IOT BASED HOME AUTOMATION SYSTEM

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

By

Mandira Das (11701616052)

Pritam Das (11701617009)

Sandip Das (11701617006)

Esha Biswas (11701616058)

Under the guidance of

Prof. Dipankar Santra

Associate Professor

Department of Electrical Engineering

Department of Electrical Engineering

RCC INSTITUTE OF INFORMATION TECHNOLOGY

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

Maulana Abul Kalam Azad Institute of Information Technology

(formerly WBUT)

CERTIFICATE

To whom it may concern

The undersigned certify that they have read and recommended to the Department of Electrical Engineering, RCC Institute of Information Technology, a final year project work entitled "IOT based Home Automation System" submitted by Mandira Das, Pritam Das, Sandip Das, Esha Biswas in partial fulfilment of the requirements for the degree of Bachelor of Technology during the academic year 2019-20 and that this project has not submitted previously for the award of any other degree, diploma and fellowship.

_____

Prof. Dipankar Santra

Associate Professor

Department of Electrical Engineering

RCC Institute of Information Technology

Countersigned by

_____

Dr. Debasish Mondal

Head of the Department,

Department of Electrical Engineering

RCC Institute of Information Technology

ACKNOWLEDGEMENT

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

Mandira Das (11701616052)

Pritam Das (11701617009)

Sandip Das (11701617006)

Esha Biswas (11701616058)

B.Tech (EE) 8th Semester,

2020, RCCIIT

To

The Head of the Department of

Electrical Engineering

RCC Institute of Information Technology

Canal South Rd. Beliagahata, Kolkata-700015

Respected Sir,

In accordance with the requirements of the degree of Bachelor of Technology in the Department of Electrical Engineering, RCC Institute of Information Technology, we present the following report entitled **"IOT Based Home Automation System"**. This work was performed under the valuable guidance of Prof. Dipanakar Santra, Associate Professor in the Dept. of Electrical Engineering.

We declare that the thesis submitted is our own, expected as acknowledge in the test and reference and has not been previously submitted for a degree in any other Institution.

Yours Sincerely,

Mandira Das (11701616052)

Pritam Das (11701617009)

Sandip Das (11701617006)

Esha Biswas (11701616058)

CONTENS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION AND ACRONYMS

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

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.

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 PROJECT MANAGEMENT

Management of any project can be briefly disintegrated into several phases. Our project has been decomposed into the following phases:

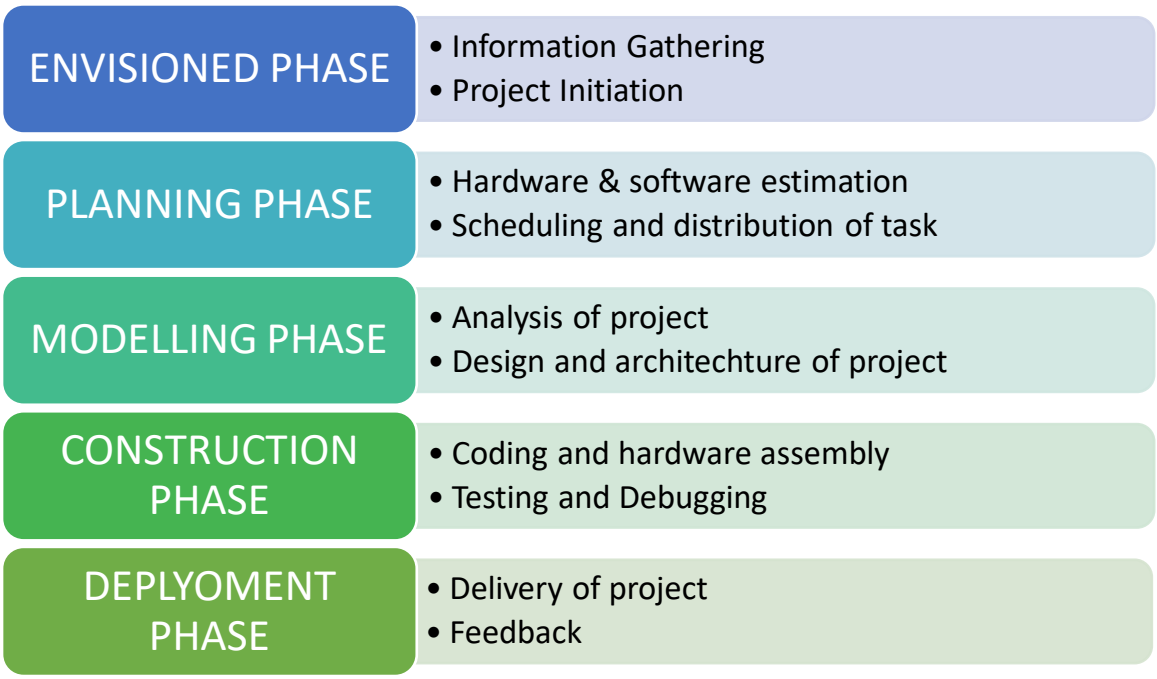


Figure 1. Model of phases in project management.

Experimentation

This phase involved discussions regarding necessary equipment regarding the project. The study of related already existing projects, gathering required theoretical learning. It also included figuring out the coding part, by developing simple algorithms and flowcharts to design the whole process

Design

This phase was, designing layout of the application, and the necessary features to be included. This involved the complete hardware assembly and installing the code to Node MCU. The power strip was designed to connect the home appliances that can be controlled via GPIO pins.

Development and testing

This phase had the development of the application. The android device was connected to the Node MCU via wireless network (WiFi) and the whole prototype was tested for identification and removal of bugs.

Real world testing

The prototype was ready to be tested into the real world and integrated with various real time electrical appliances.

1.6 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.

1.7 ORGANIZATION OF THESIS

The thesis is organised into seven chapters including the introduction. Each chapter is unique on its own and is described with necessary theory to comprehend it.

Chapter 2 deals with *Literature Review*, this chapter reflects a comprehended form of the existing projects related to the topic. It credits the projects along with a brief paragraph of summery about the project. This reflects the various people worked on this area, how different and advanced each project is from one another.

Chapter 3 has the *Theory* that has been acquired to commence the project work. This discussed about IOT, the advantages, disadvantages the network topologies and communication protocols. This chapter also briefs about the main microcontroller unit of the prototype, Node MCU. Its pin configuration, various functional units of the development board and the installation process of the device. The chapter further giver a brief overview of the project, a block diagram of the system and the circuit diagram.

Chapter 4 describes the *Hardware Modelling and setup* of the project. The chapter points the main features of the prototype, gives a layout of the project, lists the components requires. It briefly describes the various setup processes involved with the project, including hardware

interfacing and software installation and setup according to our requirement. It finally gives the hardware assembly involved.

Chapter 5 is the *Logic and operation* of the project. A flow chart presents the actions describe the working process of the prototype. It discussed the principle of operation of the system with the advantages and disadvantages of the microcontroller unit. It describes Blynk and IFTTT application and the wireless network established to attain remote control over the system. It describes the process of voice control mode and gives an overall cost estimation of the project.

Chapter 6 is the *conclusion and Future scope.* This chapter includes the result of the project work carried, the limitations it possesses, the further enhancements and modification that can be integrated into the prototype and finally concludes the project work carried so far.

Chapter 7 lists the *References* that have been used for the commencement of the project work.

Appendix A & B individual *hardware description* of the prototype and associated *data sheets.*

CHAPTER 2
LIERATURE REVIEW

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

This paper presents a step-by-step procedure of a smart home automation controller. It uses IOT to convert home appliances to smart and intelligent devices, with the help of design control. An energy efficient system is designed that accesses the smart home remotely using IOT connectivity. The proposed system mainly requires, Node MCU as the microcontroller unit, IFTTT to interpret voice commands, Adafruit a library that supports MQTT acts as an MQTT broker and Arduino IDE to code the microcontroller. This multimodal system uses Google Assistant along with a web based application to control the smart home. The smart home is implemented with main controller unit that is connected with the 24-hour available Wi-Fi network. To ensure, that the Wi-Fi connection do not turn off, the main controller is programmed to establish automatic connection with the available network and connected to the auto power backup.

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

This paper focuses on a system that provides features of Home Automation relying on IOT to operate easily, in addition to that it includes a camera module and provides home security. The android application basically converts Smartphone into a remote for all home appliances. Security is achieved with motion sensors if movement is sensed at the entrance of the house; a notification is sent that contains a photo of house entrance in real time. This notification will be received by the owner of the house via internet such that app can trigger a notification. So owner can raise an alarm in case of any intrusion or he/she can toggle the appliances like opening the door if the person is a guest. The system uses Raspberry Pi, a small sized computer which acts as server for the system. The smart home consist two modules. Home automation that consists; fan light and door controller, and security module that consists; smoke sensor motion sensor and camera module.

"A Dynamic Distributed Energy Management Algorithm of Home Sensor Network for Home Automation System", by **Tui-Yi Yang, Chu-Sing Yang, Tien-Wen Sung.**

This paper proposes an optimization of home power consumption based on PLC (Power Line Communication) for an easy to access home energy consumption. This also proposes a Zigbee and PLC based renewable energy gateway to monitor the energy generation of renewable energies. ACS and DDEM algorithm are proposed for the design of an intelligent distribution of power management system to make sure ongoing power supply of home networks. To provide efficient power management the power supply models of home sensor network are classified groups viz. main supply only, main supply and backup battery, rechargeable battery power and non-rechargeable battery power. Devices with particular features are assigned to these groups. It targets to establish real time processing scheme to address variable sensor network topologies.

"Enhance Smart Home Automation System based on Internet of Things", by **Tushar Churasia and Prashant Kumar Jain.**

This paper proposes a system that develops a model to reduce the computation overhead in existing smart home solutions that uses various encryption technologies like AES, ECHD, hybrid, etc. these solutions use intermediate gateway for connecting various sensor devices. The proposed model provides a method for automation with sensor based learning. The system uses temperature sensor for development but other sensors can also be used as per requirement. These smart home devices with sensors can configure themselves autonomously and can operate without human intervention. This work minimizes encryption decryption and focuses on authentication and automation of smart home devices with learning. The system bypasses local gateway mentioned in existing system to provide better security for smart home devices and sensor data and save computation overhead. The real time broker cloud is directly connected with smart home and manages all incoming and outgoing request between users and devices. The main purpose to use real time broker cloud is save time of cryptographic operations.

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

The paper present a vision-based machine intelligence system to sense on/off state of common home appliance. The proposed method of sensing the state of appliances results on a novel home automation system. The accessibility of the suite of devices in the home over a remote network is facilitated by the IP Addressing methods in the IOT. This project uses two boards viz. Raspberry Pi and Intel Galileo Gen 2. The communication between the User devices, Raspberry Pi and the Intel Galileo boards happens over a wireless network. The UDP protocol is deployed to facilitate the wireless communication of the nodes present in the home automation network. A Pi Cam and a USB Logitech camera attached to the rotating shaft of two different servo motor capture snapshots that are passed as inputs to the Machine Learning based models trained using dlib-C++ to detect the state of the operation of the appliances. The proposed method uses visual modality to automate the appliances, as privacy concerns may emerge while using the images from some specific places, as a counter to this issue, an SPDT switch is added to the Raspberry Pi which when turned off ensures that even if the images are taken from the webcams, they are just passed as inputs to the machine learning models and are not displayed on the website when the users access the website on the server address obtained from Raspberry Pi.

"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.**

This paper illustrates a methodology to provide a low cost Home Automation System (HAS) using Wireless Fidelity (Wi-Fi).This crystallizes the concept of internetworking of smart devices. A Wi-Fi based Wireless Sensor Network (WSN) is designed for the purpose of monitoring and controlling environmental, safety and electrical parameters of a smart interconnected home. The different

sections of the HAS are; temperature and humidity sensor, gas leakage warning system, fire alarm system, burglar alarm system, rain sensing, switching and regulation of load & voltage and current sensing. The primary requirement of HAS to monitor and control of devices is accomplished using a Smartphone application. The application is developed using Android Studio based on JAVA platform and User Interface of those are exemplified. The primary focus of the paper is to develop a solution cost effective flexible in control of devices and implementing a wide range of sensors to to capture various parameters.

"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.**

The paper focuses on the construction of a fully functional voice based Home automation system that uses Internet of Things, Artificial Intelligence and Natural Language Processing (NLP) to provide a cost-effective, efficient way to work together with home appliances using various technologies such as GSM, NFC, etc. it implements a seamless integration of all the appliances to a central console, i.e. the mobile device. The prototype uses Arduino MK1000, known as Genuino MK1000. The NLP in this project gives the user the freedom to interact with the home appliances with his/her own voice and normal language rather than complicated computer commands. The appliances are connected to the mobile device through an Arduino Board that establishes the concept of Internet of Things. The Arduino Boards are interfaced with the appliances and programmed in such a way that they respond to mobile inputs.

CHAPTER 3

THEORY

3.1 IOT (INTERNET OF THINGS)

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

3.1.1 Features of IOT

3.1.1.1 Intelligence

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

3.1.1.2 Connectivity

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

3.1.1.3 Dynamic Nature

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

3.1.1.4 Enormous Scale

The number of devices that need to be managed and that communicate with each other will be much larger than the devices connected to the current Internet. The management of data generated from these devices and their interpretation for application purposes becomes more critical. Gartner (2015) confirms the enormous scale of IOT in the estimated report where it stated that 5.5 million new things will get connected every day and 6.4 billion connected things will be in

use worldwide in 2016, which is up by 30 percent from 2015. The report also forecasts that the number of connected devices will reach 20.8 billion by 2020

3.1.1.5 Sensing

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

3.1.1.6 Heterogeneity

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

3.1.1.7 Security

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

3.1.2 Advantages of IOT

3.1.2.1 Communication

IOT encourages the communication between devices, also famously known as Machine-to-Machine (M2M) communication. Because of this, the physical devices are able to stay connected and hence the total transparency is available with lesser inefficiencies and greater quality.

3.1.2.2 Automation and Control

Due to physical objects getting connected and controlled digitally and centrally with wireless infrastructure, there is a large amount of automation and control in the workings. Without human intervention, the machines are able to communicate with each other leading to faster and timely output.

3.1.2.3 Information

It is obvious that having more information helps making better decisions. Whether it is mundane decisions as needing to know what to buy at the grocery store or if your company has enough widgets and supplies, knowledge is power and more knowledge is better.

3.1.2.4 Monitor

The second most obvious advantage of IOT is monitoring. Knowing the exact quantity of supplies or the air quality in your home, can further provide more information that could not have previously been collected easily. For instance, knowing that you are low on milk or printer ink could save you another trip to the store in the near future. Furthermore, monitoring the expiration of products can and will improve safety.

3.1.2.5 Time

As hinted in the previous examples, the amount of time saved because of IOT could be quite large. And in today's modern life, we all could use more time.

3.1.2.6 Money

The biggest advantage of IOT is saving money. If the price of the tagging and monitoring equipment is less than the amount of money saved, then the Internet of Things will be very widely adopted. IOT fundamentally proves to be very helpful to people in their daily routines by making the appliances communicate to each other in an effective manner thereby saving and conserving energy and cost. Allowing the data to be communicated and shared between devices and then translating it into our required way, it makes our systems efficient.

3.1.2.7 Automation of daily tasks leads to better monitoring of devices

The IOT allows you to automate and control the tasks that are done on a daily basis, avoiding human intervention. Machine-to-machine communication helps to maintain transparency in the processes. It also leads to uniformity in the tasks. It can also maintain the quality of service. We can also take necessary action in case of emergencies.

3.1.2.8 Efficient and Saves Time

The machine-to-machine interaction provides better efficiency, hence; accurate results can be obtained fast. This results in saving valuable time. Instead of repeating the same tasks every day, it enables people to do other creative jobs.

3.1.2.9 Saves Money

Optimum utilization of energy and resources can be achieved by adopting this technology and keeping the devices under surveillance. We can be alerted in case of possible bottlenecks, breakdowns, and damages to the system. Hence, we can save money by using this technology.

3.1.2.10 Better Quality of Life

All the applications of this technology culminate in increased comfort, convenience, and better management, thereby improving the quality of life.

3.1.3 Disadvantages of IOT

3.1.3.1 Compatibility

Currently, there is no international standard of compatibility for the tagging and monitoring equipment. I believe this disadvantage is the most easy to overcome. The manufacturing companies of these equipment just need to agree to a standard, such as Bluetooth, USB, etc. This is nothing new or innovative needed.

3.1.3.2 Complexity

As with all complex systems, there are more opportunities of failure. With the Internet of Things, failures could sky rocket. For instance, let's say that both you and your spouse each get a message saying that your milk has expired, and both of you stop at a store on your way home, and you both purchase milk. As a result, you and your spouse have purchased twice the amount that you both need. Or maybe a bug in the software ends up automatically ordering a new ink cartridge for your printer each and every hour for a few days, or at least after each power failure, when you only need a single replacement.

