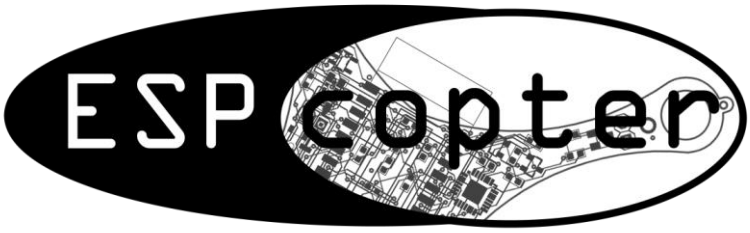
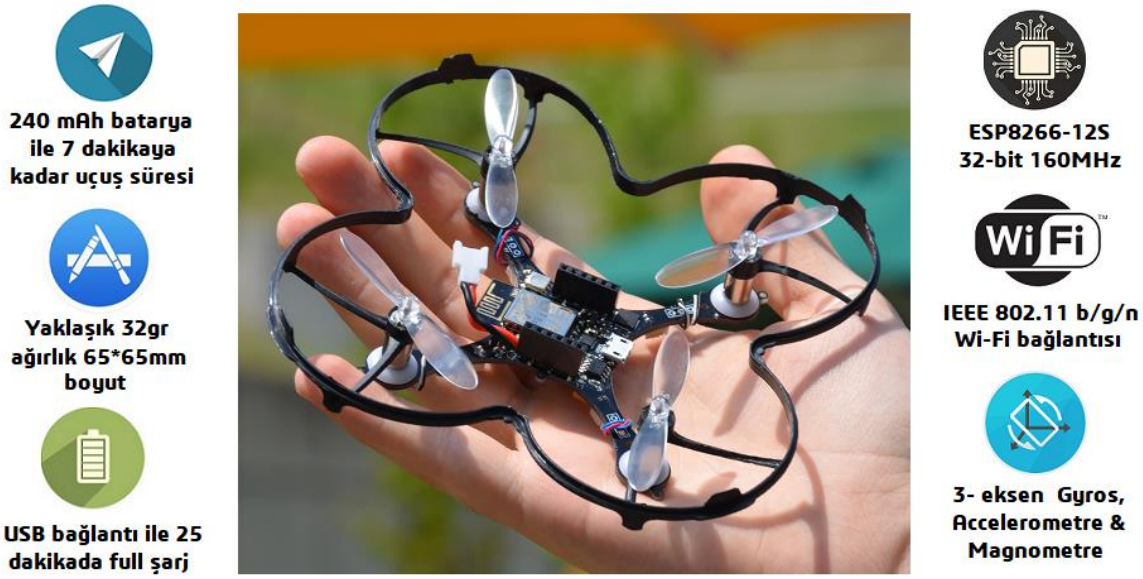


ESPcopter SDK

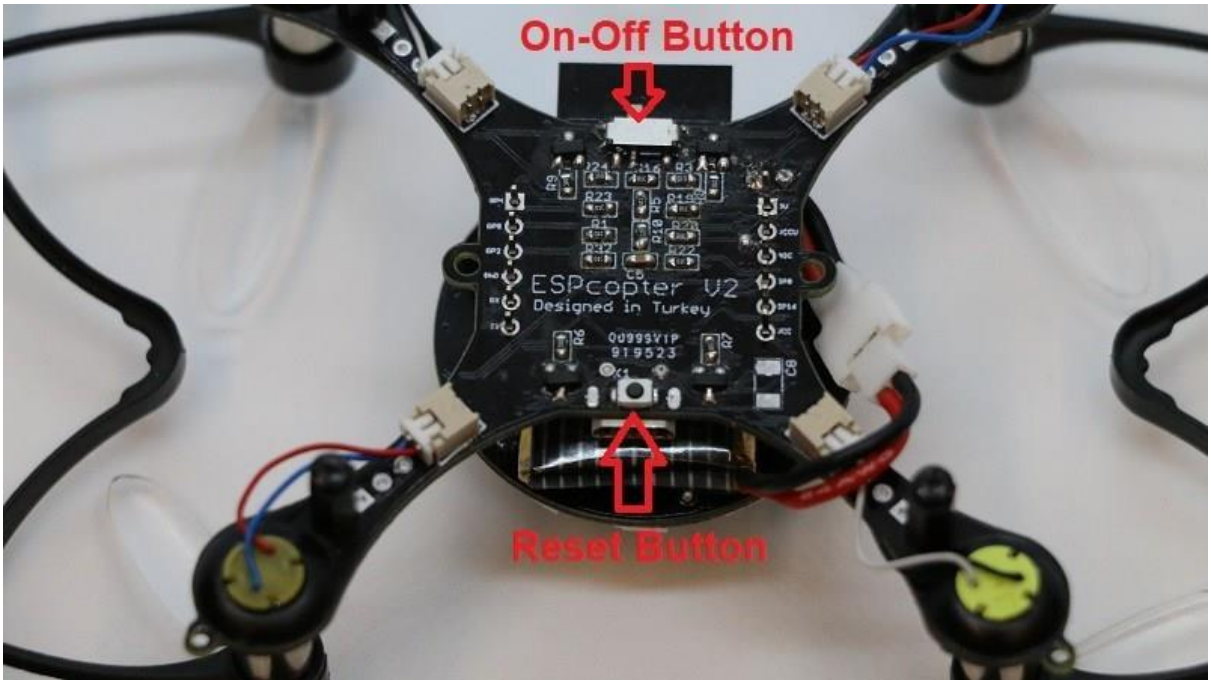
1.0.0 (Beta)

1-) Genel İnceleme

1.1-) ESPcopter'in Dahili Özellikleri:



1.2-) ESPcopter Düğme ve Buton



1.2-)Pin Çıkışları ve Pervane yönleri

Pervaneler üzerinde bulunan harflere göre takılırken motorlar kablo renklerine göre takılmalıdır.

