DESCRIPTION DOCUMENT FOR WIFI/BT QUAD RELAY BOARD

HARDWARE REVISION 0.1

Department	Name	Signature	Date
Author			
Reviewer			
Approver			

Revision History

Rev	Description of Change	Effective Date
A	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.

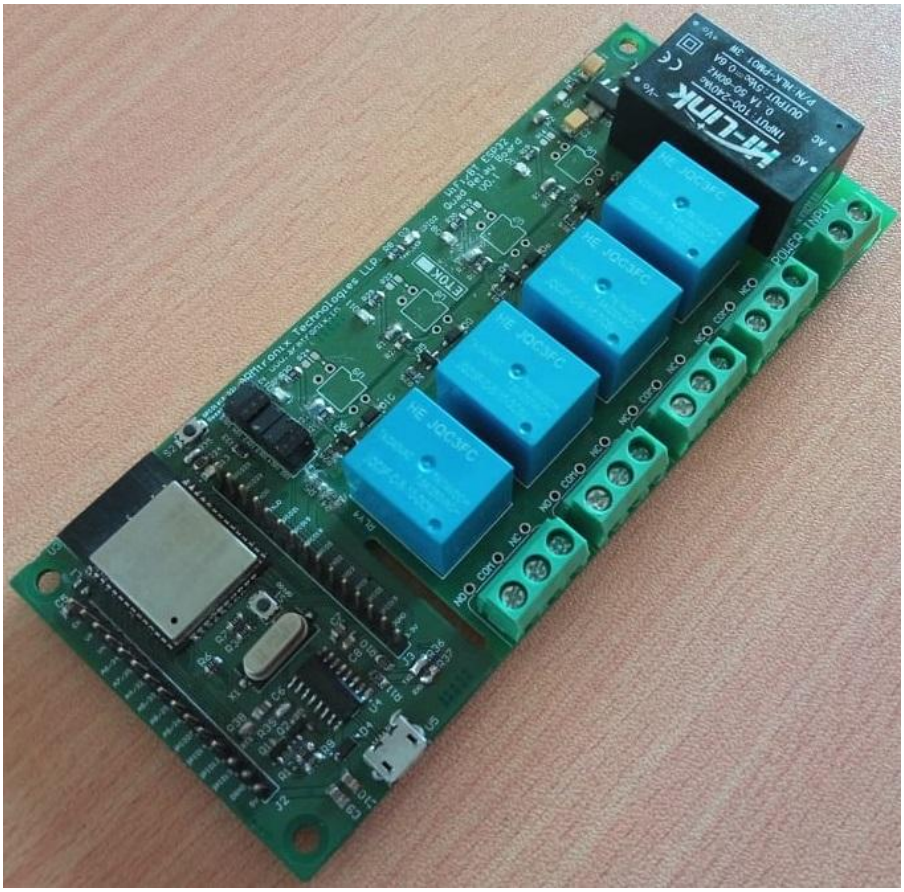


Table of Contents

Table of figures

1. ABBREVIATIONS

Term	Description
AC	Alternating Current
AP	Access Point
BT	Bluetooth
COM	Common
DC	Direct Current
HTTP	Hypertext Transfer Protocol
Hz	Hertz
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
Github Weblink	https://github.com/armtronix/Wifi_Bt_Esp32_Quad_Relay
Youtube Weblink	https://www.youtube.com/watch?v=wqkkvoWiCZI

3. PURPOSE

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

4. SCOPE

This document describes system architecture which includes Power supply, relay, WiFi/BT ESP32 Module 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 human 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 page and github. Please refer it.

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 requirement.
- Four Dry contact relay output with COM, NO and NC 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
- Basic Firmware to enter SSID and password to connect to the router
- Firmware has ability to control device through HTTP and MQTT mode.
- Push Button on board provided for configuration and Reset function.
- Board is compatible and configurable to Amazon Alexa.

7. PRODUCT DESCRIPTION

a. PHYSICAL DESCRIPTION

- AC to DC Power supply module
- Mechanical Relay – 4 numbers
- Wifi Module
- USB-UART converter

b. FUNCTIONAL DESCRIPTION

Block Diagram

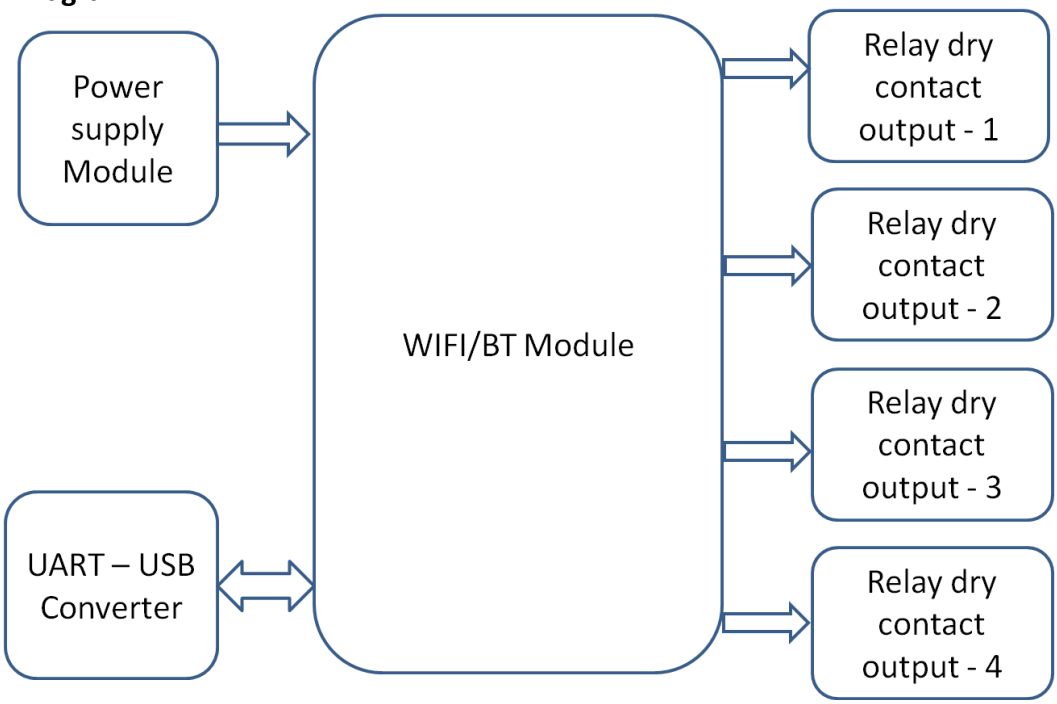


Figure 1: Block Diagram

Wifi/BT relay board is ESP32 based relay board, has on-board power supply module which takes standard AC power as input and provides required DC power as output. The DC power is used to power-up Wifi/BT module used on board to establish Wifi communication with mobile phones or wifi routers or access points. There are four relays mounted on board to control (ON/OFF) four external electrical loads independently from a mobile application using MQTT/HTTP protocol.

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-PM01. This power supply module rectifies and regulates voltage from 230 V AC to 5 V DC with output current capacity of 0.6A DC. The power of HLK-PM01 is at maximum of 3W.

The 5V supply is used to power on relay and USB-UATT 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/BT Module

Wifi module used on the board is ESP32 with all its required GPIOs are easily accessible to user for their own application. Wifi module is powered through 3.3 V DC. It works on both MQTT / HTTP protocol.

3. Mechanical Relay – 4 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. A driver circuit with an opto-isolator is used to drive the relay.

4. 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 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 Four relay board for programming purpose.

9. TECHNICAL SPECIFICATION

a. ELECTRICAL SPECIFICATION

Input Specifications				
Description	**Min**	**Typ**	**Max**	**Unit**
Voltage AC	100	220	230	Volts
Current AC	-	0.1	-	Amps
Power AC	-	3	-	Watts
Frequency	50	-	60	Hz

Relays Output Specifications (Maximum)				
Description	**Min**	**Typ**	**Max**	**Unit**
Voltage AC	-	-	240	Volts
Current AC	-	-	3	Amps
Power AC	-	-	980	Watts
Voltage DC	-	-	24	Volts
Current DC	-	-	3	Amps

b. MECHANICAL SPECIFICATION

- Mechanical Dimensions of PCB are 140 x 60 x 20 mm (Length x Width x Height)
- Mounting Holes are compatible with M3 screws pan head dia maximum of 5.65mm.

