REAL TIME BASED IOT AUTOMATION WITH FEEDBACK LOOPS

A Project Report

Submitted in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY IN

ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted by

S.SAI	(18U45A0255)
K.PAVANKALYAN	(18U45A0244)
G.KASUBABU	(18U45A0237)
S.SURESH	(18U45A0226)
A.SAI KISHORE	(18U45A0264)

Under the Esteemed guidance of

Mr. DURGA R CH NOOKESH

Assistant Professor, Department of EEE

DADI INSTITUTE OF ENGINEERING & TECHNOLOGY

(Approved by A.I.C.T.E, New Delhi &Affiliated to JNTU, Kakinada) NAAC ACCREDITED INSTITUTE

ISO 9001:2008; ISO 14001:2004 & OHSAS 18001:2007 Certified Institute

NH-5, Anakapalle-531002, Visakhapatnam, A.P. 2021

DADI INSTITUTE OF ENGINEERING &TECHNOLOGY

(Approved by A.I.C.T.E & Affiliated to JNTU, Kakinada)
NAAC ACCREDITED INSTITUTE
ISO 9001:2008; ISO 14001:2004 & OHSAS 18001:2007 Certified Institution

NH-5, Anakapalle-531002, Visakhapatnam, A.P

CERTIFICATE

This is to certify that the Project work entitled "REAL TIME BASED IOT AUTOMATION WITH FEEDBACK LOOPS" is being submitted by S.SAI (18U45A0255), K. PAVANKALYAN (18U45A0244), G. KASUBABU (18U45A0237), S. SURESH (18U45A0226), A.SAI KISHOR (18U45A0264). in partial fulfilment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY in ELECTRICAL AND ELECTRONICS ENGINEERING during the academic year 2020-21.

Mr. D.R.CH NOOKESH	Mr. A KRISHNA NAG
Assistant Professor,EEE	Associate Professor,EEE
PROJECT GUIDE	**HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINER

DECLARATION

We hereby declare that this project work entitled **"REAL TIME BASED IOT AUTOMATION WITH FEEDBACK LOOPS"** has been carried out by us and contents have been presented in the form are for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRICAL AND ELECTRONICS ENGINEERING.** We further declare that this dissertation has not been submitted elsewhere for any Diploma or Degree of University.

Submitted By

S.SAI	(18U45A0255)
K.PAVANKALYAN	(18U45A0244)
G.KASUBABU	(18U45A0237)
S.SURESH	(18U45A0226)
A.SAI KISHORE	(18U45A0264)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be in complete without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

At the outset, we would like to thank **Mr. D.R.CH NOOKESH,** Asst. Prof., Department of Electrical and Electronics Engineering our project guide for giving us an opportunity to work on this challenging topic and providing us guidance for the successful completion of our project.

We feel elated to extend our floral gratitude to the Head of the department. **A KRISHNA NAG,** Department of Electrical and Electronics Engineering for his encouragement all the way during analysis of the project.

We would like to take this opportunity to express our profound sense of gratitude to revered PRINCIPAL, **Dr. CH. Narasimham** for giving us the opportunity of doing this project and for providing us all the required facilities.

We would like express our deep sense of gratitude to honorable Chairman, **Sri DADI RATNAKAR** for the resources and infrastructure provided for working on this project without any obstacles.

We also take this opportunity to express our heartfelt thanks to teaching and nonteaching staff of the department, for their perspective comments and suggestions. Our thanks to all others, who have directly or indirectly contributed in making our project a great success.

S.SAI	**(18U45A0255)**
K.PAVANKALYAN	**(18U45A0244)**
G.KASUBABU	**(18U45A0237)**
S.SURESH	**(18U45A0226)**
A.SAI KISHORE	**(18U45A0264)**

ABSTRACT

With advancement of Automation technology, life is getting simpler and easier in all aspects. In today's world Automatic systems are being preferred over manual system. With the rapid increase in the number of users of internet over the past decade has made Internet a part and parcel of life, and IoT is the latest and emerging internet technology. Internet of things is a growing network of everyday object-from industrial machine to consumer goods that can share information and complete tasks while you are busy with other activities. Wireless Home Automation system (HAS) using IoT is a system that uses computers or mobile devices to control basic home functions and features automatically through internet from anywhere around the world, an automated home is sometimes called a smart home. It is meant to save the electric power and human energy.

The home automation system differs from other system by allowing the user to operate the system from anywhere around the world through internet connection. In this paper we present a Home Automation system (HAS) using Blynk Community that employs the integration of cloud networking, wireless communication, to provide the user with remote control of various lights, fans, and appliances within their home and storing the data in the cloud. The system will automatically change on the basis of sensors' data. This system is designed to be low cost and expandable allowing a variety of devices to be controlled.

KEW WORDS: IOT (internet of things),node MCU (WI-FI) module, Relay, BLYNK (Android application)

INDEX

LIST OF FIGURES

LIST OF TABLES

Chapter No: 1

INTRODUCTION

1. INTRODUCTION

1.1 INTRODUCTION:

IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. The present generation has been experiencing high speed internet by using 4G LTE cellular technology, which allows evolution of swifter IoT-based home automation systems. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which encompasses technologies like controlling of different home appliances like light, fan, water pump and many more. A system has been designed to connect sensor data with user's daily life. There are already many smartphone oriented remote controller products. However current products always have platform compatibility problems and additionally GUI in such systems is complex.

This work proposes an approach to enhance and simplify the controlling and monitoring experience. With the use of sensors in home environment the appliances can be controlled remotely based on environmental conditions known from sensor data. The sensor data are processed by a microcontroller and delivered to mobile application through WEB server. The results of implementation and experimentation have shown the proposed system and platform that can provide more IoT application possibilities in daily life. Increasing reliability on mobile phone applications to deal with daily life scenarios has paved the way of modeling a system which will integrate sensors and actuators.

This also allows users to observe data and send commands by using their mobile phone application. With the advancement of technology controlling and monitoring of electronics appliances using android application with the help of internet connection has become possible. It gives us the opportunity to have full control over a particular place even being far away from it. IoT allows us to control many devices simultaneously and reduces human efforts. This process is done in low cost and controlling of many devices in a simple circuit is possible. Our user-friendly interface allows a user to easily control home appliances through the internet. Relays are used to switch loads. The entire system is run by mains power using a step-down transformer, rectifier, filter and a linear dc regulator. After receiving user's commands over the internet, microcontroller processes these instructions to operate these loads accordingly and display the system status on mobile application. Besides monitoring sensor data and controlling household devices, the proposed system provides additional features of emergency notification and automatic turn off of an appliance to prevent wastage of power. Thus, this system allows efficient home automation over the internet.

The paper discusses how a system has been implemented to make use of IoT for controlling home appliances using NodeMCU and Blynk app. The different parameters of home environment are sent as input to Node MCU through the sensor-modules. NodeMCU receives instructions from user's mobile application and processes them to control actuator circuits of home appliances.

1.2 Functional Block Diagram

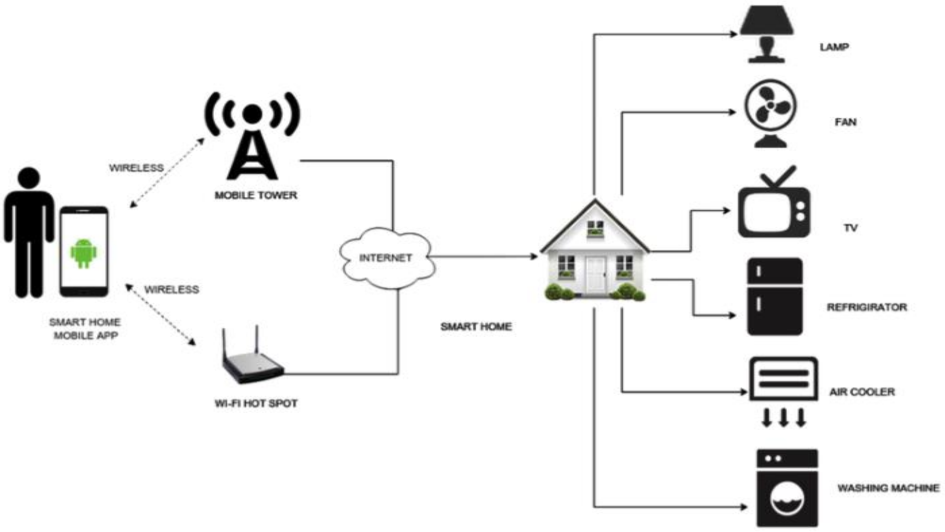


Fig 1.2 functional block diagram

1.3 DESCRIPTION OF BLOCK DIAGRAM

SMART PHONE:

Used for controlling Purpose, for giving command and gain output for this blynk android app is required

NodeMCU Esp-8266 (Wifi enable microcontroller): To take input and perform operation. obtain output as per user requirements.

RELAY DRIVER: Basically, the output of microcontroller is in Mili-volts so this output is not sufficient to run the bulky load output.

So as to run the appliances on 230v we require an Relay module so the output is fed to the relay module according to given input to the relay module it will generate output and drive various appliances and load e.g. Lamp, Fan, Tube light, T.V, etc.

OUTPUT:

These are the output Generated from the given input by the user the user can set the various output sections and can gain output through any of the section like Home appliance A for lights, Home appliance B for fans, Home appliance C for T. V .and many more……….

Home appliance A

Home appliance B

Home appliance C

Home appliance D

1.4 ORGANIZATION OF THESIS

A brief introduction on the subject matter is given in

Chapter -1 Discuss about the introduction of proposed project and the previous works are discussed briefly.

Chapter -2 it presents the literature review focusing on smart home automation using IOT

Chapter -3 introduce the internet of things, it's importance and future scope.

Chapter -4 deals the overview of the project, working operation of the circuit and different types of components, it's working and specifications.

Chapter -5 deals the Arduino software and coding.

Chapter -6 deals with blynk app.
Chapter -7 deals with results and discussion.

Chapter -8 deals with conclusion & future scope.

Chapter No: 2

LITERATURE SURVEY

2.LITERATURE SURVEY

1) BLUETOOTH BASED HOME AUTOMATION SYSTEM USING CELL PHONES:

- In Bluetooth based home automation system the home appliances are connected to the Arduino BT board at input output ports using relay.

- The program of Arduino BT board is based on high level interactive C language
  of microcontrollers; the connection is made via Bluetooth.

- The password protection is provided so only authorized user is allowed to access the appliances.

- The Bluetooth connection is established between Arduino BT board and phone for wireless communication. In this system the python script is used and it install on any of the Symbian OS environment, it is portable.

- One circuit is designed and implemented for receiving the feedback from the phone, which indicate the status of the device.

2) GSM BASED HOME AUTOMATION SYSTEM USING CELL PHONES:

- Because of the mobile phone and GSM technology, the GSM based home
  automation is lure to research.

- The SMS based home automation, GPRS based home automation and dual tone multi frequency (DTMF) based home automation, these options we considered mainly for communication in GSM.

- The home sensors and devices interact with the home network and communicates through GSM and SIM (subscriber identity module).

- The system use transducer which convert machine function into electrical signals which goes into microcontroller.

- The sensors of system convert the physical qualities like sound, temperature and
  humidity into some other quantity like voltage.

- The microcontroller analysis all signal and convert them into command to understand by GSM module.

- Select appropriate communication method among SMS, GPRS and DTFC based on the command which received GSM module.

3) WI-FI BASED HOME AUTOMATION SYSTEM USING CELL PHONES:

- Wi-Fi based home automation system mainly consist three modules, the server, the hardware interface module, and the software package.

- The figure shows the system model layout. Wi-Fi technology is used by server, and hardware Interface module to communicate with each other.

- The same technology uses to login to the server web-based application. Remote users can access server web-based application through the internet using compatible web browser. Software of the latest home automation system is split to server application software, and Microcontroller (Arduino) firmware.

- The Arduino software, built using C language, using IDE comes with the microcontroller itself. Arduino software is culpable for gathering events from connected sensors, then applies action to actuators and pre- programmed in the server.

- Another job is to report the and record the history in the server DB. The server application software package for the proposed home automation system, is a web-based application built using asp.net.

- The server application software can be accessed from internal network or from internet if the server has real IP on the internet using any internet navigator supports asp.net technology.

- Server application software is culpable of; maintain the whole home automation system, setup, and configuration.