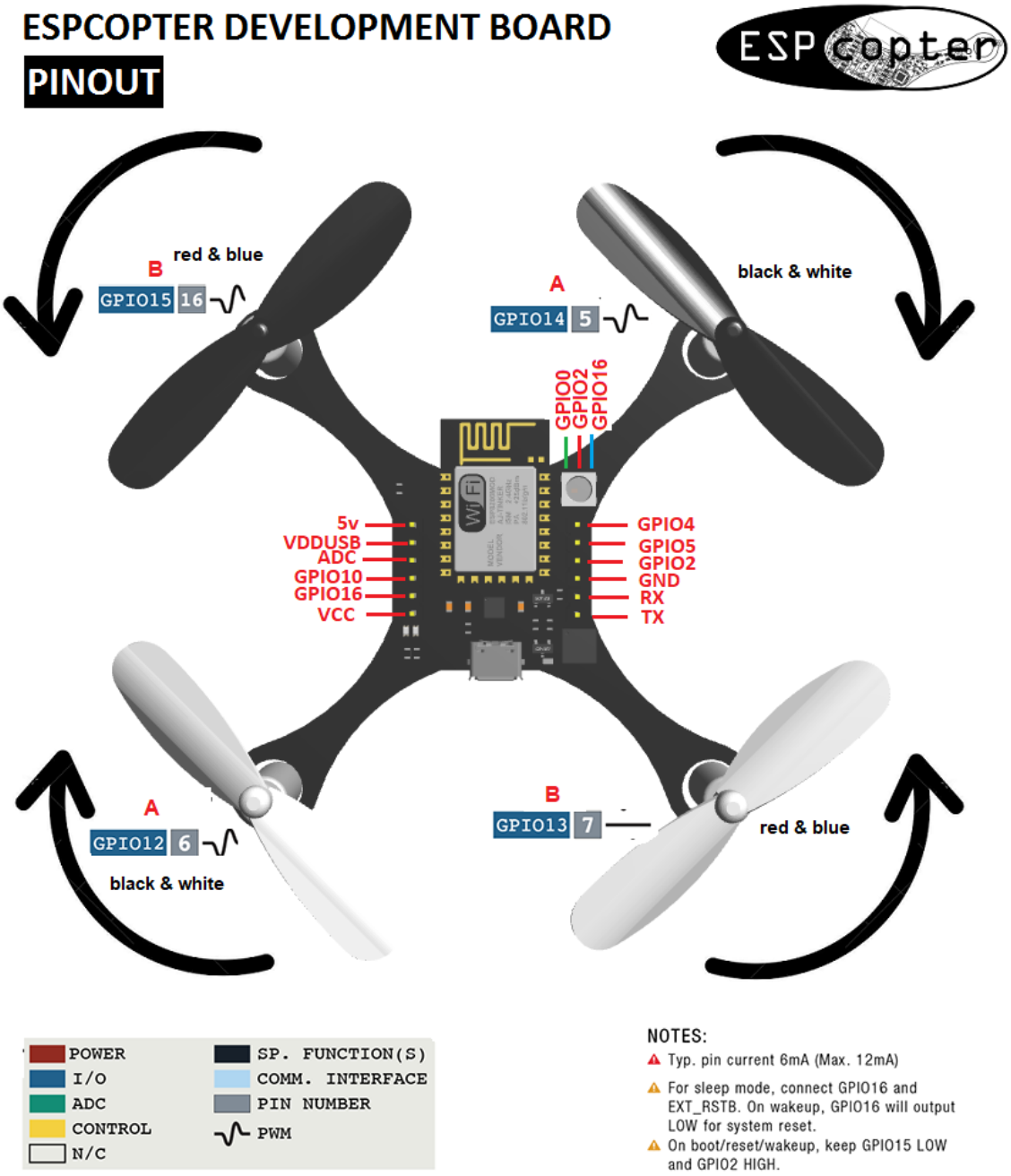
Motor konumları:

Sol Ön: B – Kırmızı, Mavi

Sağ Ön: A - Siyah, Beyaz

Sol Arka: A – Siyah, Beyaz

Sağ Arka: B – Kırmızı, Mavi



1.3-)ESPcopter'I Nasıl Şarj Edebilirim:

ESPcopter'I üzerinde buluann micro-usb ile şarj edebilirsiniz. ESPcopter'in şarj olabilmesi için üzerinde bulunan düğmenin kapalı konumda olması lazımdır.

Kırmızı Işık: Şarj oluyor

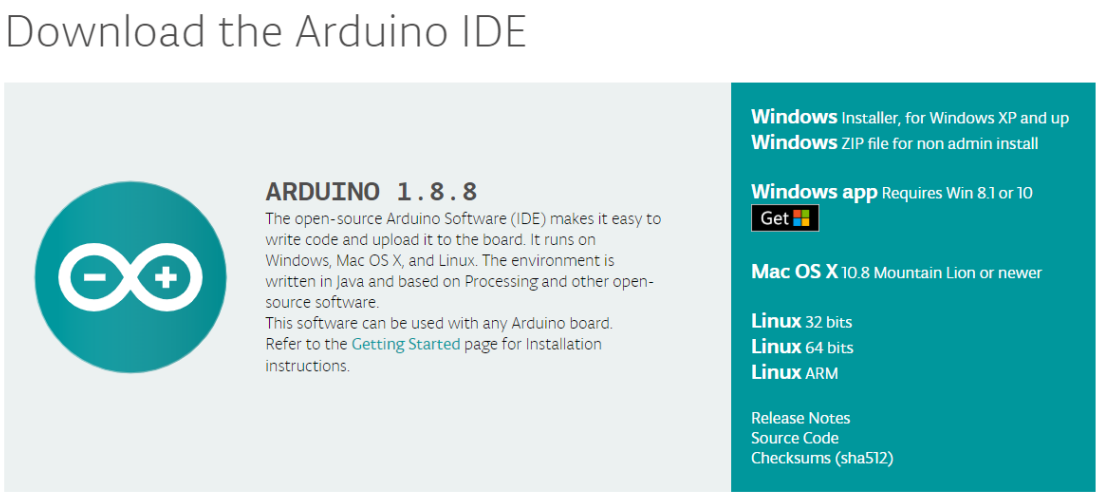
Yeşil Işık: Full Şarj

2-) Yazılım:

2.1-) Arduino Kurulumu:

Arduino'nun sitesinden en son sürümünü indirip bilgisayarınıza kurunuz.

https://www.arduino.cc/en/Main/Software



2.2-) Driver Kurulumu:

ESPcopter'in Bilgisayar tarafından tanınması için driver yazılımını bilgisayarınıza kurmanız gerekmektedir.

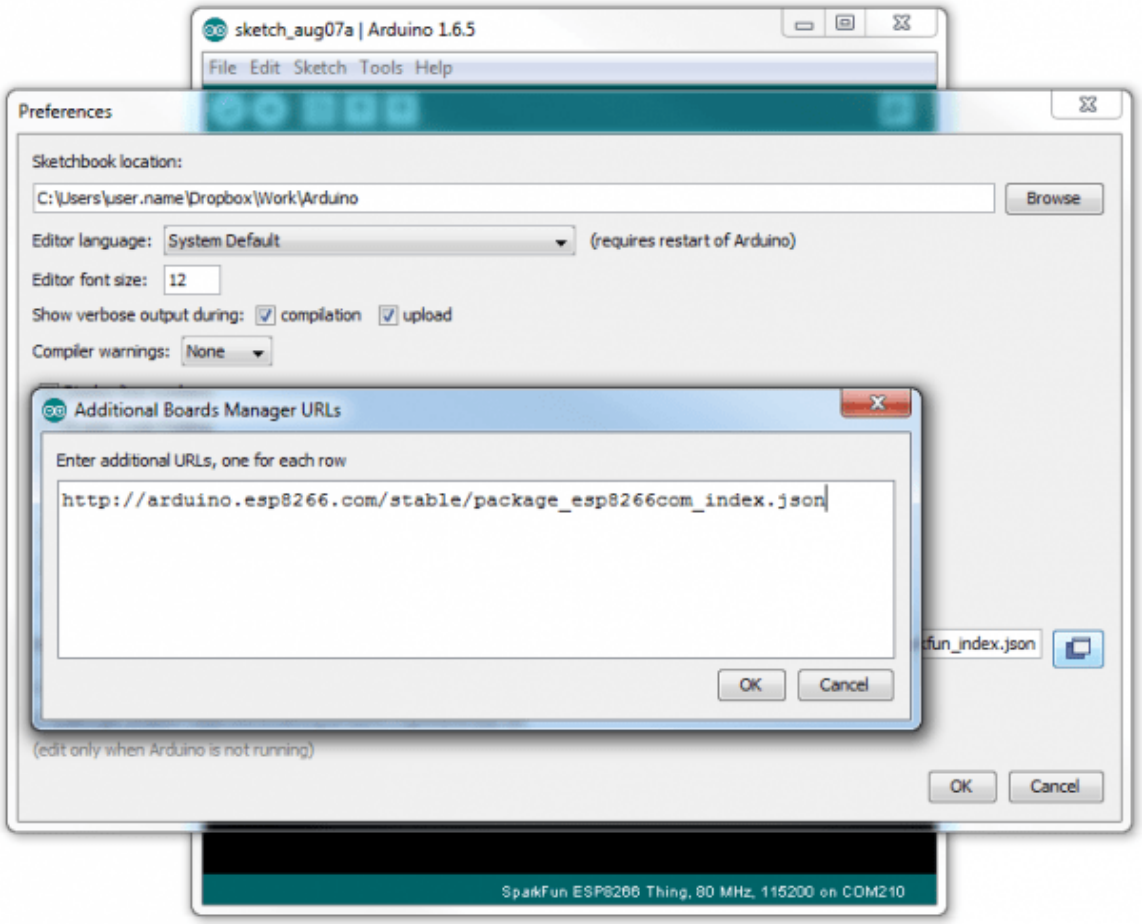
https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

