STUDY OF IOT BASED HOME AUTOMATION USING NODE MCU AND BLYNK APP

A Minor project report submitted in partial fulfilment of the requirements for

the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

BY

M. MANOJ KUMAR	(20A55A0216)
T. UPENDRARAO	(19A51A0242)
P. YAMINI	(19A51A0231)
SINTU KUMAR	(19A51A0238)

Under the esteemed guidance of

Sri B. MANMADHA KUMAR M.Tech.
Sr. Asst Professor

Department of EEE

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT
TEKKALI, SRIKALULAM (DIST.) A.P.

(Approved by AICTE, New Delhi & Permanently Affiliated to JNTKAKINADA)
2021-2022

ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An Autonomous Institution)

Department of Electrical And Electronics Engineering

CERTIFICATE

This is to certify that the minor project work entitled "**STUDY OF IOT BASED HOME AUTOMATION USING NODE MCU AND BLYNK APP** " is a bonified work done by **M. MANOJ KUMAR (20A55A0216), T. UPENDRARAO (19A51A0242), P.YAMINI (19A51A0231), SINTU KUMAR (19A51A0238),** and submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Electrical and Electronics Engineering

Project Guide :

Sri B. MANMADHA KUMAR, M.Tech

Sr. Assistant Professor

Department of E.E.E.

Head of the Department :

Dr. K. KIRAN KUMAR, M. E, Ph.D.

Head of the Department

Department of E.E.E.

ACKNOWLEDGEMENT

It is indeed with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to director **Prof.V.V.Nageswara Rao and** principal **Dr.A.S.Srinivasa Rao, for** the facilities provided to the accomplish this mini project.

We would like to thank our head of the department **Dr. K. KIRAN KUMAR,** for his SUPPORT throughout project.

We feel elated in manifesting our sense of gratitude to our supporting guide **SRI B. MANMADHA KUMAR** for his valuable guidance for **STUDY OF IOT BASED HOME AUTOMATION USING NODE MCU AND BLYNK APP.**

We are extremely grateful to my department staff members, lab technicians and non-teaching staff members for their extreme help throughout our project.

Finally, we express our heartful thanks to all my friends who helped in successful completion of this project.

M.MANOKUMAR	(20A55A0216)
T.UPENDRARAO	(19A51A0242)
P.YAMINI	(19A51A0231)
SINTU KUMAR	(19A51A0238)

CONTENTS

ABSTRACT

In this project we learn **IOT Based Home Automation Project using Blynk & Node MCU**. One of the most common & popular hobby projects you will come across the internet is Smart Home Automation Project. By Home Automation we mean controlling lights, fans, entertainment systems, and appliances without a manual switch. It may also include home security such as access control and alarm systems. When connected with the Internet, home devices are an important constituent of the Internet of Things ("IOT").

In this Home Automation System, we will control lights connected to Relay using Blynk Application. The Wi-Fi Module Node MCU will receive commands from the smartphone wirelessly through the internet. To encode the ON/OFF signal and send it to Server and to ESP8266 Board we need the best IOT Platform. So we chose Blynk as no other application can be better than this one. This project requires internet connectivity & can't work without Internet connection.

CHAPTER- 1:

1.1 INTRODUCTION:

In this project we learn how to make IOT Based Home Automation Project using Blynk & Node MCU ESP8266. One of the most common & popular hobby projects you will come across the internet is Smart Home Automation Project. By Home Automation we mean controlling lighting, climate, entertainment systems, and appliances without a manual switch. It may also include home security such as access control and alarm systems. When connected with the Internet, home devices are an important constituent of the Internet of Things ("IOT").

In this Home Automation System, we will control 4 home appliances as T.V, Fan, And Bulb, Motor, Refrigerator connected to Relay using Blynk Application. The Wi-Fi Module Node MCU ESP8266 will Receive commands from the smartphone wirelessly through the internet. To encode the ON/OFF signal and send it to Server and to ESP8266 Board we need the best IOT Platform. So we chose Blynk as no other application can be better than this one. This project requires internet connectivity & can't work without Internet connection.

1.2 COMPONENTS:

- NODE MCU
- RELAY MODULE
- BREAD BOARD
- MALE AND FEMALE PROBES
- 230V AC LAMPS
- MANUAL SWITCHES
- MICRO-USB DATA CABLE
- 5V REGULATED POWER SUPPLY
- ANDROID MOBILE WITH BLYNK SET UP

