Wifi-Control-Drone-V2

controlling the Drone using mobile with wifi

It consists of two versions "CODE FREE" and "ORIGINAL".

ORIGINAL

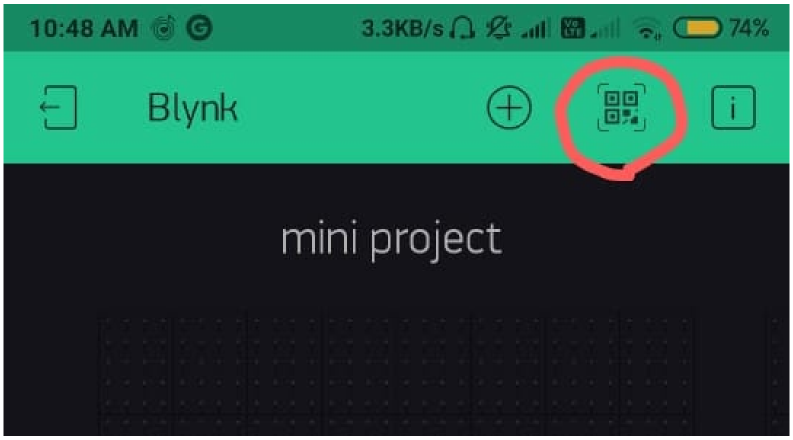
If you have experience with blynk then check out the "wifi_control_v2" folder in which you need to manually add the auth_token ,wifi_name and wifi_password.

CODE FREE

If you are new to blynk then upload then open the” Code_free_wifi_controller” folder and directly upload the code to the node_mcu or wemos_d1_mini.

CONFIGURING THE BLYNK APP:

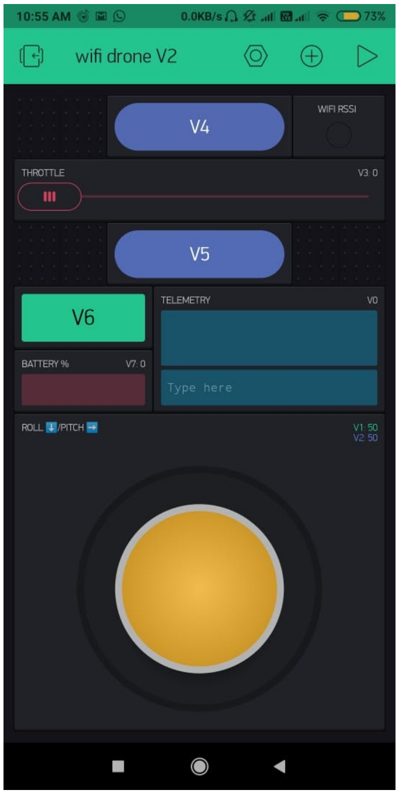
- Download the app:https://play.google.com/store/apps/details?id=cc.blynk&hl=en_IN
- Create new account or login existing account if you already have
- You will get this screen



