“Smart Home Automation”

NAAC ACCREDITED

A

PROJECT REPORT

Submitted to the

Dr. K. N. Modi University

For the Award of Degree of

BACHELOR OF COMPUTER APPLICATION

IN

Cyber Security

Submitted By:

Dhruv Sharma

Roll no 2020BCS003

ERP 02009BCS003

Bhavyansh Pareek

Roll no 2020BCS007

ERP 02009BCS007

Submitted To:

Mr. Shashi Maurya

Assistant Professor

Dr. K. N. Modi University

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Dr. K. N. MODI UNIVERSITY

NEWAI, RAJASTHAN-304021

“2023“

DECLARATION

We hereby declare that work, which is being presented in this Project Work, entitled **"Smart Home Automation"** in partial fulfillment for the award of Degree of "Bachelor of Computer Application" in Department of Computer Science & Engineering, and **submitted to the Department of Computer Science & Engineering in Dr. K. N. Modi University, Newai,** is a record of my own investigations carried under the Guidance of **"Mr. Shashi Maurya"** (Assistant Professor), Department of Computer Science Engineering, Dr. K. N. Modi University .

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Student name: Dhruv Sharma

Roll No:2020BCS003

Branch: CSE

Student name: Bhavyansh Pareek

Roll No:2020BCS007

Branch: CSE

ACKNOWLEDGEMENT

I express my sincere thanks to my project guide, **"Mr. Shashi Maurya"**, **Department of Computer Science & Engineering,** for guiding me right from the inception till the successful completion of the project. I sincerely acknowledge him for extending their valuable guidance, support for literature, critical reviews of project and above all the moral support he had provided to me with all stages of this project.

I would also like to thank the supporting all Faculty of Computer Science & Engineering Department, for their help and cooperation throughout our project.

I would like to mention thanks to my project coordinators & faculties & Lab Staff of my institution.

Last but not the least, I would like to special mention of the support, help and encouragement that I receive from my team members.

Student Name: Bhavyansh Pareek

Roll No:2020BCS007

Name: Dhruv Sharma

Roll No: 2020BCS003

INDEX

IOT based Smart Home Automation

Abstract

Smart home is basically a simple and normal home with basic utilities like fan, lights, TV and door operations in it but they are smartly controlled through a smart phone application and web interface.

Home automation system achieved great popularity in the last decades and it increases the comfort and quality of life, also this a famous and most used technology in the world. The main object of this project is to develop a home automation system with Android operating system using Wi-Fi technology. The 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 (WHAS) 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 Intel Galileo 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. This system is designed to be low cost and expandable allowing a variety of devices to be controlled. The process of controlling or operating various equipment, machinery and other applications using various control systems and with less or no human intervention is termed as automation. There are various types of automation based on the application they can be categorized as home automation, industrial automation, autonomous automation, building automation, etc.…In this project, we are discussing about wireless home automation. Home Automation is the process of controlling home appliances automatically using various control system techniques. The electrical and electronic appliances in the home such as fan, lights, outdoor lights etc., can be controlled using various control techniques. In recent years, wireless systems like Wi-Fi have become more and more common in-home networking. Also, in home and building automation systems, the use of wireless technologies gives several advantages that could not be achieved using a wired network only.

Introduction :

Goals :

Home Automation allows you to have greater control of your energy use, all while automating things like adjusting temperature, turning on and off lights, opening and closing window treatments and adjusting irrigation based on the weather.

Problem Definition :

There are basically 4 types of problems which are solved using Smart Home:

1. To monitor Power Consumption of your home
2. No Way to turn off the equipment you have accidently left on (Remote Access)
3. No Way to protect assets according to weather and dynamic changes in conditions
4. No Way to operate at your home from a place far from home (Rover inside home)

Scope :

- We can add biometric in the gates and the lockers.
- We can give instructions with the help of our voice.
- The fridge may sense outside temperature and work accordingly.
- Drones may be used to deliver packages at the right time.
- Smart mirrors can be used for playing music and TV may be used for social media.

Project Overview:

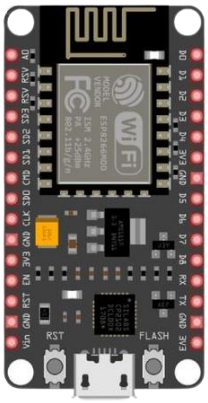
This project is basically a group of IC chips which are connected together to control power supply to various appliances in a home like fan, tv, lights etc. We have used two brains in this project one is the communication brain which is NodeMCU, responsible for internet connection and data transmission from phone or web interface directly just using SSID and Password of the wifi network provided as internet, another is the master brain which communicates to the NodeMCU and perform the tasks, it is also responsible for smart actions like automatically sensing rain and closing windows and door if they are open, it also operates basic rovers which are inside the projection area and control them according to instructions and suitable timings.

Project Pictures:

Requirements:

1. **Hardware Requirements:**

NodeMCU ESP32-S



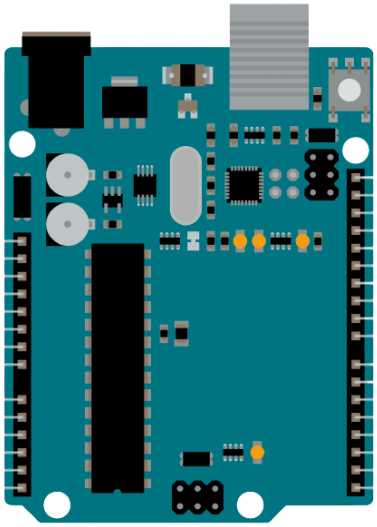
NodeMCU ESP32-S

NodeMCU is an open-source firmware for which open-source prototyping board designs are available. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). Strictly speaking, the term "NodeMCU" refers to firmware rather than the associated development kits.

It 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.

There are two available versions of NodeMCU as version 0.9 & 1.0 where the version 0.9 contains **ESP-12** and version 1.0 contains **ESP-12E** where E stands for "Enhanced".

Arduino UNO



Arduino UNO

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again.

Replaceable chip

The ATmega328P can easily be replaced, as it is not soldered to the board.

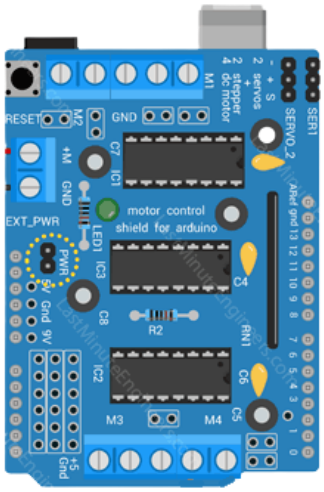
EEPROM

The ATmega328P also features 1kb of EEPROM, a memory which is not erased when powered off.

Battery Connector

The Arduino UNO features a barrel plug connector, that works great with a standard 9V battery.

Adafruit Motor Shield



Adafruit Motor Shield

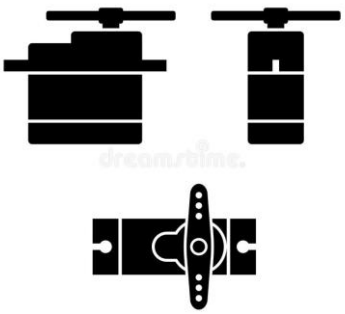
Arduino is a great starting point for electronics, and with a motor shield it can also be a nice tidy platform for robotics and mechatronics.

Here is a image for a full-featured motor shield that will be able to power many simple to medium-complexity Components or hardware parts.

Used for controlling all components on separate voltage.

- **2 connections for 5V 'hobby' servos** connected to the Arduino's high-resolution dedicated timer - no jitter!
- **Up to 4 bi-directional DC** motors with individual 8-bit speed selection (so, about 0.5% resolution)
- **Up to 2 stepper motors** (unipolar or bipolar) with single coil, double coil, interleaved or micro-stepping.
- 4 H-Bridges: L293D chipset provides 0.6A per bridge (1.2A peak) with thermal shutdown protection, 4.5V to 25V
- Pull down resistors keep motors disabled during power-up
- Big terminal block connectors to easily hook up wires (10-22AWG) and power
- Arduino reset button brought up top
- 2-pin terminal block to connect external power, for seperate logic/motor supplies.

Servo Motor



Servo Motors

A **servomotor** (or **servo motor**) is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor, although the term *servomotor* is often used to refer to a motor suitable for use in a closed-loop control system.

Used to open and close the door.

9V DC Motor

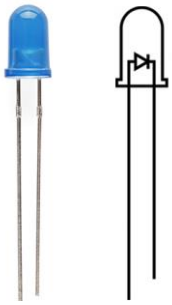
DC Motor 9V

This motor consumes 9V and provides a high velocity to any rotatable physical item for single direction at a time.

Not capable to function at a fix angle such as servo motor but are faster than servos.

Used as fan in this project.

LED White



LED (Light Emitting Diodes)

A **LED** is a semiconductor device that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The color of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device. Used as Lights and inside the TV.

Jumper Wires

Jumper Wires

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

Android Phone

Android Smart Phone

Android is a more suitable platform it has a large number of users and it is an in-hand device.

Android Phone is used as a Remote Device which runs the application and control the smart home chipset using internet.

Power Bank

Power Bank

Power Bank is 5V, 2A applicable but we can use up to 25 V with Adafruit motor shield, but we must reconfigure the power out puts of each port according to the connected device.

Power Bank is used the power source as we are implementing the project on small model

Bluetooth HC-05

Bluetooth HC-05

HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.

