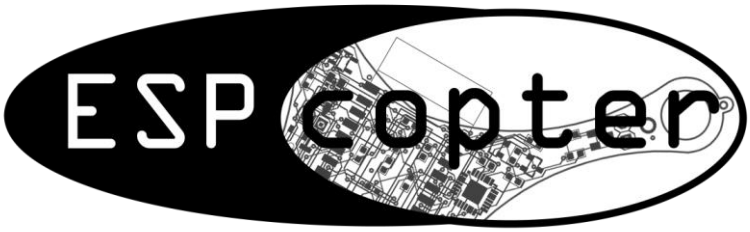


ESPcopter SDK

1.0.2 (Beta)

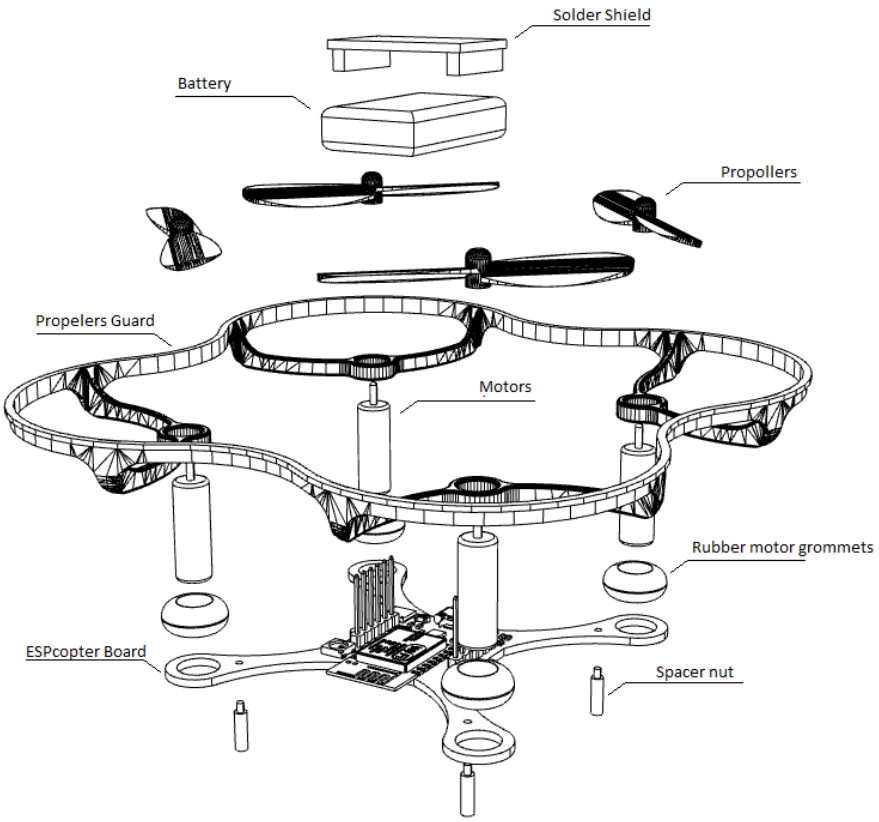
Contents

Note:

1-) General Review

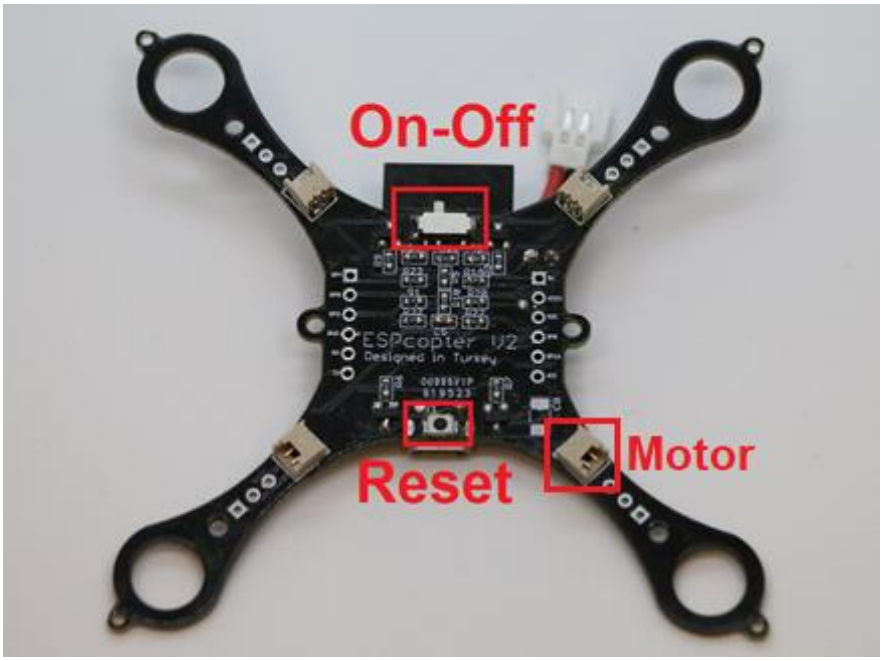
1.1-) Internal Features of the ESPcopter:

1.2-) ESPcopter Assembly Sequence:



1.3-) ESPcopter Switch and Button

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



1.4-) 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 indicates that ESPcopter currently charging

