INTRODUCTION:

ARMTRONIX WIFI Two Triac Dimmer Board V0.1

Armtronix Wifi dimmer is a IOT board it is designed for home automation. The features of the board are:

1]Wireless control

2]Small form factor

3]On board AC to DC power supp1y 230VAC to 5V DC.

4]DC virtual switch

5] Two channels(one for on and off another for dimming)

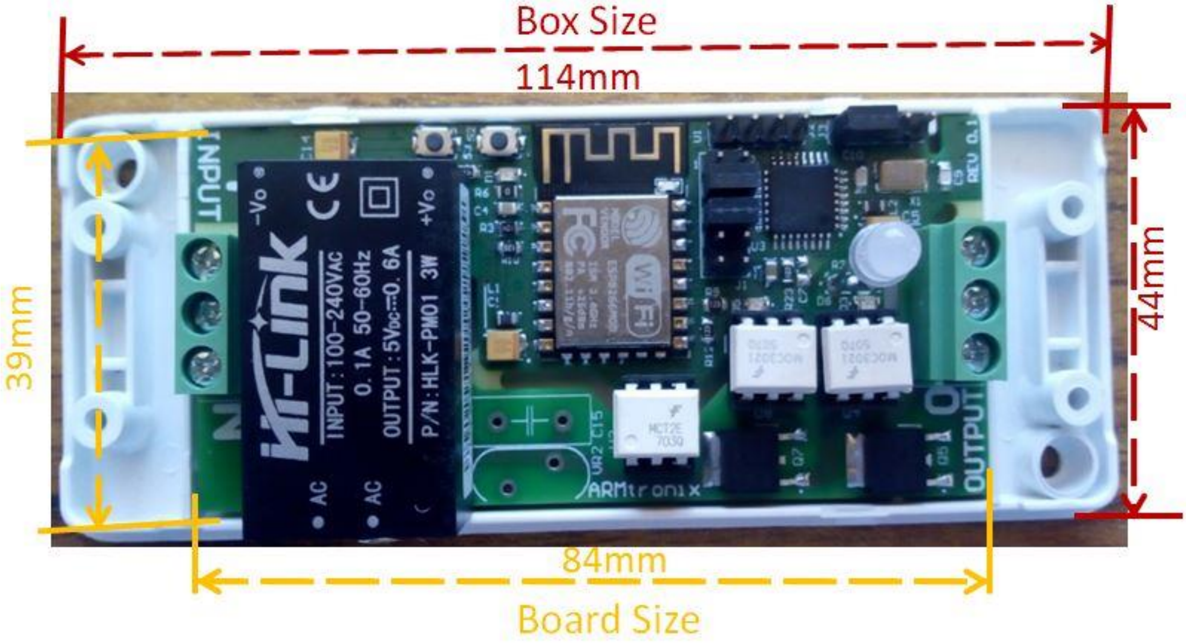


Diagram 1

The size of the board is 84mmX39mm and with box size 114 mmX44mm, as its shown in diagram1, has a capacity to drive 1Amps load. The board has the Wifi module and microcontroller (atmega328) which is used to control the triac through HTTP or MQTT. The board has two DC virtual switches which can be used to control on and off and dimming.

The board also has Power module AC to DC of 100-240VAC to 5V up to 0.6A, triac BT136 and Terminal connector. There is Zero cross detection is also available. There are two triac used one for dimming and another for switching.

