DESIGN OF NodeMCU BASED HOME AUTOMATION SYSTEMS



SONARGAON UNIVERSITY (SU)

Supervised By

Nurul Ambia Alaul

Lecturer

Department of EEE

Sonargaon University (SU)

Submitted By

Md. Mamun Ahammed	**ID: EEE1602008209**
Md. Ainul Islam	**ID: EEE1602008133**
Md. Razib Khan	**ID: EEE1602008157**
Md. Mehedi Hasan	**ID: EEE1602008134**
Md. Jahangir alam	**ID: EEE1602008210**

Department of Electrical & Electronic Engineering (EEE)
Sonargaon University (SU)
147/I, Panthapath,
Dhaka-1215, Bangladesh

Date of Submission: January 2020

Declaration

It is declared hereby that this thesis paper or any part of it has not been submitted to anywhere else for the award of any degree.

....

Md. Mamun Ahammed **Md. Ainul Islam**

....

Md. Razib Khan **Md. Mehedi Hasan**

.....

Md. Jahangir alam

Under Supervision of

.....

Nurul Ambia Alaul

Lecturer

Department of EEE

Sonargaon University (SU)

Certification

Supervisor

.....

Nurul Ambia Alaul

Lecturer

Department of Electrical and Electronic Engineer (EEE)

Sonargaon University (SU)

ACKNOWLEDGEMENT

All thanks to Almighty ALLAH, the creator and the owner of this universe, the most merciful, beneficent and the most gracious, who provided us guidance, strength and abilities to complete this research.

We are especially thankful to Nurul Ambia Alaul, our thesis supervisor, for his help, guidance and support in completion of my project.

We also thankful to the Sonargaon University Faculty Staffs of the Bsc in EEE, who have been a light of guidance for us in the whole study period at Sonargaon University, particularly in building our base in education and enhancing our knowledge.

Finally, we would like to express our sincere gratefulness to our beloved parents, brothers and sisters for their love and care. We are grateful to all of our friends who helped us directly or indirectly to complete our thesis.

ABSTRACT

This project is developed for controlling home appliances through internet in real time and also to operate them in different modes from any remote places. Our main focus is to operate the home appliances in our desired modes and control them in an easy way. We have developed an android based application for controlling appliances and operate them in different modes like optimum, sleeping, auto etc. also all the appliances can be controlled individually. Different sensors have been used for analyzing lights, temperature and motion. Based on sensor values our algorithm will operate all the appliances according to modes. So by operating them in different modes it will save power as well as make our life easier and comfortable.

TABLE OF CONTENTS

Topics	Page No
ACKNOWLEDGEMENT	iv
ABSTRACT	v
CHAPTER NAME	vi-vii
LIST OF FIGURES	viii
TABLES	ix
LIST OF ABBREVIATIONS	

CHAPTER 1: INTRODUCTION

TOPICS		Page No.
1.1	Introduction	10
1.2	Motivation	11
1.3	Proposed System	12
1.4	Advantages	13
1.5	Objectives	

CHAPTER 2: Brief introduction of all the components

2.1	Introduction	14
2.2	Node MCU	15
2.2.1	Servo Motor	16
2.2.2	Relay Module 5V-8 Channel	17
2.2.3	DHT11 SENSOR	18
2.2.4	IR SENSOR	19
2.2.5	Buck Converter	19
2.2.6	Bread board	20
2.2.7	Adapter	20
2..2.8	DC Motor	21
2.2.9	LED Light	21
2.2.10	Battery	22

CHAPTER 3: DESIGN AND FABRICATION

3.1	Introduction	23
3.2	Block Diagram	23
3.3	Circuit Diagram	26
3.4	Overview Of the project	26
3.5	Image Of the project	27
3.6	Used Materials	28
3.7	Flow Chart	28
3.8	Conclusion	28

CHAPTER 4: RESULT & DISCUSSION

4.1	Introduction	29
4.2	Result	29
4.3	Advantages of Home Automation	30
4.4	Disadvantages of smart grid	30
4.5	Limitation	30

CHAPTER 5: CONCLUSION

5.1	Conclusion	31
5.2	Limitation	31
5.3	Future Scop	31
References		32
Appendix		33-40

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
1.3.1	An example of home automation system.	12
2.2	Node MCU-32	15
2.2.1	Survo Motor	16
2.2.2.1	8 pin relays	17
2.2.3.1	DHT11 SENSOR	18
2.2.4.1	IR Sensor	19
2.2.5.1	Buck Converter	19
2.2.6.1	Breadboard	20
2.2.7.1	Adapter	20
2.2.8.1	DC Motor	21
2.2.9.1	LED light	21
2.2.10.1	Battery	22
3.2	Block Diagram of project	23
3.2.1	Node MCU Circuit	24
3.2.2.1	Top View of 8-Channel Relay	24
3.2.2.2	Blynk App Screen Short	25
3.3	Circuit Diagram of this project	26
3.4	Overview of Project	26
3.5	Top view of the home automation system	27
3.7	Flow Chart Of the project	28
4.2	View Of the project	28

LIST OF TABLE

Table.2.2	Node MCU Specification	15
Table.2.2.1	DHT sensor Specifications	18

LIST OF ABBREVIATIONS

DC	Direct Current
AC	Alternating Current.
IR	Infrared Receiver
DTH	Digital Temperature and Humidity sensor
LED	Light Emitting Diode
IC	Integrated Circuit

CHAPTER I

INTRODUCTION

1.1 Introduction

The main objective of this project is to develop a home control system using Node MCU board with Internet being remotely controlled by any Android system. As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, involving remote controlled switches. Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a most modern solution with smart phones.

A. Automatic Control In order to achieve this, a relay module is interfaced to the Node MCU board at the receiver end while on the transmitter end, a GUI application on the cell phone sends ON/OFF commands to the receiver where loads are connected. By touching the specified location on the widgets, the AC loads can be switched ON/OFF remotely through this technology.

B. Manual Control It can be achieved by connecting the switches in between the Node MCU and the relay board. So that the output load will be HIGH for any HIGH signal coming either from Node MCU or from manual switches. It will be more advantageous in case of damage to the Node MCU.

C. Also we can be controled this device with Remote control by connecting the IR sensor.

1.2 Motivation

The world has changed after the evolution of term called Automation. In every sector automation is reducing human labor. Firstly introduced in industries and now in a variety of sectors rely on automation system. Development of advanced home automation system has huge opportunity in present time. Again Internet of Things (IoT) is going to be a new era of technology in future. It is expected that 30 billion of devices will be connected within 2020[1]. So it is the right to get prepared for the future technologies.

1.3 Proposed System

This wireless Node MCU system includes controlling of home appliances like light, fan, television, room temperature system. This project proposes remotely controlling of home appliances with security of home both inside and outside. The project is composed of:

- Controlling of appliances like light, table fan, TV, etc will be controlled with Android phone through Wireless module. This is used inside and outside the house. For this we have chosen Node MCU model 32 will controlled the device.
- The system will show room temperature. For this we have chosen DHT11 will show the temperature and Humanity.
- Servo Motor used for door closer.
- IR sensor used for remote control the device.
- Used buck converter model 2596 for convert 5V to 3V for fan motor use purpose.
- Project Board for supply controlling.
- Charging Module
- Relay Module with 8 channel
- LED Light
- DC motor for fan
- Battery 18650 for Node MCU backup.

