

**T.C.**  
**MİLLİ EĞİTİM BAKANLIĞI**  
**SİVAS MERKEZ**  
**Sivas Fen Lisesi**  
**Bilgisayar Bilimi Dersi**  
**Rapor**

<b>Rapor No</b>	Proje-1
<b>Rapor Tarih</b>	09/01/2018
<b>Proje Adı</b>	Arduino ile Çizgi Takip Eden Robot

**Bilgisayar Bilimi Öğretmeni**  
**Ersin TÜTÜNCÜ**  
**2017-2018**



**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Proje Grup**

<b>Proje Görev</b>	<b>Numara</b>	<b>Ad-Soyadı</b>
Proje Yönetimi	235	Ceren AĞBABA
Doküman Yönetimi	346	Hatice Şura AYDOĞDU
Lojistik Yönetim	478	Mehmet Tekin ÇATALTEPE
Yazılım Geliştirme	315	Elif YÜCEL
Web ve GitHub Yönetimi	33	İsra AYDOĞDU
Sunum Yönetimi	216	Sudenaz ÜNAL

ÖZET .....	4
Anahtar Kelimeler .....	4
ABSTRACT .....	4
Key Words .....	4
Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması.....	5
1. Görev Dağılımı ve Sorumlusu.....	5
2. Görev Süresince Sürdürülen Eylemler .....	6
3. Görevlerin İş Yükü Şeması .....	7
4. Yoklama Çizelgeleri .....	8
5. Haftalık İş Katkı Cetvelleri .....	9
GİRİŞ .....	9
Projenin Açıklaması .....	9
YAPILAN ÇALIŞMALAR .....	10
1. Donanım Yapısı .....	10
a. Gömülü Sistemler Mimarisi ve Devre Tasarımı .....	10
b. Mekanik Sistem Mimarisinin Tanıtılması:.....	14
2. Yazılım Yapısı.....	21
a. Algoritma .....	21
b. Kodlama.....	22
SONUÇ .....	27
1. Bilgi Düzeyine Katkıları.....	27
2. Teknolojik Katkıları.....	27
3. Ekip Çalışmasının Katkıları .....	27
4. Aksayan Yönler.....	28
5. Görüş ve Öneriler.....	28
6. GitHub Adresi.....	29

## **ÖZET**

Birinci proje kapsamında kullanılan elemanlar ile çizgi takip eden robot yapılacaktır. Bu projeden belirlenmiş olan çizgiyi sorunsuz ve istenen bir şekilde izleyerek pisti tamamlaması beklenmektedir. Bu amaçla kullanılacak olan QTR8A çizgi sensörünün siyah ve beyaz gibi renk ayrımı yapabilmesi sayesinde gerekli kodlamalar ile robot istenilen siyah çizgi üzerinde pisti tamamlayacaktır. Proje gerçekleştirilirken devre tasarımı, devre elemanlarının lehimlenmesi, test ve deneme aşamaları, kodlama, sunum gibi aşamalar gerçekleştirilecektir.

## **Anahtar Kelimeler**

Arduino, Çizgi İzleyen Robot, QTR8A, Çizgi Sensörü,

## **ABSTRACT**

In this first project, the line follower robot will be made with the elements used. It is expected from this project that robot complete the runway by following the certain line without any problems. The QTR8A line sensor to be used for this purpose can distinguish between black and white color, so the robot will complete the run on the desired black line with the necessary coding. During the project, stages such as circuit design, soldering of circuit elements, testing and testing stages, coding, presentation will be realized.