Header details

The diagram2 gives the details of the headers and terminal blocks

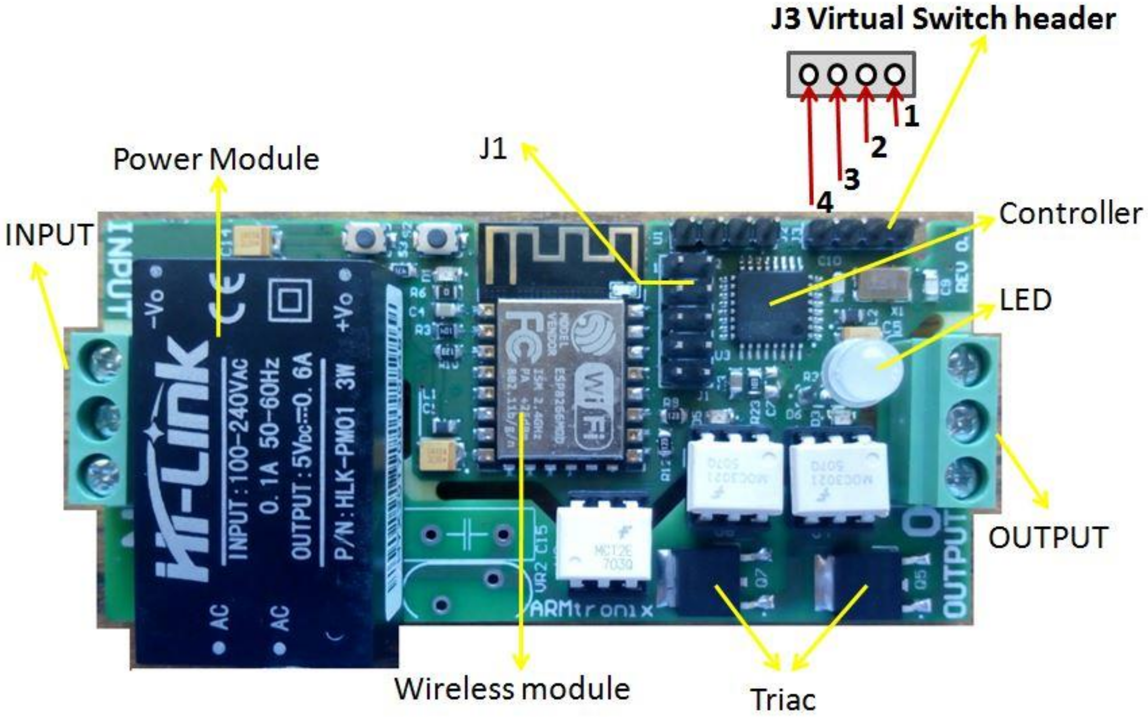
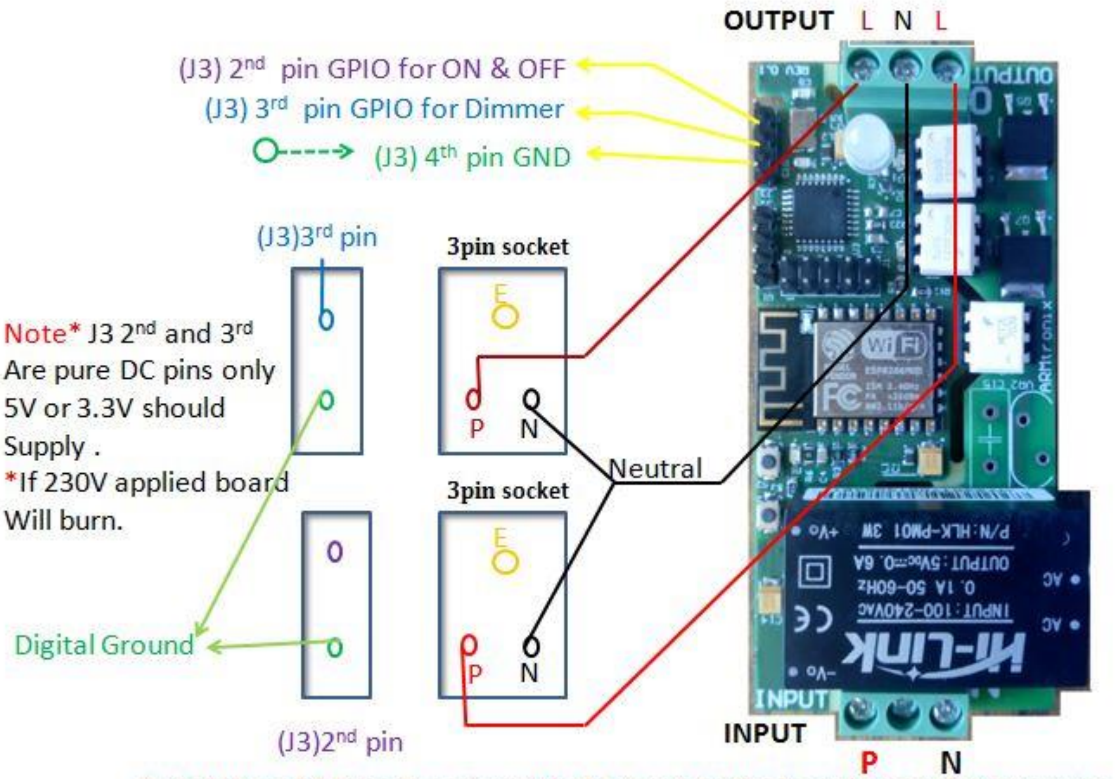


Diagram 2

To board 230VAC is applied to the input terminal block and load is applied to the output terminal block.