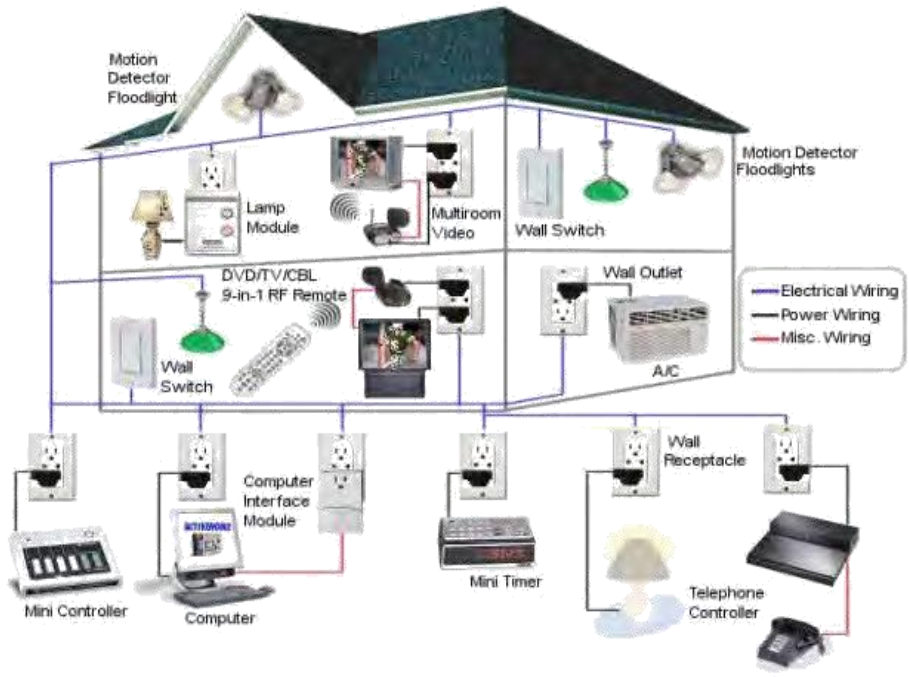


Figure 1.3.1: An example of home automation system.

1.4 Advantages:

In the present day home automation has becoming essential for the purpose of improving life style.Home automation offers a futuristic way of life in which an individual gets to control the entire house using a smart phone, from turning on a TV to locking or unlocking doors. It also offers an efficient use of energy. Automation system also allows us to control the home appliances and keep an eye on the house from a distance.

Some Advantages:

- Managing all of your home devices from one place.
- Flexibility for new devices and appliances.
- Maximum home security.
- Remote control of home functions.
- Increased energy efficiency.
- Improved appliance functionality.
- Home management insights.

1.5 Objectives:

In determining the purpose and direction of this project, there are several objectives that need to be accomplished in developing smart house.The project aims to achieve the following objectives:

- To investigate.
- To study smart house system.
- To apply.
- To maintain hundred present security system.
- To design smart house system.
- To understand and consider three factors of safety life time modeling for smart house system.

CHAPTER 2

Brief introduction of all the components

2.1 Introduction:

To begin with the project, let's get the idea of all the components that we used for the project. It is very important to know all the information about both hardware and software specifications.

The components we are using are as follows:

1.Node MCU-Model 32

2.Sensors:

(a)DHT11 for Temperature and Humanity sensor

(b)IR sensor for remote control

3.Servo motor SG90 for Door closer.

4.Buck Converter for convert from 5V to 3V.

5.Relay Module (8 Channel)

6.Breadboard

7.Adapter

8.Servo Motor

9.DC motor for fan

10.LED light

11Battery-18650

2.2.1 Node MCU-32

ESP32 is a series of low cost, low power system on a chip microcontroller with integrated Wi-Fi & dual-mode Bluetooth. It is designed to achieve the best power and RF performance, robustness, versatility, and reliability in a wide variety of applications such as voice encoding, music streaming and MP3 decoding,NodeMCU is an open source LUA based firmware developed for ESP8266 Wi-Fi chip.by exploring functionality with ESP8266 Development board/kit i.e. NodeMCU Development board.Since NodeMCU is open source platform, their hardware design is open for edit/modify/build.NodeMCU Development kit/board consist of ESP8266 WIFI enable chip. The ESP8266 is a low-cost Wi-Fi chip Development by Expressive system.

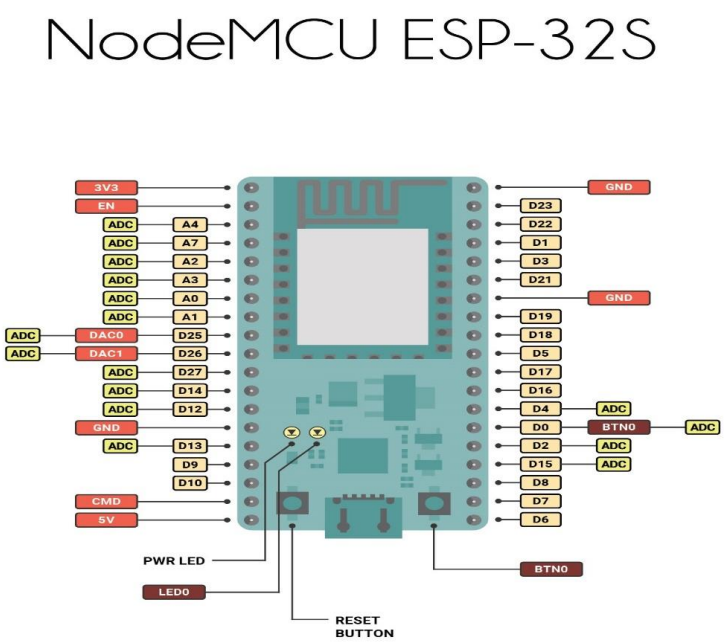


Figure 2.2 :Node MCU-32

Technical specifications:

Specifications	ESP32
MCU	Duale-Core 32 bit LX6 (600DMIPS0
Bluetooth	Bluetooth 4.2 and bellow
Typical Frequency	160 MHz
SRAM	512 KBytes
Flash	SPI Flash , up to 16 MBytes
Hardware/Software PWM	1/16 Channels
Touch Sensor	Yes
Temperature Sensor	Yes
Working Temperature	-40*C – 125*C
Power Supply	5 V

Table 2.1: NodeMCU Specification

2.2.2 Servo Motor

A **servo motor** is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through **servo mechanism.** If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

Servo Mechanism:

It consists of three parts:

Controlled device

Output sensor

Feedback system

It is a closed loop system where it uses positive feedback system to control motion and final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal. Here reference input signal is compared to reference output signal and the third signal is produces by feedback system. And this third signal acts as input signal to control device. This signal is present as long as feedback signal is generated or there is difference between reference input signal and reference output signal. So the main task of servomechanism is to maintain output of a system at desired value at presence of noises.

Working principle of Servo Motors: A servo consists of a Motor (DC or AC), a potentiometer, gear assembly and a controlling circuit. First of all we use gear assembly to reduce RPM and to increase torque of motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now difference between these two signals, one comes from potentiometer and another comes from other source, will be processed in feedback mechanism and output will be provided in term of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with potentiometer and as motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.