2.3-) ESP8266 Kütüphanesi

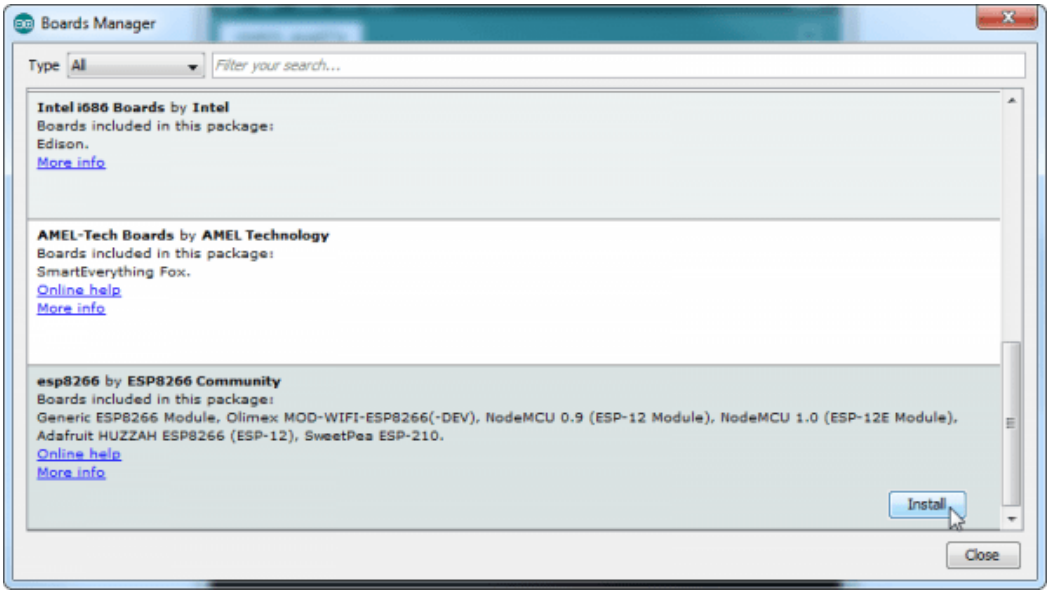
(**Dosya**> **Seçenekler**).

Aşağıda bulunan linki ” Ek kartlar URL” (“Additional Board Manager URLs” text box:). Kısmına yapıştırnız.

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

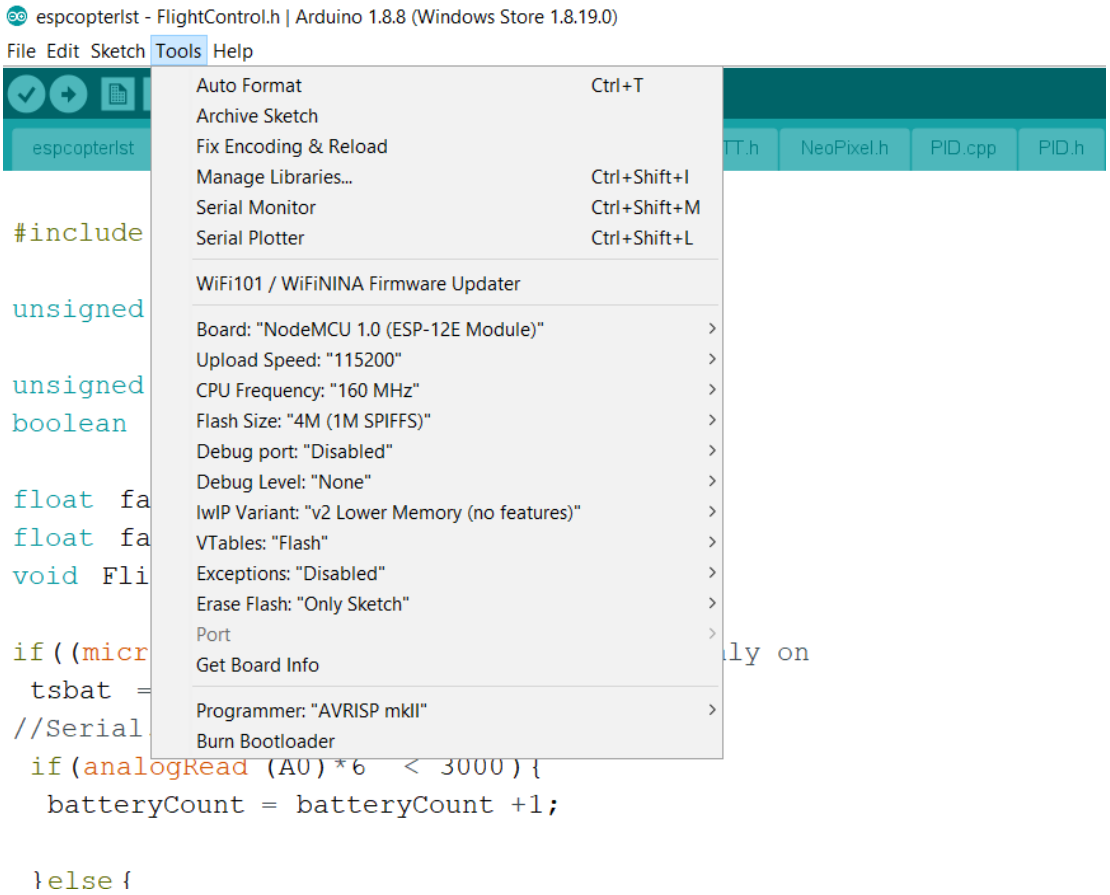


Ok tuşuna bastıktan sonra sırada board yükleme kısmı var. **Araçlar > Kartlar > Kart Yöneticisi(Tools > Boards > Boards Manager.)**. Adımlarını izleyiniz. Arama kısmına ESP yazdıktan sonra gelen seçenekler arasında **esp8266 by ESP8266 Community** seçeneceğini göreceksiniz. **2.5.0** versiyonunu indiriniz.



