DESCRIPTION DOCUMENT FOR WIFI EIGHT RELAY MODULE

HARDWARE REVISION 0.3

Department	Name	Signature	Date
Author			
Reviewer			
Approver			

Revision History

Rev	Description of Change	Effective Date
0.1	Initial Release	

ABSTRACT:

This document is a detailed product description that describes the effective features of the product. It includes a functional hardware description of the product with its internal block diagram and product images.

Table of Contents

Table of figures

1. ABBREVIATIONS

Term	Description
AC	Alternating Current
COM	Common
DC	Direct Current
HTTP	Hypertext Transfer Protocol
Hz	Hertz
MCU	Microcontroller Unit
MQTT	Message Queue Telemetry Transport
NC	Normally Closed
NO	Normally Open
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

2. REFERENCES

Company Website link	https://www.armtronix.in
Youtube Weblink	https://www.youtube.com/watch?v=N6_aVWY1ezk
Github's Weblink	https://github.com/armtronix/Wifi_eight_relay_board
Android Mobile App .APK Weblink	https://github.com/armtronix/Wifi_eight_relay_board

3. PURPOSE

The purpose of this document is to outline the design description for the Wifi Eight Relay Board. It provides a high level summary of the product.

4. SCOPE

This document describes system architecture which includes Power supply, Relay, WiFi Module, Microcontroller and UART to USB converter.

5. SAFETY AND WARNING

Note that, this board to be powered with AC 230V with required current. Work and handle carefully with AC power as it is harmful and danger for living beings. Touching live wire or board when it is ON is danger and not advisable, it may cause to death, please avoid it.

Even a 50 V AC supply is sufficient to kill you. Please Switch off the mains before you make or change connections, be very careful. If you are not sure of anything related to the AC supply lines, please call an electrician ask and him to help you with it. Do not attempt to interface to mains unless you have adequate training and access to appropriate safety equipment. Never work on high voltages by yourself when you are alone. Always ensure that you have a friend/partner who can see and hear you and who knows how to quickly turn off power in case of an accident. Use a 2A Fuse in series with the input to the board as a safety measure. Basic Wiring diagram is available on our instructables and github page. Please refer them.

Fire Hazard: Making wrong connections, drawing more than rated power, contact with water or other conducting material, and other types of misuse/overuse/malfunction can all cause overheating and risk starting a fire. Test your circuit and the environment in which it is deployed thoroughly before leaving it switched on and unsupervised. Always follow all fire safety precautions.

6. PRODUCT FEATURES

- Works directly with AC power 100 - 240 V AC 50-60 Hz.
- Device firmware can be updated/reloaded/changed as per user application.
- Eight relay outputs with NO, NC and interconnector COM pin accessible to user.
- Board can handle up-to 4 Amps of current at relay output.
- WiFi with MQTT or HTTP protocol
- On board USB – UART converter to program WiFi Module and MCU.
- Basic Firmware* to enter SSID and password to connect to the router
- Basic firmware* is capable to control device through HTTP or MQTT mode.
- Push Button on board Provided to Reset the ESP.
- Board is compatible and configurable to Amazon Alexa.

** Firmware is basic and simple only to test the basic functionality of the hardware, which may not work as per your application. The basic test code is available in our github link given above and you can achieve your application by tweaking the same or developing your own and reloading the firmware to hardware.*

7. PRODUCT DESCRIPTION

a. PHYSICAL DESCRIPTION

- AC to DC Power supply module
- Electro-mechanical Relay – 8 numbers
- Wifi Module
- USB - UART converter
- Microcontroller

b. FUNCTIONAL DESCRIPTION

Block Diagram

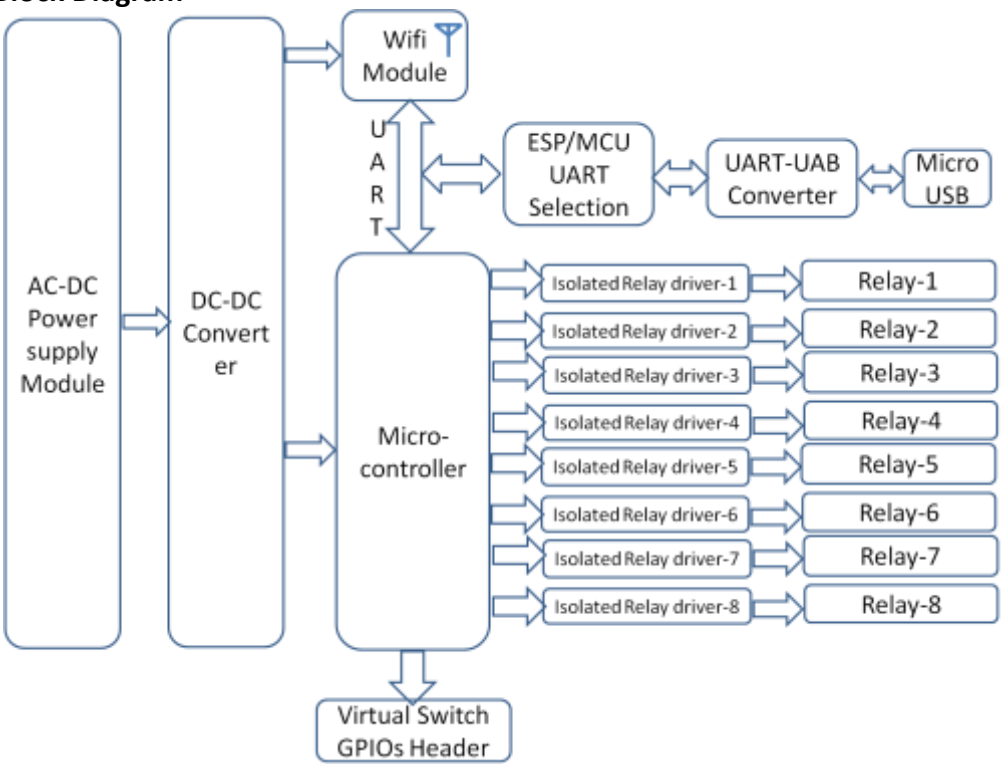

Figure 1: Block Diagram

Eight relay board has on-board power supply module which takes standard AC voltage as input and provides required DC voltage as output. The DC voltage is used to power-up Wifi module and Microcontroller Unit used on-board to establish Wifi communication with smart phones and control relays respectively. There are eight relays mounted on board with potential free contacts to control (ON/OFF) eight external electrical AC/DC loads independently from a smart phone application using MQTT/HTTP protocol. Where the COM pin of all the relays are connected to each other to make the connection easy.

8. SYSTEM OVERVIEW

1. AC to DC Power supply module

AC to DC converter is power supply module manufactured from Hi-Link part number HLK-PM05. This power supply module rectifies and regulates voltage from 230 V AC to 5 V DC with output current capacity of 1A DC. The power of HLK-PM05 is at maximum of 5W.

The 5V supply is used to power relays and USB-UART converter. There is a DC-DC converter on board to regulate voltage from 5 V DC to 3.3 V DC to supply power to Wifi module.

2. Wifi Module

Wifi module used on the board is ESP8266-12 with all its required GPIOs are easily accessible to user for their own application. Wifi module is powered on through 3.3 V DC. It works on both MQTT / HTTP protocol.

3. Microcontroller

The Microcontroller contains the relay-control algorithm to switch ON/OFF the relay, by receiving commands from ESP8266-12. Controller communicates with Wifi module through UART mode of communication to send and receive data to-and-from in connected Wifi network. ATmega328P microcontroller is used on the board to execute the algorithm, which is manufactured by Atmel incorporation and is compatible with Arduino IDE.

Controller has the capability for software up-gradation through Arduino and contains enough memory to upgrade and store the software for its own application.

4. Mechanical Relay – 8 Numbers

All relays are powered by 5 V DC. The three load terminals (COM, NO and NC) of all relays are given accessible to user to control loads independently. An opto-isolator based relay-driver circuit is used to drive the relay. All the common pins of the relay are interconnected to each other, to make ease of installation in similar load switching application.

5. USB-UART converter

USB-UART converter is an integrated chip used to convert serial UART data to high speed USB to program the Wifi module and MCU using Arduino IDE. This is much user friendly to customize the code and reload it. A micro USB connector given on board to make hassle free connection between computer and Wifi Eight relay board for programming purpose.

The same port can also be used to push messages to serial port of PC from MCU or ESP by selecting the respective one by configuring header P5. Note that you can connect only of them at an instant of time to serial port.

9. TECHNICAL SPECIFICATION

a. ELECTRICAL SPECIFICATION

Input Specifications				
Description	**Min**	**Typ**	**Max**	**Unit**
Voltage AC	100	220	230	Volts
Current AC	-	0.3	-	Amps
Power AC	-	3	-	Watts
Frequency	50	-	60	Hz

Relays Output Specifications (Maximum)				
Description	**Min**	**Typ**	**Max**	**Unit**
Voltage AC	-	-	240	Volts
Current AC	-	-	4	Amps
Power AC	-	-	980	Watts
Voltage DC	-	-	24	Volts
Current DC	-	-	4	Amps

b. MECHANICAL SPECIFICATION

- Mechanical Dimensions of PCB are 140 x 80 x 23 mm (Length x Width x Height)