Figure: 2.2.1 Survo Motor

2.2.3 Relay Module 5V-8 Channel

This is a 8-channel LOW level trigger relay and it can be applied to Arduino and Raspberry Pi. Relays are suitable for driving high power electronic devices such as lights, electric fans and air condition. A relay can be used to control high voltages with a low voltage by connecting it to an MCU.

Principle of relay: 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. Add a certain 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 contact (or you may say releasing the former and the normally closed 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.

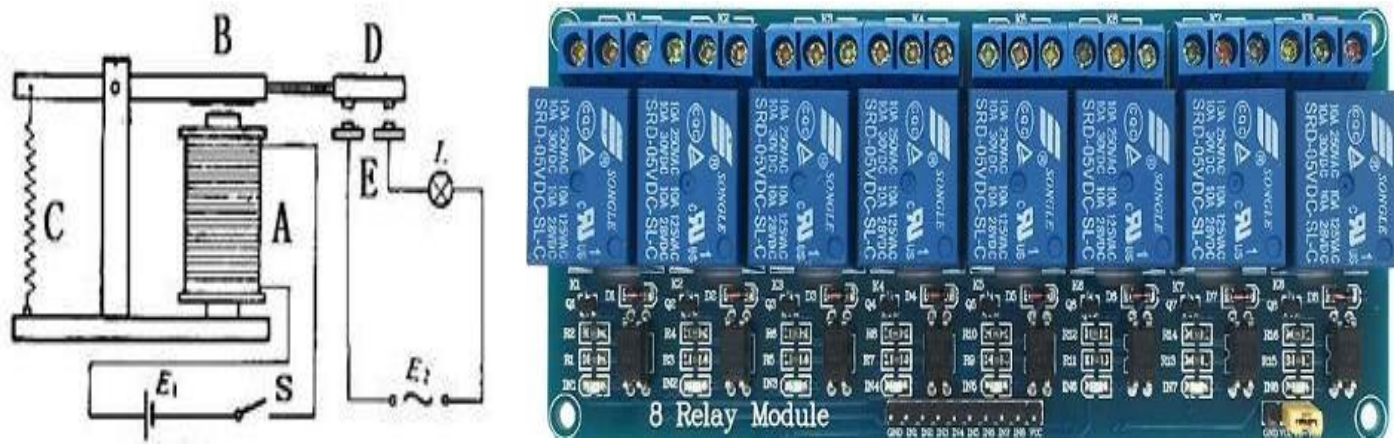


Figure: 2.2.2.1: 8 pin relays

2.2.4 DHT11 SENSOR

DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc to measure humidity and temperature instantaneously. DHT11 humidity and temperature sensor is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor. To measure the surrounding air this sensor uses a thermistor and a capacitive humidity sensor.

Working Principle of DHT11 Sensor: DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form. For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers. The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz.i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA. DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

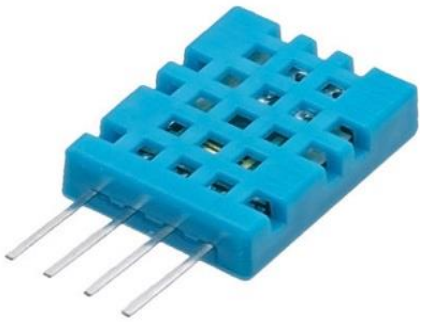


Figure: 2.2.3.1: DHT11 SENSOR

Applications:

This sensor is used in various applications such as measuring humidity and temperature values in heating, ventilation and air conditioning systems. Weather stations also use these sensors to predict weather conditions. The humidity sensor is used as a preventive measure in homes where people are affected by humidity. Offices, cars, museums, greenhouses and industries use this sensor for measuring humidity values and as a safety measure.

Specifications	DHT11
Operating Voltage	3 to 5V
Operating Voltage	2.5mA max
Humidity Range	20-80% / 5%
Temperature Range	0-50°C / ± 2°C
Sampling Rate	1 Hz (reading every second)
Body size	15.5mm x 12mm x 5.5mm
Advantage	Ultra low cost

Table: 2.2.1: DHT sensor Specifications

2.2.5 IR SENSOR

IR Sensor Working Principle: An infrared sensor circuit is one of the basic and popular sensor module in an electronic device. This sensor is analogous to human's visionary senses, which can be used to detect obstacles and it is one of the common applications in real-time. This circuit comprises of the following components

- LM358 IC 2 IR transmitter and receiver pair
- Resistors of the range of kilo-ohms.
- Variable resistors.
- LED (Light Emitting Diode).

In this project, the transmitter section includes an IR sensor, which transmits continuous IR rays to be received by an IR receiver module. An IR output terminal of the receiver varies depending upon its receiving of IR rays. Since this variation cannot be analyzed as such, therefore this output can be fed to a comparator circuit. When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM339). Thus the output of the comparator goes low, but the LED does not glow. When the IR receiver module receives signal to the potential at the inverting input goes low.

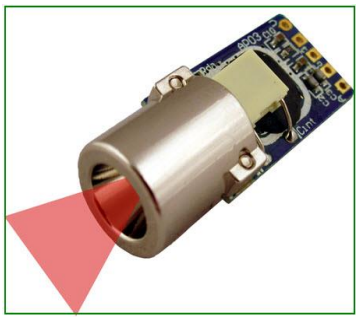


Figure:2.2.4.1 IR Sensor

2.2.5 Buck Converter

A Buck converter is a switch mode DC to DC electronic converter in which the output voltage will be transformed to level less than the input voltage. It is also called as step down converter. The name step down converter comes from the fact that analogous to step down transformer the input voltage is stepped down to a level less than the input voltage. By law of conservation of energy, the input power has to be equal to output power (assuming no losses in the circuit).Input power (P_{in}) = output power (P_{out}).Since $V_{in} > V_{out}$ in a buck converter, it follows then that the output current will be greater than the input current. Therefore, in buck converter

$$V_{in} > V_{out} \text{ and } I_{in} < I_{out}$$

Principle of operation of Buck converter: The main working principle of buck converter is that the inductor in the input circuit resists sudden variations in input current. When switch is ON the inductor stores energy in the form of magnetic energy and discharges it when switch is closed. The capacitor in the output circuit is assumed large enough that the time constant of RC circuit in the output stage is high. The large time constant compared to switching period ensures a constant output voltage $V_o(t) = V_o(\text{constant})$

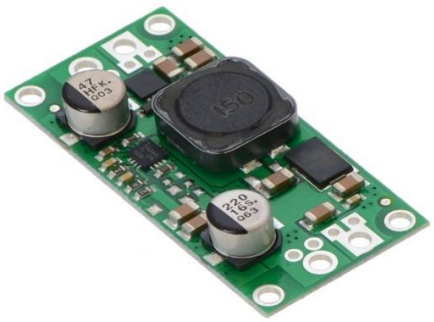


Figure: 2.2.5.1 Buck Converter

2.2.6 Bread board

When you start on your electronics journey, you will eventually need to wire up some parts to follow along with some project. And, chances are, you will be prodded towards using a breadboard. These ubiquitous pale slabs of plastic are everywhere when it comes to electronics hacking.Their popularity is not surprising - they are like a cutting board is to a cook, or a sewing machine to a tailor: indispensable, multi-purpose, durable, and inexpensive. Just about every beginner kit contains one, and almost all projects make use of one

