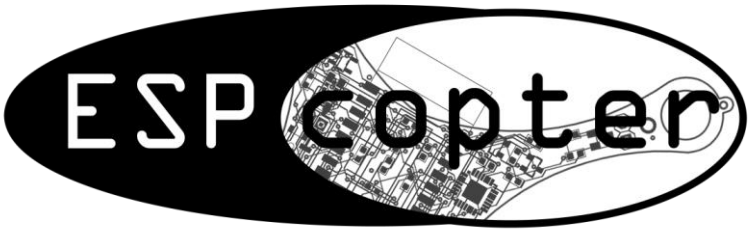


ESPcopter SDK

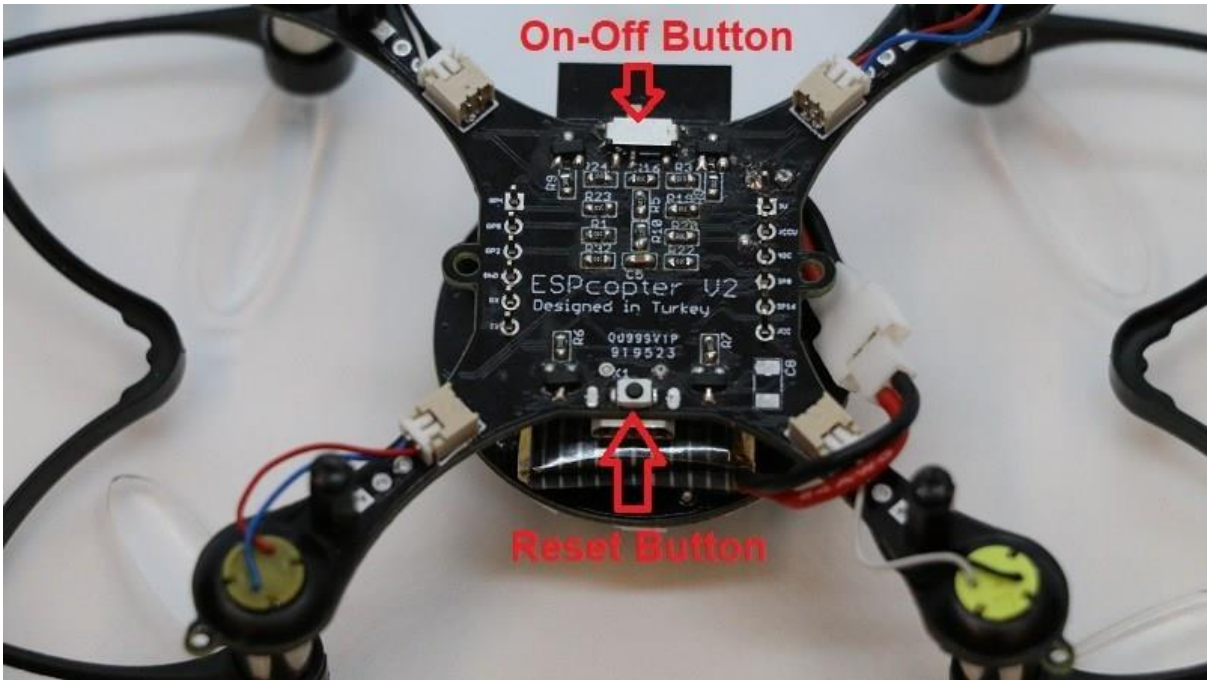
1.0.0 (Beta)

1-) General Review

1.1-) Internal Features of the ESPcopter:

1.2-) ESPcopter Switch and Button

- The button is used to reset ESPcopter MCU
- The switch is used to open and close ESPcopter



1.2-) Pinout and Propeller and Motor Directions

When installing in accordance with the letters on the propellers, the motors must be fitted according to the cable colors.

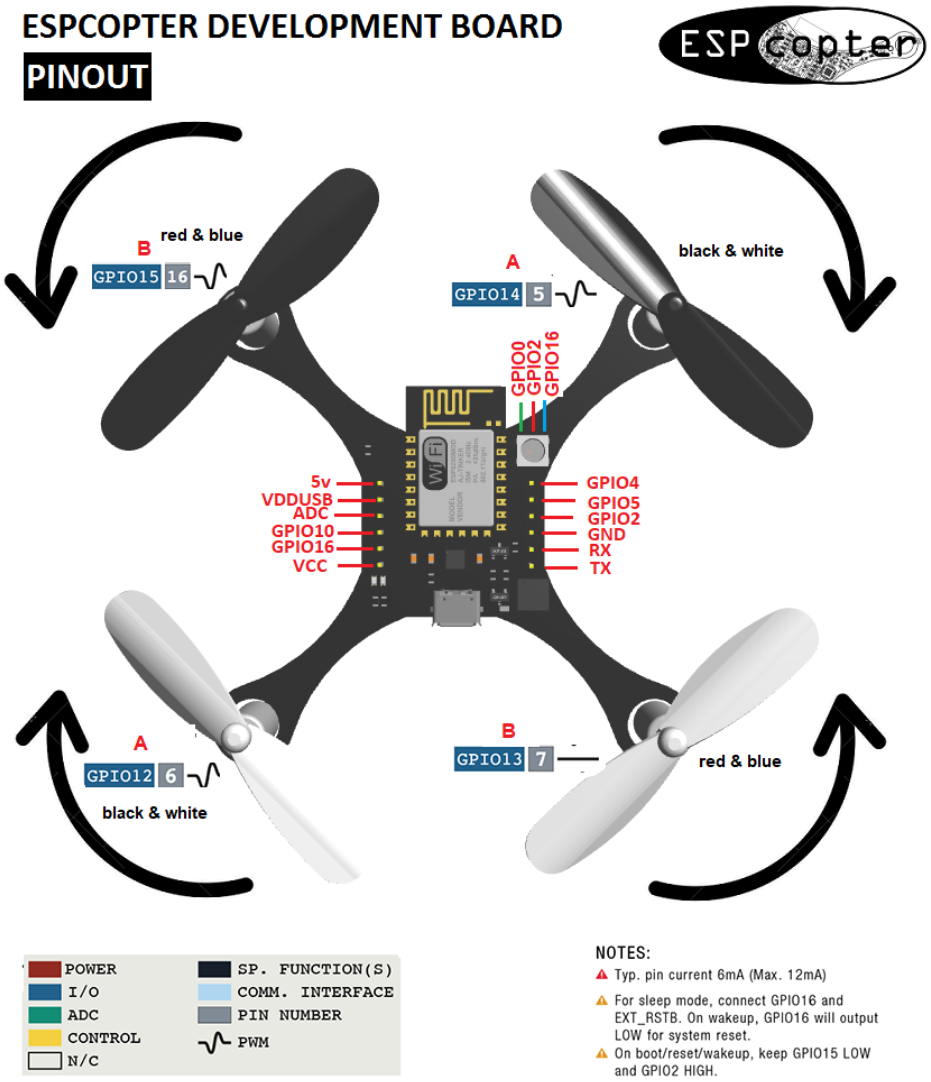
Engine positions:

Left Front: B - Red, Blue

Right Front: A - Black, White

Left Rear: A – Black, White

Right Rear: B – Red, Blue



1.3-) How to Charge the ESPcopter:

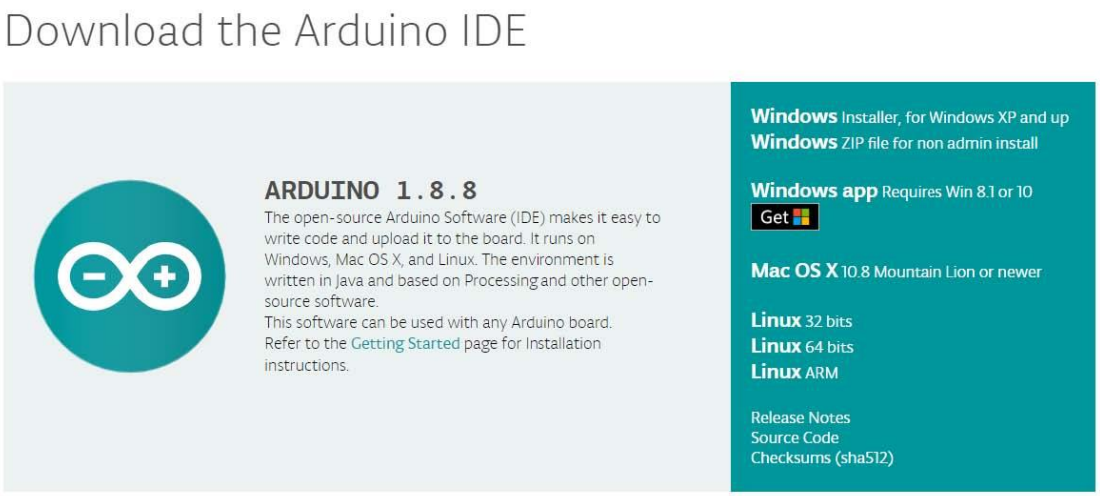
The ESPcopter will charge when connected to the micro-usb. The switch on the ESPcopter must be in the off position to charge.

Red Light: Charging
Green Light: Fully Charged

2-) Software:

2.1-) Arduino Installation:

Download and install the latest version from the Arduino web site:
https://www.arduino.cc/en/Main/Software



2.2-) Driver Installation:

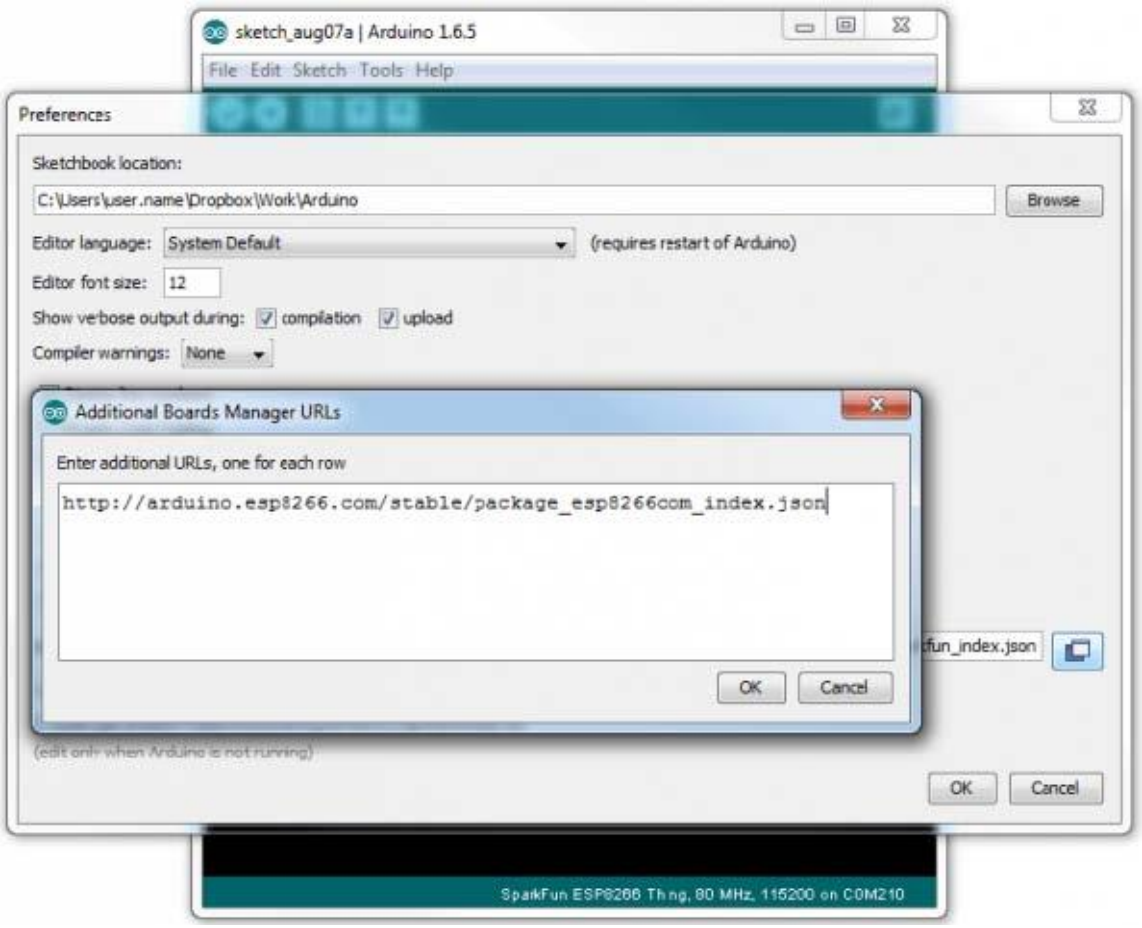
The following driver is reqired for the ESPcopter to be recognized by the computer. Download and install the approperate driver version for your OS.