10. ELECTRICAL CONNECTIONS

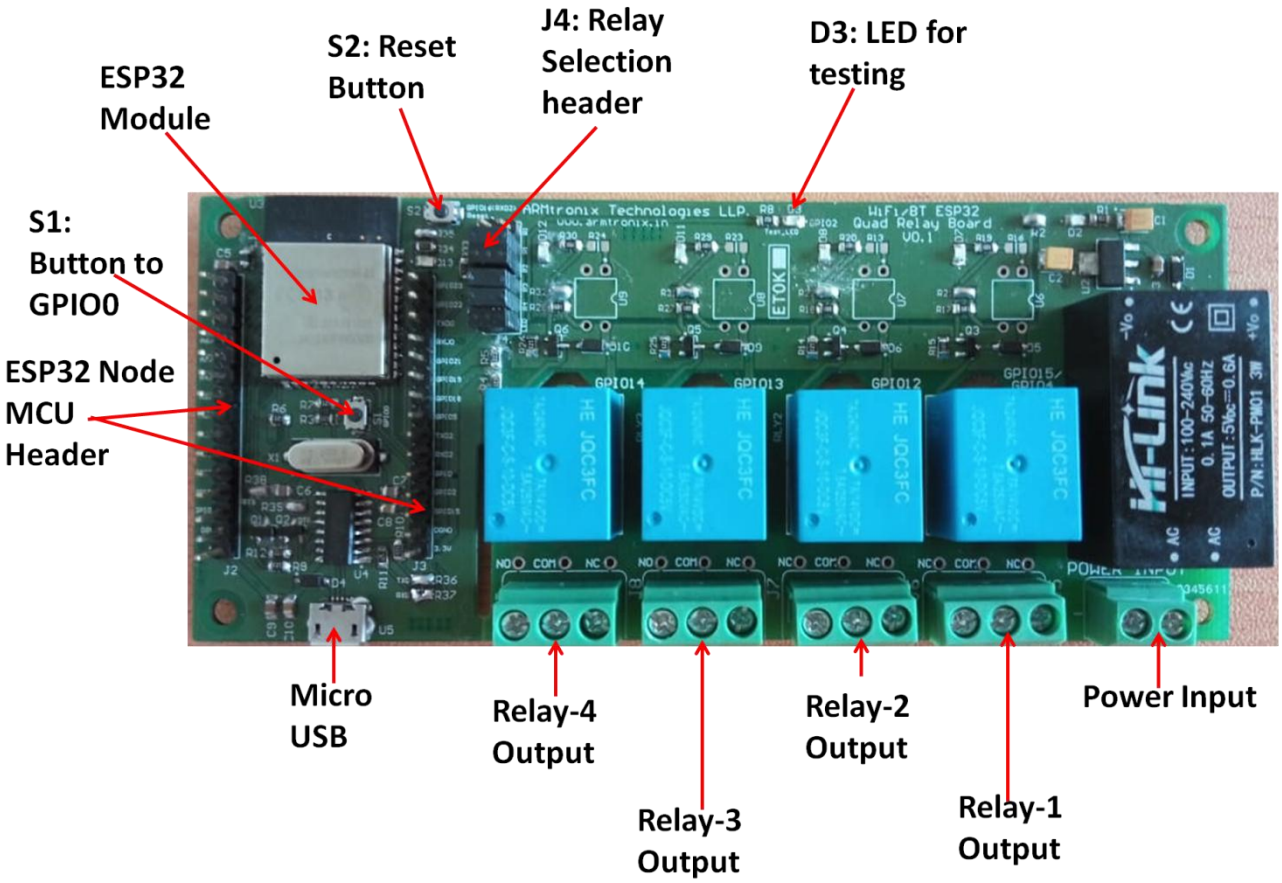


Figure 2: Header and Switch Details

Description of Header and Switches shown in Figure 1:

1. S1 Button to GPIO_0.
2. S2 Button to reset the ESP.
3. Power Input AC input terminal block.
4. J4 Relay Selection Header.
5. J2 and J3 Headers are compatible to standard ESP32 extra GPIO headers.
6. U5 Micro USB for programming.

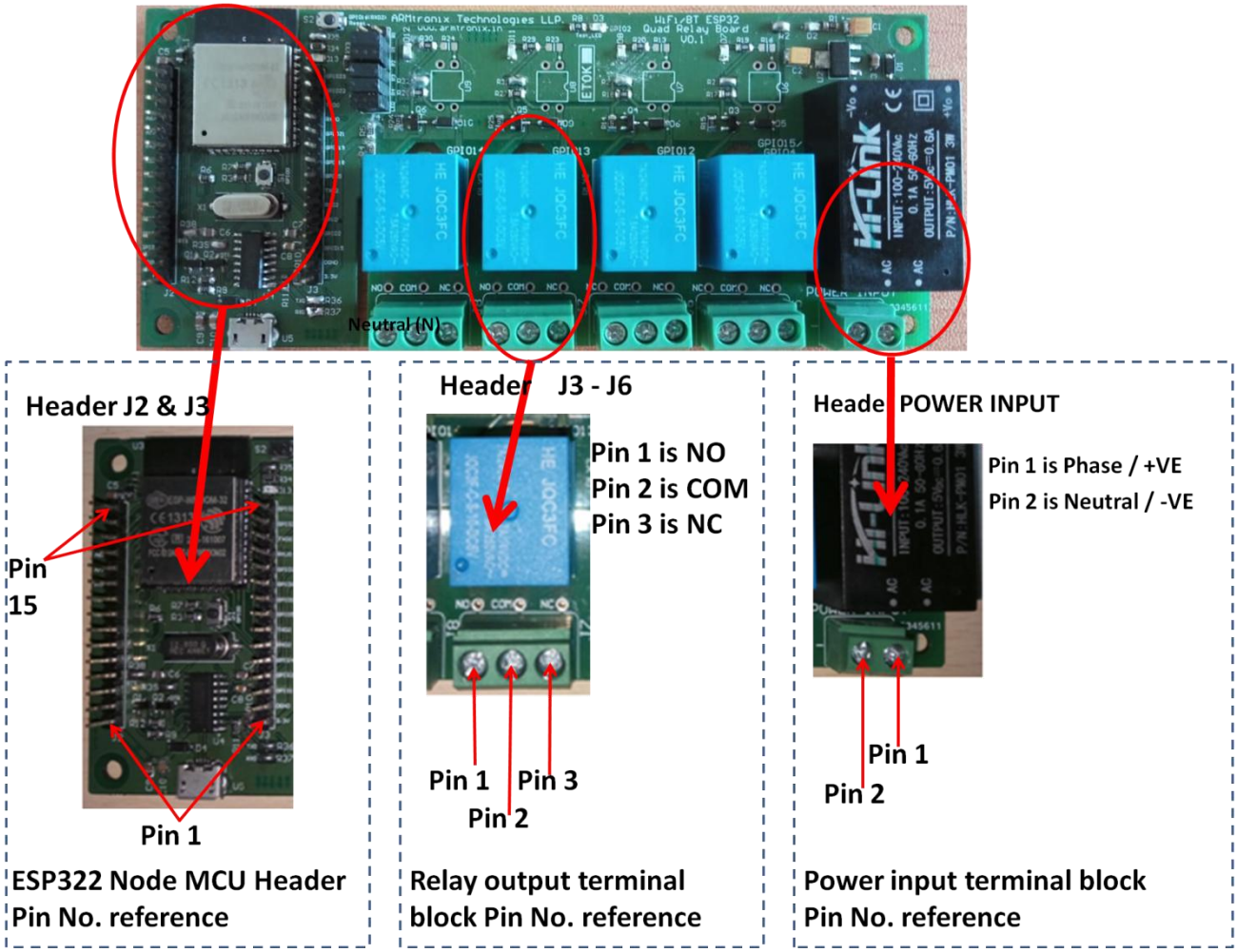


Figure 3: Header Pin number references

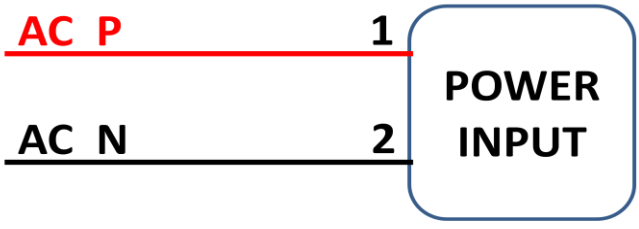


Figure 4: AC Input connection

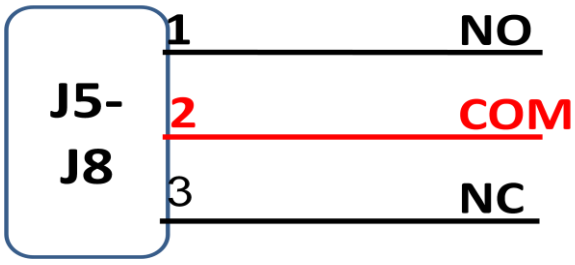


Figure 5: Relay Dry contact pin-out

Figure 3, shows pinout and connection of AC Phase and Neutral connection to POWER INPUT connector. Figure 4, shows J5 – J8 are output load connector.

a. HEADER PIN CONFIGURATION

i. HEADER J2

Header Pin	ESP Pin Number	Pin Name	Arduino Pin Name /
15	3	EN	
14	4	VP	
13	5	VN	
12	6	A6/GPIO_34	34
11	7	A7/GPIO_35	35
10	8	A4/GPIO_32	32
9	9	A5/GPIO_33	33
8	10	A8/GPIO_25	25
7	11	A9/GPIO_26	26
6	12	GPIO_27	27
5	13	GPIO_14	14
4	14	GPIO_12	12
3	16	GPIO_13	13
2	15	DGND	-
1	-	5V	-

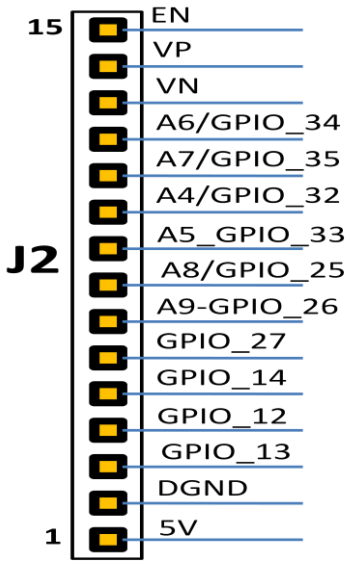


Table 1: Header J2 Pin Configuration

ii. HEADER J3

Header	ESP Pin	Pin Name	Arduino
15	37	GPIO_23	23
14	36	GPIO_22	22
13	35	TXD0	-
12	34	RXD0	-
11	33	GPIO_21	21
10	31	GPIO_19	19
9	30	GPIO_18	18
8	29	GPIO_05	05
7	28	TXD2	17
6	27	RXD2	16
5	26	GPIO_04	04
4	25	GPIO_02	02
3	23	GPIO_15	15
2	-	DGND	-
1	-	VCC_3V3	-

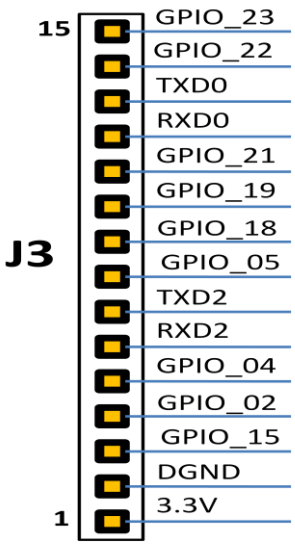


Table 2: Header J3 Pin configuration

Table 1 and 2, shows the header J2 and J3 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**

Header Pin Number	Pin Name	Header Pin Number	Pin Name
1	GPIO15	2	Relay_01
3	GPIO4	4	Relay_01
5	GPIO12	6	Relay_02
7	GPIO13	8	Relay_03
9	GPIO14	10	Relay_04
11	GPIO2	12	Test_LED (D3)

Table 3: Header J4 Pin Configuration

GPIOs mentioned in the Table 3 are used to control relays. By default the GPIOs will be shorted (using removable jumpers) with respective relay pins as mentioned in above table. If you want to use those GPIOs for your own application instead of relays, then you have to disconnect by opening the jumper and make use of them.

