

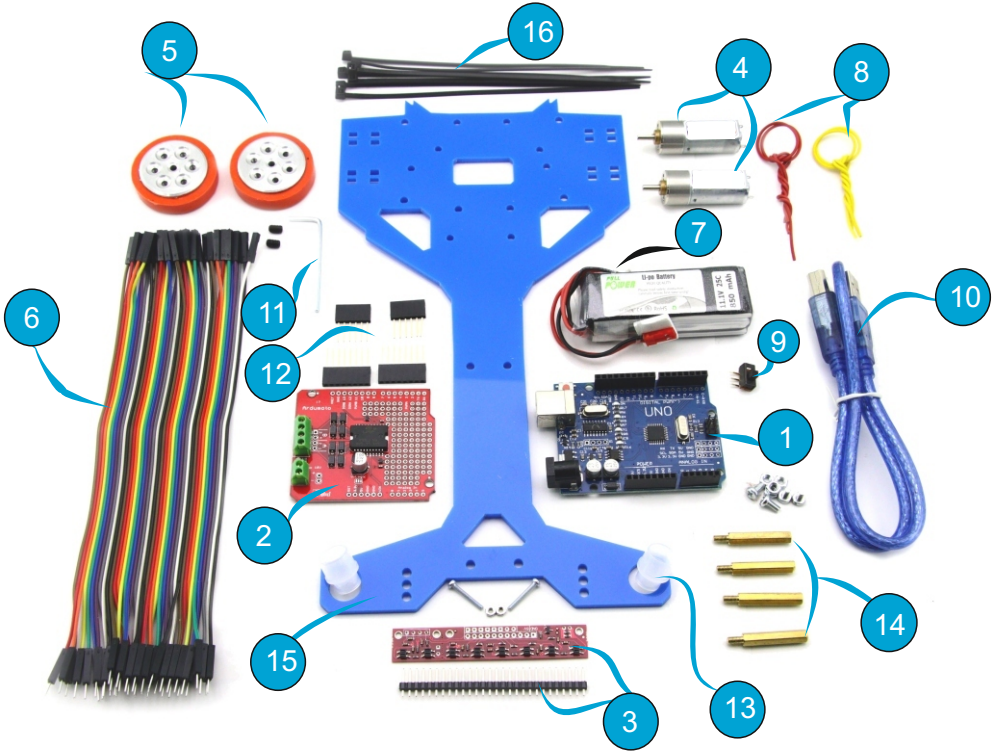
# FLİNE ARDUİNO ÇİZGİ İZLEYEN ROBOT KİTİ

## Fline Arduino Çizgi İzleyen Robot Geliştirme Kiti Montaj Klavuzu



**MONTAJ KLAUVUZU**

# Fline Çizgi İzleyen Robot Montaj Klavuzu



Fline Çizgi İzleyen Robot; 8 Sensörlü, PID programlı, Hız Kontrolü Yapılabilen Gelişmiş Çizgi İzleyen Robot Kitidir.

## Malzemeleri Tanıyalım

(1)-**Arduino Uno** Robotumuzun beyni yani kararları veren kontrol kısmıdır. Bilgisayardan USB kablo ile Arduinonun kendi dilini kullanarak (temel İngilizce komutlar) programlıyoruz. Arduino Uno en popüler ve yaygın bulunan kontrol kartlarından biridir.

(2)-**ArduMoto**: Arduino kartlarla doğrudan takılarak kullanılmaktadır. L298P motor sürücü entegresini kullanmaktadır. İki motoru bağımsız olarak sürebilmektedir.

(3)-**QTR-8RC Çizgi Sensör** Çizgi izleyen robotların vazgeçilmez sensörüdür. Üzerinde 1cm arayla yerleştirilmiş 8 IR LED/fototransistör çifti barındırmaktadır.

# Fline Çizgi İzleyen Robot Montaj Klavuzu

(4)-**ProFAST 12V 1500Rpm Dc Motor**Bu motorların çıkış milleri dakika da 1500 devir döner.Dişli kutusu düşürme oranı 10:1'dir.Mil çapı 3mm'dir.Çizgi izleyen robotlarda yüksek performans verir.

(5)-**Elektron Silikon Tekerlek:** Robotumuzda 38mm çapında,10mm uzunluğunda tekerlekler kullanıyoruz.Özel üretimimiz olan silikonları sayesinde zemine tam tutunma sağlamaktadır.

(6)-**Jumper Kablo:**Elektroniksel bağlantıları yapmak için kullanacağız.

(7)-**11,1V Lipo Batarya 850mAh 25C:**Yüksek verim alabilmek için robotumuzda 3s lipo batarya kullanıyoruz.

(8)-**Motor Bağlantı Kabloları:** Motorlar ile Ardumoto motor sürücü arasındaki bağlantıları sağlayacağımız kablolar.