It is used to connect main chipset with the remote and autonomus machines.

2. Software Requirements :

Arduino IDE

Arduino IDE

The open-source Arduino Software (IDE) makes it easy to write code and up load it to the board. This software can be used with any Arduino Board.

Blynk Application Platform

Blynk Application Platform

Blynk is an application platform for IOS and Android smartphones that is used to communicate with NodeMCU, Arduino via Internet. On this application we have created a Graphical User Interface (GUI) by compiling and providing the appropriate address on the available widgets.

The application works on a editable model in case we need to add more buttons for more function we can add them on the template and they will be updated in all at once.

Blynk also provides a web interface to access the control server from anywhere just using ID and Password provided through any web browser.

Using Blynk we can connect our model via internet without any extra charges, but blynk is having some limitation, like there is no video streaming plugin available for chipsets, so we cannot stream video using internet without port forwarding but with port forwarding it is possible and for that we have to pay for our hosting charges.

MIT App Inventor

MIT App Inventor

It is an Open Source Where you can build your own android application using a GUI Interface and there is an accurate plugin fix available, using it you can fix any error (if any) within few minutes.

We used it to implement the basic Bluetooth model and that was working fine.

Background Study :

Programming Languages and Modules studied:

C++:

We used C++ because it provides structured approach -to break problems into parts-, in it Abstract Data Types could be created using classes, it is portable and code can be run on different machines, it also includes both low level and high level language so it is known as mid-level programming language and could be used to develop system applications such as kernel, drivers etc.

C++ provides very efficient management techniques the various memory management operators help save the memory and improve the program's efficiency. These operators allocate and deallocate memory at run time, some common memory managements operators available C++ are new, delete etc.

Object oriented concepts like data hiding, encapsulation and data abstraction can easily be implemented using keyword class, private, public and protected access specifiers.

In C++ we can divide our code in separate parts using functions, it provides rich library which mean lots of inbuilt functions that makes development fast
Structured programming language.

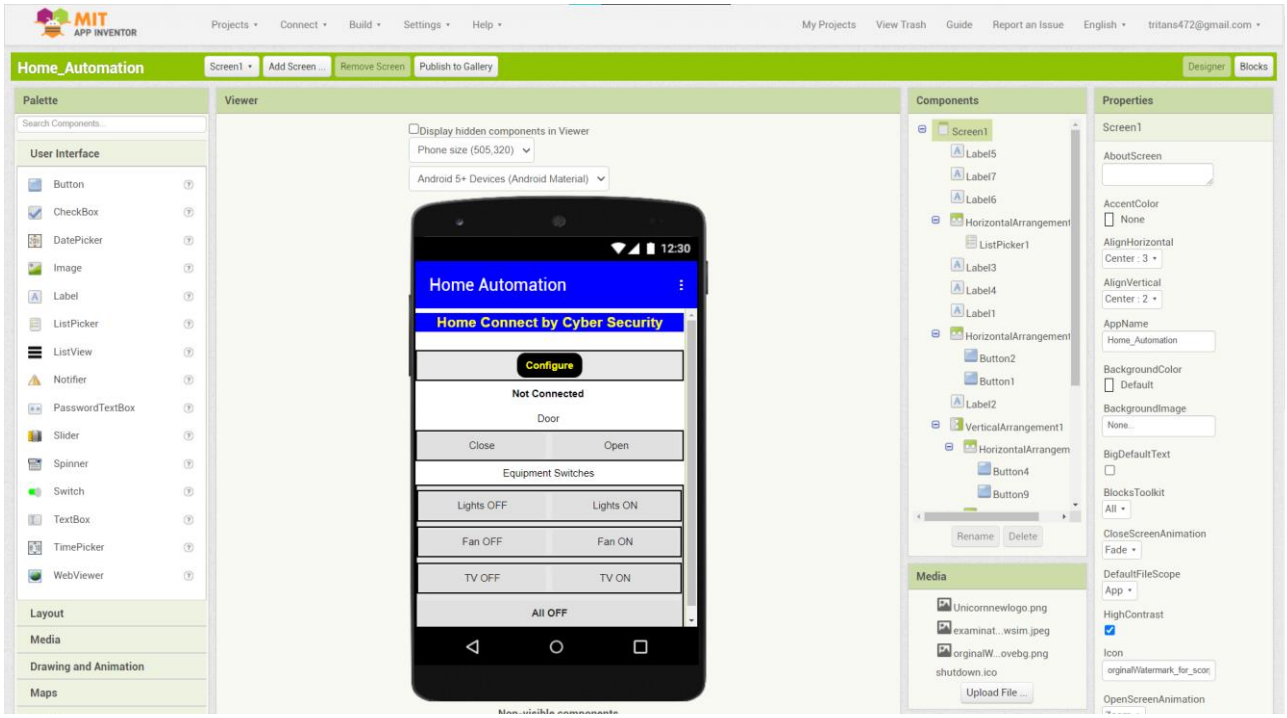
ESP8266:

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 and 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, the chip, and the software on it.

The board is self-dependent, which means it has all its needed components on itself, it is basically a brain with the Wi-Fi chipset and a programmable IC.

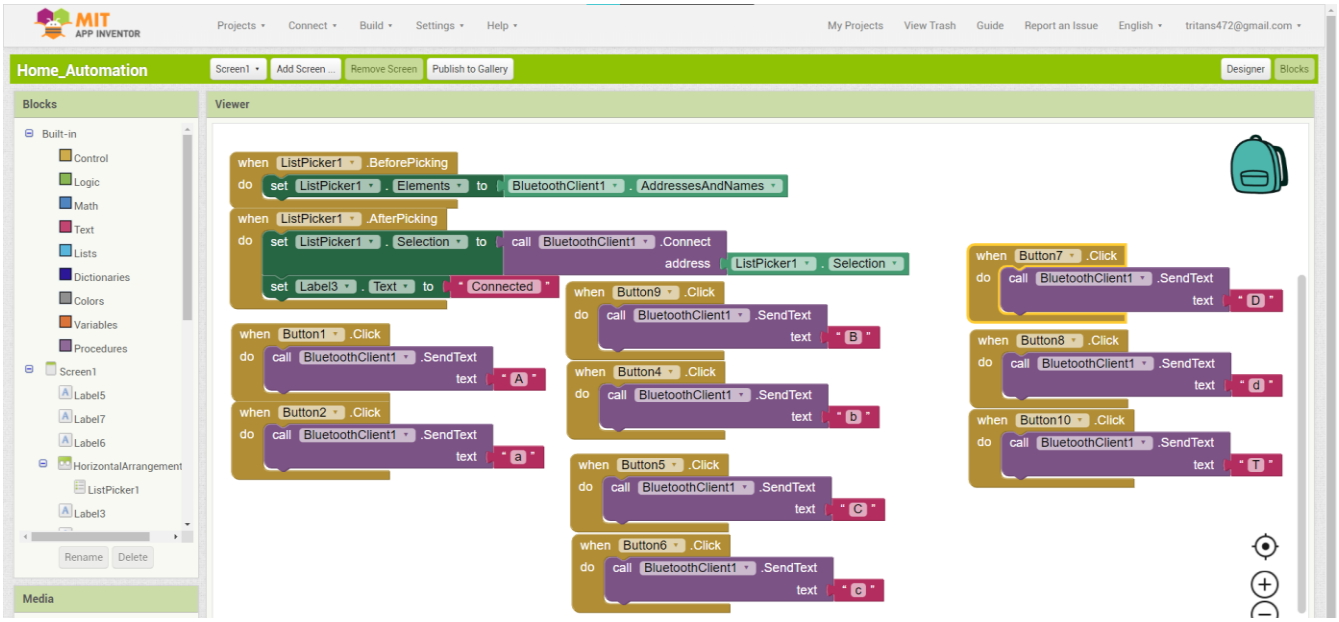
MIT App Inventor:

MIT app inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology.



It provides a GUI and MS Paint like environment to develop an android application on which we can choose colour, size complexion and number of items on the interface.

accordingly, we can also decide which type of application we want to develop and using the block view we can choose the actions performed by these GUI widgets

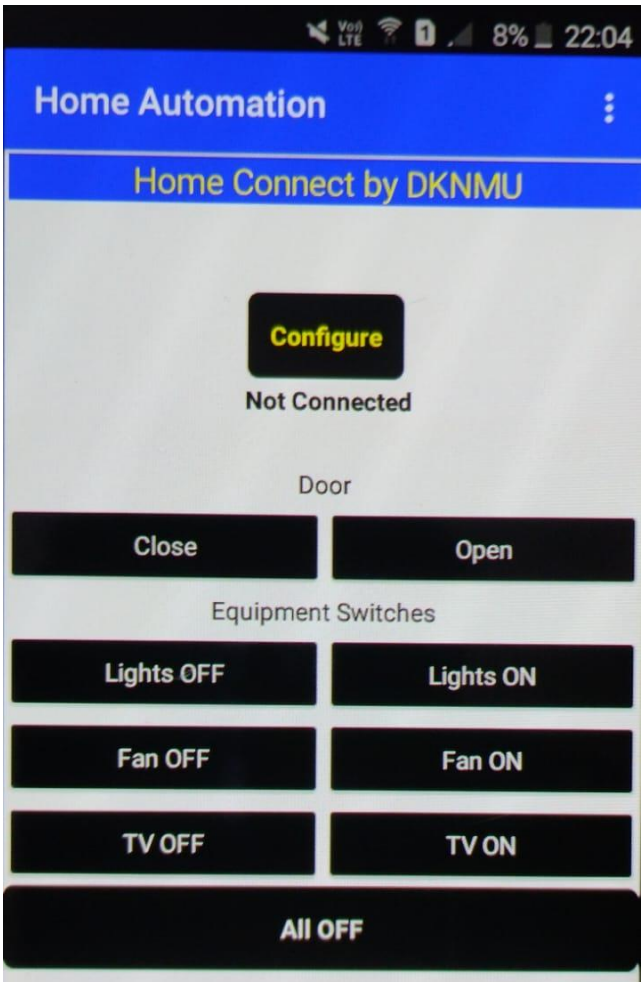


In Bluetooth version first we have to pair the bluetooth HC-05 module with the android phone directly and then we are able to connect to it in the configure section of the bluetooth application.