3.1.3.3 Privacy/Security

With all of this IOT data being transmitted, the risk of losing privacy increases. For instance, how well encrypted will the data be kept and transmitted with? Do you want your neighbours or employers to know what medications that you are taking or your financial situation?

3.1.3.4 Safety

Imagine if a notorious hacker changes your prescription. Or if a store automatically ships you an equivalent product that you are allergic to, or a flavour that you do not like, or a product that is already expired. As a result, safety is ultimately in the hands of the consumer to verify any and all automation.

As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices all are connected to the Internet, a lot of information is available on it. This information is prone to attack by hackers. It would be very disastrous if private and confidential information is accessed by unauthorized intruders.

3.1.3.5 Lesser Employment of Menial Staff

The unskilled workers and helpers may end up losing their jobs in the effect of automation of daily activities. This can lead to unemployment issues in the society. This is a problem with the advent of any technology and can be overcome with education. With daily activities getting automated, naturally, there will be fewer requirements of human resources, primarily, workers and less educated staff. This may create Unemployment issue in the society.

3.1.3.6 Technology Takes Control of Life

Our lives will be increasingly controlled by technology, and will be dependent on it. The younger generation is already addicted to technology for every little thing. We have to decide how much of our daily lives are we willing to mechanize and be controlled by technology.

3.1.4 Application Grounds of IOT

3.1.4.1 Wearables

Wearable technologies is a hallmark of IOT applications and is one of the earliest industries to have deployed IOT at its services. Fit Bits, heart rate monitors, smartwatches, glucose monitoring devices reflect the successful applications of IOT.

3.1.4.2 Smart homes

This area of application concerned to this particular project, so a detailed application is discussed further. *Jarvis*, an AI home automation employed by Mark Zuckerberg, is a remarkable example in this field of application.

3.1.4.3 Health care

IOT applications have turned reactive medical based system into proactive wellness based system. IOT focuses on creating systems rather than equipment. IOT creates a future of medicine and healthcare which exploits a highly integrated network of sophisticated medical devices. The integration of all elements provides more accuracy, more attention to detail, faster reactions to events, and constant improvement while reducing the typical overhead of medical research and organizations

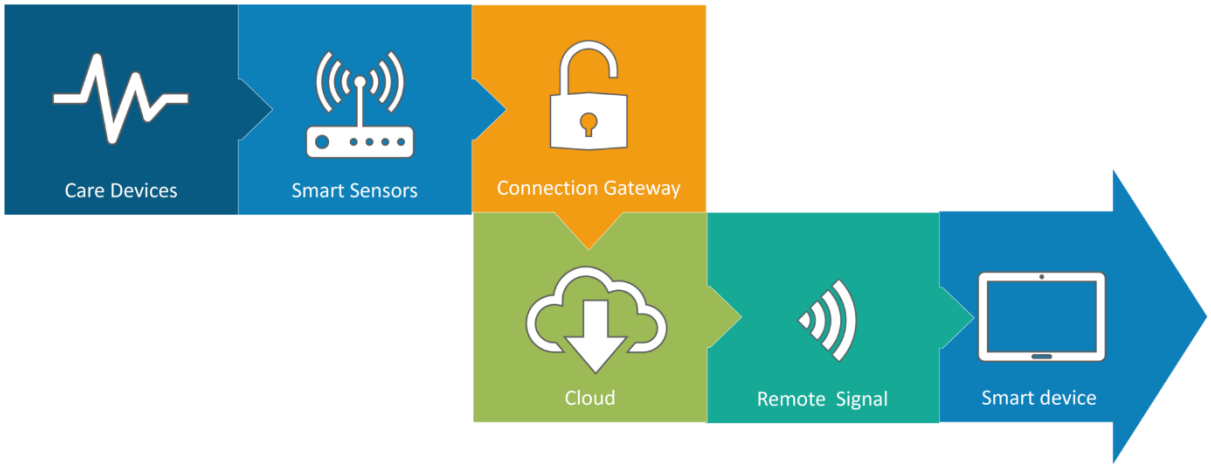


Figure 2. Working of IOT enables care devices.

3.1.4.4 Agriculture

A greenhouse farming technique enhances the yield of crops by controlling environmental parameters. However, manual handling results in production loss, energy loss, and labour cost, making the process less effective. A greenhouse with embedded devices not only makes it easier to be monitored but also, enables us to control the climate inside it. Sensors measure different

parameters according to the plant requirement and send it to the cloud. It, then, processes the data and applies a control action.

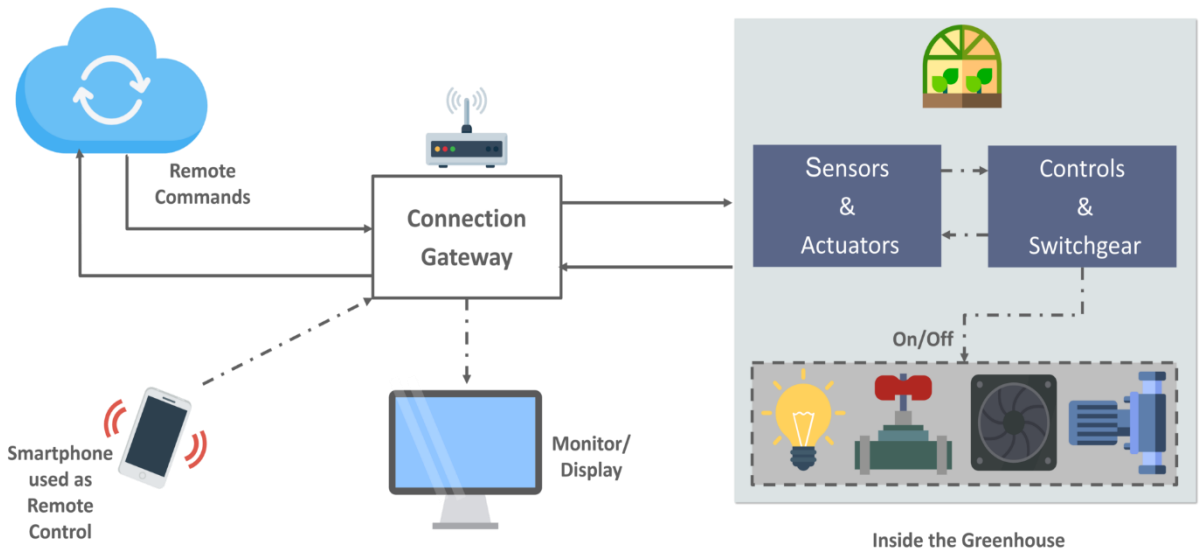


Figure 3. IOT controlled greenhouse environment.

3.1.4.5 Industrial Automation

For a higher return of investment this field requires both fast developments and quality of products. This vitality thus coined the term IIOT. This whole schematic is re-engineered by IOT applications. Following are the domains of IOT applications in industrial automation

- Factory Digitalization
- Product flow Monitoring
- Inventory Management
- Safety and Security
- Quality Control
- Packaging optimization
- Logistics and Supply Chain Optimization

3.1.4.6 Government and Safety

IOT applied to government and safety allows improved law enforcement, defence, city planning, and economic management. The technology fills in the current gaps, corrects many current flaws, and expands the reach of these efforts. For example, IOT can help city planners have a clearer view of the impact of their design, and governments have a better idea of the local economy.

3.1.5 IOT Technologies and Protocols

Several communication protocols and technologies cater to and meet the specific functional requirements of IOT system.

3.1.5.1 Bluetooth

Bluetooth is a short range IOT communication protocol/technology that is profound in many consumer product markets and computing. It is expected to be key for wearable products in particular, again connecting to the IOT albeit probably via a smartphone in many cases. The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now branded – is a significant protocol for IOT applications. Importantly, while it offers a similar range to Bluetooth it has been designed to offer significantly reduced power consumption.

3.1.5.2 Zigbee

ZigBee is similar to Bluetooth and is majorly used in industrial settings. It has some significant advantages in complex systems offering low-power operation, high security, robustness and high and is well positioned to take advantage of wireless control and sensor networks in IOT applications. The latest version of ZigBee is the recently launched 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.

3.1.5.3 Z-Wave

Z-Wave is a low-power RF communications IOT technology that primarily design for home automation for products such as lamp controllers and sensors among many other devices. A Z-Wave uses a simpler protocol than some others, which can enable faster and simpler development, but the only maker of chips is Sigma Designs compared to multiple sources for other wireless technologies such as ZigBee and others.

3.1.5.4 Wi-Fi

Wi-Fi connectivity is one of the most popular IOT communication protocol, often an obvious choice for many developers, especially given the availability of Wi-Fi within the home environment within LANs. There is a wide existing infrastructure as well as offering fast data transfer and the ability to handle high quantities of data. Currently, the most common Wi-Fi standard used in homes and many businesses is 802.11n, which offers range of hundreds of megabit per second, which is fine for file transfers but may be too power-consuming for many IOT applications.

3.1.5.5 Cellular

Any IOT application that requires operation over longer distances can take advantage of GSM/3G/4G cellular communication capabilities. While cellular is clearly capable of sending high quantities of data, especially for 4G, the cost and also power consumption will be too high for many applications. But it can be ideal for sensor-based low-bandwidth-data projects that will send very low amounts of data over the Internet.

3.1.5.6 NFC

NFC (Near Field Communication) is an IOT technology. It enables simple and safe communications between electronic devices, and specifically for smartphones, allowing consumers to perform transactions in which one does not have to be physically present. It helps the user to access digital content and connect electronic devices. Essentially it extends the capability of contactless card technology and enables devices to share information at a distance that is less than 4cm.

3.1.5.7 LoRaWAN

LoRaWAN is one of popular IOT Technology, targets wide-area network (WAN) applications. The LoRaWAN design to provide low-power WANs with features specifically needed to support low-cost mobile secure communication in IOT, smart city, and industrial applications. Specifically meets requirements for low-power consumption and supports large networks with millions and millions of devices, data rates range from 0.3 kbps to 50 kbps.

3.1.6 IOT software

IOT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IOT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

3.1.6.1 Data Collection

This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to-machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

3.1.6.2 Device Integration

Software supporting integration binds (dependent relationships) all system devices to create the body of the IOT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IOT network because without them, it is not an IOT system. They manage the various applications, protocols, and limitations of each device to allow communication.

3.1.6.3 Real-Time Analytics

These applications take data or input from various devices and convert it into feasible actions or clear patterns for human analysis. They analyse information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

3.1.6.4 Application and Process Extension

These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

3.2 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.

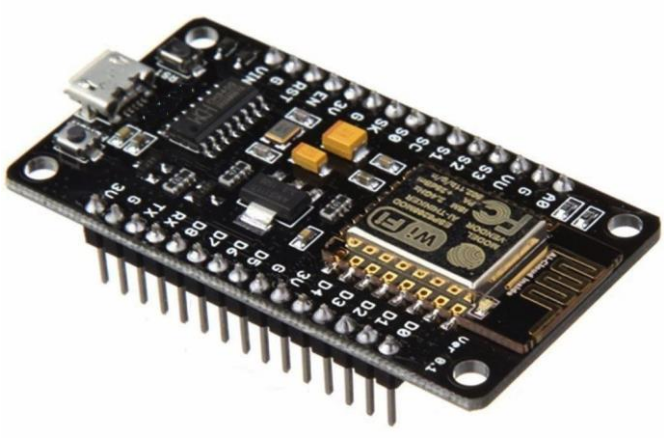


Figure 4. Node MCU Development Board.

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.

3.2.1 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.

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

Table 1. Node MCU index ↔ GPIO mapping.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

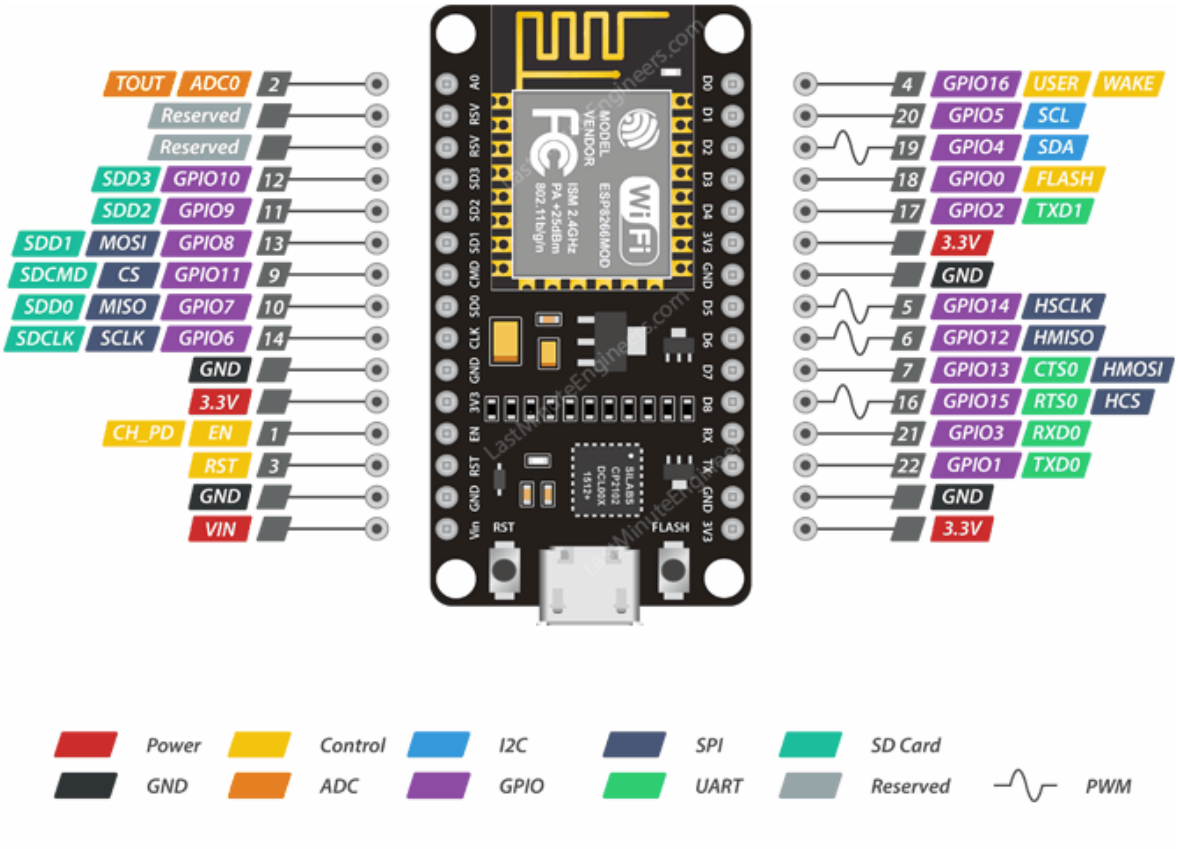


Figure 5. ESP8266 Node MCU pinout.

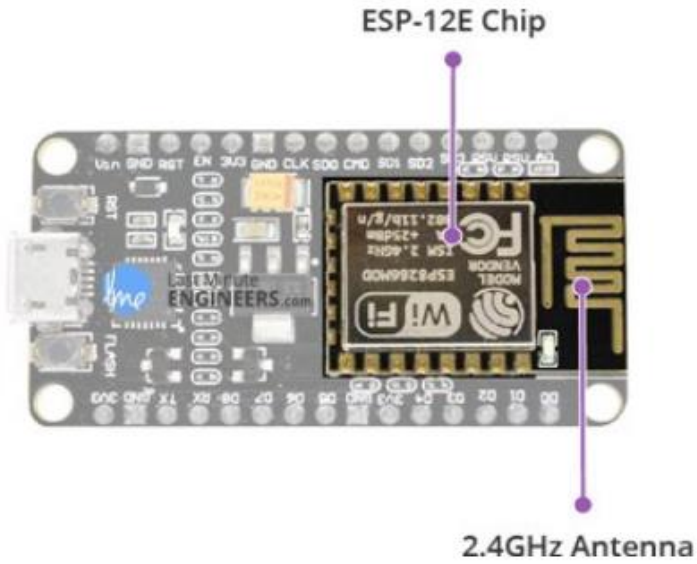
3.2.2 Parts of Node MCU Development Board

3.2.2.1 ESP 12-E Module

The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz 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 make up web pages, JSON/XML data, and everything we throw at IOT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a Wi-Fi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 Node MCU even more versatile.