https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

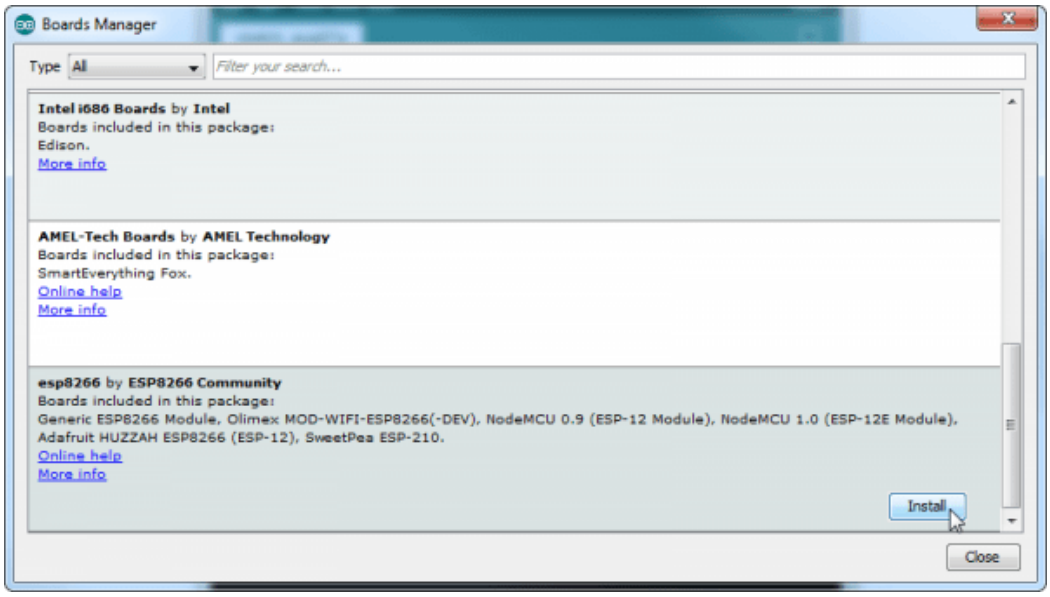
2.3-) ESP8266 Library

To begin, we'll need to update the board manager with a custom URL. Open up Arduino, then go to the Preferences (File> Preferences). Then, towards the bottom of the window, copy this
URL into the "Additional Board Manager URLs" text box:

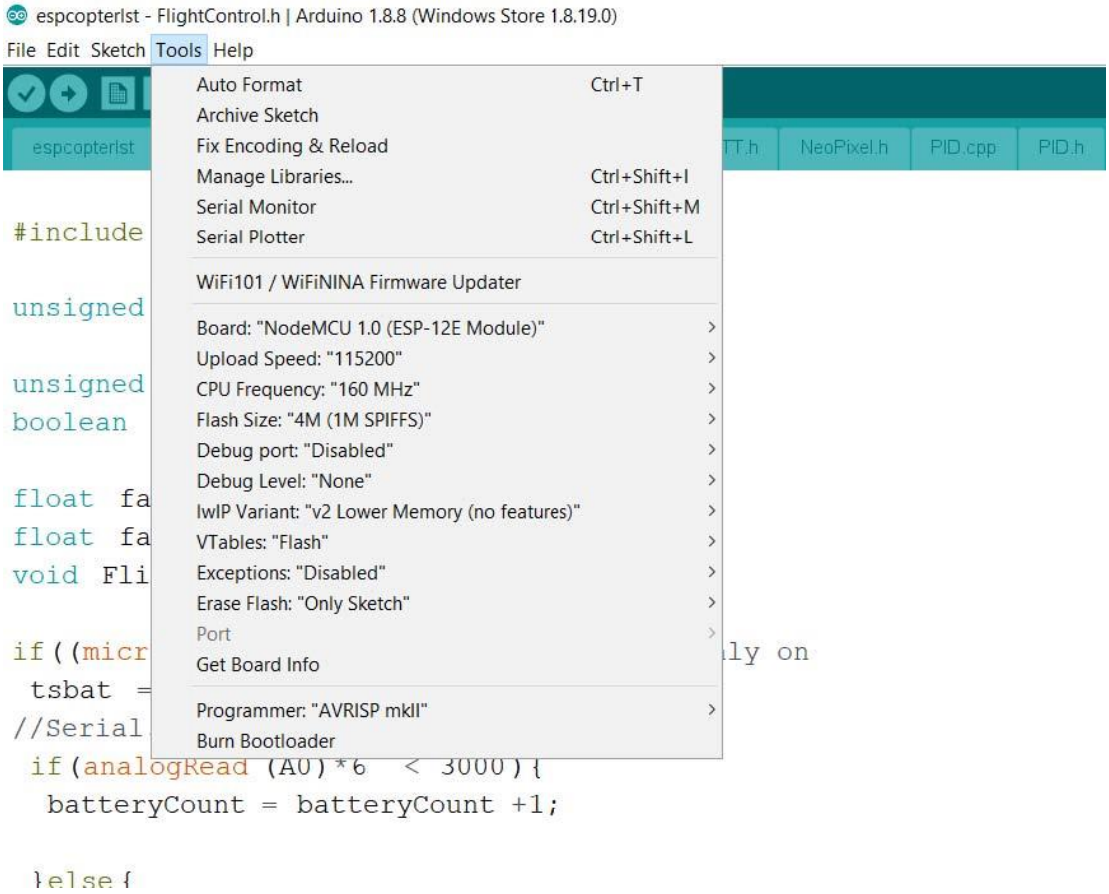
http://arduino.esp8266.com/stable/package_esp8266com_index.json



Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Look for esp8266. Click on that entry, then select Install. You need to install **2.5.0** version of esp8266 library.



The download process can take up to 10 minutes depending on your internet speed. After the download is done, select NodeMCU 1.0 from the **Tools tab** and follow the other settings.

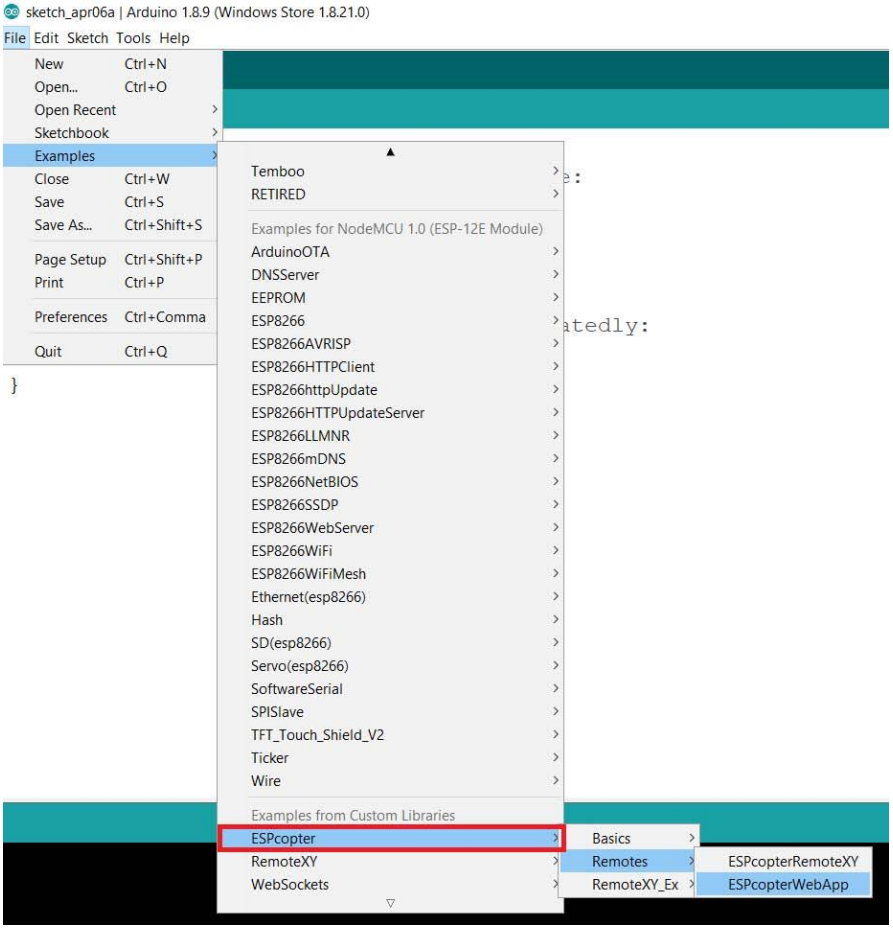


Downloading the ESPcopter library:

Before downloading the code from the website, you must use the contacts page to request source code. See the following site: **http://espcopter.com/code-release/**

After downloading, remove the zip file twice and put the file (ESPcopter "(Files> Arduino> Library) into the file. In the Examples section you will see the sample codes of the ESPcopter.