1.3 CIRCUIT DIAGRAM:

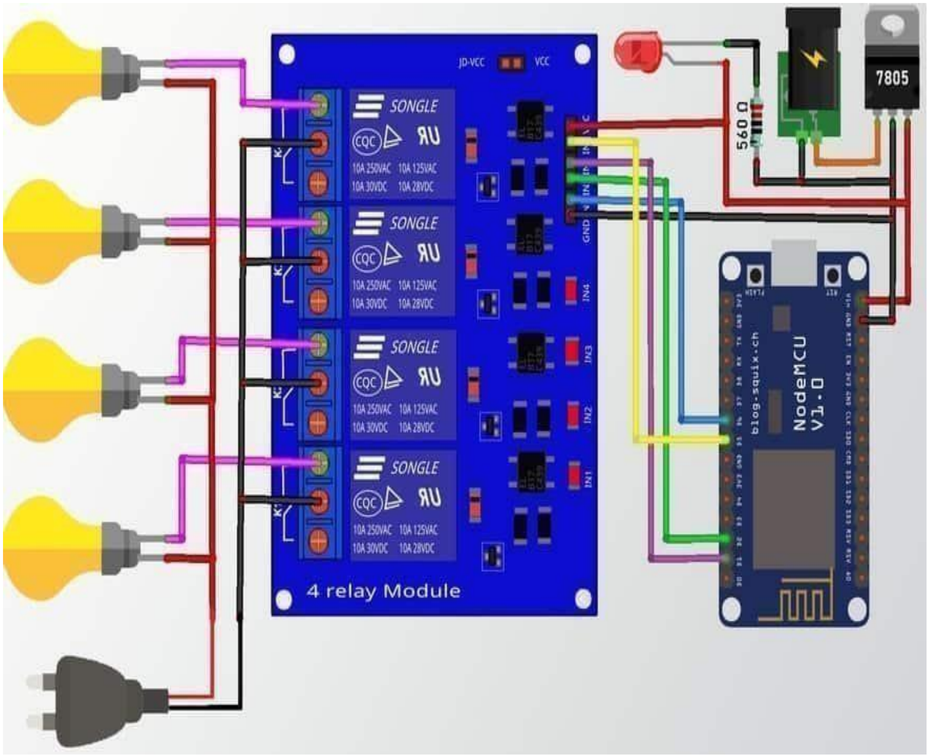


Figure 1.1 . circuit diagram

The circuit diagram for Blynk Controlled Home Automation using ESP8266 is given below. Using this circuit diagram you can assemble the circuit on Breadboard using 4 channel Relay and Node MCU Board.

CHAPTER- 2:

2.1 INTRODUCTION OF NODE MCU:

Node MCU (Node Micro controller Unit) is a low-cost open source IOT platform. Initially included firmware which runs on the ESP 8266 Wi-Fi SOC from Express if Systems, and hardware which was based on the ESP-12 module. Later support for the ESP3232-bit MCU was added.

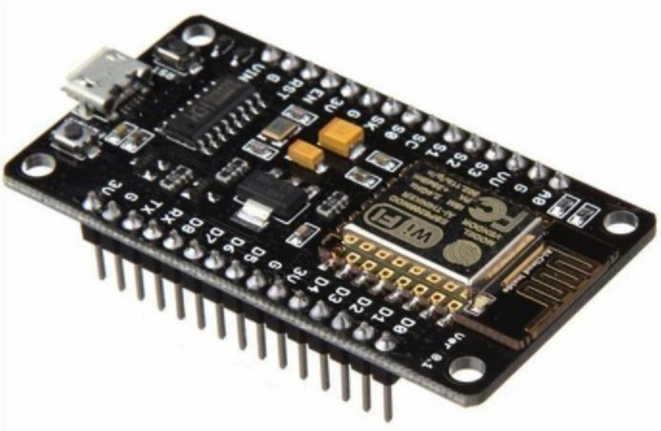


Figure2.1 Node MCU Development Board.

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

Both the firm ware and proto typing board designs are open source.

The firmware uses the Lua based coding language. The firmware is based on the Lua coding, 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.

2.2 PINS OF NODE MCU:

The ESP8266 Node MCU has total 30 pins that interface it to the out side 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.

I2 IC Pins: are used to hookup all sorts of 12C sensors and peripherals in our 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 beset 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 has2 UART interfaces, i.e.UART0 and UART1, which provide a synchronous 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 (TXD1pin) 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 80MHz 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 25MHz 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 1000µs to 10000µs, i.e., between 100Hz and 1kHz.

Control Pins:

To control ESP8266. These pins include Chip Enable pin(EN), Reset pin(RST) and WAKE pin.

- EN pin–The ESP 8266 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 ESP 8266 chip.

.