İndirme işlemi internet hızınıza göre 10 dk kadar sürebilir.

İndirme işlemi bittikten sonra **Araçlar** sekmesinden NodeMCU 1.0 seçeneğini seçiniz ve diğer ayarları uygulayınız.



ESPcopter kütüphanesinin indirilmesi:

İnternet sitesi üzerinden en son yayınlanan sürümü indiriniz.

http://espcopter.com/code-release/

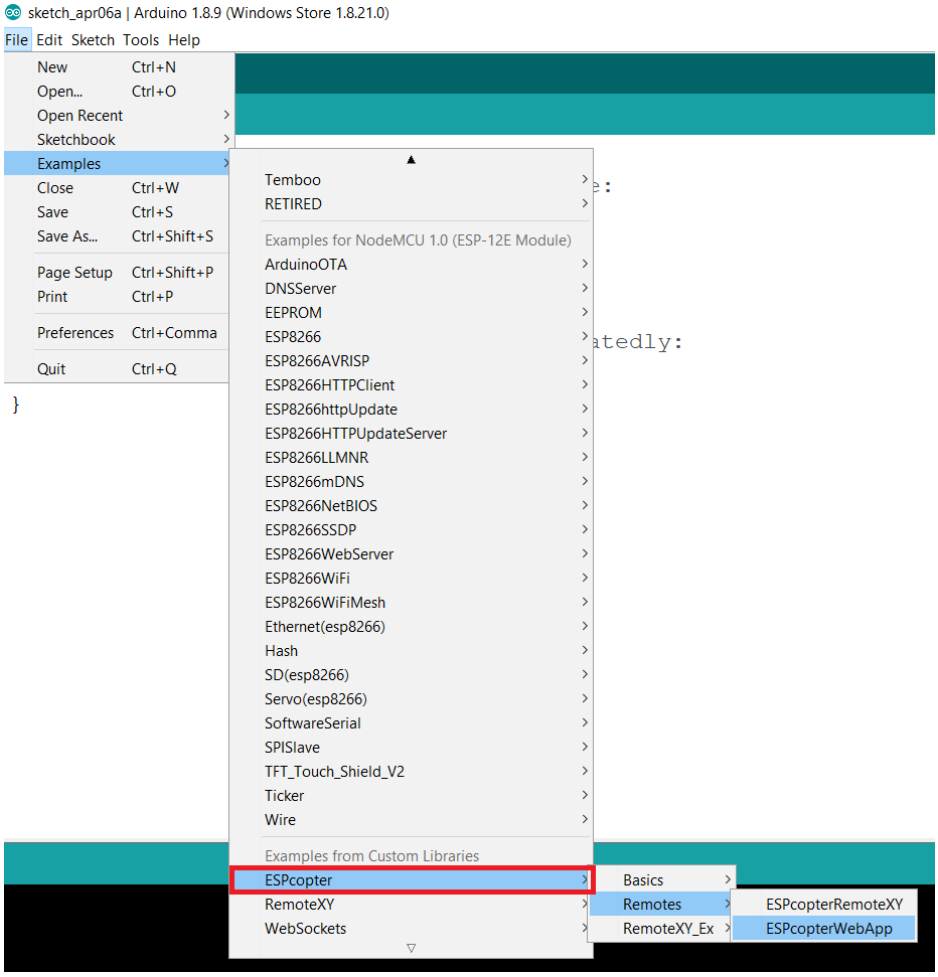
Code

DOWNLOAD ESPCOPTER ARDUINO CODE

Version 1.0.0 Beta - 13.03.2019

İndirme işlemi bittikten sonra zip dosyasını iki kez çıkartınız ve “ESPcopter” dosyasını **(**Dosyalar**>** Arduino**>**Kütüphane**)** dosyası içerisine atınız.

Örnekler bölümünde ESPcopter’in örnek kodlarını göreceksiniz.



ESPcopter standalone Code(Web APP)

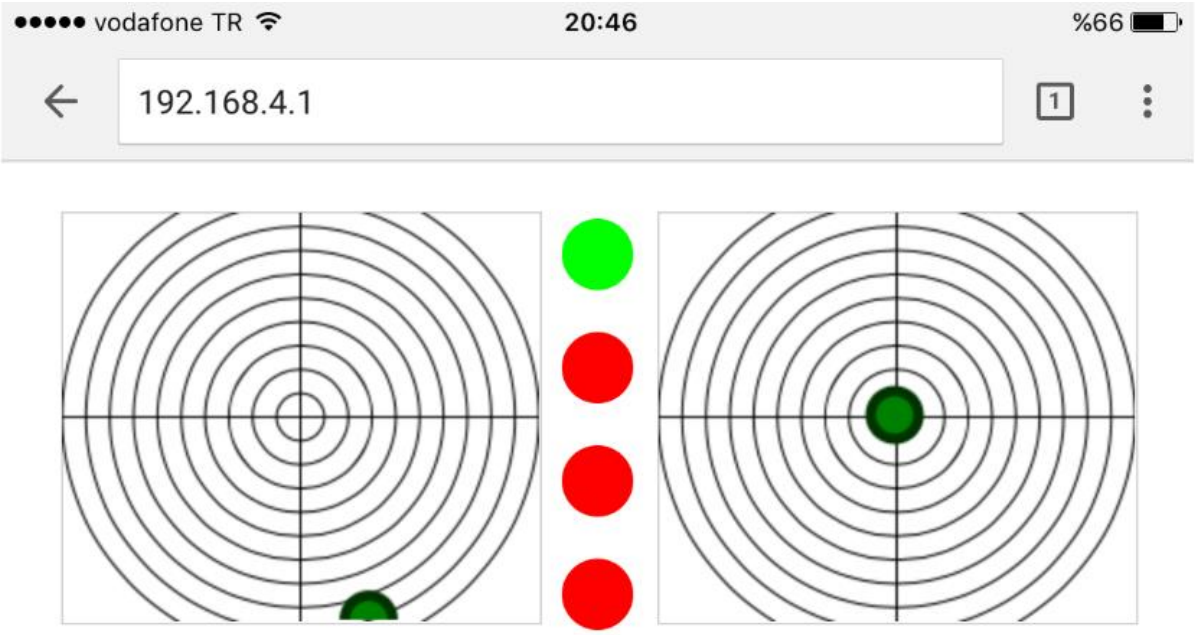
```
#define REMOTE_WEB_APP
#include <espcopter.h>
void setup() {
mainSetup();
}
void loop() {
mainLoop();
}
```

Bağlantı:

Yazılımların yüklenmesini yaptıktan sonra drone açık konuma getirin. Telefonunuz wifi sekmesinde drone'nun ismini göreceksiniz.