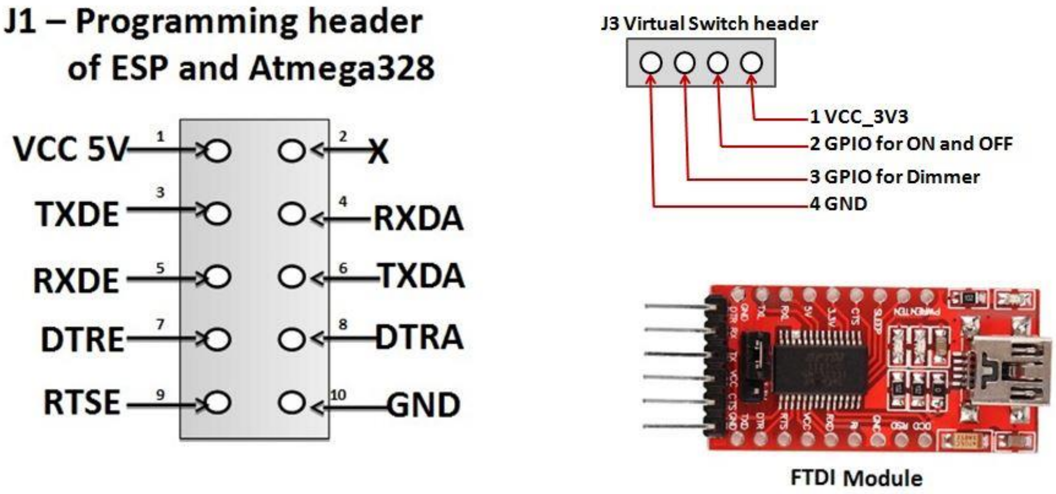
On the board J3 header is used for the dc virtual switch the header details can be referred form the diagram4. First pin is vcc3.3v, second pin is atmega gpio pin for arduino programming we need to use A4 (ON&OFF), third pin is atmega gpio pin for arduino programming we need to use A5(DIMMING) and fourth pin is ground. For dc virtual switch we are using only second and third pin i.e A4, A5 and Fourth pin i.e ground, this is mentioned in the diagram3 for connection of virtual switch.



Note*Use a 1A Fuse in series with the input to the board as a safety measure

Diagram 3

NOTE* J1,J2,J3 are pure DC headers if 230v applied to these headers bord will burn and may cause saftey issue



TXDA – TX of Atmega
RXDA –RX of Atmega

TXDE –TX of Esp
RXDE –RX of Esp

RTSE – RTS pin of ESP

DTRE –DTR pin of Esp
DTRA –DTR pin of Atmega

Diagram 4

Programming details

J1 Header is used to upload the firmware to ESP or atmega through the FTDI Module, details of headers can be found in the diagram4. To upload the new firmware to esp using FTDI

Make the following connection

1]Connect the RX of FTDI to TXDE pin of J1

2]Connect the TX of FTDI to RXDE pin of J1

3]Connect the RTS of FTDI to RTSE pin of J1

4]Connect the DTR of FTDI to DTRE pin of J1

5]Connect the Vcc5V of FTDI to VCC5v pin of J1

6]Connect the GND of FTDI to GND pin of J1

Please refer the fallowing link for code

https://github.com/armtronix/Wifi-Two-Dimmer-Board

In this code commonly used HTTP port is 80, we can change port number, whichever user needs to use based on his application , refer below

```
//##### Object instances #####

MDNSResponder mdns;

ESP8266WebServer server(80);

WiFiClient wifiClient;

PubSubClient mqttClient;

Ticker btn_timer;

Ticker otaTickLoop;
```

After making connection, connect to the USB port, initial we need to install the driver to detect it detect the com port , in this way user can program upload the firmware.