b. APPLICATION WIRING DIAGRAM

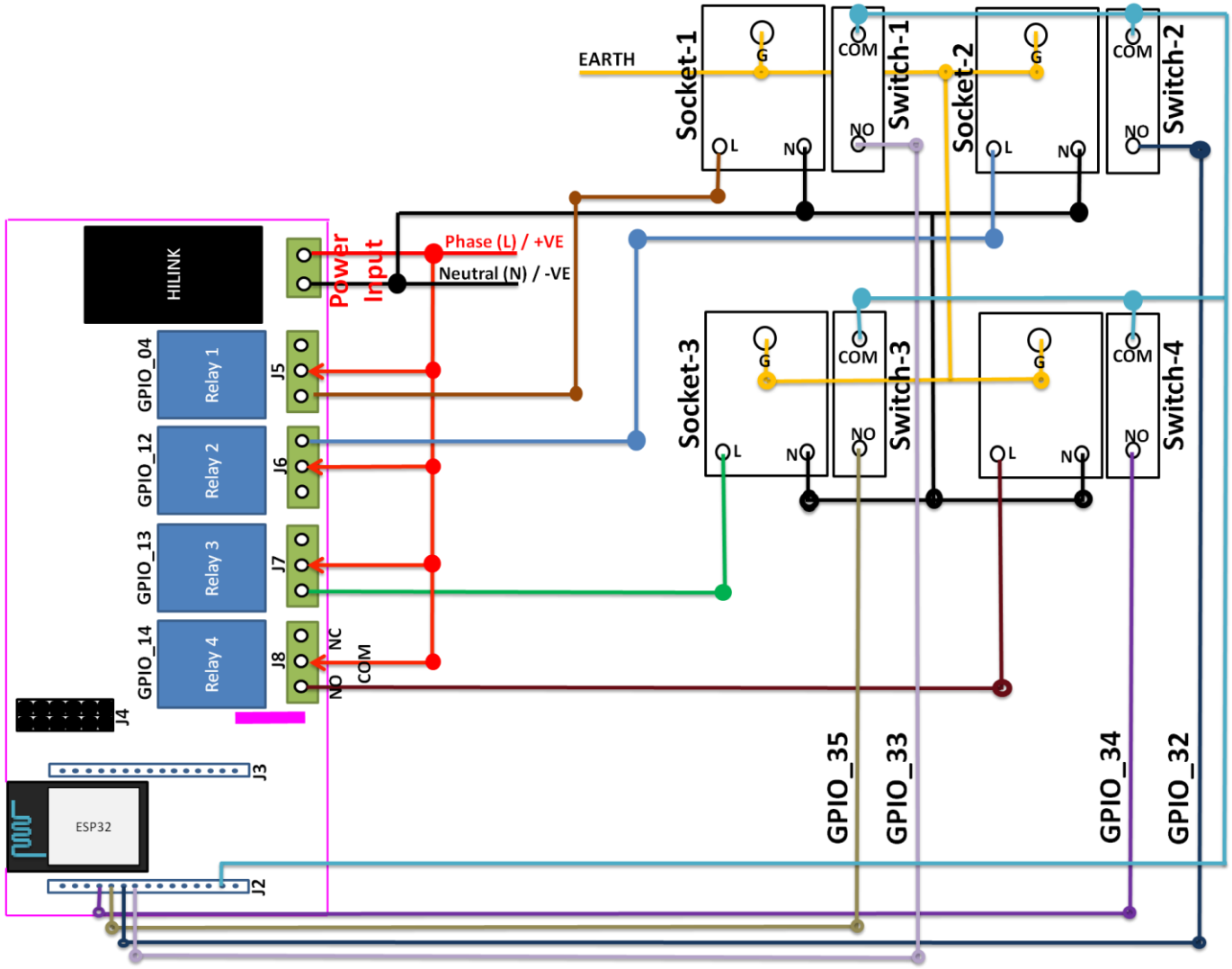


Figure 6: Application wiring example

Figure 6 represents about application connection diagram between load and board relay output (J5-J8) connectors. Phase is given to Common terminal and load shall be connected to the NO/NC terminal of the relay.

Relay No.	Relay GPIOs	Virtual Switch GPIOs
RELAY_1	GPIO_04	GPIO_33
RELAY_2	GPIO_12	GPIO_32
RELAY_3	GPIO_13	GPIO_35
RELAY_4	GPIO_14	GPIO_34

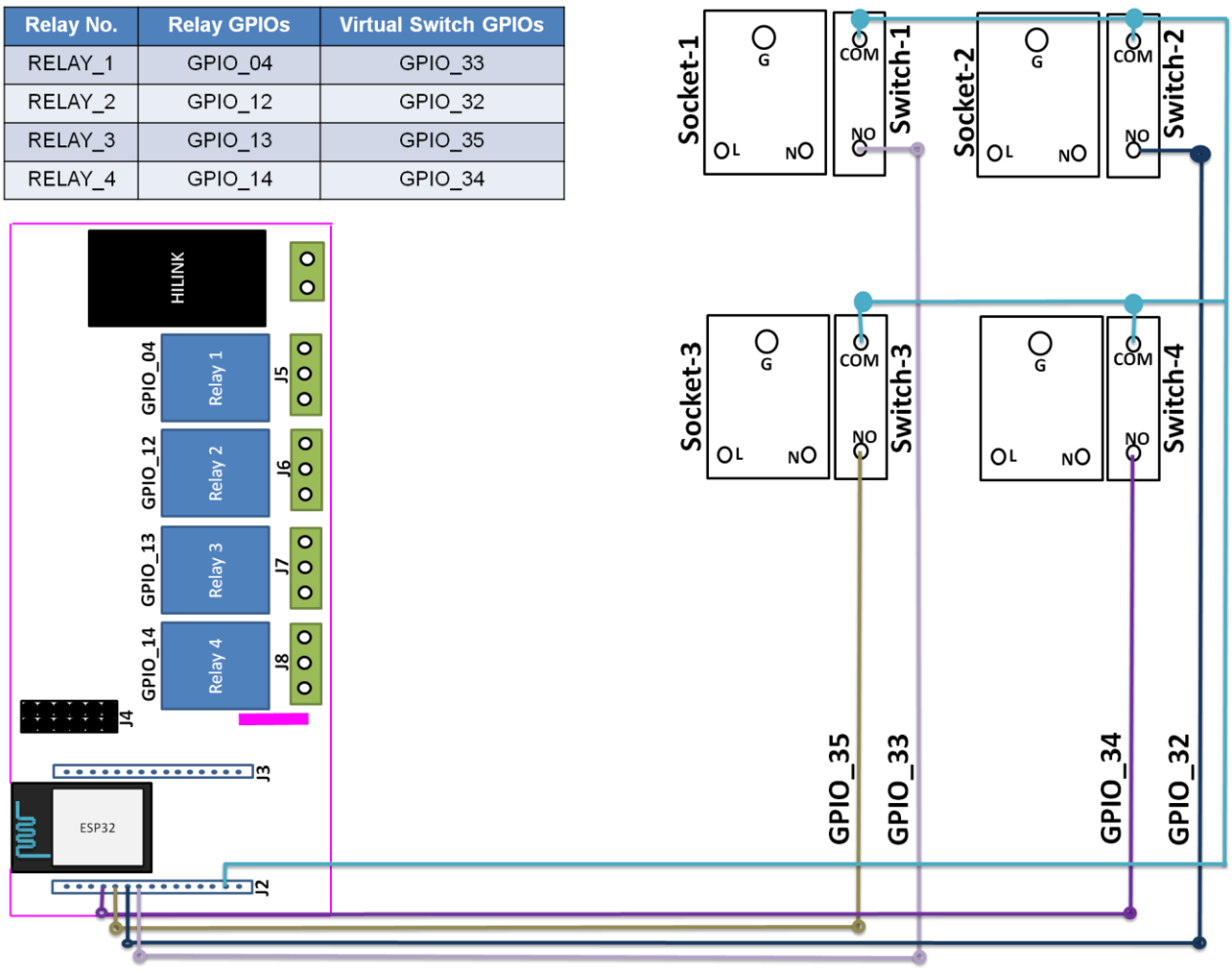


Figure 7: Application wiring example of DC connections

Note: Virtual switch GPIOs are 3.3 V DC connections, please do not connect AC lines to it. Connecting AC lines to it, may damage the board and may lead fire and cause hazardous effects.

Physical switch are connected to 4 GPIOs (refer to Table 4).

Relay No.	Relay GPIOs	Header Pin Number
Relay_01	GPIO_04	GPIO_33
Relay_01	GPIO_12	GPIO_32
Relay_02	GPIO_13	GPIO_35
Relay_03	GPIO_14	GPIO_34

Table 4: Virtual switch GPIO details

Relay No.	Relay GPIOs	Virtual Switch GPIOs
RELAY_1	GPIO_04	GPIO_33
RELAY_2	GPIO_12	GPIO_32
RELAY_3	GPIO_13	GPIO_35
RELAY_4	GPIO_14	GPIO_34