## **Key Words**

Arduino, Line Follower Robot, QTR8A, Line Follower Sensor

## Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması

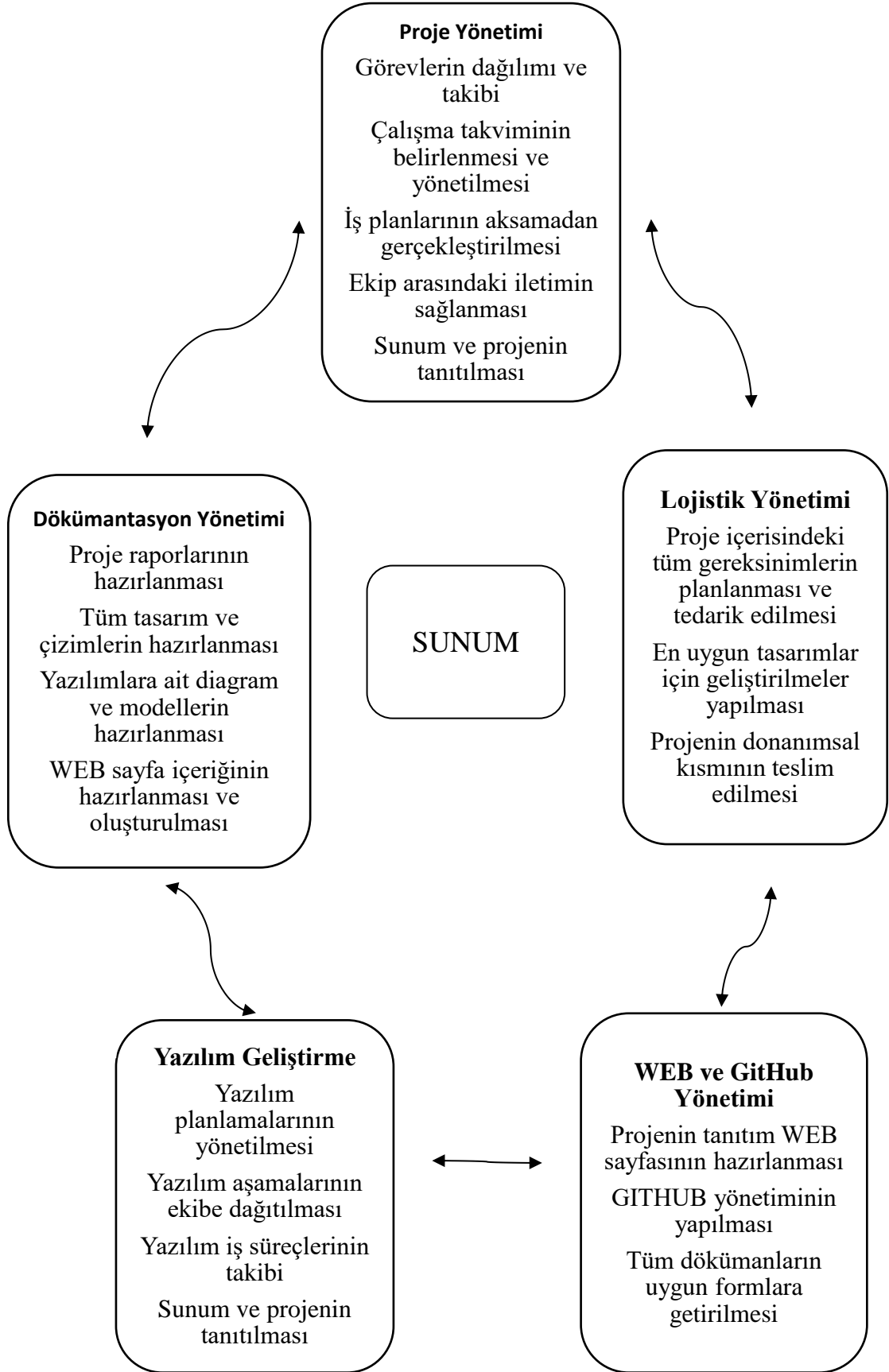
### 1. Görev Dağılımı ve Sorumlusu

Proje Yönetimi	Görev dağılımı ve takibinden sorumlu kişi, aynı zamanda proje grubunun çalışma takvimini ve düzenini ayarlamaktadır. Grupta bulunan kişilerle iletişim halinde olup projenin yönetimini sağlar.	Ceren AĞBABA
Doküman Yönetimi	Projenin tüm tasarım ve çizimlerinden, proje raporunun sunulmasından, dokümanların uygun forma getirilmesinden kodlamaya ait diyagram ve modellerin hazırlanması ve web sitesi tasarımından sorumlu olan kişidir.	Hatice AYDOĞDU
Lojistik Yönetimi	Projede kullanılacak tüm elemanların, malzemelerin belirlenmesi ve temin edilmesi, en uygun tasarımın yapılması için geliştirmelerin yapılmasıyla ve projenin donanımsal kısmının tanıtılması ile ilgilenen kişidir.	Mehmet Tekin ÇATALTEPE
Yazılım Geliştirme Yönetimi	Yazılım için araştırmaların yapılması, yazılım aşamalarının proje grubuna dağıtılması, Yazılım ile ilgili raporların hazırlanarak ilgili bölüme(doküman yönetimine) aktarılması yazılım ve süreç testlerinin gerçekleştirilmesi ile ilgilenen kişidir.	Elif YÜCEL
WEB ve GitHub Yönetimi	Proje tanıtımı için WEB sayfasının hazırlanması, projenin GitHub yönetiminin yapılması, doküman yöneticisinden almış olduğu raporlar ile WEB sitesine ve GitHub'a işlemekle sorumlu olan kişidir.	İsra AYDOĞDU
Sunum Yönetimi	Proje teslim zamanında sunumun, yapılan tüm işlemlerin uygun bir biçimde anlatılmasından, rapor ve evrakların eksiksiz bir şekilde sunulmasından ve önerilere, sorulara uygun çözümler üretmekten sorumludur.	Sudenaz ÜNAL