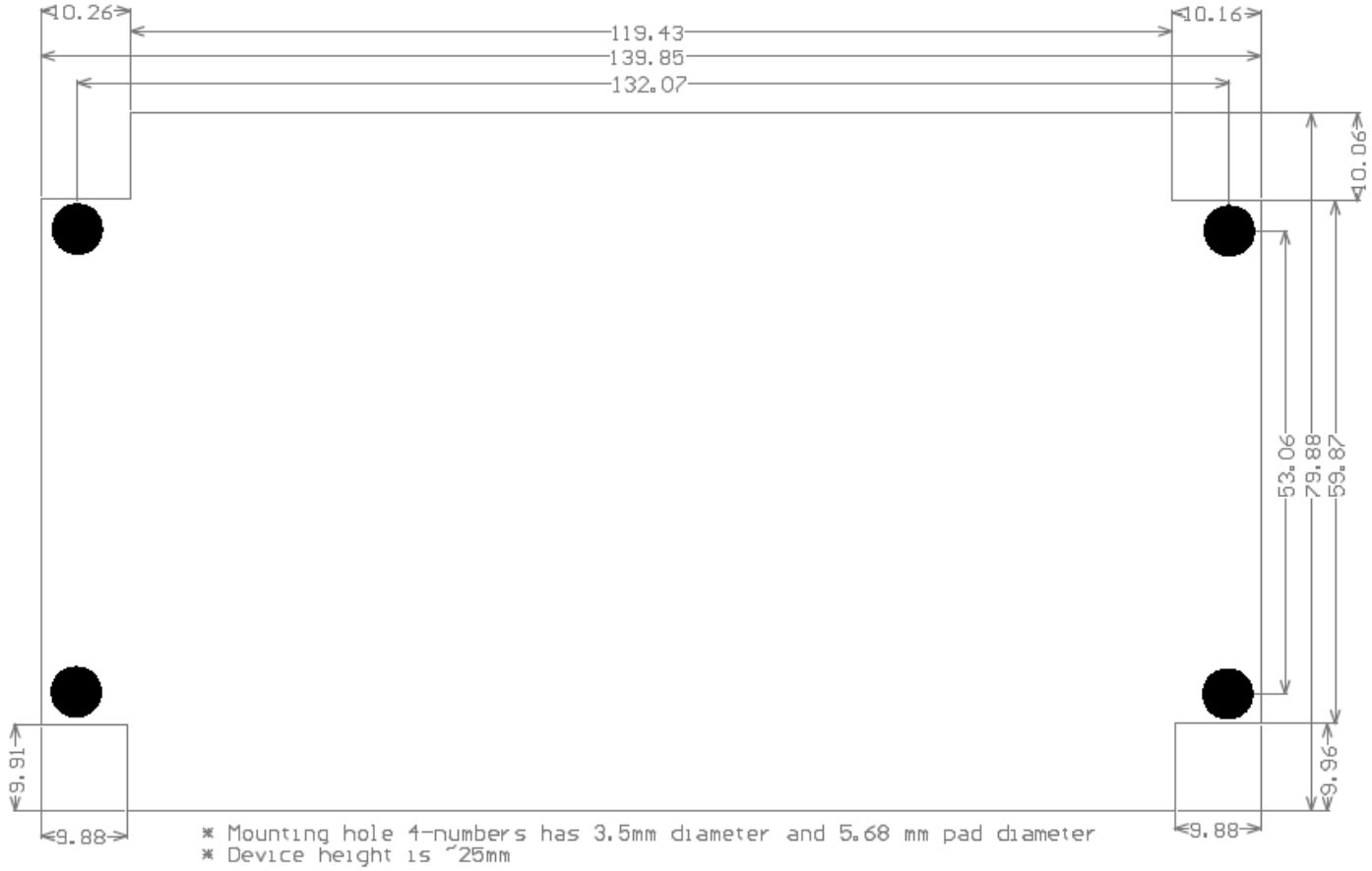

Figure 2: Mechanical dimensions of PCB

10. ELECTRICAL CONNECTIONS

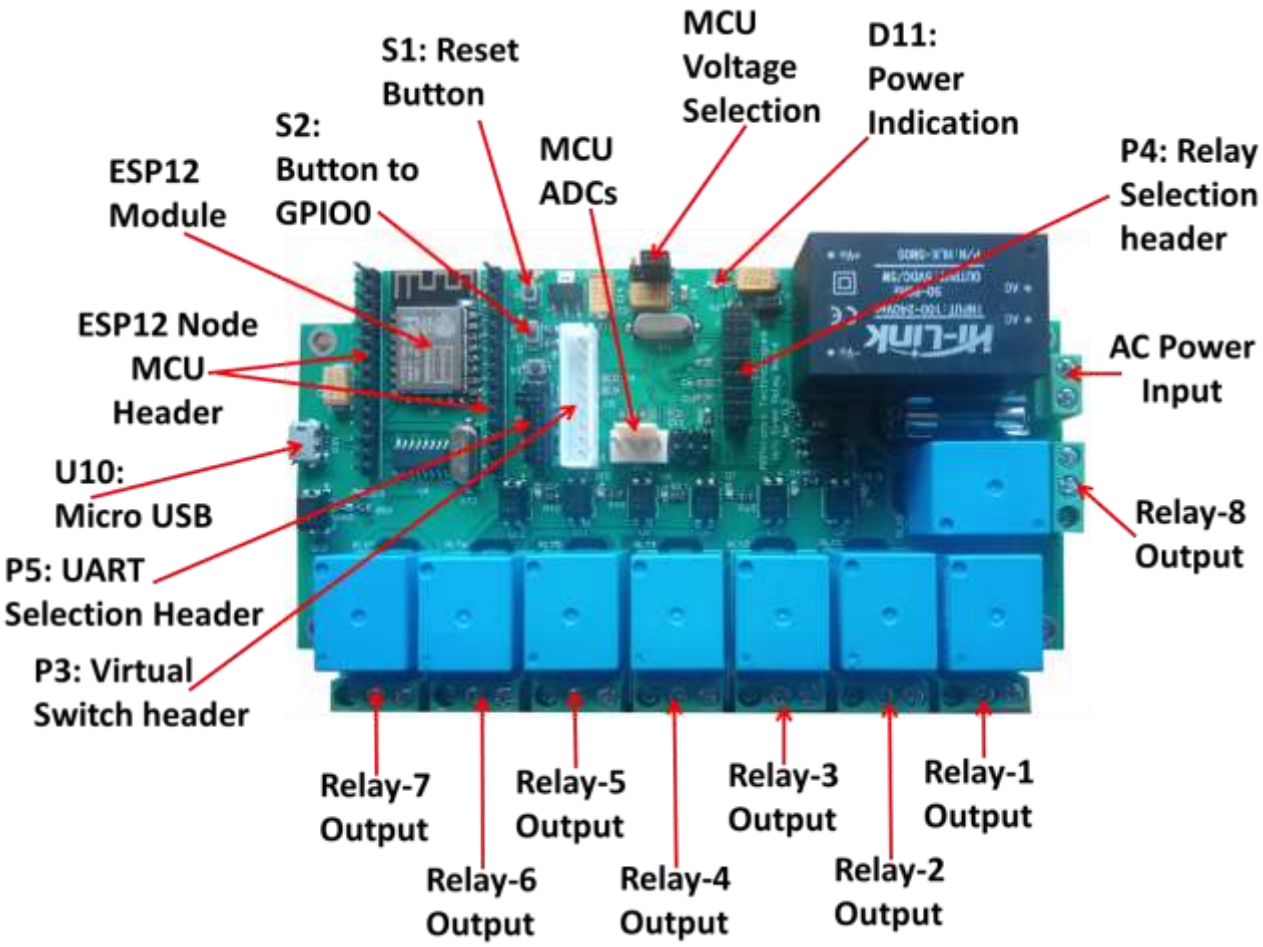

Figure 3: Header and Switch Details

Description of Header and Switches shown in Figure 3

1. S1 Button is to reset the ESP.
2. AC_IN AC power input terminal block.
3. F1 Protection replaceable glass tube fuse.
4. P4 Relay Selection Header.
5. J5 and J6 Headers are compatible to standard NodeMCU headers.
6. U10 Micro USB for programming.
7. P3 Virtual switch GPIOs header.
8. P5 UART configuration header.
9. P2 MCU Power selection header.
10. J11 MCU ADCs and I2C

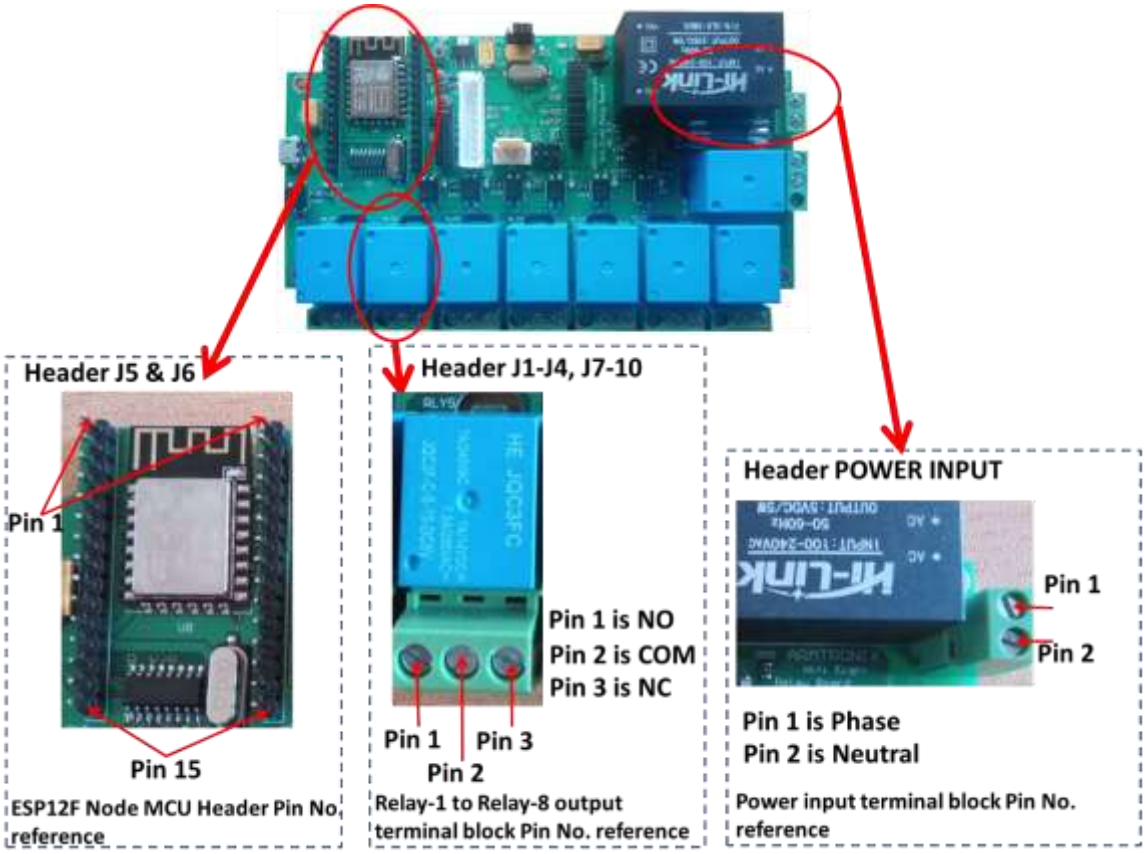

Figure 4: Header Pin number references – 1

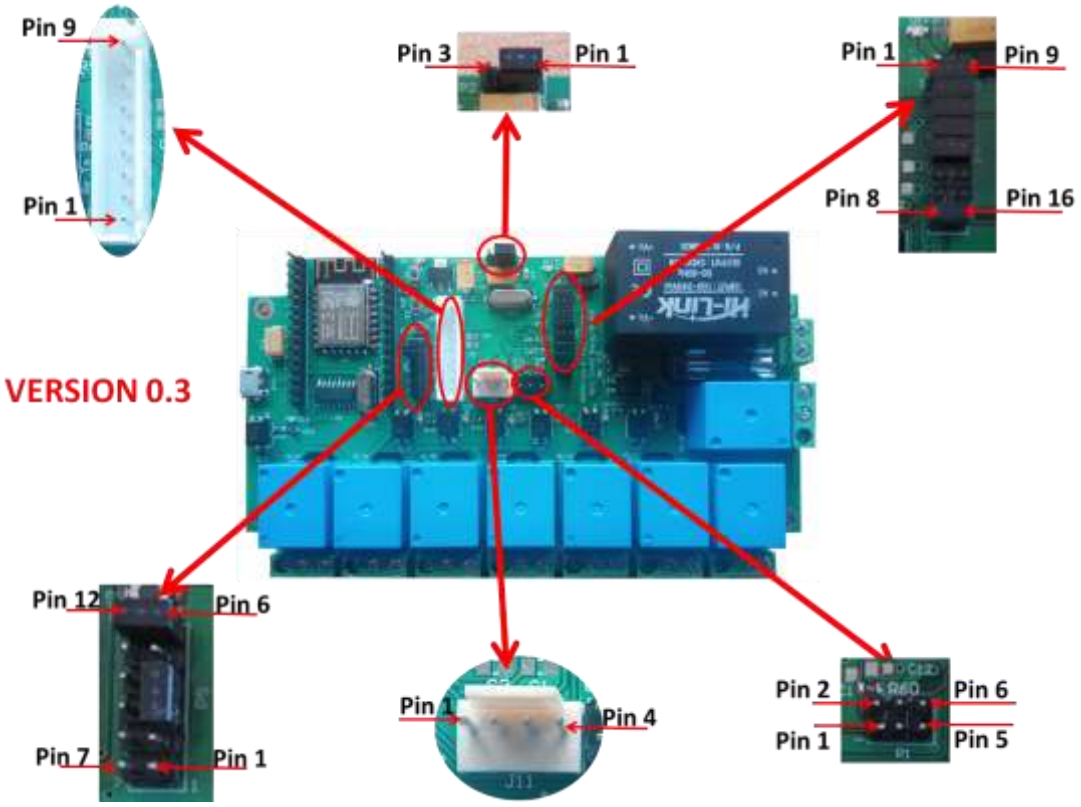

Figure 5: Header Pin number references - 2

a. HEADER PIN CONFIGURATION

i. HEADER J5

J5				
Header Pin #	**ESP Pin #**	**Arduino Port #**	**Designation**	**Description**
1	-	A0	ADC_EX	External ADC input up to 3.3Vdc
2	2	-	ADC	ADC input up-to 1.1Vdc
3	-	-	RESV	Reserved by ESP
4	-	-	SD_D3	Reserved by ESP
5	-	-	SD_D2	Reserved by ESP
6	-	-	SD_D1	Reserved by ESP
7	-	-	SD_CMD	Reserved by ESP
8	-	-	SD_D0	Reserved by ESP
9	-	-	SD_CLK	Reserved by ESP
10	-	-	GND	Ground reference
11	-	-	VCC_3V3	3.3V DC Input/output
12	3	-	EN	Enable of ESP
13	1	-	nRST	Negative of Reset
14	-	-	GND	Ground reference
15	-	-	VCC_5V	5V DC Input/output

Table 1: Header J5 Pin Configuration

ii. HEADER J6

J6				
Header Pin #	**ESP Pin #**	**Arduino Port #**	**Designation**	**Description**
1	4	D0	GPIO16	GPIO16 of ESP
2	20	D1	GPIO5	GPIO5 of ESP
3	19	D2	GPIO4	GPIO4 of ESP
4	18	D3	GPIO0	GPIO0 of ESP
5	17	D4	GPIO2	GPIO2 of ESP
6	-	-	VCC_3V3	3.3V DC Input/output
7	-	-	GND	Ground reference
8	5	D5	GPIO14	GPIO14 of ESP
9	6	D6	GPIO12	GPIO12 of ESP
10	7	D7	GPIO13	GPIO13 of ESP
11	16	D8	GPIO15	GPIO15 of ESP
12	22	D9	RXD0	UART Tx ESP
13	21	D10	TXD0	UART Rx ESP
14	-	-	GND	Ground reference
15	-	-	VCC_3V3	3.3V DC Input/output

Table 2: Header J6 Pin configuration

Table 2 and 3, shows the header J5 and J6 which are in-compatible with Node MCU headers. Freely available GPIOs are also shown in connector, can be used for user application.

iii. HEADER J4: Relay selection

J4						
Header Pin #	**MCU Pin #**	**Arduino Port #**	**Designator**		**Header Pin #**	**Description**
1	9	5	MCU_PD5		9	RELAY-1
2	10	6	MCU_PD6		10	RELAY-2
3	12	8	MCU_PB0		11	RELAY-3
4	13	9	MCU_PB1		12	RELAY-4
5	14	10	MCU_PB2		13	RELAY-5
6	32	2	MCU_PD2		14	RELAY-6
7	1	3	MCU_PD3		15	RELAY-7
8	2	4	MCU_PD4		16	RELAY-8

Table 3: Header J4 Pin Configuration

iv. HEADER P3: Virtual Switch MCU GPIOs

P3				
Header Pin #	**MCU Pin #**	**Arduino Port #**	**Designation**	**Description**
1	11	7	MCU_PD7	Can be used for virtual switch
2	15	11	MCU_PB3	Can be used for virtual switch
3	16	12	MCU_PB4	Can be used for virtual switch
4	17	13	MCU_PB5	Can be used for virtual switch
5	23	A0	MCU_PC0	Can be used for virtual switch
6	24	A1	MCU_PC1	Can be used for virtual switch
7	25	A2	MCU_PC2	Can be used for virtual switch
8	26	A3	MCU_PC3	Can be used for virtual switch
9	-	GND	GND	Ground reference

Table 4: Header P3 Pin Configuration

v. HEADER P5: UART selection for programming.