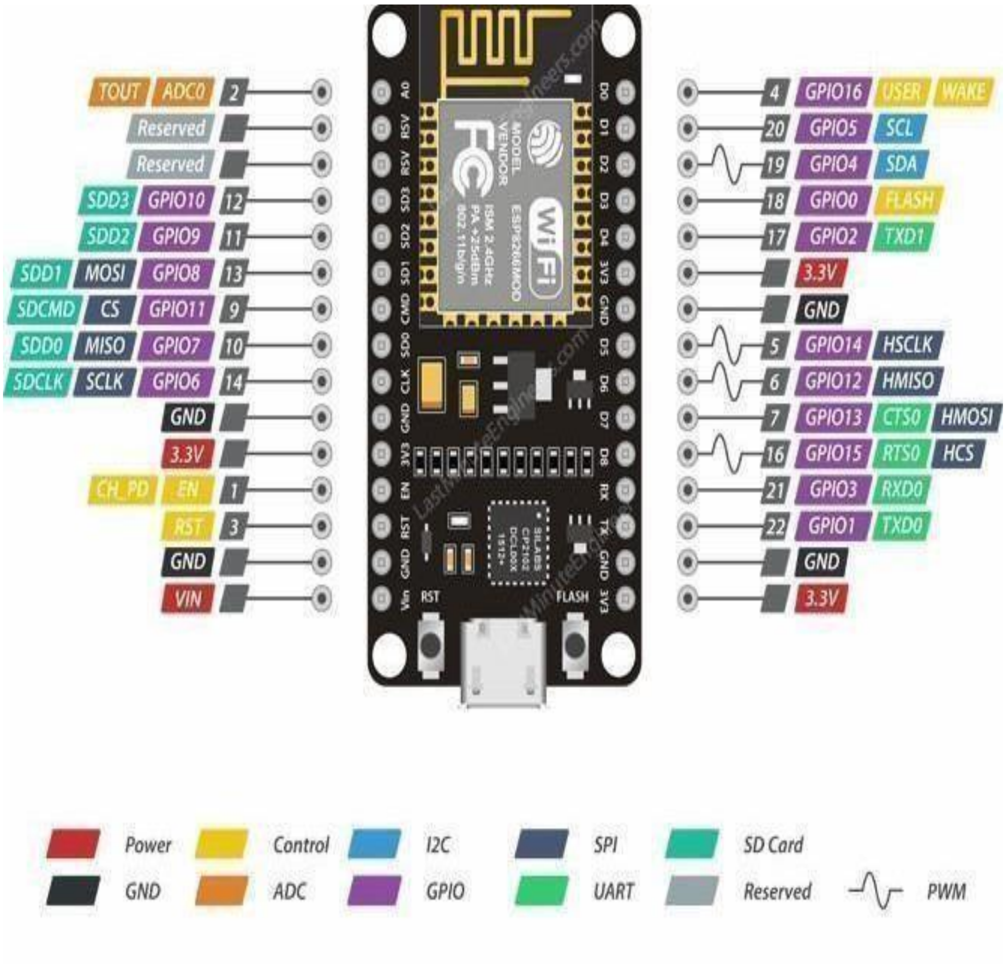


Figure2.2 ESP8266NodeMCUpinout.

2.3 Parts of Node MCU Development Board

The development board equips the ESP-12E module containing ESP8266 chip having Ten silica Xtensa®32-bit LX106RISC microprocessor which operates at 80to160MHz adjustable clock frequency and supports RTOS.

There’s also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that makeup webpages, JSON/XML data, and everything we throw at IOT devices now a days.

WHY NODE MCU ?

Before talking about Node MCU, I want to specify you why we have Chosen Node MCU for this project in spite using other development boards.

We have many development board's like Arduino, Raspberry etc. are Common choices while implementing a new lot ideas After getting programmed they has to Connect to those development boards, the external Wi-Fi Module, because IOT, stands for "Internet of things", the development board needs a way to Connect to the internet.

All the other development boards does not have a inbuilt Support for wireless network. Designers should have to connect a Wi-Fi Module externally.

But in Node Mcu, the most common feature is that it has an inbuilt support for Wi-Fi connectivity and hence IOT applications are done much easier , so that we have choose Node Mcu development board for our project.

Power Requirements:

As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labelled as 3V3. This pin can be used to supply power to external components.

Power to the ESP8266 Node MCU is supplied via the on-board Micro USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals.

- Operatingvoltage2.5Vto3.6V
- On-board3.6V600mAregulator
- 80mAoperatingcurrent
- 20µAduringsleepmode

On Board Switches and LED Indicators:

The ESP8266 Node MCU features two buttons. One marked as RST located on the top left corner is the Reset button, used of course to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware. The board also has a LED indicator which is user programmable and is connected to the D0 pin of the board.

Serial Communication:

The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip

.