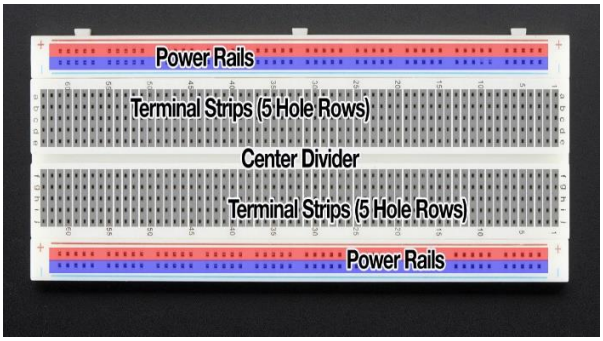


Figure: 2.2.6.1: Breadboard

2.2.7 Adapter

Introduction to Adapter Objects: Any driver that uses direct I/O and DMA must create an adapter object. The adapter object represents either a DMA controller channel or port, or a bus-master device. Two kinds of lowest-level drivers must use adapter objects:

- Drivers for devices that use the system DMA controller. Such devices are called subordinate devices and are said to "use system (or subordinate) DMA."
- Drivers for devices that are bus-master adapters. Such devices arbitrate with the system for use of the I/O bus, and thus use bus-master DMA.Drivers provide storage, usually in a device extension, for a pointer to the adapter object. To carry out DMA transfers, drivers of devices that use either of these DMA methods usually have an Adapter Control routine and call system-supplied support routines that manipulate adapter objects.



Figure: 2.2.7.1 Adapter

2.2.8 DC Motor

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances.The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of power electronics has made rep



Figure:2.2.8.1 DC Motor

2.2.9 LED Light

The LED stands for Light-emitting diode – as the name suggests – it is a pn-junction diode mainly used as the source of light. It is not a new device, it has long been serving in the field of telecommunication, electrical engineering and in hard to reach places where common light fails to deliver. The LED has a leg over common orthodox incandescent light in terms of efficiency, low consumption power, compact size, longer range and an ability to retain the quality for a longer period of time. Medical devices to embedded and control system for the indication if the relevant projects are working properly. In this post, I'll try to cover each and everything related to LED, so you don't need to wrestle your mind browsing the whole internet and find all the information in one place. Let's dive in and get down to the detail of this little electrical component.

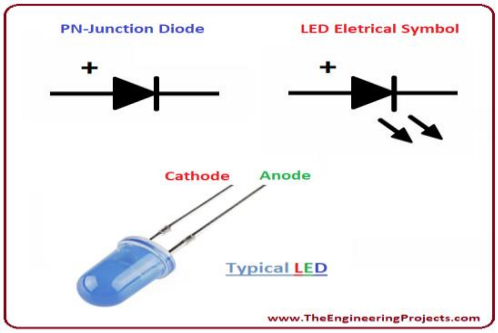


Figure: 2.2.9.1 LED light

2.2.10 Battery

Batteries create electricity using liquids or metals that are good conductors of electricity. You've probably seen that batteries have a + and a – sign on them. The positive and negative ends of the battery represent the two electrodes of the battery. Electrodes are two different kinds of metals that can move electrons between them, which creates electricity. The electrode that electrons move away from (with a negative sign) is called an anode. The electrode that electrons move towards (with a positive sign) is called a cathode. The salt water battery works just like any other battery: it will only produce an electrical current when you attach it to something that runs on electricity. Batteries can stay in a drawer or on a shelf for years without producing electricity, but as soon as you connect it to a toy or an appliance that uses electricity, it starts using the chemical energy inside it to make an electric current.When a battery is connected to a motor or other device that is powered by electricity, ions start moving from the cathode to the anode through the material that separates them, called the electrolyte. Electrons don't travel through the electrolyte, so they have to move through the wires connected to the motor, which creates an electric current.So in your salt water fuel cell, connecting the motor to the fuel cell causes electrons to travel up the wire from the anode, through the motor, and back to the cathode.



Figure: 2.2.10.1 Battery

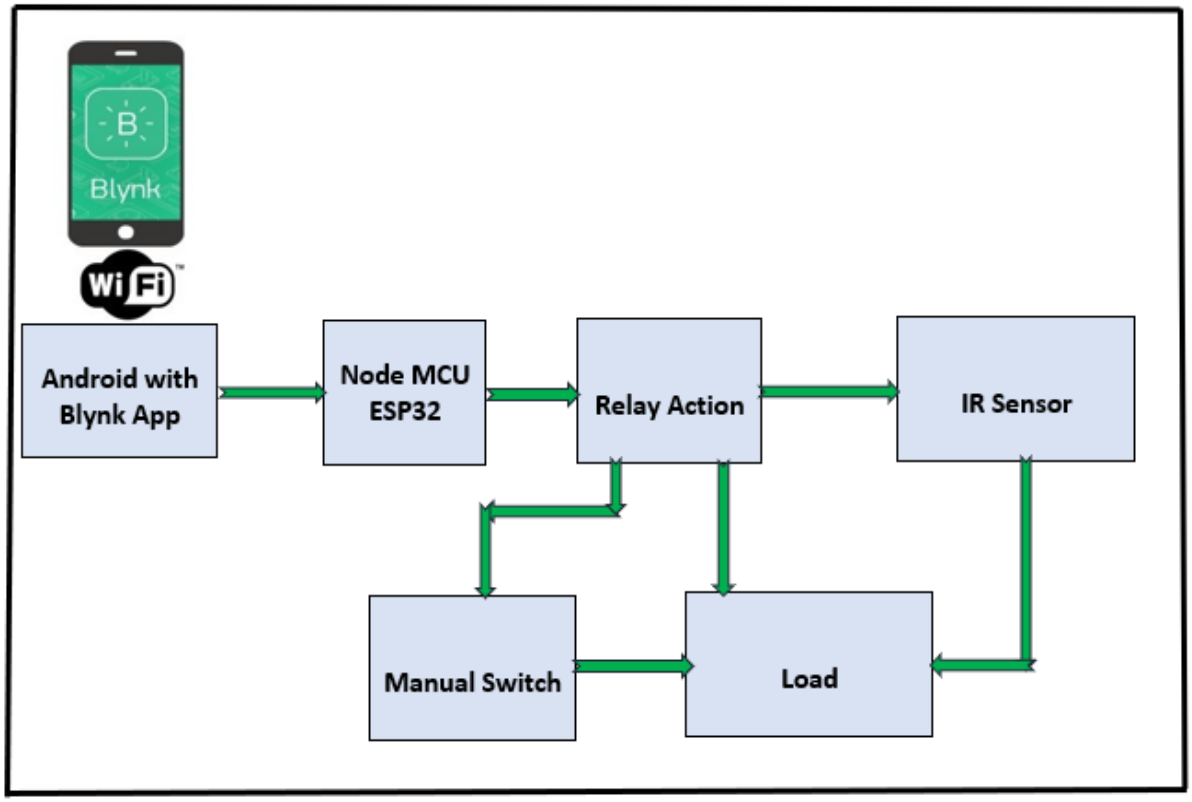
CHAPTER 3

DESING & FEBRICATION

3.1 Introduction

In this chapter fully cover with discuss design and fabrication of this project. Here we will discuss about developed block diagram and briefly describe about the circuit description and also learn about working principle.