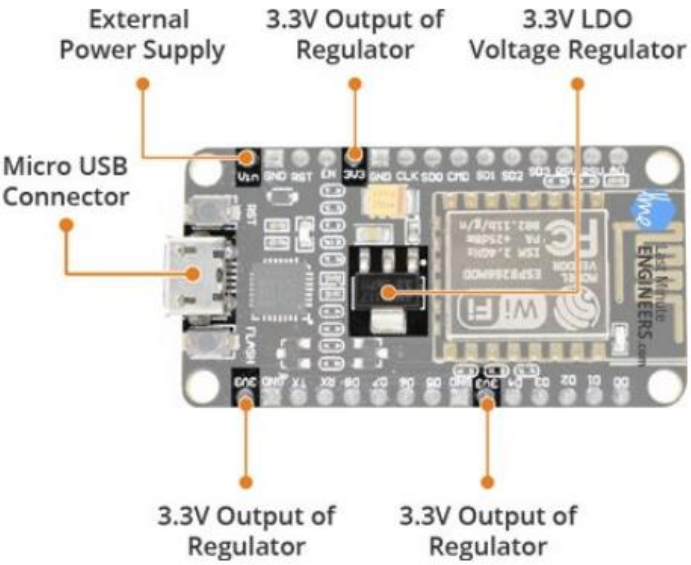
- Tensilica Xtensa® 32-bit LX106
- 80 to 160 MHz clock frequency
- 128 kb internal RAM
- 4 MB external flash
- 802.11b/g/n HT40 Wi-Fi transceiver

Figure 6. ESP 12E module in Node MCU Development board.

3.2.2.2 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 B 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.



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

Figure 7. Power module on a Node MCU development board.

3.2.2.3 Peripheral I/O

The ESP8266 Node MCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

- ADC channel – A 10-bit ADC channel.
- UART interface – UART interface is used to load code serially.
- PWM outputs – PWM pins for dimming LEDs or controlling motors.
- SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- I2S interface – I2S interface if you want to add sound to your project.

As a result of the pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin), a single GPIO pin can act as PWM/UART/SPI.

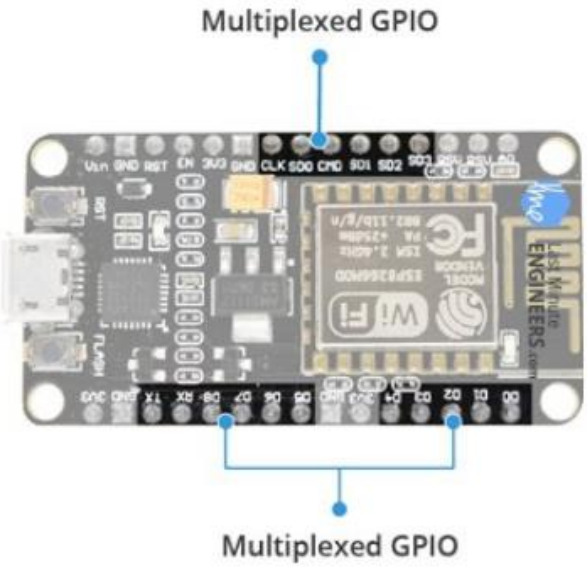
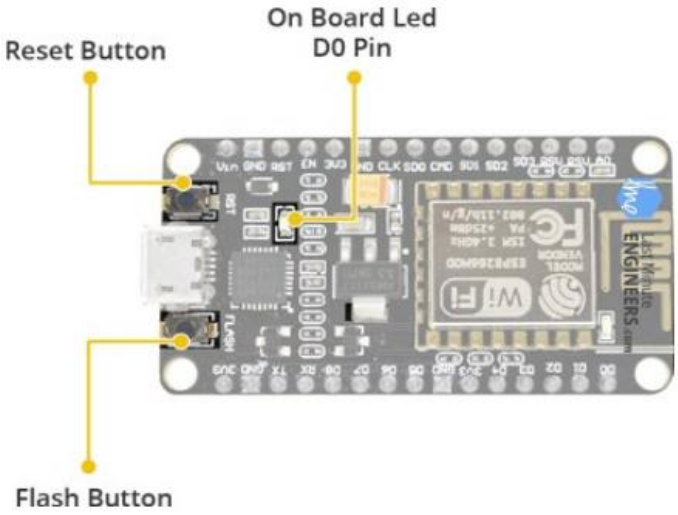


Figure 8. GPIO pins on Node MCU development board.

3.2.2.4 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.



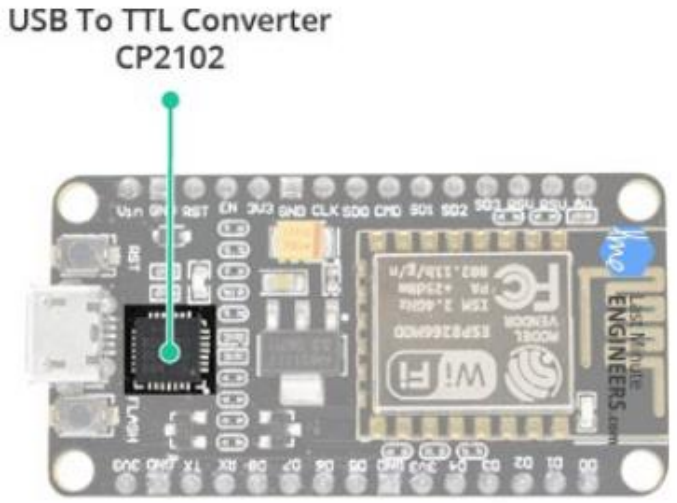
Switches and indicators

- RST: Reset the ESP8266 chip
- FLASH: Download new programs
- Blue LED: User programmable

Figure 9. ON board switches and LED indicators on Node MCU development board.

3.2.2.5 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.



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

Figure 10. CP2120 on Node MCU development board.

3.2.3 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.

3.3 BLOCK DIAGRAM

3.3.1 Block diagram of the proposed system

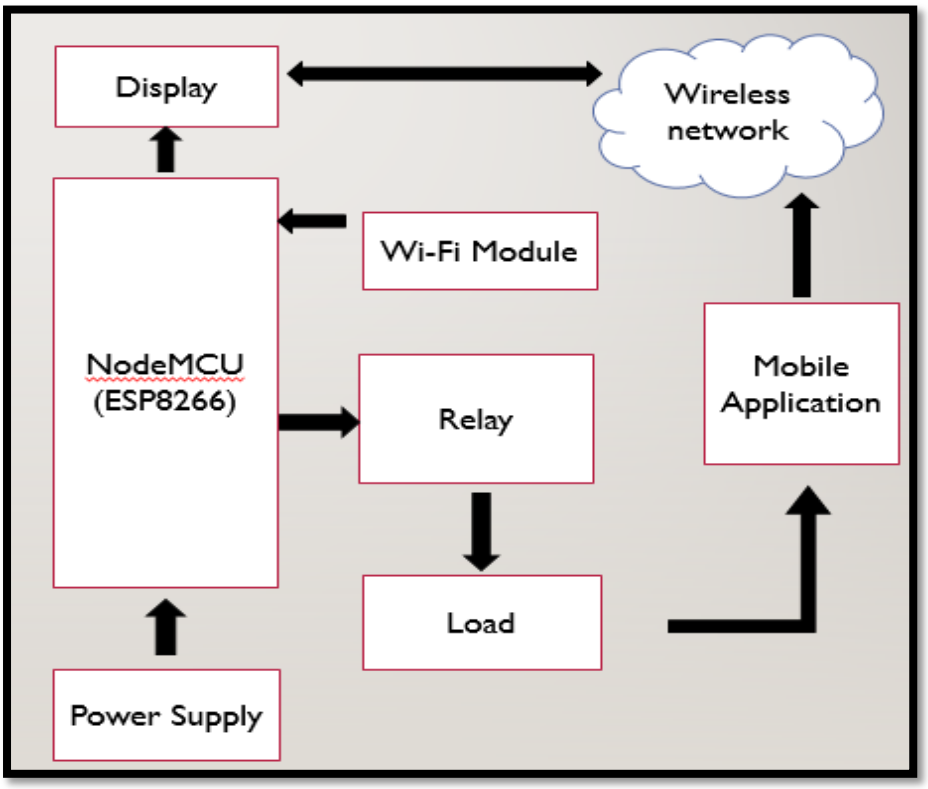


Figure 11. Block diagram of proposed system.

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

3.3.2 Proposed system

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

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

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

3.4 OVERVIEW OF PROJECT

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

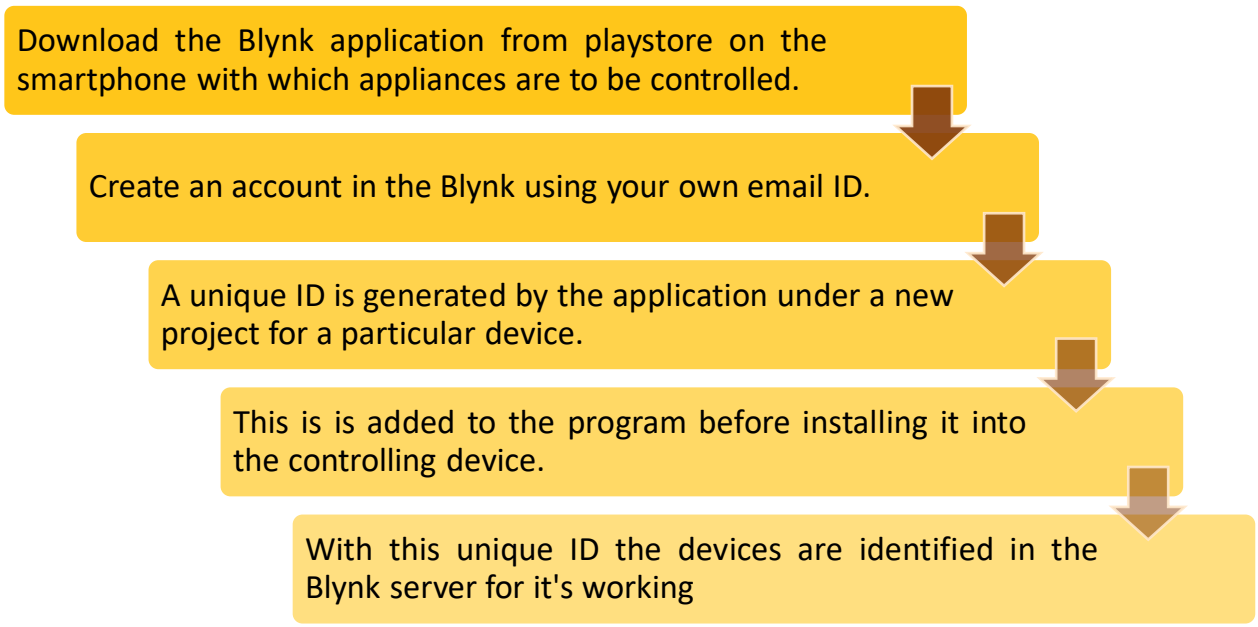


Figure 12. Creating an account and generating unique ID in Blynk Server.

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

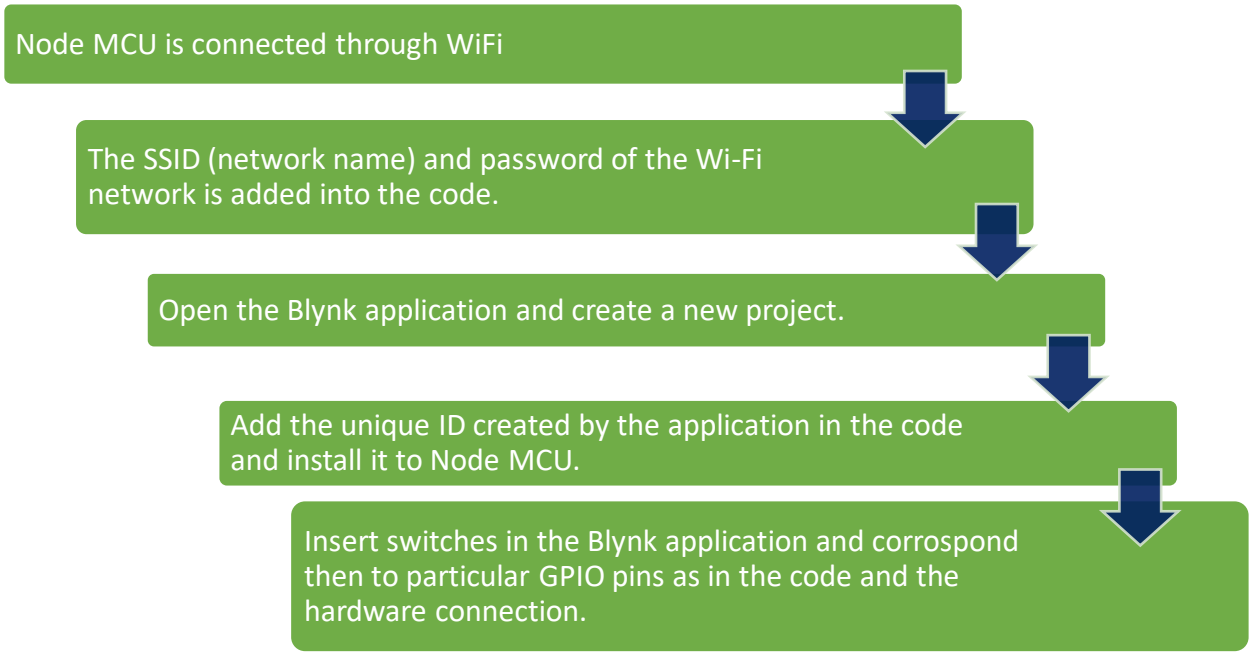


Figure 13. Setup to control Node MCU from Blynk application

3.5 CIRCUIT DIAGRAM

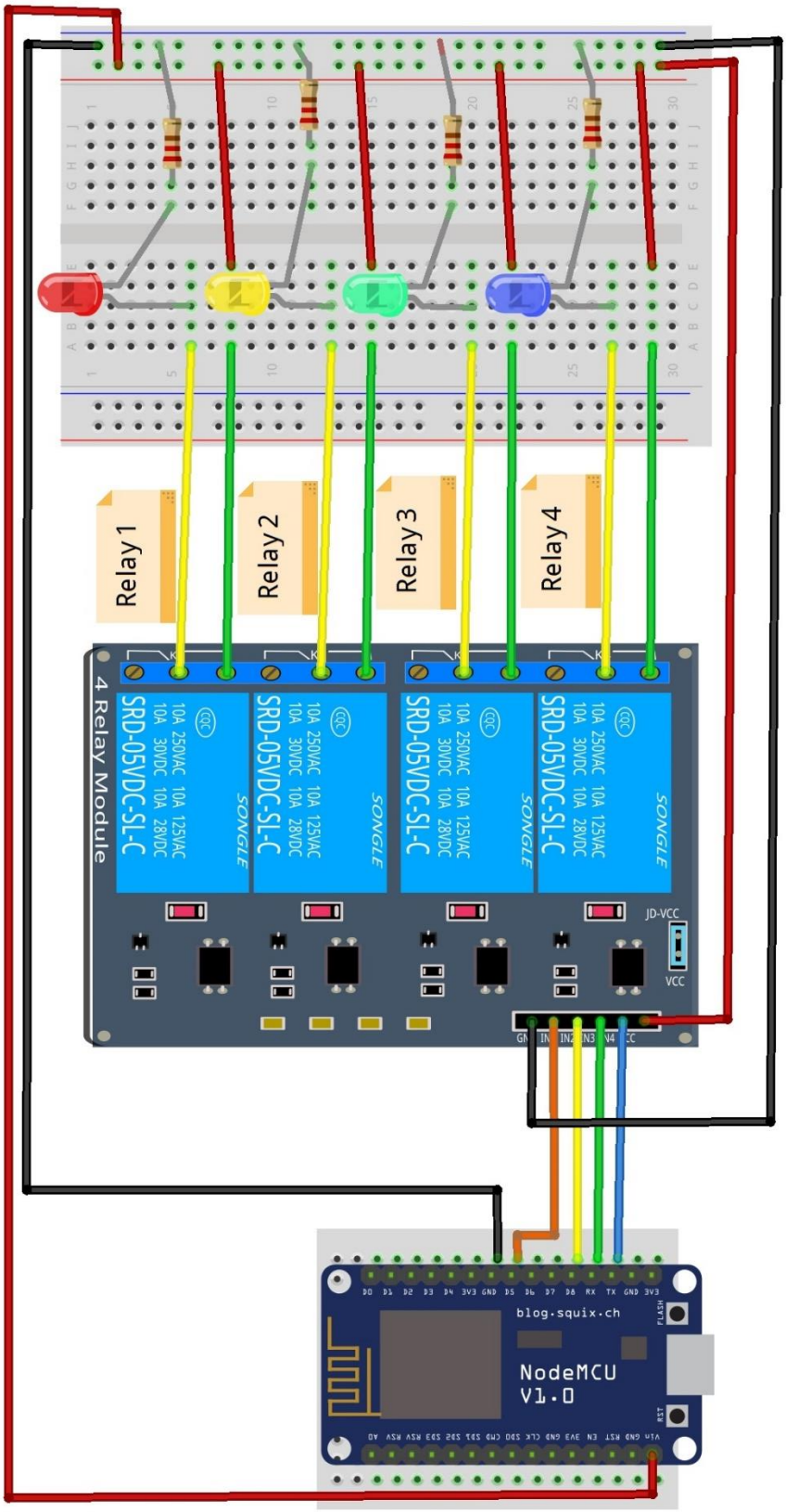


Figure 14. Connection diagram of Node MCU controlling 4 channel relay module.