(9)-**On/Off Anahtar:** Robotumuza giden güç akışını kontrol etmemizi sağlayan açma kapama anahtarıdır.

(10)-**USB Kablo**Arduino ile pc arasındaki iletişimi sağlayacak olan kablo.

(11)-**Alyan ve Setskur:**Tekerleklerin motorlar ile birleşiminde sıkıştırılması için kullanılır.

(12)-**Header**Ardumoto ile arduino arasındaki iletişimin kurulması için gerekli headerlar.

(13)-**Sarhoş Tekerlek:** Robotumuzun ön sağ ve ön sol taraflarına takılarak robotun dengede kalması ve kusursuz dönüşler yapmasını sağlar.

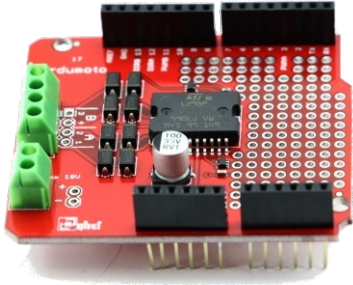
(14)-**Distance**Robot devremizin gövdeye sabitlenmesi için gerekli aparatlar.

(15)-**Gövde**Robotumuzun gövdesi.

(16)-**Plastik Kelepçe**Motorları gövdeye sabitlemek için kullanılacak.

Kullanacağımız malzemeleri öğrendik.Şimdi Ardumotoya daha da yakından bakalım.

## ARDUMOTO



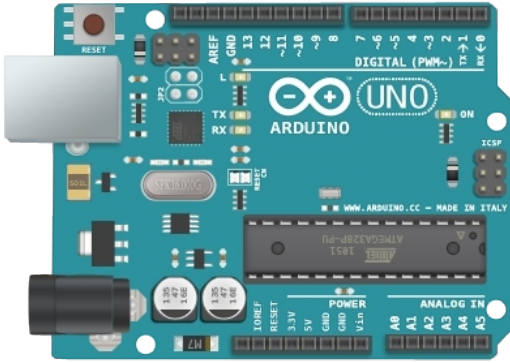
Ardumoto üzerinde motor kontrolüne yardımcı klemensler,motor sürücü ve giriş çıkış elemanları barındıran bir yardımcı motor sürücü shielddir. Arduino kartlarla doğrudan takılarak kullanılmaktadır. L298P motor sürücü entegresini kullanmaktadır.İki motoru bağımsız olarak sürebilmektedir.

# Fline Çizgi İzleyen Robot Montaj Klavuzu

## Nasıl Programlayacağız ?

Çizgi izleyen robotun ana kontrolcüsü Arduino Uno olması nedeniyle kolayca programlanabilir.Arduino Uno, Arduino.cc sitesinden indirebileceğiniz Arduino progtamı ile programlanmaktadır.Bu nedenle önce arduino programını bilgisayarınıza kurmanız gerekmektedir. Kurulum esnasında Arduino driverları da bilgisayara yüklenecektir.Kartı bilgisayarınıza taktıktan sonra Araçlar yada Tools sekmesinden kart modelinizi(Arduino Uno-Geniuno) ve bağlantı portunuzu seçtikten sonra Arduinonuzu programlayabilirsiniz.Eğer arduino konusunda kendinizi geliştirmek isterseniz, **Arduinoyu 25 Proje ile Keşfet** kitabını tavsiye ederiz.

## Arduino Nedir ve Nasıl Kurulur ?



Arduino, açık kaynak kodlu bir mikrodenetleyici kartıdır. Ana MCU(Microprocessor Control Unit - Mikroişlemcili Kontrol Ünitesi), Atmel tabanlıdır.

(MCU ya önceden bir mini program - bootlader yüklenmiştir). Kart ile robotik ve elektronik uygulamalarınızı kolayca gerçekleştirebilirsiniz.

Bu kadar popüler olmasının bir kaç nedeni vardır. Bunlar:

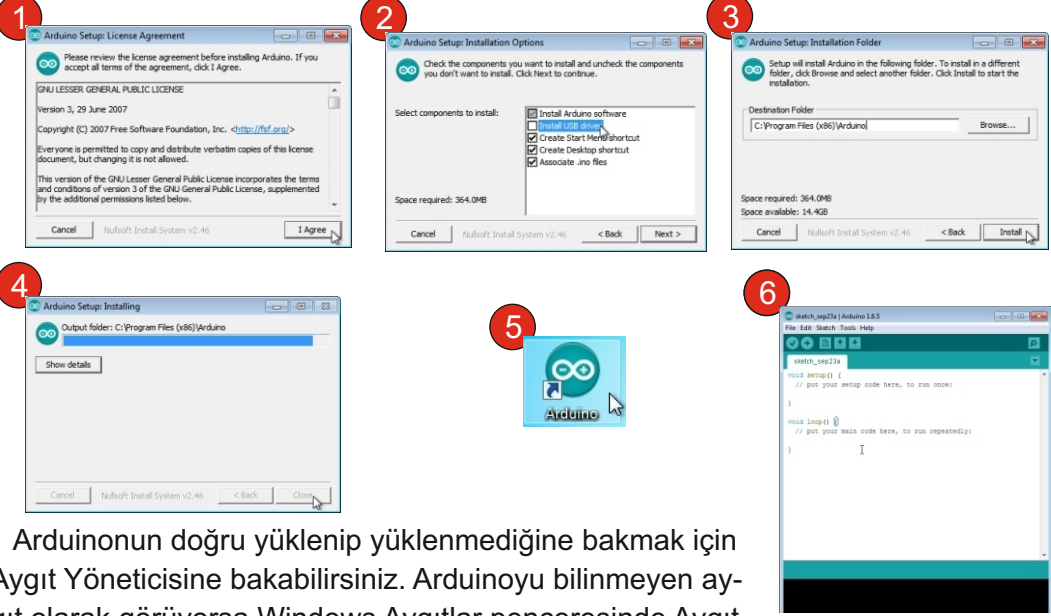
- Açık kaynak kodludur.(Devre şemasından, programlama arayüzüne kadar...)
- Program geliştirmek basittir. Gerçekten çok basittir. Aynı kart üzerinden hem programlanır hem test edilebilir.
- Arduinoda kullanılan programlama dili de basittir. Bolca örneği mevcuttur.
- Ve en önemlisi çok yaygın kullanıcısı mevcuttur. Yani büyük olasılıkla yapmak istediğiniz projelerin benzerleri de önceden yapılmıştır. Diğer proje yapımcılarıyla daha kolay bilgi alışverişinde bulunabilirsiniz.

Arduino hakkında temel bilgileri de aldık.

Sıra Arduino yazılımının bilgisayarımıza kurulumuna bakalım.

# Fline Çizgi İzleyen Robot Montaj Klavuzu

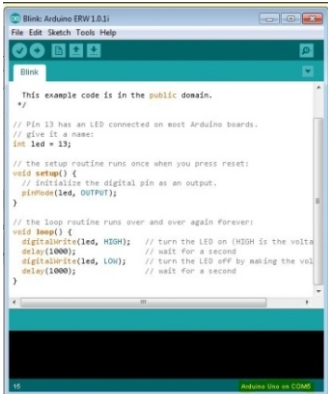
Arduinonun IDEsini <http://arduino.cc/en/Main/Software> linkinden işletim sisteminize göre en son versiyon programını indirebilirsiniz. Programı indirdikten sonra Arduinoyu bilgisayarımıza aşağıdaki adımları takip ederek kuruyoruz.



Arduinonun doğru yüklenip yüklenmediğine bakmak için Aygıt Yöneticisine bakabilirsiniz. Arduinoyu bilinmeyen aygıt olarak görüyorsa Windows Aygıtlar penceresinde Aygıt üzerine sağ tıklayın, özelliklere gidin sonrasında, Donanım sekmesine ve sonra da özellikler butonuna tıklayıp sürücü güncelleştirmesi yapın. Soru büyük olasılıkla çözülecektir.

## Arduino IDE Programı

Bu program ile Arduinoya yükleyeceğimiz yazılımları oluşturuyoruz. Program görüntüsü yan tarafta yer almaktadır. Son derece basit bir arayüzü vardır. Arduino kartınıza program yüklemeyen önce (Upload) Tool / Board seçeneğinden doğru kartı seçtiğimize emin olunuz. Karta programın derlenip yüklenmesi için Yatay okun yanındaki Tick butonu programımızı derler bu şekilde herhangi bir sorun olup olmadığını yüklemeyen önce görebilirsiniz



# Fline Çizgi İzleyen Robot Montaj Klavuzu

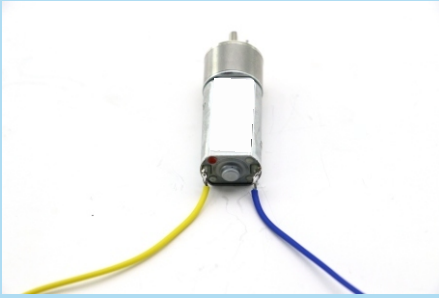
## Montaj Araç ve Gereç

- Lehim Teli
- Havya
- Yan Keski
- Tornavida
- Silikon

## Montaja Başlayalım

Motorların arka uçlarına kablo lehimleyiniz. Kabloları lehimledikten sonra birbirlerine sarabilirsiniz. Bu motorları sonrasında Ardumotonun motor bağlantı klemenslerine (SolMotor=B - SağMotor=A) bağlayacağız.