Chapter no: 3

INTERNET OF THINGS

3.1 INTRODUCTION OF IOT

The Internet of Things (IoT) is an important topic in technology industry, policy, and engineering circles and has become headline news in both the specialty press and the popular media. This technology is embodied in a wide spectrum of networked products, systems, and sensors, which take advantage of advancements in computing power, electronics miniaturization, and network interconnections to offer new capabilities not previously possible. An abundance of conferences, reports, and news articles discuss and debate the prospective impact of the ―IoT revolution‖—from new market opportunities and business models to concerns about security, privacy, and technical interoperability. The large-scale implementation of IoT devices promises to transform many aspects of the way we live. For consumers, new IoT products like Internet-enabled appliances, home automation components, and energy management devices are moving us toward a vision of the ―smart home", offering more security and energy efficiency. Other personal IoT devices like wearable fitness and health monitoring devices and network enabled medical devices are transforming the way healthcare services are delivered. This technology promises to be beneficial for people with disabilities and the elderly, enabling improved levels of independence and quality of life at a reasonable cost.1 IoT systems like networked vehicles, intelligent traffic systems, and sensors embedded in roads and bridges move us closer to the idea of ―smart cities", which help minimize congestion and energy consumption. IoT technology offers the possibility to transform agriculture, industry, and energy production and distribution by increasing the availability of information along the value chain of production using networked sensors. However, IoT raises many issues and challenges that need to be considered and addressed in order for potential benefits to be realized. A number of companies and research organizations have offered a wide range of projections about the potential impact of IoT on the Internet and the economy during the next five to ten years. Cisco, for example, projects more than 24 billion Internet–connected objects by 2019; 2 Morgan Stanley, however, projects 75 billion networked devices by 2020.3 Looking out further and raising the stakes higher, Huawei forecasts 100 billion IoT connections by 2025.4 McKinsey Global Institute suggests that the financial impact of IoT on the global economy may be as much as $3.9 to $11.1 trillion by 2025.5 While the variability in predictions makes any specific number questionable, collectively they paint a picture of significant growth and influence. Some observers see the IoT as a revolutionary fully–interconnected ―smart‖

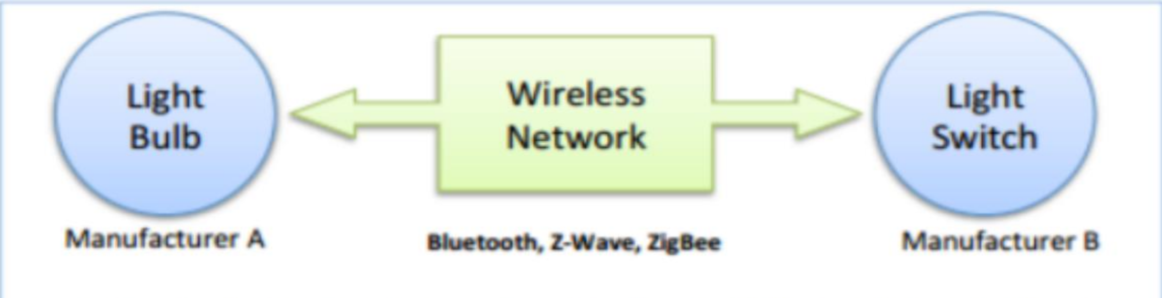
world of progress, efficiency, and opportunity, with the potential for adding billions in value to industry and the global economy.6 Others warn that the IoT represents a darker world of surveillance, privacy and security violations, and consumer lock–in. Attention-grabbing headlines about the hacking of Internet-connected automobiles,7 surveillance concerns stemming from voice recognition features in ―smart‖ TVs,8 and privacy fears stemming from the potential misuse of IoT data9 have captured public attention. This ―promise vs. peril‖ debate along with an influx of information though popular media and marketing can make the IoT a complex topic to understand. Fundamentally, the Internet Society cares about the IoT as it represents a growing aspect of how people and institutions are likely to interact with the Internet in their personal, social, and economic lives. If even modest projections are correct, an explosion of IoT applications could present a fundamental shift in how users engage with and are impacted by the Internet, raising new issues and different dimensions of existing challenges across user/consumer concerns, technology, policy and law. IoT also will likely have varying consequences in different economies and regions, bringing a diverse set of opportunities and challenges across the globe. This overview document is designed to help the Internet Society community navigate the dialogue surrounding the Internet of Things in light of the competing predictions about its promises and perils. It provides a high-level overview of the basics of IoT and some of the key issues and questions that this technology raises from the perspective of the Internet Society and the core values we promote.10,11 It also acknowledges some of the unique aspects of the Internet of Things that make this a transformational technology for the Internet. As this is intended to be an overview document, we do not propose a specific course of action for ISOC on IoT at this time. Rather, we see this document as an informational resource and starting point for discussion within the ISOC community on IoT-related issues. The Internet of Things is an emerging topic of technical, social, and economic significance. Consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way we work, live, and play. Projections for the impact of IoT on the Internet and economy are impressive, with some anticipating as many as 100 billion connected IoT devices and a global economic impact of more than $11 trillion by 2025. At the same time, however, the Internet of Things raises significant challenges that could stand in the way of realizing its potential

3.2 INTERNET OF THINGS COMMUNICATION MODELS

From an operational perspective, it is useful to think about how IoT devices connect and communicate in terms of their technical communication models. In March 2015, the Internet Architecture Board (IAB) released a guiding architectural document for networking of smart objects (RFC 7452),39 which outlines a framework of four common communication models used by IoT devices. The discussion below presents this framework and explains key characteristics of each model in the framework.

3.2.1 DEVICE TO DEVICE COMMUNICATION

The device-to-device communication model represents two or more devices that directly connect and communicate between one another, rather than through an intermediary application server. These devices communicate over many types of networks, including IP networks or the Internet. Often, however these devices use protocols like Bluetooth, Z-Wave, or ZigBee to establish direct device-to-device communications, as shown in Figure 1. These device-to-device networks allow devices that adhere to a particular communication protocol to communicate and exchange messages to achieve their function. This communication model is commonly used in applications like home automation systems, which typically use small data packets of information to communicate between devices with relatively low data rate requirements. Residential IoT devices like light bulbs, light switches, thermostats, and door locks normally send small amounts of information to each other (e.g. a door lock status message or turn on light command) in a home automation scenario. This device-to-device communication approach illustrates many of the interoperability challenges discussed later in this paper.



Source: Tschofenig, H., et. al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. Internet Architecture Board, Mar. 2015. Web. <https://www.rfc-editor.org/rfc/rfc7452.txt>.

Fig **3.2.1 Device to device communication**

As an IETF Journal article describes, ―these devices often have a direct relationship, they usually have built-in security and trust [mechanisms], but they also use device-specific data models that require redundant development efforts [by device manufacturers]‖.This means that the device manufacturers need to invest in development efforts to implement device-specific data formats rather than open approaches that enable use of standard data formats.

From the user's point of view, this often means that underlying device-to-device communication protocols are not compatible, forcing the user to select a family of devices that employ a common protocol. For example, the family of devices using the Z-Wave protocol is not natively compatible with the ZigBee family of devices. While these incompatibilities limit user choice to devices within a particular protocol family, the user benefits from knowing that products within a particular family tend to communicate well.

3.2.2 DEVICE TO CLOUD COMMUNICATION

In a device-to-cloud communication model, the IoT device connects directly to an Internet cloud service like an application service provider to exchange data and control message traffic. This approach frequently takes advantage of existing communications mechanisms like traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which ultimately connects to the cloud service. This is shown in Figure 2 This communication model is employed by some popular consumer IoT devices like the Nest Labs Learning Thermostat and the Samsung SmartTV. In the case of the Nest Learning Thermostat, the device transmits data to a cloud database where the data can be used to analyze home energy consumption. Further, this cloud connection enables the user to obtain remote access to their thermostat via a smartphone or Web interface, and it also supports software updates to the thermostat. Similarly, with the Samsung SmartTV technology, the television uses an Internet connection to transmit user viewing information to Samsung for analysis and to enable the interactive voice recognition features of the TV. In these cases, the device-to-cloud model adds value to the end user by extending the capabilities of the device beyond its native features However, interoperability challenges can arise when attempting to integrate device However, interoperability challenges can arise when attempting to integrate devices made by different manufacturers. Frequently, the device and cloud service are from the same vendor.46 If proprietary data protocols are used between the device and the cloud service, the device owner or user may be tied to a specific cloud service, limiting or preventing the use of alternative service providers. This is commonly referred to as ―vendor lock-in", a term that encompasses other facets of the relationship with the provider such as ownership of and access to the data. At the same time, users can generally have confidence that devices designed for the specific platform can be integrated

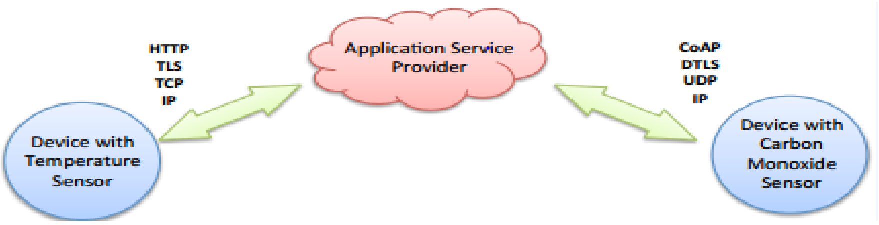

HTTP
TLS
TCP
IP
Application Service Provider
CoAP
DTLS
UDP
IP
Device with Temperature Sensor
Device with Carbon Monoxide Sensor

Fig 3.2.2 Device to cloud communication

3.3 IMPORTANCE OF IOT

IoT is regarded as the significant frontier that can improve almost all activities in our lives. Most of the devices, which have not previously been connected to the internet, can be networked and respond the same way as smart devices. By 2020, the world is set to be completely IoT oriented. Here are the benefits, which come with this technology.

- IoT promotes efficient resource utilization.
- It minimizes human efforts in many life aspects.
- Enabling IoT will reduce the cost of production and maximizing the returns
- It makes analytics decisions faster and accurately
- It boosts the real-time marketing of products
- Provide a better client experience

3.4 APPLICATIONS AND ADVANTAGES OF IOT

APPLICATIONS OF IOT:

These following are the applications of IOT.

- Water supply.
- Health.
- Traffic monitoring.
- Fleet management.
- Agriculture.
- Hospitality.
- Smart grid and energy saving.
- Smart home
- Cc cam controlling
- Door unlocking

ADVANTAGES OF IOT:

Internet of things facilitates the several advantages in day-to-day life in the business sector. Some of its benefits are given below.

- **Efficient resource utilization**: If we know the functionality and the way that how each device work, we definitely increase the efficient resource utilization as well as monitor natural resources.
- **Minimize human effort**: As the devices of IoT interact and communicate with each other and do lot of task for us, then they minimize the human effort.
- **Save time:** As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.
- **Enhance Data Collection**: Improve security: Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

Chapter no :4

PROBLEM DEFINITION & METHODOLOGY

4.1 PROBLEM DEFINITION

Today people are looking at ways and means to better their life-style using the latest technologies that are available. Any new facility for home appliance that promises to enhance their life-style is grabbed by the consumers. The more such facilities and appliances are added, it becomes inevitable to have easy and convenient methods and means to control and operate these appliances.

Conventional wall switches are located in different parts of a house and thus necessitates manual operations like to switch on or off these switches to control various appliances. It gets virtually impossible to keep track of appliances that are running and also to monitor their Performances. And Aim is to Build an system which controls home appliances with less efforts, like control
using mobile, or voice based controlled.

4.2 METHODOLOGY

METHODOLOGY:

- Make Connection as Per Circuit Diagram, Make Connection on NodeMCU.
- And Then Connect NodeMCU To the Wifi using hotspot/Router.
- Then Connect The NodeMCU pins Output To The Relay Driver Circuit.
- Then Start Programming the NodeMCU Module.
- Programme The NodeMCU Using Aurdino IDE Software.
- Download the Blynk Liberary zip File, install it from add liberary files.
- Downold the NodeMCU boards from preferences, by inserting the library link in it.
- Set the Output of NodeMCU (D0 – D14) For Different Control Function.

- Compile the Typed Programme check whether errors are occurred or not
- Upload the Programme onto NodeMCU using micro-type USB Cables.
- Then Connect the NodeMCU Module to the Internet using Router/Hotspot.
- Now Pair the NodeMCU Module with Android Application. i.e. Blynk App.