CHAPTER 4

HARWARE MODELLING AND SETUP

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

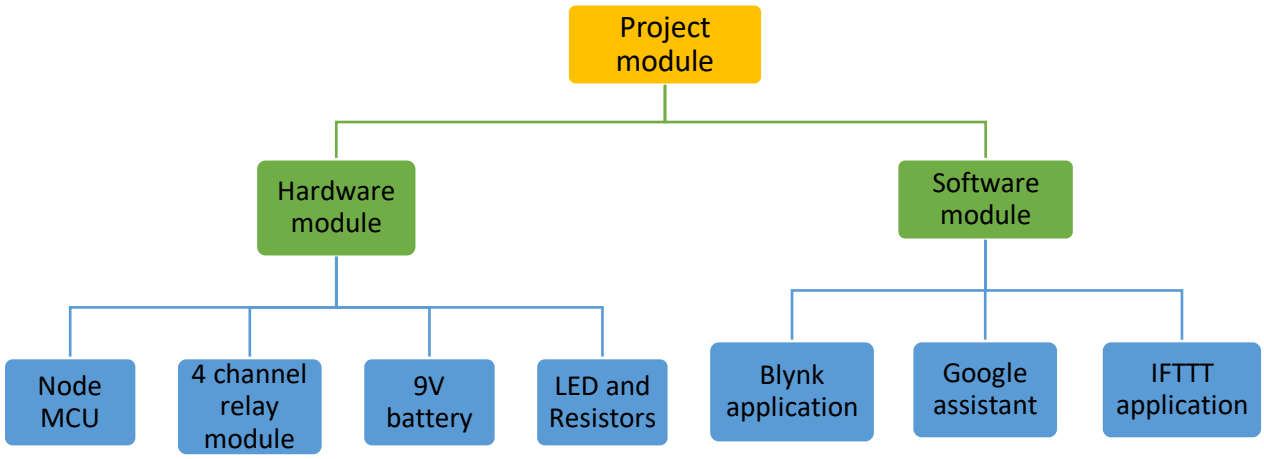


Figure 15. Layout of project module

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 REQIRED

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

Table 2. Component listing.

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.

Figure 16. Set 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 Amica.

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*

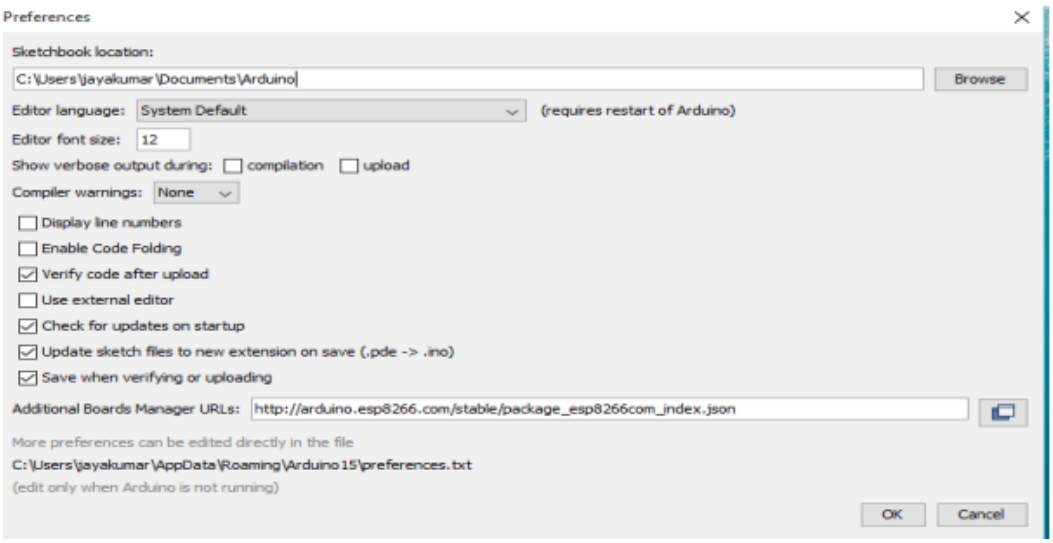


Figure 17. Arduino IDE preferences.

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.

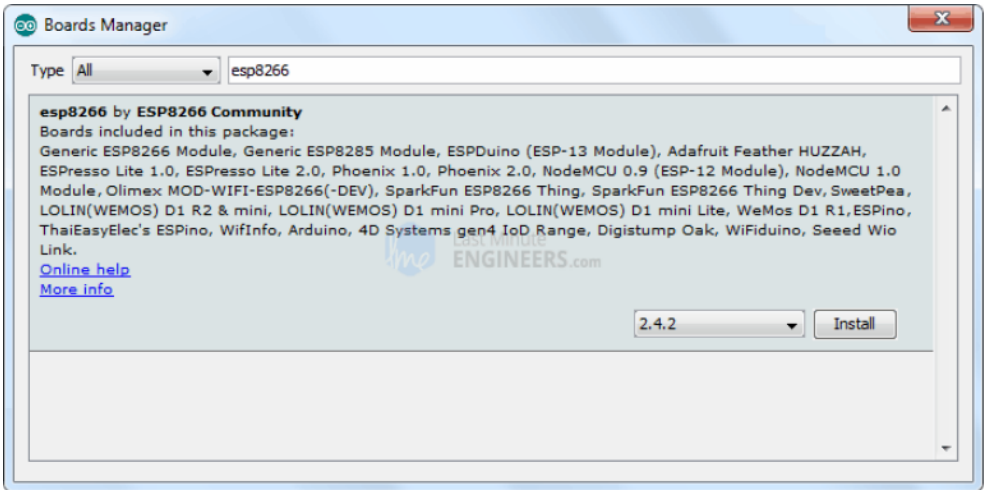


Figure 18. ESP8266 board installation in Arduino IDE.

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.

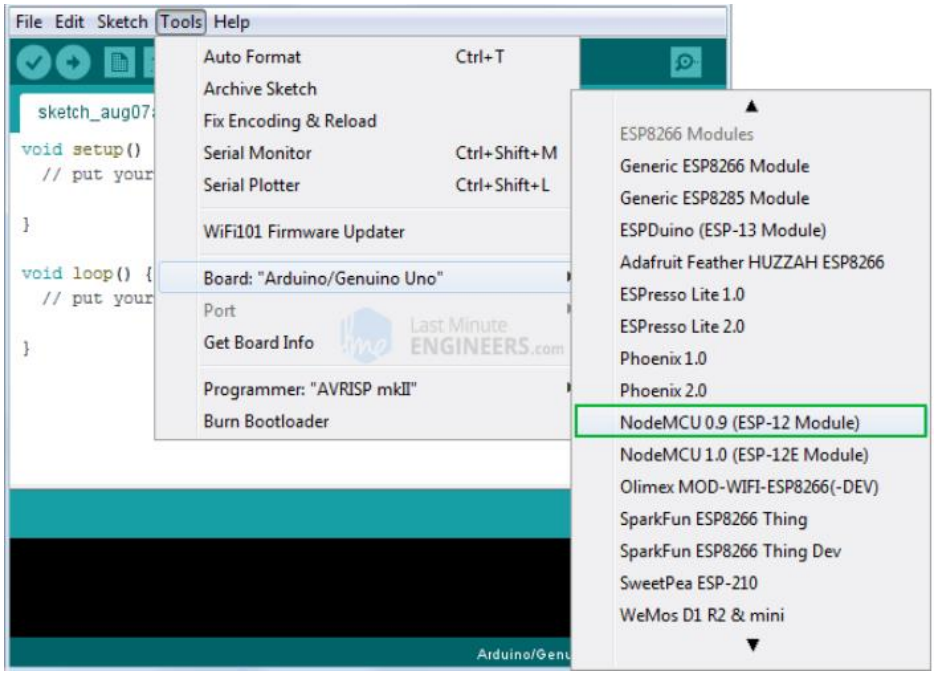


Figure 19. Arduino IDE board manager installation.

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

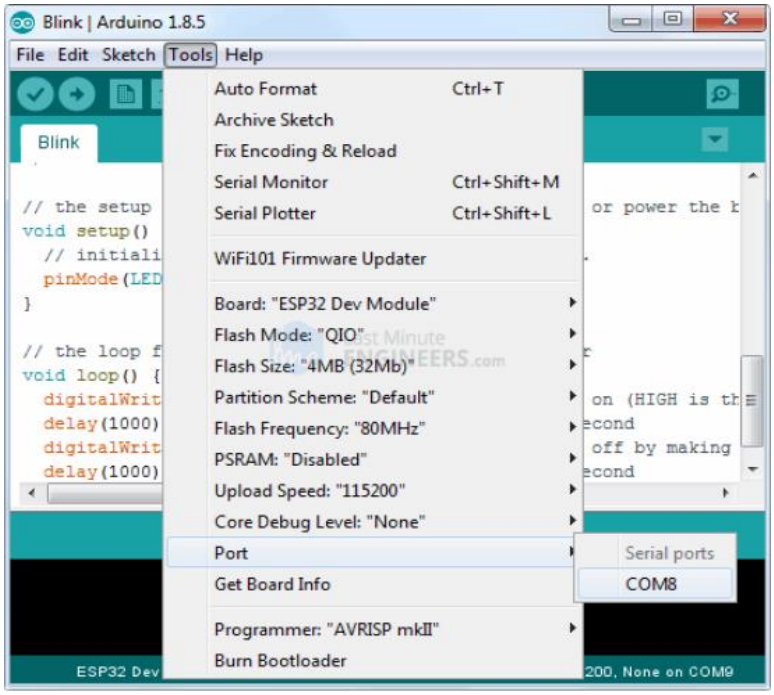


Figure 20. Assigning communication port on Arduino IDE.

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 all the settings we need to change. So now we begin writing the code.
- Select Files > Examples > Blynk > Boards_WIFI > ESP8266_Standalone. A new file with some prewritten 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.

Figure 21. Code in Arduino IDE to be installed to Node MCU.

- 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.

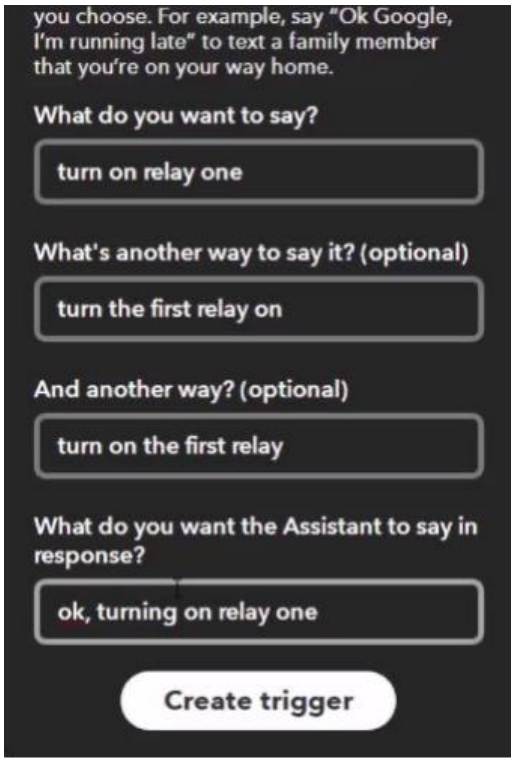


Figure 22. IFTTT configured with actions and commands.

- 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.

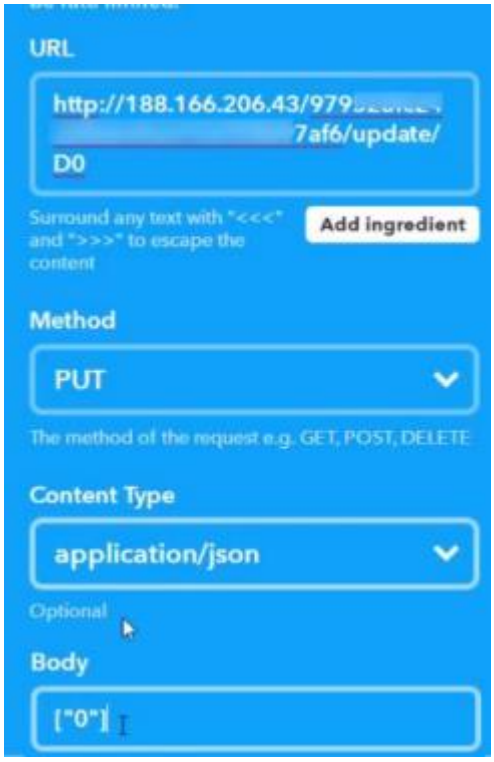


Figure 23. Configuration of applet to switch relay with voice commands.

- 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 ASSEBLY

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.

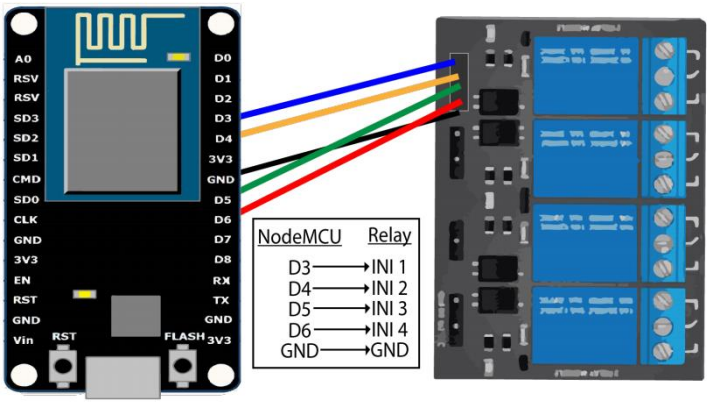


Figure 24. Node MCU & 4 channel relay connection.

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.

CHAPTER 5
LOGIC AND OPERATION

5.1 FLOW CHART

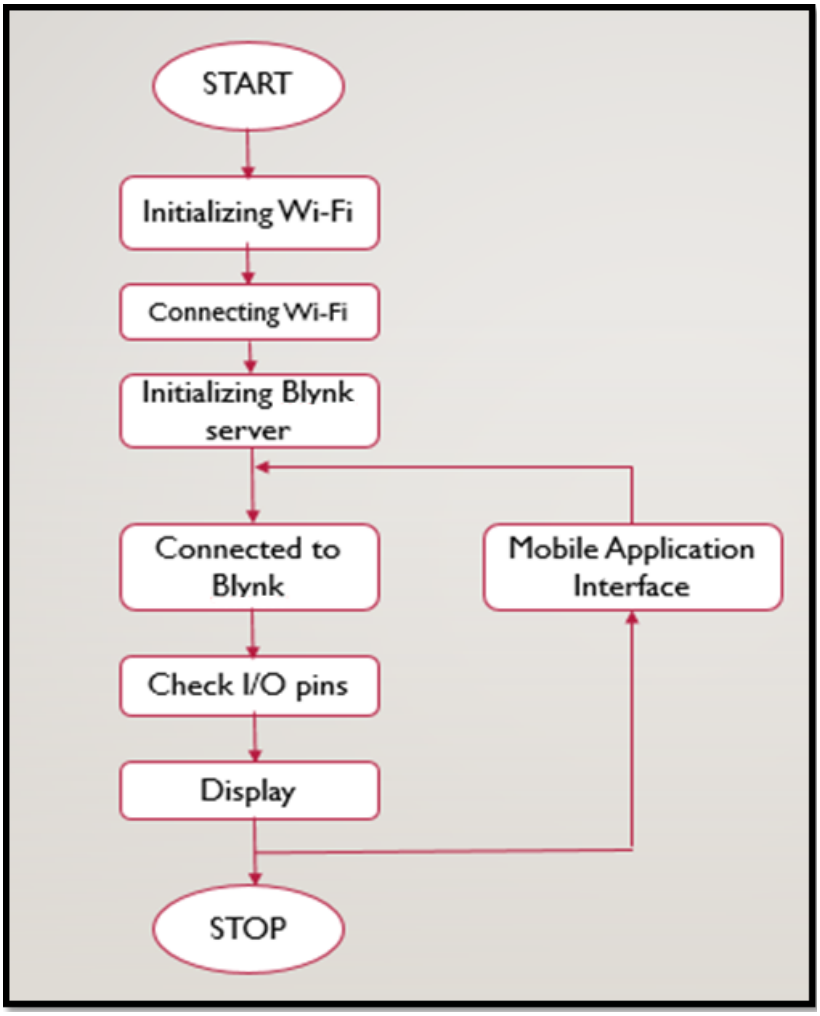


Figure 25. Flow chart of prototype function.

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.