Open the following example program:



The ESPcopter will have the flight control software to make the flight stable. This ESPcopterWebApp example is a demonstration of the ESPcopter having the additional ability to act as a self-contained web server. This will get you running your ESPcopter fast because you can control it from a web page on your phone, tablet or laptop. More advanced examples (Although this is very cool) are already available from the Arduino IDE menu.
Let's hope you have had the ESPcopter switched off and the batteries charging via the USB port so you can be ready for the first flight.

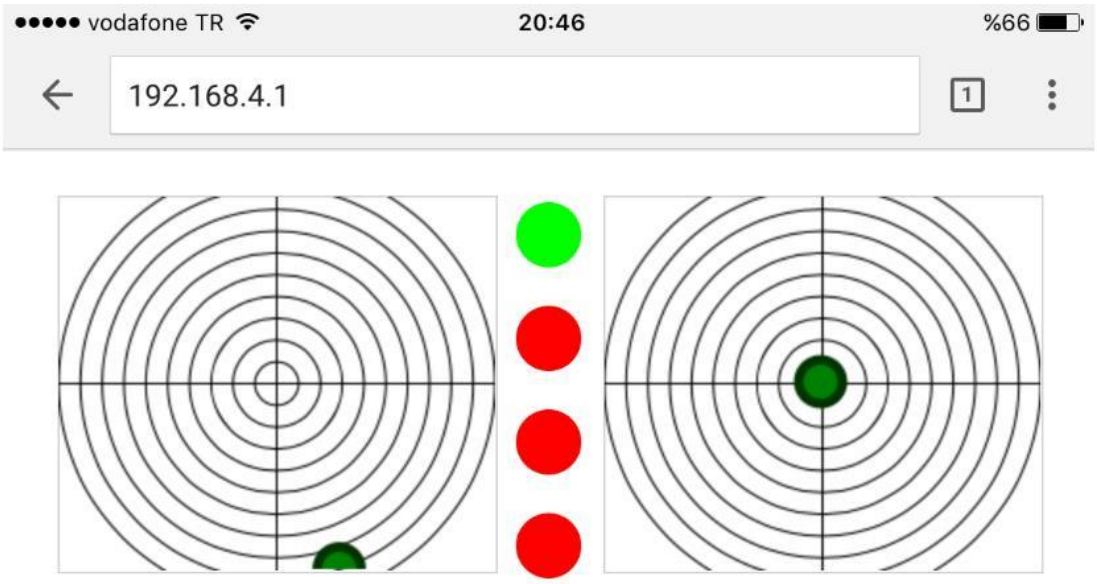
ESPcopter standalone Code(Web APP)

```
#define REMOTE_WEB_APP
#define WEB_APP_WIFI_SSID "ESPcopter"     //ESPcopter Wiffi ismi
#define WEB_APP_WIFI_PASSWORD "12345678" //ESPcopter Wiffi şifresi
#include <espcopter.h>
void setup() {
mainSetup();
}
void loop() {
mainLoop();
}
```

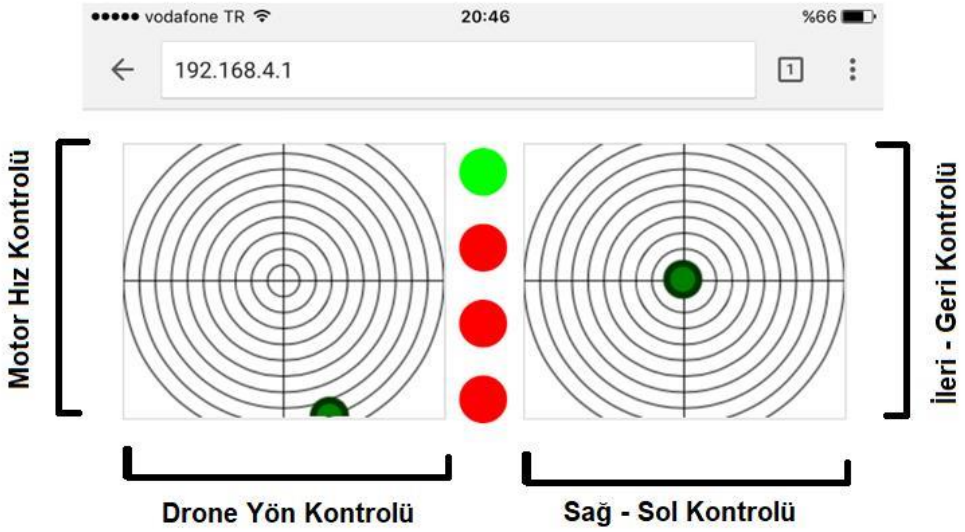
Connection:
After installing the software, turn on the drone. In the wifi tab of your phone, you will see the name of the drone.

WIFI_SSID “ESPcopter
WIFI_PASSWORD “12345678”

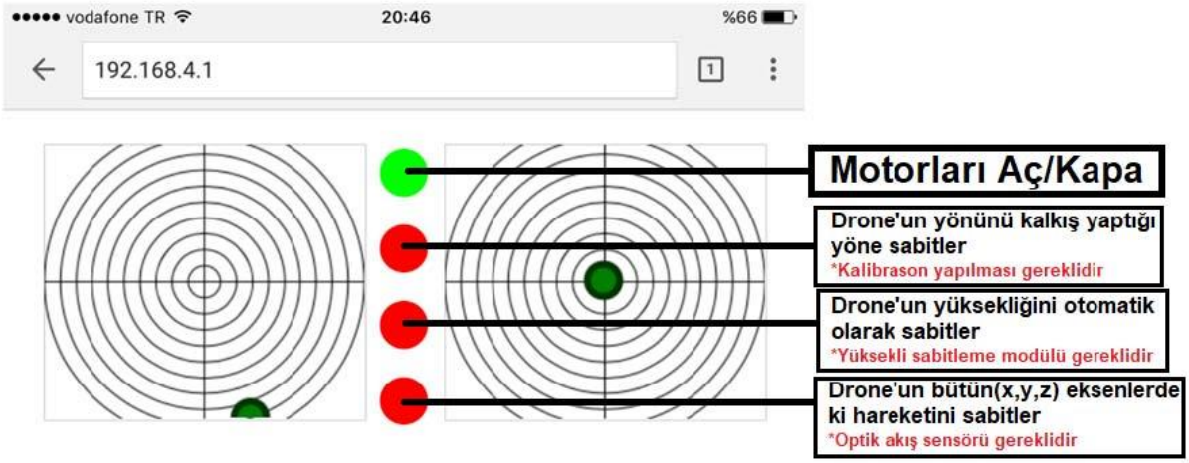
After making the connection, open any web browser application from your phone. Type 192.168.4.1 in the search line.