After connecting to it we are able to send instruction according to the buttons

We have use HC-05 Bluetooth module for this scenario which is capable to connect to the device with a highest range of 100 meters

later we changed the use of Bluetooth from controlling the device to control the autonomus bots which could be deployed for virtual attended access of a user.



Because Bluetooth is not having a larger range and it is limited to a fixed range but with the later Wi-Fi NodeMCU module we can control the devices using internet.

And now we can use Bluetooth to establish wireless connection between two or more device so they can share data wirelessly among them.

For previous version we used MIT app inventor to make a Bluetooth application to control this project using our phone, but it is having limitation to a fixed area, it could be 10-20 meters hardly 100 meters and to fix this we switched and applied this over internet using blynk.

Blynk:

Blynk is an application platform for IOS and Android smartphones that is used to communicate with NodeMCU, Arduino via Internet. On this application we have created a Graphical User Interface (GUI) by compiling and providing the appropriate address on the available widgets.

The application works on a editable model in case we need to add more buttons for more function we can add them on the template and they will be updated in all at once.

Using Blynk we can connect our model via internet without any extra charges, but blynk is having some limitation, like we are not having camera casting plugin and, in the replacement, we have to use port forwarding method. To take the live feed on the other location and the static link of broadcast is paid.

Technologies Used:

1. We have used wireless Internet connectivity instead of Bluetooth to connect to NodeMCU from android phone which ensures access from anywhere.

2. Direct Wi-Fi connectivity is used with baud rate of 9600 to ensure quick Wi-Fi connect and secure device recognition.

3. Blynk application platform is used to ensure secure connection and almost free to use WAN connectivity.

4. Two separate brains are used one for connectivity another for controls and smart actions.

5. Separate Arduino AFMotor module is used to ensure quicker and reissued speed of code looping.

6. Web Interface is also introducing to ensure connectivity even without personal device.

7. Servo Motor is used separately to ensure weight pulling and door opening on fixed angle.

8. Biometrics could be introduced to ensure safety in future models

9. Autosensing could be introduced to ensure smartness at unattended moments.

Connection Description of all Components:

There are 14 components which are 3 chipsets, 1 power supply, 6 LEDs, 1 DC motor, 1 Servo motor, Jumper Wires and a model made with wood sheets.

All are connected as described below:

1. NodeMCU is Connected to the power supply and the Arduino UNO using power cable and RX, TX and GND pins respectively and it is mainly connecting to the specific Wi-Fi of which SSID and Password are provided to it. After that it is connecting to the Blynk server through the blynk AUTH token provided by the blynk application and the Blynk module with which the code is being compiled and then it is just sending the online status logs and receives the data according to the user's command through the app and then communicates with Arduino UNO and send the appropriate letter CAPITAL CASE for ON and small case for OFF.

2. Arduino is connected to power source and NodeMCU using RX and TX pins, also it is sharing the Ground point with NodeMCU to share data without any electron collisions, After receiving letter from NodeMCU, Arduino processes and finds that which function is being executed according to the letter provided and then allows that function to be executed, after the function execution the function access the AFMotor library and orders the motor shield to provide the decided amount of power to the specific port in a specific direction decided according to the letter.

3. Motor shield is mounted on the Arduino as it is connected with all the pins of Arduino, after receiving instructions the Adafruit Motor Shield works as the instructions, we can also change the power output directions according to us using the coded algorithm.

4. And in the end, we can see the output as the functioning of our smart home.

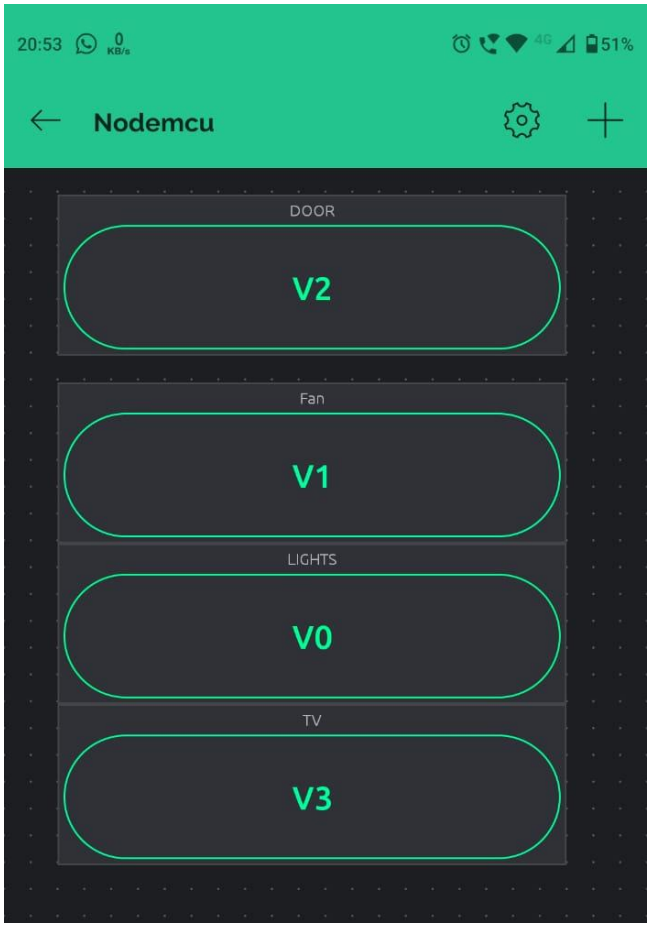
5. Servo Motor is fixed with the door to make it open and close according to user.

6. DC motor is fixed on the middle wood strip and working as a fan which is controlled by chipset.

7. 2 LED are sticked inside the Television to show it ON and OFF.

8. 3 LED are used as Lights and 1 as the door light which is turned ON when the door opens and OFF after 1 second of door closes.

Proposed system of Home Application

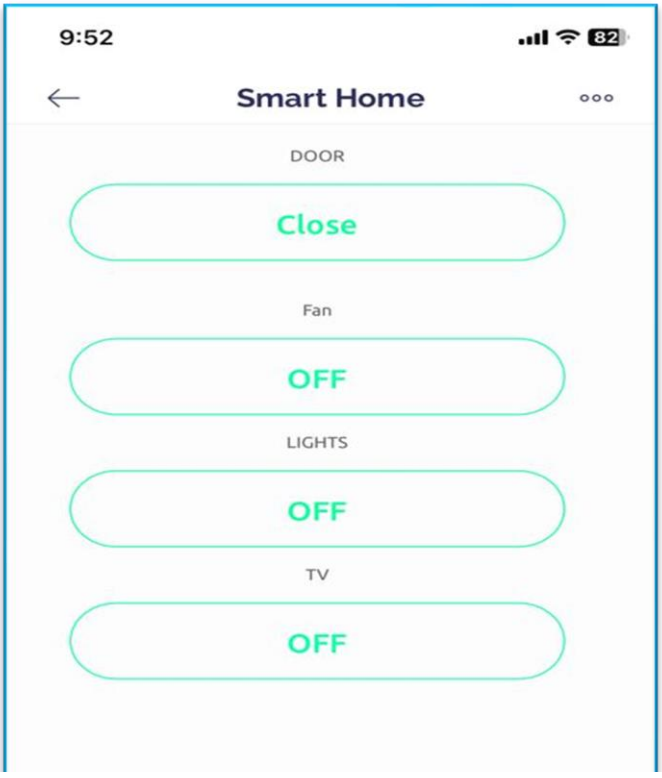


Blynk Application has a editable GUI interface using which we can easily edit the buttons and virtual pins which will perform actions in NodeMCU code.

Also, we can choose the colour and shape of the button and the whole interface and the type of button could be changed according to user requirements and design algorithm.

if we want to use a one touch button or switch button or a key button, they all are available

we just have to plug a virtual pin to the buttons and then in the NodeMCU code we have to set the action which should be performed when this virtual pin sends 1 or 0.