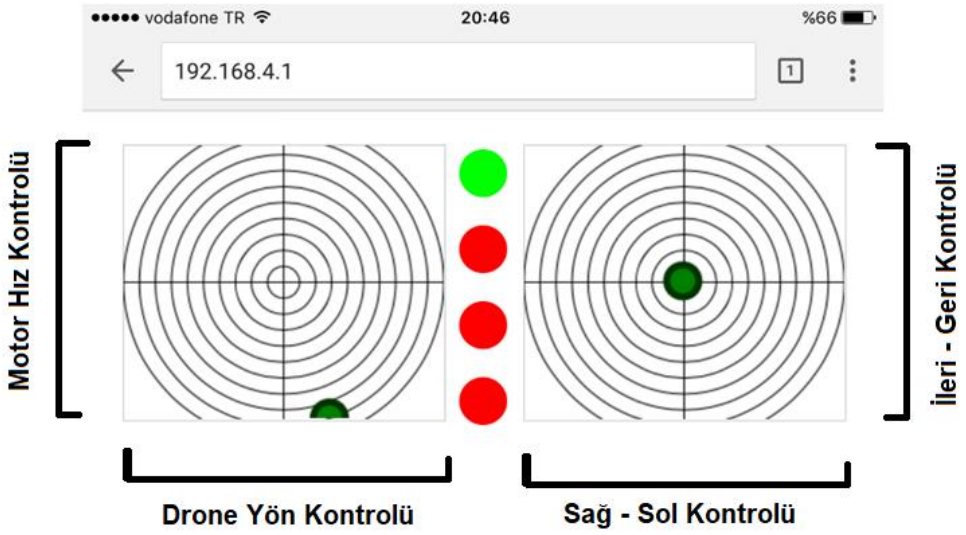
WIFI_SSID "ESPcopter"
WIFI_PASSWORD "12345678"

Bağlantıyı yaptıktan sonra telefonunuzdan herhangi bir web browser uygulamasını açınız. Arama satırına 192.168.4.1 yazıp aratınız.

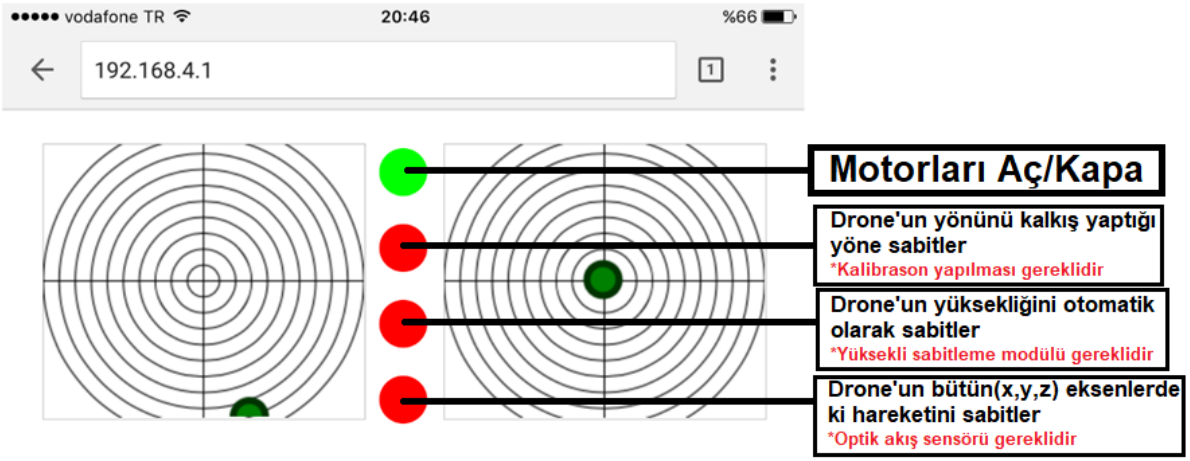


Kumanda İncelemesi:

Joistikler:



Buttonlar:



ESPcopter standalone Code(RemoteXY)

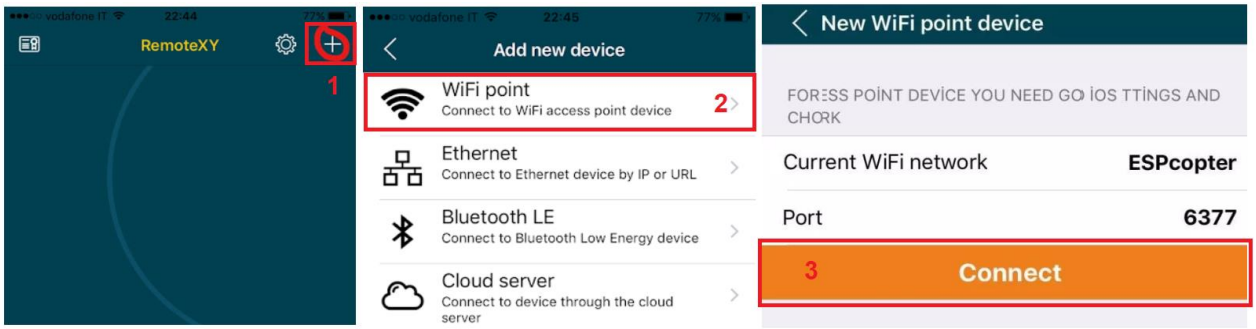
```
#define REMOTEXY_WIFI_SSID “ESPcopter_RemoteXY”
#define REMOTEXY_WIFI_PASSWORD “12345678”
#define REMOTE_XY_REMOTE
#include <espcopter.h> // library
void setup() {
  mainSetup(); // main flying setup
}
void loop() {
mainLoop ();  // main flying loop
}
```

Bağlantı:

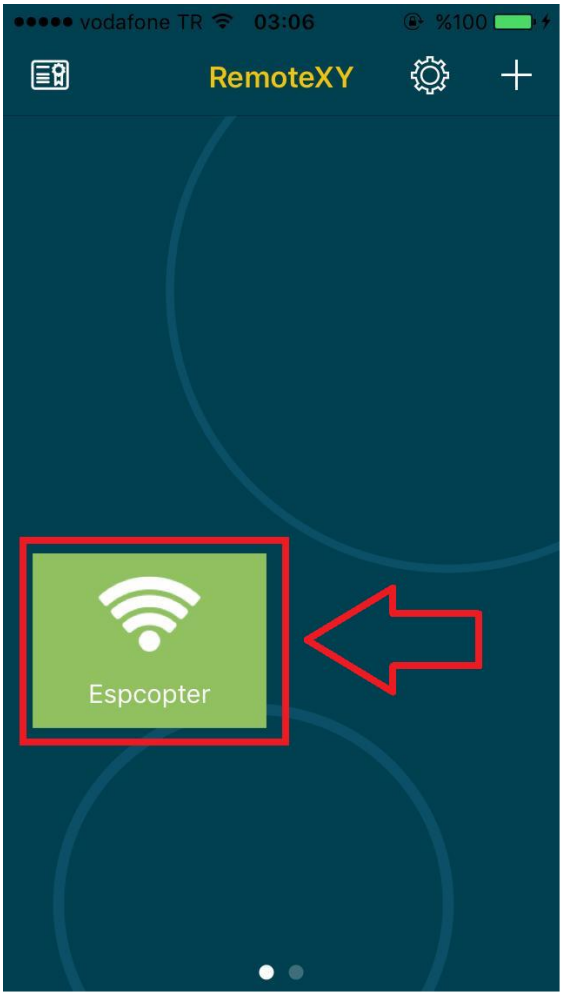
Yazılımların yüklenmesini yaptıktan sonra drone açık konuma getirin. Telefonunuz wifi sekmesinde drone’nun ismini göreceksiniz.