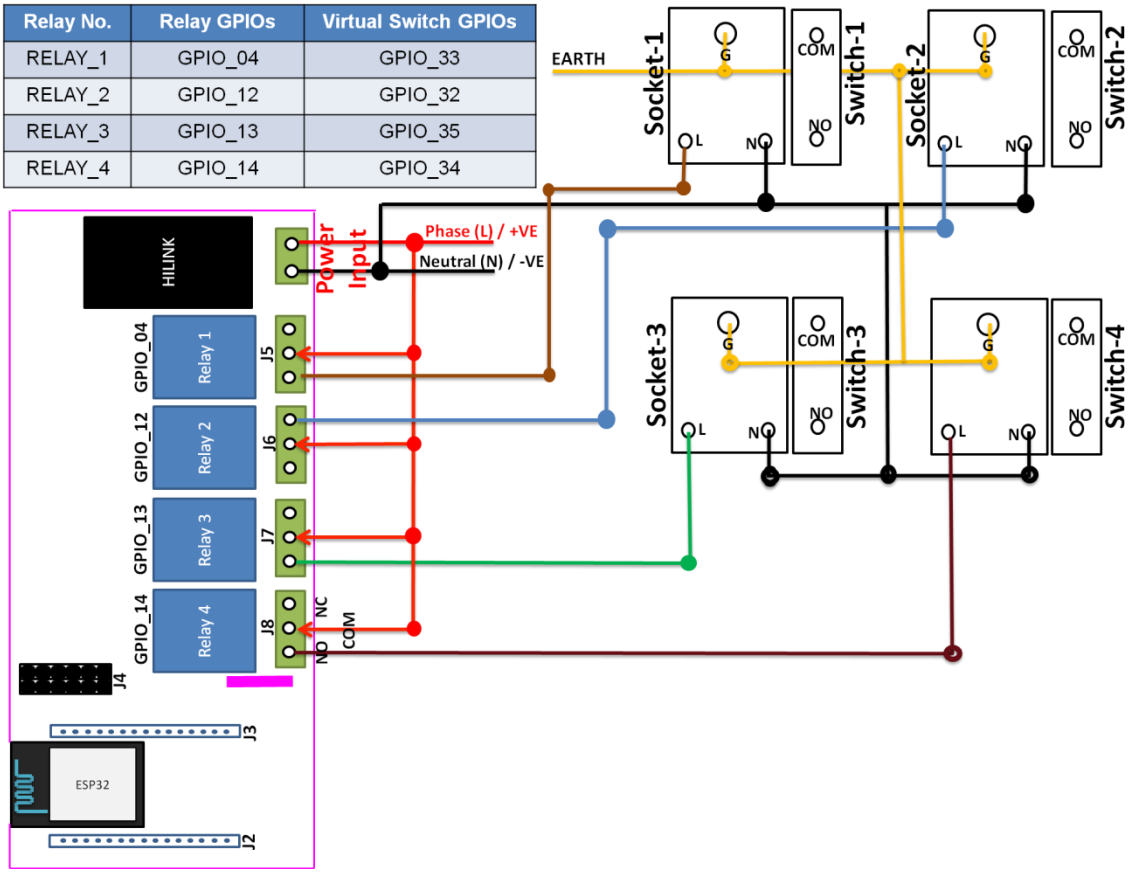


Figure 8: Application wiring example of AC connections

Outputs of relay 1-4 are connected to Socket 1-4 through NO and COM pin of relay. The socket will get power when the relay is triggered by virtual switch or mobile application. The advantage of this configuration is, it will act as two-way switch and
you can Turn-On the load through physical switch and Turn it OFF through relay or vice-versa.

The device can also be connected to Amazon Alexa by configuring using Alexa mobile App and OpenHab by configuring it with openhab sever.

11. MQTT COMMANDS TO READ INPUTS

a. COMMANDS TO TRIGGER RELAY THROUGH SUBSCRIPTION TOPIC

R4_ON ; Will turn-on the Relay_1
R4_OFF ; Will turn-off the Relay_1

R12_ON ; Will turn-on the Relay_2
R12_OFF ; Will turn-off the Relay_2

R13_ON ; Will turn-on the Relay_3
R13_OFF ; Will turn-off the Relay_3

R14_ON ; Will turn-on the Relay_4
R14_OFF ; Will turn-off the Relay_4

Note: “Subscription Topic” will be the name entered while configuring the board.

b. COMMAND TO RESET THE BOARD

Reset ; will reset the board and board will start hosting an AP.

Note: “Subscription Topic” will be the name entered while configuring the board.

c. REPONSE RECEIVED FROM THE BOARD THROUGH PUBLISHING TOPIC

On change in status of Relay_1
R04isON
R04isOFF

On change in status of Relay_2
R12isON
R12isOFF

On change in status of Relay_3
R13isON
R13isOFF

On change in status of Relay_4
R14isON
R14isOFF

Note: “Publishing Topic” will be the name entered while configuring the board.

12. HOW TO USE THE PRODUCT

a. STEPS TO CONFIGURE THE DEVICE TO NETWORK HOSTED BY YOU:

i. Switch ON the device.

ii. Make sure that Power presence indication **Green LED** is glowing.

iii. Take any Smartphone.

iv. Switch ON Wifi in it. (make sure that, its Mobile Data connection is turned OFF).

v. Search for available Wifi networks in the range

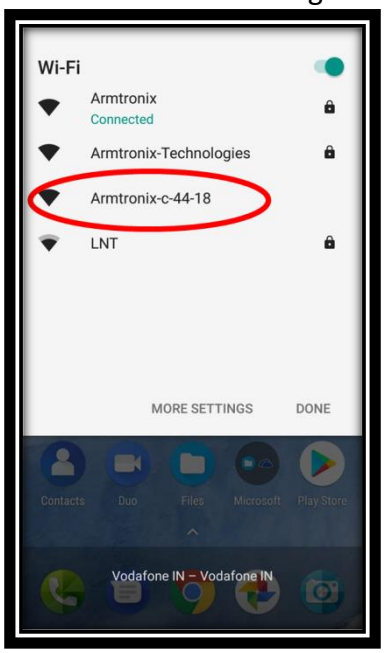


Figure 9: Available Wifi networks searched

vi. You will observe one of the available Wifi network as “Armtronix-xx-xx-xx”. Where xx: is last 6 digits of MAC address of the particular device. Click on that particular available network connect your smart phone to it. So in this scenario, the device is ‘Wifi Host’ and Smartphone is ‘Wifi Client’.

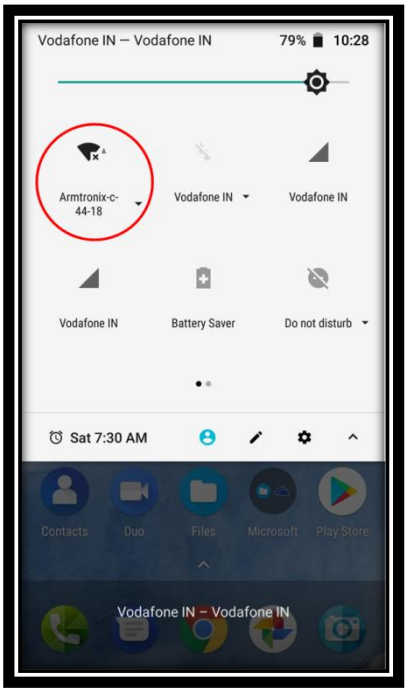


Figure 10: Smartphone Connected to Wifi hosted by board

vii. Open any web browser, enter default IP address 192.168.4.1 of the device when it is hosting its own Wifi network and click enter.

Figure 11: Default IP address entered in the Web browser

b. CONNECT VIA MQTT MODE

i. Clinking on Enter button after entering default IP address, you will be able to access its webpage as shown in Figure 9.

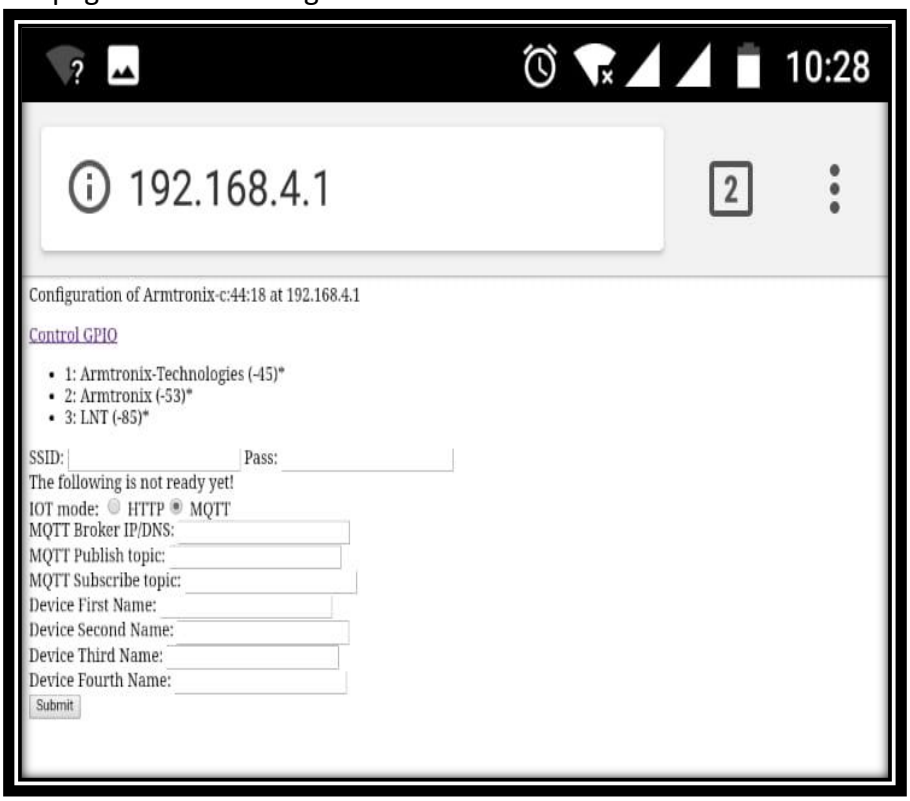


Figure 12: Accessed webpage of the device

ii. In the accessed webpage, fill-in all the required details like:

- **SSID** : SSID of Access Point
- **Pass** : Password of Access point
- **IOT Mode** : MQTT
- **MQTT Broker IP/DNS** : xxx.xxx.xxx.xxx (Ex. 192.168.0.1)
- **Publish to Topic 1 (IN)** : /I/xxx (Ex. /I/008)
- **Subscribe to topic 1 (OP)** : /O/xxx(Ex. /O/008)
- **Device Name** : Alexa command name; Which we ask Alexa to trigger

Ex. Alexa turn ON **light** ; **"**Light" is the name of the device
Ex. Alexa turn ON **Fan** ; **"**Fan" is the name of the device
Ex. Alexa turn ON **Charger** ; **"**Charger" is the name of the device
Ex. Alexa turn ON **Tube light** ; **"**Tube Light" is the name of the device

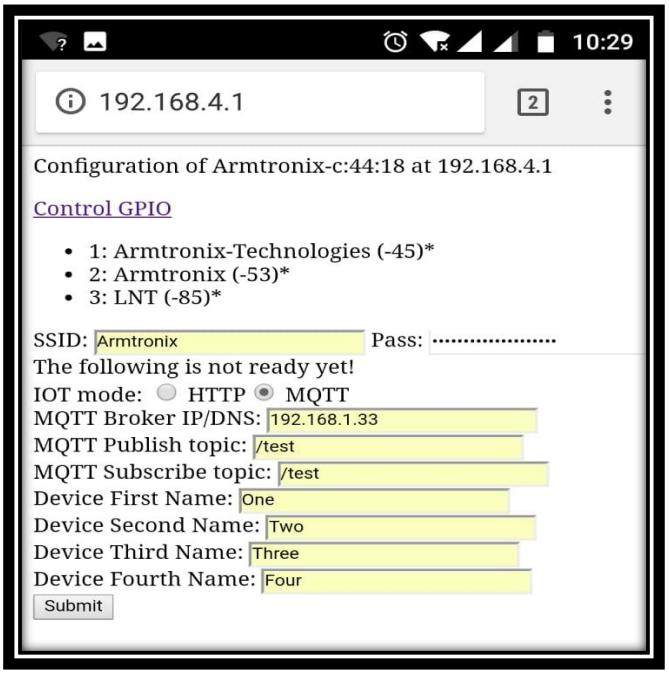


Figure 13: Entered all the required details

iii. After entering all the required details, click on Submit button. It will save the parameters you entered and reboot the device and acknowledge the user in the webpage.
Do not turn OFF the device, it will automatically reboot.

c. CONNECT VIA MQTT MODE

i. Clinking on Enter button after entering default IP address, you will be able to access its webpage as shown in Figure 9.