3.2 Block Diagram



3.2 : Block Diagram of project

To achieve the home control using smart phone, initially the mobile unit should be connected to any network and the WiFi module ESP32 which is on Node MCU should be connected to local network. By touching the specified location of widgets in the blynk app, this blynk app sends ON/OFF commands to ESP32 on Node MCU via Blynk server. Then the relays connected to Node MCU performs switching operation corresponding to the input. So that the loads connected to relays will be switched ON/OFF. Similarly, we can also control our home appliances using manual switches. This can be obtained by connecting the relays outputs to manual switches and manual switches to the loads. Whenever the relay corresponding to manual ON or OFF is switched ON then the loads corresponding to that switches will perform switching operation.

A. Hardware Components Used

1)Node MCU

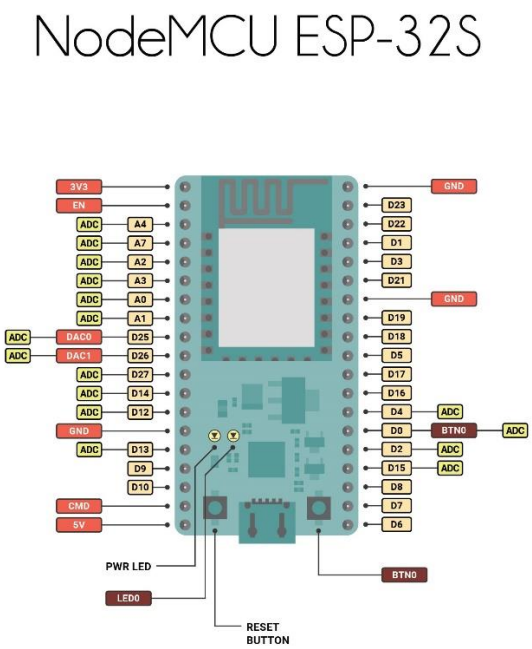


Figure: 3.2.1 Node MCU Circuit

2) Relay:A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. Here we are using SPDT (Single Pole Double Throw) relay. The Single Pole Double Throw SPDT relay is quite useful in certain applications because of its internal configuration. It has one common terminal and 2 contacts in 2 different configurations: one can be Normally Closed and the other one is opened or it can be Normally Open and the other one closed. So basically you can see the SPDT relay as a way of switching between 2 circuits: when there is no voltage applied to the coil one circuit "receives" current, the other one doesn't and when the coil gets energized the opposite is happening. We are using it in Normally Open mode.



Figure: 3.2.2.1 Top View of 8-Channel Relay

3) Smart Phone with Blynk App

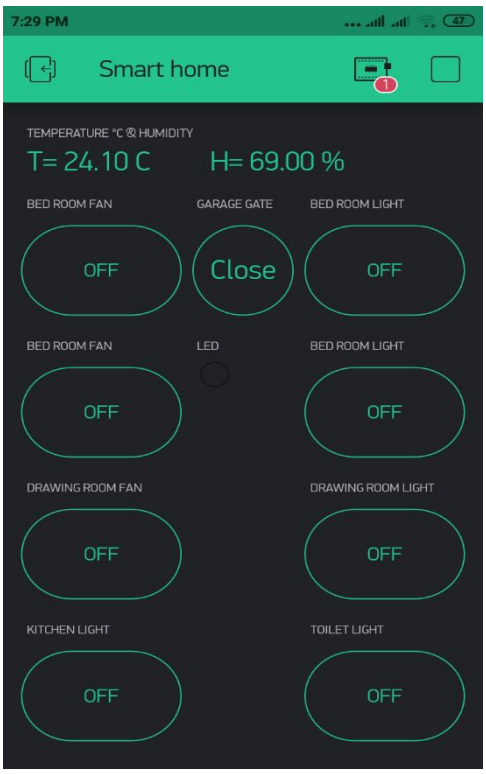


Figure: 3.2.2.2 Blynk App Screen Short

Here we are using smart phone with blynk app.

i) Blynk App

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 we can build a graphic interface for our project by simply dragging and dropping widgets. Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of our choice. Whether our Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online and ready for the Internet Of Your Things.