- Click on the QR scan icon in top right

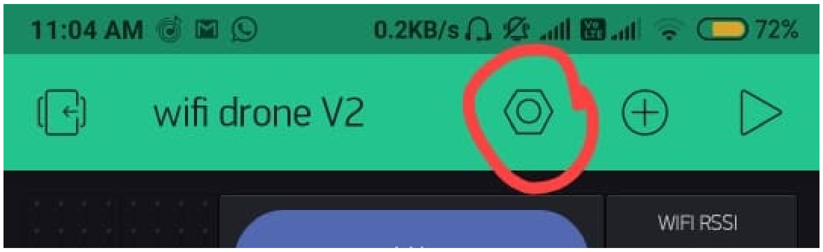
- Scan the QR

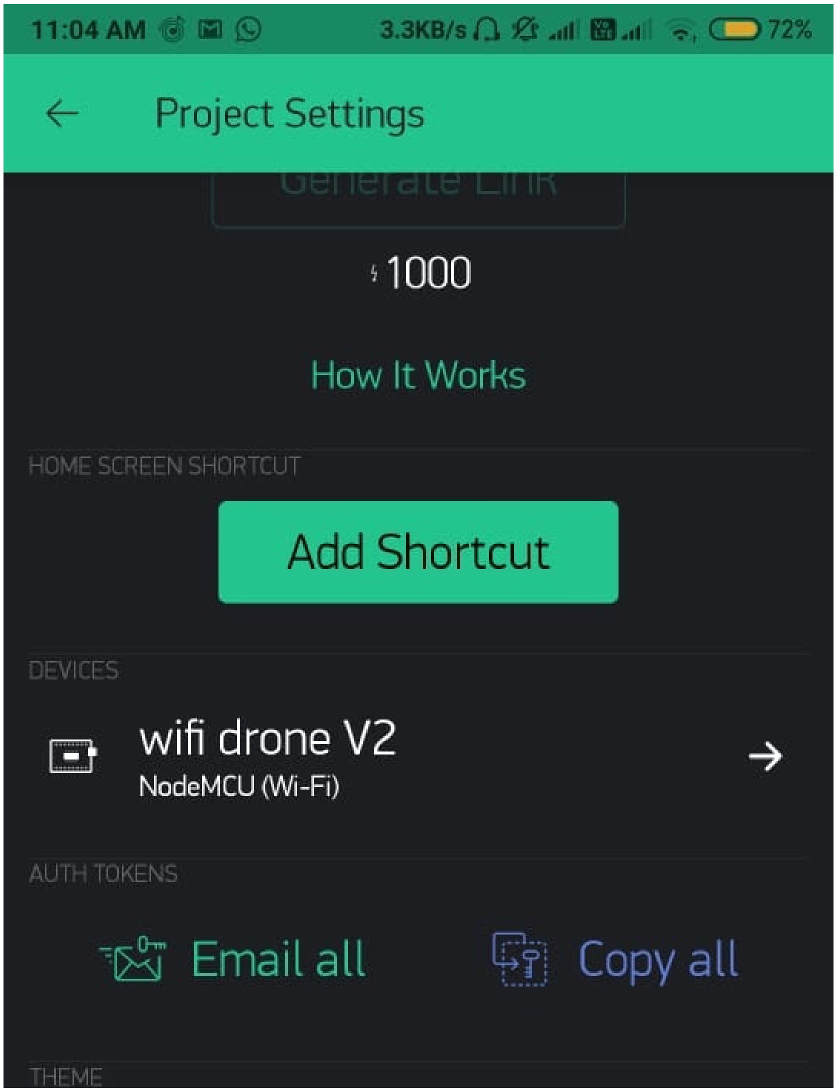
- You get all the widget in the screen



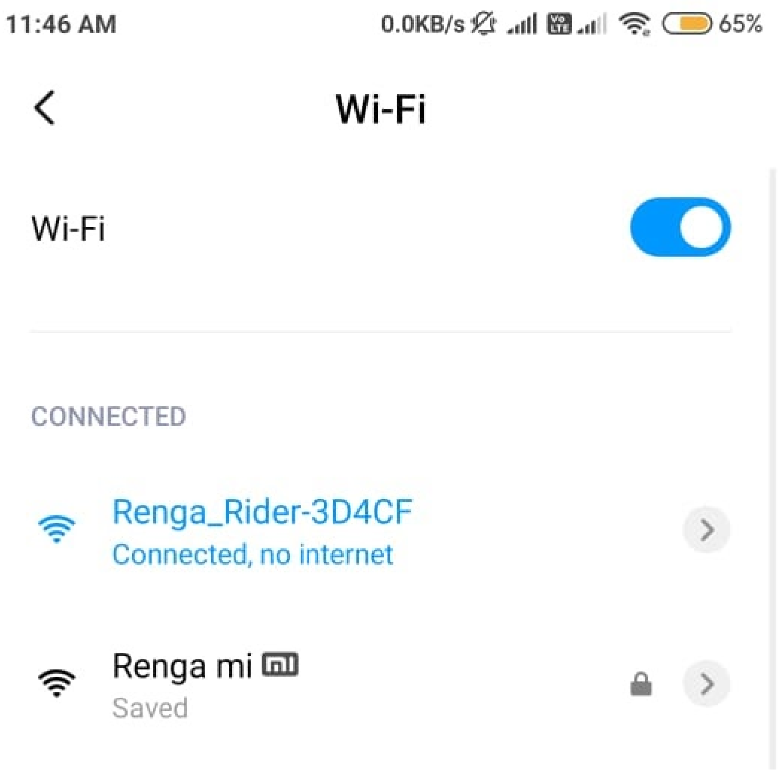
ADDING WIFI CREDENTIAL IN CODE FREE VERSION

Go to blynk setting and copy the auth code





After uploading the code free version you will notice the wifi access point named with “**Renga_Rider XXXXX**”



11:46 AM 0.0KB/s 65%

Wi-Fi

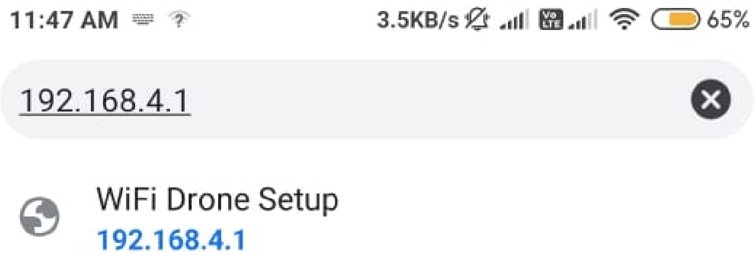
Wi-Fi

CONNECTED

Renga_Rider-3D4CF
Connected, no internet

Renga mi
Saved

Connect to that Wifi and open the browser and search**”192.168.4.1”**



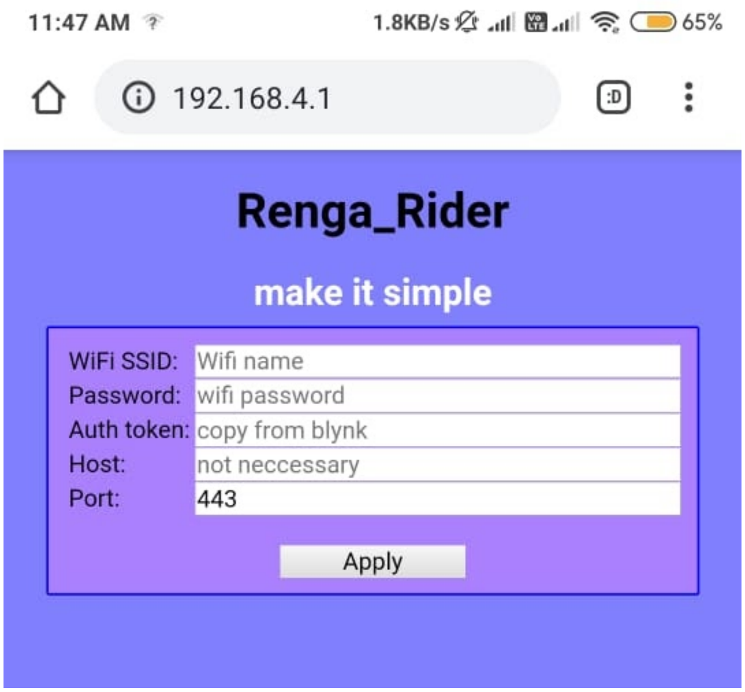
11:47 AM 3.5KB/s 65%

192.168.4.1

WiFi Drone Setup
192.168.4.1

Now u will redirected to wifi setting page

- Enter **wifi name** which drone needs to be connected(ie. Mobile Hotspot name)
- Enter **wifi password** ,leave it empty if it is open wifi
- Paste the **auth token** which you copied earlier
- Click **Apply**



11:47 AM 1.8KB/s 65%

192.168.4.1

Renga_Rider

make it simple

WiFi SSID: Wifi name
Password: wifi password
Auth token: copy from blynk
Host: not neccessary
Port: 443

Apply

Wait for few seconds as the board will automatically connect the network provided by you.

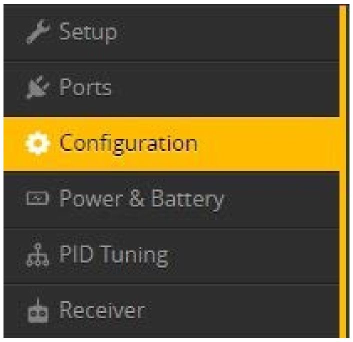
Now click the Play **Icon** in blynk app and you will find the drone online.