CONTROL TYPE:

➢ MANUAL CONTROL:

- Now Set the Function of Switches in Application.
- Checkout All the Connection First.
- Now to Test the Model

➢ VOICE CONTROL:

- Go To site IFTTT sign in Create the Google Assistant Applets (triggers for
controlling).
- Then after creating applets Configure those applets with WebHooks.
- Connect Blynk the Web-Hooks by creating the Trigger weblink to Blynk App.
- The link is available on net. just copy the link and add AuthToken and set the
pin Number.
- All set now just check the set-up with google assistant by giving com

Chapter no: 5

Overview of the project

5.1 INTRODUCTION:

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

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

5.2 circuit diagram:

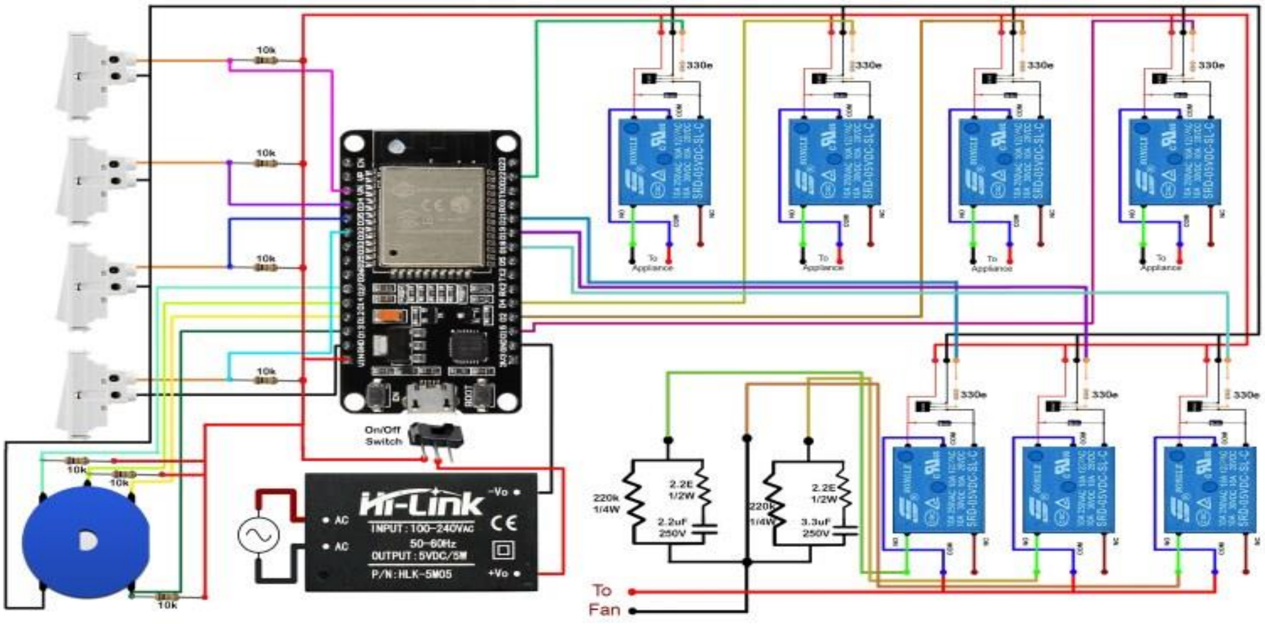


fig 5.2 circuit diagram

5.2.1 DESCRIPTION OF CIRCUIT DIAGRAM: In this project we learn how to make IoT Based Home Automation Project using Blynk & NodeMCU ESP8266. One of the most common & popular hobby projects you will come across the internet is Home Automation Project and this Home Automation System, we will control home appliances as Tv, Fan, Bulb, Motor, Refrigerator connected to Relay using Blynk Application. The Wifi Module NodeMCU ESP8266 will Receive commands from the smartphone wirelessly through the internet. To encode the ON/OFF signal and send it to Server and to ESP8266 Board we need the best IoT Platform. So, we chose Blynk as no other application can be better than this one. Using this circuit diagram, you can assemble the circuit on Breadboard using 4 channel Relay and NodeMCU Board. NODE MCU is very popular in-Home Automation. It’s WiFi capabilities and Arduino IDE support making it easier for IoT Applications. It is very tiny and has many Digital I/O pins, Serial Communication and I2C Communication. NodeMCU has a micro USB port to program it using your existing mobile cable (no additional programmer needed). There is a successor called ESP32 Development board which has more Analog pins and Digital pins. You can use any one of them for this project according to your requirements. Here we will be using NodeMCU and the relay driver is used to Basically, the output of microcontroller is in Mili-volts so this output is not sufficient to run the bulky load output. So as to run the appliances on 230v we require a Relay module so the output is fed to the relay module according to given input to the relay module it will generate output and drive various appliances and load e.g. Lamp, Fan, Tube light, T.V, etc. A relay is an electrical device which is generally used to control high voltages using very low voltage as an Input. This consists of a coil wrapped around a pole and a two small metal flaps(nodes) that are used to close the circuit. One of the nodes is fixed and other is movable. Whenever an electricity is passed through the coil, it creates a magnetic field and attracts the moving node towards the static node and the circuit gets completed. So, just by applying small voltage to power up the coil we can actually complete the circuit for the high voltage to travel. Also, as the static node is not physically connected to the coil there is very less chance that the Microcontroller powering the coil gets damaged if something goes wrong. And the hole operation is based on micro controller i.e. node MCU and the esp8266 first we give supply directly to load at one end and the output of controller is 3v and the relay is controlled all the load operation .and the controlling is done by BLYNK APP

5.2.2 COMPONENTS USED:

The following are the list of components used in the circuit design

- NODE MCU (wi-fi module)
- Relay -module
- Rectifier unit
- PCB board
- BC547 transistor
- 17007 diode
- 1kΩ resistors
- Connecting probes

5.3 NODE MCU

5.3.1 INTRODUCTION OF ESP8286:

The Node MCU (Node Micro Controller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK. That makes it an excellent choice for the Internet of Things (IoT) projects of all kinds However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate ana log voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

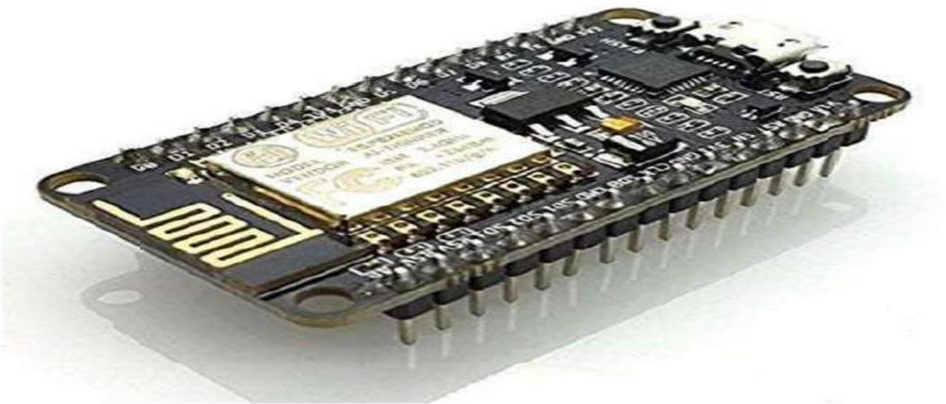


Fig. 5.3.1 Node MCU ESP8286

There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have Wi-Fi capabilities, and some even have a serial data port instead of a USB port

1. NodeMCU 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.…
2. The term NodeeMCU stands for Node Microcontroller unit It has open source hardware and software environment. NodeMCU is also called as Devkit 1.0. It is on board system on chip (SOC) called ESP-8266.The ESP8266 is a low-cost Wi-Fi micrrochip with full TCP/IP stack and microcontroller capability.
3. 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.
4. 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 flashh it without any trouble, just like Arduino. It is also immediately breadboard friendly.

5.3.2 ESP8266:

The chip first came to the attention of Western makers in August 2014 with the ESP01 module, made by a third-party manufacturer Ai-Thinker.

This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using Hayes-style commands.

However, at first there was almost no English-language documentation on the chip the commands it accepted.

The very low price and the fact that there were very few external components on the module, which suggested that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, chip, and the software on it, as well as to translate the Chinese documentation.

The ESP8285 is an ESP8266 with 1 MiB of built-in flash, allowing for single chip capable of connecting to Wi-Fi.

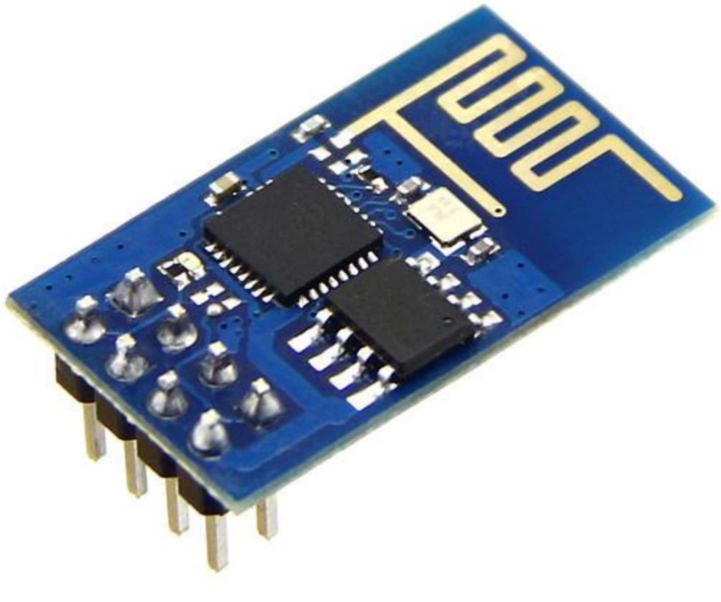


Fig 5.3.2 esp8266

5.3.3 PIN DIAGRAM OF ESP8286

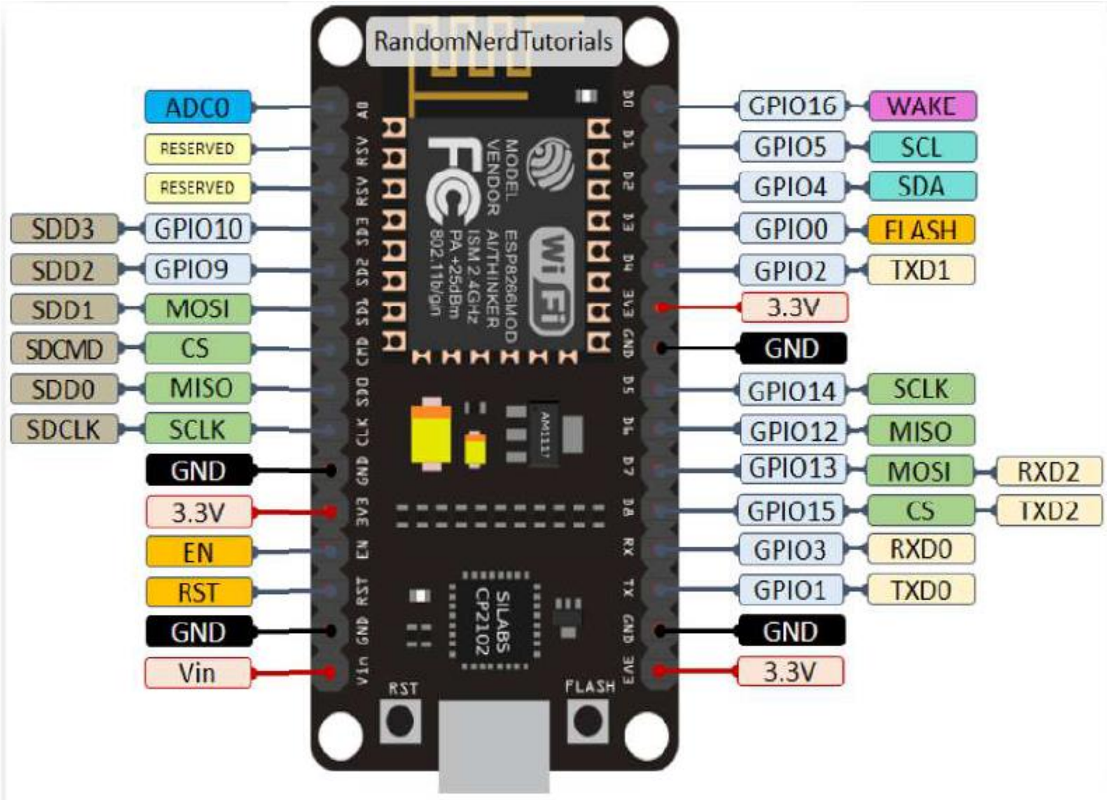


Fig 5.3.3 Pin Diagram of ESP8286

5.3.4 TYPES OF PINS

❖ **Power Pins**:

There are four power pins. VIN pin and three 3.3V pins.