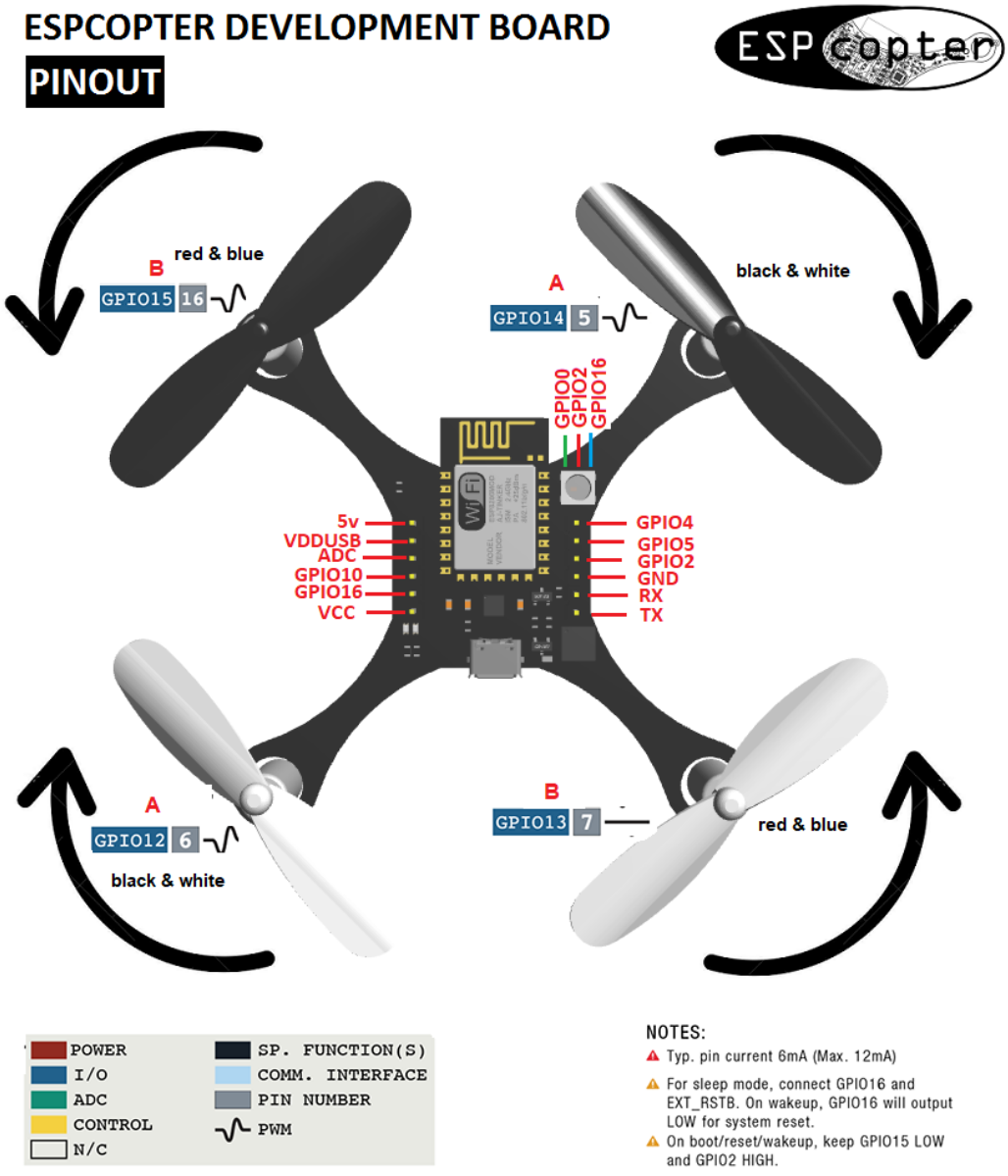
1.5-) Pinout and Propeller and Motor Directions

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**



1.6-) ESPcopter Control Methods:

1.7-) ESPcopter Default Wifi Information:

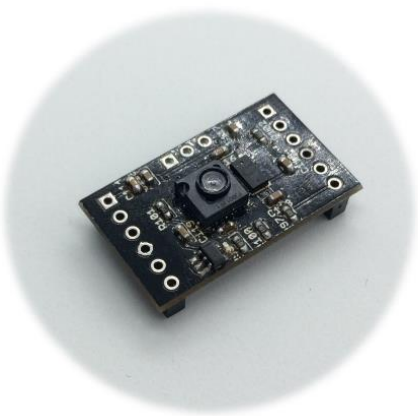
SSİD: ESPcopter

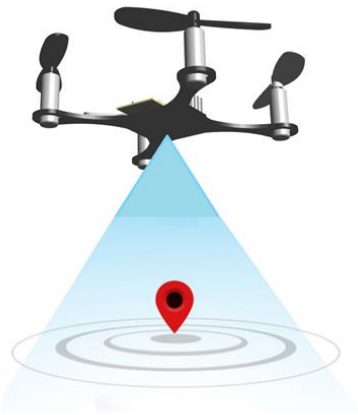
Pass: 123456789

2-) ESPcopter Geliştirme Modülleri

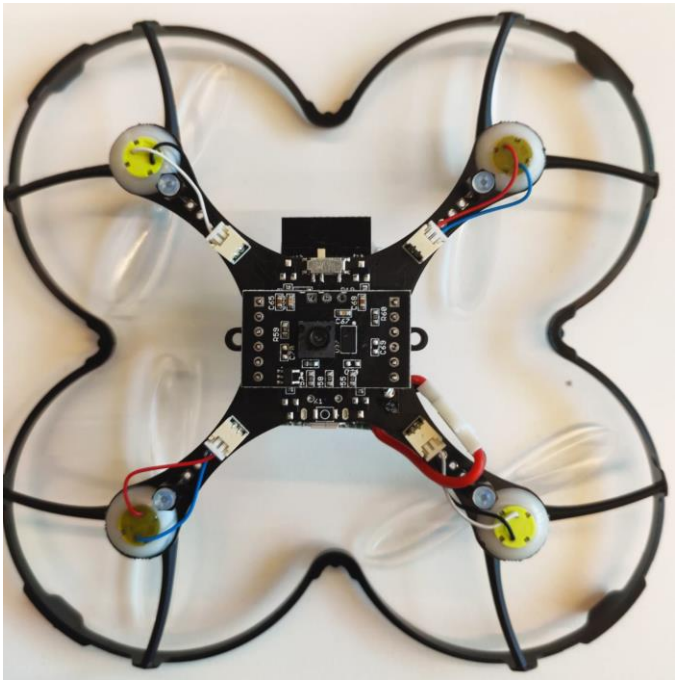
2.1-) Optic Flow Module:

Optic flow module understands the drone's movement via processing the images of the ground. In this way drone can stay in the same location or it can move autonomously.