3.3 Circuit Diagram

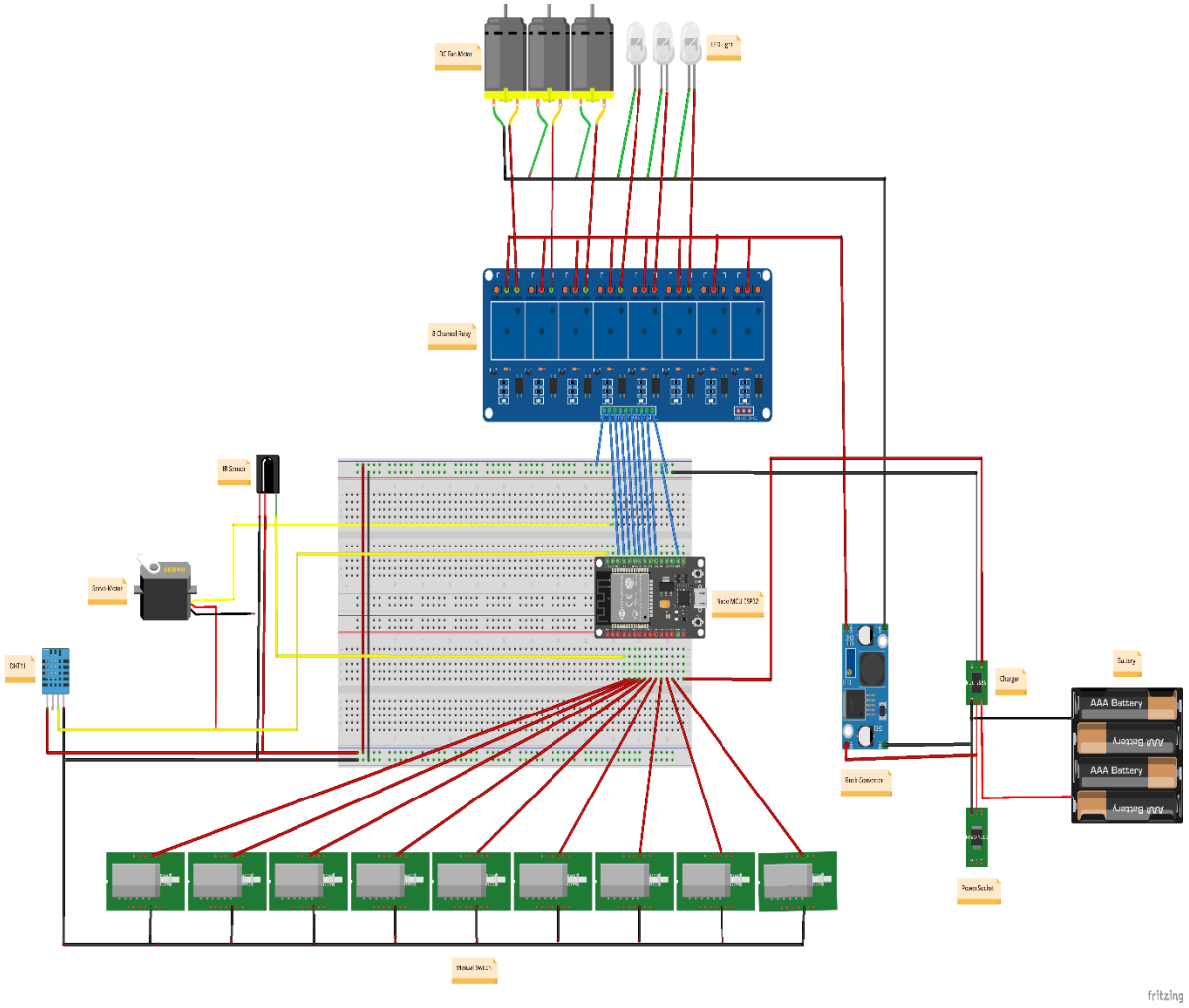


Figure 3.3 : Circuit Diagram of this project

3.4 Overview Of the project.

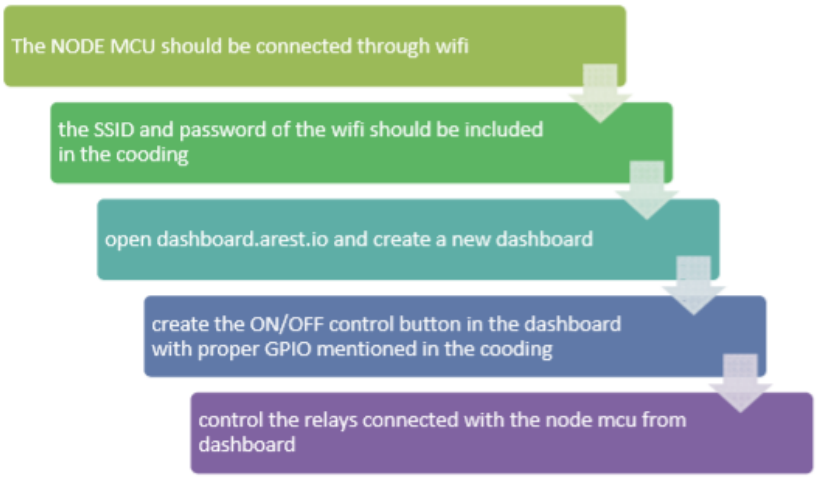


Figure: 3.4 Overview of Project

3.5 Image Of the project

After connecting and programming all the components with the, we conducted the experiment. We have run all the components according to the proposed system all modules and microcontroller are kept together with a lot of wires. This part is the main center of the home automation system. The sensors are placed inside the room. All commands are given from Android App and remote control.



Figure: 3.5 ‘Top view of the home automation system’

3.6 Used Materials

1.	Node MCU ESP32	1 pcs
2.	Battery	4 pcs
3.	Buck Converter	1 pcs
4.	Switch	8 pcs
5.	LED light	5 pcs
6.	Power sources (5v)	1 pcs
7.	DHT11	1 pcs
8.	IR Sensor	1 pcs
9.	Servo Motor	1 pcs
10.	Fan motor(3V)	3 pcs

3.7 Flow Chart

After assembling the system, what remains is to observe its operation and efficiency of the system. The total system is divided in several sub systems, like

- **Node MCU**
- **Bread Board**
- **Relay Module**
- **Connecting Wire**

The operation of the whole circuit is depending on every sections performance. 3.7 Flow Chart

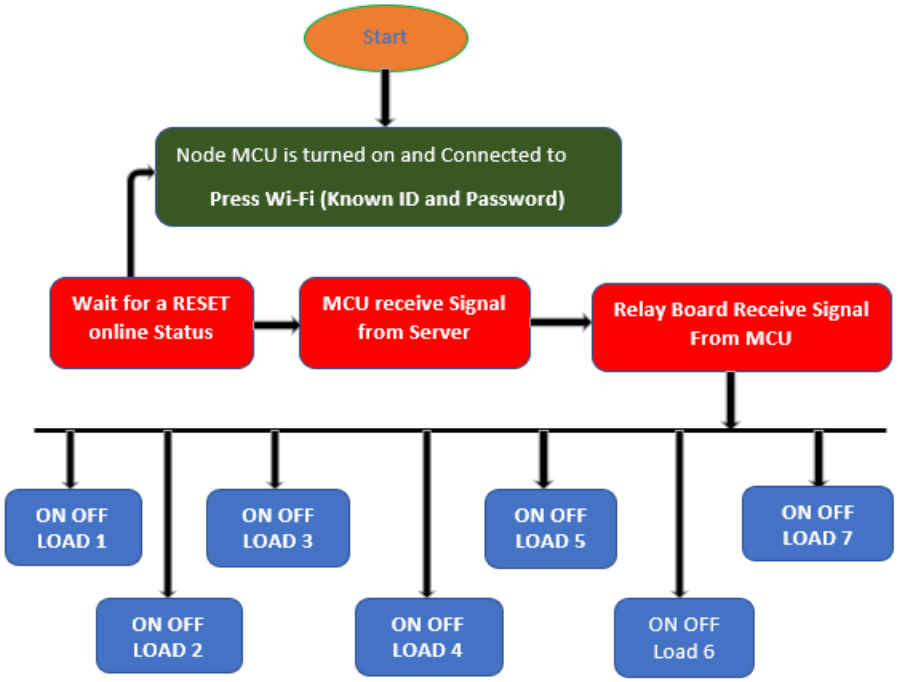


Figure: 3.7 Flow Chart Of the project

3.8 Conclusion

In this chapter we have discussed the block diagram,circuit discription,Flow Chart,Blynk App Screen Short,show the real image of the project and others. In the later part of this paper we discussed the result & discussion of the project to make the concept clear to anyone.

Chapter 4

RESULT & DISCUSSION

4.1 Introduction

This chapter contains the results obtained and discussion about the project. We have also covered discussions about advantages, disadvantages Of this project home automation system.

4.2 Result

After connecting and programming all the components with the, we conducted the experiment. We have run all the components according to the proposed system. We have designed a home automation system with Node MCU Microcontroller.All modules and microcontroller are kept together with a lot of wires. This part is the main centre of the home automation system.



Figure: 4.2 View Of the project

4.3 Advantages

- Intelligent home automation system
- Home appliances can be controlled in just one click
- Android based application to control appliances
- Easy user interface of the android application
- Multiple modes to control and run appliances when necessary
- Individual appliances can be controlled from the app
- Home appliances can be controlled from any remote places
- This system turns a home into a smart home

4.4 Disadvantages

- Internet Reliance
- Setup and Configuration
- More Technical Security Threats

4.5 Limitations

Besides all the advantages there are also some limitations in our projects which we look forward to overcome in our future works. The limitations of this system are discussed below:

- Wireless connection was designed based on dynamic IP addresses which changes every time after switching to a different network
- Home controlling devices must be connected to internet with a good bandwidth otherwise controlling could be disrupted
- If no internet is available in the client side or in the home devices controller side than this system cannot be controlled but it will keep executing the last command that was received by the controller
- User have to give IP addresses of the controller every time it connects with a new network

Chapter 5

CONCLUSION

5.1 Conclusion

The paper has been experimentally proven to work successfully. We can control the loads using the Blynk app or manually using the normal switches also Remote control with IR sensor. The designed system can be controlled from anywhere throughout the world. This will help the user by saving the power by switching off the home appliances in case of forgot to switch off. This will also help for physically disabled people who might otherwise require care givers.