1

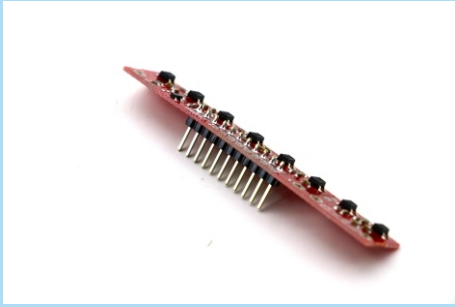


2

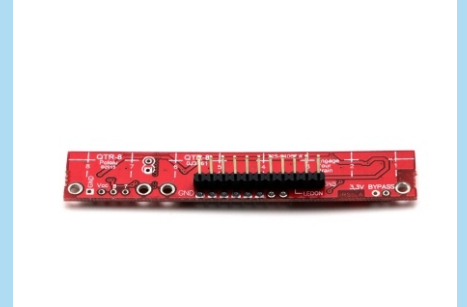


Resim 2'de görüldüğü üzere ardumotonun üzerindeki pinlere headerlarımızı lehimliyoruz. Lehimlediğimiz bu headerlar sayesinde ardumoto ile arduino uno bağlantımızı kolay bir şekilde yapacağız.

3



4



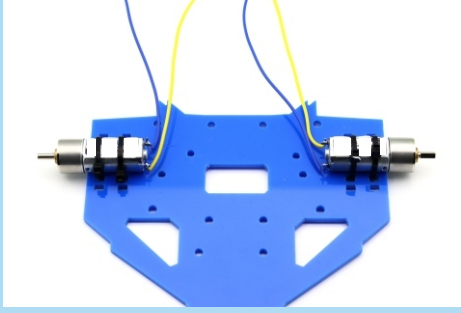
Ardumoto headerlarını lehimledikten sonra robotumuzun çizgi takibinde kullandığımız QTR8-RC çizgi sensörünün headerlarını lehimliyoruz.



# Fline Çizgi İzleyen Robot Montaj Klavuzu

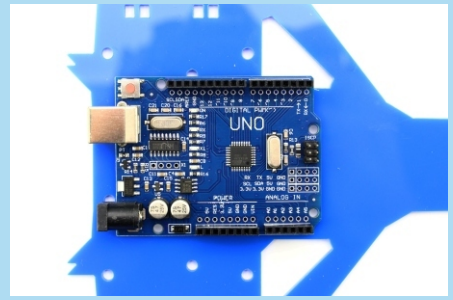
Lehimleme işlemini bitirdikten sonra bağlantı kablolarını lehimlediğimiz motorları gövdemize, plastik kelepçe yardımıyla Resim 5'te görüldüğü gibi sabitliyoruz.

5



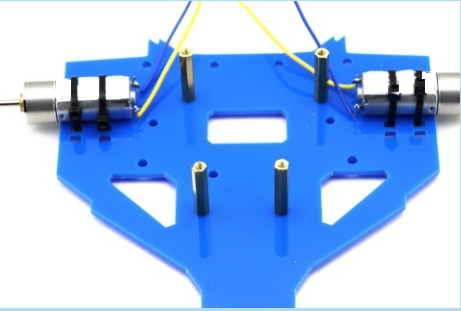
ÖNEMLİ

!

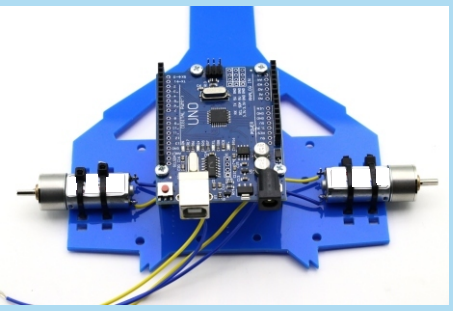


Burada dikkat etmemiz gereken nokta gövdemizin ön yüzünü belirlemek. Arduino Unomuzun montaj delikleri ile gövdemizin deliklerinin uyuşması gerekmektedir. Sağ üstteki resimdeki gibi gövdemizin ön yüzünü belirleyebilirsiniz.

6



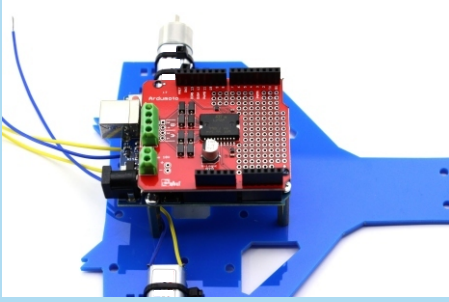
7



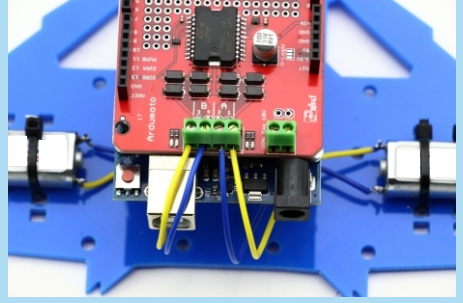
Motorlarımızı sabitledikten sonra arduinomuzu gövdemize sabitlemek için gerekli olan distansları civatalar yardımıyla montaj deliklerine takıyoruz. Ardından arduino unomuzu distanslar üzerine resim 7'deki gibi sabitliyoruz.