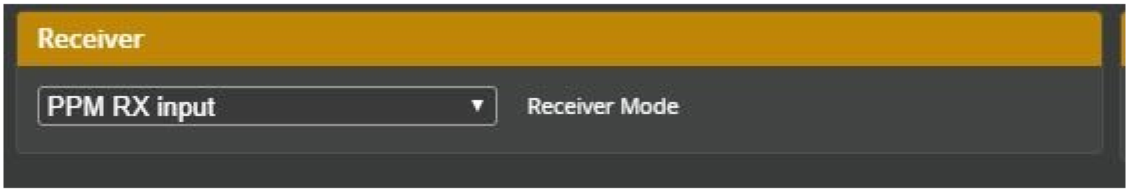
Connecting to flight controller

- Connect **5v** from flight controller to **vin** in node mcu
- Connect **gnd** to **gnd**
- Connect **ppm inpu**t to **D1** in node mcu

Configuring in beta flight

- Connect the flight controller to the laptop
- Give connect in beta flight
- Go bottom in configuration tab and make sure the **receiver is in ppm mode**

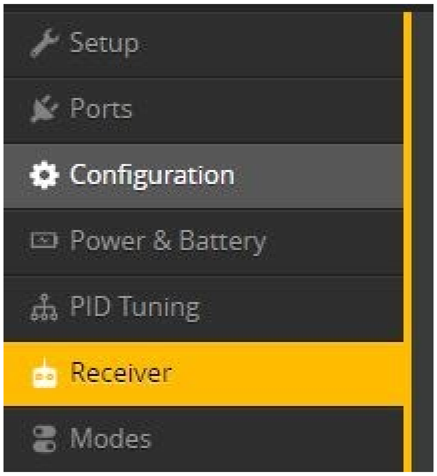




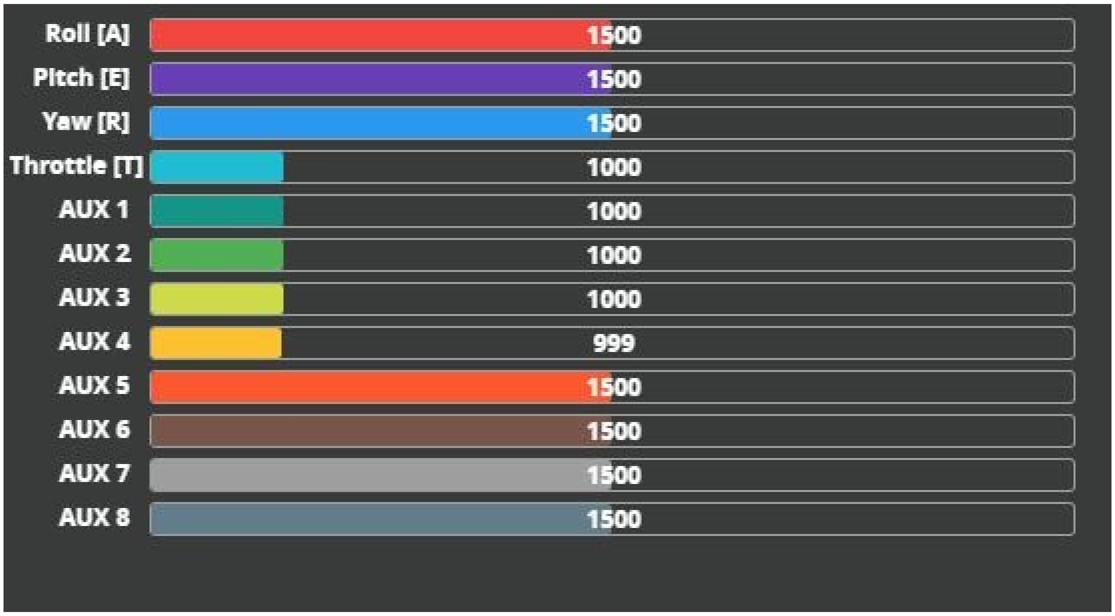
Receiver

PPM RX input ▾ Receiver Mode

- Now open the receiver tab

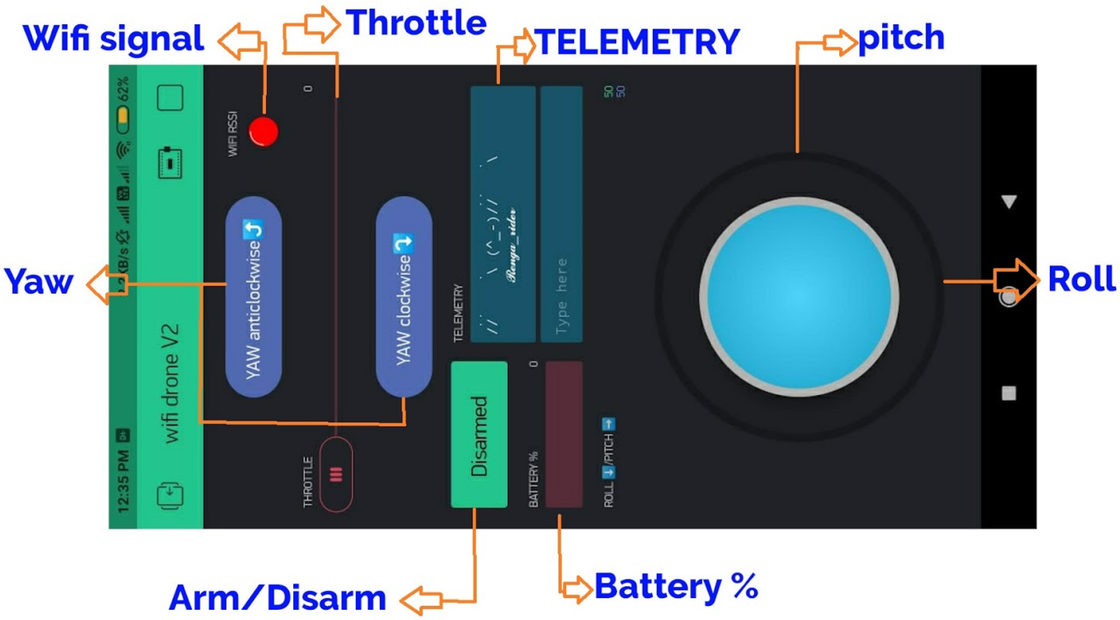


- Now check all the channels are working perfectly



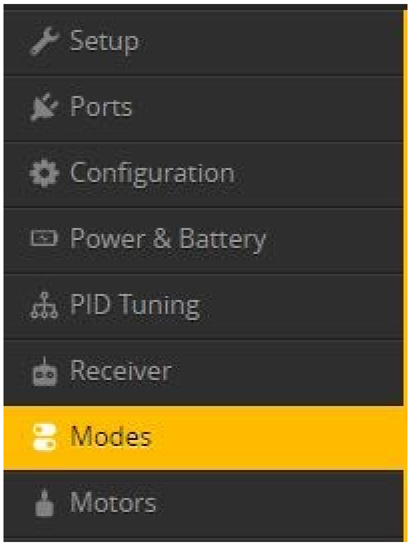
Roll [A]	1500
Pitch [E]	1500
Yaw [R]	1500
Throttle [T]	1000
AUX 1	1000
AUX 2	1000
AUX 3	1000
AUX 4	999
AUX 5	1500
AUX 6	1500
AUX 7	1500
AUX 8	1500