VIN can be used to directly supply the Node MCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the Node MCU module – you can also supply 5V regulated to the VIN pin.

3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components.

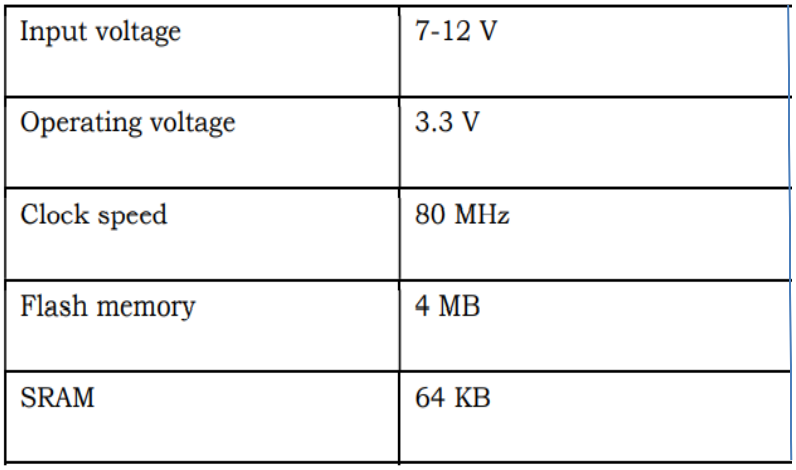
- ❖ **GND are the ground pins of Node MCU/ESP8266.**

15 I2C Pins are used to connect I2C sensors and peripherals. 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**: Node MCU/ESP8266 has 17 GPIO pins which can be assigned to 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. 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**: Node MCU/ESP8266 has 2 UART interfaces (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. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.
- ❖ **SPI Pins**: Node MCU/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:** Node MCU/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 up to 10000 us (100 Hz and 1 kHz).
- **Control Pins** are used to control the Node MCU /ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.
- **EN:** The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- **RST**: RST pin is used to reset the ESP8266 chip.
- **WAKE**: Wake pin is used to wake the chip from deep-sleep.

5.3.5 SPECIFICATIONS OF ESP8266:



Input voltage	7-12 V
Operating voltage	3.3 V
Clock speed	80 MHz
Flash memory	4 MB
SRAM	64 KB

TABLE 5.3.5 SPECIFICATIONS OF ESP8266

5.3.6 FEATURES AND APPLICATIONS:

Features:

- Open-source
- Arduino-like hardware

- Status LED
- Micro USB port
- Reset/Flash buttons
- Interactive and Programmable
- Low cost
- ESP8266 with inbuilt wi-fi
- USB to UART converter
- GPIO pins
- Arduino-like hardware IO

APPLICATIONS:

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

5.4 FOUR CHANNEL RELAY

5.4.1 INTRODUCTION: The 4 – Channel Relay module is a convenient board which can be used to control high voltage, high current load such as motor, solenoid valves, lamps and AC load. It is designed to interface with microcontroller such as Arduino. The relays terminal (COM, NO and NC) is being brought out with screw terminal. It also comes with a LED to indicate the status of relay. This is a 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.

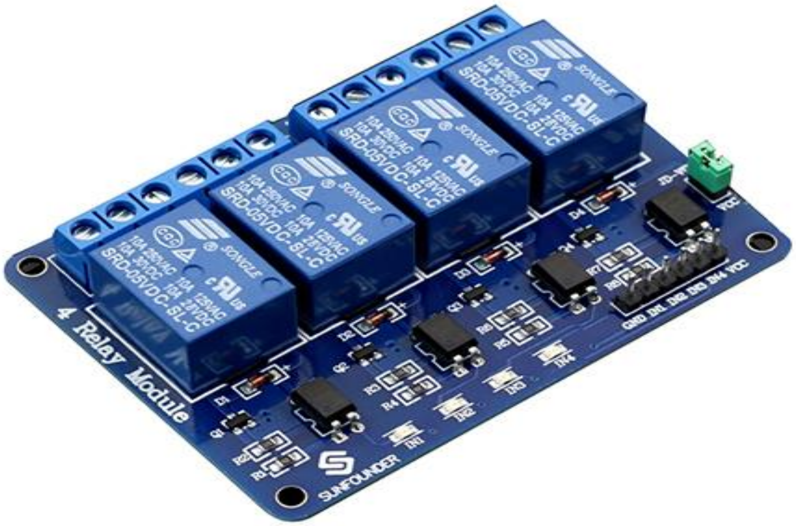


Fig 5.4 Four channel relay

5.4.2 PRINCIPLE:

From the picture below, you can see that 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, and 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.

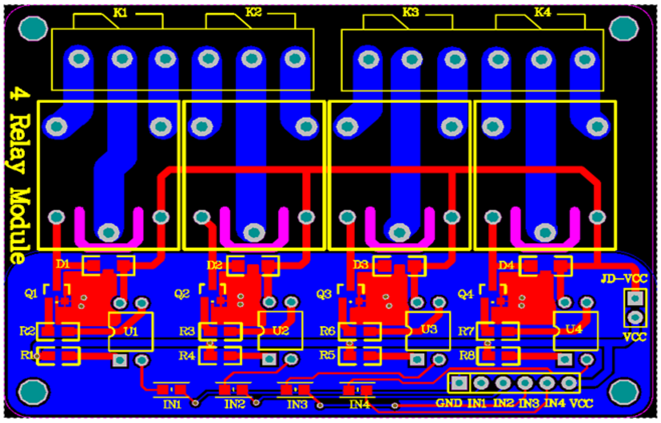
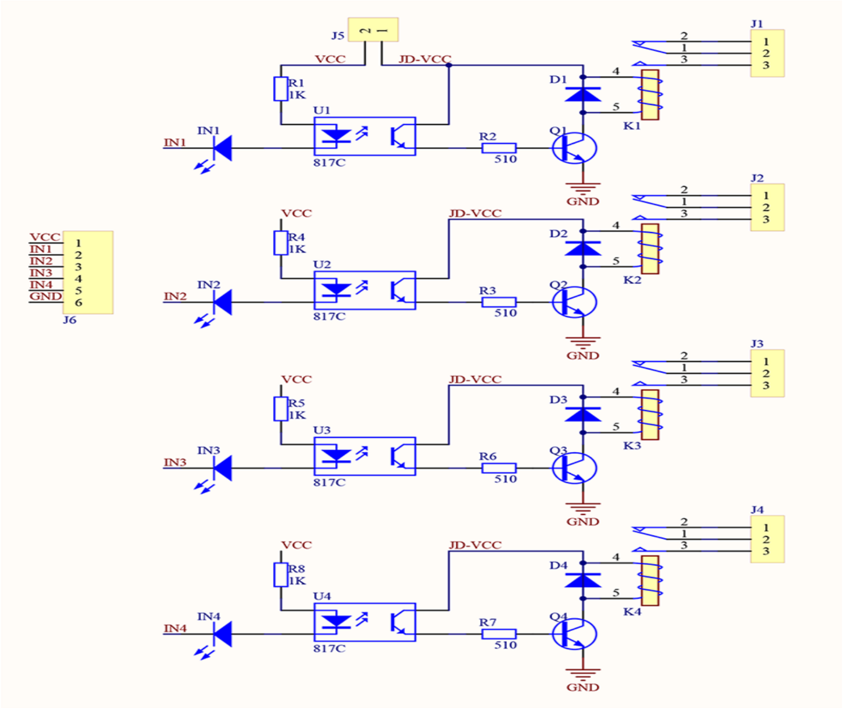


FIG 5.4.2 PRINCIPLE OF RELAY



5.4.3. PIN DISCRIPTION:

1) Input: 0-5 V
2) VCC: Positive supply voltage
3) GND: Ground
4) IN1--IN4: Relay control port
5) Output: supports various types of loads
6) Connect a load, DC 30V/10A · AC 250V/10A

5.4.4 FEATURES:

1. Size: 75mm (Length) * 55mm (Width) * 19.3mm (Height)
2. Weight: 61g
3. PCB Colour: Blue
4. There are four fixed screw holes at each corner of the board, easy for install and fix. The diameter of the hole is 3.1mm
5. High quality Single relay is used with single pole double throw, a common terminal, a normally open terminal, and a normally closed terminal
6. Optical coupling isolation, good anti-interference.
7. Closed at low level with indicator on, released at high level with indicator off
8. VCC is system power source, and JD_VCC is relay power source. Ship 5V relay by default. Plug jumper cap to use
9. The maximum output of the relay: DC 30V/10A, AC 250V/10A

5.5 POWER SUPPLY

5.51 INTRODUCTION:

Today almost every electronic device needs a DC supply for its smooth operation and they need to be operated within certain power supply limits. This required DC voltage or DC supply is derived from single phase ac mains

A regulated power supply can convert unregulated an AC (alternating current or voltage) to a constant DC (direct current or voltage). A regulated power supply is used to ensure that the output remains constant even if the input changes. A regulated DC power supply is also called as a linear power supply, it is an embedded circuit and consists of various blocks.

The regulated power supply will accept an AC input and give a constant DC output.

So, we have Powered the Kit/ set with the Adaptor Which gives exact 5V output voltage we have used the Micro-type USB cables for connection of Controller with Adaptor.

BLOCK DIAGRAM OF POWER SUPPLY:

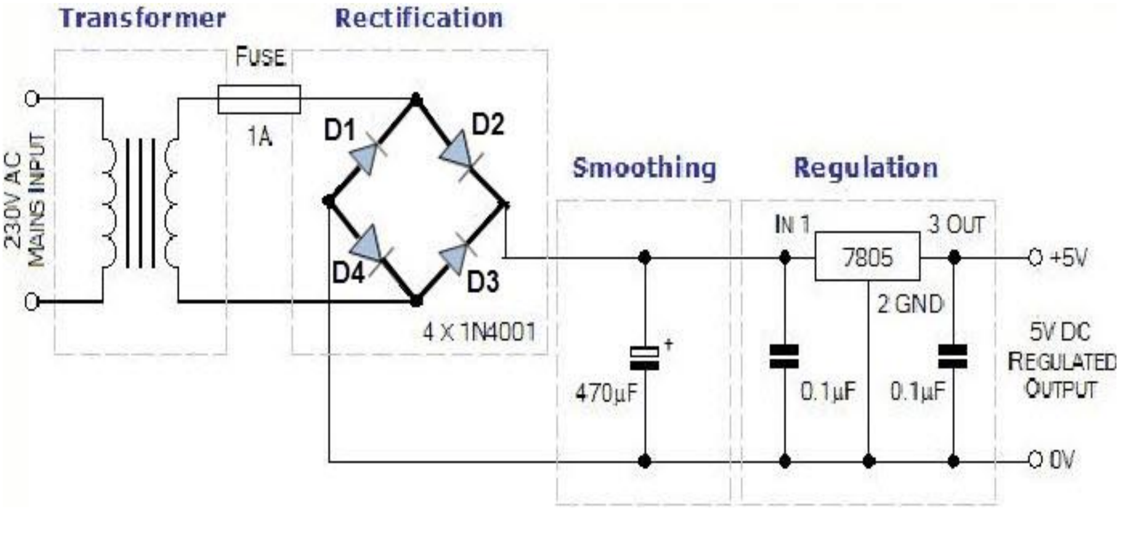


Fig no 5.51 BLOCK DIAGRAM OF POWER SUPPLY

DESCRIPTION:

1. A step down transformer
2. The basic building blocks of a regulated DC power supply are as follows:

A rectifier

3. A DC filter
4. A regulator

Step Down Transformer:

A step-down transformer will step down the voltage from the ac mains to the required voltage level. The turn's ratio of the transformer is so adjusted such as to obtain the required voltage value. The output of the transformer is given as an input to the rectifier circuit.