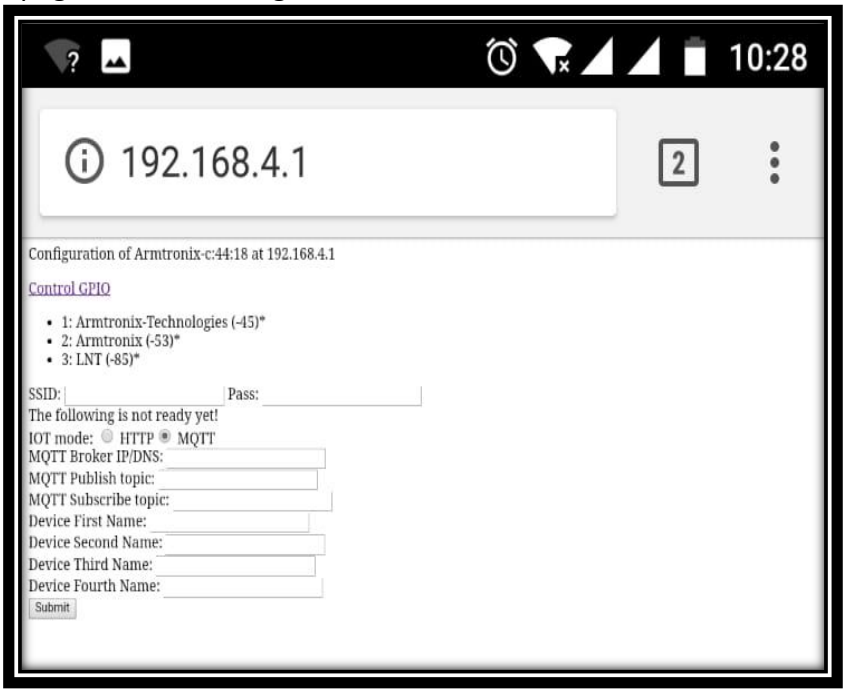


Figure 14: Accessed webpage of the device

ii. In the accessed webpage, fill-in all the required details like:

- **SSID** : SSID of Access Point
- **Pass** : Password of Access point
- **IOT Mode** : HTTP
- **Device Name** : Alexa command name; Which we ask Alexa to trigger

Ex. Alexa turn ON **light** ; **“**Light” is the name of the device
Ex. Alexa turn ON **Fan** ; **“**Fan” is the name of the device
Ex. Alexa turn ON **Charger** ; **“**Charger” is the name of the device
Ex. Alexa turn ON **Tube light** ; **“**Tube Light” is the name of the device

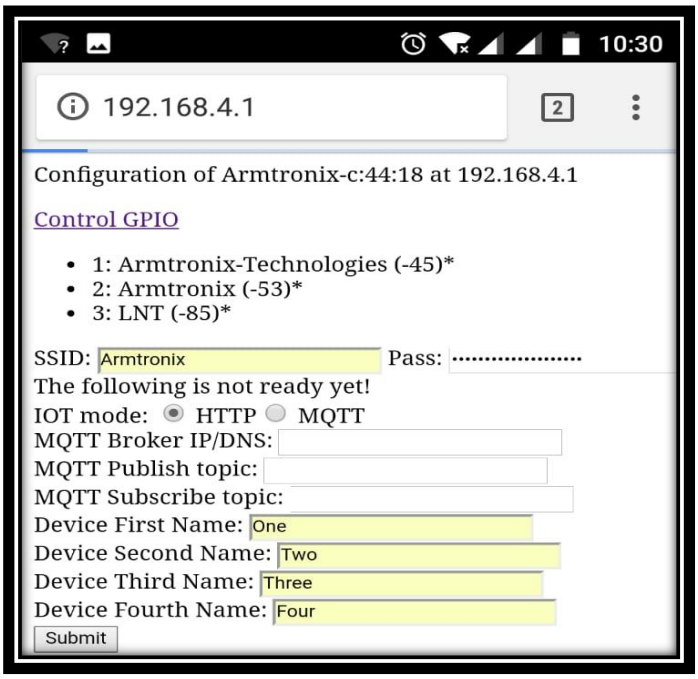


Figure 15: Entered all the required details

iii. After entering all the required details, click on Submit button. It will save the parameters you entered and reboot the device and acknowledge the user in the webpage.
Do not turn OFF the device, it will automatically reboot.

d. STEPS TO CONNECT SMARTPHONE TO MQTT BROKER / WIFI ROUTER / ACCESS POINT:

i. Disconnect Smartphone from any other Wifi network if connected.

ii. Search for available Wifi network where the MQTT broker / WIFI ROUTER / ACCESSVPOINT is running.
In our case it is “Armtronix-Home” is the wifi network where our MQTT broker is running.

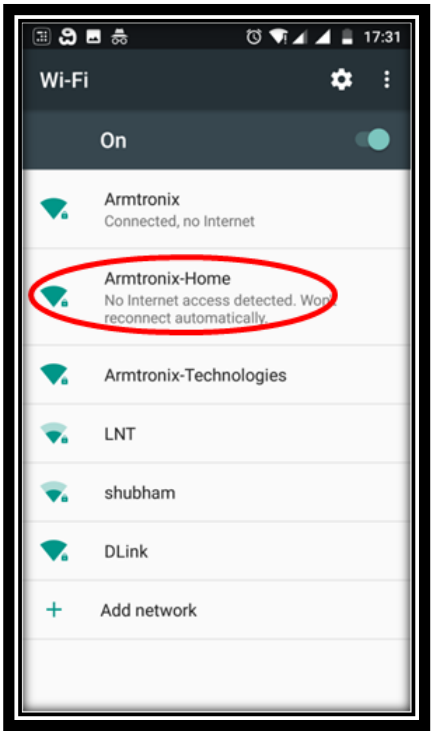


Figure 16: Smartphone searched for available Wifi networks

iii. Click on that particular available network to connect your smart phone to it.

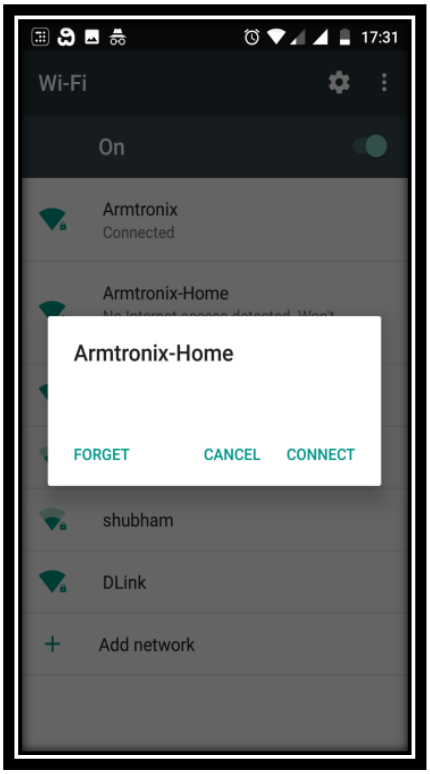


Figure 17: Trying to connect to pre-configured MQTT broker

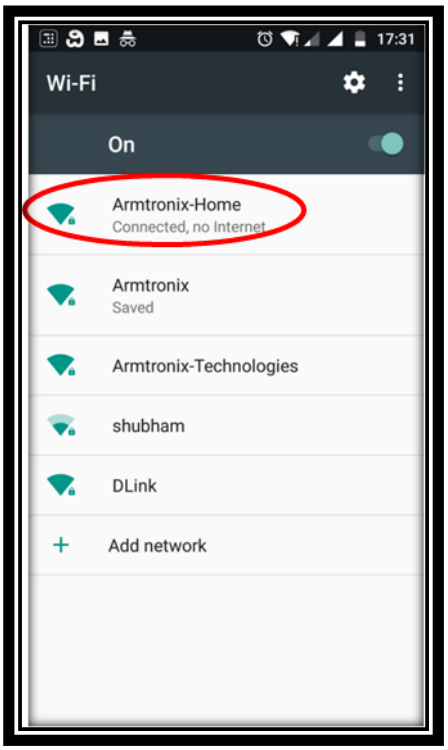


Figure 18: Smartphone connected to MQTT broker

e. STEPS TO TEST THE DEVICE USING SMARTPHONE AND MQTT BROKER:

i. Install 'MyMQTT' Android app in to a Smartphone you would use for testing.

ii. Open an app 'MyMQTT' app Smartphone.

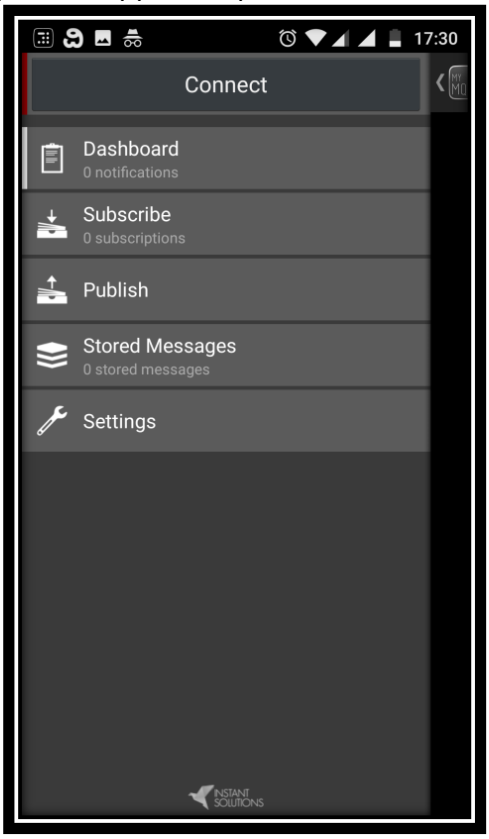


Figure 19: MyMQTT app menu page

iii. Click on settings option.

iv. Enter MQTT broker IP address and default Port number as 1883 (if not changed)
Our MQTT broker IP address is 192.168.0.1

Figure 20: MQTT broker IP address and port number entered

v. On the completion of your IP address and port number entry, Save the settings by clicking on **Save** button. Popup will indicate once the settings saved.