WIFI_SSID “RemoteXY”
WIFI_PASSWORD “12345678”

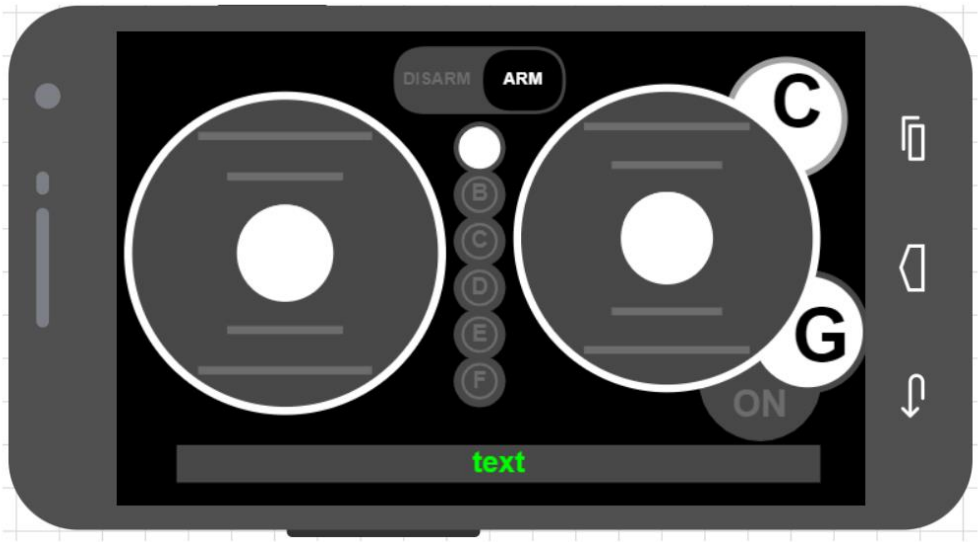
Bağlantıyı yaptıktan sonra telefonunuzdan RemoteXY uygulamasını açınız ve aşağıda bulunan adımları izleyiniz.



İlk bağlantı yapıldıktan sonra RemoteXY ekranında ESPcopter adında bir kısa yol oluşacaktır. Bu kısa yola basarak direk kumandaya girişi yapabilirsiniz.

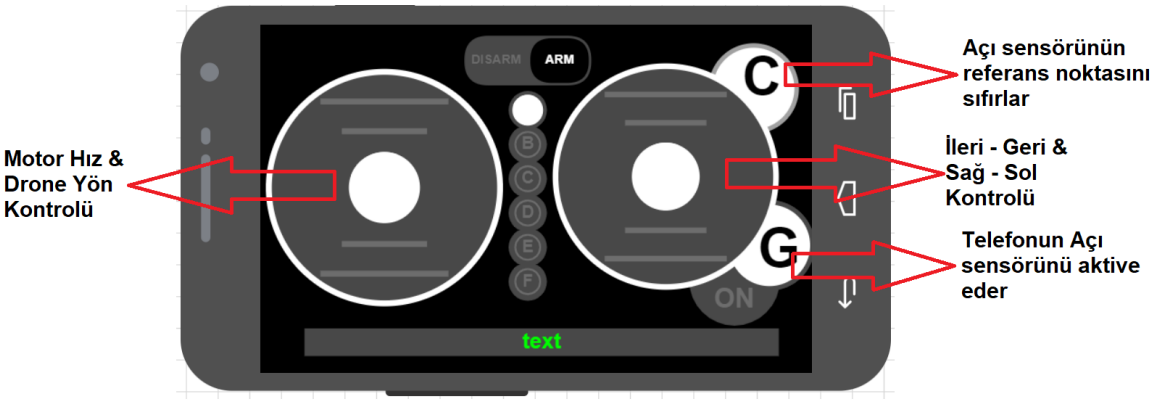


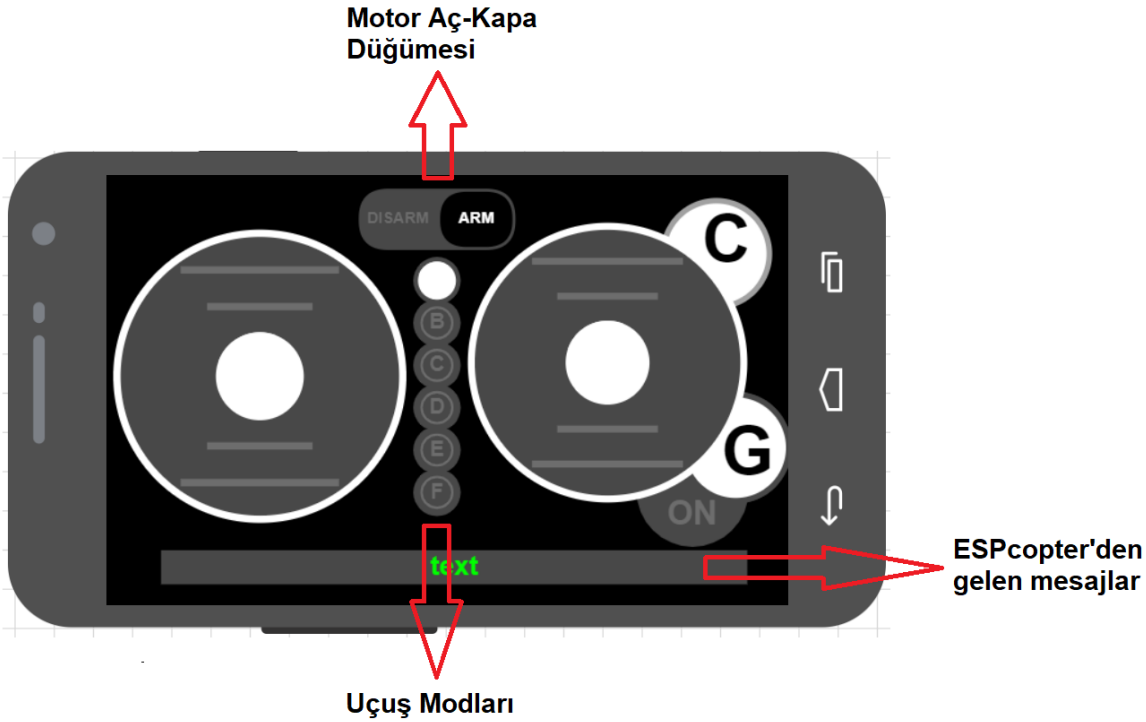
Drone kontrol ekranı otomatik otomatik olarak açılacaktır.