Rectification:

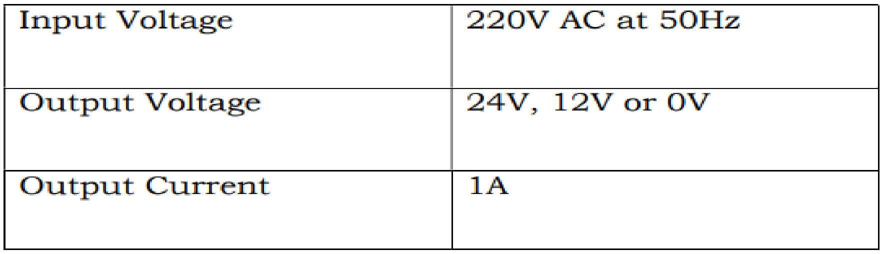
Rectifier is an electronic circuit consisting of diodes which carries out the rectification process. Rectification is the process of converting an alternating voltage or current into corresponding direct (DC) quantity. The input to a rectifier is ac whereas its output is unidirectional pulsating DC. Usually a full wave rectifier or a bridge rectifier is used to rectify both the half cycles of the ac supply (full wave rectification). Figure below shows a full wave bridge rectifier.'

DC Filtration:

The rectified voltage from the rectifier is a pulsating DC voltage having very high ripple content. But this is not we want, we want a pure ripple free DC

Regulation:

This is the last block in a regulated DC power supply. The output voltage or current will change or fluctuate when there is change in the input from ac mains or due to change in load current at the output of the regulated power supply or due to other factors like temperature changes. This problem can be eliminated by using a regulator. A regulator will maintain the output constant even when changes at the input or any other changes occur. Transistor series regulator, Fixed and variable IC regulators or a Zener diode operated in the Zener region can be used depending on their applications. IC's like 78XX and 79XX are used to obtained fixed values of voltages at the output. waveform. Hence a filter is used. Different types of filters are used such as capacitor filter. LC filter, choke input filter.



Input Voltage	220V AC at 50Hz
Output Voltage	24V, 12V or 0V
Output Current	1A

Table no 5.52 Specifications of Transformer

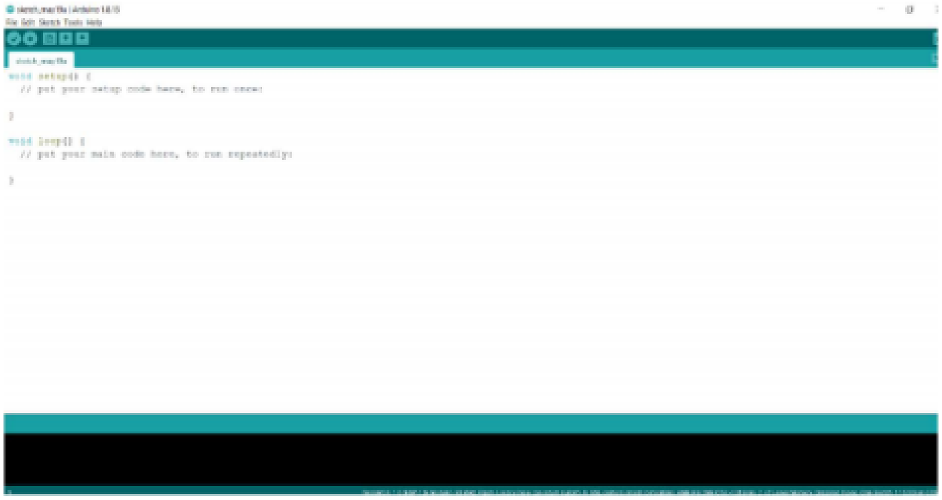
Chapter no: 6

ARDUINO IDE SOFTWARE AND CODING & BLYNK APP

6.1 ARDUINO IDE SOFTWARE

6.1.1 INTRODUCTION:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Arduino IDE is an open-source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

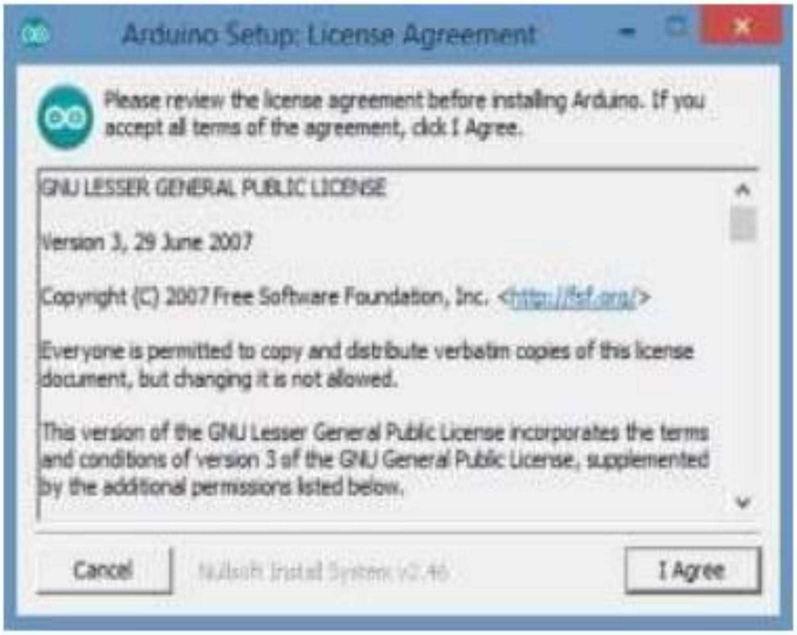
Fig no 6.1.1 Arduino IDE Software

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

6.1.2 PROCEDURE FOR INSTALLING ARDUINO IDE

- ❖ Visit http://www.arduino.cc/en/main/software to download the latest Arduino IDE version for your computer’s operating system. There are versions for Windows, Mac, and Linux systems. At the download page, click on the “Windows Installer” option for the easiest installation.

- ❖ Save the .exe file to your hard drive.
- ❖ Open the .exe file.
- ❖ Click the button to agree to the licensing agreement:


Arduino Setup: License Agreement
Cancel
Nullsoft Install System v2.46
I Agree

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

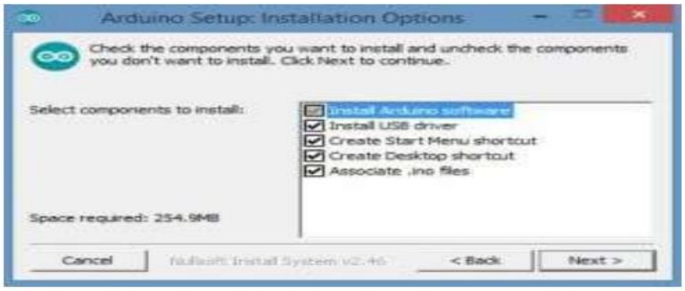
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

Fig 6.1.2 Arduino Setup: License Agreement

- Decide which components to install, then click “Next”:



Arduino Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

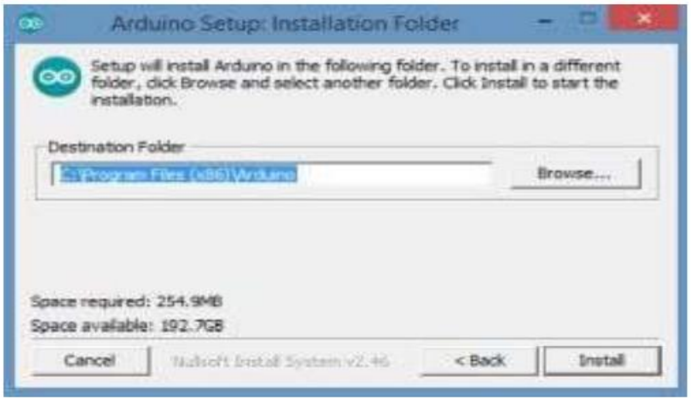
Select components to install:

☑ Install Arduino software
☑ Install USB driver
☑ Create Start Menu shortcut
☑ Create Desktop shortcut
☑ Associate .ino files

Space required: 254.9MB

Fig. No. 6.1.3 Arduino Setup: Installation Options

- Select which folder to install the program to, then click “Install”



Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

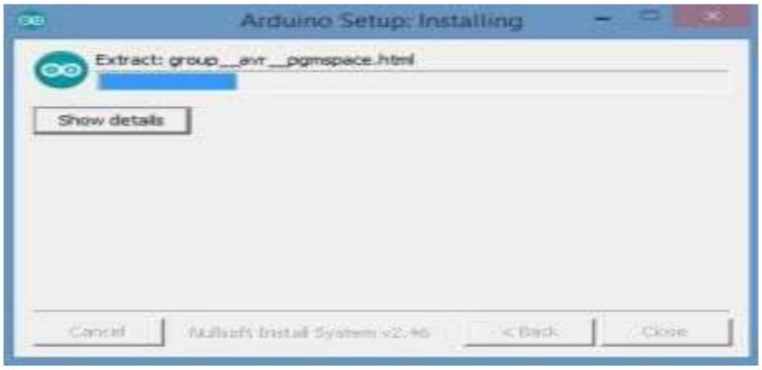
C:\Program Files (x86)\Arduino

Space required: 254.9MB

Space available: 192.7GB

Fig. No. 6.1.4 Arduino Setup: Folder

- Wait for the program to finish installing, then click “Close”:

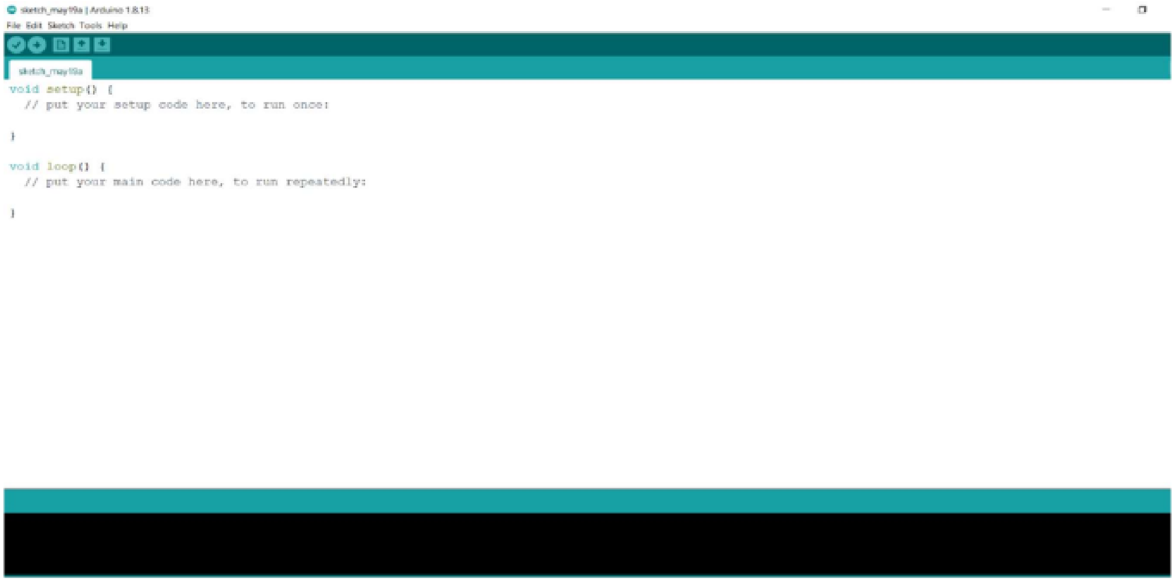


Arduino Setup: Installing

Extract: group__avr__pgmspace.html

Fig. No. 6.1.5 Arduino Setup: Installing

- ❖ Now find the Arduino shortcut on your Desktop and click on it. The IDE will open up and you'll see the code editor



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig. No. 6.1.6 Code editor

- ❖ Open your first project.

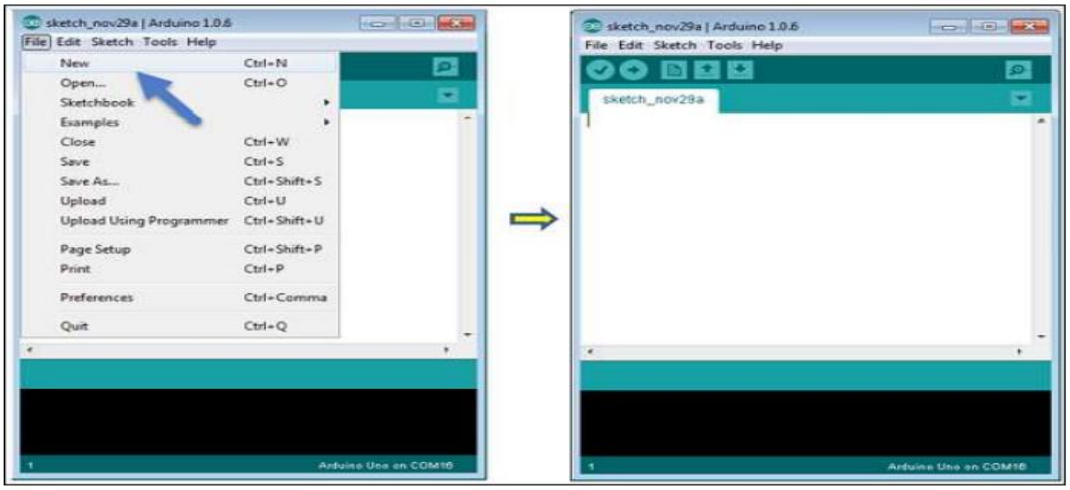


Fig. No. 6.1.7 Create new project

6.1.2 LIBRARIES USED

- Adafruit_Sensor-master.
- Blynk-library-master.
- DHT_sensor_library-1.3.8.
- ESP8266wifi-master.
- Simple Timer-master.