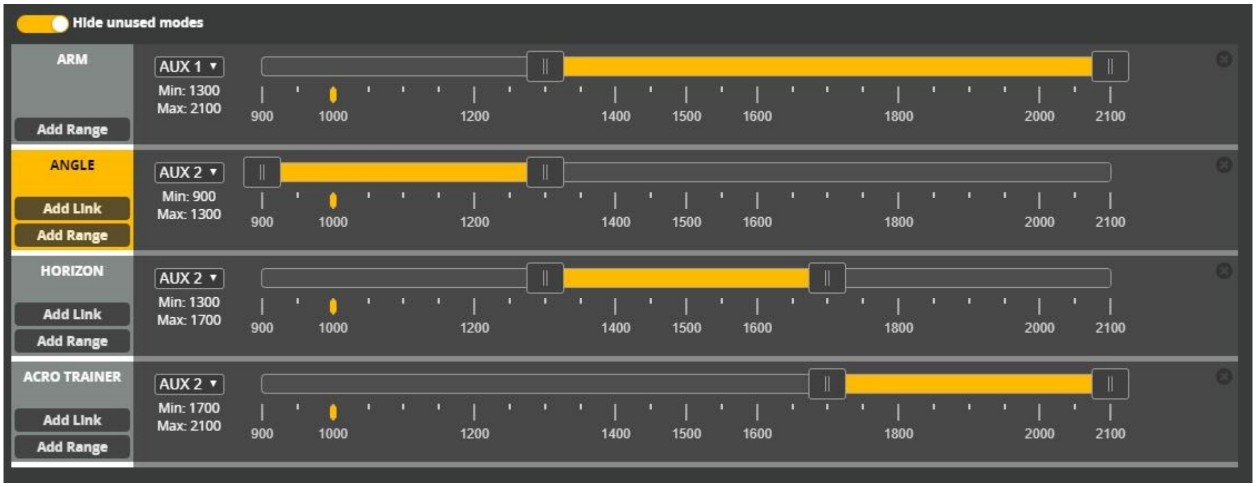
Make sure that pitch,roll,yaw and throttle and aux1 works properly.



Choosing mode

Go to mode tab and add this range for mode




Hide unused modes

ARM
AUX 1
Min: 1300
Max: 2100

ANGLE
AUX 2
Min: 900
Max: 1300

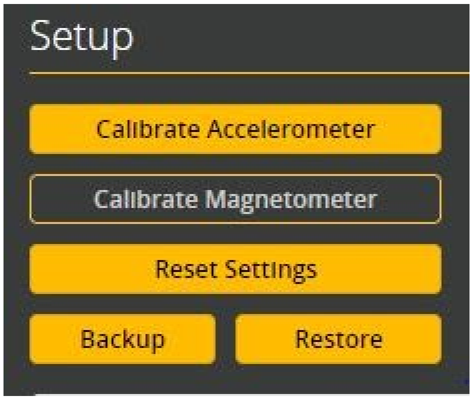
HORIZON
AUX 2
Min: 1300
Max: 1700

ACRO TRAINER
AUX 2
Min: 1700
Max: 2100

Click **save**

Calibrating

- Now go to **setup tab**
- Keep the drone in the perfectly flat surface and give calibrate accelerometer
- Wait for few seconds



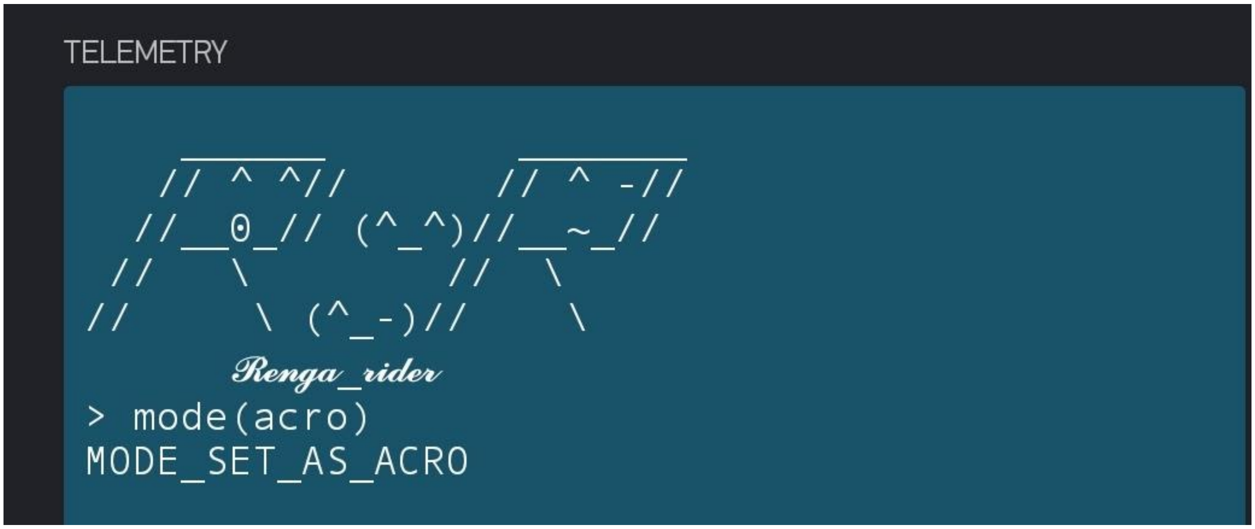
Ready to fly

- Connect the battery
- Click **disarmed** button
- Motor will start rotate
- Increase the throttle the drone will start to fly

Telemetry command

Mode selection:

For selecting the mode the command will **mode(type)** ie **mode(acro)**
Total three modes acro,angle & horizon,**default mode is angle**.