Kumanda İncelemesi:

Joistikler:





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

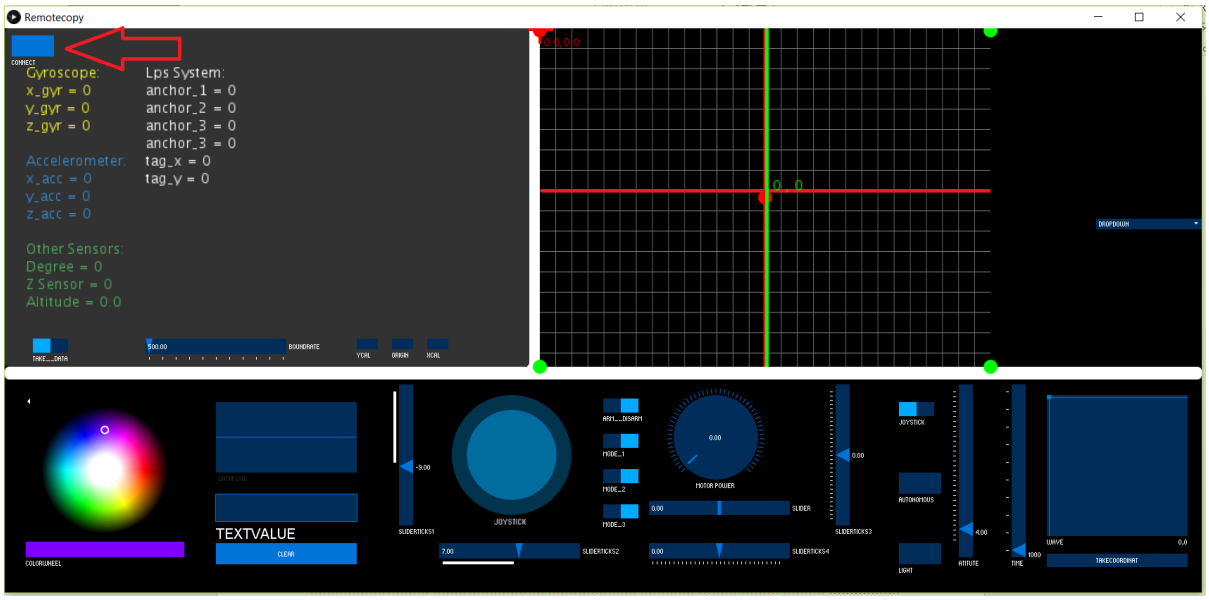
Mod B ve Mod D ikisi bir arada

Bilgisayar İle Kontrol(Processing)

```
#define PROCESSING_WIFI_SSID “ESPcopter”
#define PROCESSING _WIFI_PASSWORD “12345678”
#define PROCESSING_REMOTE
#include <espcopter.h> // library
void setup() {
  mainSetup(); // main flying setup
}
void loop() {
mainLoop ();  // main flying loop
}
```

Bağlantı:

Processing kodunu ESPcopter’e yükeledikten sonra ESPcopter açık konumda iken bilgisayarınız ile ESPcopter’i wiffi ile bağlayınız. Bağlantı tamamlandıktan sonra “Connect” tuşana basarak ESPcopter’I bilgisayarınıza bağlayabilirsiniz



Trim Ayarı:

Trim ayarı drone'nun kalkışta ve uçuşta bir yöne doğru sürekli olarak gitmesi durumunda yapılır. Çoğu drone'da bu ayar kumanda üzerinde ki ayarlardan yapılır. ESPcopter programlanabilir bir drone olduğu için trim ayarı yazılım yüklenerek değiştrilebilir.

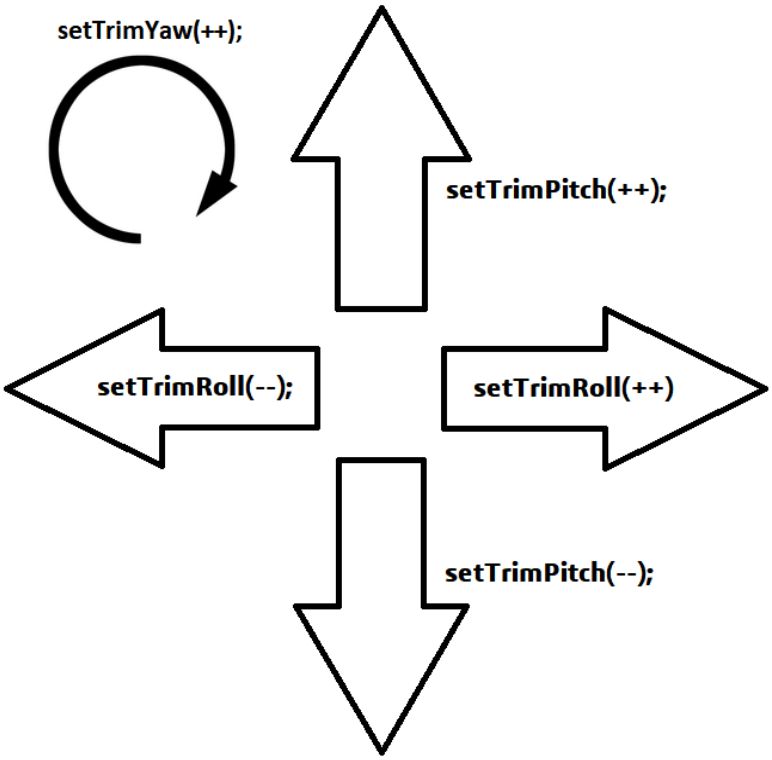
Trim ayarı yapmak için uçuş kodu içerisine "setTrim" fonksiyonlarının yazılması yeterlidir. Fonksiyon içinde ki değer değiştirerek drone'nunuz için en uygun trim değerini bulabilirsiniz. Optik flow modülüne ship olan ESPcopterler trim ayarını otomatik olarak yapabilir. Optik flow modülüne ship olamayan dronelar da bu ayar manuel olarak yapılmalıdır.

setTrimRoll(x); // x değeri -500 / +500

setTrimPitch(x); // x değeri -500 / +500

setTrimYaw(x); // x değeri -2000/ +2000

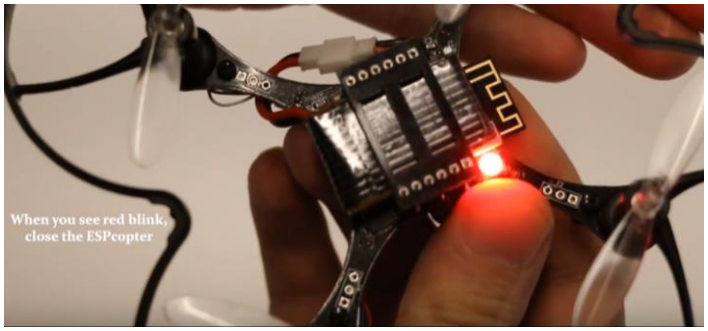
Aşşağıda bulunan görsel setTrim ayarının drone'nun üzerinde ki etkisinin yönlerini göstermektedir. Örneğin, eğer setTrimPitch(x); fonksiyonun daki x değeri artırırsanız. Drone uçarken öne doğru gitmeye başlıyacaktır.



Kalibrasyon:

1- Kalibrasyonu Başlatma

ESPcopter'i ilk çalıştırdığınız da anda kırmızı ışık yanıp sönecektir. Eğer bu aşamada ESPcopter'i kapatırsanız bir sonra ki açılışta kalibrasyon algoritmasını başlatacaktır.



2- Soft Iron Manyetik Alan Kalibrasyonu:

ESPcopter'i açtıktan sonra etrafında metal olmayan düz bir zemine koyunuz. Mavi ışık manyetik alan kalibrasyonun başladığını göstermektedir. Bu aşamada ESPcopter'i olduğu yerde iki tur çevirmeniz gerekmektedir.

3- Hard Iron Manyetik Alan ve Gyro Kalibrasyonu :

Soft iron kalibrasyonunu yaptıktan sonra ESPcopter'in ışığı mor olcaktır. Bu aşamdan sonra ESPcopter'e dokunmayınız. Motorlarını yerden havalanmayacak şekilde bir kaç saniyeliğine çalıştıracaktır.