6.1.3 CODING

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
BlynkTimer timer;

#define DEBUG_SW 1

// Pins of Fan Regulator Knob
#define s1 27
#define s2 14
#define s3 12
#define s4 13

// Pins of Switches
#define S5 32
#define S6 35
#define S7 34
#define S8 39

// Pins of Relay (Appliances Control)
#define R5 15
#define R6 2
#define R7 4
#define R8 22

// Pins of Relay (Fan Speed Control)
#define Speed1 21
#define Speed2 19
```

```
#define Speed4 18

// By default, the mode is with_internet
int MODE = 0;

char auth [] = "AUTH_TOKEN";
char ssid [] = "SSID";
char pass [] = "PASS";

bool speed1_flag = 1;
bool speed2_flag = 1;
bool speed3_flag = 1;
bool speed4_flag = 1;
bool speed0_flag = 1;


int switch_ON_Flag1_previous_I = 0;
int switch_ON_Flag2_previous_I = 0;
int switch_ON_Flag3_previous_I = 0;
int switch_ON_Flag4_previous_I = 0;

BLYNK_WRITE(V0)
{
int fan_speed = param. asInt (); // assigning incoming value from pin V1 to a
variable
if (fan_speed == 0)
{
speed0();
}
if (fan_speed == 1)
{
speed1();
}
if (fan_speed == 2)
{
speed2();
}
if (fan_speed == 3)
{
speed3();
}
if (fan_speed == 4)
{
speed4();
}
}

BLYNK_WRITE(V1)
{
int pinValue = param. asInt (); // assigning incoming value from pin V1 to a variable
digitalWrite (R5, pinValue); // process received value
}

BLYNK_WRITE(V2)
{
int pinValue = param. asInt (); // assigning incoming value from pin V2 to a variable
digitalWrite (R6, pinValue); // process received value
}

BLYNK_WRITE(V3)
{
int pinValue = param. asInt (); // assigning incoming value from pin V3 to a variable
digitalWrite (R7, pinValue); // process received value
}

BLYNK_WRITE(V4)
{
int pinValue = param. asInt (); // assigning incoming value from pin V4 to a variable
digitalWrite (R8, pinValue); // process received value
}

void setup ()
{
// put your setup code here, to run once:
pinMode (s1, INPUT);
pinMode (s2, INPUT);
pinMode (s3, INPUT_PULLUP);
pinMode (s4, INPUT);
pinMode (S5, INPUT);
pinMode (S6, INPUT);
pinMode (S7, INPUT);
pinMode (S8, INPUT);

pinMode (R5, OUTPUT);
pinMode (R6, OUTPUT);
pinMode (R7, OUTPUT);
pinMode (R8, OUTPUT);
pinMode (Speed1, OUTPUT);
pinMode (Speed2, OUTPUT);
pinMode (Speed4, OUTPUT);

Serial.begin(9600);
WiFi.begin(ssid, pass);

timer.setInterval(3000L, checkBlynk); // check if connected to Blynk server every 3 seconds

Blynk.config(auth);//, ssid, pass);
}

void loop ()
{

if (WiFi.status()! = WL_CONNECTED)
{
if (DEBUG_SW) Serial.println("Not Connected");
}
else
{
if (DEBUG_SW) Serial.println(" Connected");
Blynk.run ();
}


timer.run (); // Initiates SimpleTimer
```

```
if (MODE == 0)
with_internet ();
else
without_internet ();
// put your main code here, to run repeatedly:
}

void with_internet ()
{
// FOR FAN
if (digitalRead(s1) == LOW && speed1_flag == 1)
{
speed1();
Blynk.virtualWrite(V0, 1);
speed1_flag = 0;
speed2_flag = 1;
speed3_flag = 1;
speed4_flag = 1;
speed0_flag = 1;
}
if (digitalRead(s2) == LOW && digitalRead(s3) == HIGH && speed2_flag == 1)
{
speed2();
Blynk.virtualWrite(V0, 2);
speed1_flag = 1;
speed2_flag = 0;
speed3_flag = 1;
speed4_flag = 1;
speed0_flag = 1;
}
if (digitalRead(s2) == LOW && digitalRead(s3) == LOW && speed3_flag == 1)
{
speed3();
Blynk.virtualWrite(V0, 3);
speed1_flag = 1;
speed2_flag = 1;
speed3_flag = 0;
speed4_flag = 1;
speed0_flag = 1;
}
```

```
if (digitalRead(s4) == LOW && speed4_flag == 1)
{
speed4();
Blynk.virtualWrite(V0, 4);
speed1_flag = 1;
speed2_flag = 1;
speed3_flag = 1;
speed4_flag = 0;
speed0_flag = 1;
}
if (digitalRead(s1) == HIGH && digitalRead(s2) == HIGH && digitalRead(s3) ==
HIGH && digitalRead(s4) == HIGH && speed0_flag == 1)
{
speed0();
Blynk.virtualWrite(V0, 0);
speed1_flag = 1;
speed2_flag = 1;
speed3_flag = 1;
speed4_flag = 1;
speed0_flag = 0;
}

// FOR SWITCH
if (digitalRead(S5) == LOW)
{
if (switch_ON_Flag1_previous_I == 0)
{
digitalWrite (R5, HIGH);
if (DEBUG_SW) Serial.println("Relay1- ON");
Blynk.virtualWrite(V1, 1);
switch_ON_Flag1_previous_I = 1;
}
if (DEBUG_SW) Serial.println("Switch1 -ON");

}
if (digitalRead(S5) == HIGH)
{
if (switch_ON_Flag1_previous_I == 1)
{
digitalWrite (R5, LOW);
```

```
if (DEBUG_SW) Serial.println("Relay1 OFF");
Blynk.virtualWrite(V1, 0);
switch_ON_Flag1_previous_I = 0;
}
if (DEBUG_SW) Serial.println("Switch1 OFF");
}


if (digitalRead(S6) == LOW)
{
if (switch_ON_Flag2_previous_I == 0)
{
digitalWrite (R6, HIGH);
if (DEBUG_SW) Serial.println("Relay2- ON");
Blynk.virtualWrite(V2, 1);
switch_ON_Flag2_previous_I = 1;
}
if (DEBUG_SW) Serial.println("Switch2 -ON");

}
if (digitalRead(S6) == HIGH)
{
if (switch_ON_Flag2_previous_I == 1)
{
digitalWrite (R6, LOW);
if (DEBUG_SW) Serial.println("Relay2 OFF");
Blynk.virtualWrite(V2, 0);
switch_ON_Flag2_previous_I = 0;
}
if (DEBUG_SW) Serial.println("Switch2 OFF");
//delay (200);
}

if (digitalRead(S7) == LOW)
{
if (switch_ON_Flag3_previous_I == 0)
{
digitalWrite (R7, HIGH);
if (DEBUG_SW) Serial.println("Relay3- ON");
Blynk.virtualWrite(V3, 1);
```

```
switch_ON_Flag3_previous_I = 1;
}
if (DEBUG_SW) Serial.println("Switch3 -ON");

}
if (digitalRead(S7) == HIGH)
{
if (switch_ON_Flag3_previous_I == 1)
{
digitalWrite (R7, LOW);
if (DEBUG_SW) Serial.println("Relay3 OFF");
Blynk.virtualWrite(V3, 0);
switch_ON_Flag3_previous_I = 0;
}
if (DEBUG_SW) Serial.println("Switch3 OFF");
//delay (200);
}

if (digitalRead(S8) == LOW)
{
if (switch_ON_Flag4_previous_I == 0)
{
digitalWrite (R8, HIGH);
if (DEBUG_SW) Serial.println("Relay4- ON");
Blynk.virtualWrite(V4, 1);
switch_ON_Flag4_previous_I = 1;
}
if (DEBUG_SW) Serial.println("Switch4 -ON");

}
if (digitalRead(S8) == HIGH)
{
if (switch_ON_Flag4_previous_I == 1)
{
digitalWrite (R8, LOW);
if (DEBUG_SW) Serial.println("Relay4 OFF");
Blynk.virtualWrite(V4, 0);
switch_ON_Flag4_previous_I = 0;
}
if (DEBUG_SW) Serial.println("Switch4 OFF");
```

```
//delay (200);
}


}


void without_internet ()
{

// FOR FAN
if (digitalRead(s1) == LOW && speed1_flag == 1)
{
speed1();

speed1_flag = 0;
speed2_flag = 1;
speed3_flag = 1;
speed4_flag = 1;
speed0_flag = 1;
}
if (digitalRead(s2) == LOW && digitalRead(s3) == HIGH && speed2_flag == 1)
{
speed2();

speed1_flag = 1;
speed2_flag = 0;
speed3_flag = 1;
speed4_flag = 1;
speed0_flag = 1;
}
if (digitalRead(s2) == LOW && digitalRead(s3) == LOW && speed3_flag == 1)
{
speed3();

speed1_flag = 1;
speed2_flag = 1;
speed3_flag = 0;
speed4_flag = 1;
speed0_flag = 1;
```

```
}
if (digitalRead(s4) == LOW && speed4_flag == 1)
{
speed4();

speed1_flag = 1;
speed2_flag = 1;
speed3_flag = 1;
speed4_flag = 0;
speed0_flag = 1;
}
if (digitalRead(s1) == HIGH && digitalRead(s2) == HIGH && digitalRead(s3) ==
HIGH && digitalRead(s4) == HIGH && speed0_flag == 1)
{
speed0();

speed1_flag = 1;
speed2_flag = 1;
speed3_flag = 1;
speed4_flag = 1;
speed0_flag = 0;
}


// FOR SWITCH
digitalWrite (R5, digitalRead(S5));
digitalWrite (R6, digitalRead(S6));
digitalWrite (R7, digitalRead(S7));
digitalWrite (R8, digitalRead(S8));
}

// Fan Speed Control

void speed0()
{
  //All Relays Off - Fan at speed 0
  if (DEBUG_SW) Serial.println("SPEED 0");
  digitalWrite (Speed1, LOW);
  digitalWrite (Speed2, LOW);
  digitalWrite (Speed4, LOW);
```

```
}

void speed1()
{
  //Speed1 Relay On - Fan at speed 1
  if (DEBUG_SW) Serial.println("SPEED 1");
  digitalWrite (Speed1, LOW);
  digitalWrite (Speed2, LOW);
  digitalWrite (Speed4, LOW);
  delay (1000);
  digitalWrite (Speed1, HIGH);
}

void speed2()
{
  //Speed2 Relay On - Fan at speed 2
  if (DEBUG_SW) Serial.println("SPEED 2");
  digitalWrite (Speed1, LOW);
  digitalWrite (Speed2, LOW);
  digitalWrite (Speed4, LOW);
  delay (1000);
  digitalWrite (Speed2, HIGH);
}

void speed3()
{
  //Speed1 & Speed2 Relays On - Fan at speed 3
  if (DEBUG_SW) Serial.println("SPEED 3");
  digitalWrite (Speed1, LOW);
  digitalWrite (Speed2, LOW);
  digitalWrite (Speed4, LOW);
  delay (1000);
  digitalWrite (Speed1, HIGH);
  digitalWrite (Speed2, HIGH);

}

void speed4()
{
  //Speed4 Relay On - Fan at speed 4
  if (DEBUG_SW) Serial.println("SPEED 4");
  digitalWrite (Speed1, LOW);
  digitalWrite (Speed2, LOW);
  digitalWrite (Speed4, LOW);
  delay (1000);
  digitalWrite (Speed4, HIGH);
}