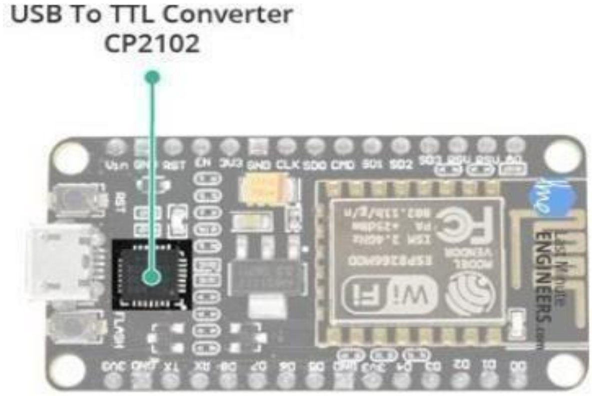


Figure2.3. CP2120 on Node MCU development board

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

CHAPTER-3

3.1 BLOCK DIAGRAM:

Block diagram of the proposed system

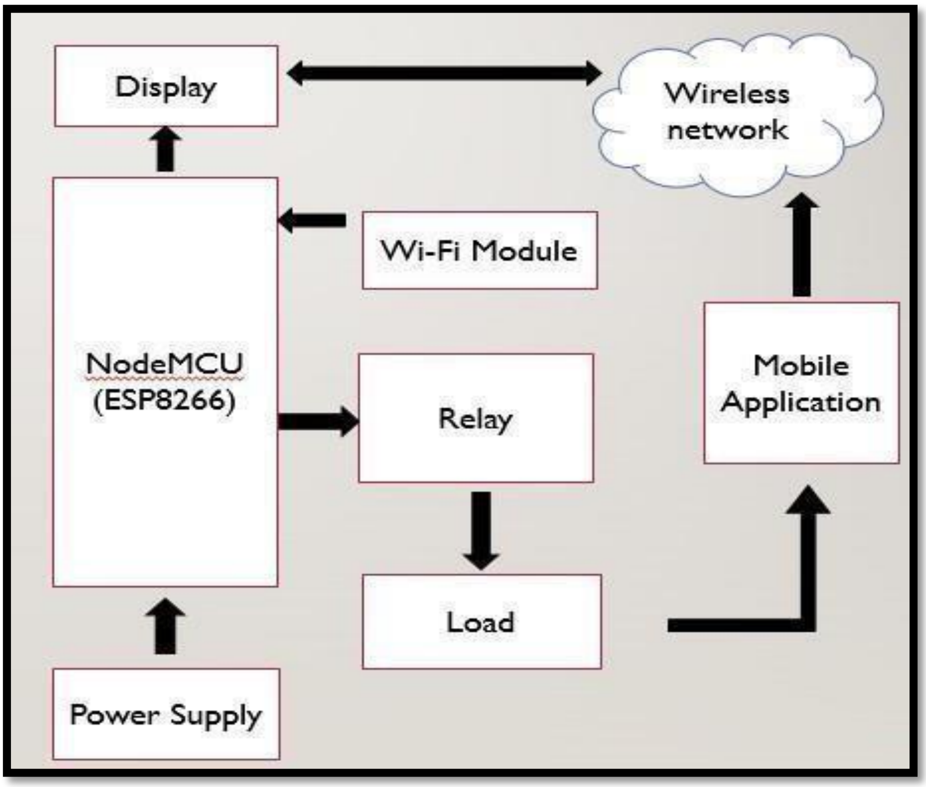


Figure 3.1 Block diagram of proposed system.

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. There is also a display unit that displays the status of the application.

3.2 Proposed system:

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

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

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

3.3 CIRCUIT DIAGRAM:

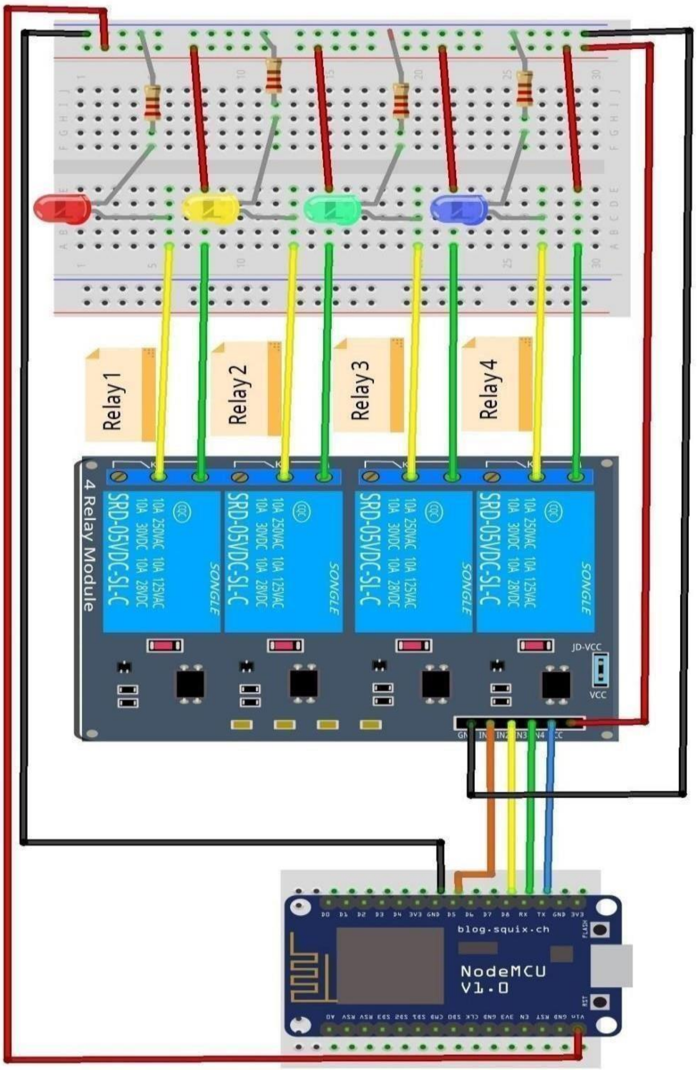


Figure 3.2 Connection diagram of Node MCU controlling 4 channel relay module.

CHAPTER-4

4.1 HARWARE MODELLING AND SETUP

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 customized by adding a name and selecting the digital pin it will correspond to. This section will actually affect the hardware connection as the relays will be physically connected to the digital pins corresponded here.
- The setup of Blynk application is now complete.

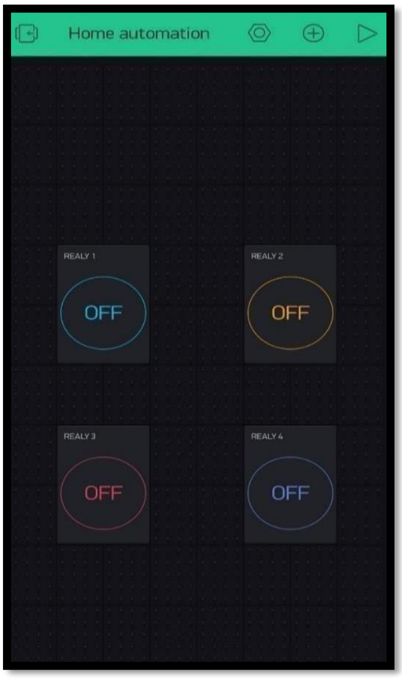


Figure 4.1 blynk app

4.2 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.