After all configuration it will provide us a clean and clear interface for all GUI usage

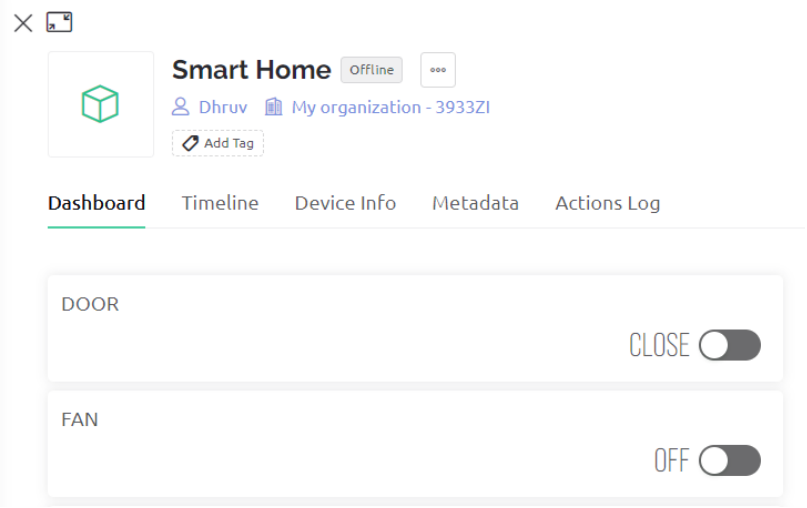
using blynk we just have to download blynk app and login our user id and password in it and our application will be live on it.

Web interface provides us access to our smart home from anywhere and from any device whether its ours or not and
it is also helpful in a condition if we accidently lost our smartphone.

Then we can access our smart home from our friend's or neighbour's device

we can also design a web interface using blynk, with live online and offline status update,

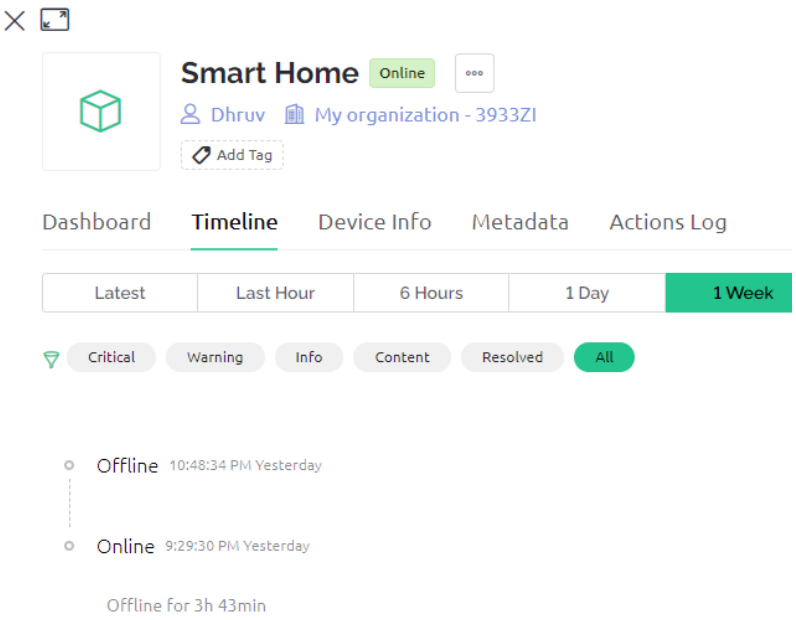
like we have used single tone buttons in android application



but on web interface we have used switch button which are more attractive and more suitable according to web accessibility.

We also have a timeline feature here which provides us all the activity details of smart home chip that when it was last online

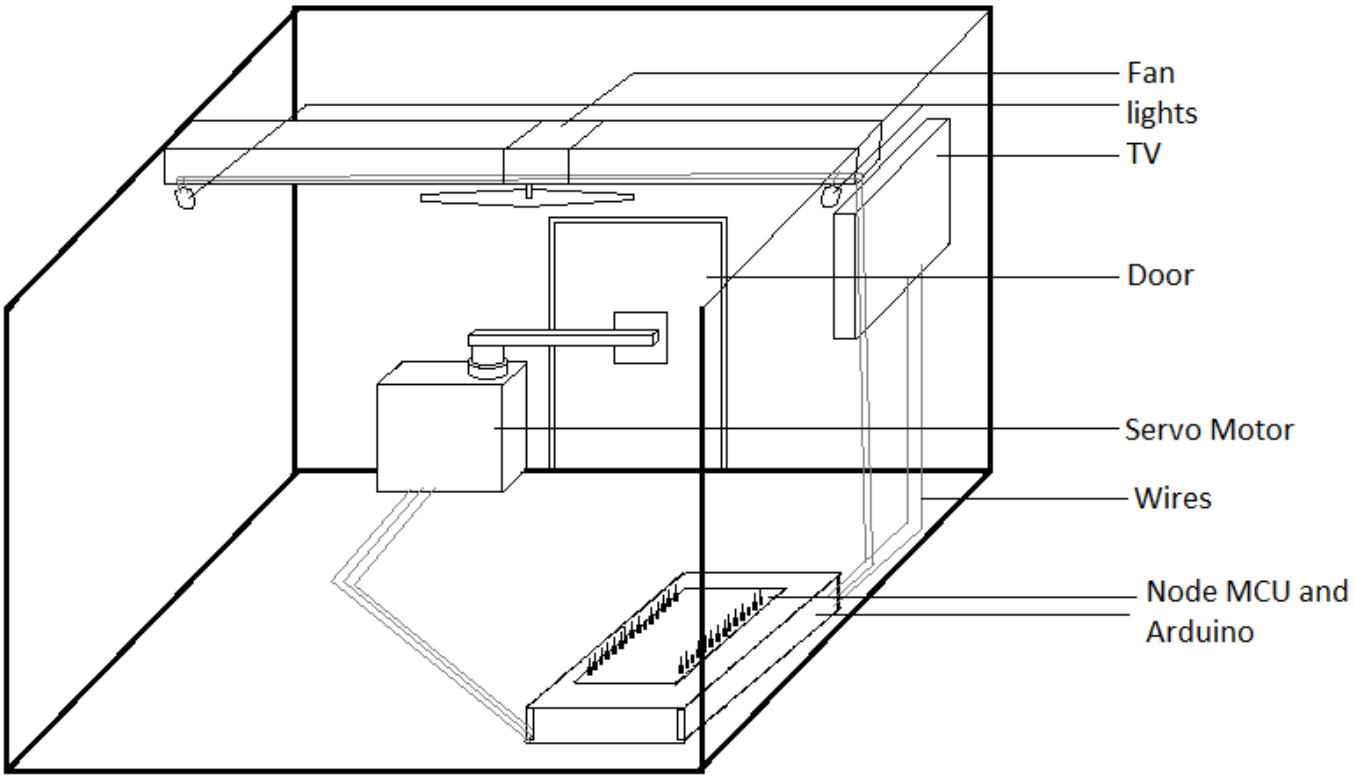
or any warning if there are smoke or rain sensor deployed inside home.

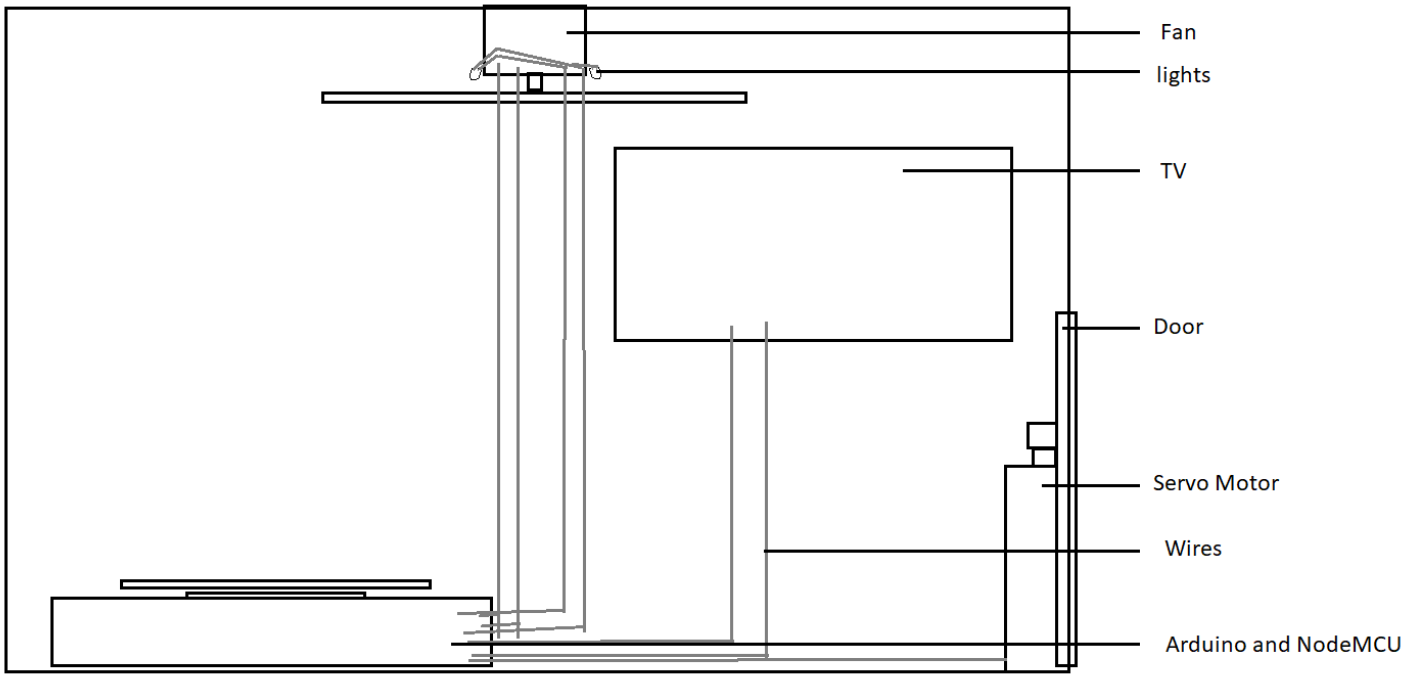


That's how our whole application works whether it is web or android or IOS.