From the project carried out, we find the system effectively low cost and user friendly. The whole house remains under the user's control all the time.

5.2 Limitations

There are some limitations observed throughout our project. Since our project is internet based, our home appliances are totally controlled through internet access. As we control the whole system through internet, we need to confirm high speed of internet. Otherwise, the system delay will occur as Ethernet shield will not be working as a network provider to the circuit.

The project relies on power supply. So if the power supply fails, the internet connection will be halted. Then database access will be stopped. For this, without security system the whole system will not be worked. In our project, security system is powered by another power source for security safety.

Though it needs less circuitry but its cost is not in minimal range. To get facility, users need to expense for this. The cost of installing a home automation system can be to a certain extent expensive. But it depends on the apparatus. The more sophisticated the system is the more expensive it will be.

5.3 Future Scope

As we have mentioned earlier this thesis is not a complete project. This is just a basic structure of another complete system. We have done all the basic necessities of a typical house.

Reference

[1] Circuit Diagram Design making with fritzing software

[2] Block Diagram nd Flow Chart Making with MS Word nd paint 3D

[3] https://docs.zerynth.com/latest/official/board.zerynth.nodemcu_esp32/docs/index.html

[4] https://circuitdigest.com/article/servo-motor-basics nd working principle

[5] https://www.robot-r-us.com/vmchk/sensor-temp/humid/dht11-temperature-and-humidity-sensor.html

[6] https://www.elprocus.com/infrared-ir-sensor-circuit-and-working/

[7] http://wiki.sunfounder.cc/index.php?title=8_Channel_5V_Relay_Module

[8] https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all

[9] https://en.wikipedia.org/wiki/DC_motor

[10] Google Search PDF Support nd also support from previous note book formate.

Appendix

Setup Code

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <SimpleTimer.h>
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRrecv.h>
#include <IRutils.h>
#include <ESP32Servo.h>
#include "DHT.h"
#define DHTPIN 23

#define DHTTYPE DHT11

const uint16_t kRecvPin = 36;

IRrecv irrecv(kRecvPin);

decode_results results;

DHT dht(DHTPIN, DHTTYPE);

char auth[] = "FG-0_i4LPv0N4lQ6r6nlFM_xTftVrlb3";

//WiFi credentials.
char ssid[] = "mobilewifi";
char pass[] = "12345678";

const int btnPin0 = 39;          // pin for physical push button switch.
const int btnPin1 = 34;          // pin for physical push button switch.
const int btnPin2 = 35;          // pin for physical push button switch.
const int btnPin3 = 32;          // pin for physical push button switch.
const int btnPin4 = 33;          // pin for physical push button switch.
const int btnPin5 = 25;          // pin for physical push button switch.
const int btnPin6 = 26;          // pin for physical push button switch.
const int btnPin7 = 27;          // pin for physical push button switch.
const int btnPin8 = 14;          // pin for physical push button switch.
const int RelayPin1 = 15;       // pin for relay output.
const int RelayPin2 = 2;       // pin for relay output.
const int RelayPin3 = 4;       // pin for relay output.
const int RelayPin4 = 5;       // pin for relay output.
const int RelayPin5 = 18;       // pin for relay output.
const int RelayPin6 = 19;       // pin for relay output.
const int RelayPin7 = 21;       // pin for relay output.
const int RelayPin8 = 3;       // pin for relay output.

ESP32Servo myservo;
int pos = 0;
int flag = 0;

BlynkTimer timer;

void checkPhysicalButton();
```

```
int btnPinState0 = LOW;         // ON
int btnPinState1 = LOW;         // ON
int btnPinState2 = LOW;         // ON
int btnPinState3 = LOW;         // ON
int btnPinState4 = LOW;         // ON
int btnPinState5 = LOW;         // ON
int btnPinState6 = LOW;         // ON
int btnPinState7 = LOW;         // ON
int btnPinState8 = LOW;         // ON
int RelayPinState0 = HIGH;      // OFF
int RelayPinState1 = HIGH;      // OFF
int RelayPinState2 = HIGH;      // OFF
int RelayPinState3 = HIGH;      // OFF
int RelayPinState4 = HIGH;      // OFF
int RelayPinState5 = HIGH;      // OFF
int RelayPinState6 = HIGH;      // OFF
int RelayPinState7 = HIGH;      // OFF
int RelayPinState8 = HIGH;      // OFF
void setup()
{
  Serial.begin(115200);
  irrecv.enableIRIn();
  Blynk.begin(auth, ssid, pass);
  pinMode(RelayPin1, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin2, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin3, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin4, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin5, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin6, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin7, OUTPUT);          // initialize your pin as output.
  pinMode(RelayPin8, OUTPUT);          // initialize your pin as output.
  pinMode(btnPin0, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin1, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin2, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin3, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin4, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin5, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin6, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin7, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  pinMode(btnPin8, INPUT_PULLUP);      // initialize your pin as input with enable internal pull-up resistor
"input_pullup"
  digitalWrite(RelayPin1, RelayPinState1);
  digitalWrite(RelayPin2, RelayPinState2);
  digitalWrite(RelayPin3, RelayPinState3);
  digitalWrite(RelayPin4, RelayPinState4);
  digitalWrite(RelayPin5, RelayPinState5);
  digitalWrite(RelayPin6, RelayPinState6);
  digitalWrite(RelayPin7, RelayPinState7);
  digitalWrite(RelayPin8, RelayPinState8);

dht.begin();
myservo.attach(22);
```

```
// Setup a function to be called every 100 ms
timer.setInterval(100L, checkPhysicalButton);
}
// Every time we connect to the cloud...
BLYNK_CONNECTED()
{
// Request the latest state from the server
Blynk.syncVirtual(V0);
Blynk.syncVirtual(V1);
Blynk.syncVirtual(V2);
Blynk.syncVirtual(V3);
Blynk.syncVirtual(V4);
Blynk.syncVirtual(V5);
Blynk.syncVirtual(V6);
Blynk.syncVirtual(V7);
Blynk.syncVirtual(V8);
}

// When App button is pushed - switch the state
// Map this Virtual Pin to your Mobile Blynk apps widget.
BLYNK_WRITE(V0)
{
RelayPinState0 = param.asInt();
if(RelayPinState0 == LOW && flag == 0){
Serial.println("11");
for(int posDegrees = 0; posDegrees <= 120; posDegrees++) {
myservo.write(posDegrees);
delay(20);
}
flag = 1;
}
else if(RelayPinState0 == HIGH && flag == 1){
Serial.println("22");
for(int posDegrees = 120; posDegrees >= 0; posDegrees--) {
myservo.write(posDegrees);
delay(20);
}
flag = 0;
}

}
BLYNK_WRITE(V1)
{
RelayPinState1 = param.asInt();
digitalWrite(RelayPin1, RelayPinState1);
}
BLYNK_WRITE(V2)
{
RelayPinState2 = param.asInt();
digitalWrite(RelayPin2, RelayPinState2);
}
BLYNK_WRITE(V3)
{
RelayPinState3 = param.asInt();
digitalWrite(RelayPin3, RelayPinState3);
```