4.3 PROJECT LAYOUT:

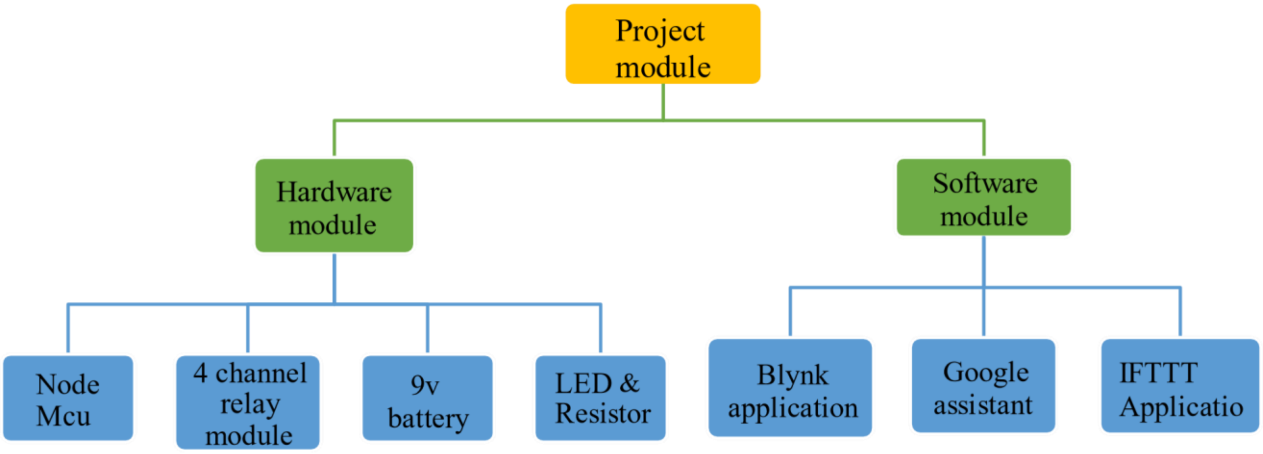


Figure 4.2 Layout of project module

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 primarily uses Blynk application to sense commands from user to the hardware over wireless network

4.4 COMPONENTS REQIRED

1. NODE MCU------1
2. 4 CHANNEL RELAY BOARD---1
3. 9V BATTERY-----1
4. LED-----4
5. 2.2K OHMS RESISTOR----4
6. MALE PIN HEADER-----1
7. FEMALE PIN HEADER—1
8. JUMPER WIRES-------8
9. USB CABLE----------1

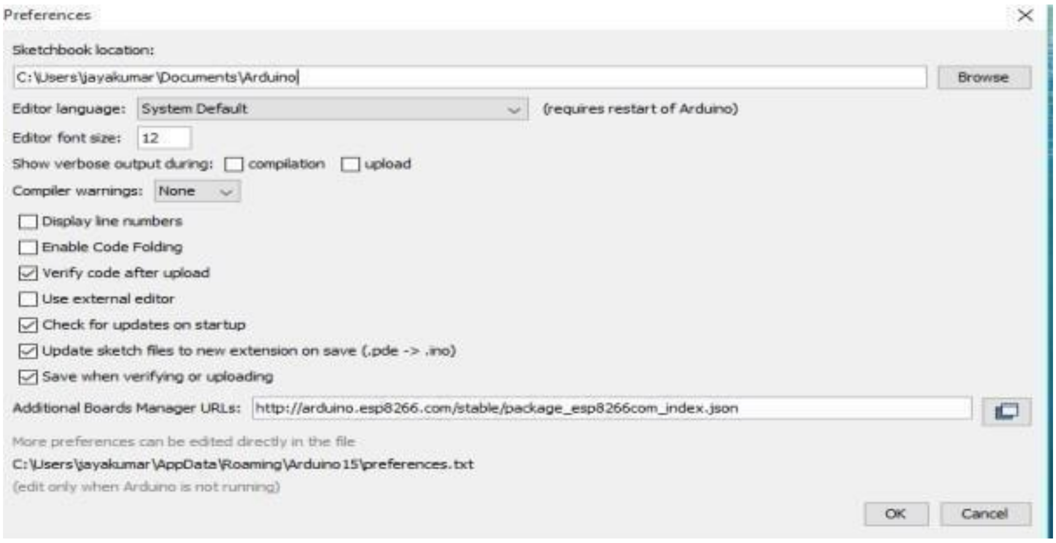
CHAPTER-5

5.1 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.