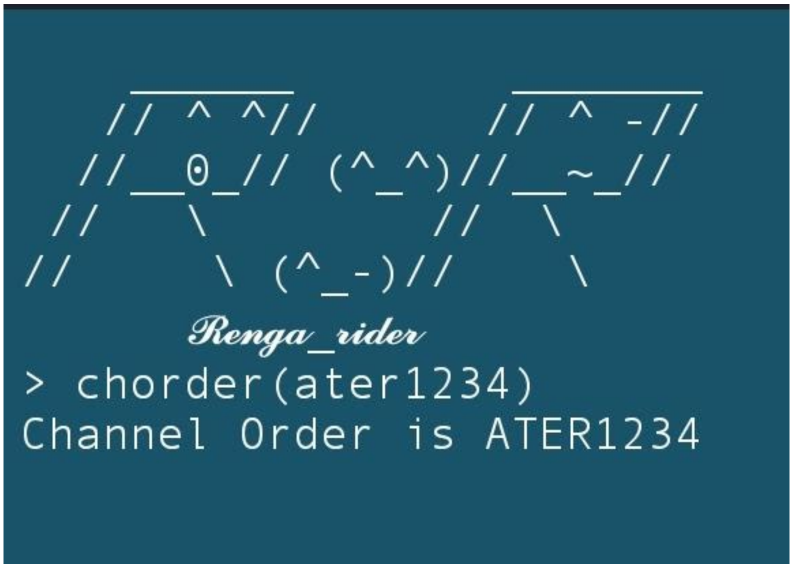


```
TELEMETRY

    ______        _______
   // ^ ^//      // ^ -//
  //__0_// (^_^)//__~_//
 //   \        //  \
//     \ (^_-)//    \
      Renga_rider
> mode(acro)
MODE_SET_AS_ACRO
```

DEBUG FUNCTIONS

- **CHANGING CHANNEL ORDER:**
  Incase if you are using some other flight controller (ie pixhawk ,apm) we can't change the channel order in the flight controller.so we can change using blynk app command **chorder("order")** ie as **chorder(ATER1234)**
  The channel order will be fixed as ATER1234
  By **default** the order is **AETR1234**(Roll,Pitch,Throttle,Yaw,Aux1,Aux2,Aux3,Aux4)



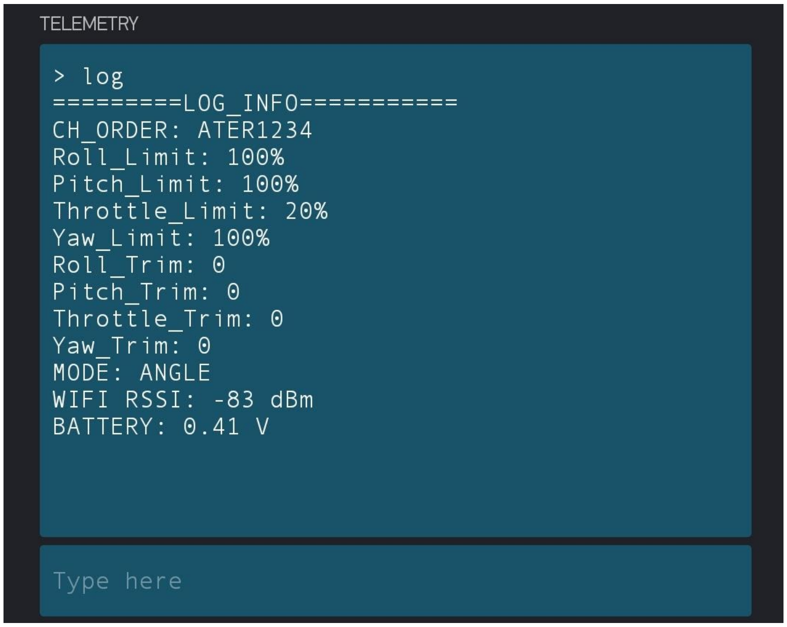
```
    ______        _______
   // ^ ^//      // ^ -//
  //__0_// (^_^)//__~_//
 //   \        //  \
//     \ (^_-)//    \
      Renga_rider
> chorder(ater1234)
Channel Order is ATER1234
```

Commands are **not case sensitive**

- **LOG INFO:**

  Getting all the information about our drone .

  Command **log**



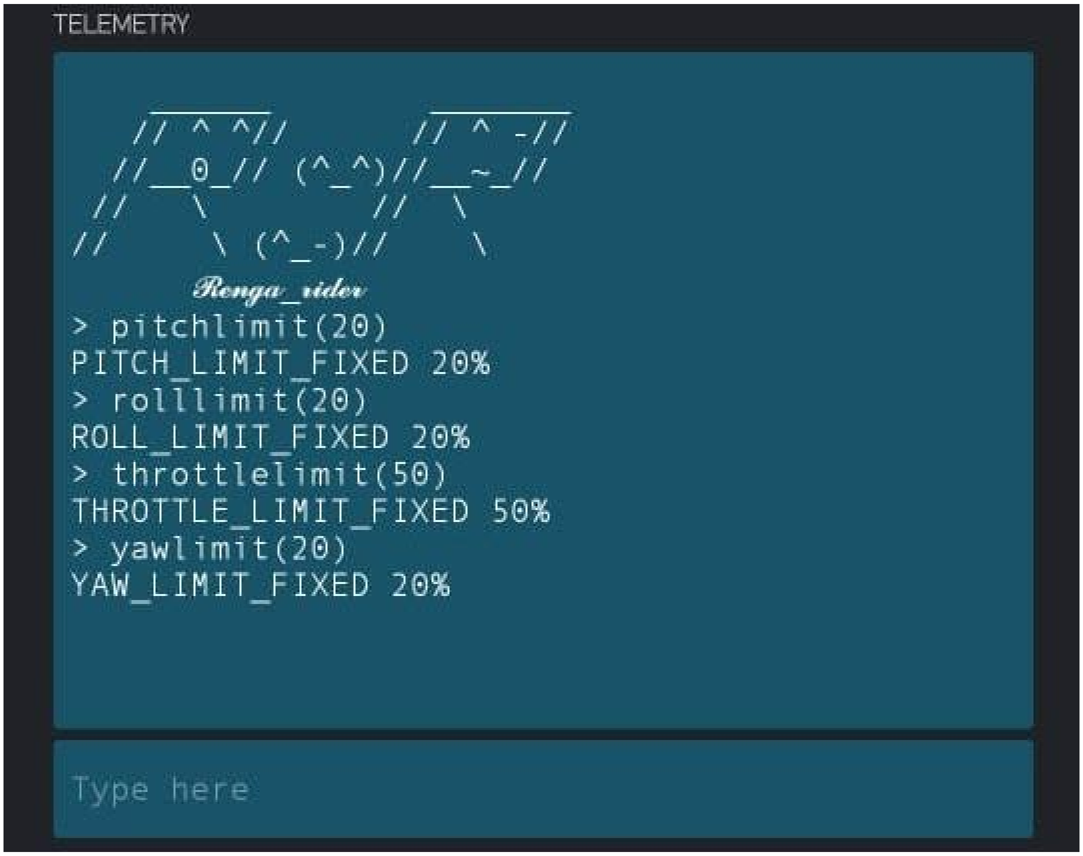
```
> log
=========LOG_INFO===========
CH_ORDER: ATER1234
Roll_Limit: 100%
Pitch_Limit: 100%
Throttle_Limit: 20%
Yaw_Limit: 100%
Roll_Trim: 0
Pitch_Trim: 0
Throttle_Trim: 0
Yaw_Trim: 0
MODE: ANGLE
WIFI RSSI: -83 dBm
BATTERY: 0.41 V
```

- **LIMITING THE CHANNEL**

  In case you want to fly at low height , you want to limit the throttle to 50%(1500).