## 2. Görev Süresince Sürdürülen Eylemler

<b>1.Hafta</b>	Proje Hakkında Bilgi Edinme
<b>2.Hafta</b>	Malzeme Seçimi
<b>3.Hafta</b>	Mekanik ve Elektronik Tasarım
<b>4.Hafta</b>	Yazılım
<b>5.Hafta</b>	Grup elemanlarına ait iş yükünün tamamlanması (rapor, web, github)

### 3. Görevlerin İş Yükü Şeması



#### 4. Yoklama Çizelgeleri

<b>Tarihler</b> <b>Grup</b>	28.11.2017	05.12.2017	12.12.2017	19.12.2017	26.12.2017
İsra AYDOĞDU	+	+	+	+	+
Sudenaz ÜNAL	+	+	+	+	+
Ceren AĞBABA	+	+	+	+	+
Elif YÜCEL	+	+	+	+	+
Hatice AYDOĞDU	+	+	+	+	+
Mehmet Tekin ÇATALTEPE	+	+	+	İZİNLİ	+



## 5. Haftalık İş Katkı Cetvelleri

Tarihler	Yapılan iş
28.11.2017	Proje araştırması
05.12.2017	Malzeme Teminatı
12.12.2017	Elektronik ve Mekanik Tasarım
19.12.2017	Yazılım
26.12.2017	Deneme ve Test Aşamaları

## GİRİŞ

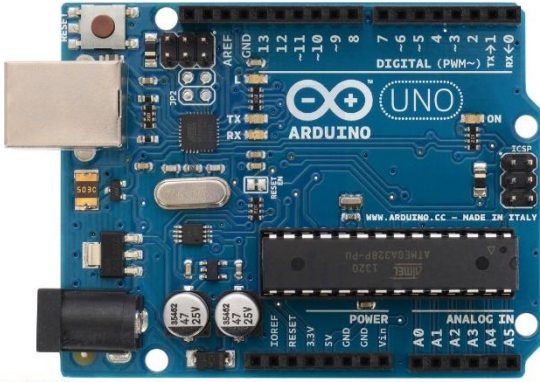
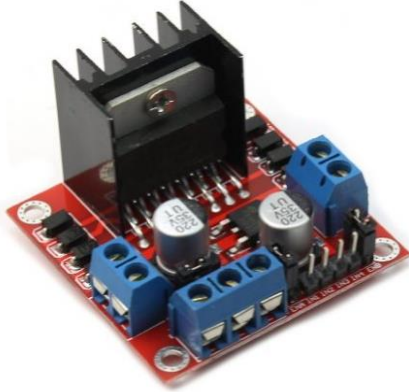
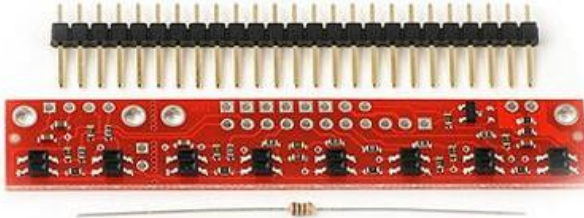
### Projenin Açıklaması

Yapılan projede amaç; hazırlanan pistte, çizgiyi takip edebilecek bir çizgi izleyen robot tasarımı gerçekleştirebilmektir. Robot yolu öz denetimli olarak takip etmelidir. Robotun çalışma prensibi; QTR-8A çizgi sensörü ile döndürülen değerler şart yapıları ile hesaplamaları sayesinde robotun konumunu güncelleyerek pisti tamamlamaktır. Robot ilerlerken hedeflenen çizgi robotun sağ tarafında kalırsa motorlar gerekli komutlarla sağ tarafa, sol tarafında kalırsa sol tarafa yönelecektir. Dönme işlemi de hedeflenen çizgiye gelene kadar devam edecektir. Çizgi üzerindeyken robot düz devam edecektir.