P5						
Header Pin #	**Designator**	**Description**		**Header Pin #**	**Designator**	**Description**
1	RXDA	UART_Rx_MCU		7	TXD0	Programmer UART_Tx
2	TDXE	UART_Tx_ESP		8	RXD0	Programmer UART_Rx
3	RXDE	UART_Rx_ESP		9	TXD0	Programmer UART_Tx
4	TXDA	UART_Tx_MCU		10	RXD0	Programmer UART_Rx
5	DTR_E	UART_DTR_ESP		11	DTR	Programmer UART_DTR
6	RESET	MCU Reset		12	DTR	Programmer UART_DTR

Table 5: Header P5 Pin Configuration

vi. HEADER P1: ISP/SPI Header

P1							
Header Pin #	**MCU Pin #**	**Arduino Port #**	**Designator**	**Header Pin #**	**MCU Pin #**	**Arduino Port #**	**Designator**
1	16	12	MCU_MISO	2	-	-	MCU_VCC
3	17	13	MCU_SCK	4	15	11	MCU_MOSI
5	29	-	MCU_RESET	6	-	-	MCU_GND

Table 6: ISP Programmer / SPI Header

vii. HEADER J11: MCU ADCs and I2C

J11				
Header Pin #	**MCU Pin #**	**Arduino Port #**	**Designation**	**Description**
1	27	A4	MCU_PC4_SDA	Can be used as GPIO or I2C or ADC
2	28	A5	MCU_PC5_SCL	Can be used as GPIO or I2C or ADC
3	22	A7	MCU_ADC7	Can be used only as ADC
4	19	A6	MCU_ADC6	Can be used only as ADC

Table 7: MCU ADCs and I2C

viii. HEADER P2: MCU Power selection header

P2		
Header Pin #	**Designator**	**Description**
1	VCC_5V	5VDC
2	VCC_MCU	MCU voltage selection either 5V/3.3V
3	VCC_3V3	3.3VDC

Table 8: MCU Power selection Header

b. APPLICATION WIRING DIAGRAM

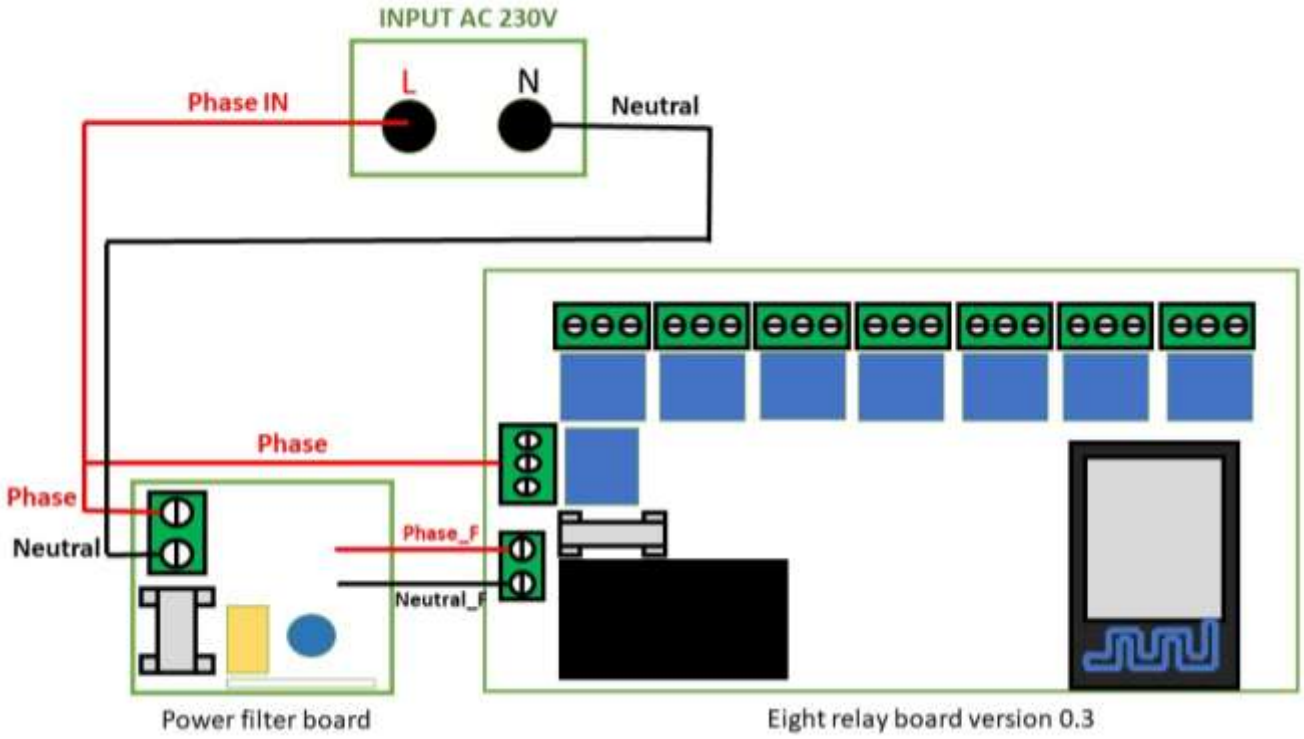

Figure 6: Power Filter board connection wtih BA012

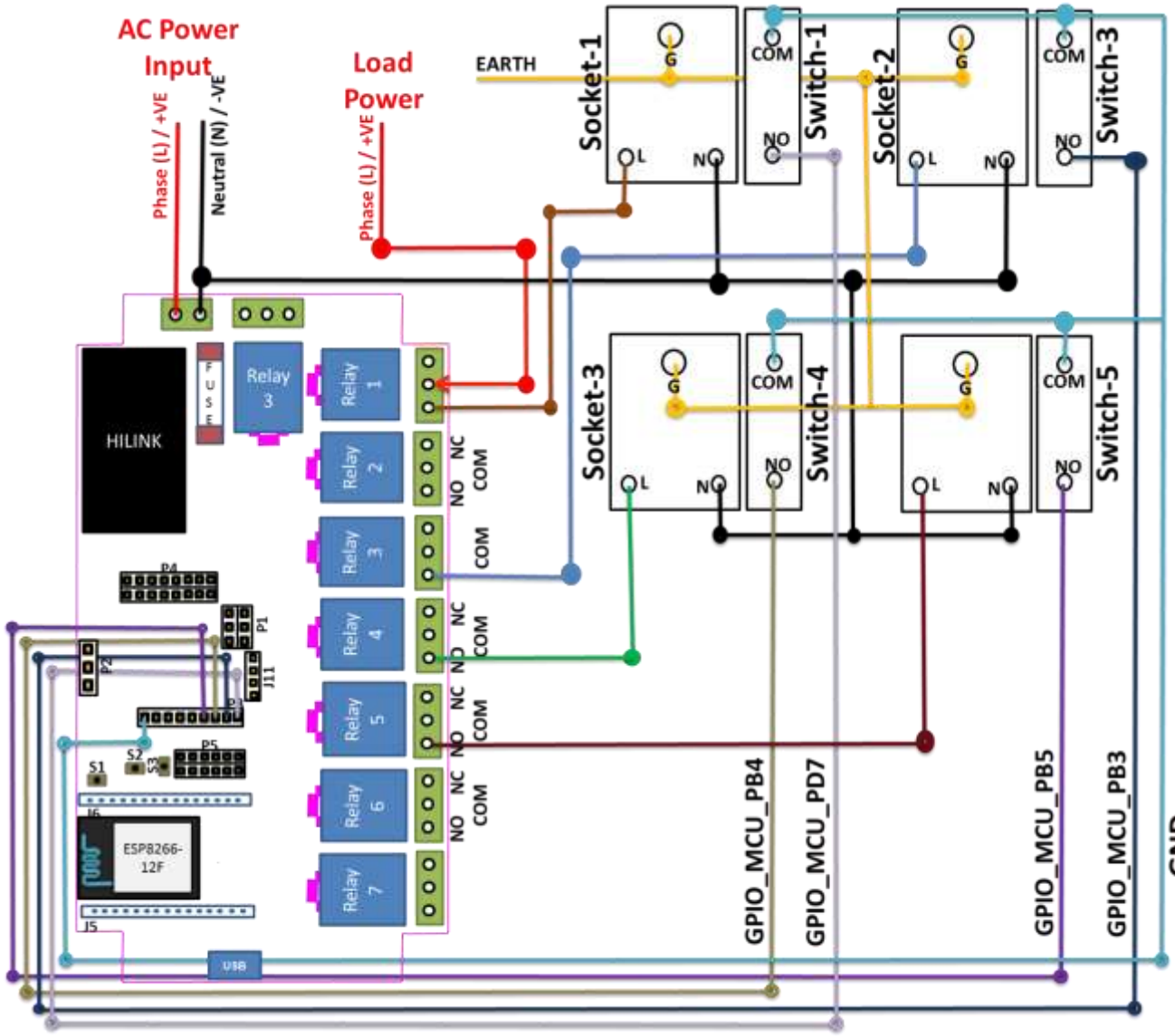

Figure 7: Application wiring example

Figure 6 represents about connection between load and relay output (J1-J4, J7-J10) connectors. Phase is given to Common terminal and load shall be connected to the NO/NC terminal of the relay. Output of relay-1, 3, 4, 5 are used to represent example application connection diagram. In an above wiring diagram, an output of relays are connected to Three-pin sockets respectively through NO pin of relay. That means the socket will get power only when the relay is triggered.
If the socket connected by NC, then socket will get power by default and disconnects power when the relay is triggered.