  Command **tlimit(range)** or **throttlelimit(range) ,**range-**10 to100**,ie **tlimit(20)** in full throttle the output will be only **1200**

  Similarly you can fix for all channels

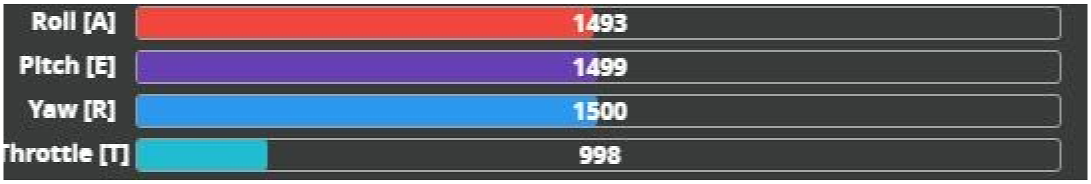


TELEMETRY

```
   _______        _______
   // ^ ^//      // ^ -//
  //__0_// (^_^)//__~_//
 //   \        //   \
//     \ (^_-)//     \
      Renga_rider
> pitchlimit(20)
PITCH_LIMIT_FIXED 20%
> rolllimit(20)
ROLL_LIMIT_FIXED 20%
> throttlelimit(50)
THROTTLE_LIMIT_FIXED 50%
> yawlimit(20)
YAW_LIMIT_FIXED 20%
```

- **CHANNEL TRIM**

  In drone transmitter we have trim buttons for making the output value exactly to mid(1500)



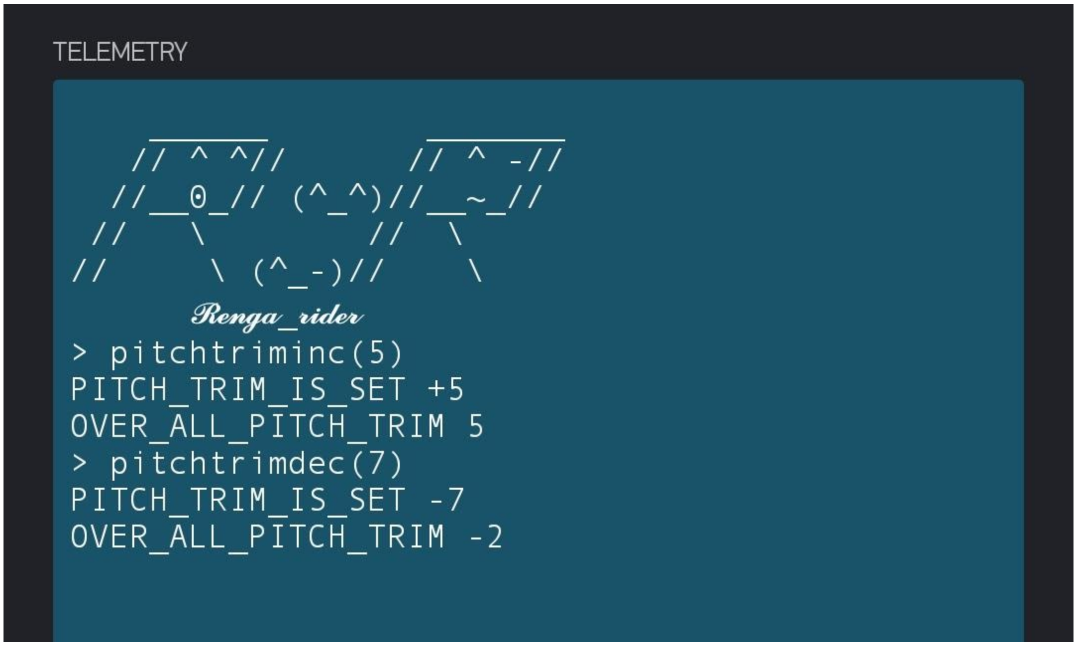
```
Roll [A]      1493
Pitch [E]     1499
Yaw [R]       1500
Throttle [T]  998
```

In above case the Roll is **not at 1500** which will cause the drone to **drift in left side** after take off

Command **rolltriminc(value)** ie in above case **rolltriminc(7)**

If the channel is above 1500 so we need to decrease. In that case the command will be **rolltrimdec(value)** ie **rolltrimdec(2)**

Similarly you can use for all channels (roll,pitch,throttle and yaw)

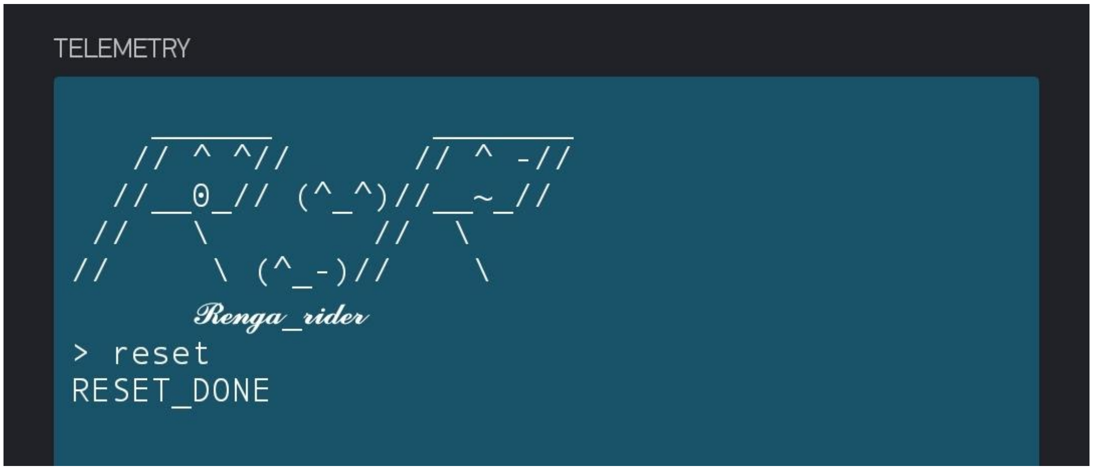


TELEMETRY

```
    ______        _______
   // ^ ^//      // ^ -//
  //__0_// (^_^)//__~_//
 //   \         //  \
//      \ (^_-)//     \
        Renga_rider
> pitchtriminc(5)
PITCH_TRIM_IS_SET +5
OVER_ALL_PITCH_TRIM 5
> pitchtrimdec(7)
PITCH_TRIM_IS_SET -7
OVER_ALL_PITCH_TRIM -2
```