Design of Proposed Work

Model Design:

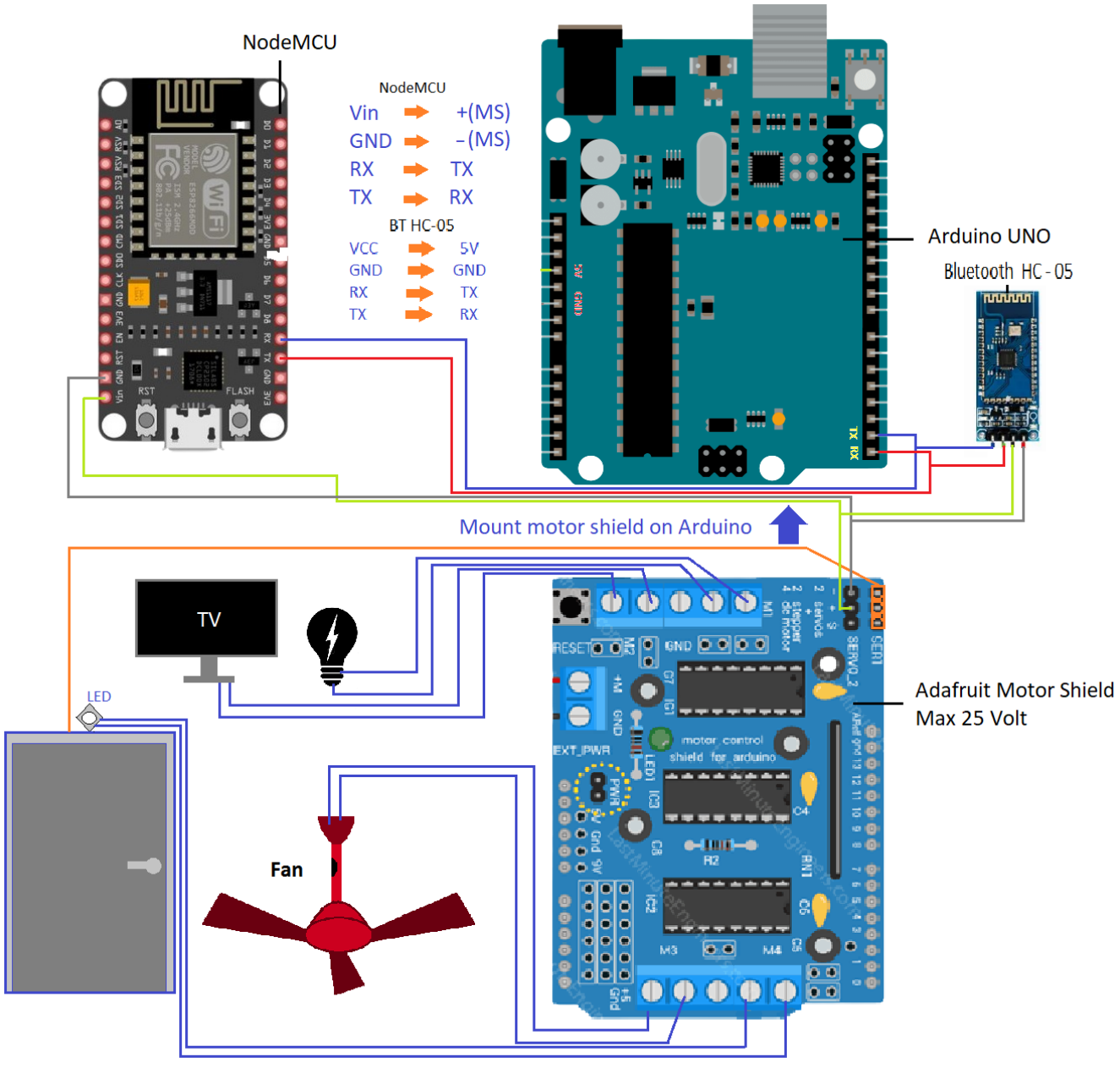




Description: this model consist of the basic wood body which is used as its frame to hold all other components and provide a look of small sized home

Secondary it consists all the electrical chipsets like Arduino UNO which is used as the commanding brain or master brain, NodeMCU which is used as the communication brain, Adafruit Motor Shield which is used to control all the components on specific voltage, a 9V DC motor used to make a small sized fan model, 6 LED (light Emitting Diodes) in which 3 of them are used as Lights, 1 as the door opening light, and remaining 2 are used in TV, one servo motor to show the opening and closing sessions of the door and to operate the door on the specific angle because single rotational motor can cause damage to other components and also consumes more voltage and to connect arduino and chipsets we have used jumper wires and for components twin phase wires.

Chipset Assembly:



Connection Procedure:

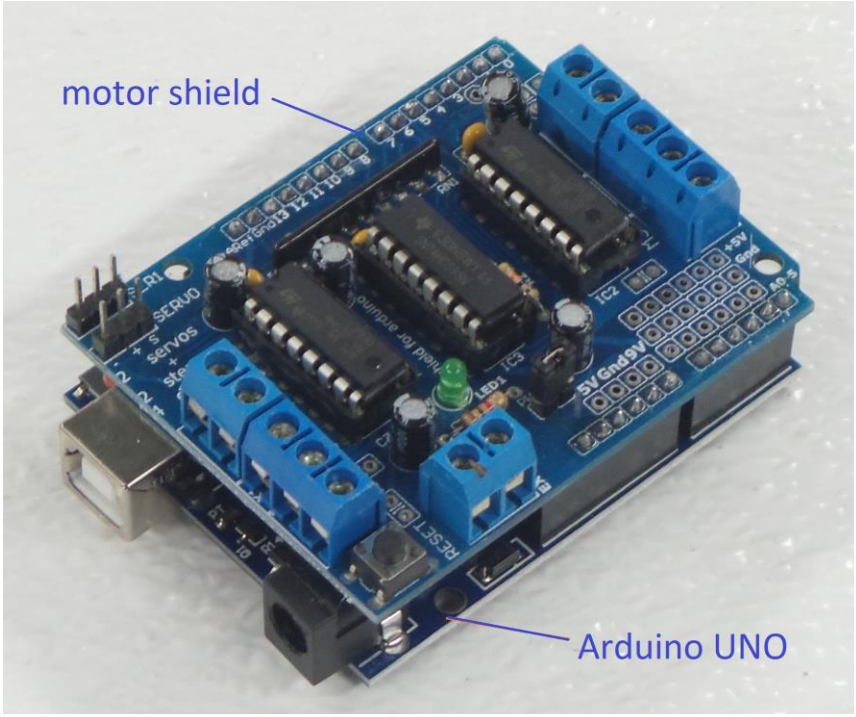
NodeMCU to Arduino = we have two brains for separate work and independent task procedure but we have to connect them to operate them on the basis of our commands use jumper wires to connect them.

First, we have to establish a current flow between these two brains so, connect Vin pin of NodeMCU to + servo2 point of motor shield and GND (ground) pin to GND to each other as shown in chip assembly diagram above.

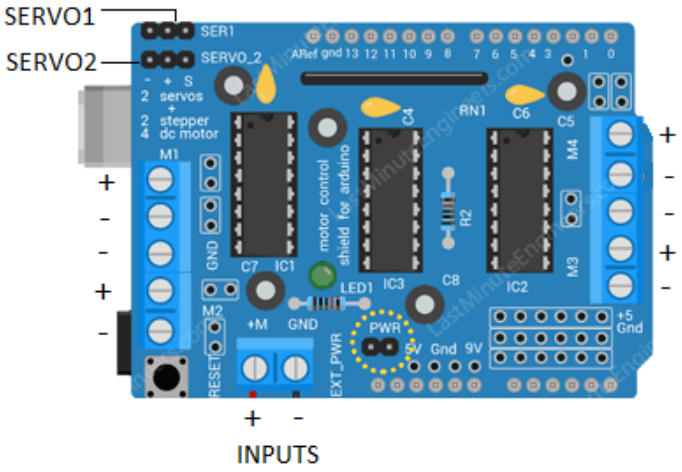
Secondary we have to establish a data sharing connection between them so connect RX (receive) pin to TX (transmit) of each other, remember we have to cut and fix the wires in RX TX ports of Arduino because we have to mount motor shield on it and it will also acquire the RX and TX ports, after all they could share data such as numbers and alphabets between them.

Now our brain assembly is ready so we should mount motor shield on Arduino.

Mount motor shield on Arduino According to the pins and the image shown here.



After that we must connect all component wires to the motor shield according to this figure.