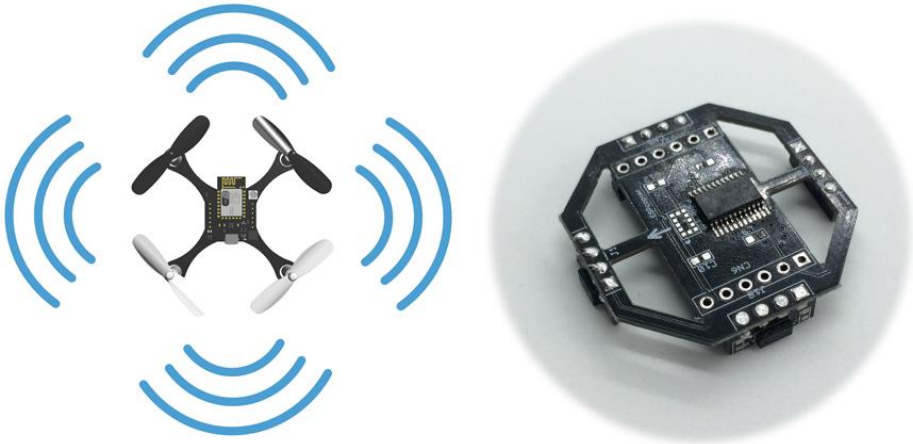


2.1.1-) Optical Flow Module Connection**:**



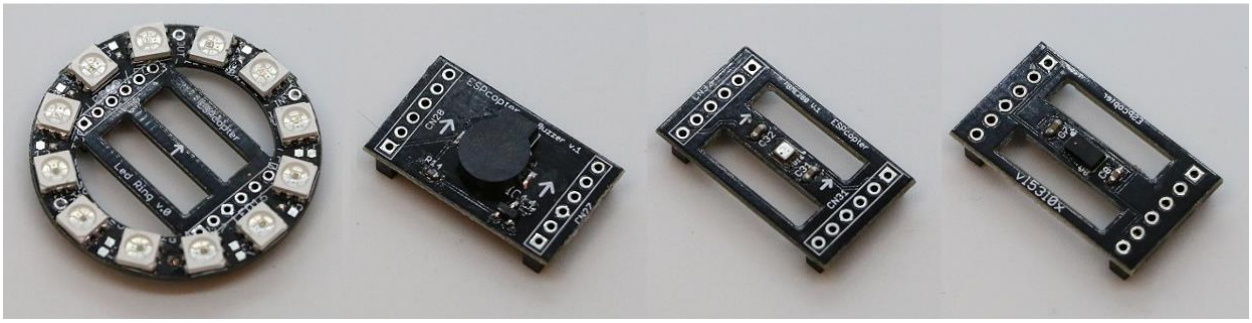
2.1-) Multi-Ranger Module:

There are 4 laser sensors on drone. Those sensors can understand the distance up to 1 meter. With the help of this you can make anti-collision system, hand control system, autonomous flight system etc.



2.3-) Other Modules:

ESPcopter Neopixel Module | ESPcopter Buzzer Module | ESPcopter Temperature, Pressure and Humidity Module | ESPcopter Altidute Hold Module



2.3.1-) ESPcopter Neopixel Module:

There are 12 NeoPixels in this circular card. It can connect to the ESPcopter's top input pins. You can use the NeoPixel module to make various light shows while flying with the ESPcopter.

2.3.2-) ESPcopter Buzzer Module:

There is one buzzer in the buzzer module. It can connect to the ESPcopter's top input pins. You can play music through the Buzzer module when you are not flying, and you can hear the warning sounds when you fly.

2.3.3-) ESPcopter Temperature Pressure and Humidity module:

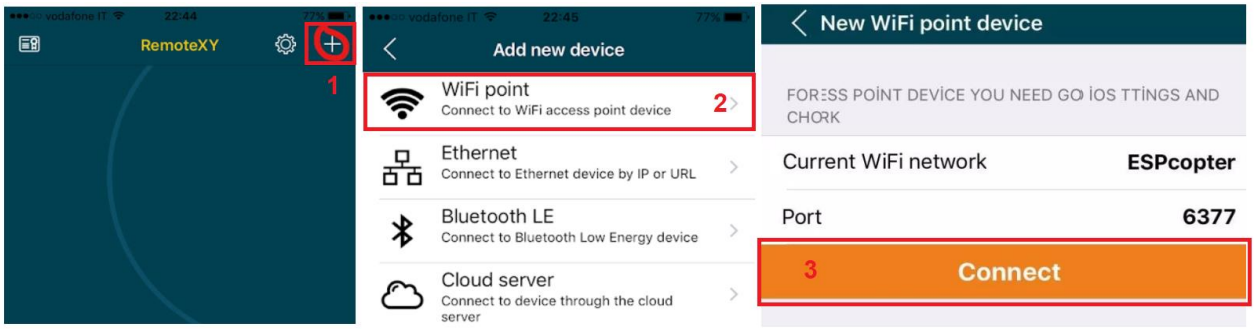
This module has one BEM280 sensor. It can connect to the ESPcopter's top input pins. You can use this module to record weather data while flying or you can send these data to your phone or computer over the internet in your IoT project.

3-) ESPcopter Control Application (RemoteXY):

Connection:

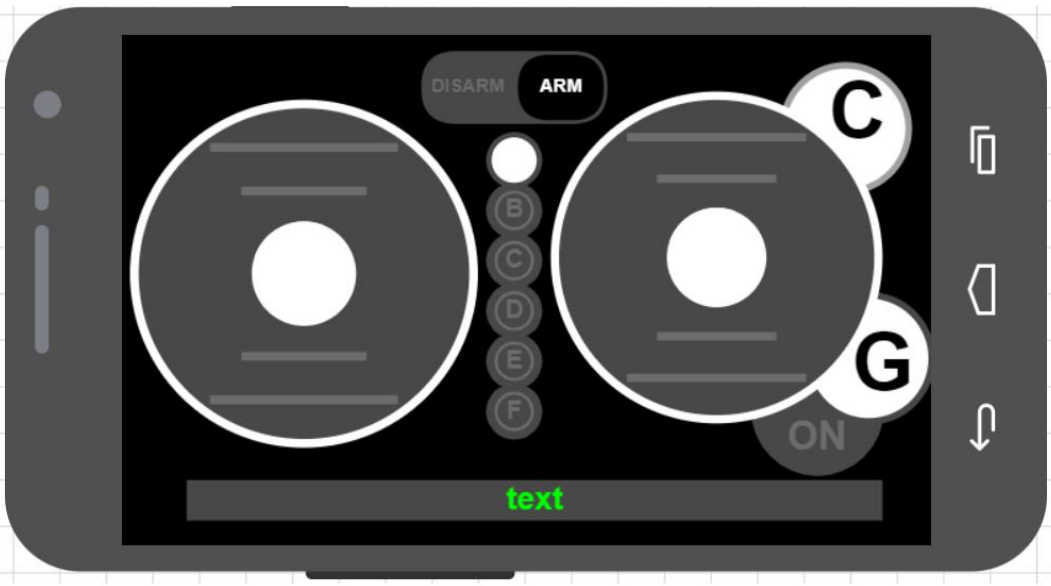
After installing the RemoteXY application on your phone, turn ESPcopter on and connect your phone and ESPcopter via wifi network.