Figure 21: Saved the settings

f. CONTROL OUTPUTS VIA SMARTPHONE:

i. Connect Smartphone to network hosted having MQTT broker as said in section 13.d.
ii. Open MyMQTT app in Smartphone.
iii. Tap on the screen, it will open menu window.
iv. Click on the Publish option.
v. Enter topic “as you entered while configuring”
vi. Ex. Message given in the section 11.a” (Device will trigger respective relay(from Relay-1 to Relay-4,) based on message you publish)

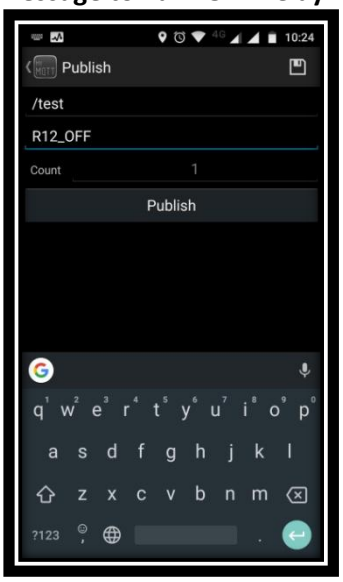
Message to Turn ON Relay-2	**Message to Turn OFF Relay-2**
	

Figure 22 Entered topic and message to control outputs

vii. Click on Publish button to publish the topic.

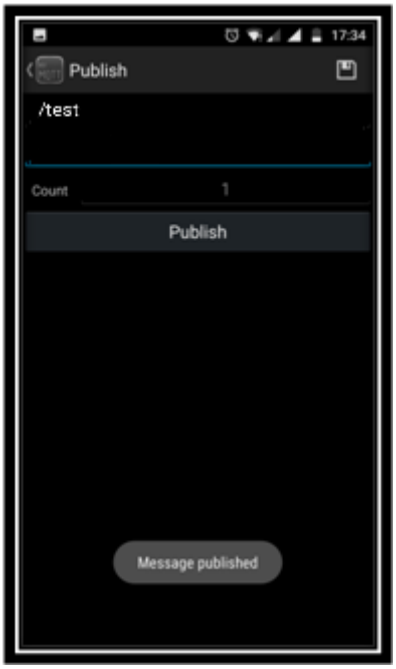


Figure 23: Message and topic published to control outputs

viii. On publishing the topic, popup will arrive as ‘*Message Published*’ the device will take action on the outputs.

g. RESET THE DEVICE USING MQTT COMMAND VIA SMARTPHONE.

i. Connect Smartphone to network hosted having MQTT Broker as said in section 13.b.
ii. Open MyMQTT app in Smartphone.
iii. Tap on the screen, it will open

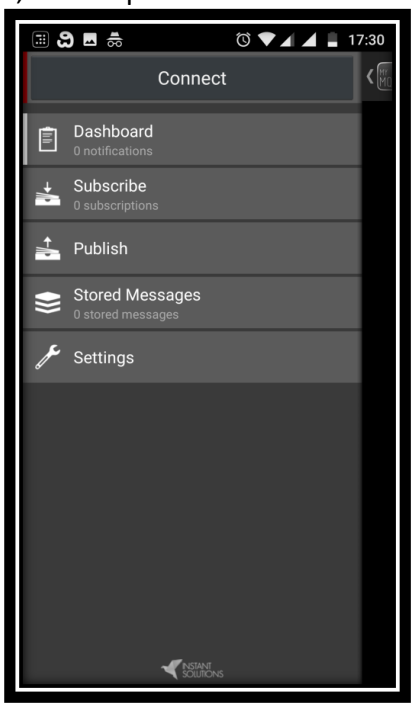


Figure 24: Taped on the default screen

iv. Click on the Publish option.
v. Enter topic “as you entered while configuring”
vi. Enter Message as “Reset” (Device will get RESET)

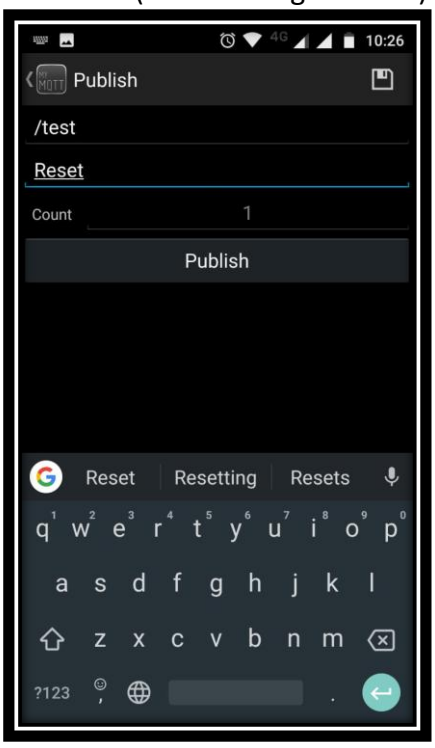


Figure 25: Clicked on the publish option and entered the message to be displayed on LCD

vii. Click on Publish button on the screen to publish the topic.

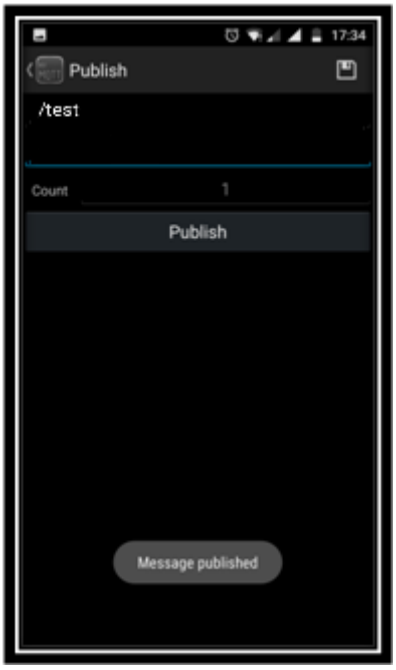


Figure 26: Published the message by clicking on Publish button

viii. On publishing on the topic, popup will arrive as '*Message Published*' the device will take action on the outputs.

h. READ DIGITAL INPUTS VIA SMARTPHONE.

i. Connect Smartphone to network hosted having MQTT Broker as said in section 13.b.
ii. Open MyMQTT app in Smartphone.
iii. Tap on the screen, it will open menu window.

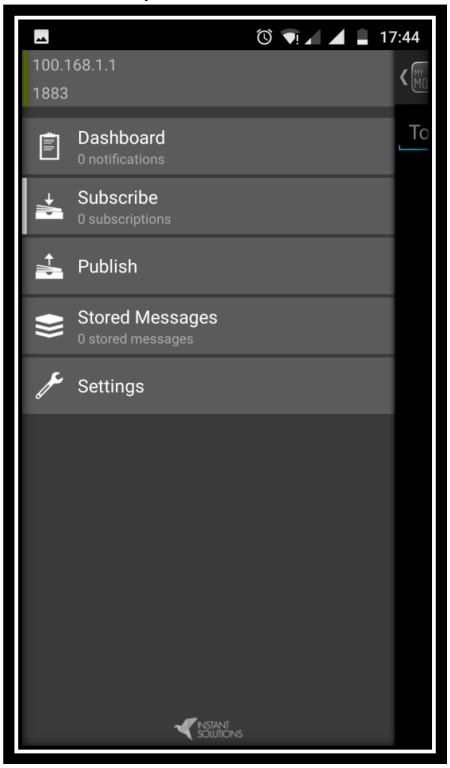


Figure 27: Tapped on the home screen

iv. Click on the Subscribe option.

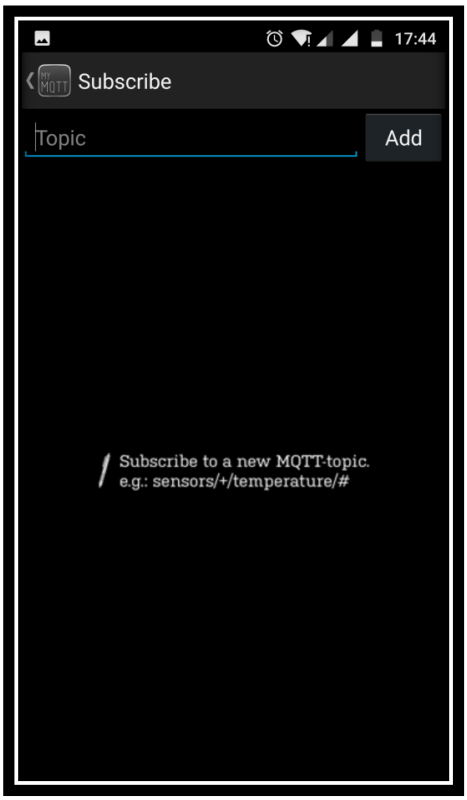


Figure 28: Clicked on the Subscribe option

i. Enter subscription topic “as you entered while configuring”

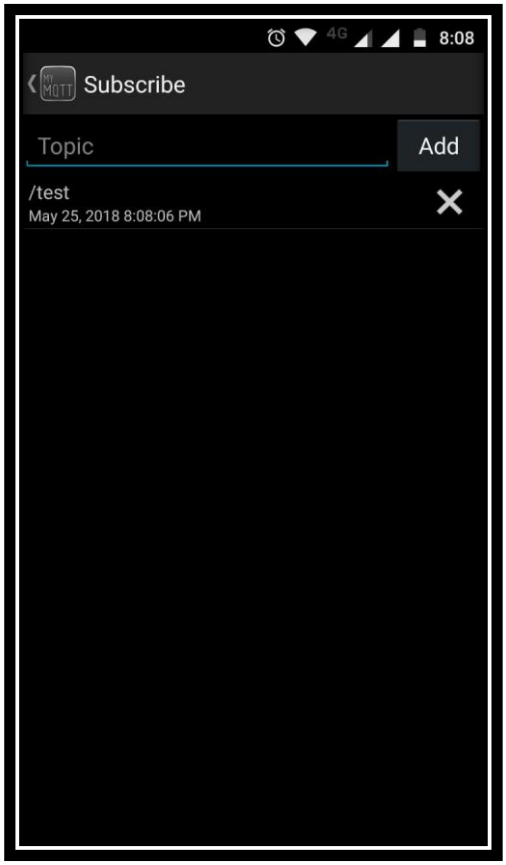


Figure 29: Entered the Subscription topic and clicked on the Add button

v. Click on back button located at left-top-corner of the screen.

vi. Tap on the screen. It will open the menu.

vii. Open the dashboard by clicking on Dashboard option in the menu.