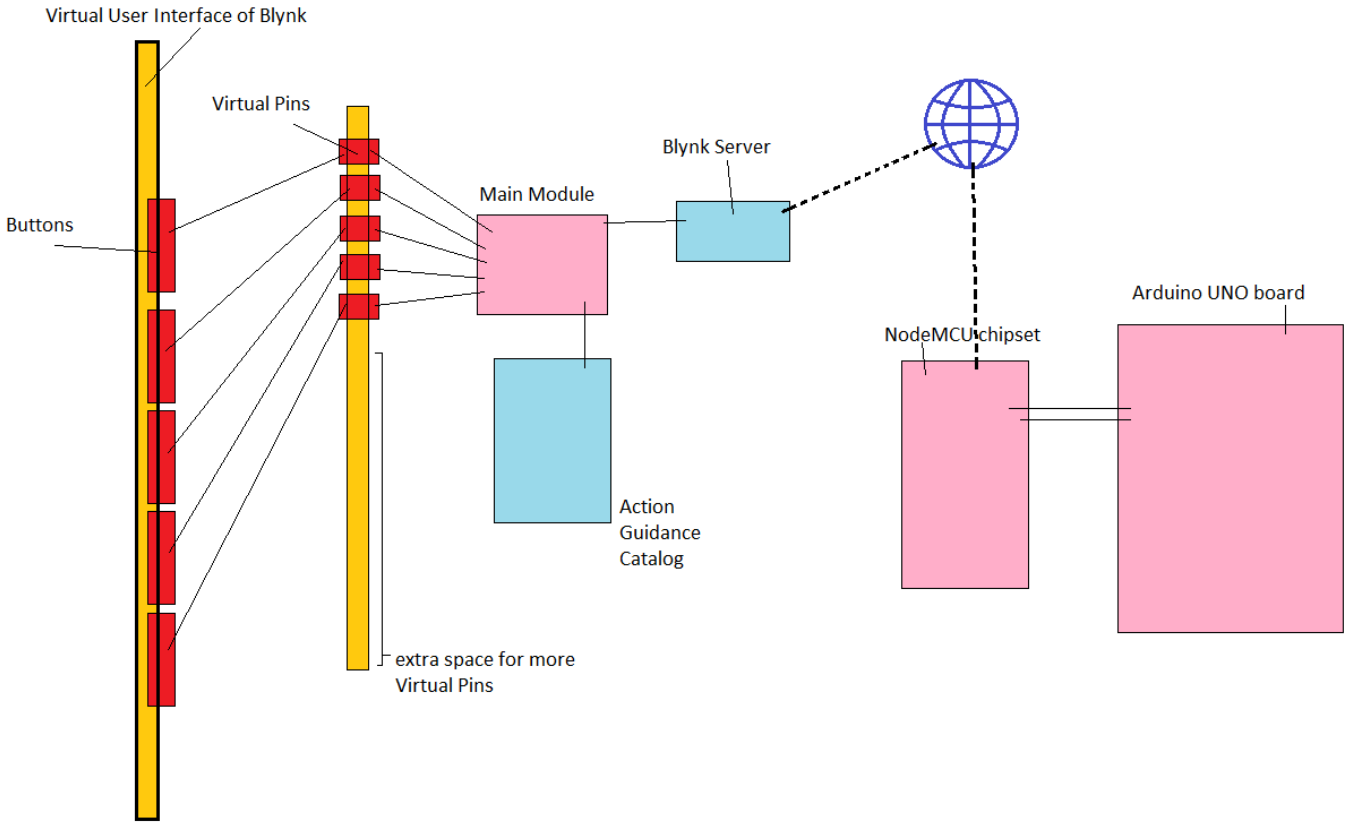


Connect DC motor, Door LED, TV and LEDs to the + - terminals of the shield at Separate points and Servo(Door Motor) to the Servo points.

========== Warning! Remove Power Jumper if Using Input Power Sources========= After all connections, we fixed all the component in the model and managed to fix the door with servo motor as described in the design.

Working Procedure of Smart Home Application

Application Working:



The working of the application is a little bit complicated

Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet.

This application is used to create a graphical interface or human machine interface (HMI) by compiling and providing the appropriate address of virtual pin on the available widgets.

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.

There are three major components in the platform:

Blynk App: – It allows you to create amazing interfaces for your projects using various widgets which are provided.

Blynk Server: – It is responsible for all the communications between the smartphone and hardware.

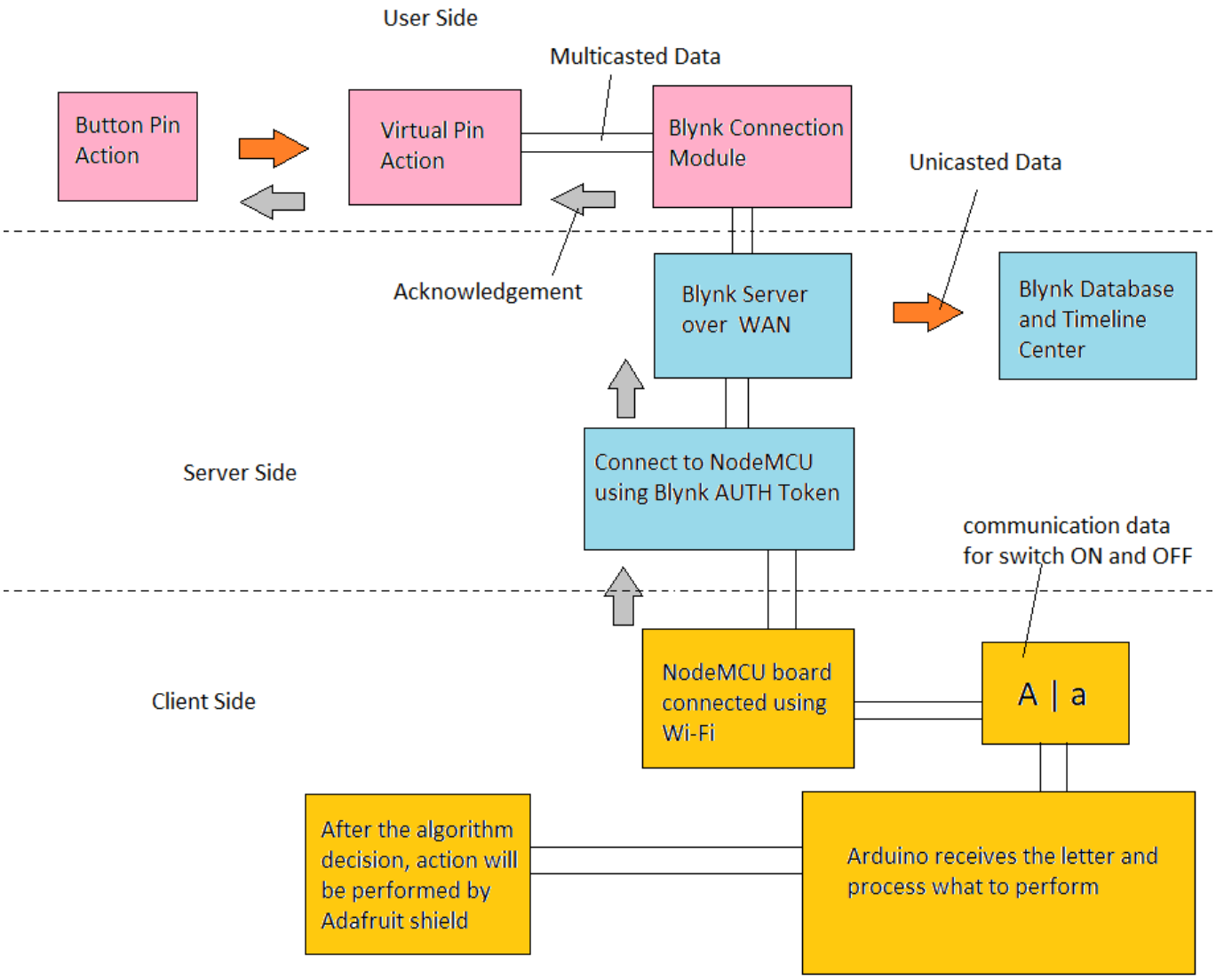
You can use the Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

Blynk Libraries: – It enables communication, for all the popular hardware platforms, with the server and process all the incoming and outgoing commands.

The process that occurs when someone presses the Button in the Blynk application is that the data will move to Blynk Cloud, where data finds its way to the hardware that has been installed using Blynk AUTH Token which is provided on the Blynk account.

It works in the opposite direction and everything happens in a blink of an eye. All this Data flow is described in Dataflow Diagram

Data Flow Diagram:



Journey starts from the button widget of user side interface, then it hits Virtual Pin which is configured to trigger an action, after that blynk connection module generates the data packets according to action triggered by the virtual pin and it will be uploaded to the blynk server, where the server generates the log of the action and timeline and stores it in database, while forwarding the flow towards NodeMCU, who is continuously trying to get a request from the AUTH Token provided to it.

The NodeMCU sends acknowledgement to the blynk user application that the action is successfully performed and sends appropriate letter to the Arduino Board and it orders motor shield to perform the action and all this happens very quickly in a blink of an eye.

Project:

Whole Idea Description:

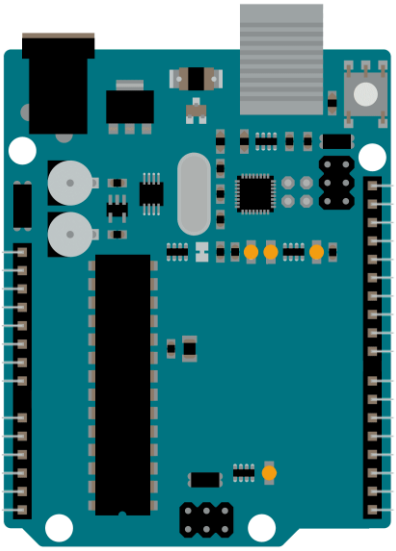
If we simply imagine about a smart home, then we can notice that according to our imagination a smart home actually is a home that is smart enough to manage day to day tasks without user's attention, it automatically senses that its raining outside and shuts the windows if any are open,

it is responsible if any type of fire is ignited without any known reason inside the home and takes appropriate actions to stop it or if it is out of control then it will inform the user as well as call the fire brigade team to handle the situation, it also senses the heat and smoke produced from cooking or from any other reason, Smart home is a home with brain who can manage the fix or predefined action tasks without your help.