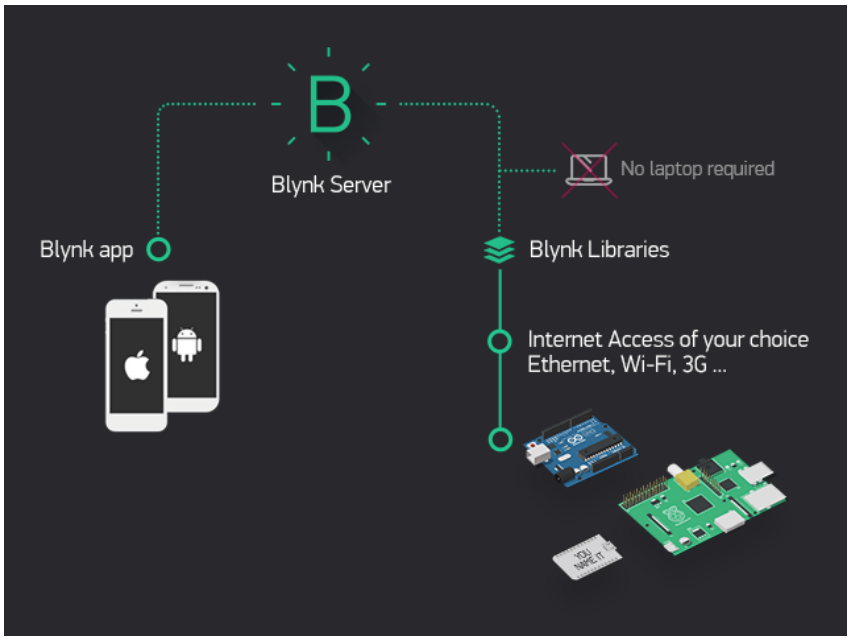


Figure 26. Working principle of Blynk application.

5.4 WIRELSS 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.

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 frequency hopping	Adaptive frequency hopping	Dynamic frequency selection	Dynamic frequency selection, transmit power control
Basic cell	Piconet	Picomet	Star	BSS
Extension of basic cell	Scattemet	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

Table 3. Comparison chart of Wi-Fi with other wireless communication protocols.

5.5 VOICE MODE CONTROL

The prototype works in both switch mode and voice mode of control. The switch mode is simply be accessing the radio buttons on the Blynk application, and the process of control has been discussed earlier in this chapter in the section before. Here we will discuss the voice mode control of the prototype. We use application IFTTT and Google assistant on smart phone to achieve control by voice commands.IFTTT stand for 'If This Then That', is an interface that provides web based service in which devices are connected to mobile application.

We cannot connect the Google Assistant to the Node MCU directly, and that is the only reason we are using the Blynk app. Blynk app can directly connect to the Node MCU and send data to it. So, if we can send the voice commands interpreted by Google assistant directly to the Blynk app, the Blynk app can then forward those commands to the NodeMCU. But the problem is Google Assistant cannot directly understand foreign commands like "turn on the fan" or "turn on relay one" etc. on its own. So, to solve this we use another intermediate application/website called 'IFTTT'.

Simply, to control our home appliances over the internet we are using Node MCU and to connect Node MCU with the home appliances we use a relay board. Now to send on or off signals to the Node MCU we use our smartphone, and we do this using the Blynk app. But we want to send the on or off signals using voice commands. To do this we use google assistant in our smartphone and an app called IFTTT.

So, in the end what will happen is, when we say a voice command like "ok google turn on the light" to the Google Assistant, Google Assistant sends that this foreign command to IFTTT. IFTTT interprets this command and sends an on or off signal to the Blynk app via the Blynk Server. Blynk will then send this signal to the Node MCU and then to our electrical appliances.

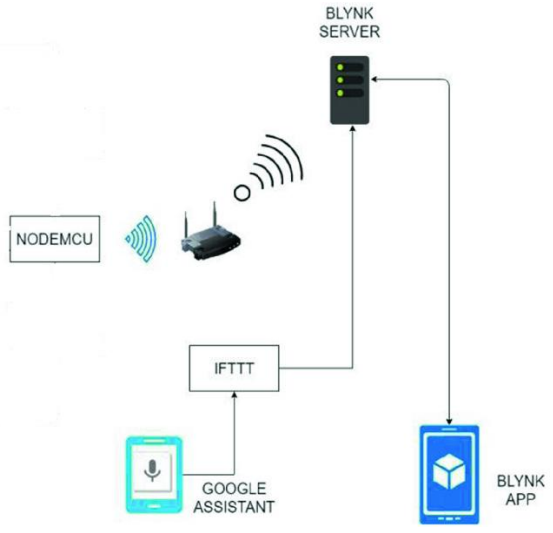


Figure 27. Voice and switch mode control.

5.6 COST ESTIMATION

SL. NO	COMPONENTS	QUANTITY	PRICE
1.	Node MCU	1	₹350
2.	4 channel relay board	1	₹120
3.	9V battery	1	₹50
4.	LED	4	₹8
5.	2.2K Ω Resistor	4	₹4
6.	Blank PCB (KS100)	1	₹40
7.	Male pin header	1	₹5
8.	Female pin header	1	₹5
9.	Jumper wires	8	₹40
10.	USB Cable	1	₹50
Total			₹672

Table 4. Costing of Project.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

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 LIMITATIONS

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 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

6.4 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.

CHAPTER 7
REFFERENCES

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/

APPENDIX A

HARDWARE DESCRIPTION

NODE MCU

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

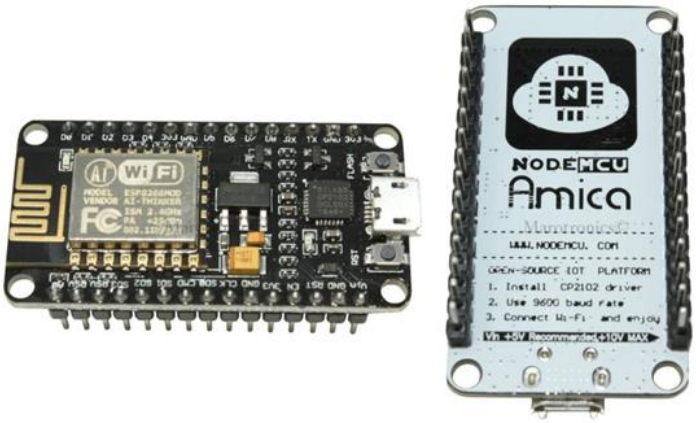


Figure 28. Node MCU module.

RESISTOR

Figure 29. Resistor.

Resistance is the opposition of a material to the current. It is measured in Ohms Ω. All conductors represent a certain amount of resistance, since no conductor is 100% efficient. To control the electron flow (current) in a predictable manner, we use resistors. Electronic circuits use calibrated lumped resistance to control the flow of current. Broadly speaking, resistor can be divided into two groups viz. fixed & adjustable (variable) resistors. In fixed resistors, the value is fixed & cannot be varied. In variable resistors, the resistance value can be varied by an adjuster knob. It can be divided into (a) Carbon composition (b) Wire wound (c) Special type. The most common type of resistors used in our projects is carbon type. The resistance value is normally indicated by colour bands. Each

resistance has four colours, one of the bands on either side will be gold or silver, this is called fourth band and indicates the tolerance, others three band will give the value of resistance (see table). For example, if a resistor has the following marking on it say red, violet, gold. Comparing these coloured rings with the colour code, its value is 27000 ohms or 27 kilo ohms and its tolerance is ±5%. Resistor comes in various sizes (Power rating).The bigger the size, the more power rating of 1/4 watts. The four colour rings on its body tells us the value of resistor value.

Colour Code of Resistor

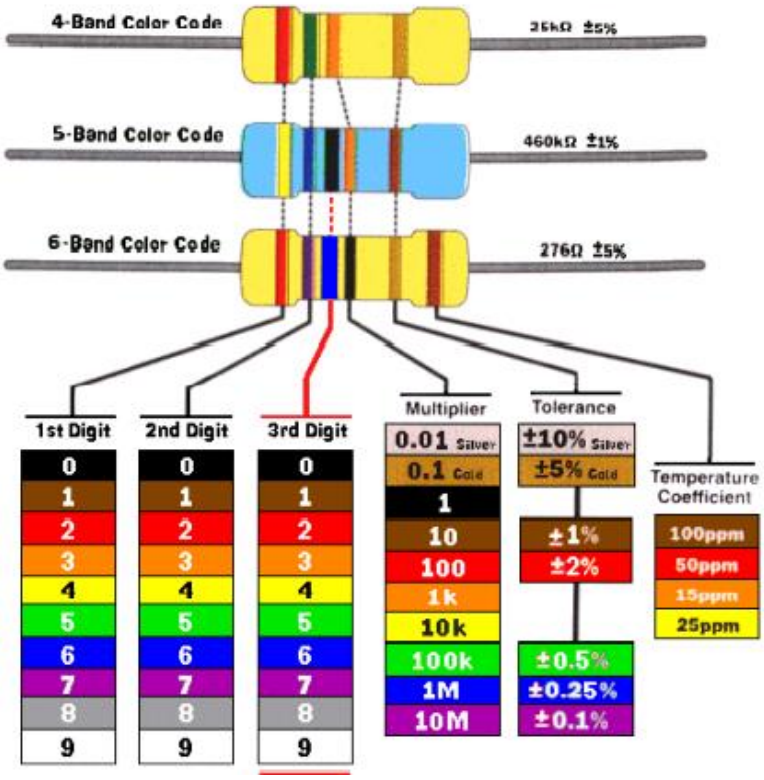


Figure 30. Colour code of resistor.

RELAY

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches.

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.

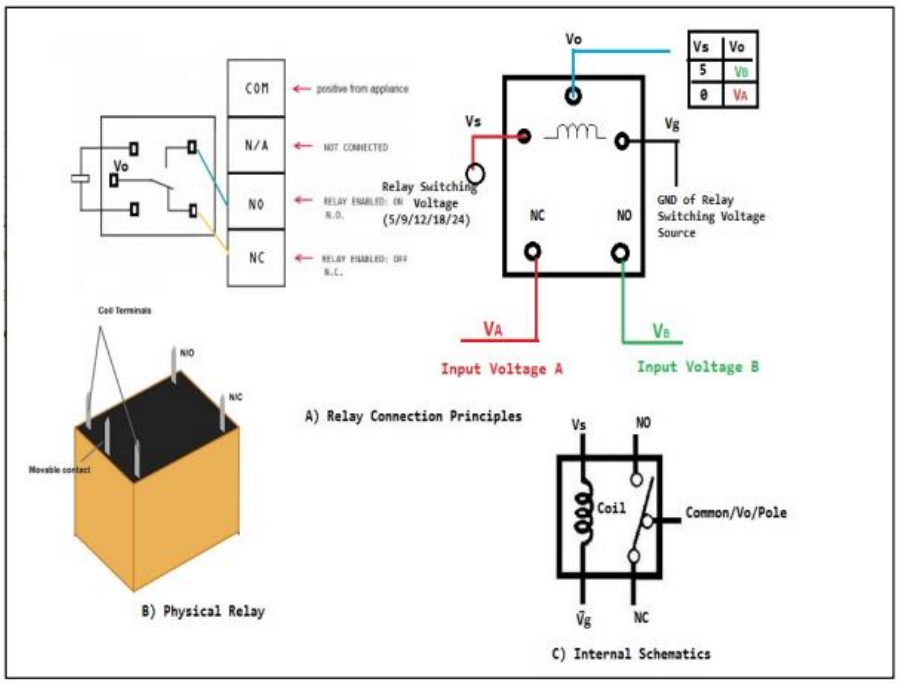


Figure 31. 6V Cube relay.

4 CHANNEL 5V RELAY MODULE

Figure 32. 4 Channel 5V Relay Module.

In a 5V 4-channel relay interface board, 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.

When the signal port is at low level, the signal light will light up and the optocoupler 817c (it transforms electrical signals by light and can isolate input and output electrical signals) will conduct, then the transistor will conduct, the relay coil will be electrified, and the normally open contact of the relay will be closed. When the signal port is at high level, the normally closed contact of the relay will be closed. So you can connect and disconnect the load by controlling the level of the control signal port.

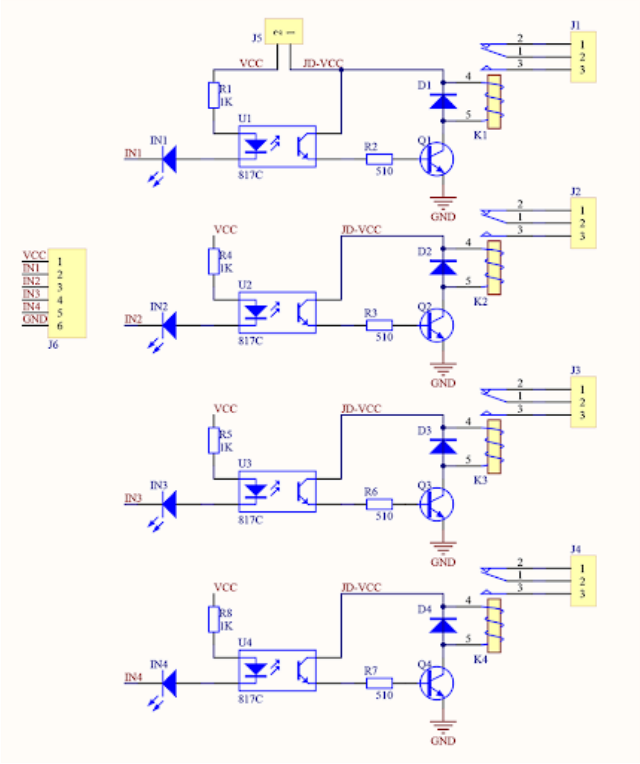


Figure 33. Schematic of relay module.

BLANK PCB

A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer (outer and inner layers). Multi-layer PCBs allow for much higher component density. Conductors on different layers are connected with plated-through holes called vias. Advanced PCBs may contain components - capacitors, resistors or active devices - embedded in the substrate.

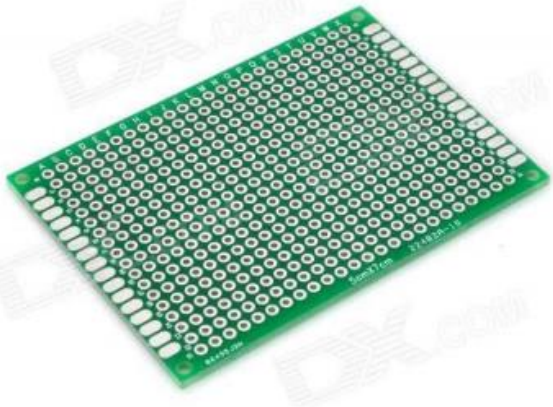


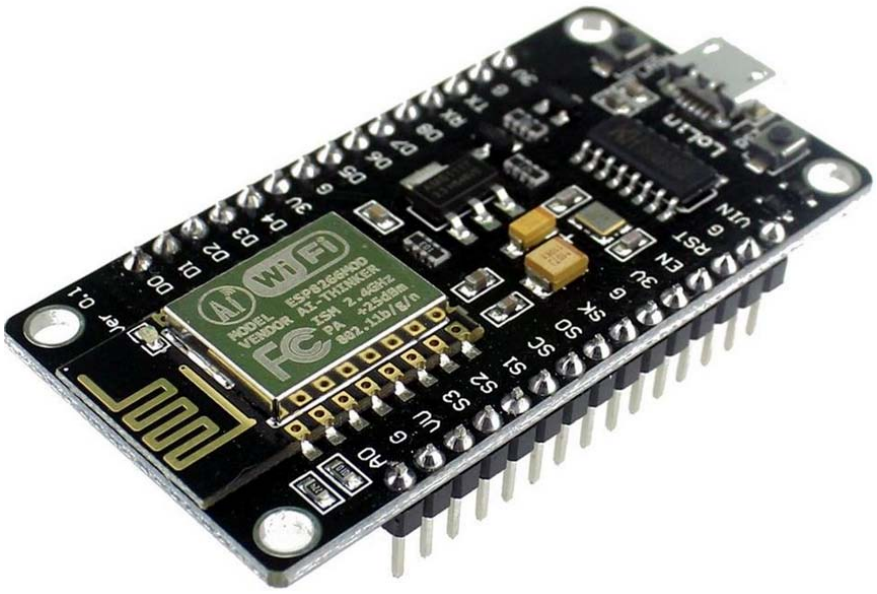
Figure 34. Blank glass epoxy PCB board.

FR-4 glass epoxy is the primary insulating substrate upon which the vast majority of rigid PCBs are produced. A thin layer of copper foil is laminated to one or both sides of an FR-4 panel. Circuitry interconnections are etched into copper layers to produce printed circuit boards. Complex circuits are produced in multiple layers. Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs require the additional design effort to lay out the circuit, but manufacturing and assembly can be automated. Manufacturing circuits with PCBs is cheaper and faster than with other wiring methods as components are mounted and wired with one single part. Furthermore, operator wiring errors are eliminated.