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*



Sketchbook location:
C:\Users\jayakumar\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
□ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file
C:\Users\jayakumar\AppData\Roaming\Arduino15\preferences.txt
(edit only when Arduino is not running)

Figure 5.1.interfacing of node mcu with ardunio

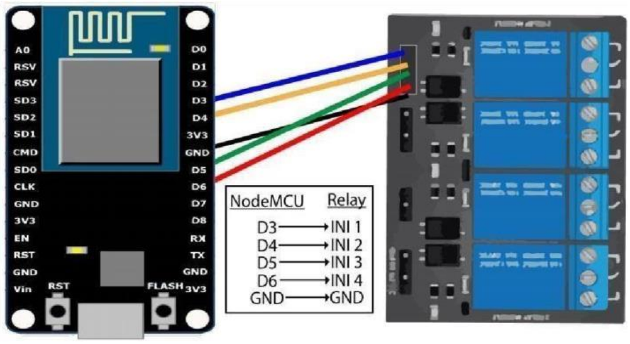
Uploading code to Node MCU

- Node MCU 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 'Node MCU 1.0 (ESP-12E Module)' as the board. And that's all the settings we need to change. So now we begin writing the code.
- Select Files > Examples > Blynk > Boards WI-FI > ESP8266_Standalone. A new file with some prewritten code opens. The following changes to the code are made.

5.2 HARDWARE ASSEMBLY

Hardware assembly mainly includes connecting specific digital pins of Node MCU 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.



NodeMCU	Relay
D3 →	INI 1
D4 →	INI 2
D5 →	INI 3
D6 →	INI 4
GND →	GND

Figure 5.2 Node MCU & 4 channel relay connection.

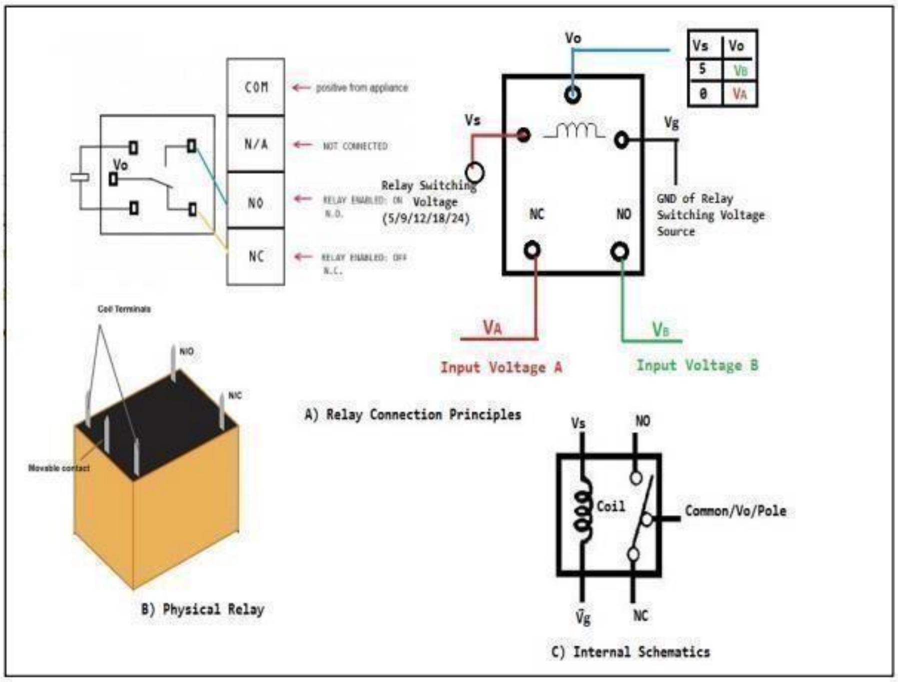
In this prototype instead of real home appliances, we connect the relays to LEDs, (according to circuit diagram) to just ensure the functionality of the prototype. The prototype is given a supply from a 9V battery.

5.3 RELAY MODULE :

The relay’s switch connections are usually labelled COM (POLE), NC and NO:

COM/POLE= Common, NC and NO always connect to this, it is the moving part of the switch. NC = Normally Closed, COM/POLE is connected to this when the relay coil is not magnetized.

NO = Normally Open, COM/POLE is connected to this when the relay coil is MAGNETIZED and vice versa.



Vs	Vo
5	VB
0	VA

Figure 5.3 6V Cube relay.

5.4 ESP8266 NodeMCU WiFi Devkit:

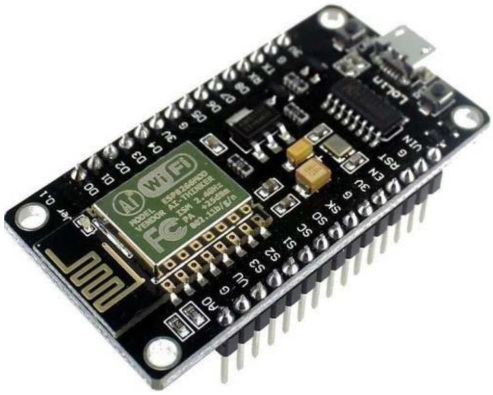


Figure 5.4_Node Mcu WiFi Devkit

The ESP8266 is the name of a micro controller designed by Systems. The ESP8266 itself is a self-contained Wi-Fi networking solution offering as a bridge from existing micro controller to Wi-Fi and is also capable of running self-contained applications.

This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect Node-MCU development kit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

Specification:

- Voltage: 3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Ten silica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.

5.5 Using Arduino IDE:

The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. What we recommend is using the very cool Arduino ESP8266 project, which is a modified version of the Arduino IDE that you need to install on your computer. This makes it very convenient to use the ESP8266 chip as we will be using the well-known Arduino IDE. Following the below step to install ESP8266 library to work in Arduino IDE environment.