The idea was to implement almost all possible ideas and techniques in this project but we have implemented basic utilities and control action.

Arduino Nano

Arduino UNO



First we have tried to use Arduino Nano for this one, but unfortunately it was not a perfect choice for Adafruit motor shield, working fine but unable to take acknowledgement from the shield, because of insufficient pins to plug with the shield, and another reason is that it has less amount of IC storage and ram to run the program, that's why we have selected Arduino UNO for the project,

This is a shorter area concept that's why UNO is enough but if we want to implement it in different rooms or in a whole building then we must use Arduino Leonardo as it is having much bigger storage and ram to handle a lot of extra instructions, and it is having more pins to communicate with ever room controller separately,

So first we have used UNO and adafruit shield as they are working well with each other, manually they are working well, then we deployed Bluetooth HC-05 module on baud rate of 9600 in slave mode as it is being controlled by the master device (Android).

Using Bluetooth it was working fine it is connecting quickly, working on a sufficient speed (not so fast), but our professor suggested a problem and that was the Bluetooth is having a limitation of shorter range maximum 100 meter or less than it, that's why we dropped the idea of controlling the project with Bluetooth,

And used Bluetooth for communication among devices, like in the master controller (the main chipset controller) a Bluetooth module will be plugged, for the time when internet is not working or it is down in your area, but you are inside your home, then you are enable to use smart services using Bluetooth, and Bluetooth will also work to control unwired or autonomous devices which could not be connected using wires

On internet based project we first decided to use our own application with own server and all supported components, then our professor suggested to use Blynk Platform Service, which is like an empty container who can host our application over the internet or WAN network, and Arduino is also having its library so we can use it directly in NodeMCU board, which is a Wi-Fi board and is having a separate brain to work independently, so we don't need any PC, Laptop or server including raspberry Pi boards which are continuously connecting the chipset to the Blynk server and our NodeMCU is working on its own while replacing the bigger machine by performing same task.

Then we have studied about all the plugins of blynk and ESP8266 as it is needed to configure NodeMCU, when we are confident to apply this whole scenario on the ground then we request our professor to help and guide us to perform the testing of chipset and after all we have successfully deployed all our components inside the wood built model.

The whole project comprises a lot of work done and it took around 15 days – 1 month from learning the programming to configuring and deploying it on the ground.

Coding and Explanation:

We used Arduino IDE to program all the chipsets including NodeMCU.

Arduino IDE 

Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. It's hardware products are licensed under a CC BY-SA license, while the software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the **Arduino Programming Language**, inspired by the Processing language, and used with a modified version of the Processing IDE. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) and a command line tool developed in Go.

The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

The name *Arduino* comes from a bar in Ivrea, Italy, where some of the project's founders used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

Communication Code for NodeMCU

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>



#define BLYNK_TEMPLATE_NAME "nodemcu"
#define BLYNK_AUTH_TOKEN "TTAdrred_ejdi928_authtokenfromyourblynkID"
char auth[] = BLYNK_AUTH_TOKEN;     // providing Bynk Auth Token



//   //////////WIFI here
char ssid[] = "DKNMU1";   // providing wifi name with which NodeMCU will connect
char pass[] = "knmu123";    //  password of the wifi




////////// appliances code from here
BLYNK_WRITE (V0){                // Declaring virtual pin 0 for  Lights
  int value = param.asInt();
  // Serial.println(value);
  if(value == 1){
    Serial.println(" A");                // transferring capital letter for switch ON
  }
  if(value == 0){
    Serial.println(" a");             // transferring small letter for switch OFF
  }
}



BLYNK_WRITE (V1){                // Declaring virtual pin 1 for  FAN
  int value = param.asInt();
  // Serial.println(value);
  if(value == 1){
    Serial.println(" B");
  }
  if(value == 0){
    Serial.println(" b");
  }
}
```

```
BLYNK_WRITE (V2){          // Declaring virtual pin 2 for  DOOR
  int value = param.asInt();
  // Serial.println(value);
  if(value == 1){
    Serial.println(" C");
  }
  if(value == 0){
    Serial.println(" c");
  }
}



BLYNK_WRITE (V3){              // ////// Declaring virtual pin 3 for   TV
  int value = param.asInt();
  // Serial.println(value);
  if(value == 1){
    Serial.println(" D");
  }
  if(value == 0){
    Serial.println(" d");
  }

}



void setup()
{
  // Debug console
  Serial.begin(9600);             // baud rate declared at 9600 Hz for Arduino

  Blynk.begin(auth, ssid, pass); // providing wifi details to NodeMCU
}




void loop()
{
  Blynk.run();       // Starting the program
}
```

Code Explanation

- In this code first we are including ESP8266 and Blynk libraries to use them.

- Then we are defining our board name and blynk authentication Id, so that our board can connect to our mobile application online.

- Then we are providing our Wi-Fi details to which our NodeMCU board will connect and access the internet for data communication.

- Then we are configuring appliance code in which we are setting virtual pins and their 0, 1 input point with if else and transferring Capital case and small case letters according to them.

- In this section it is first storing the value in variable and then matching it and accordingly send the letter.

- After that in Void setup section we have declared our baud rate at 9600 Hz and input the Wi-Fi details so that our board will be connected to it.

- And then we have started the code in void loop section so our board could run it from here.

Action code for Arduino:

```
#include <AFMotor.h>
#include <Servo.h>

Servo servo;


//initializing motors pins
AF_DCMotor motor1(1, MOTOR12_1KHZ);   // setting motor frequencies
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);


char val;
int Speeed = 255;                    //setting speed for motor
long a = 0;                  // declaring led blinker variable to check
                            // continuous working of chipset


void setup() {
  Serial.begin(9600);          // declaring baud rate frequency at 9600 Hz for NodeMCU
  servo.attach(10);                 // declaring port on which servo motor is attached
  pinMode(LED_BUILTIN, OUTPUT);        // declaring action output at builtin LED

}


void loop() {
  // builtin led blink function to check continous functioning
  a = a+1;      //it will add 1 in existing value of a with one loop


  if (a == 300000){      // after reaching 300000 cycles it will changes the value of a to 0
    a=0;
  }


  if(a == 1){               // just after changing the value of a LED will be turned ON
    digitalWrite(LED_BUILTIN, HIGH);
  }


  if (a==100000){                  // after reaching 100000 cycles LED will be turned OFF
  digitalWrite(LED_BUILTIN, LOW);
  }
```

```
////////////////////// motor operation code


 if(Serial.available() > 0){    // checking if there is something receiving from NodeMCU
  val = Serial.read();         //here it will store the letter which is received from NodeMCU
  Serial.println(val);         // it will also print the value received on the serial monitor

      if (val == 'A'){      // then it will check the letter and execute the function
        lights();                                  //where it will match
      }

      if (val =='a'){
         motor3.setSpeed(0); //Define minimum velocity // means lights OFF
         motor3.run(RELEASE); //stop the motor when release the button
      }

      if (val == 'B'){
      fan();
      }

      if (val == 'b'){
         motor2.setSpeed(0); //Define minimum velocity
         motor2.run(RELEASE);
      }

      if (val == 'C'){                    //////////////////////Door Opening
      motor1.setSpeed(150);              //turning on the door light
      motor1.run(FORWARD);
      dooropen();
      }

      if (val == 'c'){                    ////////////////////////Door Closing
         doorclose();
         delay(1000);              // waiting for 1 second
         motor1.setSpeed(0); //turning off the door light
         motor1.run(RELEASE);
      }


        if (val == 'D'){
      tv();   //tv on
      }


      if (val == 'd'){
         motor4.setSpeed(0); //Tv off
         motor4.run(RELEASE);
      }
```

```
        if (val == 'T'){
        Stop();   all turned off
        }
  }
  }

// declaring usable functions

void doorclose()
{
  servo.write(360);       // setting servo at 360 degree angle
}

void dooropen()
{
  servo.write(90);       // setting servo at 90 degree angle
}

void fan()
{
  motor2.setSpeed(150); //Define maximum velocity
  motor2.run(FORWARD); //rotate the motor anti-clockwise
}

void lights()
{
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD); //rotate the motor anti-clockwise
}

void tv()
{
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD); //rotate the motor anti-clockwise
}