APPENDIX B
DATASHEETS

User Manual V1.2

ESP8266 NodeMCU WiFi Devkit



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

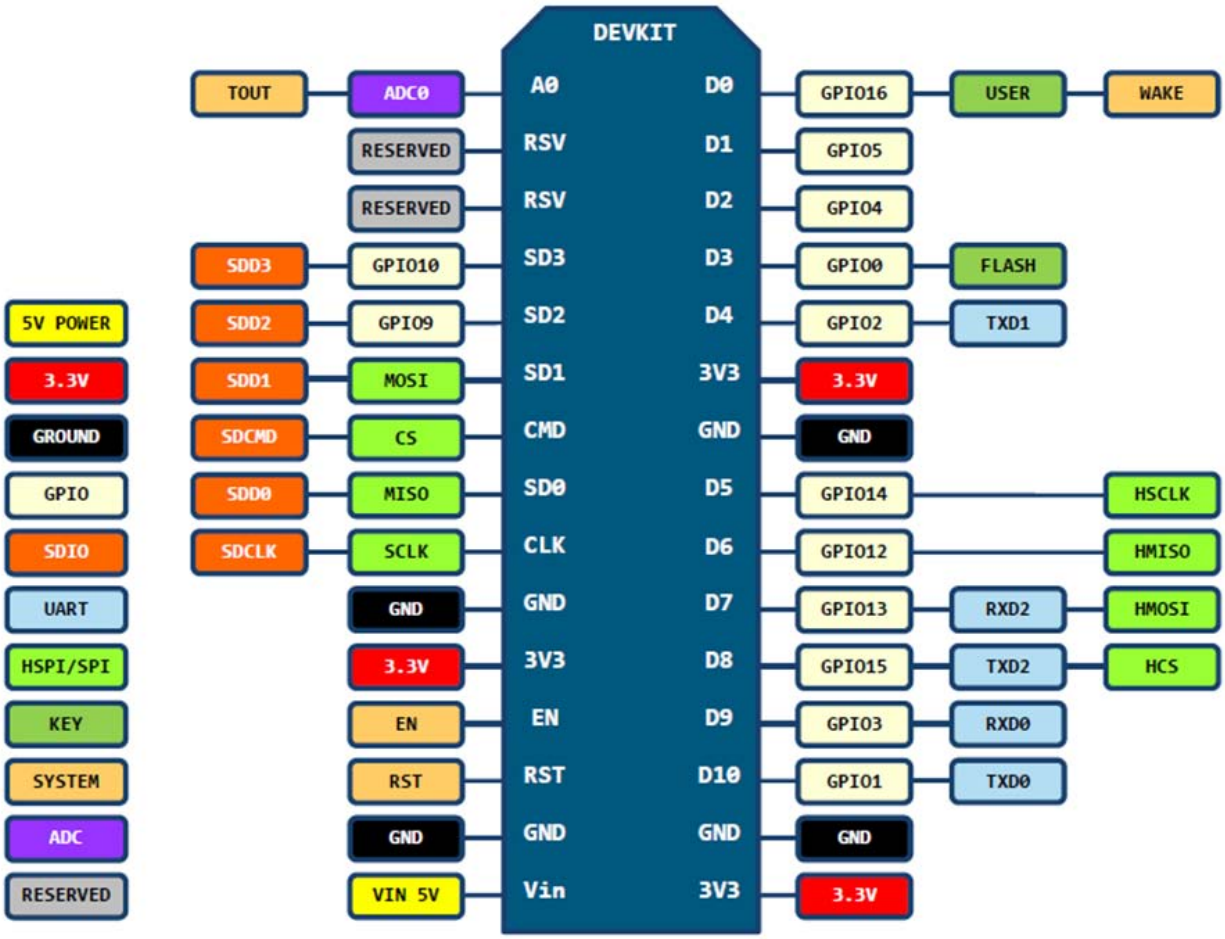
This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

Table of Contents

1. 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: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).
- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

2. Pin Definition:



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

3. 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.

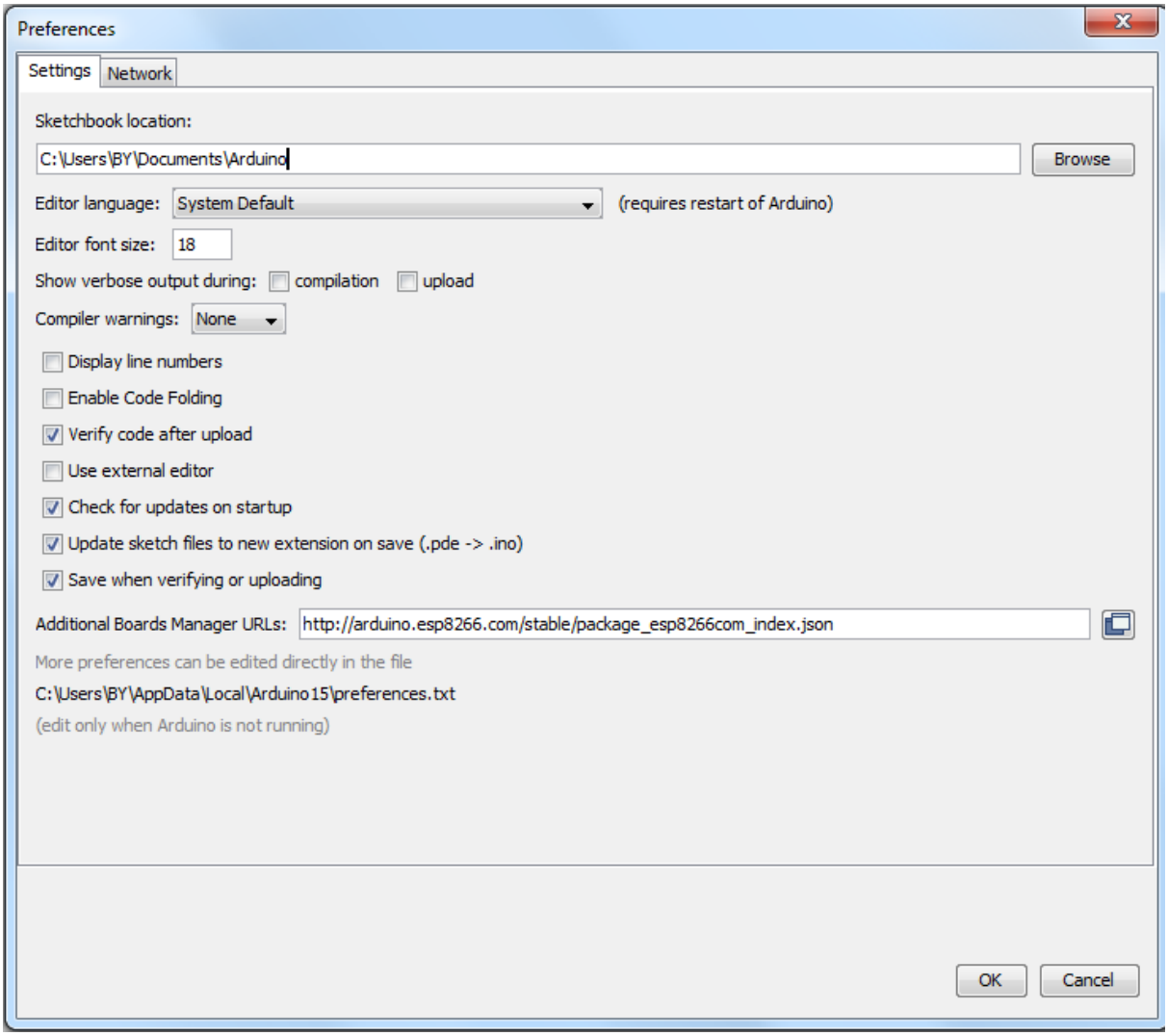
3.1 Install the Arduino IDE 1.6.4 or greater

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.

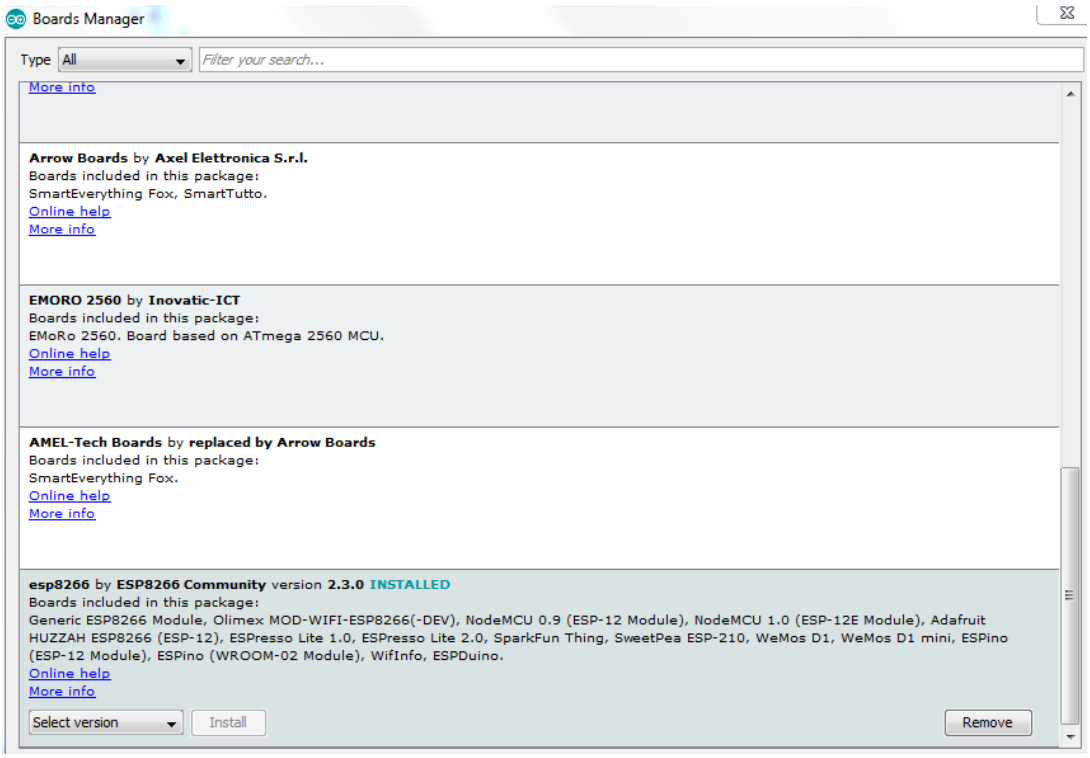
3.2 Install the ESP8266 Board Package

Enter ***http://arduino.esp8266.com/stable/package_esp8266com_index.json*** into *Additional Board Manager URLs* field in the Arduino v1.6.4+ preferences.



Click 'File' -> 'Preferences' to access this panel.

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

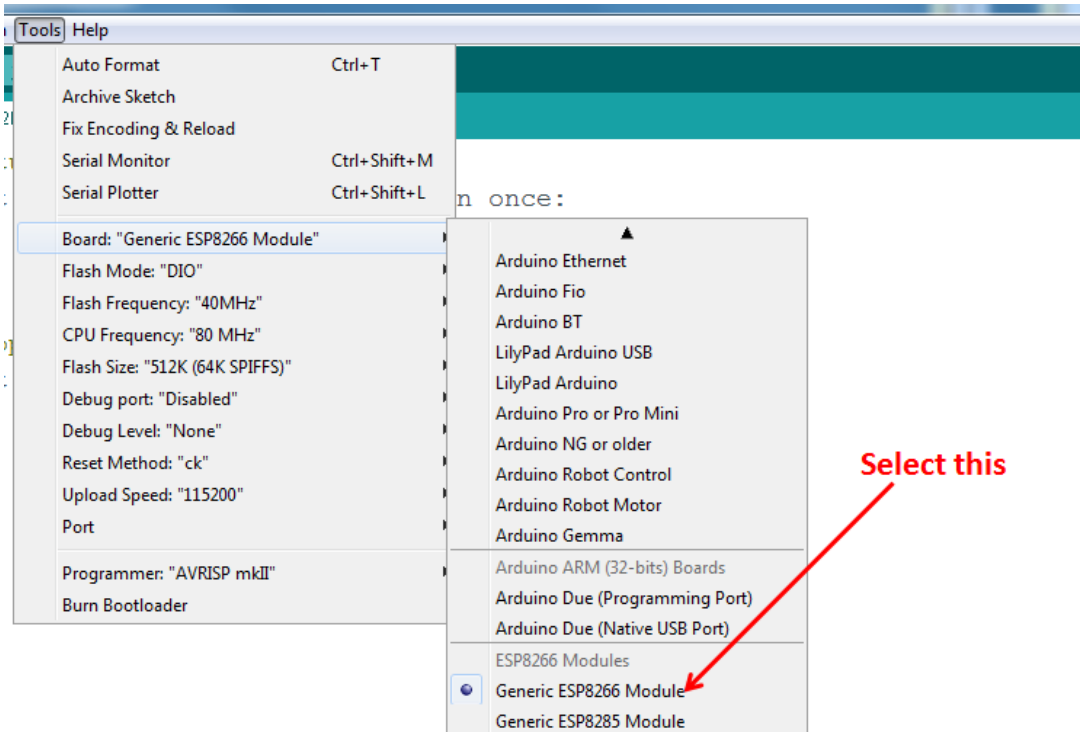


Click 'Tools' -> 'Board:' -> 'Board Manager…' to access this panel.

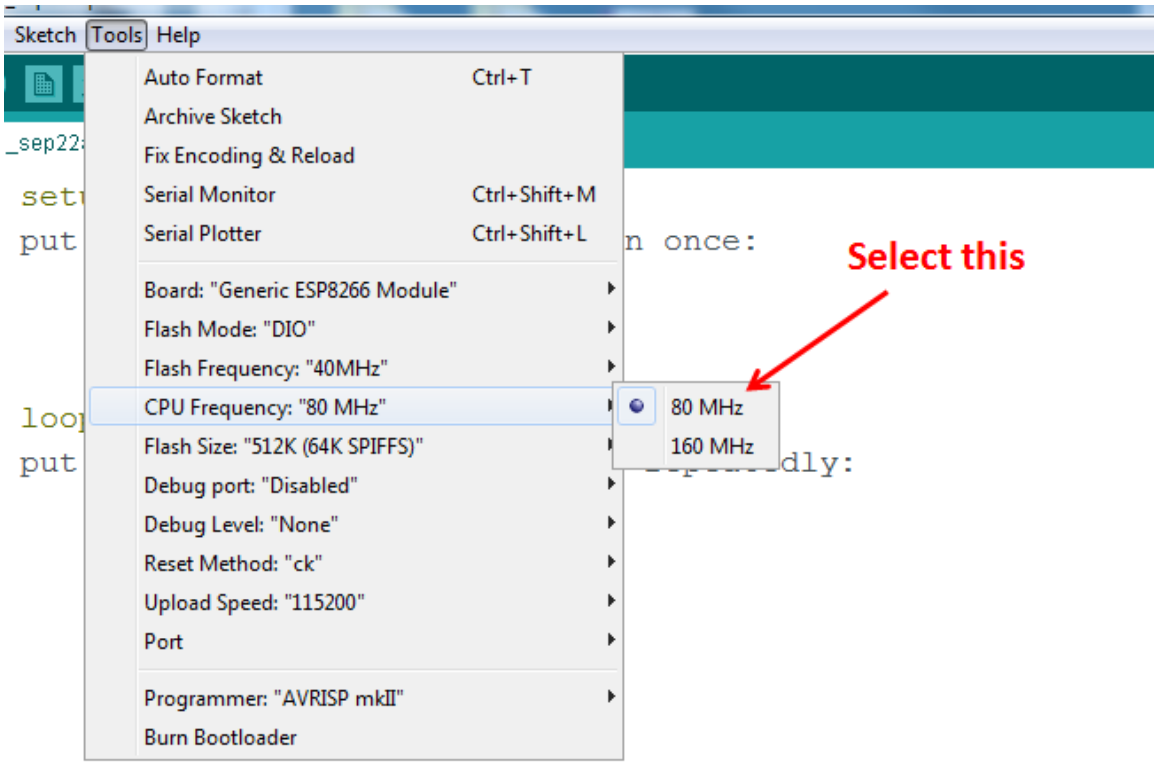
Scroll down to ' esp8266 by ESP8266 Community ' and click "Install" button to install the ESP8266 library package. Once installation completed, close and re-open Arduino IDE for ESP8266 library to take effect.

