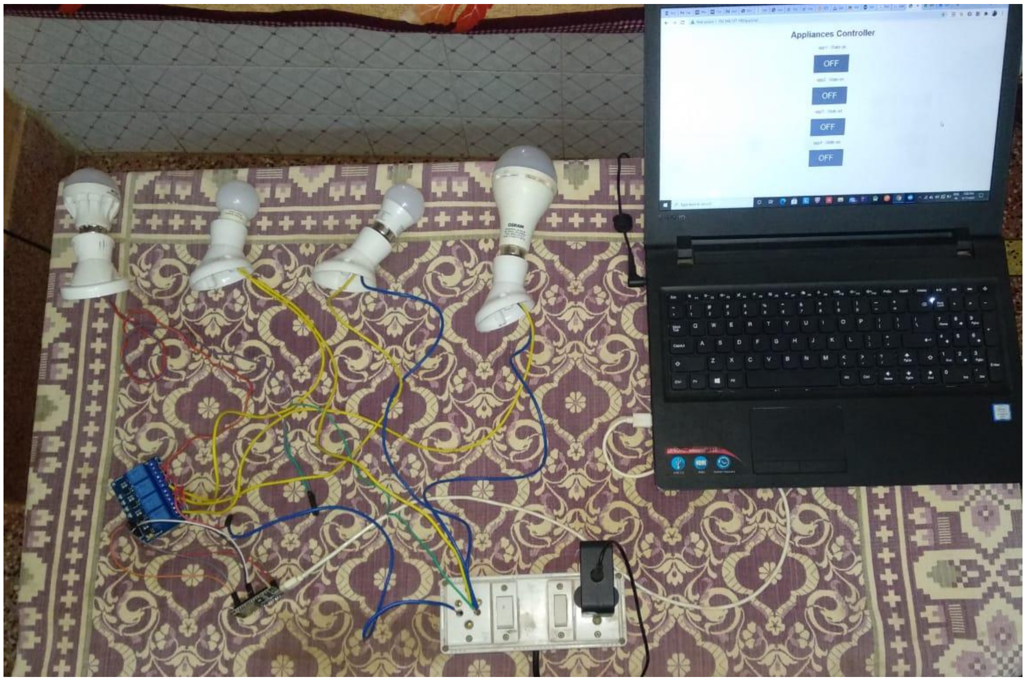


Figure 5.3.1 Output Hardware Connection 1

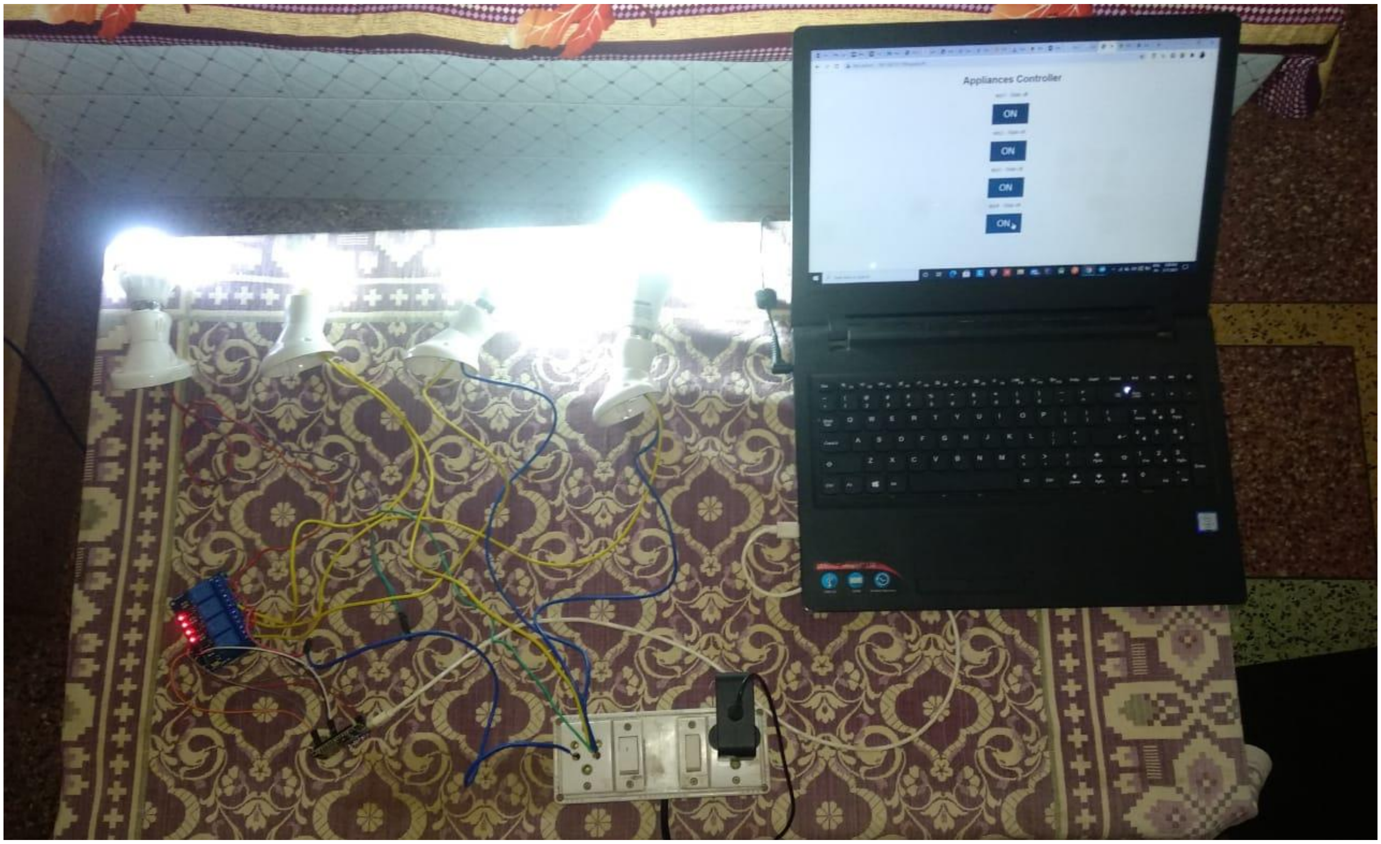


Figure 5.3.2 Output Hardware Connection 2

In the above figures, 4-Channel Relay module is interfaced with the Node MCU and Four AC power loads are connected to the Relay i.e at IN1,IN2,IN3,IN4. User can operate this loads using GUI/Web Server. Appliances can be turned ON/OFF by sending On/OFF commands from the web application. The processor then controls the switches based on the commands received from the user and also updates the user about the status of the switches after the control operation is performed to the cloud.

CHAPTER 6
CONCLUSION

Based on the results of analysis of all data obtained by testing the smart home with the Internet of Things based Node MCU ESP6288 module, the following conclusions can be drawn:

1) Smart Home with Internet of Things (IoT) based Node MCU ESP8266 Module can be designed with various components hardware and software support so that it can be arranged into a smart home system that is controlled with the Web application according to what is intended.

2) The Smart Home with this Internet of Things (IoT) based Node MCU ESP8266 Module can be implemented to control some of the home electronics performance including lighting controls, fan control, temperature monitoring, early warning systems and etc.