//Check Blynk Connectivity every 3 seconds

void checkBlynk ()
{
  bool isconnected = Blynk.connected();
  if (isconnected == false)
  {
    MODE = 1;
  }
  if (isconnected == true)
  {
    MODE = 0;
  }
}
```

6.2 BLYNK APP

6.2.1 INTRODUCTION:

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 you can build a graphic interface for your project by simply dragging and dropping widgets.

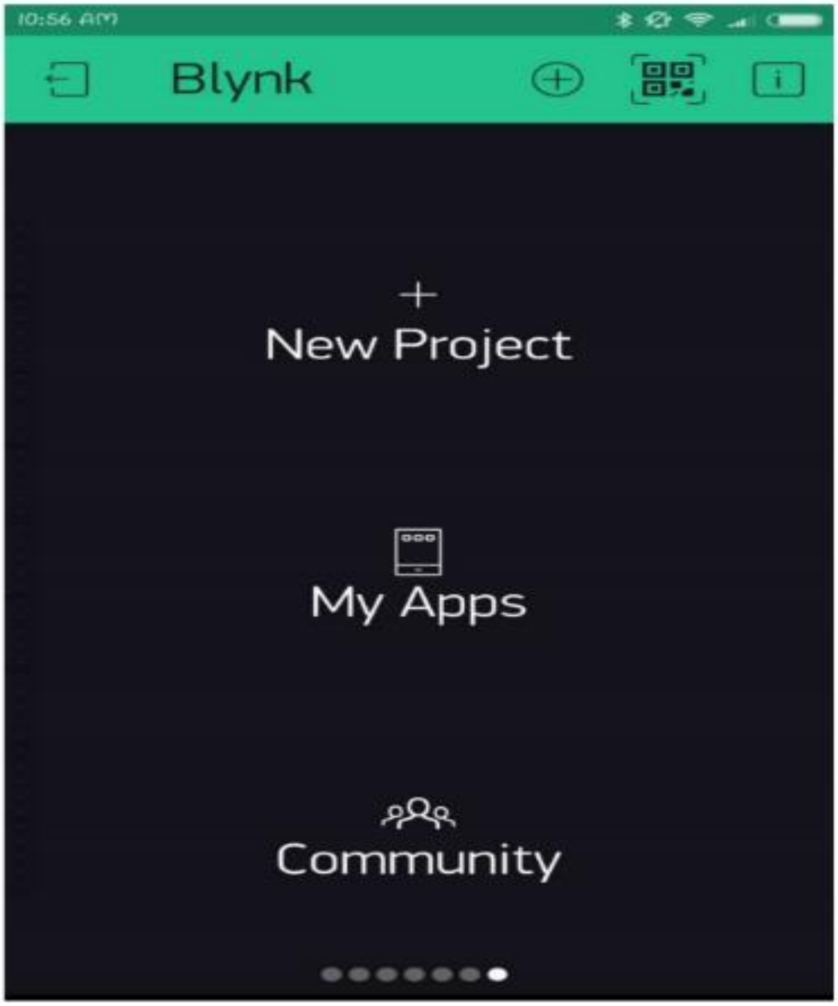


Fig. No. 6.2.1 Blynk App

6.2.2 PROCEDURE FOR INSTALLING AND CREATING IOT PROJECT

- Blynk application can be found from the following links
  - Android blynk app:

https://play.google.com/store/apps/details?id=cc.blynk

  - IOS blynk app:

https://apps.apple.com/us/app/blynk-control-

- After downloading the app, create an account and log in. (If possible than log in with your real mail id for better connectivity later).

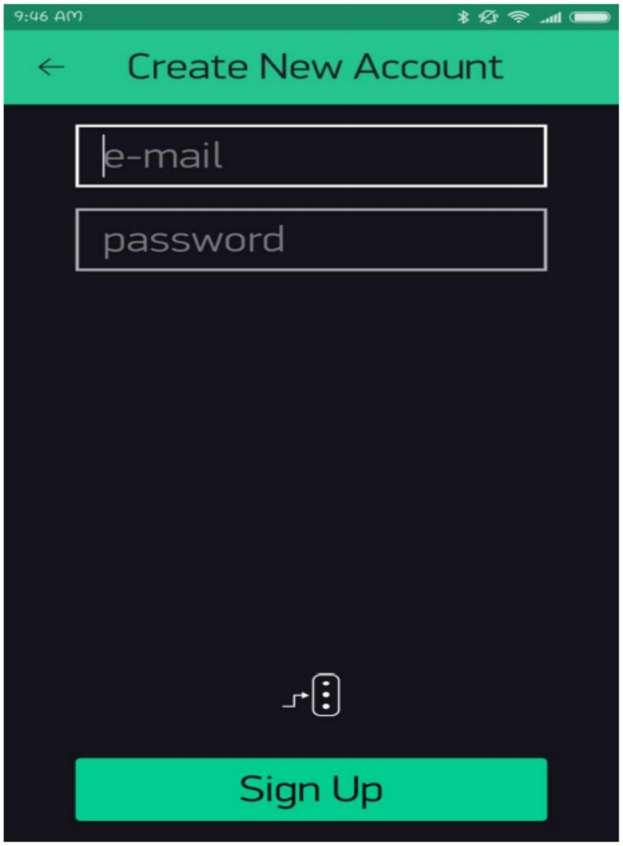
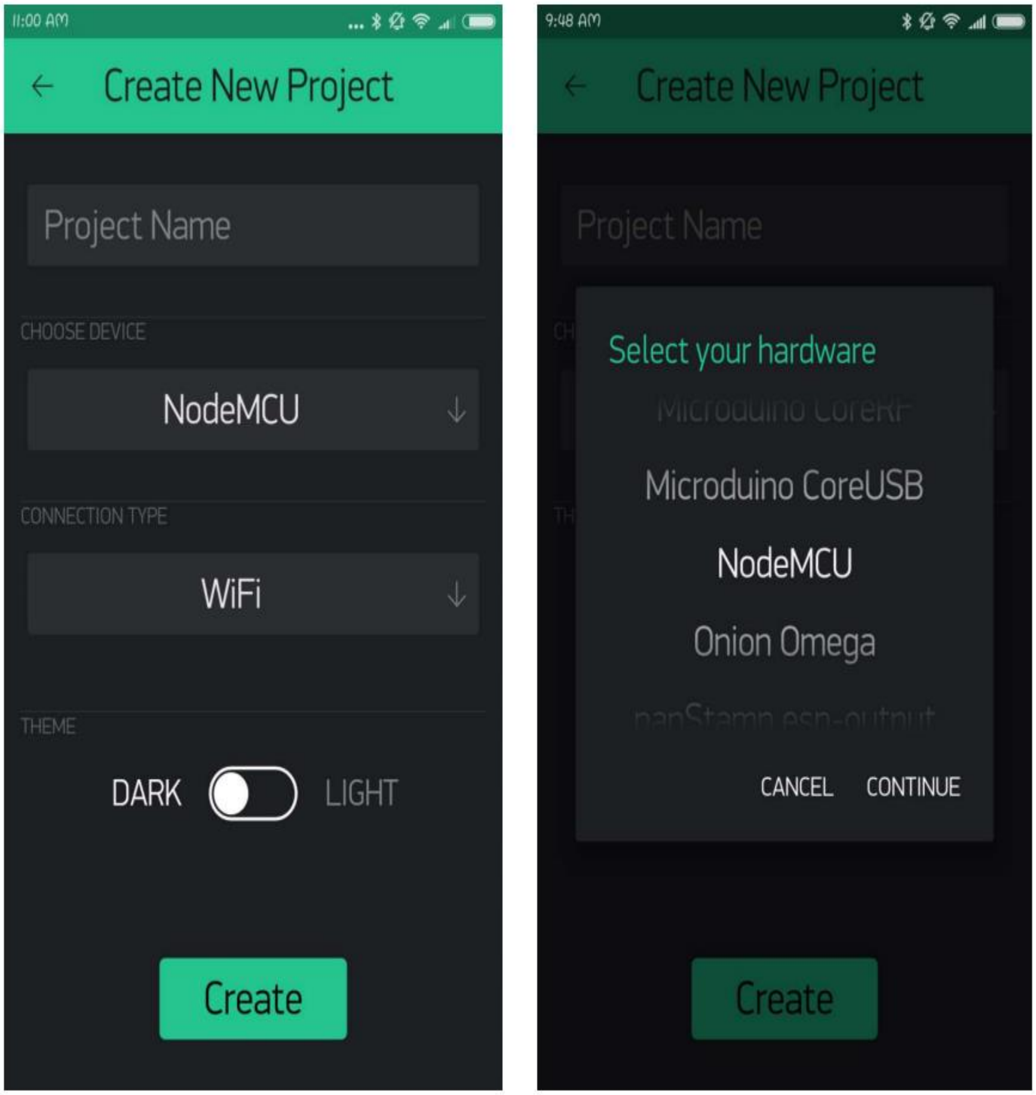


Fig. No. 6. 2 .2Creating account

- Click the “Create New Project” in the app to create a new Blynk app. Give it any name.

    - Blynk works with hundreds of hardware models and connection types. Select the Hardware type. After this, select connection type. In this project we have select Wi-Fi connectivity.



Create New Project

Project Name

CHOOSE DEVICE

NodeMCU

CONNECTION TYPE

WiFi

THEME

DARK LIGHT

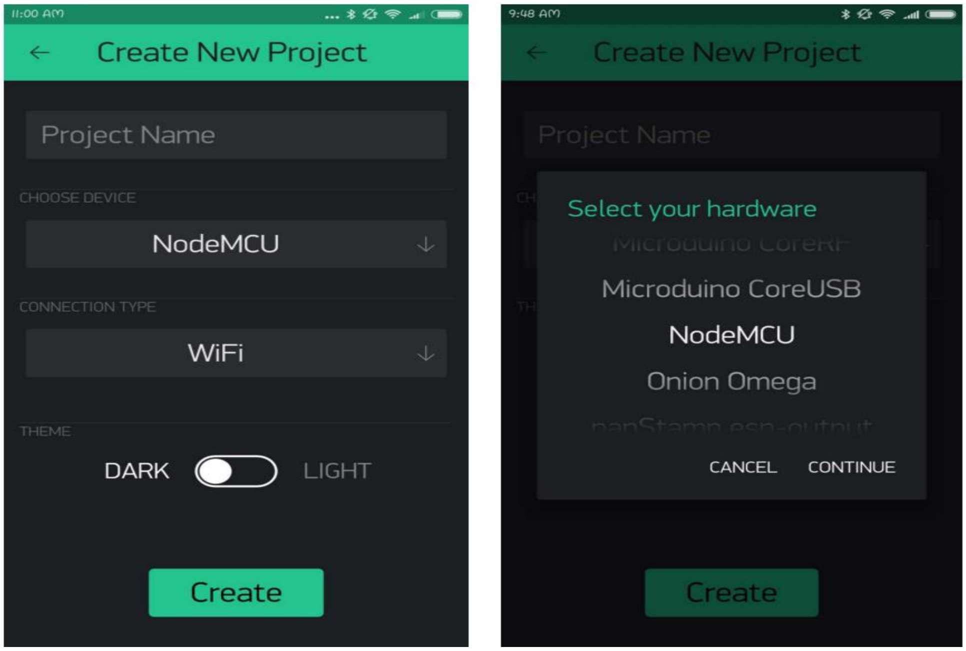
Fig. No. 6.2.3 Creating new project

- ❖ Add Widgets to The Project

Then you'll be presented with a blank new project. To open the widget box, click in the project window to open. We are selecting a button to control Led connected with Node MCU.

- ➢ Click on Button.
- ➢ Give name to Button say led.
- ➢ Under OUTPUT tab-
- ➢ Click pin and select the pin to which led is connected to Node MCU, here it is digital pin 2, hence select digital and under pin D2. And Click continue.

Under MODE tab- Select whether you want this button as “push button” or “Switch”.



CHOOSE DEVICE
NodeMCU

CONNECTION TYPE
WiFi

THEME
DARK LIGHT

Fig. 6.2.4 Add Widgets to The Project

- ❖ **Upload the Firmware**

Now that your Blynk project is set-up, open Arduino and navigate to the ESP8266_Standalone example in the File > Examples > Blynk > Boards Wi-Fi> ESP8266_Standalone menu.

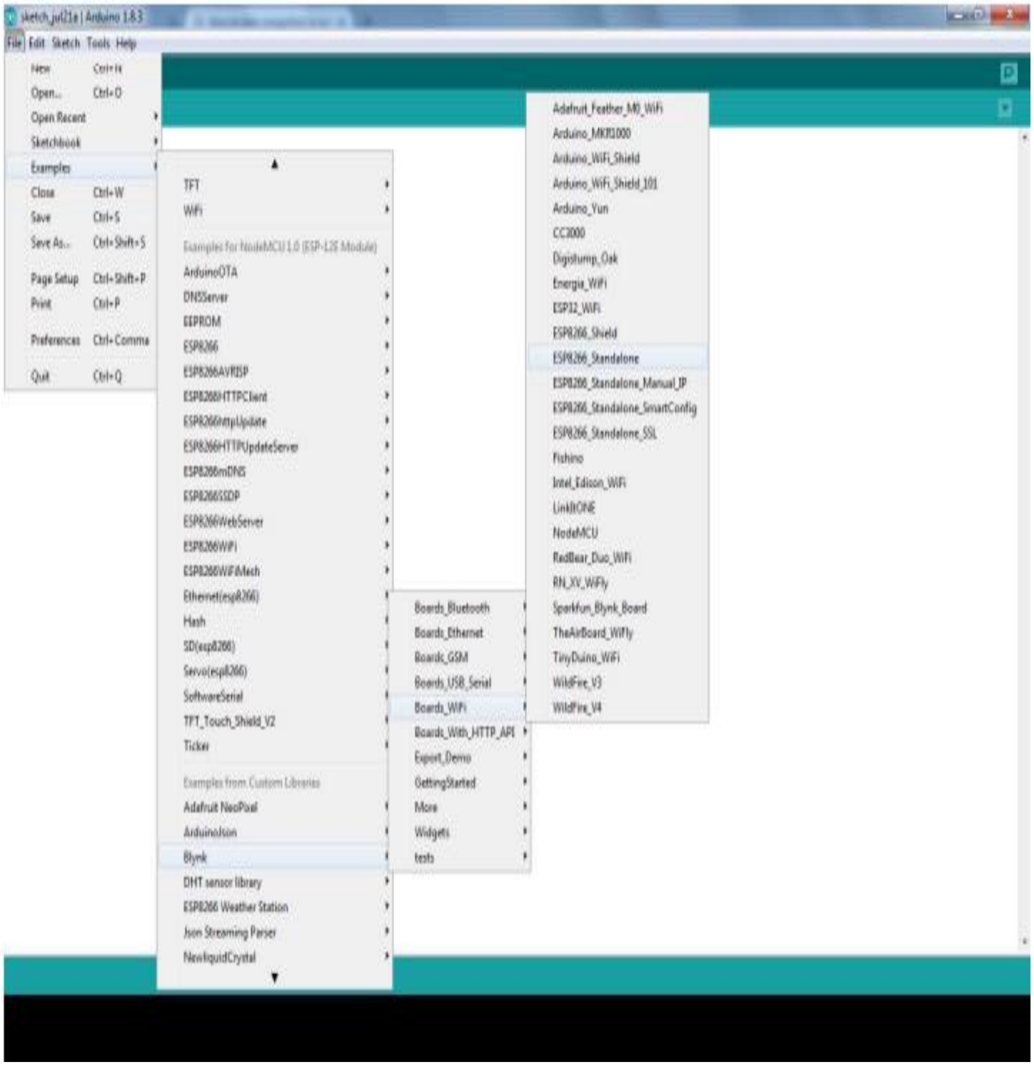


Fig. No. 6.2.5 Upload the Firmware

CHAPTER No: 7

Advantages and applications

ADVANTAGES

1. Savings: Smart thermostats and smart light bulbs save energy, cutting utility costs over time. Some home automation technologies monitor water usage, too, helping to prevent exorbitant water bills. Certain devices even offer rebates.

2. Convenience: Because home automation technology performs rote tasks automatically, end users experience great convenience. Lots of smart gadgets are compatible with one another, and you can set different triggers between devices to automate regular home processes. For instance, you could set your smart locks to turn on your smart lighting when you unlock the front door.

.

3. Control: Consumers also choose smart home devices to better control functions within the home. With home automation technology, you can know what's happening inside your home at all times.

4. Comfort: Some people use smart technology to record shows or to play music throughout the home. Connected devices can also help create a comfortable atmosphere they provide intelligent and adaptive lighting, sound, and temperature, which can all help create an inviting environment.

5. Peace of Mind: Finally, many consumers invest in home automation technology for peace of mind. A new mom or dad can check on their little one thanks to smart cameras and other technologies. Or, if you can't remember whether you closed the garage after you left, you can verify remotely with an app.

APPLICATIONS

1. Heating, ventilation and air conditioning (HVAC): it is possible to have remote control of all home energy monitors over the internet incorporating a simple and friendly user interface.

2. Lighting control system: A "smart" network that incorporates communication between various lighting system inputs and outputs, using one or more central computing devices.

3. Occupancy-aware control system: It is possible to sense the occupancy of the home using smart meters and environmental sensors like CO2 sensors, which can be integrated into the building automation system to trigger automatic responses for energy efficiency and building comfort applications.

4. Appliance control and integration: With the smart grid and a smart meter, taking advantage, for instance, of high solar panel output in the middle of the day to run washing machines.

5. Home robots and security: A household security system integrated with a home automation system can provide additional services such as remote surveillance of security cameras over the Internet, or access control and central locking of all perimeter doors and windows.

6. Leak detection: Smoke and CO detectors.

7. Home automation for the elderly and disabled.

8. Smart Kitchen and Connected Cooking: Using Voice control devices like Amazon Alexa or Google Home to kitchen appliances.

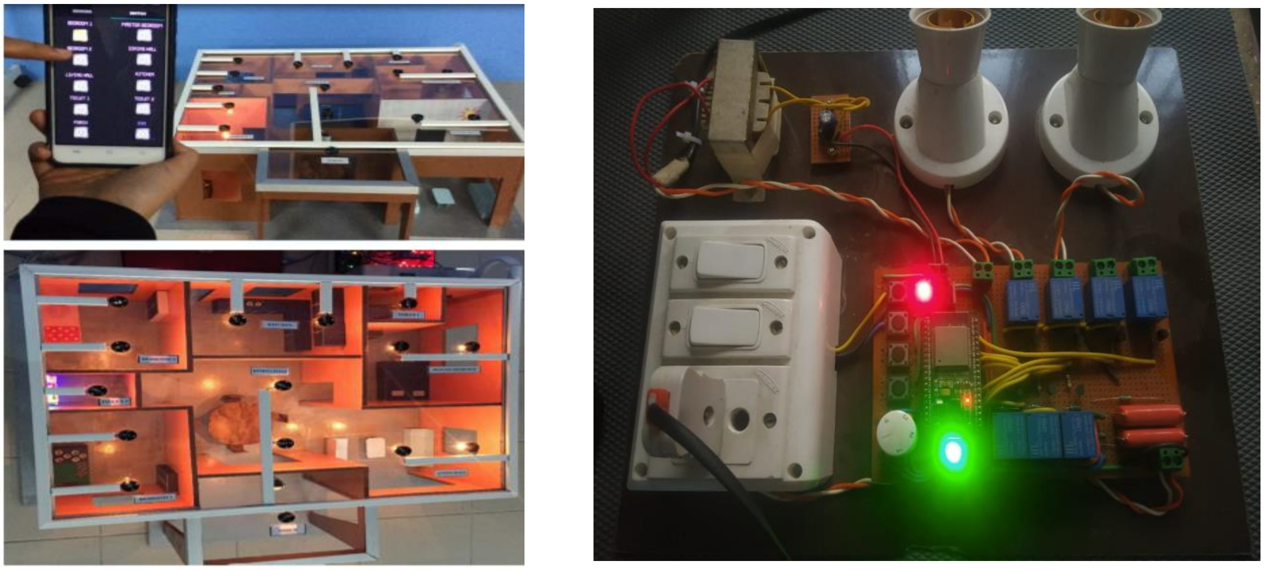
9.gas detector

Chapter No:8

RESULT AND DISCUSSIONS

This section presents an example to verify and validate the implementation of the proposed HAS. The preliminary results that we obtained in this paper will be used for further extension and enhancement This paper was managing to successfully develop a Wi-Fi-Based Automation System for Smart Home Using Android Mobile Phone.