3.3 Setup ESP8266 Support

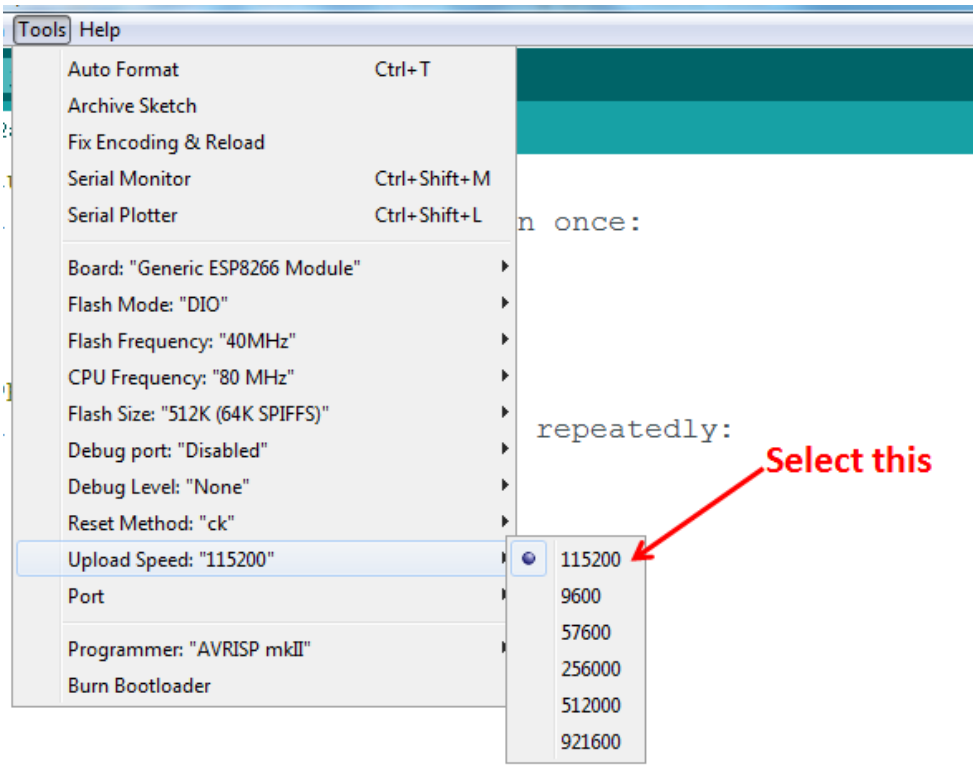
When you've restarted Arduino IDE, select 'Generic ESP8266 Module' from the 'Tools' -> 'Board:' dropdown menu.



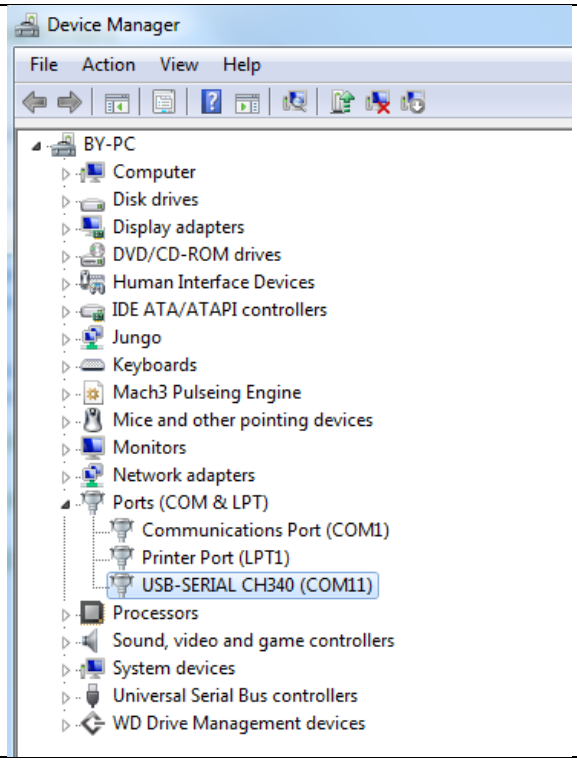
Select 80 MHz as the CPU frequency (you can try 160 MHz overclock later)



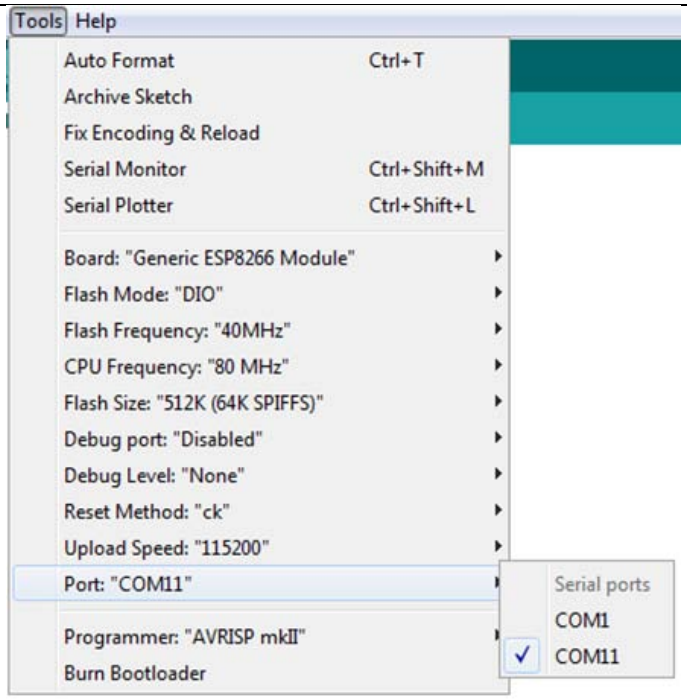
Select '115200' baud upload speed is a good place to start - later on you can try higher speeds but 115200 is a good safe place to start.



Go to your Windows 'Device Manager' to find out which Com Port 'USB-Serial CH340' is assigned to. Select the matching COM/serial port for your CH340 USB-Serial interface.



Find out which Com Port is assign for CH340



Select the correct Com Port as indicated on 'Device Manager"

Note: if this is your first time using CH340 " USB-to-Serial " interface, please install the driver first before proceed the above Com Port setting. The CH340 driver can be download from the below site:

https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers

3.4 Blink Test

We'll begin with the simple blink test.

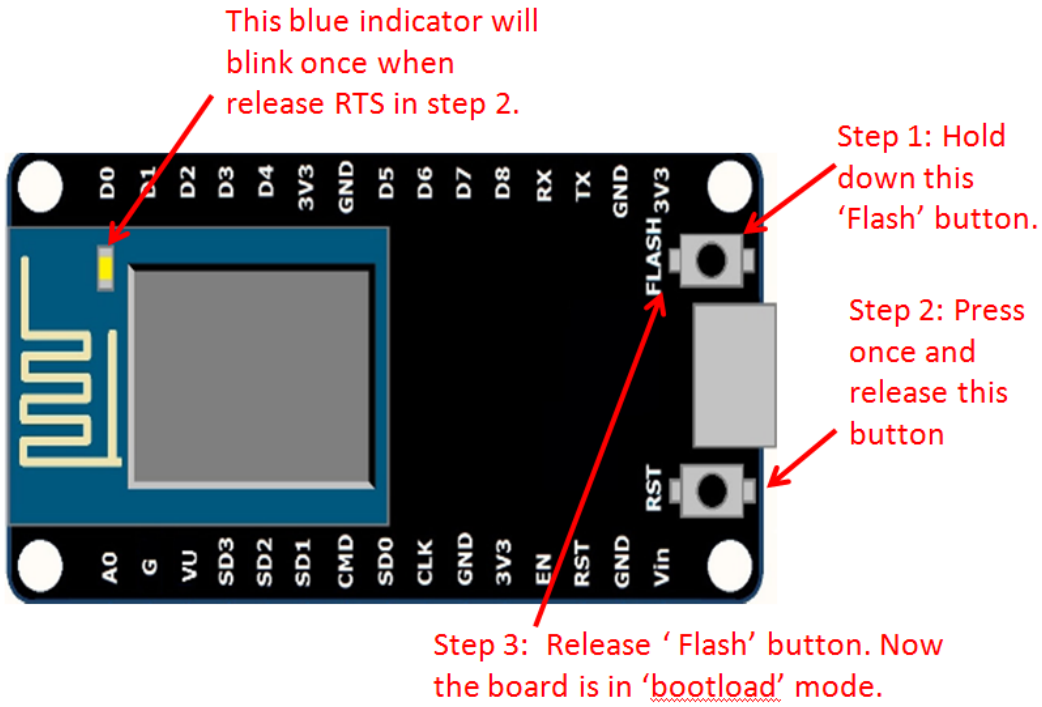
Enter this into the sketch window (and save since you'll have to). Connect a LED as shown in Figure3-1.

```
void setup() {
  pinMode(5, OUTPUT);   // GPIO05, Digital Pin D1
}

void loop() {
  digitalWrite(5, HIGH);
  delay(900);
  digitalWrite(5, LOW);
  delay(500);
}
```

Now you'll need to put the board into bootload mode. You'll have to do this before each upload. There is no timeout for bootload mode, so you don't have to rush!

- Hold down the 'Flash' button.
- While holding down ' Flash', press the 'RST' button.
- Release 'RST', then release 'Flash'

- When you release the ‘RST’ button, the blue indication will blink once, this means its ready to bootload.



Once the ESP board is in bootload mode, upload the sketch via the IDE, Figure 3-2.

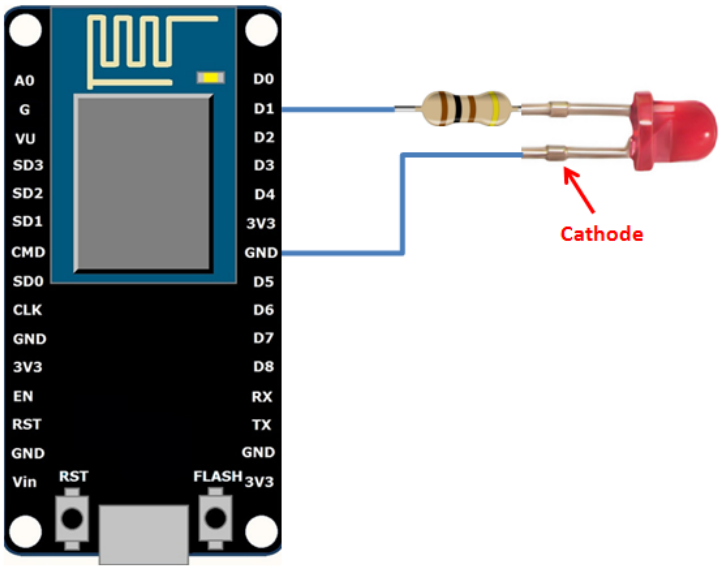


Figure3-1: Connection diagram for the blinking test

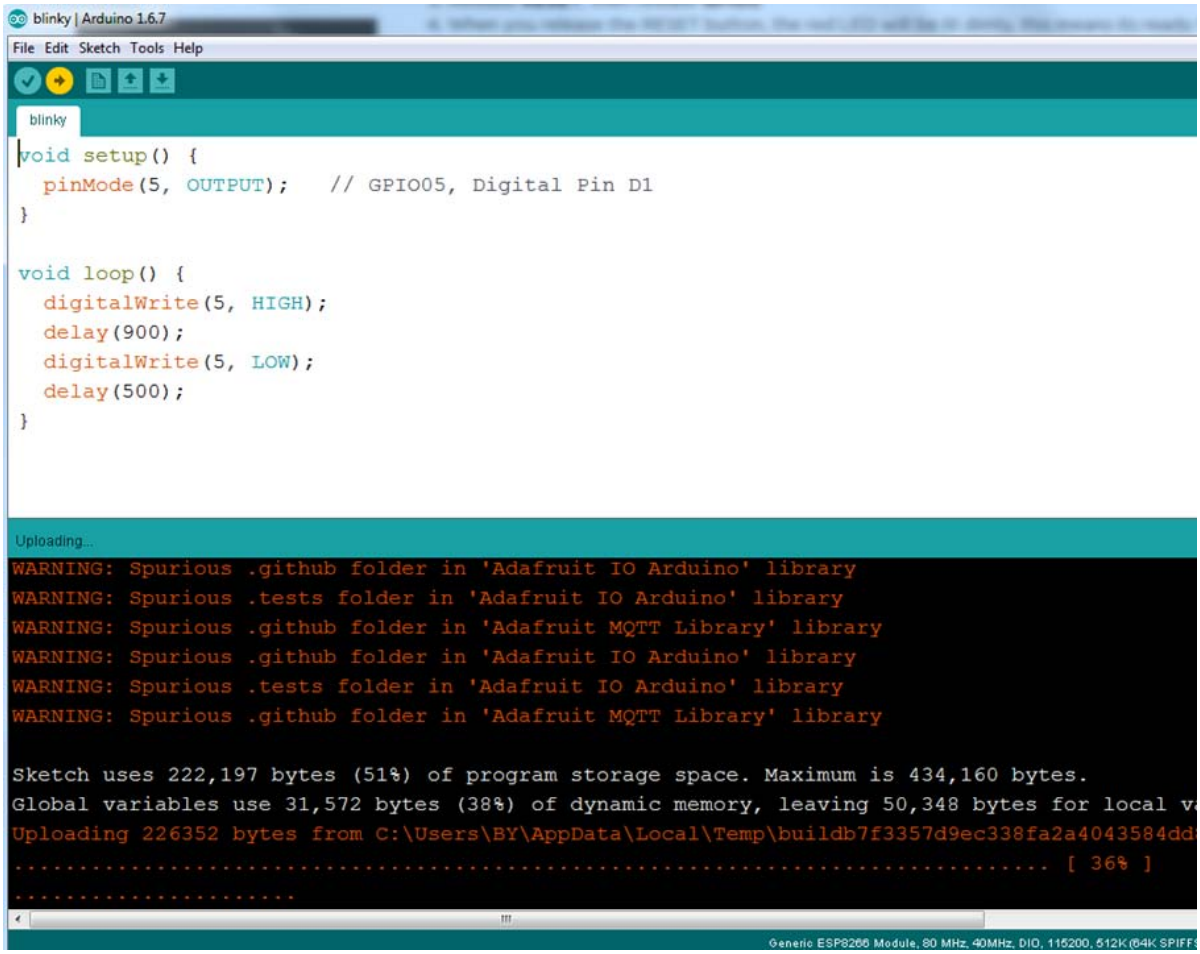


Figure 3.2: Uploading the sketch to ESP8266 NodeMCU module.

The sketch will start immediately - you'll see the LED blinking. Hooray!

3.5 Connecting via WiFi

OK once you've got the LED blinking, let's go straight to the fun part, connecting to a webserver. Create a new sketch with this code:

Don't forget to update:

const char* ssid = "yourssid";

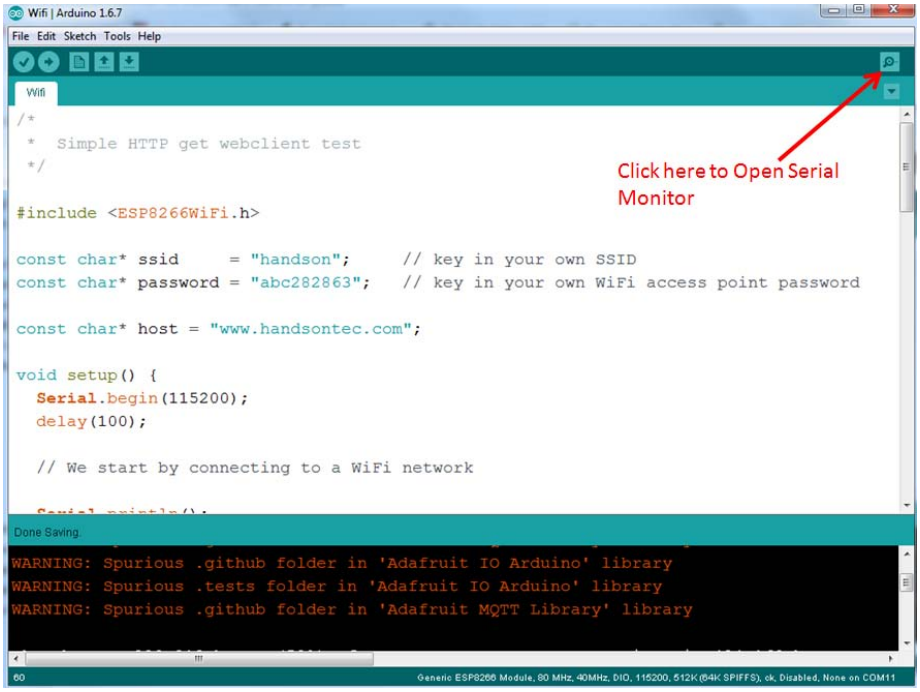
const char* password = "yourpassword";

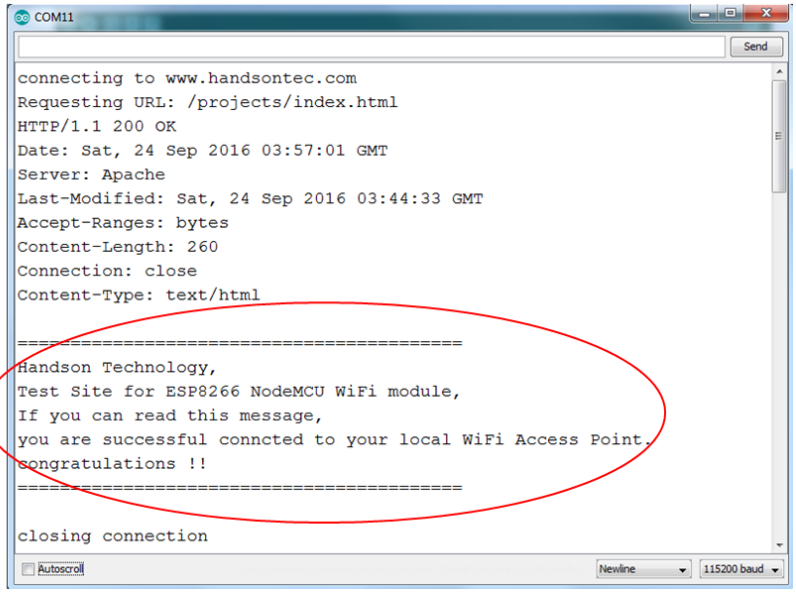
to your WiFi access point and password, then upload the same way: get into bootload mode, then upload code via IDE.

```
/*
 *  Simple HTTP get webclient test
 */

#include <ESP8266WiFi.h>

const char* ssid     = "handson";     // key in your own SSID
const char* password = "abc1234";     // key in your own WiFi access point
password
```

```
const char* host = "www.handsontec.com";

void setup() {
  Serial.begin(115200);
  delay(100);

  // We start by connecting to a WiFi network

  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

int value = 0;

void loop() {
  delay(5000);
  ++value;

  Serial.print("connecting to ");
  Serial.println(host);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("connection failed");
    return;
  }

  // We now create a URI for the request
  String url = "/projects/index.html";
  Serial.print("Requesting URL: ");
  Serial.println(url);

  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" +
               "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
  delay(500);

  // Read all the lines of the reply from server and print them to Serial
  while(client.available()){
    String line = client.readStringUntil('\r');
    Serial.print(line);
  }

  Serial.println();
  Serial.println("closing connection");
}
```

Open up the IDE serial console at 115200 baud to see the connection and webpage printout!