Similarly to uplaod the firmware to the atmega follow the connection

1]Connect the RX of FTDI to TXDA pin of J1

2]Connect the TX of FTDI to RXDA pin of J1

3]Connect the DTR of FTDI to DTRA pin of J1

4]Connect the Vcc5V of FTDI to VCC5v pin of J1

5]Connect the GND of FTDI to GND pin of J1

Please refer the fallowing link for code

https://github.com/armtronix/Wifi-Two-Dimmer-Board

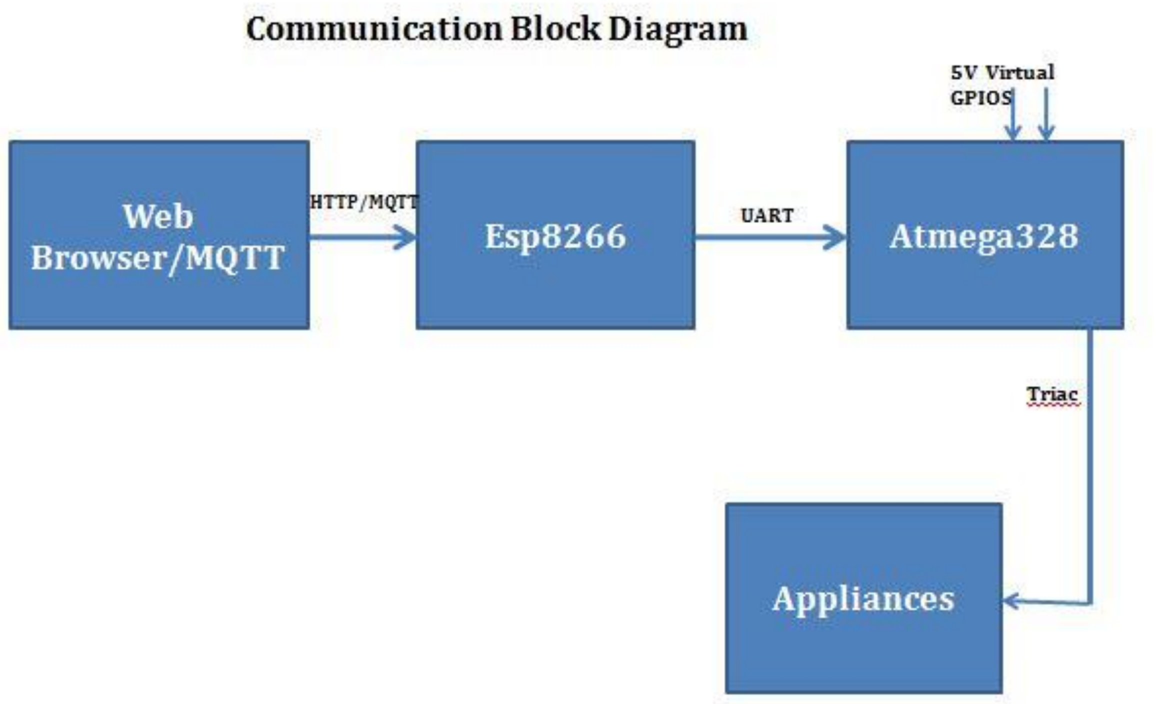
we are using 6gpios to control two triac dimmer board, two for controlling triac, two for controlling LED, anther two for controlling virtual switch. Gpios are

```
//Triac no.
#define NON_DIMMABLE_TRIAC 8 //Gpio 8
#define DIMMABLE_TRIAC     9 //Gpio 9
/*Dual colour LED*/
#define DLED_RED   3
#define DLED_GREEN 4

//manual switch
#define SWITCH_INPIN1 A5 //switch 1
#define SWITCH_INPIN2 A4 //switch 2
```

After making the connection user can able to upload the firmware to atmega. After programming both ESP and Atmega we have to establish connection between ESP and Atmega by shorting pins 3-4 of J1 header and 5-6 of J1 header using jumpers setting.