3) Main purpose of home automation system is to provide ease to people to control differnt home appliances with the help of the web application using their mobile phones or desktop and to save time and money.

REFERENCES

1) BOHORA, Bharat; MAHARJAN, Sunil; SHRESTHA, Bibek Raj. IoT Based Smart Home Using Blynk Framework. Zerone Scholar, [S.l.], v. 1, n. 1, p. 26-30, dec. 2016. ISSN 25422774. google scholar.

2) DC-DC Step Down Converter Power Supply Provides Regulated 5VDC Output with Range Input of 10-32VDC, Model GTD21088L-1505-T2.

3) Home Automation Using Internet of Thing 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) Published: 2016. Google Scholar.

4) Internet of Things in Home Automation and Energy Efficient Smart Home Technologies Simon G. M. Koo Department of Computer Engineering, Santa Clara University, CA 95053, USA

5) Low Cost Implementation of Smart Home Automation Ravi Kishore Kodali Department of Electronics and Communication Engineering National Institute of Technology, Warangal , 506004 India

6) Mobile based home automation using Internet of Things (IoT) 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) Published: 2015

7) NodeMCU Features and Pinout. A Brief Tutorial on the Introduction to NodeMCU V3.

8) Yoyosteven in Circuits Microcontrollers. NODEMCU 1.0 (ESP8266) CONTROLLED RELAY USING BLYNK (OVER THE WEB).

9) 5V 4-Channel Relay Interface Board, Standard Interface that can be Controlled Directly by Microcontroller.

10) 15-17 March 2018 U. Venkanna IoT Based Smart Home Automation System Using Sensor Node. Google Scholar.