Control review:



Buttons:



First Button: ARM – DISARM
Second Button: AUTO YAW CONTROL
Third Button: AUTO ALTİTUDE CONTROL
Forth Button: AUTO MISSION CONTROL

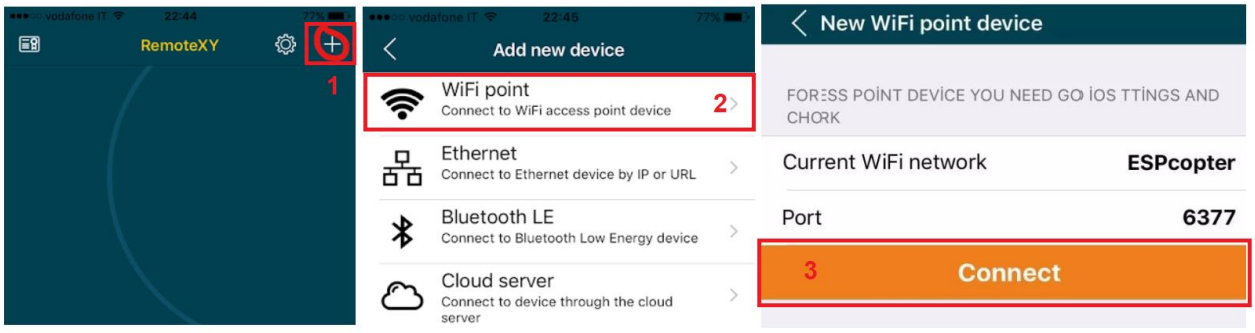
ESPcopter standalone Code (RemoteXY)

```
#define REMOTEXY_WIFI_SSID “RemoteXY”
#define REMOTEXY_WIFI_PASSWORD “12345678”
#define REMOTE_XY_REMOTE
#include <espcopter.h> // library
void setup() {
    mainSetup(); // main flying setup
}
void loop() {
mainLoop ();    // main flying loop
}
```

Connection:
After installing the software, turn on the drone. In the wifi tab of your phone, you will see the name of the drone.

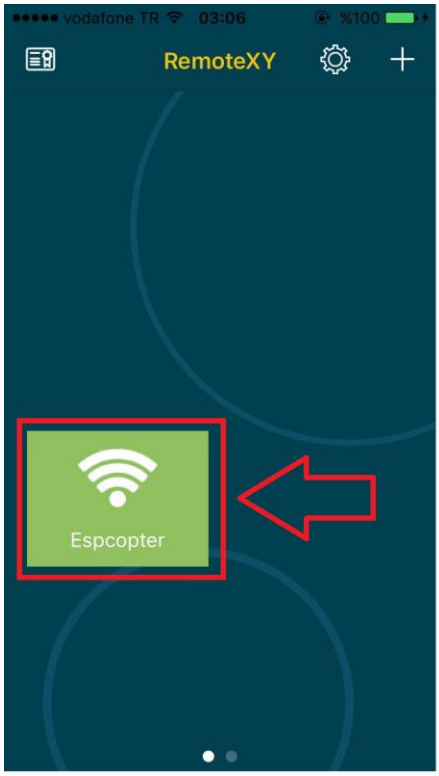
WIFI_SSID “RemoteXY
WIFI_PASSWORD “12345678”

After you make the connection, open RemoteXY from your phone.

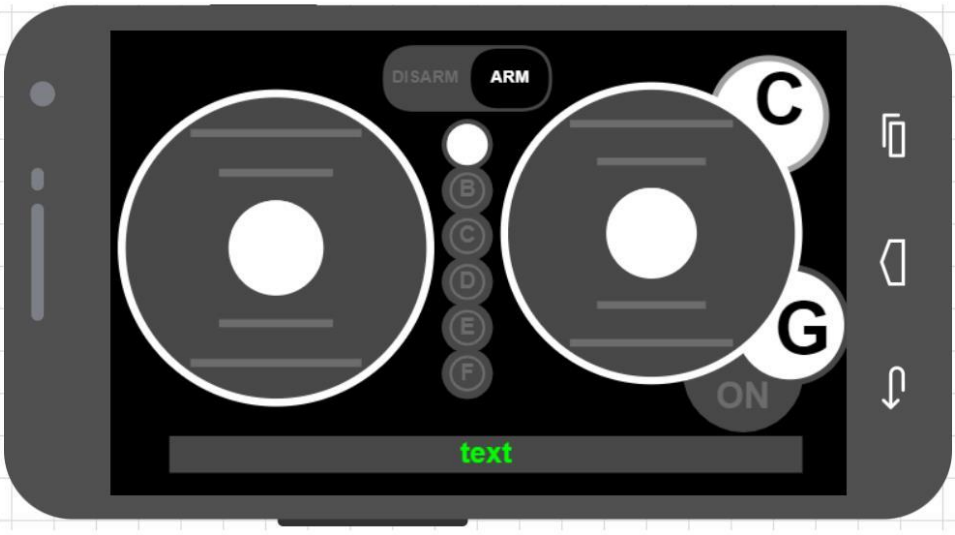


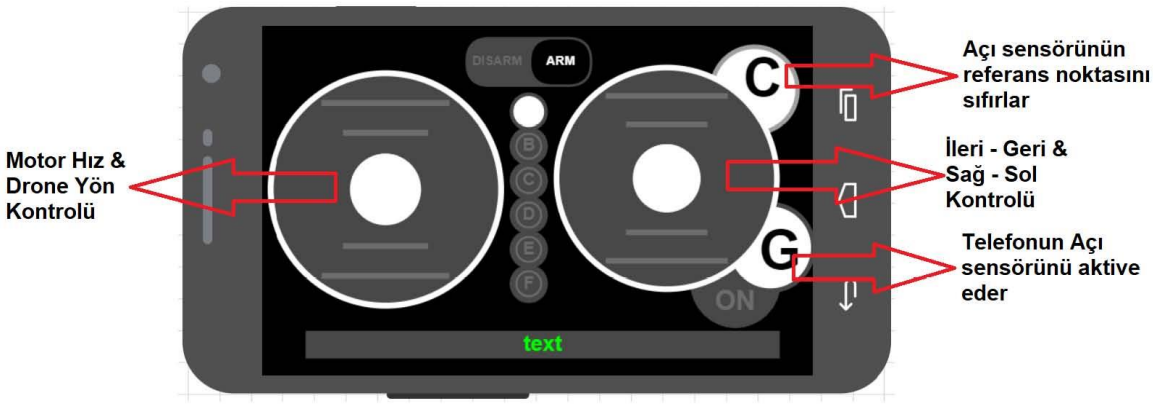
After making the connection, the screen below will open automatically.