- **RESET**

  All the **data**(chorder,trim & limit) will be **stored in cloud** ,incase if you want to set trim back to zero , limit to 100% and chorder to AETR1234. Reset command is used Command **reset**



TELEMETRY

```
    ______        _______
   // ^ ^//      // ^ -//
  //__0_// (^_^)//__~_//
 //   \         //  \
//      \ (^_-)//     \
        Renga_rider
> reset
RESET_DONE
```

- **CLEAR**

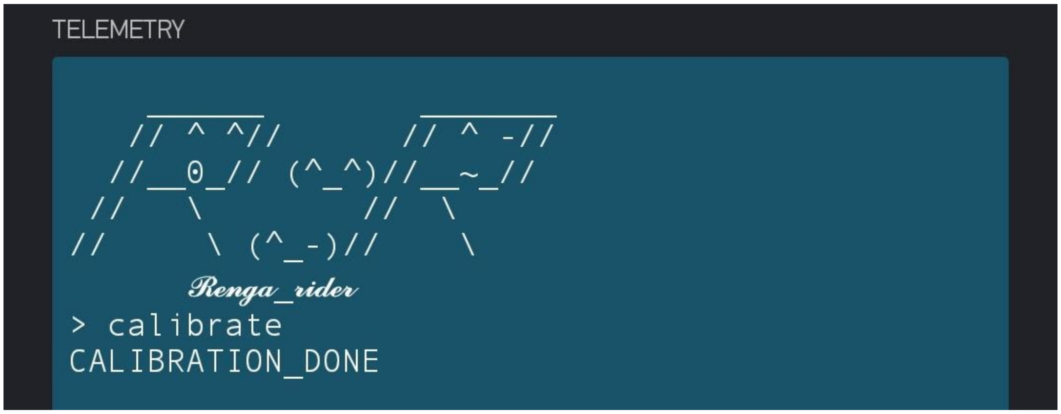
  Clearing the terminal window .

  Command **clear** ie it will clear the terminal and display the logo

- **CALIBRATION**

  After crashing the drone we need to calibrate the accelerometer again ,so everytime we can't have a laptop to calibrate. We can calibrate the accelerometer using calibration command

Command **calibration** ie keep the drone in flat surface and give the command

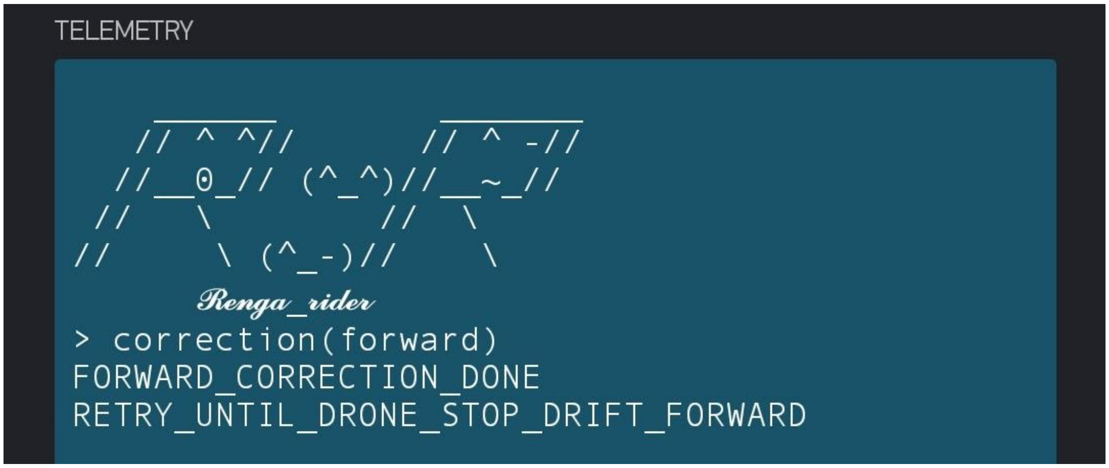


```
    ______        _______
   // ^ ^//      // ^ -//
  //___0_// (^_^)//___~_//
 //   \        //  \
//     \ (^_-)//    \
      Renga_rider
> calibrate
CALIBRATION_DONE
```

- **CORRECTION**

  Even after perfect trim and calibration the drone will tend to move in a particular direction. In that case the correction command is useful

  Command **correction(direction)** ie if the **drone drift forward** the command will be **correction(forward)**



```
    ______        _______
   // ^ ^//      // ^ -//
  //___0_// (^_^)//___~_//
 //   \        //  \
//     \ (^_-)//    \
      Renga_rider
> correction(forward)
FORWARD_CORRECTION_DONE
RETRY_UNTIL_DRONE_STOP_DRIFT_FORWARD
```

Similarly for **all direction** (forward,backward,left & right)

- Disarm the drone and give the command
- Arm again and check the direction of drift
- Repeat the steps until the drone stop drift

Check this video for better understanding (https://www.youtube.com/watch?v=otJNzZrE-yk&feature=youtu.be)