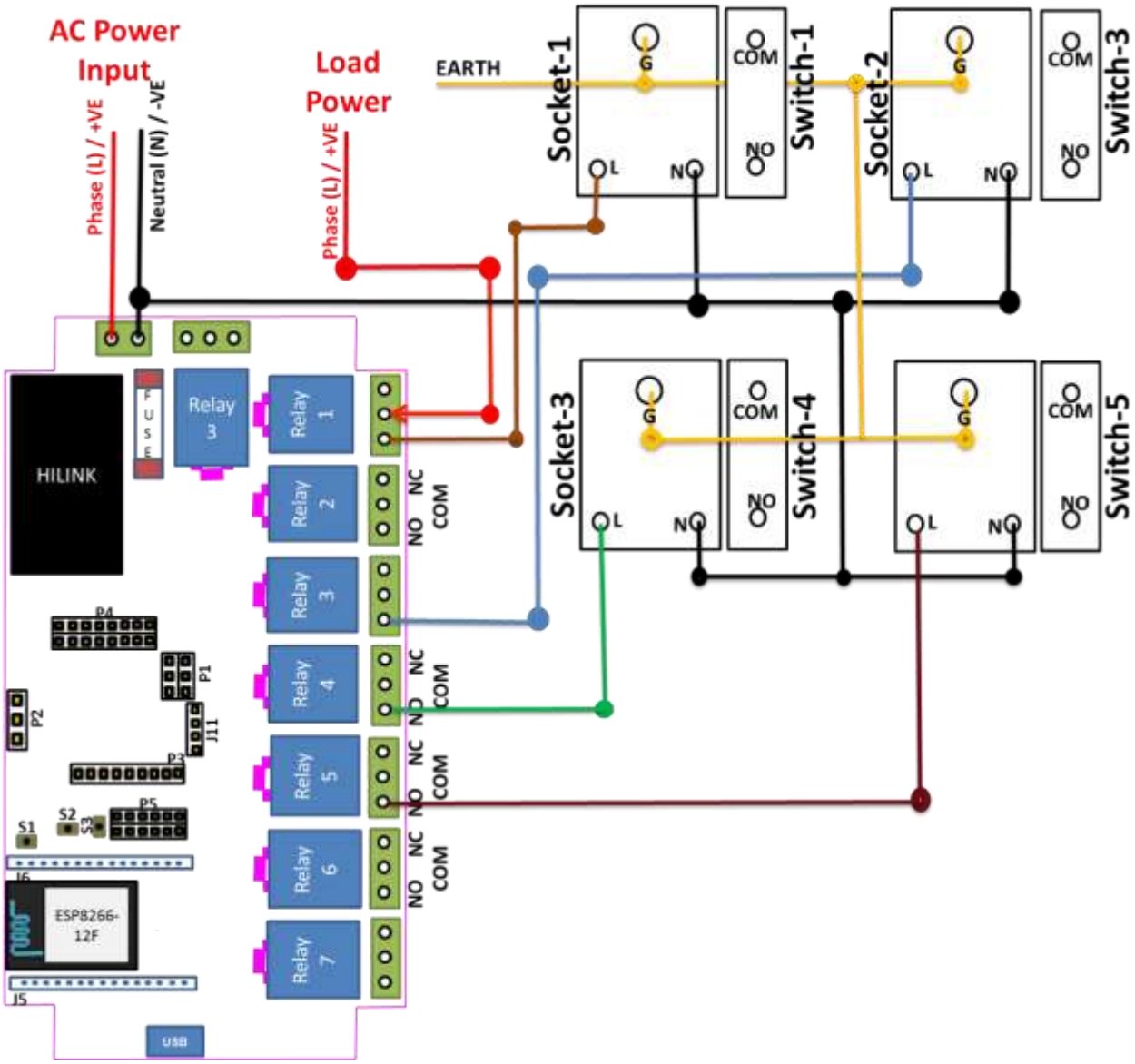

Figure 8: Relay output Connections diagram

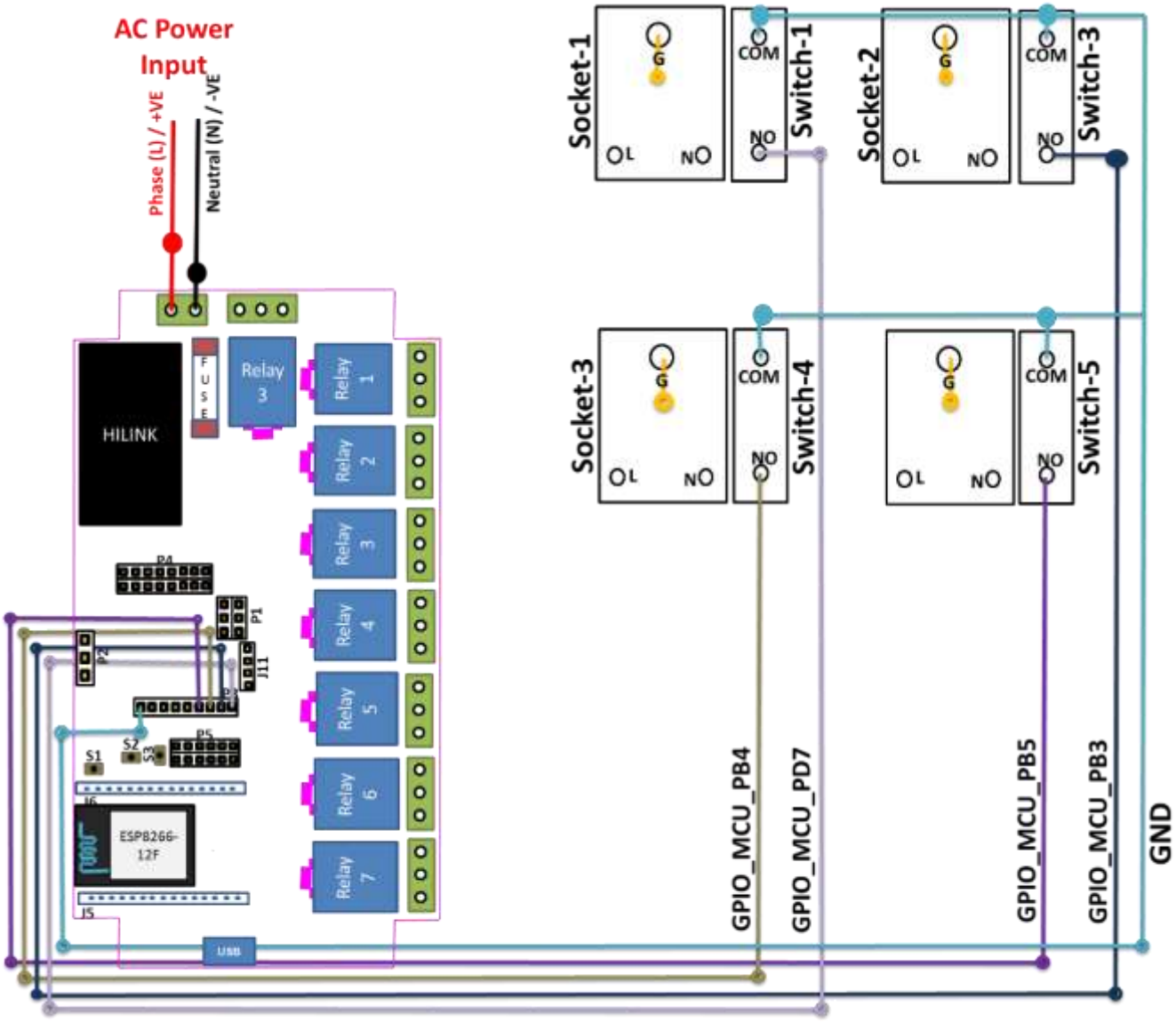

Figure 9: Virtual Connections.

11. HOW TO USE THE PRODUCT

Power ON the device, so that, it will host the access point as shown in Figure 10,

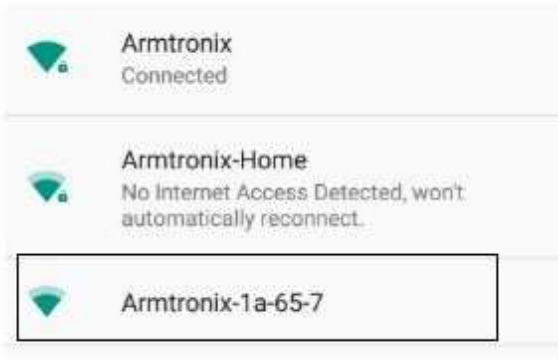

Figure 10: Device hosting Access point

Connect the mobile to access point with Armtronix-(MAC ID). EX: Armtronix-1a-65-7 as shown in Figure 10.

Figure 11: Access point name

After connecting, open browser and enter 192.168.4.1 IP address, it will open the web server as shown in the Figure 12,

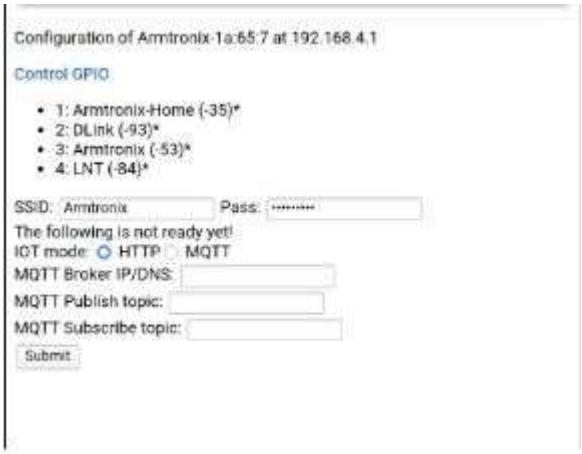

Figure 12: Web server

fill the SSID and password and select HTTP, if user wants to connect to MQTT then user has to

select MQTT radio button, enter MQTT broker IP address, enter MQTT publish topic then MQTT subscribe topic and submit.

After submitting configuration, the ESP 8266 will connect to the router and router assigns IP address to the ESP. Open that IP address in the browser to control the switch (Relay).

Without configuring the SSID and Password we can control the Wifi Switch by connecting to the access point of the device and open the IP address of device i.e 192.168.4.1 the web server page will show the link with the name Control GPIO as shown in the Figure 8, by clicking this link we can control the Wifi Switch board but the response will be slow.

a. IF DEVICE LOADED WITH TASMOTA FIRMWARE:

1. Make an input AC phase and Neutral connection as shown in Figure 5.
2. Use an external electrical fuse and MCB with rating 2A/250V, in series to input connections for the purpose of safety.
3. Make sure that there is no short circuit between phase and neutral.
4. Ensure that, required safety precautions are taken care.
5. Once the device Powered ON, it will host an access point as shown in Figure 13 below:

Figure 13: Hosting device with tasmota name

1. After connecting, open any browser and enter the IP address as 192.168.4.1 on your Smartphone it opens the Web Server as shown in the Figure 10 below, then enter the SSID and Password and you can give the Host name of your choice (just for your reference) then click on SAVE button.

Figure 14: Web Server

Once the **Configuration** is done it displays the message as shown in the figure 15 below,

Figure 15: Saving Configuration

Then user get access to control the load i.e he can Switch ON / OFF using TOGGLE button as shown in the figure 16 below.

Figure 16: Controlling a load

2. After the device get restarted, you will get a home as shown in figure 15 below and device is ready to use.

3. One more advantage is that the user can easily **Configure** and **Upgrade** his **Firmware** through smartphone and an option also provided to choose file directly from the phone if it is available as shown in figure 17.

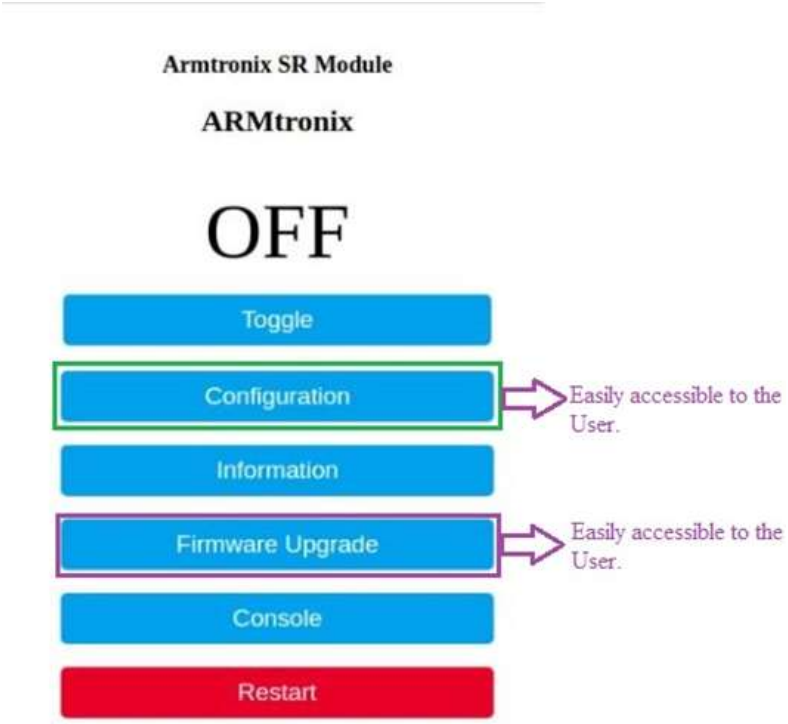

Figure 17: Main Menu

i. CONFIGURING DEVICE FOR MQTT MODE:

4. click on the Configuration option as shown in the figure 18,

Figure 18: Configuration Option

5. Click on Configure MQTT option provided as shown in the figure 19 below,

Figure 19: Option to configure to MQTT

6. User shall click on the SAVE button as shown in figure 20 below, once after entering required details to get access to MQTT protocol from the web server.

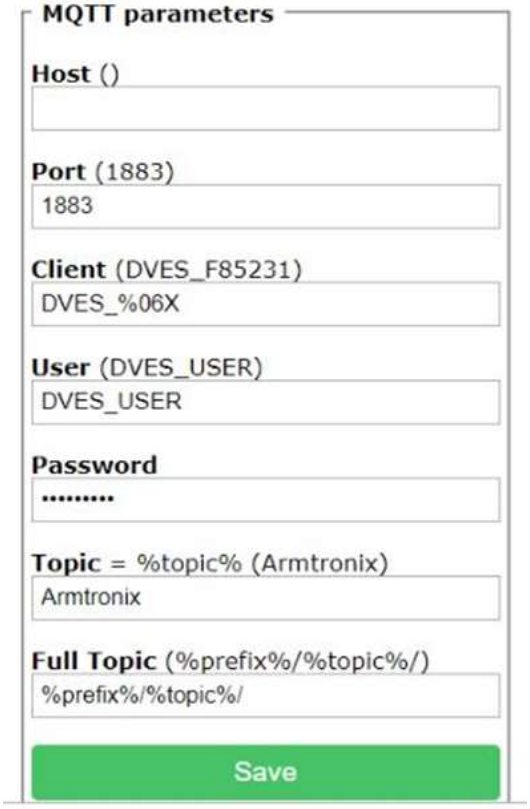

Figure 20: Option to enter required MQTT details

12. HOW TO CUSTOMISE FIRMWARE

You can program this board using Arduino IDE. Please follow the below steps to program the board by yourself with easy steps as mentioned below:

a. STEPS TO LOAD PROGRAM TO ESP8266:

1. Use external USB-UART converter between computer and this board.
2. Short Pin-2 and Pin-8 of bergstick-header P5 using removable jumper.
3. Short Pin-3 and Pin-9 of bergstick-header P5 using removable jumper.
4. Connect Micro USB cable between your computer and U5 of “Wifi Eight Relay Board”.
5. Open your code in Arduino IDE as shown.
6. Click on Tools Tab, move mouse pointer on “Board: xxxxxxxxxx” and click on “NodeMCU0.9 (ESP-12 Module)” as shown in figure 21.

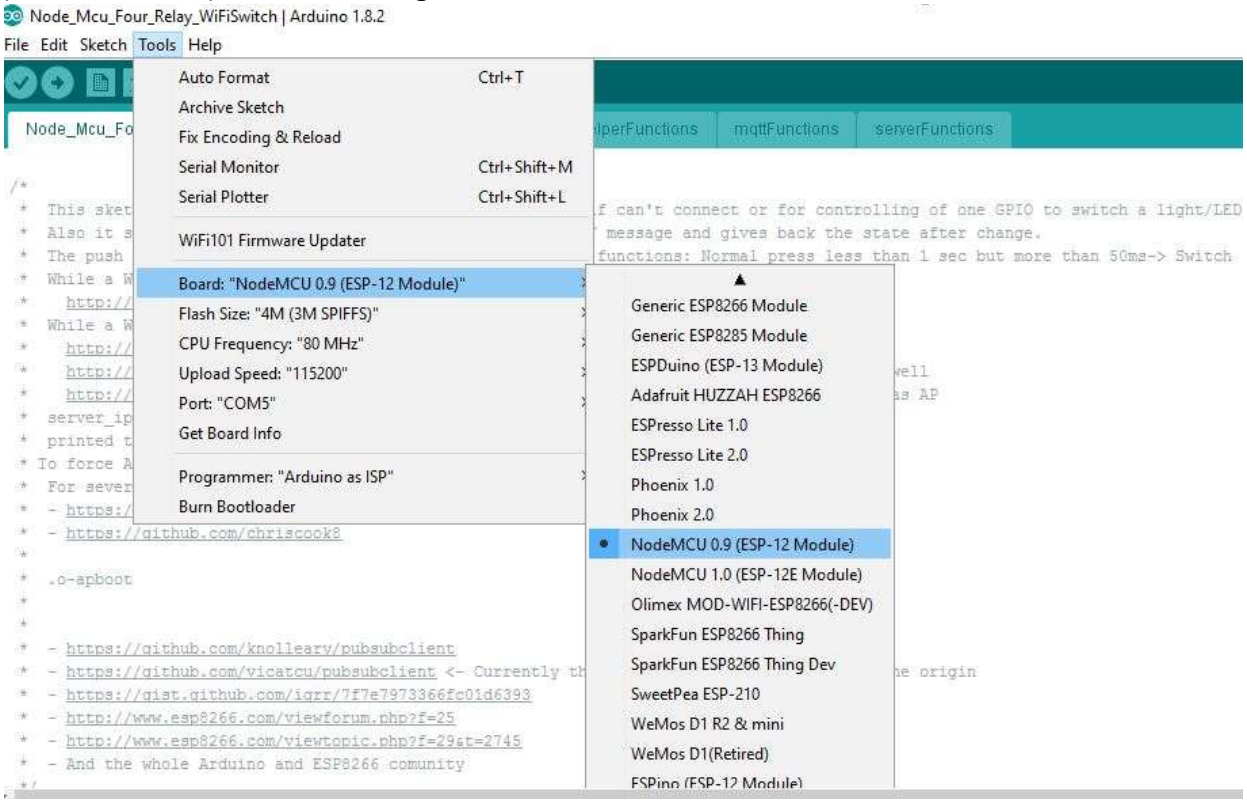

Figure 21: Board Selection

7. Select Upload Speed as “115200”.
8. Select flash size as “4M (SPIFFS)”
9. Select CPU Frequency as “80Mhz”.

10. Click on tools tab, move mouse pointer to “Programmer: “Arduino as ISP’”, under this click on “Arduino as ISP” to select. Refer to figure 22.