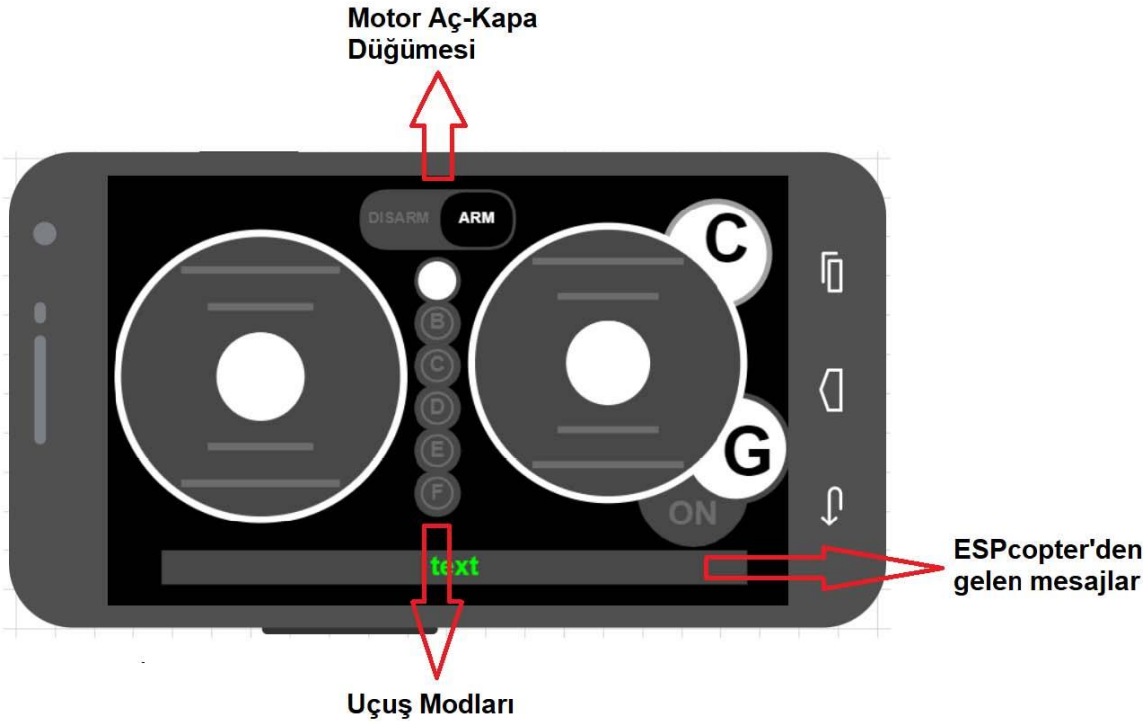
After the first connection. There will be ESPcopte box in RemoteXY app. You can connect ESPcopter by cliking this box.



Controller Review:







Başlangıç Düğmesi : Görevi yok

Drone'un yönünü kalkış yaptığı yöne sabitler : Kalibrasyon gerekli

Drone'un yüksekliğini otomatik olarak sabitler : Yükseklik sabitleme modülü gerekli

Drone'un bütün(x,y,z) eksenlerdeki hareketini sabitler : Optik akış sensörü gerekli

Mod B ve Mod C ikisi bir arada

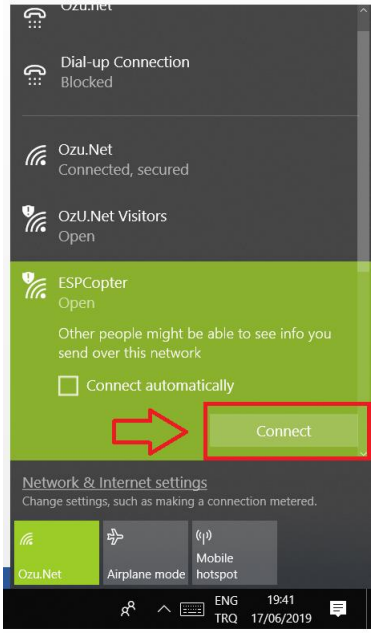
Mod B ve Mod D ikisi bir arada

Computer Control (Processing)

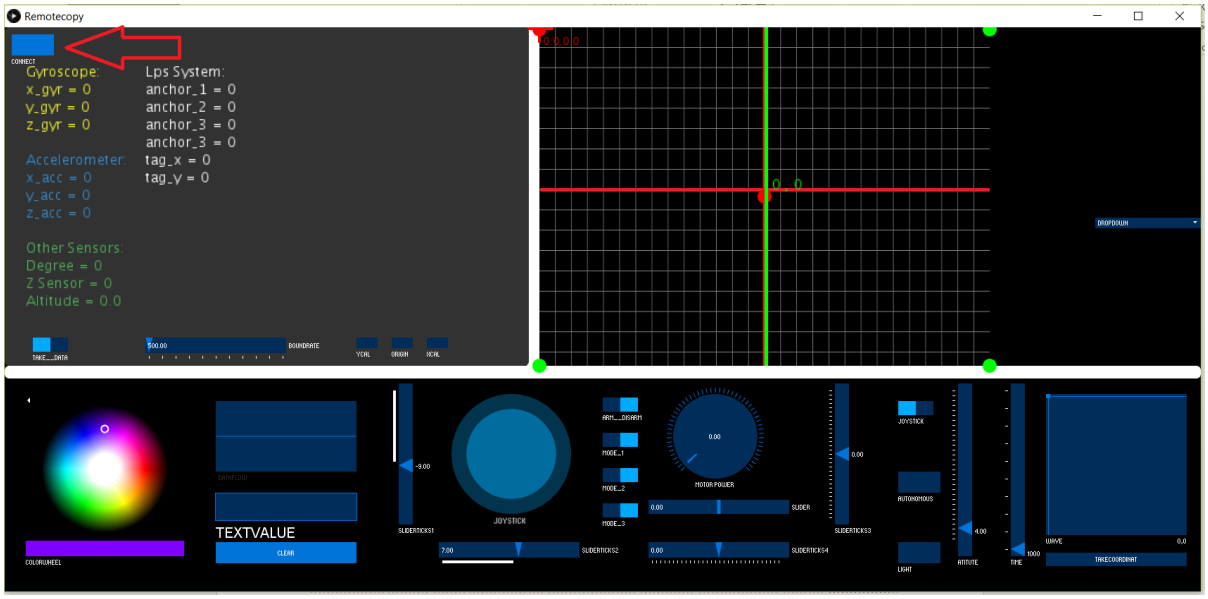
```
#define PROCESSING_REMOTE
#include <espcopter.h>
void setup() {
  mainSetup();
  setTrimRoll(0);
  setTrimPitch(0);
  setTrimYaw(0);
}
void loop() {
  mainLoop ();
}
```

Connection:

After the uploading the Processing control code to ESPcopter. Open the wiffi screen and then connect the ESPcopter