LIST OF COMMANDS

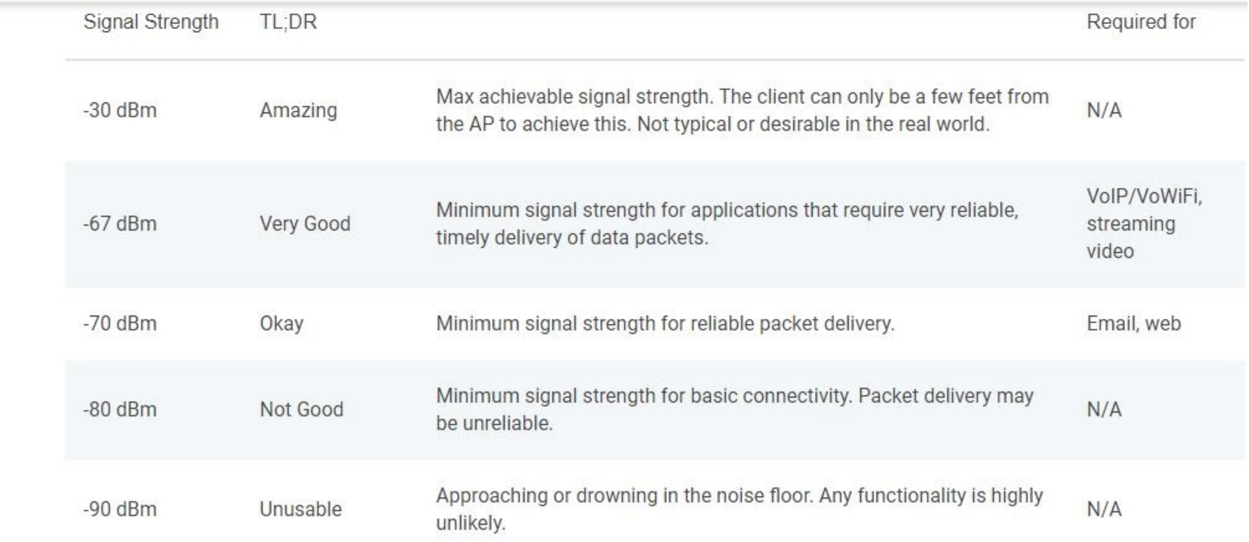
1. Log
2. Reset
3. Clear
4. Calibration
5. Chorder(order) // order->(aetr1234),(ater1234)....etc
6. Mode(type) // type->angle,acro,horizon
7. Pitch limit(value) or p limit(value) or Pitchlimit(value) or plimit(value)//value (10 to 100)
8. Roll Limit(value) or r limit(value) or RollLimit(value) or rlimit(value) //value (10 to 100)
9. Yaw Limit(value) or y limit(value) or YawLimit(value) or ylimit(value) //value (10 to 100)
10. Throttle limit(value) or t limit(value) or Throttlelimit(value) or tlimit(value) //value (10 to 100)
11. Roll triminc(value) or r triminc(value) or rolltriminc(value) or rtriminc(value)//value- 1 to 99
12. Roll trimdec(value) or r trimdec(value) or rolltrimdec(value) or rtrimdec(value) //value- 1 to 99
13. Pitch triminc(value) or p triminc(value) or pitchtriminc(value) or ptriminc(value) //value- 1 to 99
14. Pitch trimdec(value) or p trimdec(value) or pitchtrimdec(value) or ptrimdec(value) //value- 1 to 99
15. Throttle triminc(value) or t triminc(value) or throttletriminc(value) or ttriminc(value)//value- 1 to 99
16. Throttle trimdec(value) or t trimdec(value) or throttleltrimdec(value) or ttrimdec(value) //value- 1 to 99
17. Yaw triminc(value) or y triminc(value) or yawltriminc(value) or ytriminc(value) //value- 1 to 99
18. Yaw trimdec(value) or y trimdec(value) or yawtrimdec(value) or ytrimdec(value) //value- 1 to 99

ALL COMMANDS ARE NOT CASE SENSITIVE !!!!!

WIFI RSSI:

Wifi signal strength indicator ,incase if you are flying out ranging it will be indicated

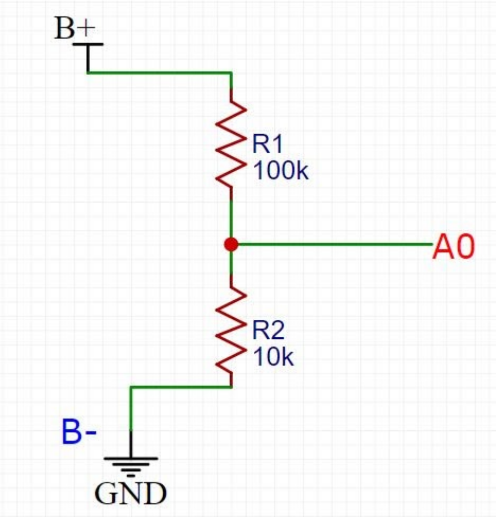
- Green-Amazing
- Yellow-Very good
- Orange-Okay
- Red-Not good



Signal Strength	TL;DR		Required for
-30 dBm	Amazing	Max achievable signal strength. The client can only be a few feet from the AP to achieve this. Not typical or desirable in the real world.	N/A
-67 dBm	Very Good	Minimum signal strength for applications that require very reliable, timely delivery of data packets.	VoIP/VoWiFi, streaming video
-70 dBm	Okay	Minimum signal strength for reliable packet delivery.	Email, web
-80 dBm	Not Good	Minimum signal strength for basic connectivity. Packet delivery may be unreliable.	N/A
-90 dBm	Unusable	Approaching or drowning in the noise floor. Any functionality is highly unlikely.	N/A

BATTERY PERCENTAGE :

Connect voltage divider of **100k and 10k** between the battery and connect the output to **A0** in node mcu

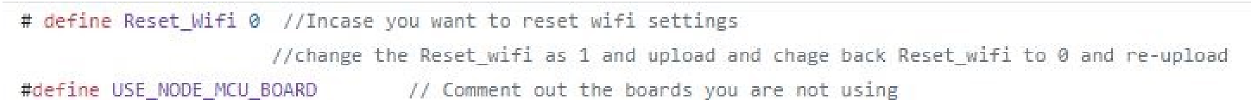


If you are using the esp8266 12e standalone then change the **float ADC** value in the program to **1.0,**the maximum voltage the esp8266 12e standalone ADC accept is 1V ,in case of node mcu it will be 3.3V by default .

RESETTING THE WIFI CREDENTIALS IN CODE FREE VERSION

Wifi ssid,wifi password & auth token will be stored in **EEPROM** even reuploading of the program will not delete it .

For deleting the previous access



```
# define Reset_Wifi 0  //Incase you want to reset wifi settings
                     //change the Reset_wifi as 1 and upload and chage back Reset_wifi to 0 and re-upload
#define USE_NODE_MCU_BOARD        // Comment out the boards you are not using
```

- Change **Reset_wifi** as **1**
- And upload the program
- Once done uploading
- Change **Reset_wifi** back **0**
- And reupload the program
- Then you can find the **Renga_rider_xxxxx** access point
- Now enter new wifi credential