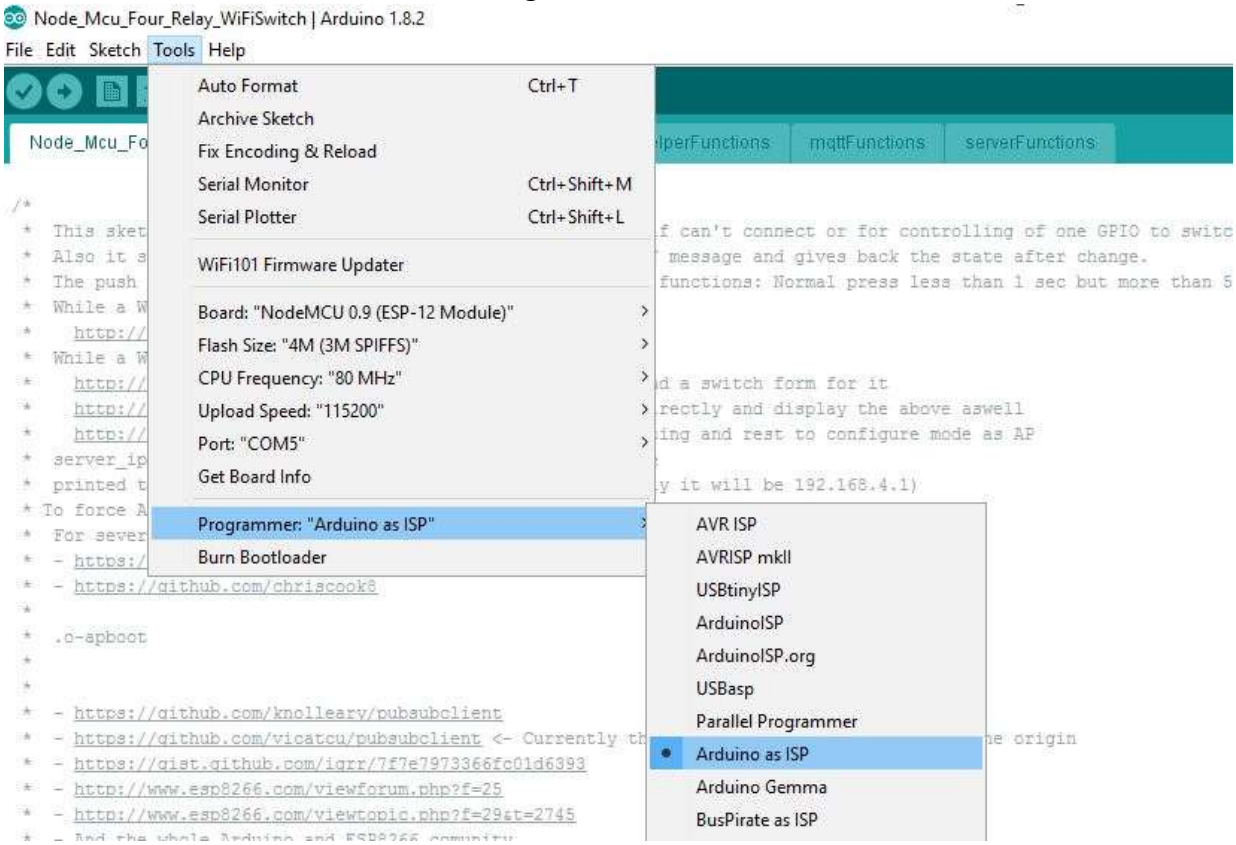

Figure 22: IDE Selection

11. Click on tools tab, move mouse pointer to “Port: “COMx”, under this click on “COMx” to select. (“x” refers to port number available in your computer) Refer to figure 23.

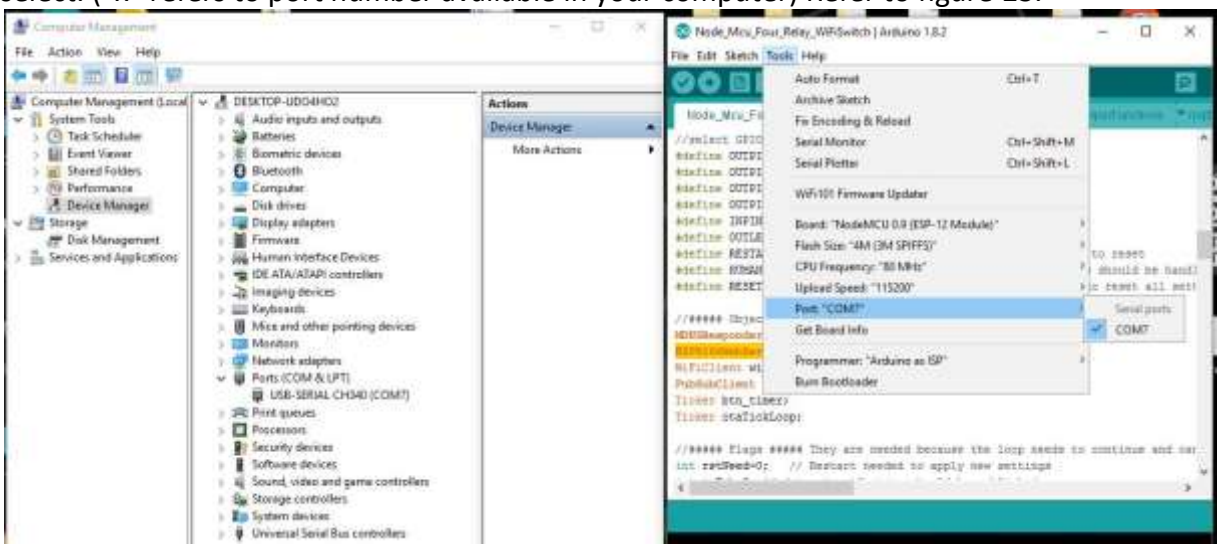

Figure 23: COM port selection.

12. Run the program. Refer to Figure 24.

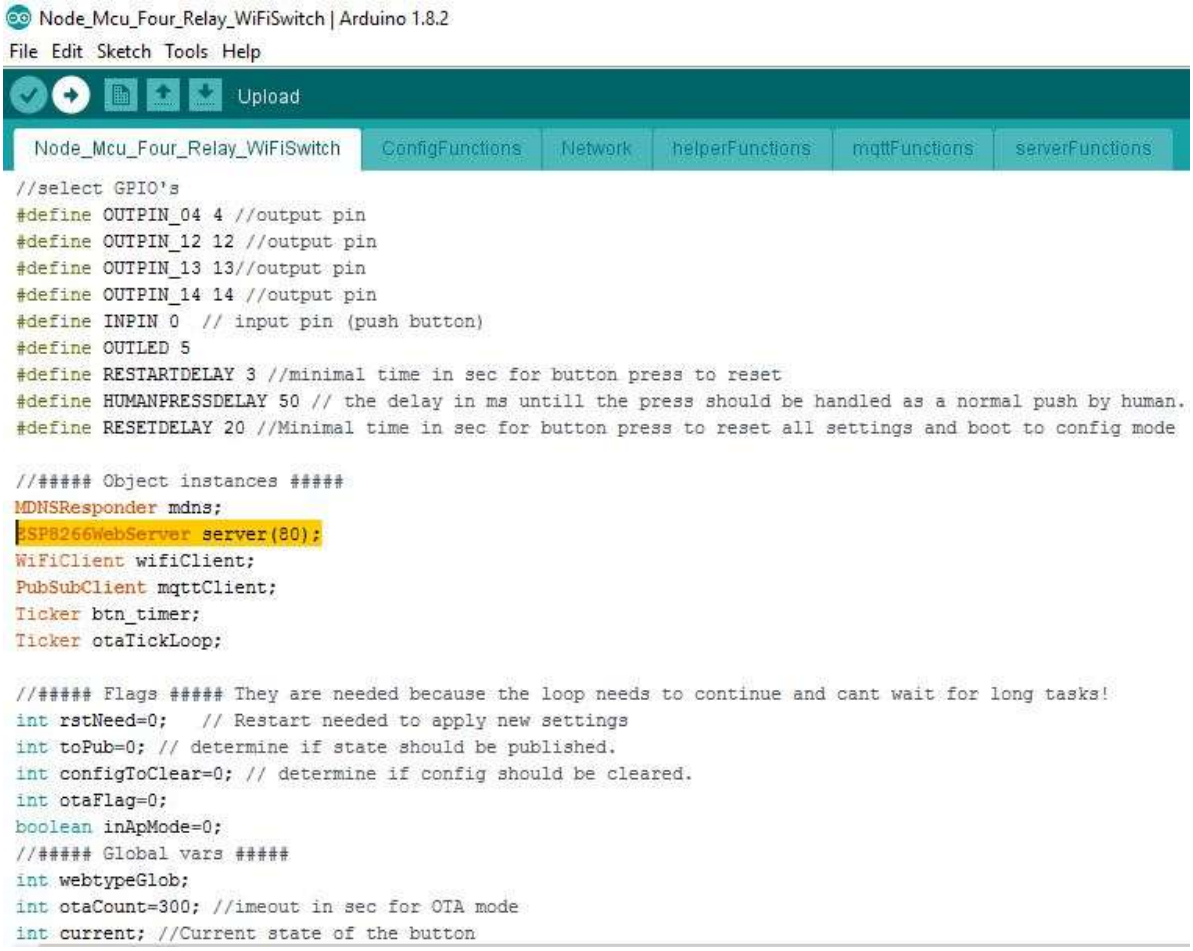

Figure 24: Executing code

b. STEPS TO LOAD PROGRAM TO ATMEGA328P:

1. Use external USB-UART converter between computer and this board.
2. Short Pin-1 and Pin-7 of bergstick-header P5 using removable jumper.
3. Short Pin-4 and Pin-10 of bergstick-s header P5 using removable jumper.
4. Connect Micro USB cable between your computer and U5 of “Wifi Eight Relay Board”.
5. Open your code in Arduino IDE as shown.
6. Click on Tools Tab, move mouse pointer on “Board: xxxxxxxxxx” and click on “Arduino Uno” as shown in figure 10.
7. Select Upload Speed as “115200”.
8. Click on tools tab, move mouse pointer to “Programmer: “Arduino as ISP’”, under this click on “Arduino as ISP”
9. Click on tools tab, move mouse pointer to “Port: “COMx”, under this click on “COMx” to select. (“x” refers to port number available in your computer).
10. Run the program.

Once your loading of program is completed, switch OFF the board and disconnect converter from it and short Pin-1.RXDA to Pin-2.TXDE, short Pin-3.RXDE to Pin-4.TXDA, to check the board functionality and use. If the board is working as per your code, then you can install it for your application.

13. INTEGRATION WITH "OPENHAB"

a. THINGS REQUIRED TO INSTALL OPENHAB ON RASPBERRY PI:

- Raspberry Pi 3 X 1
- SD Card 16 Gb X1
- 2A Power Supply 5V with USB data cable (micro)/USB Charger X1
- Raspberry Pi Case (Optional)
- SD Card Reader
- Monitor
- Keyboard
- Mouse
- HDMI to VGA Cable (optional in case Monitor is of VGA type
- Amazon echo/ Amazon echo dot

i. Important things to be downloaded:

- NOOBS from Raspberry Pi Website
- (https://www.raspberrypi.org/downloads/noobs/)
- SD Card Formatter (https://www.sdcard.org/downloads/formatter_4/eula_windows/index.html)
- You should have active internet connection to update the OS and install Java updated version
- Putty for SSH in case you are using windows (http://www.putty.org/)
- FileZilla (optional) in case you want to transfer some demo files directly to pi (https://filezilla-project.org/download.php)

ii. Installing Raspberry Pi OS and Updating it:

- First Step is to setup your Raspberry Pi. To do that, NOOBS should be extracted into a folder, example let's say folder "NOOBS".
- In this folder you will find around three folders (defaults, OS, overlays) and around fourteen files (Not mentioned here).
- Plug in your SD card reader with the 16GB SD card inserted to your computer USB. Format it using the SD card formatter (FAT32 format)
- Dump/Copy the contents of NOOBS folder into the formatted SD card(3 Folders + around 14 files)
- Eject the SD card reader, plug out the SD card and place it on your Raspberry Pi SD card holder
- Connect Power Supply/Charger, Monitor via HDMI /VGA(HDMI to VGA Cable required in case monitor is of VGA type), Keyboard, Mouse to the raspberry pi.
- Power on the Charger/Power supply
- Raspberry Pi takes you through the setup process and it is quite simple and intuitive.
- A few points to be considered while installing are, select Debian/Raspbian OS while installing, select Us keyboard /Language for most of the countries except UK and give access to the internet, you have to set up your SSID and Password.
- Once the installation is done please cross check that you are connected to the internet.
- If you are not connected to the internet then on the right-hand corner of the computer you will see an icon/Wifi icon (something like a computer icon) click it, it will show you various Wifi connections available.
- Pick the appropriate SSID. It will ask you to enter its Key (password).
- Once that is done open the terminal (greyish black color monitor image) found on left hand side mid corner.
- Type "**sudo apt-get update**"
- After the source list is updated on the same terminal type "**sudo apt-get upgrade**"
- This will take some time but will update your raspberry pi OS
- This will setup your Raspberry Pi
- Now we have few things to install and will be discussed in the next step.