# Fline Çizgi İzleyen Robot Montaj Klavuzu

8

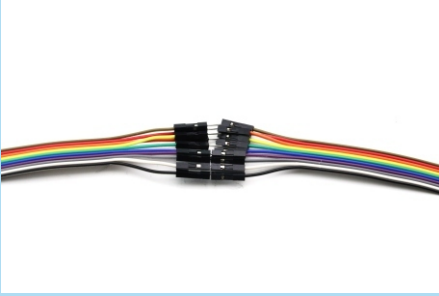


9

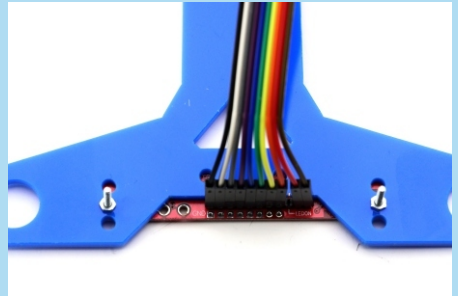


Sabitlediğimiz arduino uno'nun üzerindeki pinlere geçiriyoruz. Ardından motor bağlantı kablolarını Resim 9'daki gibi motor klemenslerine Sol motor B, Sağ motor A'ya gelecek şekilde takıp sıkıştırıyoruz.

10



11

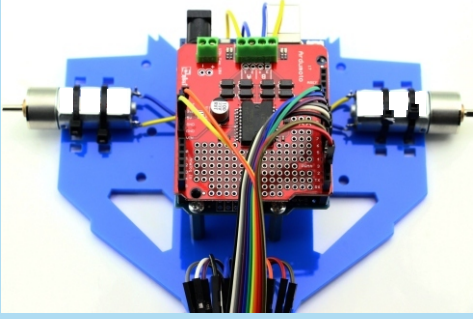


Motor kablolarını klemenslere sıkıştırdıktan sonra çizgi sensörünün bağlantısı için Resim 10'daki gibi kabloları birbirine geçiriyoruz. Ardından gövdemizin ön kısmına çizgi sensörümüzü vida ve somunlar yardımı ile montaj deliklerinden çizgi sensörleri robotun altına gelecek şekilde sabitliyoruz. Sonrasında birbirine eklediğimiz kabloları çizgi sensörü pinlerine takıyoruz. Kabloları takarken **Ledon** pininin boş olmasına dikkat edelim.

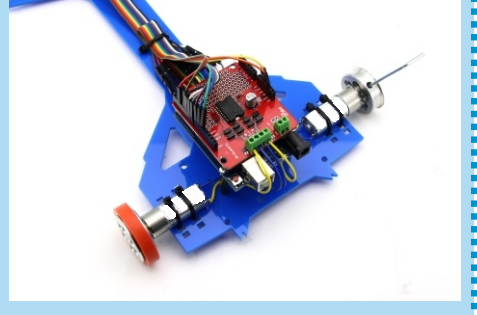


# Fline Çizgi İzleyen Robot Montaj Klavuzu

12



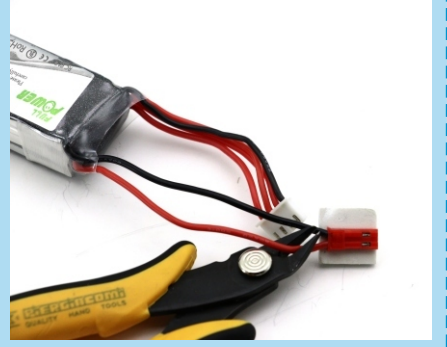
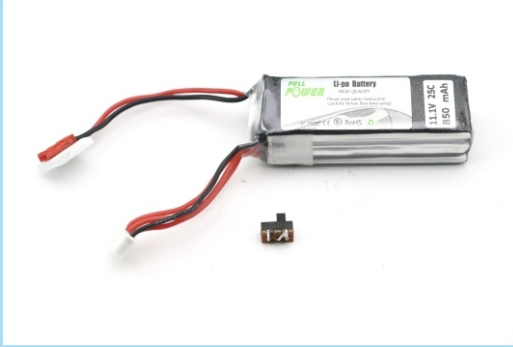
13



QTR 8RC'nin kablolarında **5v** ve **GND** pininden gelenleri arduino üzerindeki **5v** ve **GND** pinlerine, daha sonrasında da sırasıyla **digital 2.pin**den başlayarak takıyoruz. 3.pinin boş olmasına dikkat edelim. Ardından silikon tekerleklerimizi Resim 13'de görüldüğü gibi alıyın ve setskur yardımı ile motorlarımıza takıp sıkıştırıyoruz.