Özel komutlar:

ESPcopter fonksiyonlar:

Bu liste de bulunan fonksiyonlar kullanılarak ESPcopte üzerinde bulunan led ve motorlar kontrol edilebilir.

Fonksiyonun Yazılışı	Aldığı değerler	Görevi
esp.redLed_Digital();	0 - 1 yada FALSE - TRUE	Kırmızı ledi yakar yada söndürür
esp.blueLed_Digital();	0 - 1 yada FALSE - TRUE	Mavi ledi yakar yada söndürür.
esp.greenLed_Digital();	0 - 1 yada FALSE - TRUE	Yeşil ledi yakar yada söndürür.
esp.redLed_Analog();	0 - 255	Kırmızı ledi istenilen parlaklıkta yakar
esp.blueLed_Analog();	0 - 255	Mavi ledi istenilen parlaklıkta yakar
esp.greenLed_Analog();	0 - 255	Yeşil ledi istenilen parlaklıkta yakar
esp.motorFL_Analog();	0 - 255	Ön sol motoru istenilen güçte çalıştırır.
esp.motorFR_Analog();	0 - 255	Ön sağ motoru istenilen güçte çalıştırır.
esp.motorRL_Analog();	0 - 255	Arka sol motoru istenilen güçte çalıştırır.
esp.motorRR_Analog();	0 - 255	Arka sağ motoru istenilen güçte çalıştırır.

ESPcopter Kontrol Tanımları:

Bu listede bulunan tanımlar kullanılarak ESPcopter'in kontrol yötemi değiştirilebilir. Bu listeden sadece bir tanım aktif edilmesi gerekmektedir.

Komutun Yazılışı	Görevi	Cihaz
#define REMOTE_XY_REMOTE	RemoteXY ile kontrol	Telefon - Tablet
#define BLYNK	BLYNK ile kontrol	Telefon - Tablet
#define PROCESSING_REMOTE	Processing ile kontrol	bilgisayar
#define MQTT	MQTT ile kontrol	bilgisayar
#define REMOTE_WEB_APP	Web uygulaması ile kontrol	Telefon - Tablet
#define PPM_REMOTE	PPM alıcı ile kontrol	Uzaktan kumanda

ESPcopter Değişken Tanımları:

Değişkenin Yazılışı	Görevi	Aldığı değerler
setTrimRoll();	X ekseninde ki hatayı düzetir.	-500 - 500
setTrimPitch();	Y ekseninde ki hatayı düzetir.	-500 - 500
setTrimYaw();	Zekseninde ki hatayı düzetir.	-500 - 500
setArmControl();	Motor Aç-kapa	false - true
setFlyMode_1();	Z eksen sabitlemeyi aç- kapa	false - true
setFlyMode_2();	Yükseklik sabitleme aç- kapa	false - true
setFlyMode_3();	Optik akış modülü ile hareket sabitleme aç- kapa	false - true
landing();	İnişe geçer	false - true
setMotorMax();	Maximum motor gücünü ayarla	600-900
getRX_throttle();	Motor güç değeri	0 – (motorMax)
getRX_roll();	X ekseninde ki kumada değeri	-100 : + 100
getRX_pitch();	Y ekseninde ki kumada değeri	-100 : + 100
getRX_yaw();	Z ekseninde ki kumada değeri	-100 : + 100

Otonom Uçuş Komutları:

Değişkenin Yazılışı	Görevi	Aldığı değerler
takeOff(Y, T);	Komut satırı bir kere çalıştığında drone otomatik olarak kalkış yapar.	Y: 200 - 1000 Yükseklik değeri T: Uçuş süresi
goforward(T);	T Süresi boyunca drone ileri hareket eder .	T: Uçuş süresi
goBack(T);	T Süresi boyunca drone geri hareket eder.	T: Uçuş süresi
goLeft(T);	T Süresi boyunca drone sola hareket eder.	T: Uçuş süresi
goRight(T);	T Süresi boyunca drone sağ hareket eder.	T: Uçuş süresi
turnRight(D);	D açısı kadar kendi çervesinde sağ döner.	D: dönüş açısı
turnLeft(D);	D açısı kadar kendi çervesinde sola döner.	D: dönüş açısı
delay_(T);	Hareketler arası bekleme yapamaya yarar. Her	T: Bekleme süresi

	komuta arasına mutlaka yazılmalıdır	
Land();	Otonom uçuş yazılımının sonuna yazılmak zorundadır.	

Yüklseklik Sabitleme Modülü:

Değişkenin Yazılışı	Görevi	Aldığı değerler
setVl5310xControl ();	vl5310x modülü aç- kapa	0 - 1
setTargetOto();	vl5310x modülü ile yükseklik sabitle	250 - 1000
getOtoMeasure	Drone'nun yükselik verisi	0- 1000

Buzzer Modülü:

Değişkenin Yazılışı	Görevi	Aldığı değerler
esp.buzzer();	0 - 1 yada FALSE - TRUE	Buzzer modülünü çalıştırır yada kapatır.

Neopixel Modülü:

Değişkenin Yazılışı	Görevi	Aldığı değerler
#define NeoPixel	Neopixel modülünü açıp kapatır	TANIM
ESPrainbow();	Gök kuşağı efekti yapar	Yok(Loop içine yazılır)
ESPsetPixel (x,r,g,b);	Modül üzerinde ki her bir ledi ayrı ayrı kullnmak için kullanılır.	X= 1 – 12 R(KIRMIZ)= 0 - 255 G(Yeşil)= 0 - 255 B(Mavi)= 0 - 255
ESPpixelShow();	Yapılan değişiklikleri uygular.ESPsetPixel(); komutu ile birlikte kullanılır.	

Optik Akış Modülü:

Değişkenin Yazılışı	Görevi	Aldığı değerler
SetPointOpt[0]	Drone'nun harketini optic akış sensörünü kullarak set eder. Eğer bu değer sıfıra eşitlenirse drone x	-15 - +15

	ekseninde sabit durur. +15 eşitlenirse sağ gider, -15 e eşitlenirse sola gider	
SetPointOpt[1]	Drone'nun harketini optic akış sensörünü kullarak set eder. Eğer bu değer sıfıra eşitlenirse drone y eksenin de sabit durur. +15 eşitlenirse ileri gider, -15 e eşitlenirse geri gider	-15 - +15
deltaCalX	Optik akış sönsöründen gelen x ekseni akış verisi.	Drone'nun yere göre hızına bağlı
deltaCalY	Optik akış sönsöründen gelen y ekseni akış verisi.	Drone'nun yere göre hızına bağlı

Çoklu Mesafe Modülü:

Değişkenin Yazılışı	**Görevi**	**Aldığı değerler**
#define HandControl #define AntiCollision	El ile kontrol yada çarpışma önleme sistemi	Tanım
Distance_Y_1();	Y(+) ekseni mesafe verisi	50-1000
Distance_Y_0();	Y(-) ekseni mesafe verisi	50-1000
Distance_Y_1();	X(+) ekseni mesafe verisi	50-1000
Distance_Y_0();	X(-) ekseni mesafe verisi	50-1000

Diğer Komutlar

Arduino ve ESP8266 kütüphanesinde ki aşağıda ki listed bulunan komutlar hariç hepsi kullanılabilir.

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