Block Diagram



Web browser/MQTT

We can control this device through HTTP/MQTT. Http client sends an http request to esp8266 according to http standard, specifying the information the client like to retrieve from the esp8266. MQTT stands for MQ Telemetry Transport. It is a nice lightweight publish and subscribe system where you can publish and receive messages as a client. It makes it really easy to establish a communication between multiple devices. It is a simple messaging protocol, designed for constrained devices and with low-bandwidth.

ESP8266

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. WiFi is a technology that uses radio waves to provide network connectivity. A WiFi connection is established using a wireless adapter to create hotspots areas in the vicinity of a wireless router that are connected to the network and allow users to access internet services. Programming to esp8266 is explained above and configuration details explained below.

Atmega328

This is 32 pin controller, Applications low-powered, low-cost micro-controller is needed. Perhaps the most common implementation of this chip is on the popular Arduino development platform, namely the Arduino Uno and Arduino Nano models. We used 6 gpios from this controller two for triac controlling other two for LED another two gpios are DC 5v gpios to control virtual switch.

Appliances

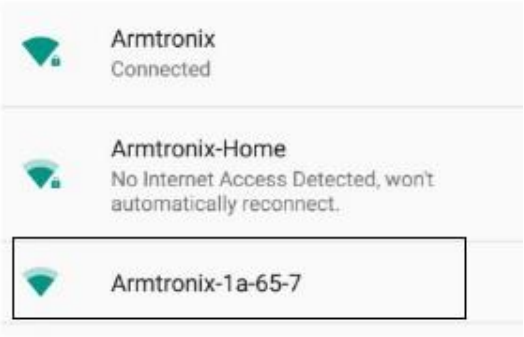
Home appliances like light and fan, this board provides two channels one for switching and other for dimming , you can also be use two channels as switching, for this application we already build the code, you can also be use two channels as dimming for this application you need modify our code. For code please refer this link https://github.com/armtronix/Wifi-Two-Dimmer-Board

Configuration details

Power the board with Input with 230V AC the device will host the access point as shown in diagram5, connect the mobile to access point with Armtronix-(mac) EX: Armtronix-1a-65-7 as shown in diagram6.After connecting open browser and type 192.168.4.1 IP address in the browser,it will open the web server as shown in the diagram7, fill the SSID and password and select http, if user wants to connect to mqtt then he has to select mqtt radio button and enter mqtt broker IP address and enter mqtt publish topic and mqtt subscribe topic and submit.

After configuring the submit 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 relay.

Without configuring the SSID and Password we can control the relay 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 diagram7 by clicking this link also we can control the relay but the response will be slow.

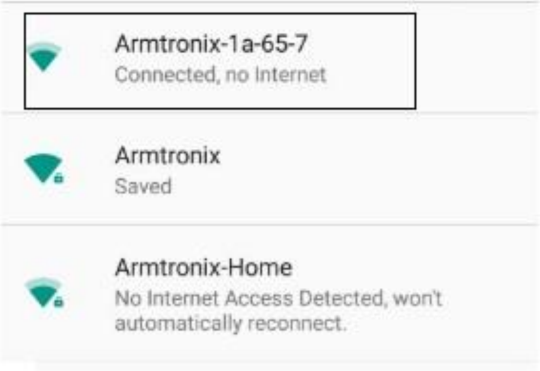


Armtronix
Connected

Armtronix-Home
No Internet Access Detected, won't automatically reconnect.

Armtronix-1a-65-7

Diagram 5

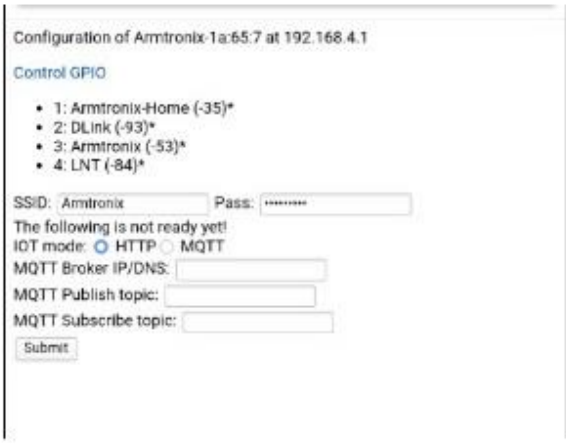


Armtronix-1a-65-7
Connected, no Internet

Armtronix
Saved

Armtronix-Home
No Internet Access Detected, won't automatically reconnect.