**ÖNEMLİ**

14



Robotumuzun güç akışını kontrol etmek için lipo bataryamıza on/off anahtarı ekliyoruz. Burada dikkat etmemiz gereken bir nokta ise lipo batarya kablolarını aynı anda kesmemek. Eğer aynı anda kesersek pilimiz kısa devre olacaktır. Bunun için resimde görüldüğü üzere ayrı ayrı kesiyoruz.

# Fline Çizgi İzleyen Robot Montaj Klavuzu

15

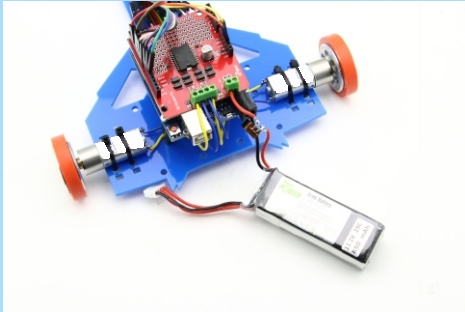


16

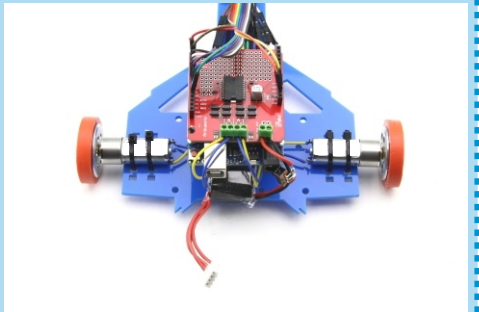


Kabloları kestikten sonra kırmızı kabloyu Resim 15'deki gibi açma kapama anahtarının ayaklarına lehimliyoruz. Bu arada siyah kabloyla kırmızı kabloyu kısa devre olmaması için birbirine değdirmemeye dikkat edelim. Ayaklara kabloları lehimledikten sonra siyah kablo ile kırmızı kablo ucunu Resim 17'deki gibi power klemenslerine yerleştirip tornavida yardımı ile sıkıştırıyoruz.

17



18



Power klemenslerine yerleştirdikten sonra lipo bataryamızı gövdemiz ile arduino unomuzun arasına yerleştirebiliriz.

**Tebrikler montajınızı başarıyla tamamladınız.**

Şimdi robotumuzun program kısmına geçelim.

# Şimdi Kodlamayı Öğrenelim !

`#include <QTRSensors.h>/QTR Sensors kütüphanesi eklendi.`

`#define sagtabanhiz 200 //Sağ motorun taban hızı 200 olarak tanımlandı.`

`#define soltabanhiz 200 //Sağ motorun taban hızı 200 olarak tanımlandı.`

`#define sagmotoryon 13 //Sağ Motor yön Arduinonun 13 numaralı pinine tanımlandı.`

`#define sagmotorpwmpin 11 //Sağ Motor Pwm Arduinonun 11 numaralı pinine tanımlandı.`

`#define solmotoryon 12 //Sol Motor Yön Arduinonun 12 numaralı pinine tanımlandı.`

`#define solmotorpwmpin 3 //Sol Motor Pwm Arduinonun 3 numaralı pinine tanımlandı.`

`int sensor=10;//sensor adlı değişken Arduinonun 10 numaralı pinine tanımlandı.`

`QTRSensorsRC qtrrc((unsigned char[]) { 2, 4, 5, 6, 7 , 8, 9, 10} ,8, 2500,  
QTR_NO_EMITTER_PIN);qtr sensör pinleri tanımlandı.`

`unsigned int sensorValues[8]; //sensorValues adında 8'li dizi oluşturuldu.`

`void setup()`

`{`

`delay(2000); //2000 milisaniye beklendi.`

`pinMode(sensor, INPUT); //sensor pini giriş olarak ayarlandı.`

`pinMode(sagmotoryon, OUTPUT); //sagmotoryon pini çıkış olarak ayarlandı.`

`pinMode(sagmotorpwmpin, OUTPUT); //sagmotorpwmpin pini çıkış olarak ayarlandı.`

`pinMode(solmotoryon, OUTPUT); //solmotoryon pini çıkış olarak ayarlandı.`

`pinMode(solmotorpwmpin, OUTPUT); //solmotorpwmpin pini çıkış olarak ayarlandı.`

`int i; //i Adında integer türünde değişken oluşturuldu.`

`digitalWrite (13,HIGH); //13.pin aktif yapıldı.`

`for (int i = 0; i < 200; i++) //for döngüsü oluşturuldu. int i sıfıra eşitlendi. i değişkeni  
200'den küçük olduğu hallerde i'yi bir arttır koşulları yazıldı.`

`{`

`if( 0 <= i && i < 5 ) hafifsagadon();//eğer i 0'dan büyük eşitse ve 5'ten küçükse  
hafifsagadon metodu çalışacak.`

`if( 5 <= i && i < 15 ) hafifsoladon();//eğer i 5'ten büyük eşitse ve 15'ten küçükse  
hafifsoladon metodu çalışacak.`

# Şimdi Kodlamayı Öğrenelim !

`if ( 15 <= i && i < 25 ) hafifsagadon();` //eğer i 15'ten büyük eşitse ve 25'ten küçükse hafifsagadon metodu çalışacak.