## YAPILAN ÇALIŞMALAR

### 1. Donanım Yapısı

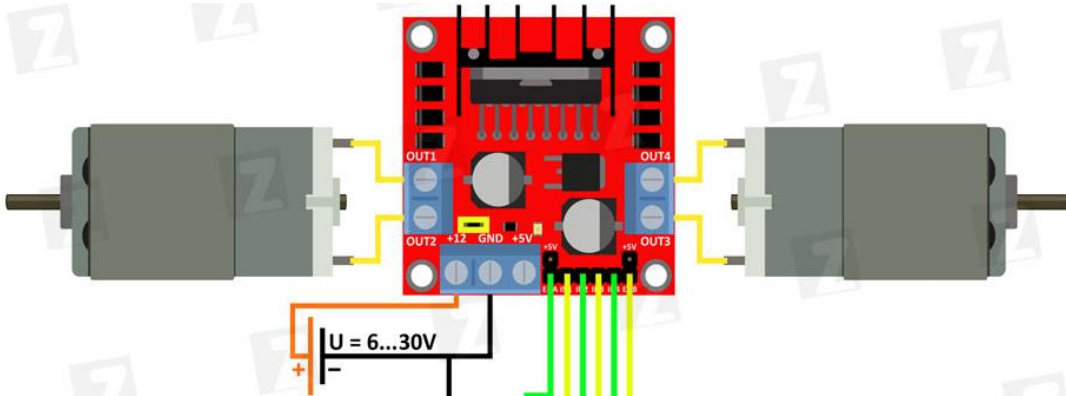
#### a. Gömülü Sistemler Mimarisi ve Devre Tasarımı

 <p>www.pololu.com</p>	<p><b>Arduino Mega 2560</b> ATMega2560 mikro denetleyici içeren bir Arduino kartıdır. 54 tane dijital I/O pini vardır. Bunlardan 15 tanesi PWM çıkışı olarak kullanılabilir. Ayrıca 16 adet analog girişi vardır. Çalışma gerilimi: 5V DC. Mikro denetleyici kartın çalıştırılabilmesi için USB, harici bir adaptör veya batarya ile beslemek gereklidir.</p>
	<p><b>L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı</b> 24V'a kadar olan motorları sürmek için hazırlanmış olan bu motor sürücü kartı iki kanallı olup, kanal başına 2A akım vermektedir. Kart üzerinde L298N motor sürücü entegresi kullanılmıştır. DC motorlardan ayrı olarak step motor kontrolüne de imkân sağlamaktadır.</p>
	<p><b>QTR-8A Çizgi Sensörü</b> Çizgi izleyen robotlar için yapılmış olan bu sensör kartı 1cm arayla yerleştirilmiş 8 IR LED/fototransistor çifti barındırmaktadır. LED çiftlerinin her biri ayrı birer MOSFET transistörle sürülmektedir ve ek hassasiyet ya da güç tasarrufu için LED'ler kapatılabilir. Kart üzerindeki her sensör ayrı bir analog voltaj çıkışı sağlar. Her bir sensöre bir pull-up direnci bağlanmıştır.</p>