Install the Arduino IDE 1.6.4 or greater

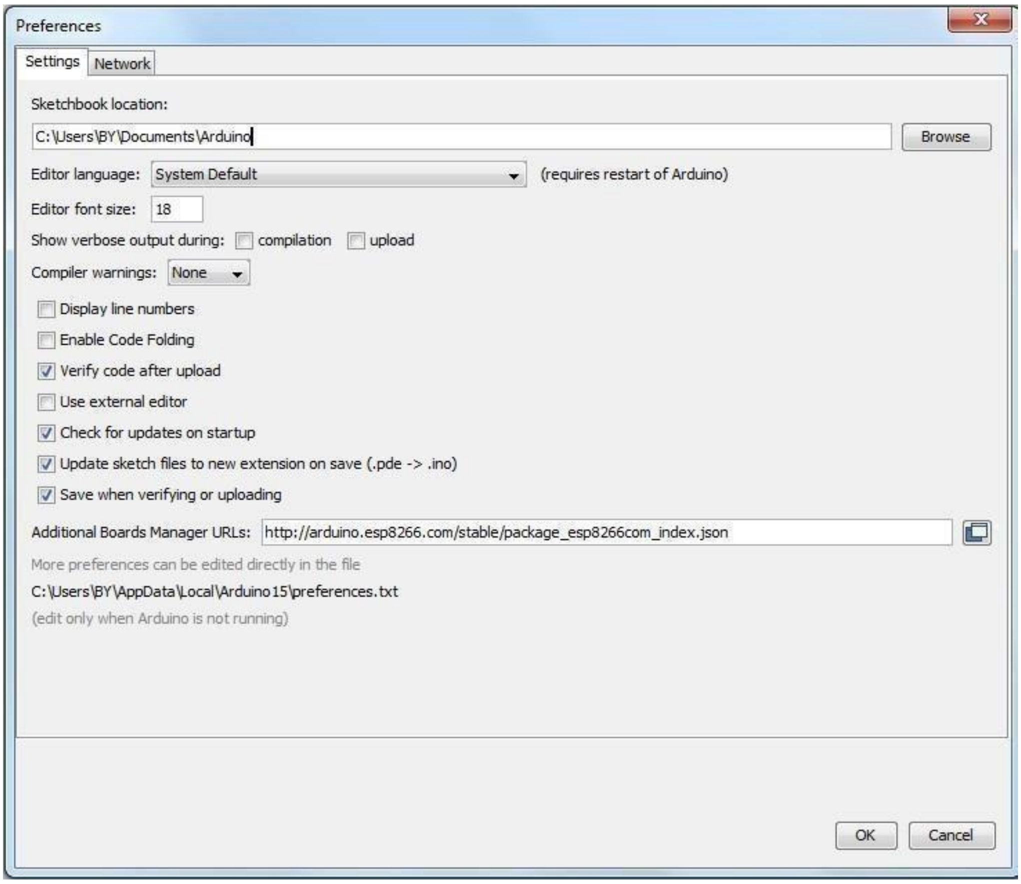
Download Arduino IDE from Arduino.cc (1.6.4 or greater) - don't use 1.6.2 or lower version! You can use your existing IDE if you have already installed it.

You can also try downloading the ready-to-go package from the ESP8266-Arduino project, if the proxy is giving you problems.

Install the ESP8266 Board Package

Enter ***http://arduino.esp8266.com/stable/package_esp8266com_index.json*** into *Additional Board Manager URLs*

field in the Arduino v1.6.4+ preferences.



Sketchbook location: C:\Users\BY\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 18

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers

□ Enable Code Folding

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

C:\Users\BY\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

Figure 5.5 Click ‘File’ -> ‘Preferences’ to access this panel.

Next, use the Board manager to install the ESP8266 package.

Scroll down to ‘ esp8266 by ESP8266 Community ’ and click “Install” button to install the ESP8266 library package. Once installation completed, close and re-open Arduino IDE for ESP8266 library to take effect.

Setup ESP8266 Support

When you've restarted Arduino IDE, select ‘Generic ESP8266 Module’ from the ‘Tools’ -> ‘Board:’ dropdown menu. Select 80 MHz as the CPU frequency (you can try 160 MHz)

5.6 4 Channel 5V Optical Isolated Relay Module

This is a LOW Level 5V 4-channel relay interface board, and each channel needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.

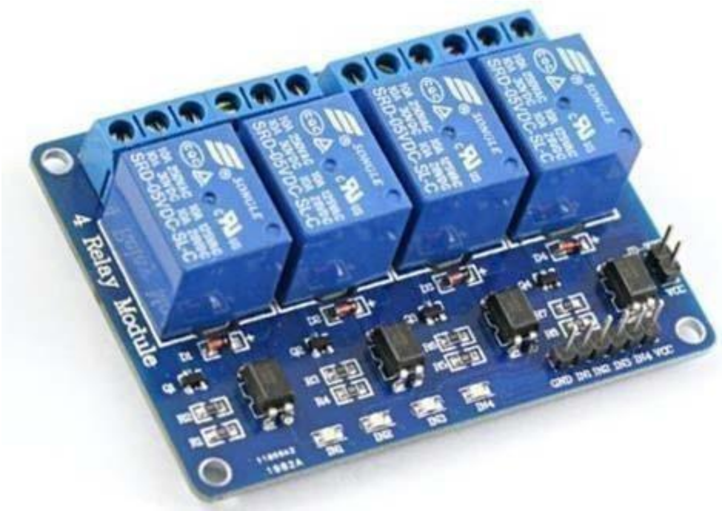


Figure 5.6 4 Channel 5V Optical Isolated Relay Module

Brief Data:

Relay Maximum output: DC 30V/10A, AC 250V/10A.