`if( 25 <= i && i < 35 ) hafifsoladon();` // eğer i 25'ten büyük eşitse ve 35'ten küçükse hafifsoladon metodu çalışacak.

`if( 35 <= i && i < 45 ) hafifsagadon();` //eğer i 35'ten büyük eşitse ve 45'ten küçükse hafifsagadon metodu çalışacak.

`if( 45 <= i && i < 55 ) hafifsoladon();` //eğer i 45'ten büyük eşitse ve 55'ten küçükse hafifsoladon metodu çalışacak.

`if( 55 <= i && i < 65 ) hafifsagadon();` //eğer i 55'ten büyük eşitse ve 65'ten küçükse hafifsagadon metodu çalışacak.

`if( 65 <= i && i < 75 ) hafifsoladon();` //eğer i 65'ten büyük eşitse ve 75'ten küçükse hafifsoladon metodu çalışacak.

`if( 75 <= i && i < 85 ) hafifsagadon();` //eğer i 75'ten büyük eşitse ve 85'ten küçükse hafifsagadon metodu çalışacak.

`if( 85 <= i && i < 90 ) hafifsoladon();` //eğer i 85'ten büyük eşitse ve 90'dan küçükse hafifsoladon metodu çalışacak.

`if ( i >= 90 ) {frenle(); delay(5);}` //eğer i 90'dan büyük eşitse frenle metodu çalışacak ve 5 milisaniye beklenecek.

`qtrrc.calibrate();` //qtrrc kalibre edildi.

`delay(4);` //4 milisaniye beklendi.

}

`digitalWrite (13,LOW);` //13.pin pasif yapıldı.

`delay(2000);` //2000 milisaniye beklendi.

`Serialbegin (9600);` //Seri haberleşme başlatıldı.

`int sonhata = 0;` //integer türünde sonhata adında değişken oluşturuldu ve 0'a eşitlendi.

`float Kp = 0.06;` //Float türünde Kp adında değişken oluşturuldu ve 0.06'ya eşitlendi.

`float Kd = 0.8;` // Float türünde Kd adında değişken oluşturuldu ve 0.8'e eşitlendi.

`int sagmotorpwm = 0;` //sagmotorpwm değişkeni oluşturuldu ve 0'a eşitlendi.

`int solmotorpwm = 0;` //solmotorpwm değişkeni oluşturuldu ve 0'a eşitlendi.

`int zemin=0;` //zemin değişkeni oluşturuldu ve 0'a eşitlendi.

# Şimdi Kodlamayı Öğrenelim !

```
void loop()
{
  // Pozisyon hesabı QTRlibrary tarafından yaptırılıyor
  unsigned int sensorValues[8];
  unsigned int position = qtrrc.readLine(sensorValues,1,zemin);
  int hata = position-3500;

  if ( sensorValues[0]<100 && sensorValues[7]<100 ) { zemin=0; } //Beyaz zemin- siyah çizgi yani eğer sensorValues[0] değeri 100'den küçük ve sensorValues[7] değeri 100'den küçükse zemin değişkeni 0'a eşitlendi.
  if ( sensorValues[0]>500 && sensorValues[7]>500 ) { zemin=1; } //siyah zemin - beyaz çizgi yani eğer sensorValues[0] değeri 500'den büyükse ve sensorValues[7] değeri 500'den büyükse zemin değişkeni 1'e eşitlendi.

  //////////// motorlara verilecek hız düzeltme oran hesabı
  int duzeltmehizi = Kp * hata + Kd*(hata - sonhata);
  sonhata = hata;//sonhata adlı değişken hata adlı değişkene eşitlenir.

  //////////// Motorlara uygulanacak kesin hız ayarları
  sagmotorpwm = sagtabanhiz + duzeltmehizi ;
  solmotorpwm = soltabanhiz - duzeltmehizi ;

  //// Burada motorlara uygulanacak PWM değerlerine sınırlandırma getirilmiştir.
  sagmotorpwm = constrain(sagmotorpwm, -100, 110);
  solmotorpwm = constrain(solmotorpwm, -100, 110);
  motorkontrol(sagmotorpwm,solmotorpwm);

  // Seri monitörde hata, sağ ve sol motor hız değerlerini göstermek için alttaki satırı aktif yapınız
  // Serial.print(hata); Serial.print(" "); Serial.print(sagmotorpwm);
  //Serial.print(" "); Serial.println(solmotorpwm); delay(100);

}
```