Control Screen

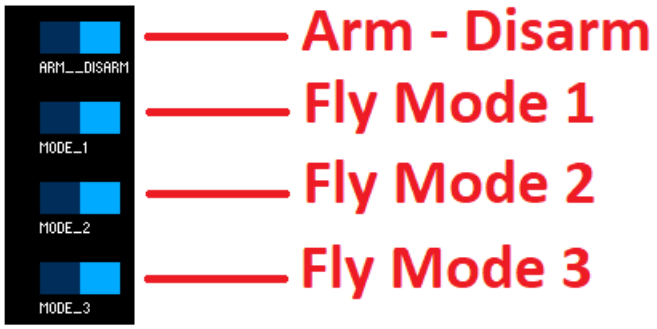


Controller Review:

1-) Control Switches:

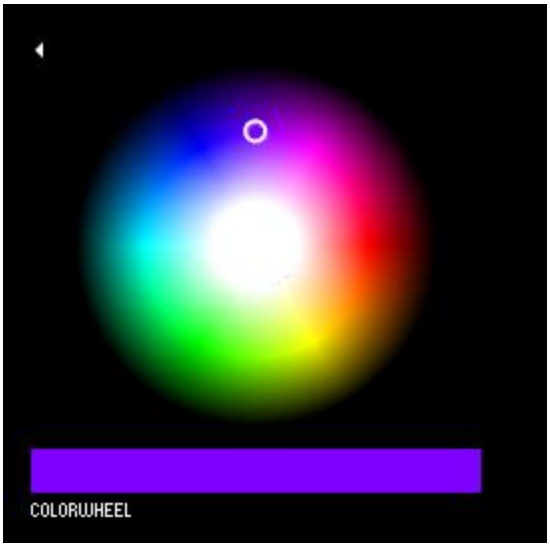
If ESPcopter did not connect, reset it and then click the reconnect button two times.

2-) Control Switches:



Button: ARM – DISARM
Fly Mode 1: YAW CONTROL
Fly Mode 2 AUTO ALTİTUDE CONTROL
Fly Mode 3: AUTO ALTİTUDE AND FLİGHT MISSION CONTROL

3-) Control RGB LED:

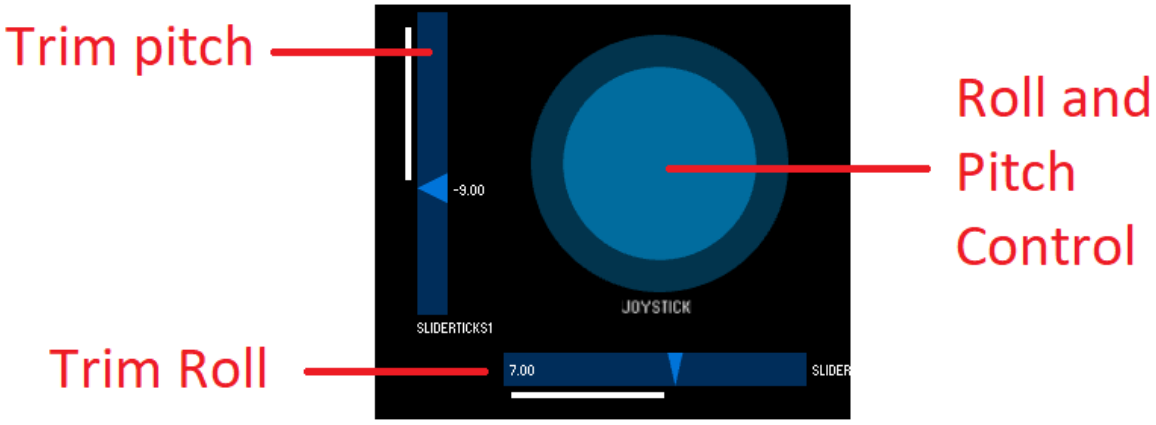


It is not working current code. We will solve the problem next version.

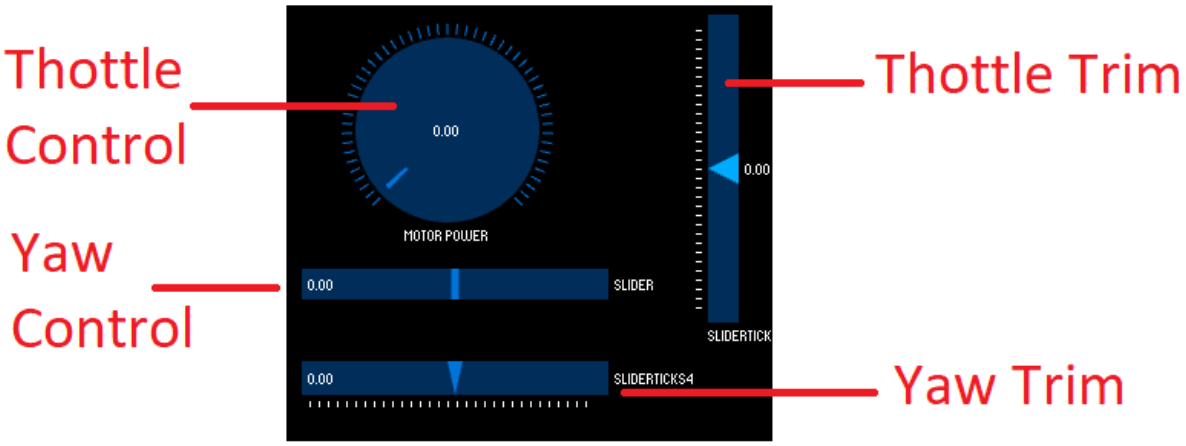
4-) Graph:

It will show something like connection speed. Current code is not working.

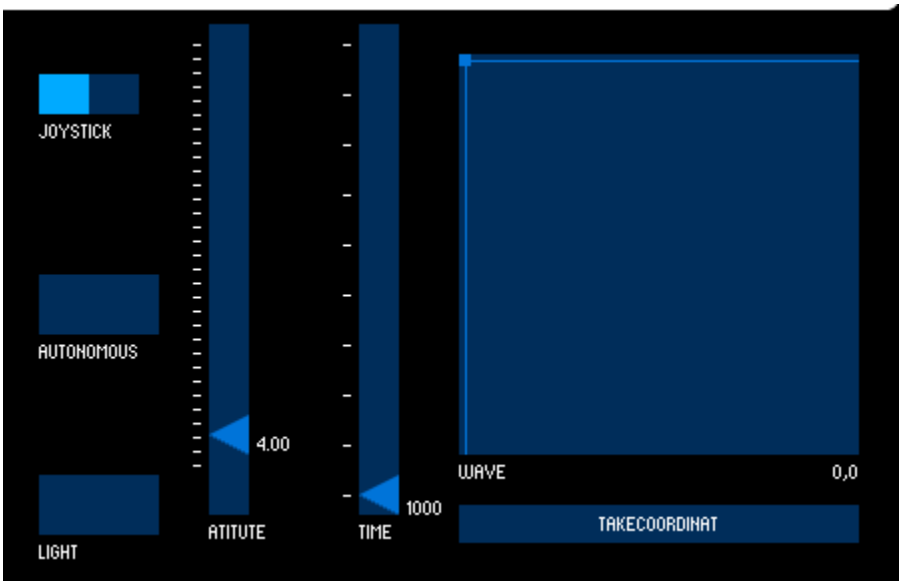
5-) Control Joystick 1:



6-) Control Joystick 2:



7-) Auto Flight Route Control Part

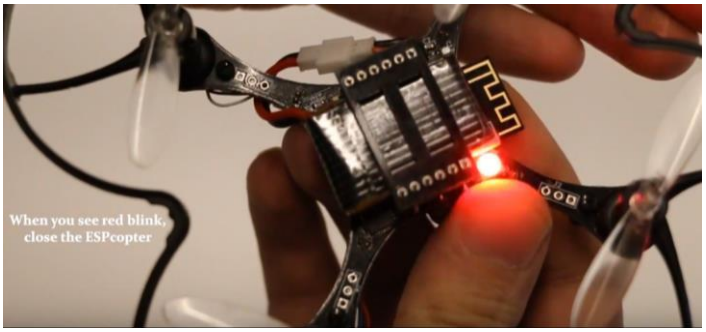


8-) Auto Flight Route Simulation:

Calibration:

1- Start Calibration

The ESPcopter has at a minimum a Gyro, acceleration and Earth magnetic field sensors. They must be calibrated when you first turn on the ESPcopter. Calibration mode starts with the red light flashing. Do not turn off the ESPcopter until the calibration algorithm has finished. If you turn it off, the calibration will run again at the next boot.



2- Magnetic Field Calibration (Compass):

Now place it on a flat, non-metal surface. The blue light indicates that the magnetic field calibration has started. At this stage you have to turn the ESPcopter around twice.

3- Gyro Calibration:

After you have calibrated the compass the ESPcopter's light will be purple. Do not touch the ESPcopter after this step. It will run its engines for a few seconds without taking off from the ground.

Calibration is completed!!!

Custom commands:

ESPcopter functions:

The LEDs and motors on the ESPcopter can be controlled using the functions found in this list.

Function	Acceptable Values	Description
esp.redLed_Digital();	0 - 1 or FALSE - TRUE	Controls Red LED on/off
esp.blueLed_Digital();	0 - 1 or FALSE - TRUE	Controls Blue LED on/off
esp.greenLed_Digital();	0 - 1 or FALSE - TRUE	Controls Green LED on/off
esp.redLed_Analog();	0 - 255	Controls Red LED brightness
esp.blueLed_Analog();	0 - 255	Controls blue LED brightness
esp.greenLed_Analog();	0 - 255	Controls green LED brightness
esp.motorFL_Analog();	0 - 255	Operates front left engine at desired power.

esp.motorFR_Analog();	0 - 255	The front right operates the engine at the desired power.
esp.motorRL_Analog();	0 - 255	Operates the rear left engine at the desired power.
esp.motorRR_Analog();	0 - 255	The rear right operates the engine at the desired power.

ESPcopter Control Table:

The control method of the ESPcopter can be changed using the definitions in this list. Only one definition should be activated from this list.

Function	Description	Control Device
#define REMOTE_XY_REMOTE	Control with RemoteXY	Phone - Tablet
#define BLYNK	Control with BLYNK	Phone - Tablet
#define PROCESSING_REMOTE	Control with processing	Computer
#define MQTT	Control with MQTT	Computer
#define REMOTE_WEB_APP	Web application control	Phone - Tablet
#define PPM_REMOTE	Control with PPM receiver	Standard RC Remote

ESPcopter Global Variable Definitions:

Function	Description	Value Range
setTrimRoll();	Trim on the X-axis.	-500 - 500
setTrimPitch();	Trim on the Y-axis.	-500 - 500
setTrimYaw();	Trim on the Yaw..	-500 - 500
setArmControl();	Motor Enable	false - true
setFlyMode_1();	Z-axis stabilization on-off	false - true
setFlyMode_2();	Height fixing on-off	false - true
setFlyMode_3();	Optical flow module with motion stabilization	false - true
landing();	Landing	false - true
setMotorMax();	Set maximum motor power	600-900
getRX_throttle();	Motor power rating	0 – (motorMax)
getRX_roll();	The value in the X-axis	-100 : + 100
getRX_pitch();	The value in the Y-axis	-100 : + 100

getRX_yaw();	The value in the Z-axis	-100 : + 100

Autonomous Flight Commands:

Function	Description	Value Range
takeOff(Y, T);	When the command line runs, the drone automatically takes off.	Y: 200 - 1000 Height T: Flight time
goforward(T);	The drone moves forward during the duration.	T: Flight time
goBack(T);	During the T Time the drone moves back.	T: Flight time
goLeft(T);	The drone moves to the left during the duration.	T: Flight time
goRight(T);	The drone moves right through the time.	T: Flight time
turnRight(D);	D rotates right up to its own angle in angle.	D: Rotation angle
turnLeft(D);	The angle of D turns to the left in its own frame	D: Rotation angle
delay_(T);	It allows you to wait before executing the next command	T: Standby time
Land();	In autonomous flight mode, this must be at the end of the commands.	

Altitude Hold Module

Function	Description	Value Range
setVl5310xControl ();	vl5310x module on-off	False - true
setTargetOto();	Height stabilizer with vl5310x module	250 - 1000
getOtoMeasure();	Drone's elevation data	0- 1000

Buzzer Module:

Function	Description	Value range
esp.buzzer();	0 - 1 or FALSE - TRUE	On- Off buzzer

Neopixel Module:

Function	Description	Value range

#define NeoPixel	Turns the Neopixel module on and off	Include in the program
ESPrainbow();	Makes an automatic rainbow effect	
ESPsetPixel (x,r,g,b);	Set each led separately. After setting pixels call	X= 1 – 12 R(Red)= 0 - 255 G(Green)= 0 - 255 B(Blue)= 0 - 255
ESPpixelShow();	Applies the changes made with	

Optical Flow Module:

Function	Description	Value range
SetPointOpt[0]	Sets the speed of Drone using the optic flow sensor. If this value is equal to zero, the drone remains stationary in the x-axis. Positive moves right, Negative moves left	-15 - +15
SetPointOpt[1]	Sets the speed of Drone using the optic flow sensor. If this value is equal to zero, the drone stops at the y axis. Positive goes forward, Negative- goes back	-15 - +15
deltaCalX	X-axis flow data from the optical flow extender.	Relative to the drone current speed
deltaCalY	Y-axis flow data from the optical flow damper.	Relative to the drone current speed

Multi-Distance Module:

Function	Description	Value range
#define HandControl #define AntiCollision	Manual control or collision prevention system Must choose one or the other	
Distance_Y_1();	Y (+) axis distance data	50-1000
Distance_Y_0();	Y (-) axis distance data	50-1000
Distance_X_1();	X (+) axis distance data	50-1000

Distance_X_0();	X (-) axis distance data	50-1000

Other Commands

All other common Arduino and ESP8266 commands can be used in the library except for the following which will interfere with drone operation.

delay();
analogWrite();
Tone();