iii. Installing/Updating required sotware/drivers:

a) To update Java please follow the instructions mentioned below:

- Check Java version by typing “java -version” in the terminal
- If your java version is lesser than "1.8.0_101" please follow the below commands

- First remove openjdk
- sudo apt-get purge openjdk*
- add packet source
- sudo nano /etc/apt/sources.list
- add the following lines at the end of the sources.list file
- deb http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main
- deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu trusty main

b) Install Java 8

- sudo apt-get update sudo apt-get install oracle-java8-installer sudo apt-get install oracle-java8-set-default
- remove old Java
- sudo apt-get purge openjdk*
- sudo apt-get purge java7*
- sudo apt-get autoremove
- check success of upgrade by typing
- java -version
- Check if it is above "1.8.0_101"

c) Install Mosquitto Mqtt Broker by using the following command

- sudo apt-get install mosquitto
- To install mqtt client use
- sudo apt-get install mosquitto_client

d) Activating SSH if it is not Open in your Raspberry pi’s terminal Type

- sudo raspi-config
- Go to Interfacing Options and press enter
- There you will find P2 SSH Enable/Disable SSH
- Select that by using up/down arrow key and press enter
- We are basically enabling the ssh option for further use.

e) Installing OpenHab

We can now continue to install OpenHab on raspberry pi. To do this, open terminal again and type the following commands

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install screen mc vim git htop
- First, add the openHAB 2 Bintray repository key to your package manager and allow Apt to use the HTTPS Protocol:
- Wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?
- username=openhab' | sudo apt-key add - sudo apt-get install apt-transport-https
- echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' |
- sudo tee/etc/apt/sources.list.d/openhab2.list
- sudo apt-get update

f) Now install OpenHAB with the following commands

- sudo apt-get install openhab2
- When you choose to install an add-on, openHAB will download it from the internet on request. If you plan to disconnect your machine from the internet, then you have to install the add-ons package.
- sudo apt-get install openhab2-addons
- If everything goes well, you can start openHAB and register, it will be automatically executed at system startup.
- sudo systemctl start openhab2.service sudo systemctl status openhab2.service
- sudo systemctl daemon-reload
- sudo systemctl enable openhab2.service
- The first start may take up to 15 minutes, you should be able to reach the openHAB 2 Dashboard at http://your_raspberry_pi_ip:8080 at this point from any computer in the same network, any browser.
- Once you open it, click on the Paper UI and then go to addons.
- Inside addons click on bindings.
- Here go the Http Binding and install it (here instead of mqtt we are configuring armtronix boards under Http mode).

After installing the http binding go to the tab MISC and install openHAB Cloud Connector, this is required for internet access and to interface it with Alexa.

You can also go to the USER interface tab and install the Basic/Classic UI to control your appliances after integration.

b. STEPS EXPLAINED ON, HOW TO USE OPENHAB IN WINDOWS

Step 1: Open Putty Configuration application and enter the IP address 193.168.1.33 as shown in the figure below and make sure to select the connection type as SSH and click on OPEN.Putty for SSH in case you are using windows (http://www.putty.org/)

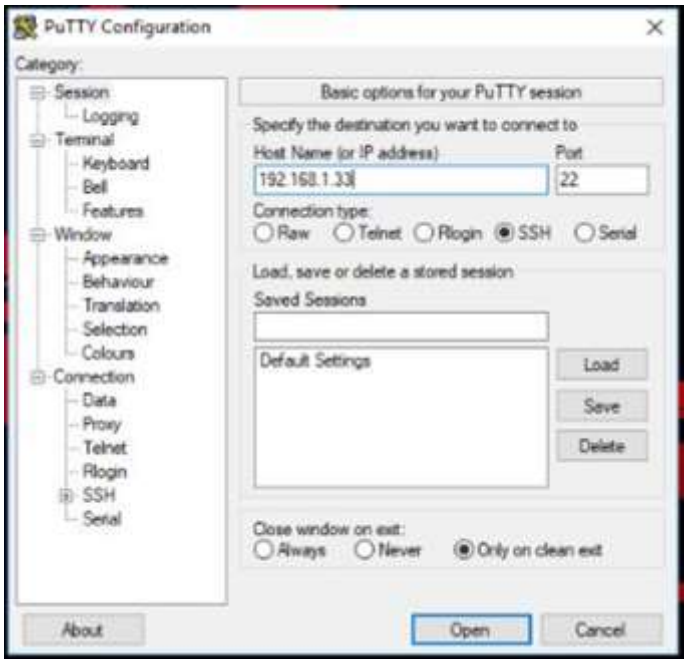

Figure 25: Putty application for Raspberry Pi

Step 2: Login as Pi and enter the password as raspberry.
Login as : pi

root@192.168.1.33's
Password : raspberry

Step 3: Type the command as etc/openhab2 on the Terminal (greyish black color monitor) to enter into Openhab2 file as shown in the figure below.

Figure 26: Path for the file

Step 4: cd /etc/openhab2/sitemaps/ You will be in the site map folder you need to create a file with extension as .sitemap, To do that
Type nano
Amtronix_Office.sitemap Example: "Armtronix_Office.sitemap" is the name of that file . Once you press enter, it will give you a blank file for editing. Site map is basically a layout.

```
Group item=w10 label="Single relay" icon="group"
{
        Switch item=w101 label="Relay"
}
```

Step 7: Next thing is to create the file with **.item** as an extension. To do this change to your item directory by typing **cd /etc/openhab2/items**. For http mode, the URL http://(**IP address of the board**)/ay?0=1, (IP address is different for different boards) for toggling (**ON** / **OFF**).

Example: http://192.168.1.22/ay?0=1

Figure 27: OpenHAB sub folders, Accessing item folder

Group w101

Switch w101 "Relay"(w10,Lights)
{http=">[ON:POST:http://192.168.1.22/ay?o=1] >[OFF:POST:http://192.168.1.22/ay?o=1]"}

Step 8: Code for Item file in MQTT mode, where 'sub' indicates the subscribe topic which is mentioned during the configuration.
Switch sr "Relay"(w10)
{mqtt">[broker:test_sub/test/:command:ON:D2_ON],>[broker:test_sub/test/:command:OFF:D2_OFF],>[broker:/w10:state:MAP(w10d2.map)]",autoupdate="false"}

Step 9: You can Register yourself in OpenHAB by typing the URL https://myopenhab.org, if you have already registered then you can enter the E-mail address and password and then click on **Sign in**, as shown in the figure below.

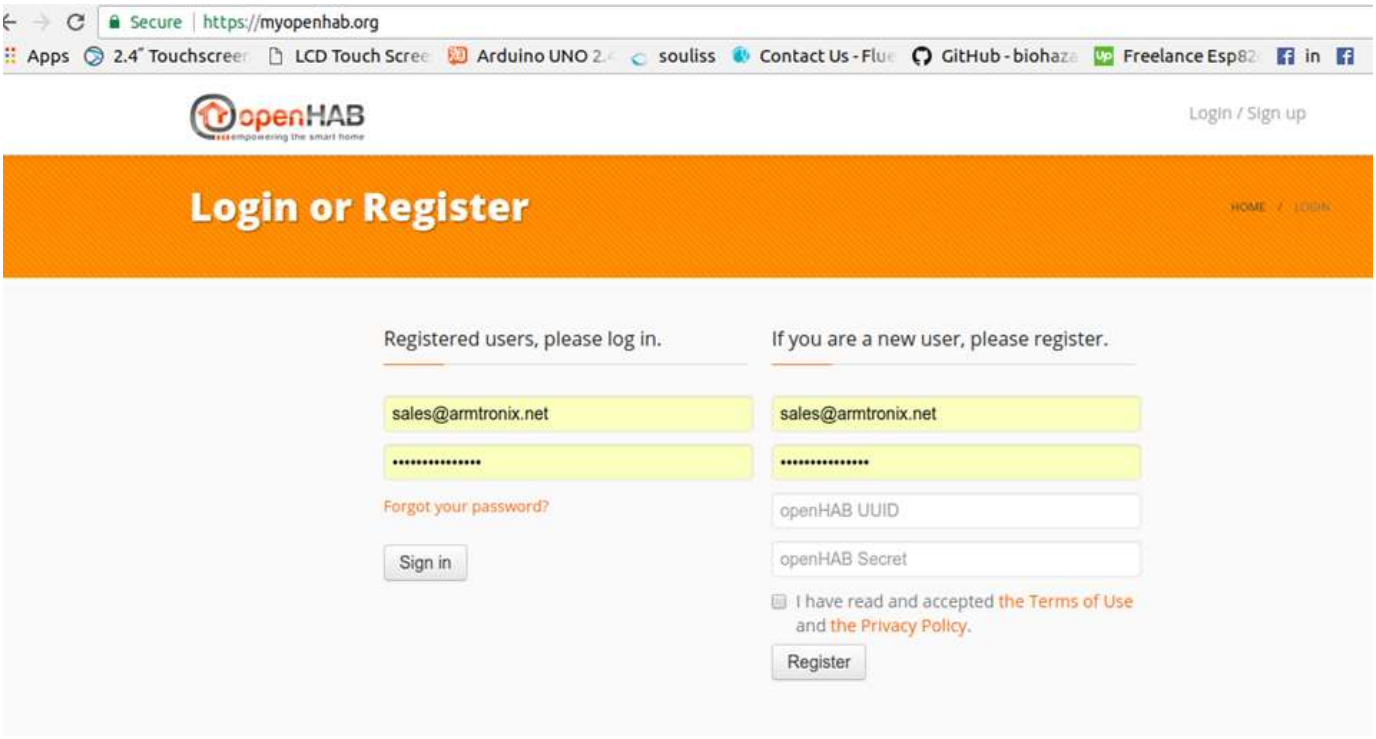

Figure 28: Login page of OpenHab

Step 10: Once you register yourself with the OpenHab you will be taken to the following page as shown in the figure below.

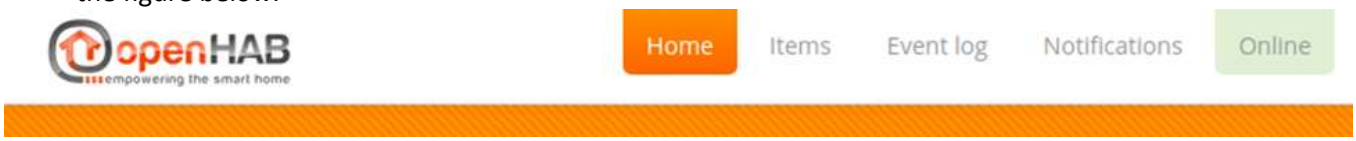

Figure 29: Menu bard of OpenHab

Step 11: Enter the IP address for which your Board is configured, then you will be taken to the following page as shown in the figure below.
Example: 192.168.1.11

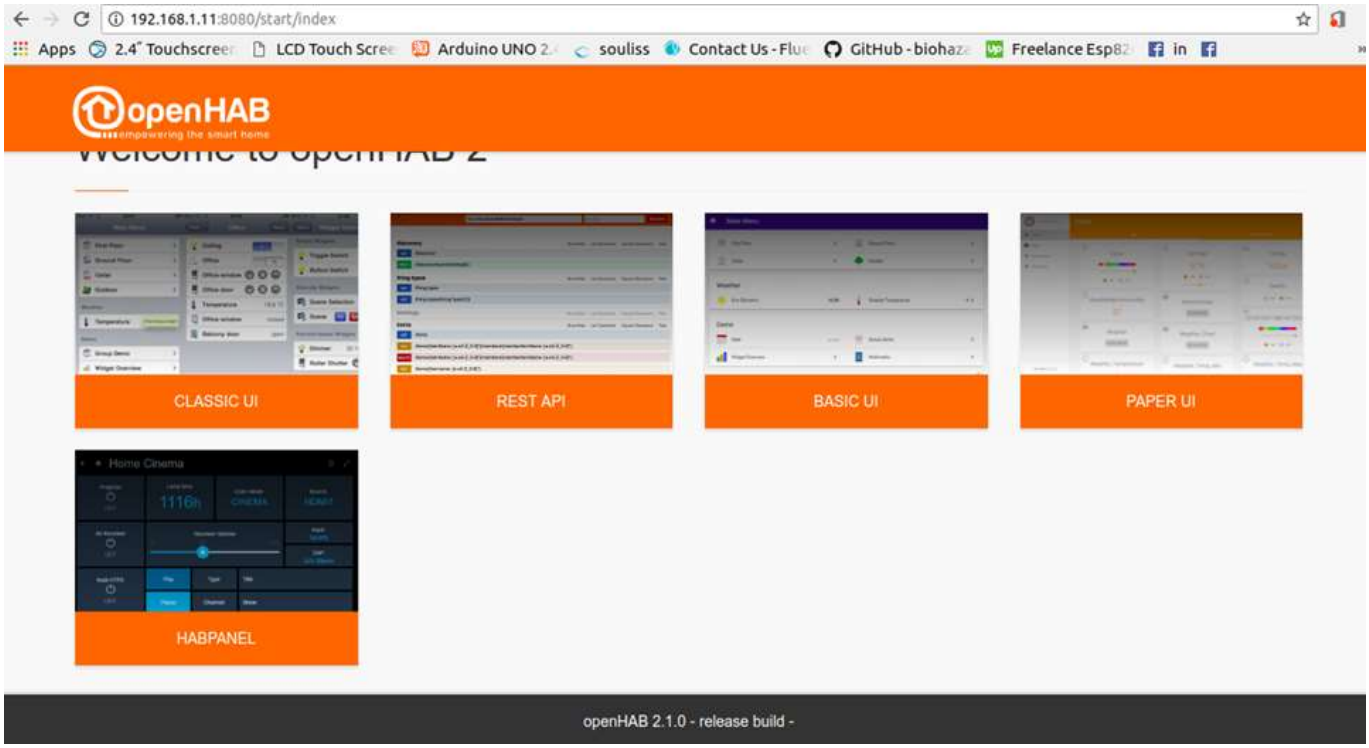

Figure 30: Home page of device with entered IP

Step 12: After creating the Sitemap file in **.sitemap** extension and Item file in **.item** extension as mentioned in the previous steps, the external appearance in OpenHab application is as shown in the figure below.
Example: ARMtronix Office.