void Stop()
{
  motor1.setSpeed(0); //Define minimum velocity
  motor1.run(RELEASE); //stop the motor when release the button
  motor2.setSpeed(0);
  motor2.run(RELEASE);
  motor3.setSpeed(0);
  motor3.run(RELEASE);
  motor4.setSpeed(0);
  motor4.run(RELEASE);
}
```

Explanation of Arduino Code:

- First, we are including the needed AFMotor and Servo libraries, to connect the code and control hardware shield.

- Initializing pins and frequencies for the ports, on which appliances are being connected.

- declaring variables to store the character value which is being received from serial communication, the motor speed indication quantity which will hold the voltage supplied to the port on which fan motor will be connected and variable named “a” to store the number of cycles the code is executed.

- In void setup section we are declaring the communication baud rate at 9600 Hz so that it can communicate with NodeMCU easily, also declaring the pin number on which the servo motor is connected and declaring the action output of code repetition on built-in LED.

- Then in void loop first of all,we have declared a function to blink the built-in LED of the Arduino board so we could check that code is running or stopped for some unknown reason, so 1 will be added in the existing value of “a” so it is the same value as the code is executed.

- Then if the value of loop cycle storing variable reached 300000 then it will automatically store 0 in a.

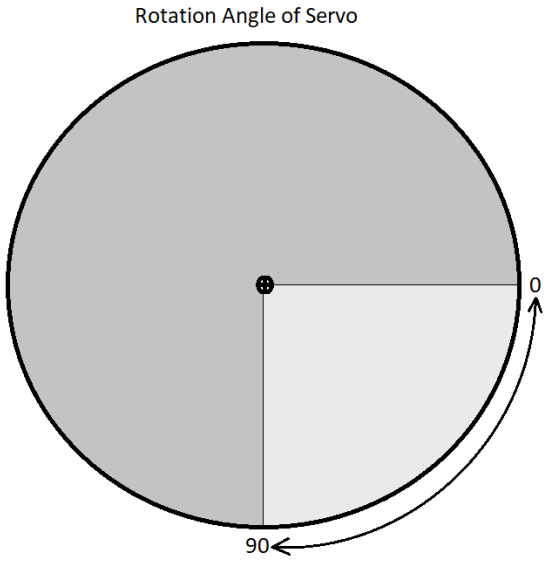
- Now when the code will execute next time it will store 0+1 =1, so the LED will be turned ON and when the value reaches 100000 cycles the LED will be turned OFF. So, the LED will Continuously blink after completing the cycles and it will work as an indicating mechanism and indicates that the code is running.

- Now, we are on at the motor code section here if Arduino receives something from NodeMCU on its communication pin, then it will execute the internal code which is to compare it, else it will continue to recheck.

- If it will receive something then it will store it in “val” variable and pass it from all the comparing if else statements and execute the appropriate function.

- The algorithms is like if letter is Capital case it will indicated the switch ON situation and if it is small case it will indicate switch OFF situation.

- In case of door the servo motor is set at the angle of 360/0 on closed and 90 on open situation.



- Before door open power will be supplied to the door LED so the door light will be switched on.

- And after door close there is a delay of 1 second and then door LED will be switched OFF.

- After the loop syntax all functions are declared here to use separately and for better readability.

- There is also a stop function which works to stop all the components, it is not implemented yet but it is being used by alarming protocols and quick emergency shutdown type mechanisms so it could cut the power from whole in case of fire or any other disaster situations.

Future Technical Upgrade:

Here are the upgrades mentioned, which could be implemented in this project.

- We may use drone to establish a better virtual attendance in the surrounding area.

- Autonomous devices could be used as they are more efficient to maintain tasks of home in comparison to drone, and also they could be manually controlled.

- Bio-Metric authentication could be used in the doors and inside the safety vaults to keep our money and jewellery safe.

- They might be a voice recognition and control or an AI involved in the scenario, who helps to manage the home properly, with AI we can even use face and voice recognition and could use it as authentication to enter the house.

- With the involvement of AI and much more sensors we could use them for fridge, like if its raining outside the fridge will decrease the cooling accordingly and if there is a sunshine with high temp then it will increase the cooling.

- We could introduce a new user caring feature, like if a user wants to switch on the AC on a specific temperature before it reaches home, so the whole room is already cooled, the AI could track his latest location continuously and when the user is on its way to home in a 10 min radius the AI will switch ON the AC on its own.

- Smart Mirrors could be used for playing music, TV could be used to display important mails, social media content.

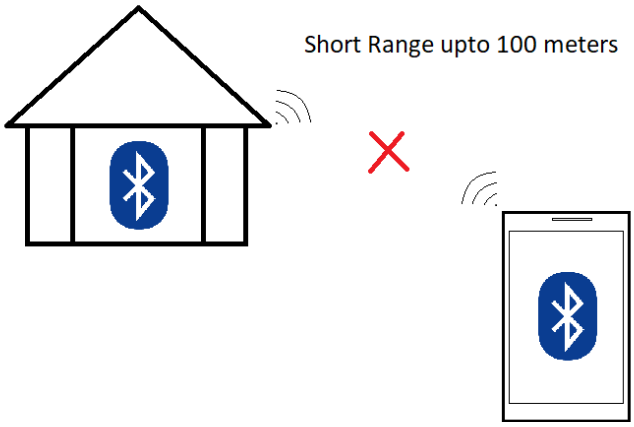
- In Company buildings the parking could be managed by and AI.

- Also, the day-to-day cleaning tasks like window cleaning, sweeping, dusting, clothes washing, toilet cleaning could be done by AI using the autonomous devices

- If someone known arrives on home in absence of user then AI will recognize his face with the camera and notify the user that your relative is outside the house, do you want to let him in and then the user could choose either to let him in or to tell him to meet him at the office or somewhere else.

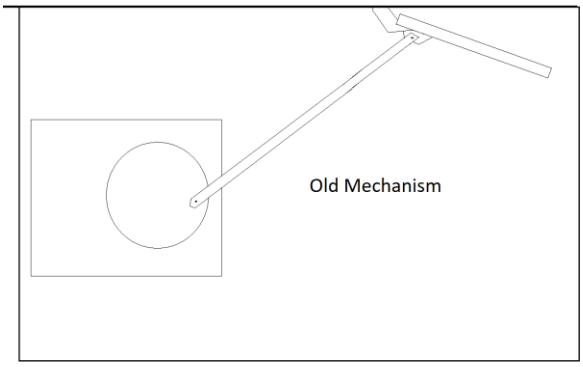
Problems solved and final output:

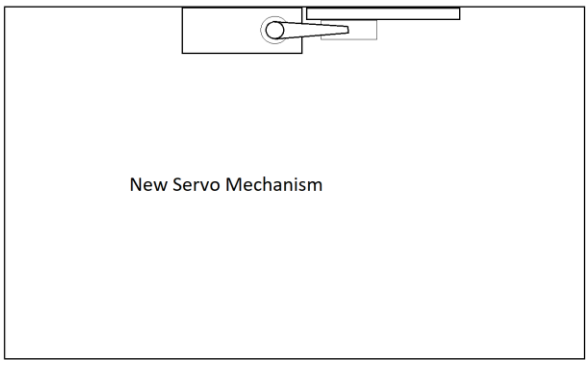
There were two major problems we faced while working on this project:



- First, the short range of Bluetooth in previous model for which we have discussed with our professor to find a substitute of Bluetooth, we were thinking to add local area network (LAN) to connect to the chipset, but our professor suggested that if you are already trying to solve the connection range problem why should we stick on a technique which is also having limited range which is larger than Bluetooth and could be increased but still it is not sufficient.

  So, as the final solution our professor suggested to use internet for connectivity and then they advised to learn to use blynk plugins, because using it we don't need to host our project over the internet.





- Second, previously we are having another mechanism to open and close the door, but it was not working sufficiently, it was slower, consumes more power and a lot of space, not accurate as it was opening the door randomly sometime the door is half open, sometimes it was not closed properly.

  To fix this issue we have used the servo motor for the door opening mechanism, servo is faster, consumes less power and lesser space, has an accurate opening and closing angle.

Bibliography

References and Sources:

1. Valuable Guidance tendered by Mr. Shashi Maurya, Dr. Rahul Kumar.

2. Websites for learning:

   ⇨ https://www.tutorialspoint.com/cplusplus/index.htm

   ⇨ https://www.youtube.com

   ⇨ https://www.learnvern.com

   ⇨ https://www.simplilearn.com

3. For Required Parts and Chipsets

   ⇨ https://www.amazon.in

   ⇨ https://www.flipkart.com

   ⇨ https://www.flyrobo.in/

4. To automate over WAN, we used Blynk –

   ⇨ https://blynk.io/

   ⇨ https://www.esp8266.com/

Conclusion:

Technically, this smart home project is a set of some few chipsets or circuits which are connected and programmed in such a way so they can communicate with your mobile devices and can control your home electricity flow according to your instructions.

Previously, this project was implemented using the Bluetooth HC-05 module, but as everything is having its limitations similarly Bluetooth also has its limitation effects on this idea, the communication range of this project with mobile phone or laptop, is limited to a fixed 50–100 meter area. Later with help of our professors we have successfully implemented the internet connectivity module in it using Blynk which is an IoT platform for IOS or android smartphones that is used to control Arduino, NodeMCU and other boards via internet.

Presently, this project is having basic implementation in it, which includes basic control of home appliances and door opening and closing functions, but later on we are having a lot of techniques to make this project futuristic and up to date like

1. Cameras for surveillance and drones for better virtual access of home.

2. Bio-Metric authentication could be used in the doors and inside the safety vaults to keep our money and jewellery safe.

3. With the involvement of AI and much more sensors we could use them for fridge, livelihood equipment in the home.

4. In Company buildings the parking could be managed by and AI.

5. Face Recognition like if someone known arrives on home in absence of user then AI will recognize his face with the camera and notify the user that your relative is outside the house, do you want to let him in and then the user could choose either to let him in or to tell him to meet him at the office or somewhere else.

6. Also, the day-to-day cleaning tasks like window cleaning, sweeping, dusting, clothes washing, toilet cleaning could be done by AI using the autonomous devices.

This smart home project has demonstrated the potential of technology to transform our living spaces and make our lives more comfortable, convenient and sustainable. With the advancements made in this project, we have opened up new opportunities for innovation and development in the field of smart home technology. As we continue to explore new ways to integrate technology. As we continue to explore new ways to integrate technology into our homes, we have the power to create a future where our homes are not just smart, but truly intelligent and adaptive to our needs.