Diagram 6



Configuration of Armtronix-1a:65:7 at 192.168.4.1

Control GPIO

- 1: Armtronix-Home (-35)*
- 2: DLink (-93)*
- 3: Armtronix (-53)*
- 4: LNT (-84)*

SSID: Armtronix Pass: *********

The following is not ready yet!

IOT mode: ● HTTP ○ MQTT

MQTT Broker IP/DNS:

MQTT Publish topic:

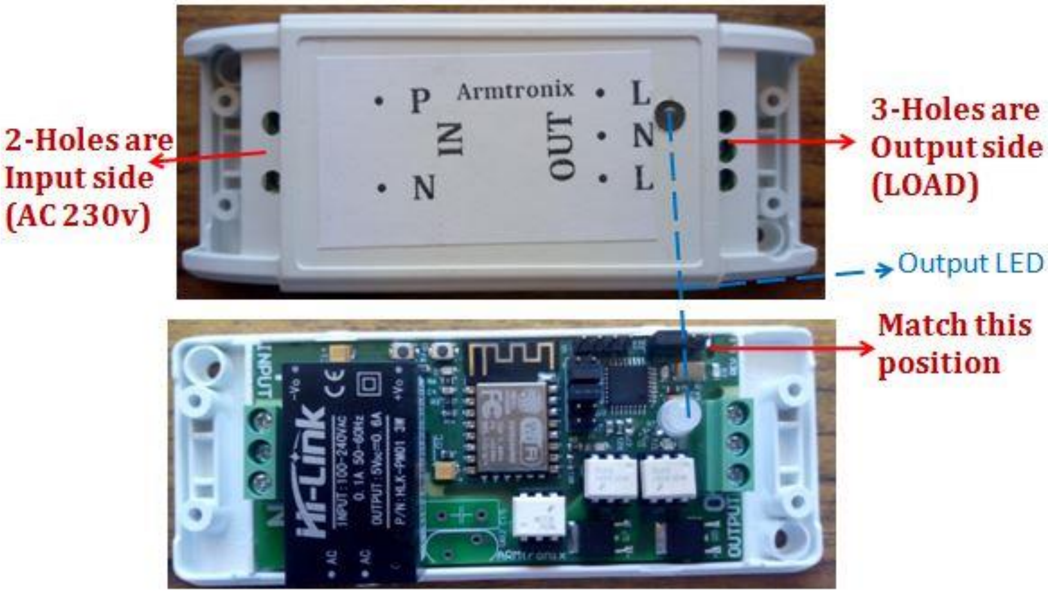
MQTT Subscribe topic:

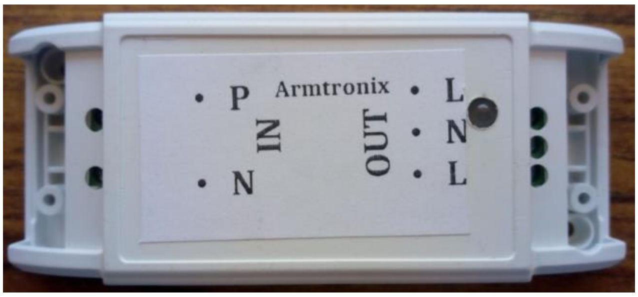
Diagram 7

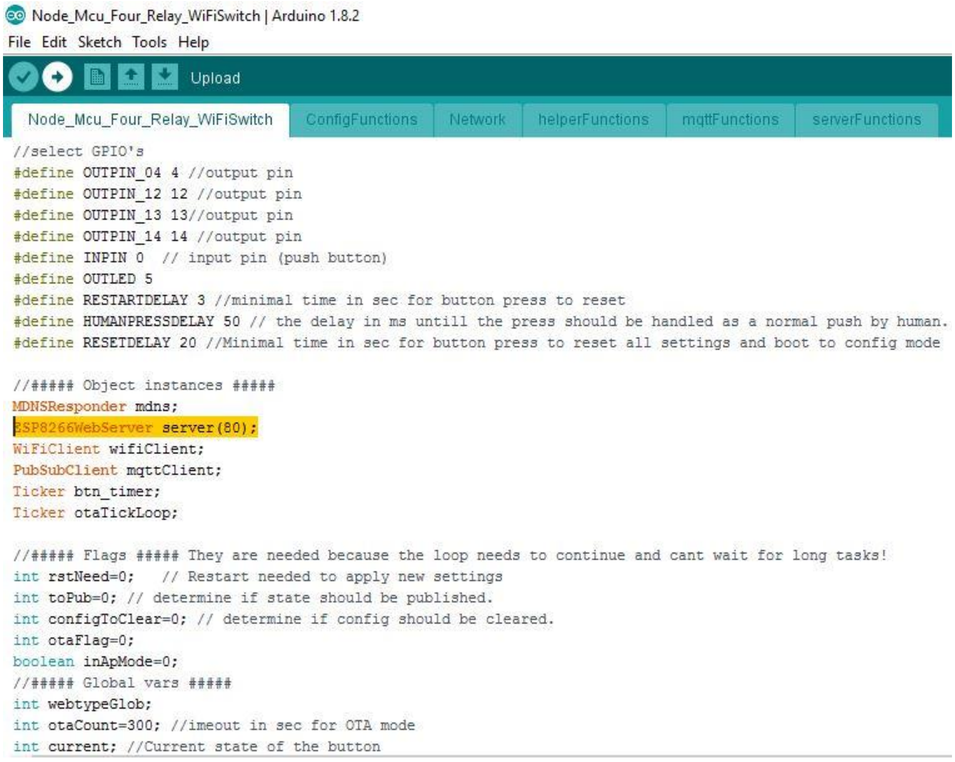
Wiring

The wiring diagram is shown in the diagram3 to input terminal block 230VAC Phase(P) and Neutral(N) is given .The output can be used as dimmer to the dimmable light to control the intensity of light and also to control the speed of fan. The output also be controlled via DC virtual switch as shown in the diagram3 Gpio A4,A5 of second and third pin of J3 header of atmega is used for virtual switch and J3 header fourth pin Ground is also used to connect virtual switch. For best dimming output use 10K pot.

Some images which may help you


2-Holes are Input side (AC 230v)
P
Armtronix
L
IN
OUT
N
N
L
3-Holes are Output side (LOAD)
Output LED
Match this position




Node_Mcu_Four_Relay_WiFiSwitch | Arduino 1.8.2
File Edit Sketch Tools Help
Upload
Node_Mcu_Four_Relay_WiFiSwitch
ConfigFunctions
Network
helperFunctions
mqttFunctions
serverFunctions

```
//select GPIO's
#define OUTPIN_04 4 //output pin
#define OUTPIN_12 12 //output pin
#define OUTPIN_13 13//output pin
#define OUTPIN_14 14 //output pin
#define INPIN 0  // input pin (push button)
#define OUTLED 5
#define RESTARTDELAY 3 //minimal time in sec for button press to reset
#define HUMANPRESSDELAY 50 // the delay in ms untill the press should be handled as a normal push by human.
#define RESETDELAY 20 //Minimal time in sec for button press to reset all settings and boot to config mode

//##### Object instances #####
MDNSResponder mdns;
ESP8266WebServer server(80);
WiFiClient wifiClient;
PubSubClient mqttClient;
Ticker btn_timer;
Ticker otaTickLoop;

//##### Flags ##### They are needed because the loop needs to continue and cant wait for long tasks!
int rstNeed=0;   // Restart needed to apply new settings
int toPub=0; // determine if state should be published.
int configToClear=0; // determine if config should be cleared.
int otaFlag=0;
boolean inApMode=0;
//##### Global vars #####
int webtypeGlob;
int otaCount=300; //imeout in sec for OTA mode
int current; //Current state of the button
```

Safety Warnings

If you are considering buying this item, you probably already know all this but in the interest of your safety, we feel compelled to state all this clearly. So take a few minutes to read it carefully before buying.

AC mains are very dangerous -- Even a 50 V AC supply is more than enough to kill you.

Please switch off the mains before making or changing connections, be very very careful. If you are not sure of anything related to the Ac supply lines please call an electrician ask him to help you out 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 when you are alone by yourself. 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 1A 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 go through 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.