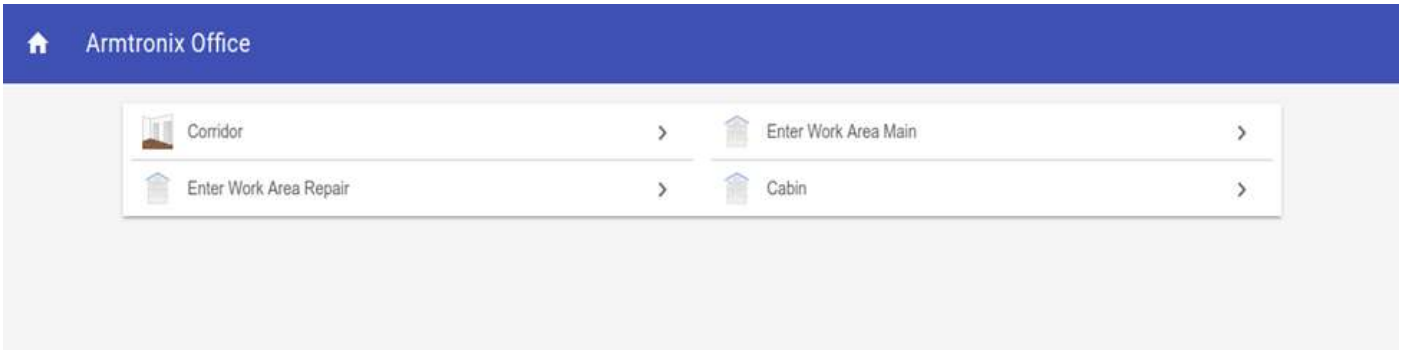

Figure 31: Devices are categorized as Groups under frame

Step 13: In OpenHab application you will be provided with the options such as Control, Inbox, Configuration, Add-on and Preferences. If you click on Add-on option then you will be provided with many options as shown in the figure below.

Example: Add-on option in that Binding option and you can select any binding of your choice and install it (basically ARMtronix boards are configured for HTTP mode), which is suitable for your application.

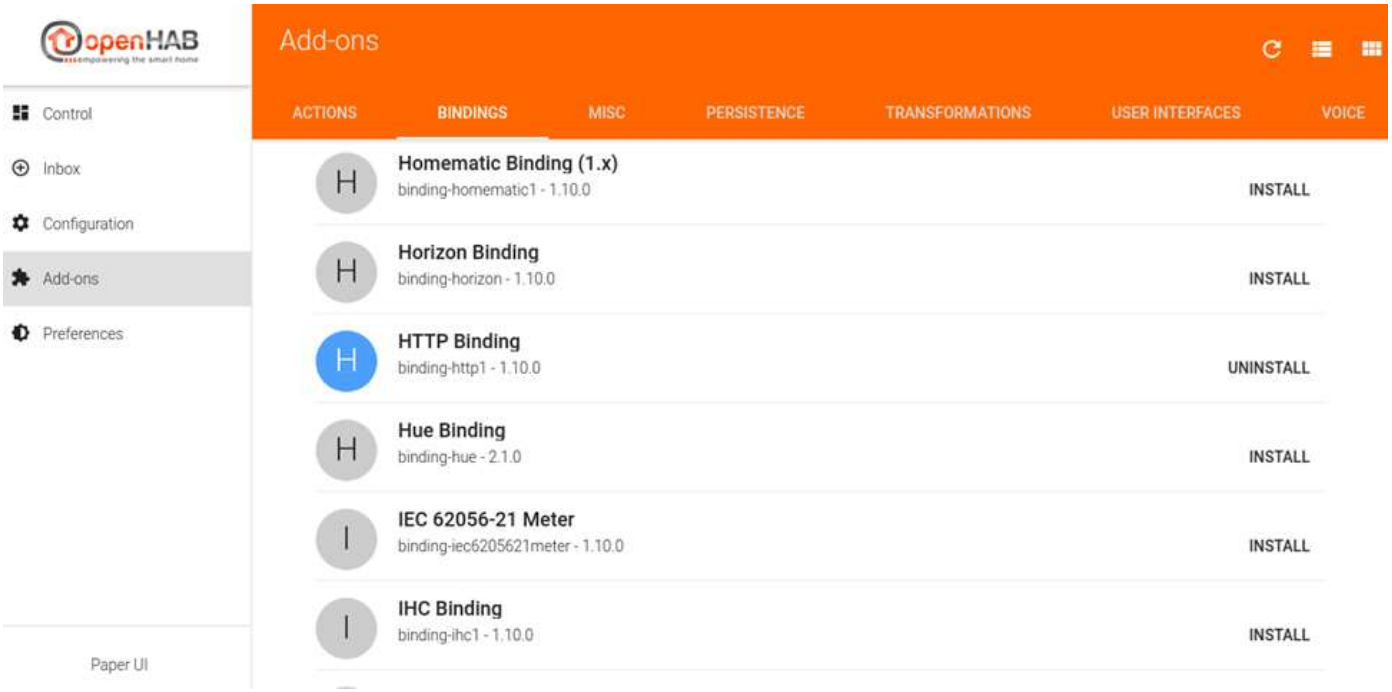

Figure 32: OpenHAB Add-ons binding option selection

14. FAQ ON OPENHAB

i. How To Configure OpenHab Application?

Consider the following steps to configure your OpenHab application.

Step 1: Make sure that your Smartphone is connected to the OpenHab cloud connector, if not then click on Add-on option in that you have to select the MISC option under that check whether OpenHab cloud connector is installed first, if not install it (It is required basically to interface with Alexa).

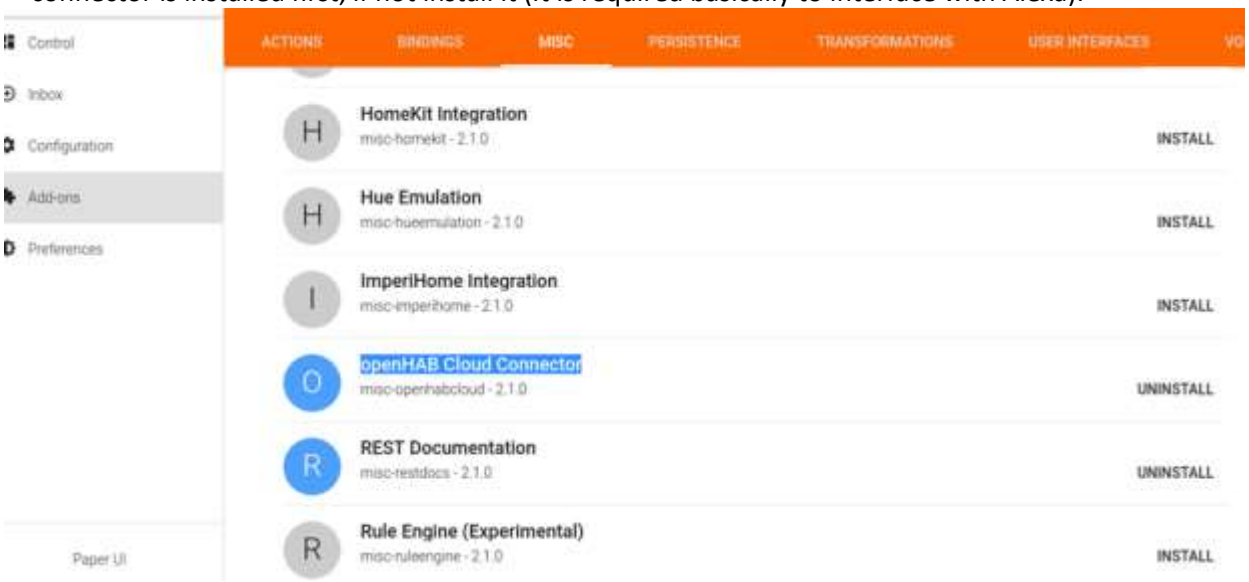

Figure 33: Binding option selected as OpenHAB

Step 2: In OpenHab application there is Configuration option under that select the option Services then a dialog box appears where you are able to see Configure OpenHab cloud and you can check the Mode, for that you will be provided with the dropdown where you can select the suitable Mode for your application, Base URL for OpenHab cloud server and items to expose to apps and you can select by clicking on the checkboxes provided in front of the options and you can save it by clicking on the SAVE option, as shown in the figure below.

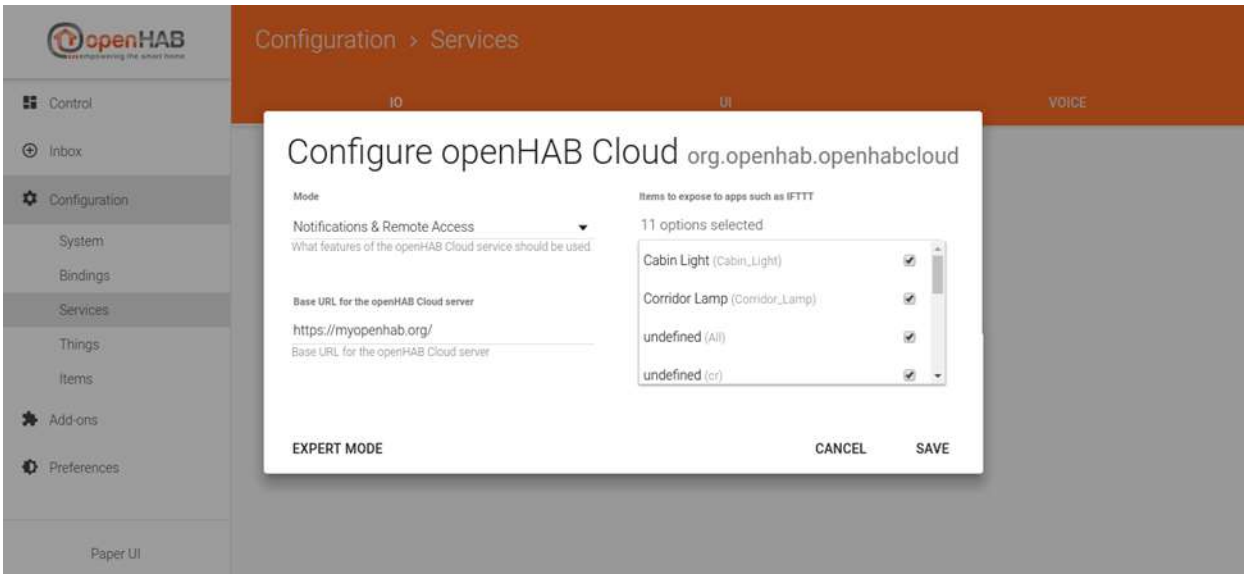

Figure 34: Device selection to configure with OpenHAB

15. INTEGRATION WITH "AMAZON ALEXA"

a. STEPS TO INSTALL ALEXA APP IN SMARTPHONE

- Download the Amazon Alexa app from the Google Play Store.
- Open it and sign in to your Amazon account.
- Open Settings on your Android phone
- Open Apps
- Tap "Default Apps"
- Select "Assist & voice input"
- Choose Alexa instead of Google Assistant.

After you've followed the steps above, you'll see a prompt to talk to Alexa. It'll ask for permissions, so tap "Allow" to let Alexa hear your requests and control devices in your smart home.

Now you'll be able to hold the home button to pull up Alexa. You can use Alexa to turn on smart lights at home ("Alexa, turn on the living room lights," for example)

Following steps explain how to use Amazon Alexa through your Smartphone
Step 1: Click on Amazon Alexa app installed in your Smartphone.

Figure 35: Alexa app bootup screen

Step 2: Click on the  icon which is present on the right-bottom corner of your Smartphone as

shown in the figure below.

Figure 36: Alexa app home screen

Step 3: Click on **All Devices** option to check which devices are connected to the Amazon Alexa as shown in the figure below.

Figure 37: Check # of devices connected

Step 4: You can make groups of your choice and it shows how many devices are connected to that particular group as shown in the figure below,

Example: Cabin, Lab, Entrance, Office etc.