After you make the connection, open RemoteXY and do following steps.

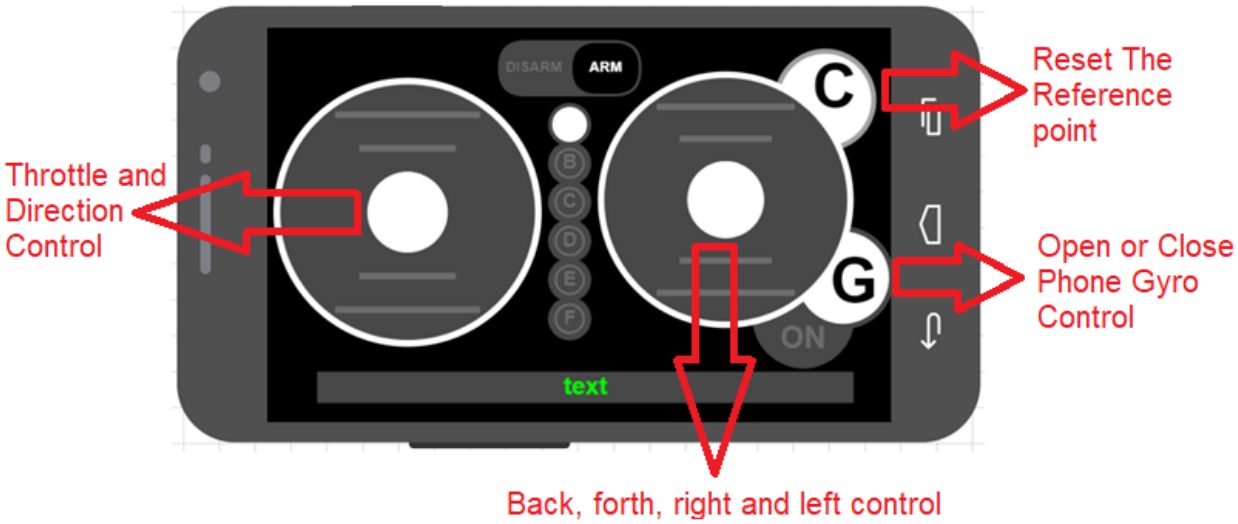


After making the connection, the screen below will open automatically.

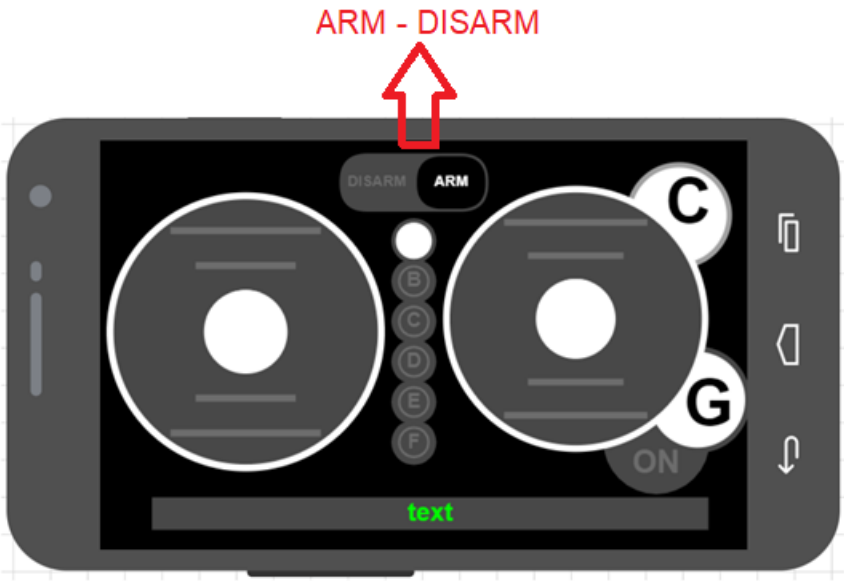
After the first connection. There will be ESPcopte box in RemoteXY app. You can connect ESPcopter by cliking this box.



3.1-) Control Review:



Joysticks:

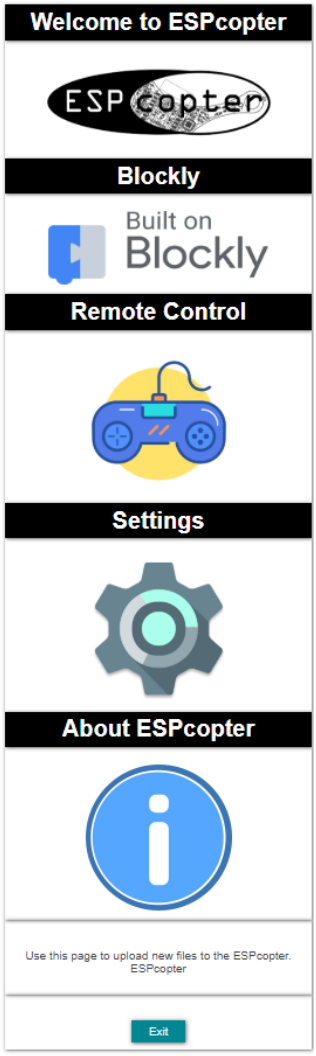


Free Flight(All mode is inactive)
Direction stabilization mode
Altitude Hold Mode
B and C Mode
Optical Flow stabilization mode
B, C and E Mode

4-) ESPcopter Web Interface:

Thanks to the ESPcopter website, you can control the ESPcopter from your phone tablet and computer without any application installation and program it wirelessly thanks to the blocks. Follow the following steps to control the ESPcopter with the Web Interface.