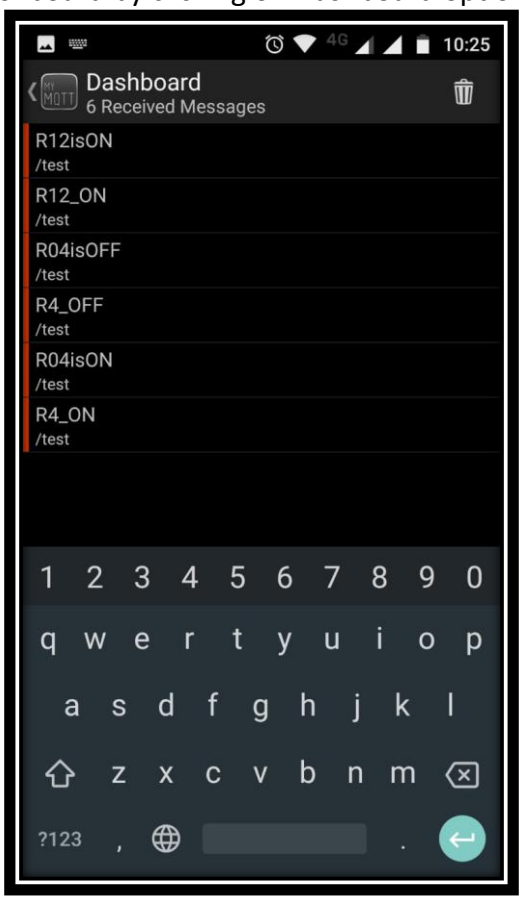


Figure 30: Dashboard window to monitor status of Digital Inputs

viii. You will receive the status of load, as and when there is a change in status.

13. Openhab Example

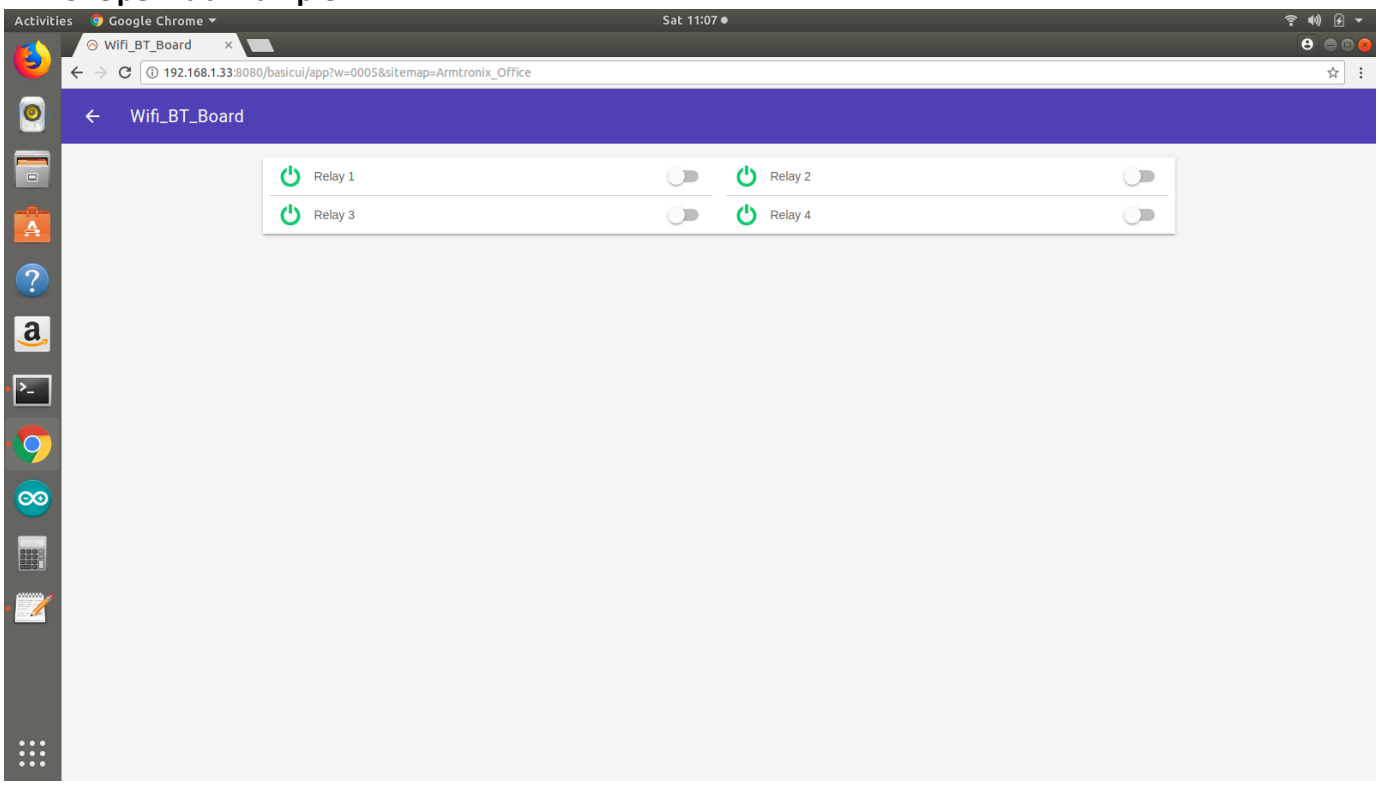


Figure 31: Openhab image of 4 relay board

a. Example of Openhab files in MQTT mode

In our case,

- MQTT broker's name is "**broker**"
- Topic for publishing and subscription is "**/test**"
- Sitemap file name is **Armtronix_Office.sitemap**
- Item file name is **Armtronix_Office.items**
- Map file names are **r4.map**, **r12.map, r13.map and r14.map**

Path of sitemap file: **/etc/openhab2/sitempas/Armtronix_Office.sitemap**

Code of sitemap file:

```
sitemap demo label="Armtronix Office"
{
 Frame
  {
Group item=tsu label="Wifi_BT_Board" icon="group"
  {
    Switch item=GPIO4 label="Relay 1"
    Switch item=GPIO12 label="Relay 2"
    Switch item=GPIO13 label="Relay 3"
    Switch item=GPIO14 label="Relay 4"
  }
  }
}
```

Path of item file: **/etc/openhab2/items/Armtronix_Office.items**

Code of item file:
Group All
Group tsu (All)

Switch GPIO4 "Relay 1" (tsu,Lights)
{mqtt=">[broker:/test:command:ON:R4_ON],>[broker:/test:command:OFF:R4_OFF],[broker:/test:state:MAP(r4.map)]",autoupdate="false"}

Switch GPIO12 "Relay 2" (tsu,Lights)
{mqtt=">[broker:/test:command:ON:R12_ON],>[broker:/test:command:OFF:R12_OFF],<[broker:/test:state:MAP(r12.map)]",autoupdate="false"}

Switch GPIO12 "Relay 2" (tsu,Lights)
{mqtt=">[broker:/test:command:ON:R12_ON],>[broker:/test:command:OFF:R12_OFF],<[broker:/test:state:MAP(r12.map)]",autoupdate="false"}

Switch GPIO14 "Relay 4" (tsu,Lights)
{mqtt=">[broker:/test:command:ON:R14_ON],>[broker:/test:command:OFF:R14_OFF],<[broker:/test:state:MAP(r14.map)]",autoupdate="false"}

Path of map file: **/etc/openhab2/transform/r4.map**

Code of map file r4:
R04isON=ON
R04isOFF=OFF

Path of map file: **/etc/openhab2/transform/r12.map**

Code of map file r12:
R12isON=ON
R12isOFF=OFF

Path of map file: **/etc/openhab2/transform/r13.map**

Code of map file r13:
R13isON=ON
R13isOFF=OFF

Path of map file: **/etc/openhab2/transform/r14.map**

Code of map file r14:
R14isON=ON
R14isOFF=OFF

14. 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 ESP32:

1. Use external mobile USB A type to micro USB data cable between computer and device.
2. Connect Micro USB cable between your computer and U5 of “Wifi/BT Quad Relay Board”.
3. Open your code in Arduino IDE as shown.