```
}
BLYNK_WRITE(V4)
```

```
{
RelayPinState4 = param.asInt();
digitalWrite(RelayPin4, RelayPinState4);
}
BLYNK_WRITE(V5)
{
RelayPinState5 = param.asInt();
digitalWrite(RelayPin5, RelayPinState5);
}
BLYNK_WRITE(V6)
{
RelayPinState6 = param.asInt();
digitalWrite(RelayPin6, RelayPinState6);
}
BLYNK_WRITE(V7)
{
RelayPinState7 = param.asInt();
digitalWrite(RelayPin7, RelayPinState7);
}
BLYNK_WRITE(V8)
{
RelayPinState8 = param.asInt();
digitalWrite(RelayPin8, RelayPinState8);
}

void checkPhysicalButton()
{
if (digitalRead(btnPin0) == LOW)
{
if (btnPinState0 != LOW)
{
RelayPinState0 = !RelayPinState0;
Blynk.virtualWrite(V0, RelayPinState0);
if(RelayPinState0 == LOW){
Serial.println("33");
for(int posDegrees = 0; posDegrees <= 120; posDegrees++) {
myservo.write(posDegrees);
delay(20);
}
}
else{
Serial.println("44");
for(int posDegrees = 120; posDegrees >= 0; posDegrees--) {
myservo.write(posDegrees);
delay(20);
}
}
}

btnPinState0 = LOW;
}
else
{
btnPinState0 = HIGH;
}
if (digitalRead(btnPin1) == LOW)
{
```



```
if (btnPinState1 != LOW)
{
```

```
RelayPinState1 = !RelayPinState1;
digitalWrite(RelayPin1, RelayPinState1);
Blynk.virtualWrite(V1, RelayPinState1);
}

btnPinState1 = LOW;
}
else
{
btnPinState1 = HIGH;
}
if (digitalRead(btnPin2) == LOW)
{
if (btnPinState2 != LOW)
{
RelayPinState2 = !RelayPinState2;
digitalWrite(RelayPin2, RelayPinState2);
Blynk.virtualWrite(V2, RelayPinState2);
}

btnPinState2 = LOW;
}
else
{
btnPinState2 = HIGH;
}
if (digitalRead(btnPin3) == LOW)
{
if (btnPinState3 != LOW)
{
RelayPinState3 = !RelayPinState3;
digitalWrite(RelayPin3, RelayPinState3);
Blynk.virtualWrite(V3, RelayPinState3);
}

btnPinState3 = LOW;
}
else
{
btnPinState3 = HIGH;
}
if (digitalRead(btnPin4) == LOW)
{
if (btnPinState4 != LOW)
{
RelayPinState4 = !RelayPinState4;
digitalWrite(RelayPin4, RelayPinState4);
Blynk.virtualWrite(V4, RelayPinState4);
}

btnPinState4 = LOW;
}
else
{
btnPinState4 = HIGH;
}
if (digitalRead(btnPin5) == LOW)
```



```
{
if (btnPinState5 != LOW)
```

```
{
RelayPinState5 = !RelayPinState5;
digitalWrite(RelayPin5, RelayPinState5);
Blynk.virtualWrite(V5, RelayPinState5);
}

btnPinState5 = LOW;
}
else
{
btnPinState5 = HIGH;
}
if (digitalRead(btnPin6) == LOW)
{
if (btnPinState6 != LOW)
{
RelayPinState6 = !RelayPinState6;
digitalWrite(RelayPin6, RelayPinState6);
Blynk.virtualWrite(V6, RelayPinState6);
}

btnPinState6 = LOW;
}
else
{
btnPinState6 = HIGH;
}
if (digitalRead(btnPin7) == LOW)
{
if (btnPinState7 != LOW)
{
RelayPinState7 = !RelayPinState7;
digitalWrite(RelayPin7, RelayPinState7);
Blynk.virtualWrite(V7, RelayPinState7);
}

btnPinState7 = LOW;
}
else
{
btnPinState7 = HIGH;
}
if (digitalRead(btnPin8) == LOW)
{
if (btnPinState8 != LOW)
{
RelayPinState8 = !RelayPinState8;
digitalWrite(RelayPin8, RelayPinState8);
Blynk.virtualWrite(V8, RelayPinState8);
}

btnPinState8 = LOW;
}
else
{
btnPinState8 = HIGH;
}
```



```
float h = dht.readHumidity();
float t = dht.readTemperature();
```

```
String stringOne, stringTwo, stringThree, stringFour, stringFive;
stringOne = String("T= ");
stringTwo = String(" C         H= ");
stringThree = String(" %");
stringFour = String();
stringFour = stringOne + t + stringTwo + h + stringThree;
Blynk.virtualWrite(V9, stringFour);
}

void loop()
{
Blynk.run();                          // Run Blynk
timer.run();                          // Run SimpleTimer
if(irecv.decode(&results)){
// print() & println() can't handle printing long longs. (uint64_t)
serialPrintUint64(results.value);
Serial.println("");
irecv.resume();  // Receive the next value
if(results.value == 16724175){
Serial.println("1");
RelayPinState1 = !RelayPinState1;
digitalWrite(RelayPin1, RelayPinState1);
Blynk.virtualWrite(V1, RelayPinState1);
delay(100);
}
else if(results.value == 16718055){
Serial.println("2");
RelayPinState2 = !RelayPinState2;
digitalWrite(RelayPin2, RelayPinState2);
Blynk.virtualWrite(V2, RelayPinState2);
delay(100);
}
else if(results.value == 16743045){
Serial.println("3");
RelayPinState3 = !RelayPinState3;
digitalWrite(RelayPin3, RelayPinState3);
Blynk.virtualWrite(V3, RelayPinState3);
delay(100);
}
else if(results.value == 16716015){
Serial.println("4");
RelayPinState4 = !RelayPinState4;
digitalWrite(RelayPin4, RelayPinState4);
Blynk.virtualWrite(V4, RelayPinState4);
delay(100);
}
else if(results.value == 16726215){
Serial.println("5");
RelayPinState5 = !RelayPinState5;
digitalWrite(RelayPin5, RelayPinState5);
Blynk.virtualWrite(V5, RelayPinState5);
delay(100);
}
else if(results.value == 16734885){
Serial.println("6");
RelayPinState6 = !RelayPinState6;
```



```
digitalWrite(RelayPin6, RelayPinState6);
Blynk.virtualWrite(V6, RelayPinState6);
```

```
delay(100);
}
else if(results.value == 16728765){
Serial.println("7");
RelayPinState7 = !RelayPinState7;
digitalWrite(RelayPin7, RelayPinState7);
Blynk.virtualWrite(V7, RelayPinState7);
delay(100);
}
else if(results.value == 16730805){
Serial.println("8");
RelayPinState8 = !RelayPinState8;
digitalWrite(RelayPin8, RelayPinState8);
Blynk.virtualWrite(V8, RelayPinState8);
delay(100);
}
else if(results.value == 16753245){
Serial.println("off");
RelayPinState0 = !RelayPinState0;
Blynk.virtualWrite(V0, RelayPinState0);
if(RelayPinState0 == LOW){
Serial.println("33");
for(int posDegrees = 0; posDegrees <= 120; posDegrees++) {
myservo.write(posDegrees);
delay(20);
}
}
else{
Serial.println("44");
for(int posDegrees = 120; posDegrees >= 0; posDegrees--) {
myservo.write(posDegrees);
delay(20);
}
}
delay(100);
}
else{
delay(500);
}
}

}
```