- 4 Channel Relay Module with Optocoupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.
- Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- OptoCoupler isolation, for high voltage safety and prevent ground loop with microcontroller.

Module Layout:

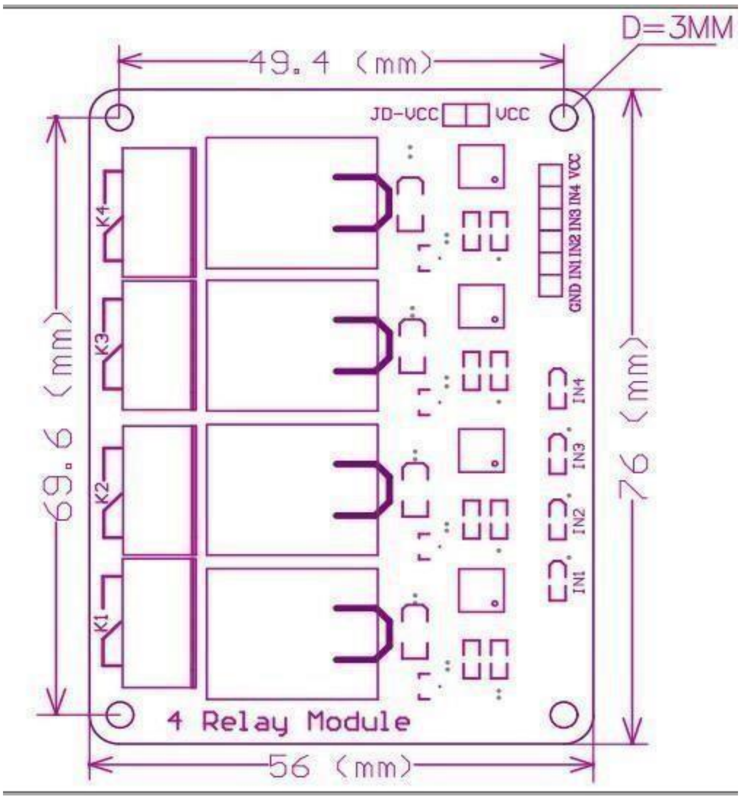
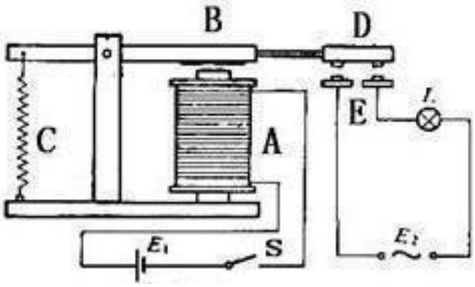


figure 5.7 relay module

Operating Principle

See the picture below: A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.



Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

Input:

VCC : Connected to positive supply voltage (supply power according to relay voltage)

GND : Connected to supply ground.

IN1: Signal triggering terminal 1 of relay

IN2: Signal triggering terminal 2 of relay

IN3: Signal triggering terminal 3 of relay

IN4: Signal triggering terminal 4 of relay

Output:

Each module of the relay has one NC (normally close), one NO (normally open) and one COM (Common) terminal. So there are 4 NC, 4 NO and 4 COM of the channel relay in total. NC stands for the normal close port contact and the state without power. NO stands for the normal open port contact and the state with power.

COM means the common port. You can choose NC port or NO port according to whether power or not.

CHAPTER -6:

6.1 ADVANTAGES:

- Simplest method to automate home appliances.
- Helps physically challenged in their daily activities.
- Electrical and electronic appliances can be controlled any where from the world.

6.2 DISADVANTAGES:

- Security issues
- Cost: Extremely expensive.
- Human errors.

6.3 APPLICATIONS:

- Lighting control
- Lawn/Gardening management
- Smart Home Appliances Improved Home safety and security
- Home air quality and water quality monitoring
- Natural Language-based voice assistants
- Al-driven digital experiences
- Smart Switches
- Smart Locks
- Smart Energy Meters

CHAPTER-7

CONCLUSION:

The home automation system are economical in cost and flexible and programmable. It supports wide variety of peripherals and accessories. And all the home appliances were controlled from a wireless mobile device.

The systems can be accessed from any internet based device including handheld devices such as mobile phones.

REFERENCES:

1. Sam Polniak, “RFID case study book: RFID application stories from around the globe” Abhisam software, 2007.

2. S. P. Singh, M. Mccartney, J. Singh, And R. Clarke, “RFID Research And Testing For Packages Of Apparel, Consumer Goods, 2007.

3. Packaging Technology and Science, Vol. 21,pp. 91–102, 2008.

4. D. Parkash, M.V. Kartikeyan, “Planer antennas for passive. ” Progress in Electromagnetics Research, Vol 91, pp. 95– 212, 2009.

5. Kamran Ahsan, Hanifa Shah and Paul Kingston, “RFID Applications: An Introductory and Exploratory Study,” IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 3, January 2010.