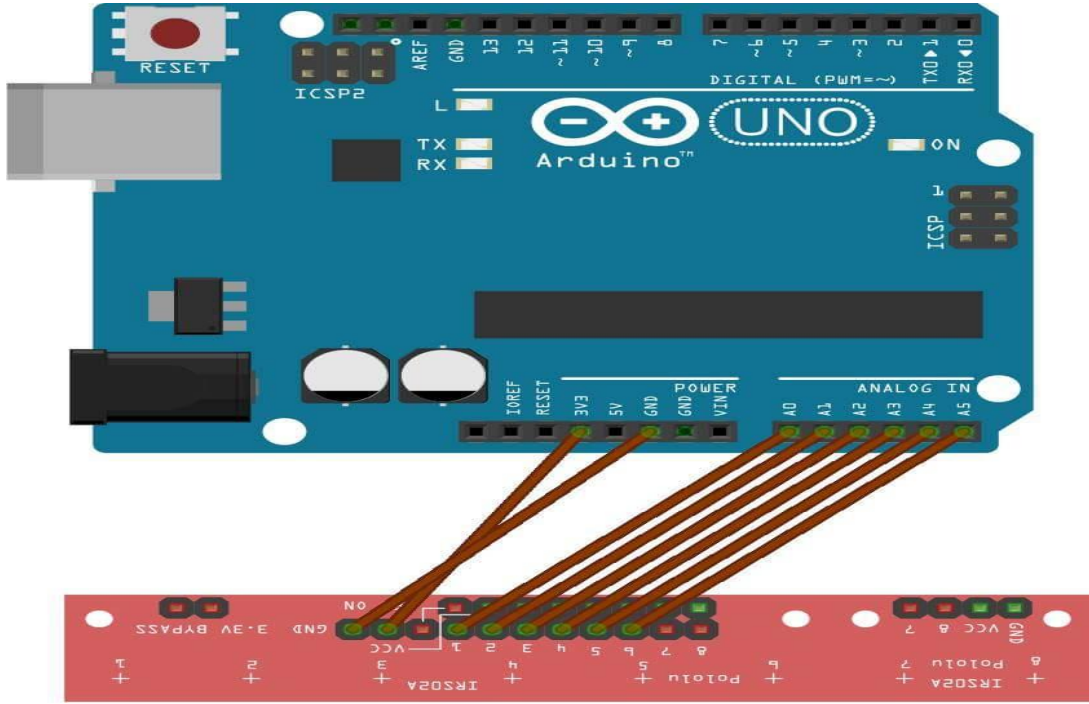
	<p><b>Pertinaks</b> Prototip devre elemanlarını pertinaks üzerine lehimleyip farklı uygulamalar için kullanabiliriz.</p>
	<p><b>Jumper Kablo (E-E/D-E)</b> Devre elemanlarının bağlantılarını gerçekleştirmek için kullanabiliriz.</p>
	<p><b>Tekerlek</b> 42 mm çapında yumuşak ve yüksek sürtünmeli bir tekerlektir. 16mm çaplı, mikro ve mini metal redüktörlü motorlarla tam uyumludur. Çapı:42mm Genişlik:19mm Ağırlık:16gr</p>

	<p><b>6V 250 Rpm Motor</b>  <b>Motor Özellikleri:</b></p> <p>Çalışma Voltajı: 3-12V  Redüksiyon Oranı: 1:48  Hız: 250 Rpm(@6V)  Akım: 95mA (maks. 160mA)  Ağırlık: 29gr</p>
	<p><b>Sarhoş Teker 19,05mm</b>  Plastik toplu bu küçük sarhoş tekerin bilye çapı 12,7 mm'dir. Birleştirilmiş halde kit uzunluğu 20,32 ile 27,94mm arasında değişmektedir.</p>

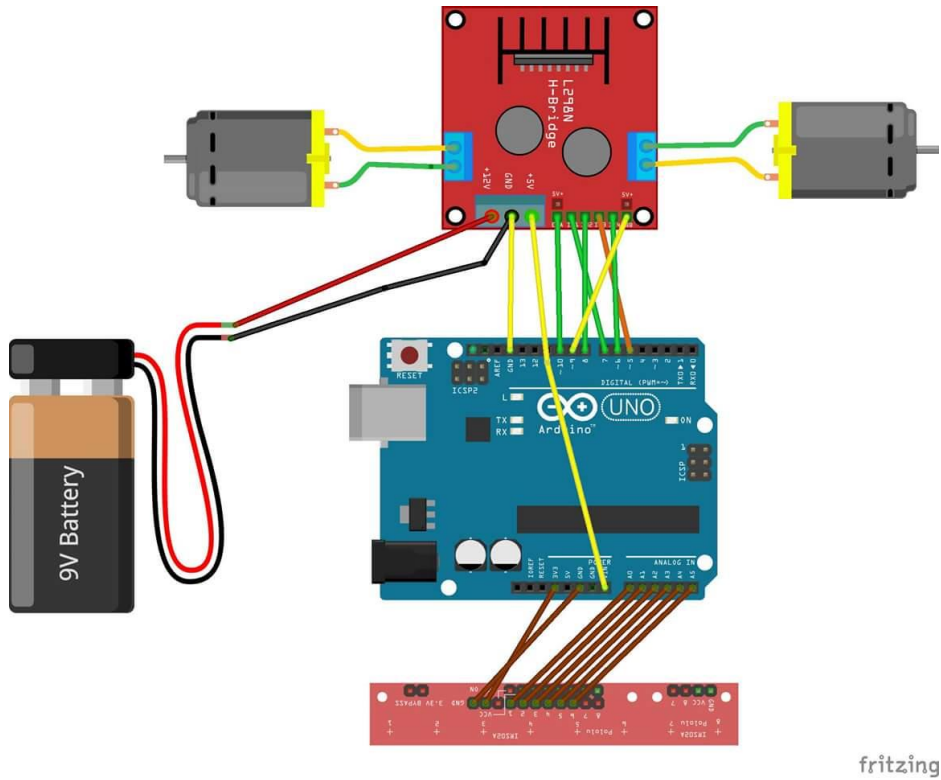
#### L298N bağlantısı:



QTR8-A bağlantısı:

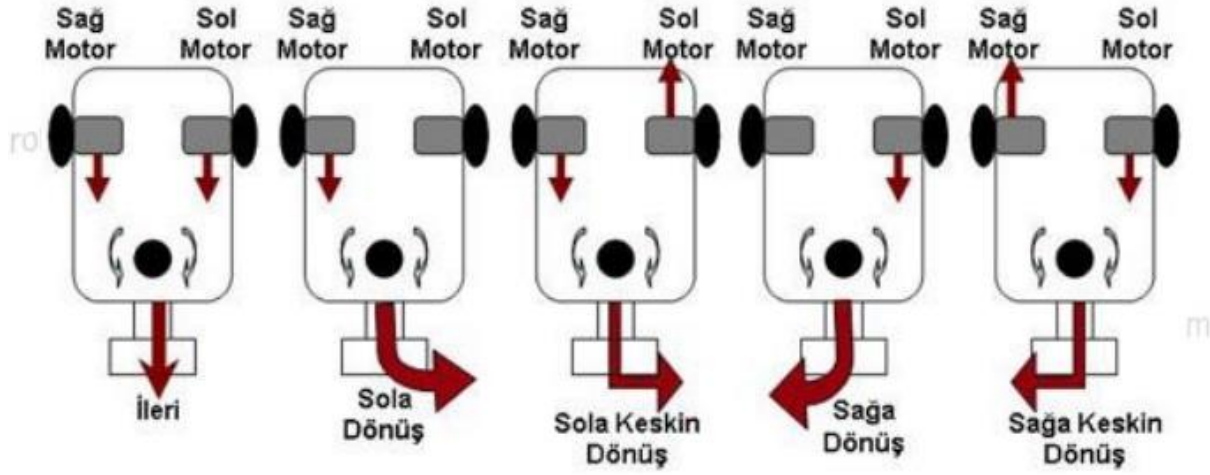


Devre Şeması:

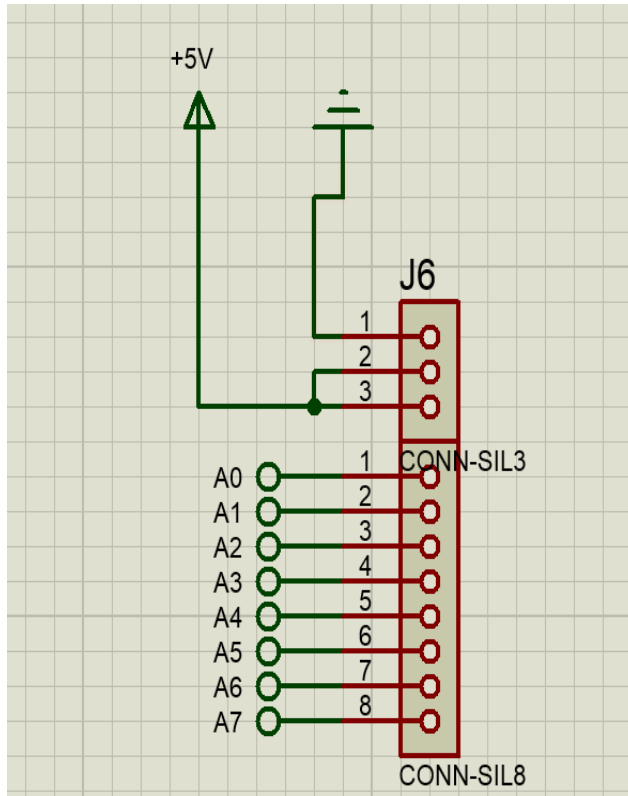


## b. Mekanik Sistem Mimarisinin Tanıtılması:

Robotun dönüşü birbirinden bağımsız çalışan iki motorun arasındaki hız farkıyla sağlanır. Aşağıdaki şemada bu sürüş sisteminin çalışması gösterilmiştir.