1. Login to the site by typing http://192.168.4.1 into the Chrome web browser Search bar.
2. The ESPcopter website will open automatically.



With this web interface;

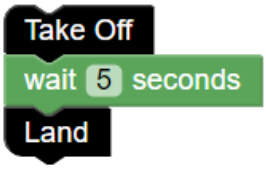
- You can open the block programming interface
- You can open the web control interface
- You can change some drone settings such as Wi-Fi password in settings section.

4.1-) Block Programming with Web Interface:

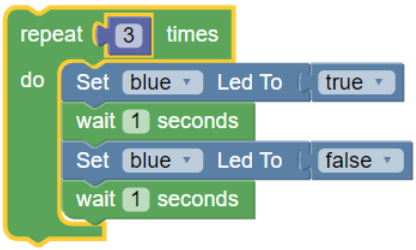
You can open the web blockly control interface by clicking the blockly icon from the web interface

You can code ESPcopter wirelessly with Blocks

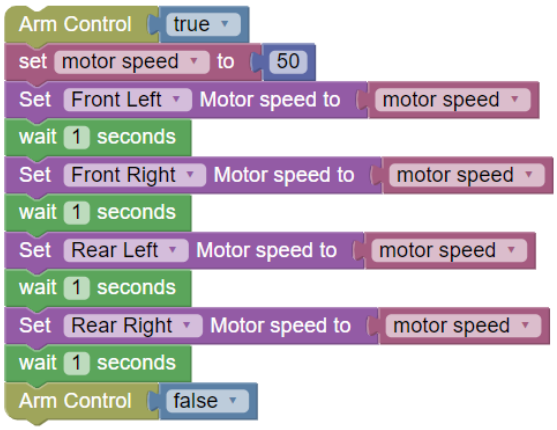
- **Simple Flight Code:**



- **Led Control Code:**

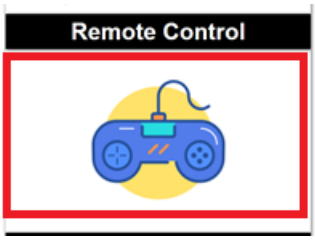


- **Motor Control Code:**



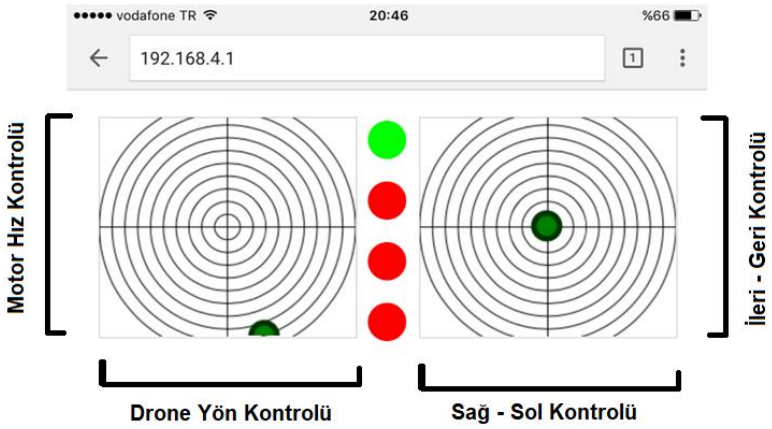
4.2-) Web Interface Control Controller:

You can open the web control by clicking on the remote control icon from the web interface

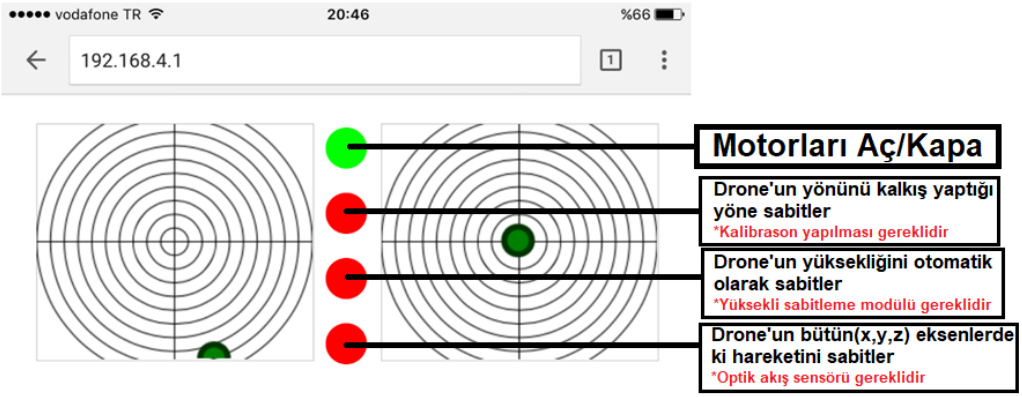


4.2.1-) Control Review:

Joysticks Review:

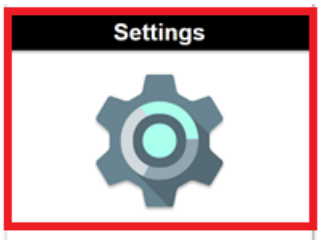


Buttons:



4.3-) Web Interface Settings:

You can open your settings page by clicking the setting icon from the web interface:



You can change the ESPcopter wifi name and password via the settings page.