# Şimdi Kodlamayı Öğrenelim !

// Motor kontrol alt programı

```
void motorkontrol( int sagmotorpwm, int solmotorpwm){ //sagmotorpwm ve solmotorpwm parametrelili motorkontrol metodu oluşturuldu.
```

```
if(sagmotorpwm<=0) { //eğer sagmotorpwm küçük eşitse 0'a aşağıdaki kod bloğu çalışacak.
```

```
    sagmotorpwm=abs(sagmotorpwm);  
    digitalWrite(sagmotoryon, LOW);  
    analogWrite(sagmotorpwmpin, sagmotorpwm);
```

```
}
```

```
else { //yukarıdaki şart doğru değilse aşağıdaki kod bloğu çalışacak.
```

```
    digitalWrite(sagmotoryon, HIGH);  
    analogWrite(sagmotorpwmpin, sagmotorpwm);
```

```
}
```

```
if(solmotorpwm<=0) { //eğer solmotorpwm küçük eşitse 0'a aşağıdaki kod bloğu çalışacak.
```

```
    solmotorpwm=abs(solmotorpwm);  
    digitalWrite(solmotoryon, LOW);  
    analogWrite(solmotorpwmpin, solmotorpwm);
```

```
}
```

```
else { //yukarıdaki şart doğru değilse aşağıdaki kod bloğu çalışacak.
```

```
    digitalWrite(solmotoryon, HIGH);  
    analogWrite(solmotorpwmpin, solmotorpwm);
```

```
}
```

```
}
```

```
void frenle(){motorkontrol(0,0);} //frenle metodu oluşturuldu .
```

```
void hafifsagadon(){motorkontrol(-100,100);} //hafifsagadon metodu oluşturuldu .
```

```
void hafifsoladon(){motorkontrol(100,-100);} //hafifsoladon metodu oluşturuldu.
```



# Program Sözlüğü

**void** setup(); - Reset butonuna bastıktan sonra rutin bir şekilde bir kez çalışır. Bazı ayarlar ve tanımlar burada yapılır.

**void** loop(); - Programımızın çalıştığı döngüdür.

**int** değişkenismi; - Oluşturulan değişken boyutu 4 byte'tır ve virgülden sonrasını da kapsar. Program ondalık değişkenler için kullanılır ve aynı zamanda pin tanımlamada kullanılır.

**pinMode**(pinadı,OUTPUT/INPUT); - Kullanacağımız pinin çıkış(OUTPUT) yada giriş (INPUT) olarak ayarlamamızı sağlar.

**Serialbegin** (9600); - Bilgisayar ile arduino arasında seri iletişim başlatmak için **void** setup() fonksiyonu altında seri iletişim açılır. Buradaki 9600 alınan ve gönderilen bilgilerin, verilerin hızıdır. Arduino tarafına 9600 baud rate yazdıysak bilgisayar tarafında da aynı baud rate ile çalışmalıyız. Çünkü veri kaybı yaşarız. 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 ve 115200 baud ratelerden birini seçebilirsiniz. Seri iletişim halindeyken arduinodaki rx, tx, yani digital 0,1 pinleri kullanılmaz.

**Serialprint** (); - Serial monitöre yanyana yazı yazdırdığımız komut.

**Serialprintln** (); - Serial.print ile tek farkı bu komutla ekrana yazdırma işlemi bittikten sonra alt satıra geçer.

**delay**();Parantez içine yazılan süre boyunca bekletir. Mili saniye cinsinde yazılır. (1000 milisaniye 1 saniye);

**digitalWrite** (pinadı,HIGH/LOW); - Çıkış olarak tanımlanan pine 5V( ) veya 0V ( ) yapmamızı sağlar.

**analogWrite** (pinadı,değer); - Pwm pinlerinde yazılan değere göre çıkış sağlar. Değer aralığı 0-255dir.

**if**(koşul){} - **else**{}

 - İf karşılaştırmak için kullanılan bir operatördür. İf in içindeki koşul sağlanır ise o süslü parantez arasındaki kodlar çalışır. Koşul sağlanmaz ise elsenin süslü parantez içindeki kodu çalışır.

# Fline Çizgi İzleyen Robot Montaj Klavuzu



## Arduino Programı

Resmi arduino yazılımını arduinonun web sitesinden ücretsiz bir şekilde indirip bilgisayarınıza kurabilirsiniz. Kurulum dosyası driver yüklemek istediğinde onaylayınız. Download <https://www.arduino.cc/en/Main/Software> sayfası linkindedir. Aşağıdaki QR kodunu okutarak arduino ile ilgili daha fazla bilgi alabilirsiniz.

## Robotunuzu Nasıl Geliştirebilirsiniz ?

ESP8266 wifi modülü ile aracınızı internet ortamına bağlayarak internet olan her yerde aracınızın kontrolünü sağlayabilirsiniz. Bu bizim verebileceğimiz bir örnek, gerisi sizin hayal gücünüze kalmış :)