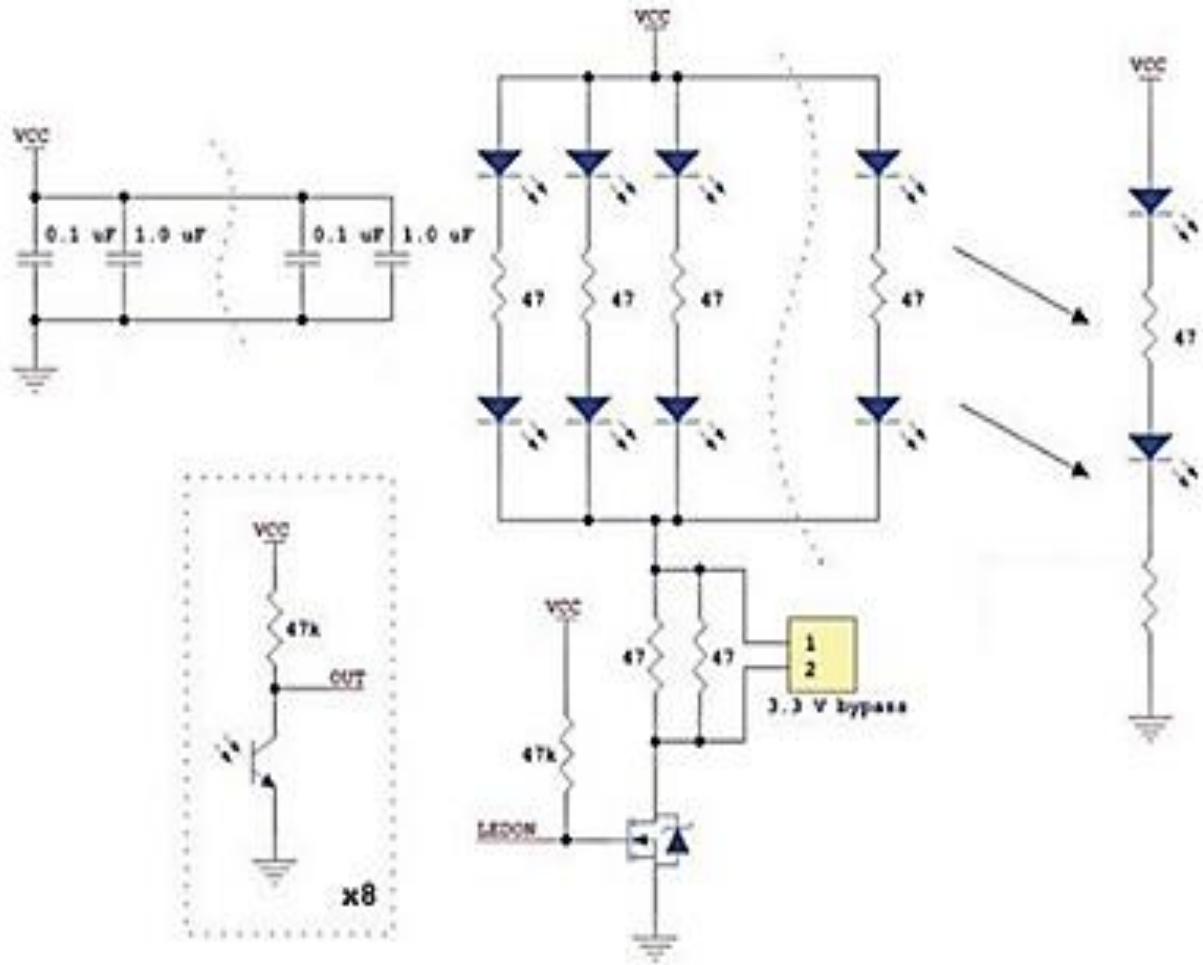
## 1.QTR-8A Çizgi Sensörü



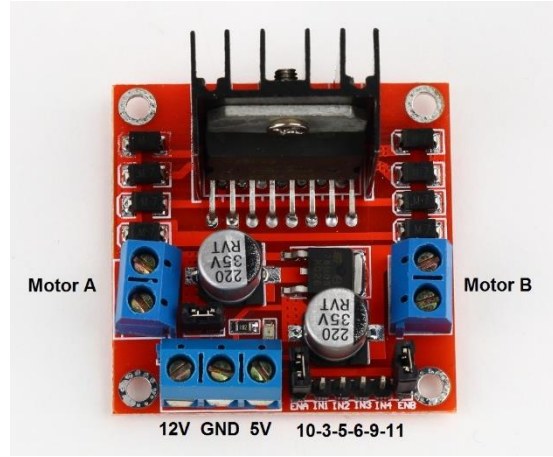
Burada bulunan 8 tane çizgi sensörü pin'i arduinonun analog bacaklarına bağlıyoruz. Vcc ve led on pinini +5V'a Gnd pinini de toprağa bağlıyoruz. QTR-8A Sensör kartı bir kaç farklı şekilde çalıştırılabilir. Bir mikro denetleyicinin analog giriş pinlerine bağlanarak ADC (Analog Dijital Çevirici) işlemine tabi tutulabilir. Eşik değeri ayarlanabilir bir karşılaştırıcı kullanılarak gelen analog voltajı dijitale çevirerek işlemler yaptırılabilir. Her çıkışı mikro denetleyicinin I/O pinlerine