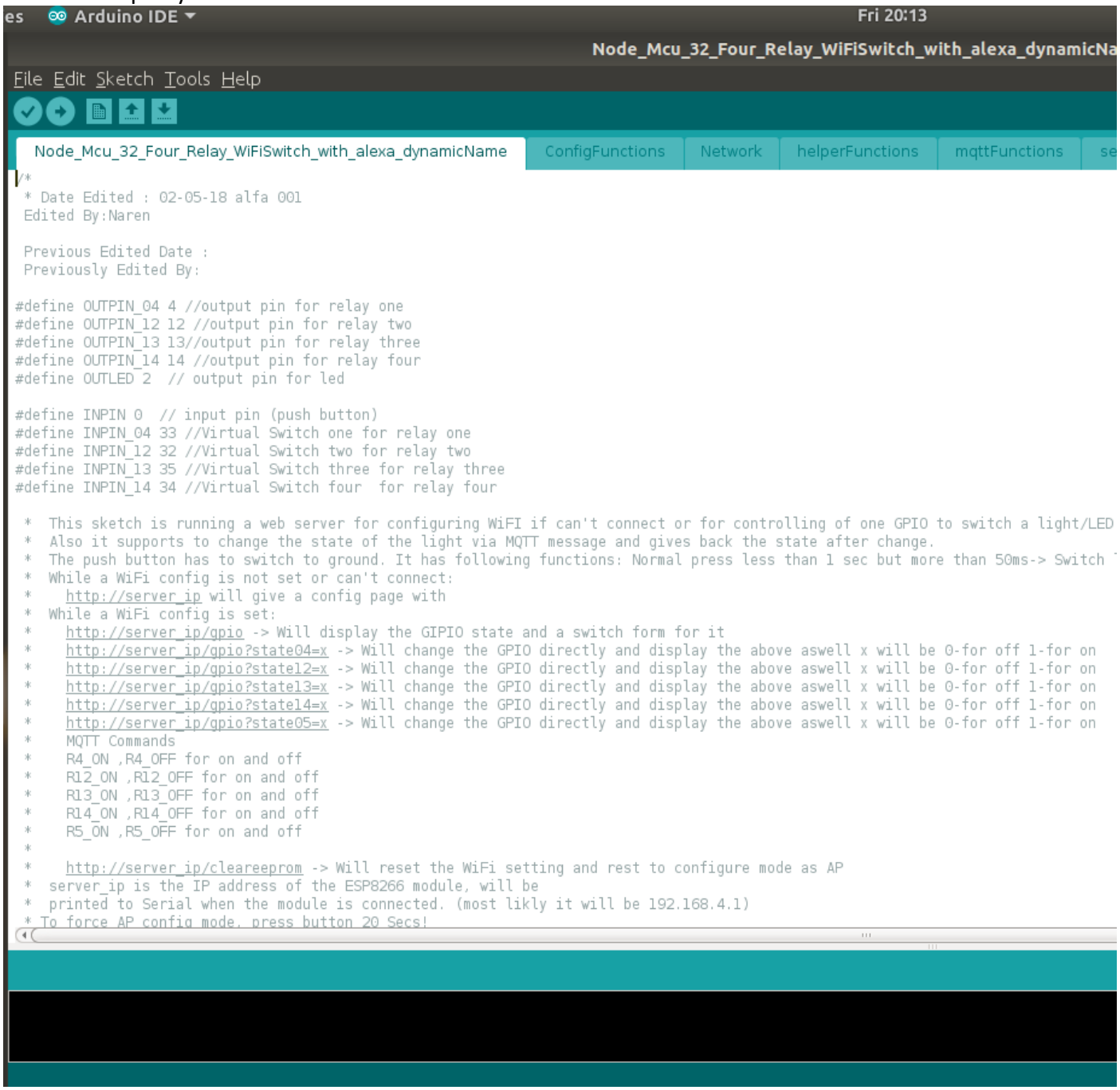


Figure 32: Program Opened in IDE

4. Click on Tools Tab, move mouse pointer on “Board: xxxxxxxxxx” and click on “NodeMCU-32S” as shown in figure 10.

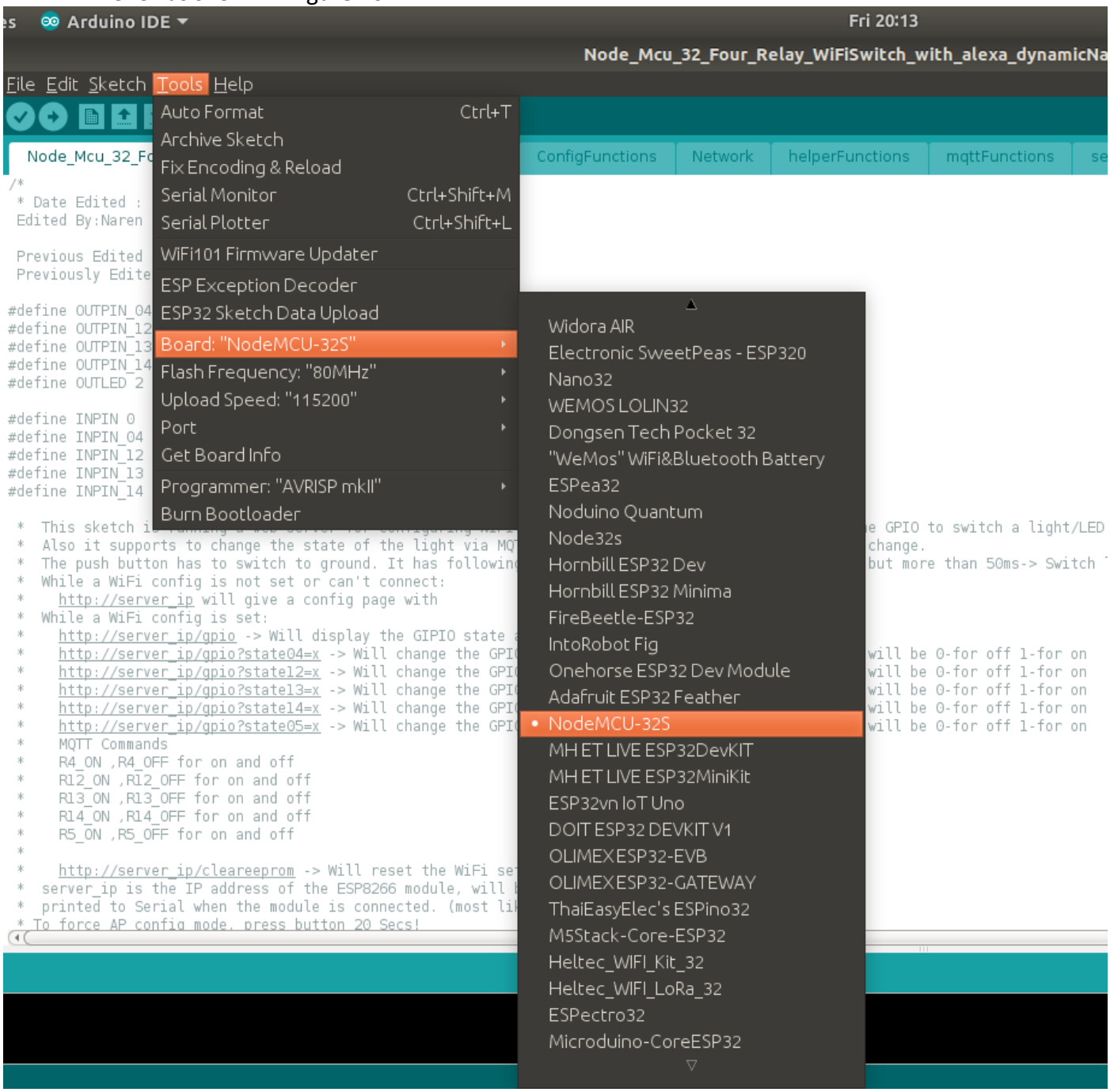


Figure 33: Board Selection

5. Select Upload Speed as “115200”.

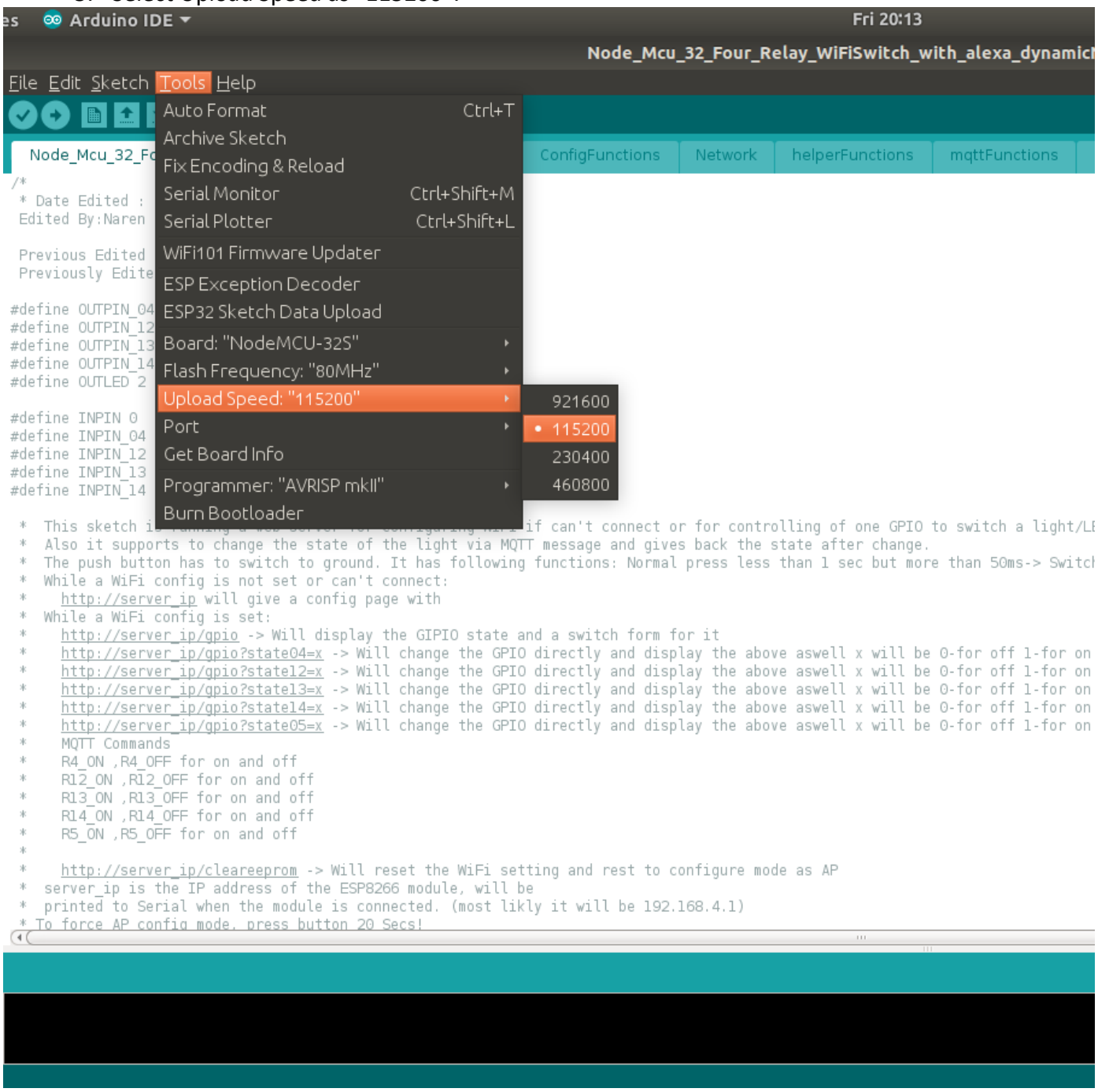


Figure 34: Baudrate selection

6. Click on tools tab, move mouse pointer to “Programmer: “Arduino as ISP’”, under this click on “Arduino as ISP”.
7. 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).
8. Run the program. Refer to Figure 13.

IMPORTANT NOTICE

ARMtronix Technologies LLP and its subsidiaries reserve the right to make corrections, enhancements, improvements and other changes to its products and services and 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----------