4.5-) Web Interface Information:

You can open your information page by clicking the information icon from the web interface:

5-) Software:

5.1-) Arduino Installation:

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

5.2-) Driver Installation:

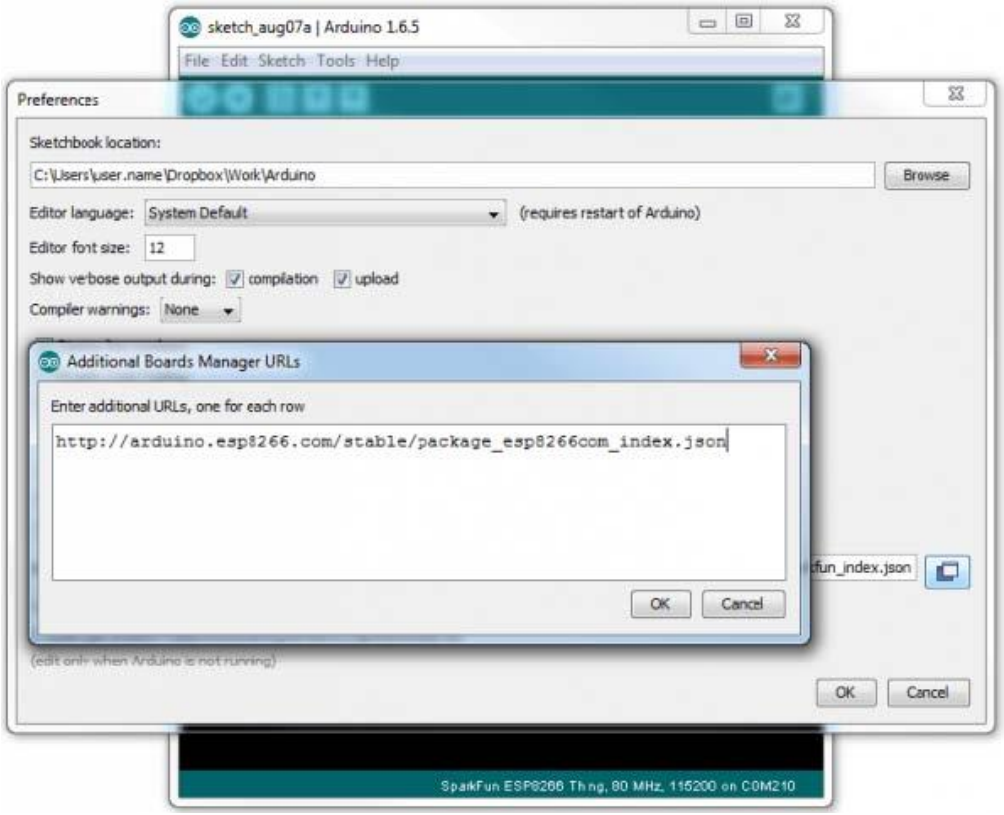
The following driver is required for the ESPcopter to be recognized by the computer. Download and install the appropriate driver version for your OS.

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

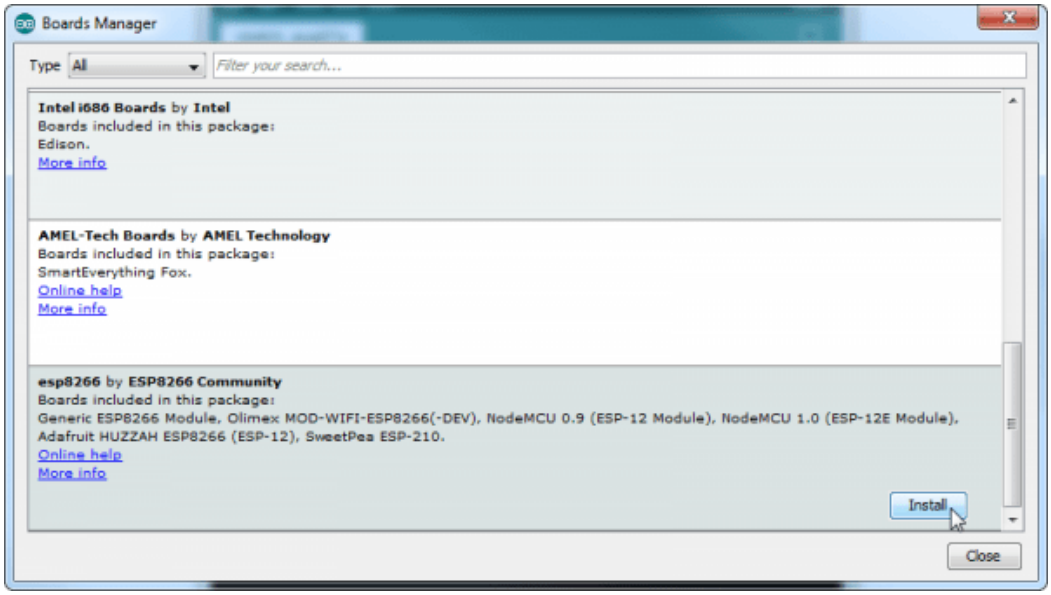
5.3-) ESP8266 Library

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:

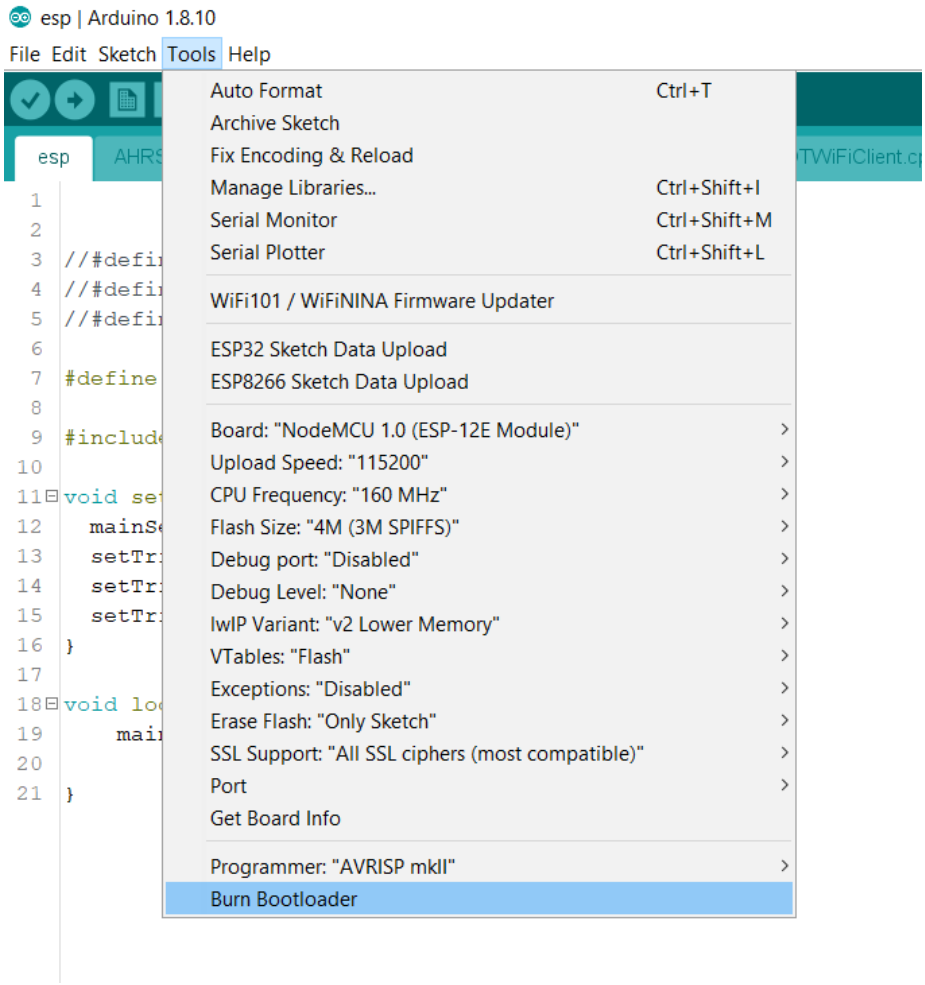
http://arduino.esp8266.com/stable/package_esp8266com_index.json



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 **2.6.3** 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.

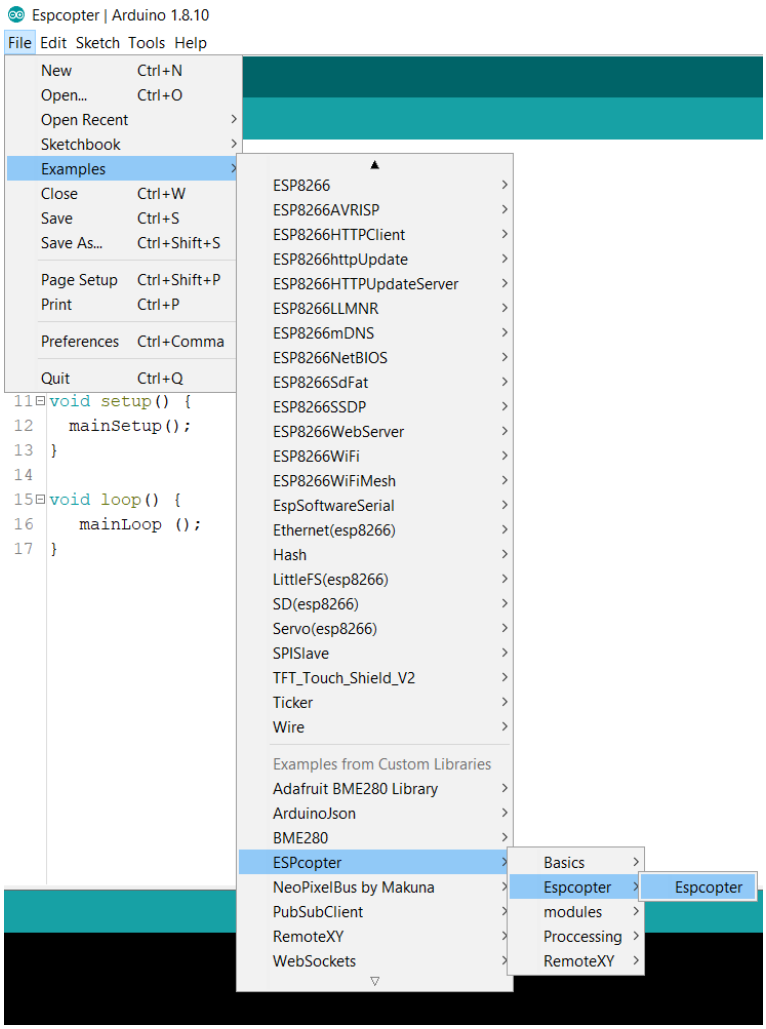


5.4-) Downloading the ESPcopter library:

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:



5.5-) Spiffs Memory Updater Installation:

5.5.1-) What is Spiffs Memory?

Flash File System (SPIFFS) is a SPI Flash (64kBytes to 3Mbyte) In this flash memory ESP stores the program. This filing system can be used to store infrequently changing data such as; web pages, configurations, sensor calibration data etc.

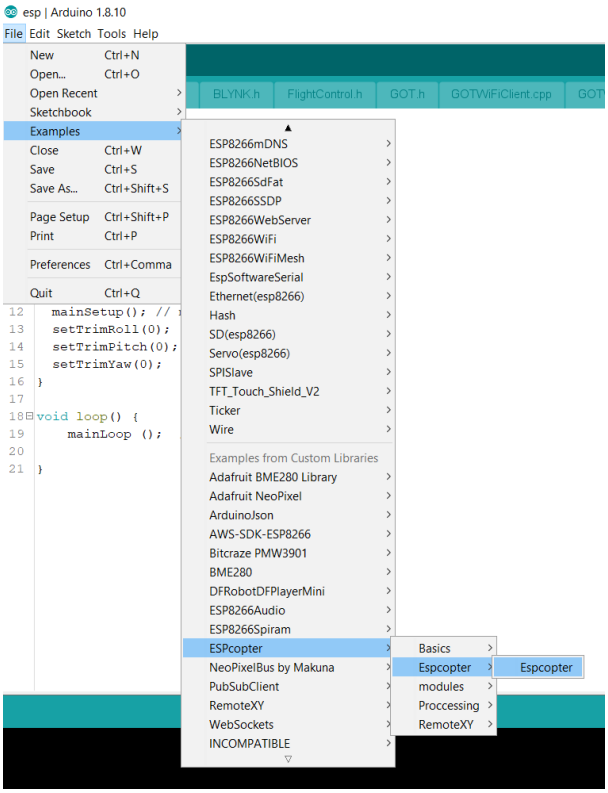
5.5.1-) Install Spiffs Memory Uploader?