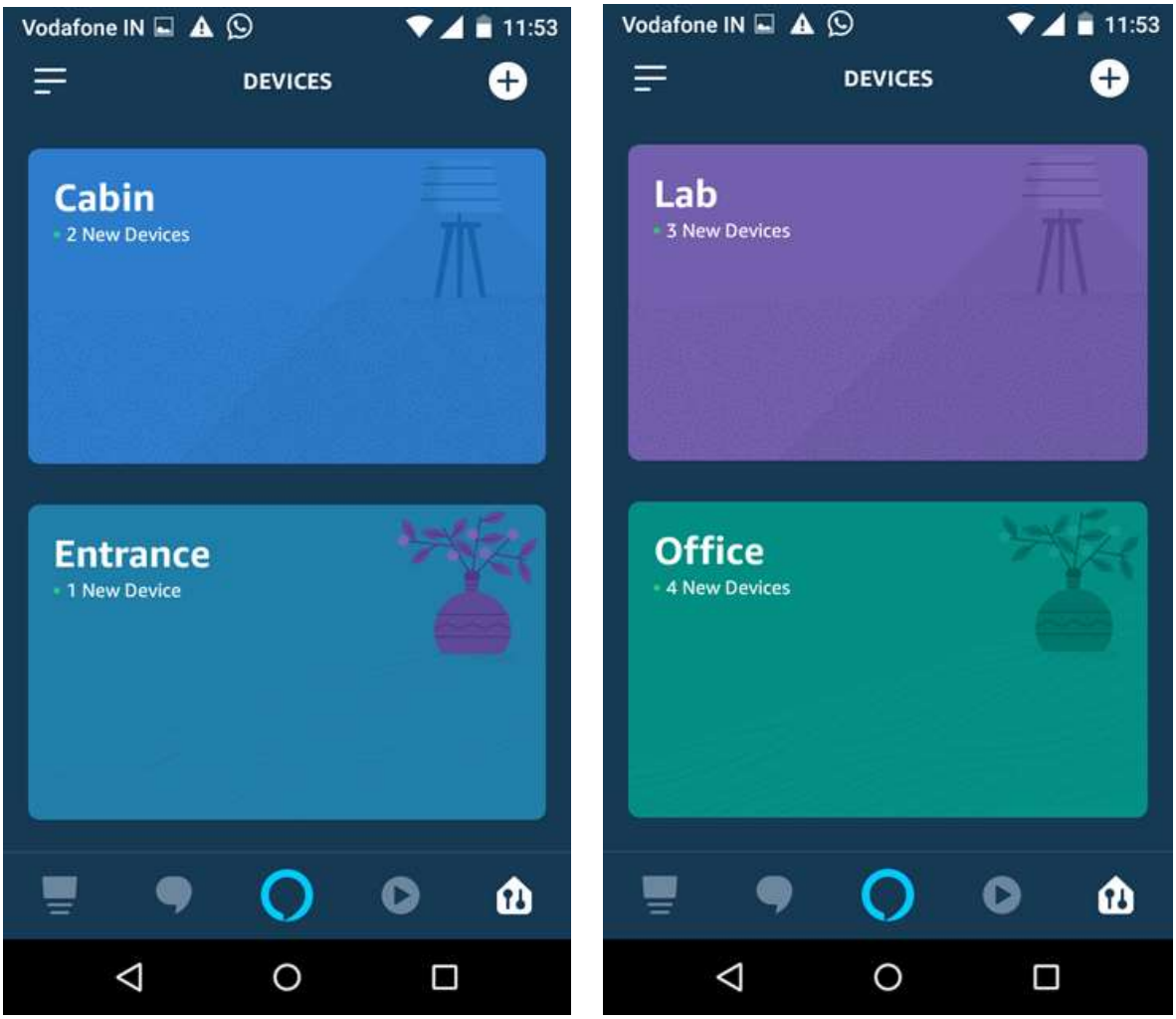

Figure 38: Devices devided as groups in Alexa app

Step 5: Upon voice recognition by Alexa you can control Loads (Fan, Lights) by giving commands.

Example: **Alexa turn ON Cabin Lights**
Alexa turn OFF Cabin Lights

Alexa turn ON Office Lights
Alexa turn OFF Office Lights

Note: Ensure that to interface with Alexa you are connected to the ”OpenHAB Cloud Connector”.

Step 6: Ensure that when you click on All option under groups, it displays the device name which you have mentioned during the configuration time as shown in the figure below.

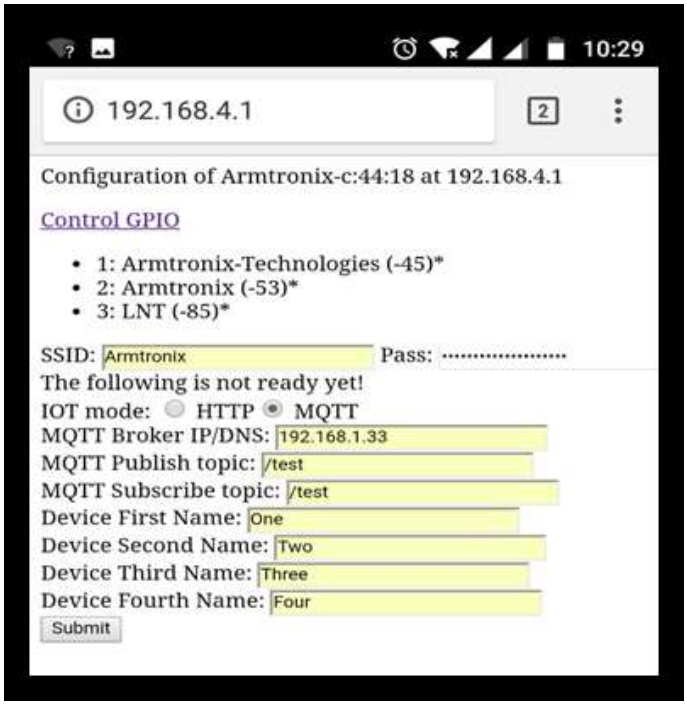

Figure 39: Names are defined for loads

16. INTEGRATION WITH "GOOGLE HOME"

a. FOLLOWING STEPS EXPLAIN, HOW TO INTEGRATE GOOGLE HOME WITH OPENHAB

To use the OpenHab integration for Google Assistant on your smartphone, you will need the Google Assistant or Google Home app (iOS or Android)
Before you start integration, make sure that the OpenHAB is installed and configured with our devices.

Step 1: Make sure Google Play Services is up to date

- Visit "Google Assistant" app entry in Google Play Store on Android
- Set up the voice-activated speaker, Pixel, or Android phone (version 6+) with the *same test account*
- Make sure you're the correct user
- Start the updated Google Assistant app on your phone

Step 2: Open Google Assistant app in your Smartphone as shown in the figure below

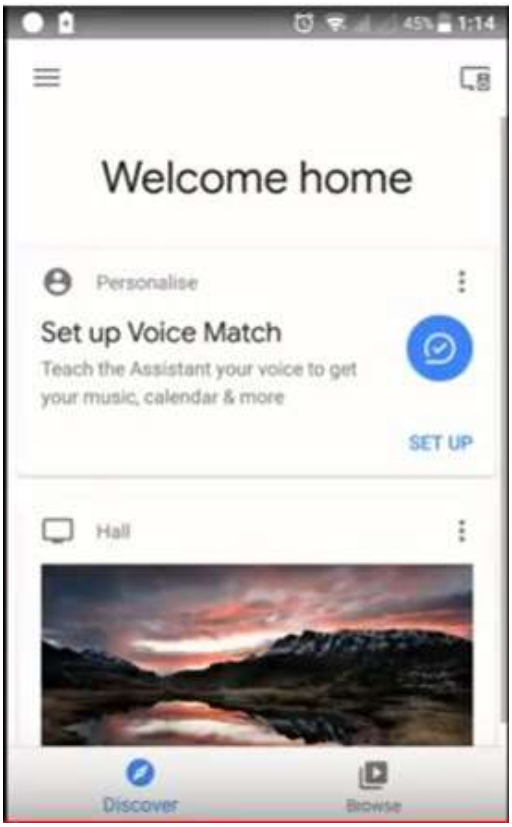

Figure 40: Google Home app's Home screen

Step 3: Go to Settings, under that go to services there you will be able to see the option Home control, click it.

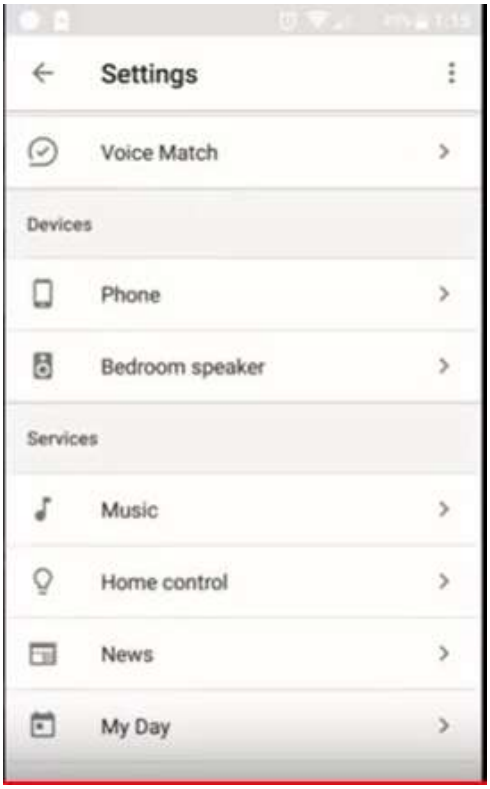

Figure 41: Google Home app's Settings screen

Step 4: Click on the "+" sign under Home control which will take you to the Add devices

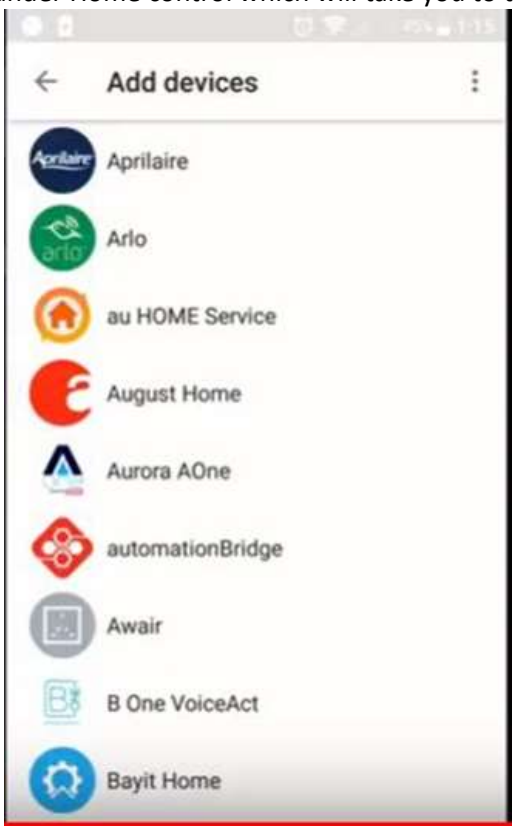

Figure 42: Google Home app's screen to Add IoT devices

Step 5: Under Add devices select the option OpenHab as shown in the figure below
Note: Ensure that your Smartphone is connected to the OpenHab cloud connector

Figure 43: Screen to Add cloud service provider

Step 6: Click on OpenHab and enter the E-mail ID and Password.

Figure 44: Login Screen

Step 7: Click on Allow button as it is asking permission for Authentication with Google home.

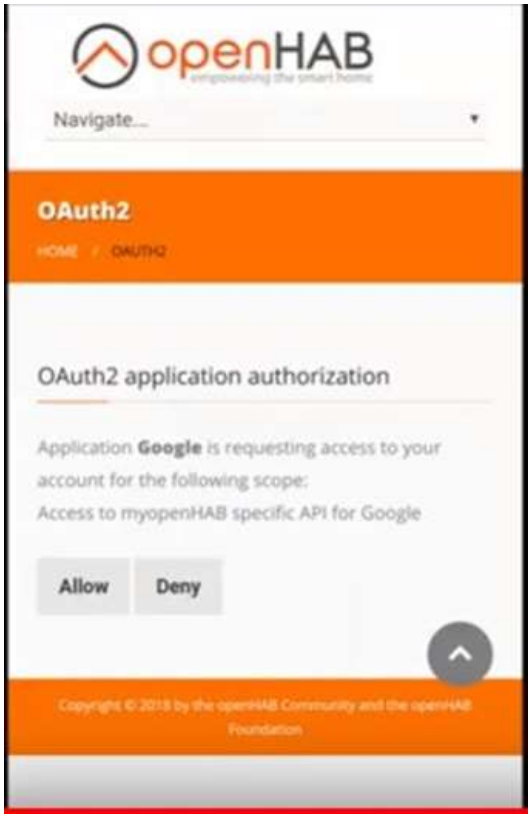

Figure 45: Apps's permission athentication screen

Step 8: It will display the devices connected with the OpenHab as shown in the figure below.

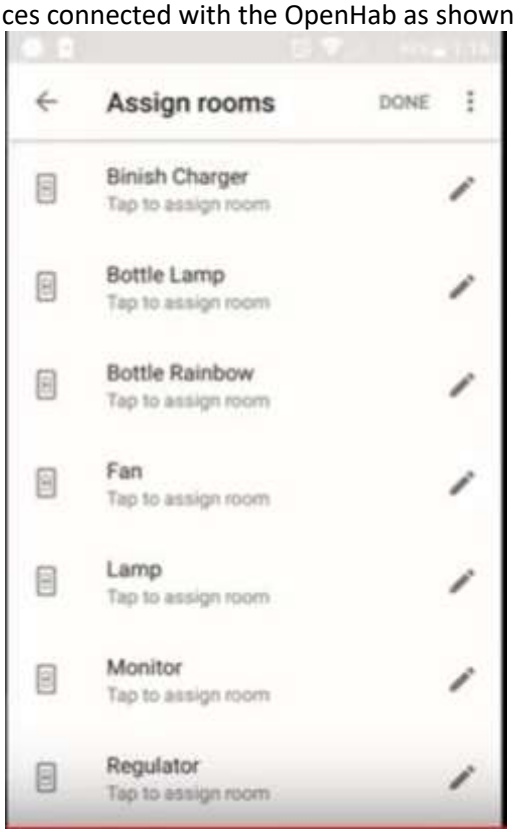

Figure 46: List of added devices

Step 9: Now it is ready to give commands and upon voice recognition Google home respond to the commands given.

Example: **Google turn on fan**
Google turn off fan

Google turn on Lights
Google turn off Lights

Figure 47: Device control screen

IMPORTANT NOTICE

ARMtronix Technologies LLP and its subsidiaries reserve the right to make corrections, enhancements, improvements and other changes to its products, services and documents and also to discontinue any product or service. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to ARMtronix Technologies LLPs terms and conditions of sale supplied at the time of order acknowledgment.

The information in this document is subject to update without notice. The contents of this document thereof must not be used for any unauthorized purpose.

----------END OF DOCUMENT----------