Firstly, we need to connect our Android-Based smart phone with the available Wi-Fi. Then open Virtuino application in our mobile and fill in the IP address before connect. The IP address can be found in commend prompt in PC. Connect Virtuino application to Wi-Fi. Now we can control all the electrical appliances and able to monitor the motion, temperature and humidity of the house. The electrical appliances such as bulb and fan can be controlled and monitored using Android app which is Virtuino as demonstrated in Fig. 10. Further, the sensor DHT22 can measure and monitor the temperature and humidity in the house from the android-based mobile phone and these parameters can be used in the next step to fully automated the AC system and fans. The motion sensor also can detect if there is any motion in a certain area and give alarm via buzzer, which can be used later on in the security system or in automated the operation of lights.



FIGURES 8.1 : RESULT

At the end of this stage of our project, all the electrical appliances can be controlled by using mobile apk I.e. BLYNK APP

CONCLUSION

While wearing down this endeavor we have grabbed a lot of finding out about various modules being used in this errand. We are glad we can Built this Project as a part in this endeavour and set up new musings. We believe the assignment completes as needed and the data grabbed in the midst of this period will be used in our future corporate life. Additionally, we might want to include that home computerization is the fate of places of new world. Home automation is a resource which can make home environment Automated. People can control their electrical devices via. Smartphone These home automation devices and set-up controlling action through mobile. In future these products may have high potential for marketing.

FUTURE SCOPE

- The going with stage for home robotization advertise will happen subject to a couple of key overhauls in the progression open in Automation, for example, improvement in Wireless Automation blueprints and moreover bringing down of regard appears as the market starts perceive Home mechanization use in more noteworthy volumes. A couple of examples that we foresee for this time of the business are

- The going with stage for home robotization advertise will happen subject to a couple of key overhauls in the progression open in Automation, for example, improvement in Wireless Automation blueprints and moreover bringing down of regard appears as the market starts perceive Home mechanization use in more noteworthy volumes. A couple of examples that we foresee for this time of the business are

- Solution commitments will bit by bit move to an all the more straightforward structure, where next to two or three key parts, customers will have the ability to buy and use the Automation things themselves without the guide of any specific ace.
- Future Scope for the home automation system involves making homes even smarter. Homes can be interfaced with the sensors including the motion sensors, light sensors and temperature sensors and thus this may provide the automatic toggling of the devices according to the conditions.
- More energy can be conserved by ensuring occupation of the house before turning on devices and checking the brightness and turning off the light if not necessary.
- The system can be integrated closely with the home security solutions enhancing the safety for home owners

REFERENCES

1. E. S. A. Ahmed, “Internet of things applications, challenges and related future technologies,” *Internet of Things (IoT) Applications*, vol. 67, no. 2, pp. 126–148, 2017. At google scholar

2. R. Piyare, “Internet of things: ubiquitous home control and monitoring system using android based smart phone,” *International Journal of Internet of Things*, vol. 2, no. 1, pp. 5–11, 2013.View at: Publisher Site | Google Scholar

3. K. Mandula, R. Parupalli, C. A. Murty, E. Magesh, and R. Lunagariya, “Mobile based home automation using internet of things (IoT),” in *Proceedings of the 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 340–343, IEEE, Kumaracoil, India, December 2015.View at: Publisher Site | Google Scholar

4. B. Kang, S. Park, T. Lee, and S. Park, “IoT-based monitoring system using tri-level context making model for smart home services,” in *Proceedings of the 2015 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 198-199, IEEE, Las Vegas, NV, USA, January 2015.View at : Publisher Site | Google Scholar

5. G. Lobaccaro, S. Carlucci, and E. Löfström, “A review of systems and technologies for smart homes and smart grids,” *Energies*, vol. 9, no. 5, p. 348, 2016. View at : Publisher Site | Google Scholar

6. A. ElShafee and K. A. Hamed, “Design and implementation of a wifi based home automation system,” *World Academy of Science, Engineering and Technology*, vol. 68, pp. 2177–2180, 2012. View at : Publisher Site | Google Scholar

7. M. Fahim, I. Fatima, S. Lee, and Y. K. Lee, “Daily life activity tracking application for smart homes using android smartphone,” in *Proceedings of the 2012 14th International Conference on Advanced Communication Technology (ICACT)*, pp. 241–245, IEEE, PyeongChang, South Korea, February 2012 View at : Publisher Site | Google Scholar

8. A. S. Abdulraheem, A. A. Salih, A. I. Abdulla et al., “Home automation system based on IoT,” 2020. View at : Publisher Site | Google Scholar