- Download the tool: https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.5.0/ESP8266FS-0.5.0.zip
- In your Arduino sketchbook directory, create `tools` directory if it doesn't exist yet.
- Unpack the tool into `tools` directory (the path will look like `<home_dir>/Arduino/tools/ESP8266FS/tool/esp8266fs.jar`) If upgrading, overwrite the existing JAR file with the newer version.
- Restart Arduino IDE.

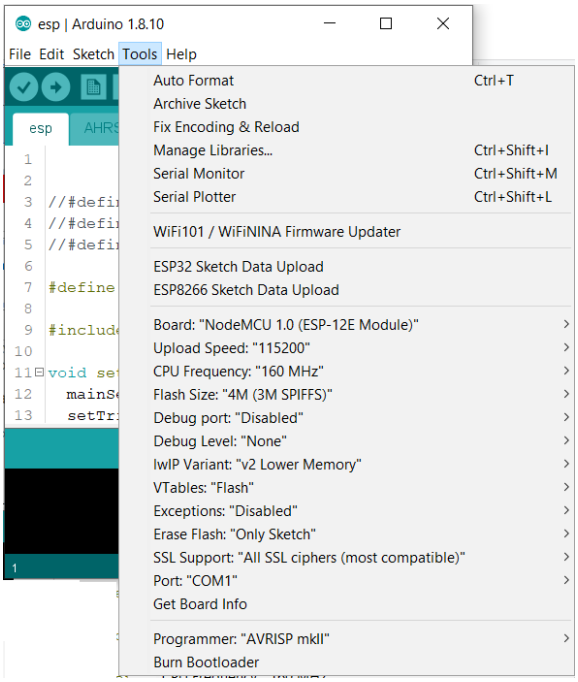
6-) Update Software:

6.1-) Update Main Software:

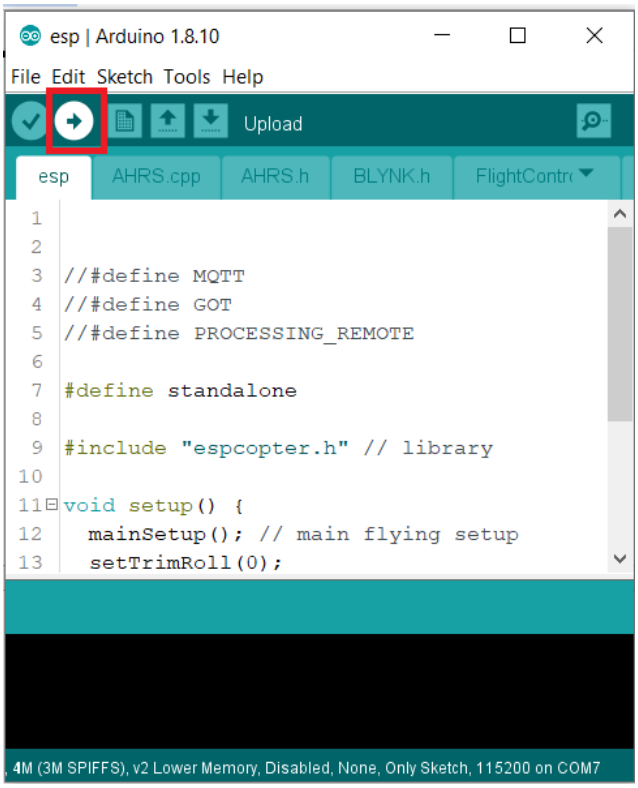
- To update the code, the drone must be turned on and connected to the computer via USB cable. Open the ESPcopter code from the ESPcopter Library in the Examples section.



- Make the necessary settings from the Tools section.

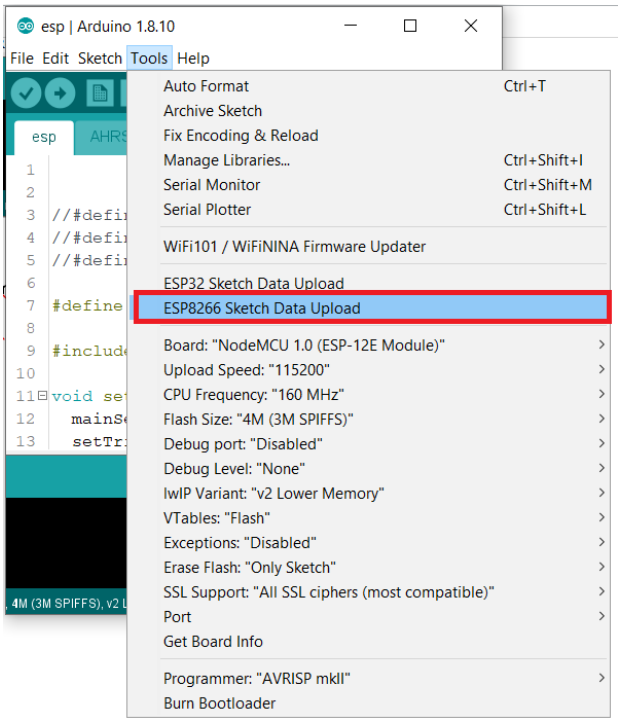


Upload the code by clicking the Upload button.



6.2-) Update Spiffs Memory:

- Make sure you have selected a board, port, and closed Serial Monitor.
- If your board requires you to press a button (or other action) to enter bootload mode for flashing a sketch, do that now.
- Select Tools > ESP8266 Sketch Data Upload. This should start uploading the files into ESP8266 flash file system. When done, IDE status bar will display `SPIFFS Image Uploaded` message.



7-) ESPcopter Arduino Special Commands:

7.1-) 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.

7.2-) 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

7.3-) 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

7.4-) 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.	

7.5-) 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

7.6-) Buzzer Module:

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

7.7-) 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	

7.8 -) 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	-15 - +15

	x-axis. Positive moves right, Negative moves left	
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

7.9-) 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

7.10 -) 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();