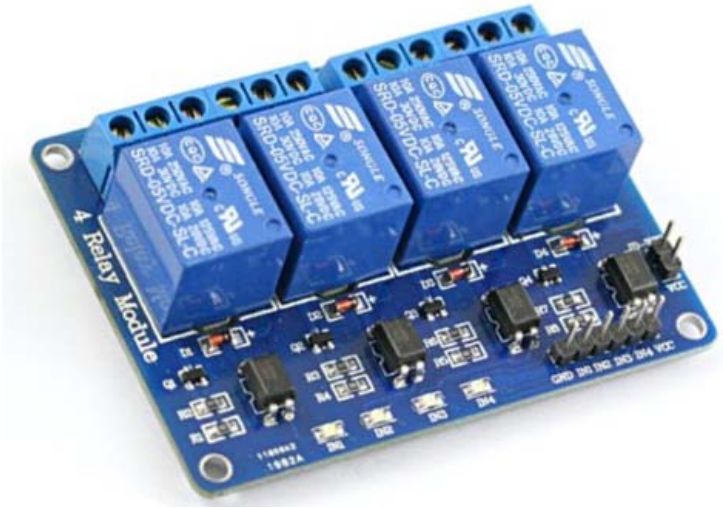


That's it, pretty easy right ! This section is just to get you started and test out your module.

User Guide

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.



Brief Data:

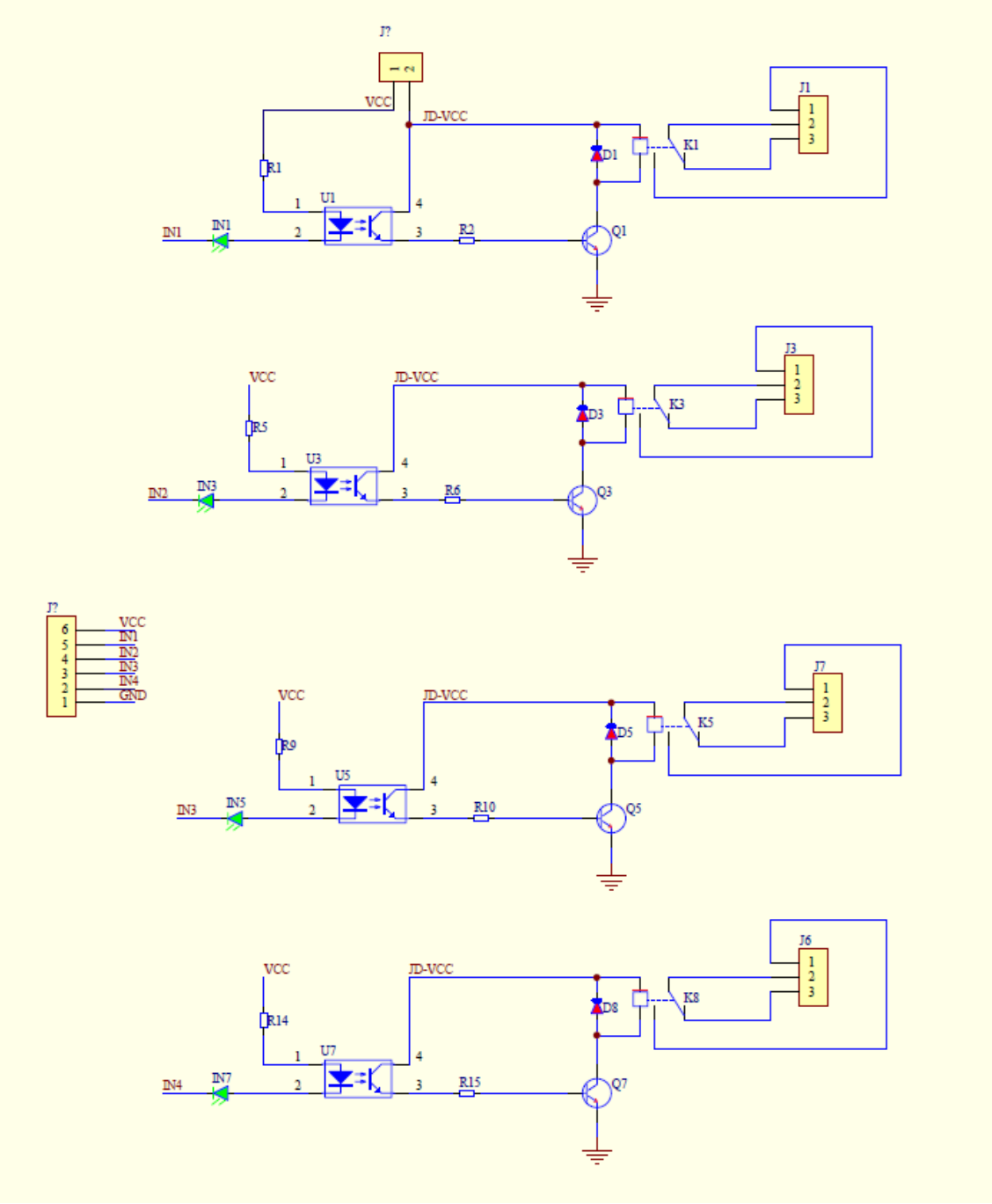
- Relay Maximum output: DC 30V/10A, AC 250V/10A.
- 4 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.
- Standard interface that can be controlled directly by microcontroller (8051, AVR, *PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.

Schematic:

VCC and RY-VCC are also the power supply of the relay module. When you need to drive a large power load, you can take the jumper cap off and connect an extra power to RY-VCC to supply the relay; connect VCC to 5V of the MCU board to supply input signals.

NOTES: If you want complete optical isolation, connect "Vcc" to Arduino +5 volts but do NOT connect Arduino Ground. Remove the Vcc to JD-Vcc jumper. Connect a separate +5 supply to "JD-Vcc" and board Gnd. This will supply power to the transistor drivers and relay coils.

If relay isolation is enough for your application, connect Arduino +5 and Gnd, and leave Vcc to JD-Vcc jumper in place.



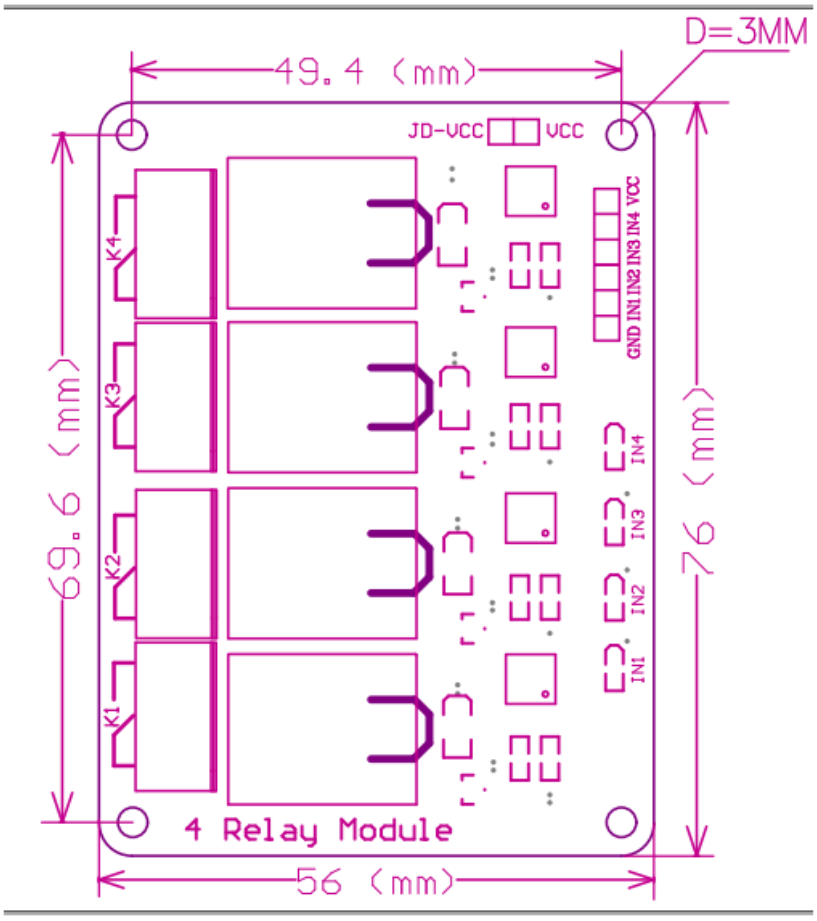
4 Channel Relay Module Schematic

It is sometimes possible to use this relay boards with 3.3V signals, if the JD-VCC (Relay Power) is provided from a +5V supply and the VCC to JD-VCC jumper is removed. That 5V relay supply could be totally isolated from the 3.3V device, or have a common ground if opto-isolation is not needed. If used with isolated 3.3V signals, VCC (To the input of the opto-isolator, next to the IN pins) should be connected to the 3.3V device's +3.3V supply.

NOTE: Some Raspberry-Pi users have found that some relays are reliable and others do not actuate sometimes. It may be necessary to change the value of R1 from 1000 ohms to something like 220 ohms, or supply +5V to the VCC connection.

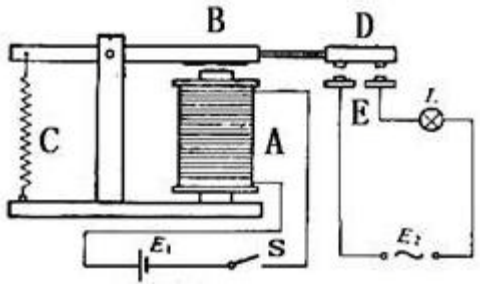
NOTE: The digital inputs from Arduino are Active LOW: The relay actuates and LED lights when the input pin is LOW, and turns off on HIGH.

Module Layout:



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 module

IN2: Signal triggering terminal 2 of relay module

IN3: Signal triggering terminal 3 of relay module

IN4: Signal triggering terminal 4 of relay module

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.

Testing Setup:

When a low level is supplied to signal terminal of the 4-channel relay, the LED at the output terminal will light up. Otherwise, it will turn off. If a periodic high and low level is supplied to the signal terminal, you can see the LED will cycle between on and off.

For Arduino：

Step 1：

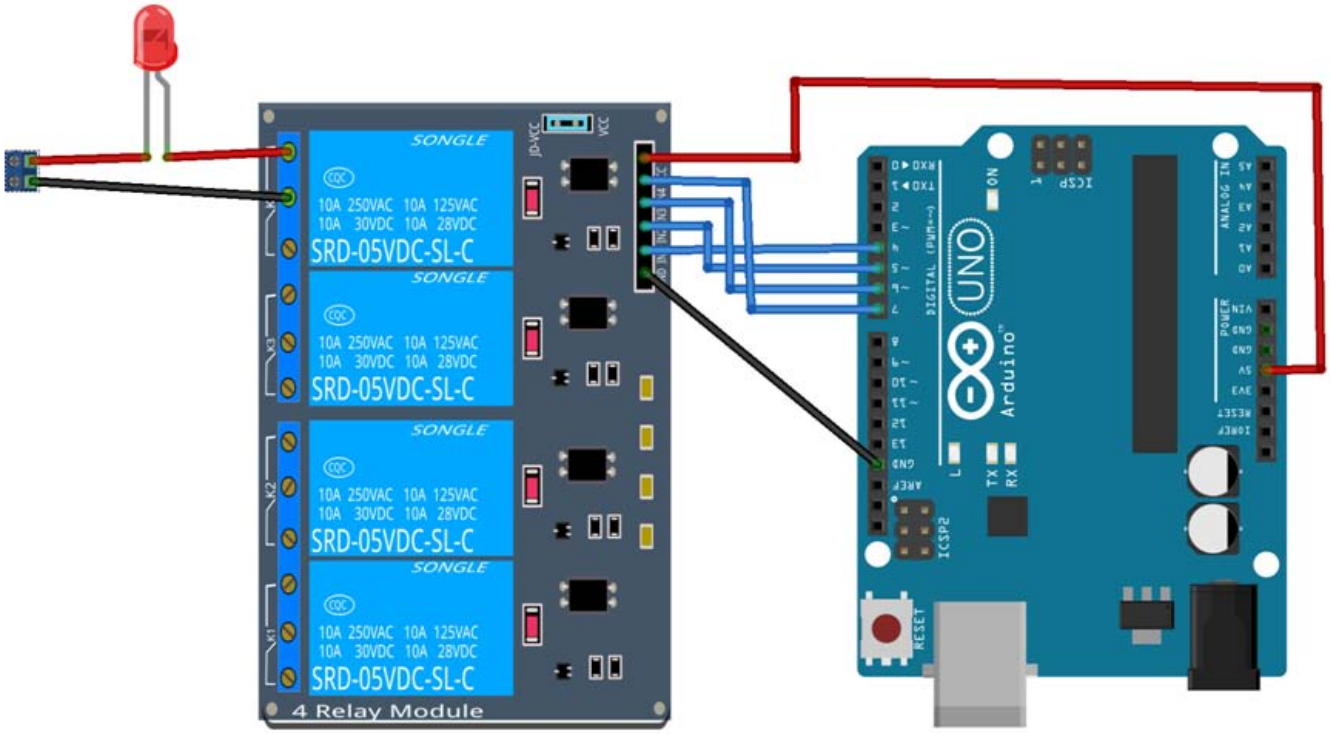
Connect the signal terminal IN1、IN2, IN3 & IN4 of 4-channel relay to digital pin 4, 5, 6, 7 of the Arduino Uno or ATMega2560 board, and connect an LED at the output terminal.

IN1> 4; IN2> 5; IN3>6; IN4>7

Step 2:

Upload the sketch "4 Channel Relay Demo " to the Arduino Uno or ATMega2560 board. Then you can see the LED cycle between on and off.

The actual figure is shown below:



Arduino Sketch: 4 Channel Relay Demo

```
/*************************************************
   Name: 4 channel_relay
   Description: control the 4 channel relay module to ON or OFF
   Website: www.handsontec.com
   Email: techsupport@handsontec.com
**************************************************/

//the relays connect to

int RelayControl1 = 4;    // Digital Arduino Pin used to control the motor
int RelayControl2 = 5;
int RelayControl3 = 6;
int RelayControl4 = 7;


void setup()
{
  Serial.begin(9600);
  pinMode(RelayControl1, OUTPUT);
  pinMode(RelayControl2, OUTPUT);
  pinMode(RelayControl3, OUTPUT);
  pinMode(RelayControl4, OUTPUT);
}


void loop()
{

  digitalWrite(RelayControl1,HIGH);// NO1 and COM1 Connected (LED on)
  delay(1000);
```

```
  digitalWrite(RelayControl1,LOW);// NO1 and COM1 disconnected (LED off)
  delay(1000);
  digitalWrite(RelayControl2,HIGH);
  delay(1000);
  digitalWrite(RelayControl2,LOW);
  delay(1000);
  digitalWrite(RelayControl3,HIGH);
  delay(1000);
  digitalWrite(RelayControl3,LOW);
  delay(1000);
  digitalWrite(RelayControl4,HIGH);
  delay(1000);
  digitalWrite(RelayControl4,LOW);
  delay(1000);
}
```