bağlanarak mikro denetleyici içindeki karşılaştırıcı kullanılarak okuma yapılabilir. (Bu yöntemle yüksek yansıma olan ortamlarda daha iyi sonuç alınır.).QTR-8A sensörü 8 sensörün tamamını kullanmak istemeyen kullanıcılara da imkân sağlamaktadır. Ayrıca sensör yakınlık ve cisim algılama sensörü olarak da kullanılabilir. Çalışma voltajı 3.3 volt ile 5 volt arasındadır. Çektiği maksimum akım 100mA'h'tir. İdeal algılama mesafesi 3mm dir. Bu uzaklıktan fazla olduğunda kararlı çalışmaz. Sorunlar çıkarabilir.



## QTR sensör içyapısı:

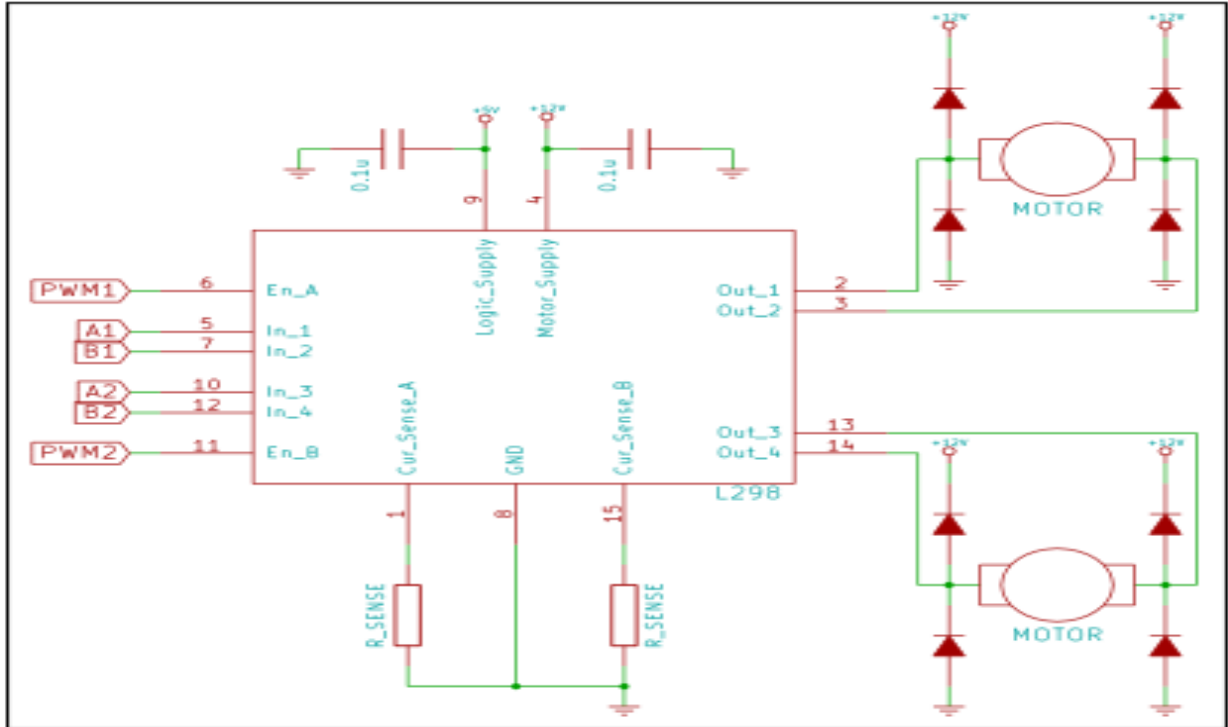


## 2.L298N Motor Sürücü Kartı:



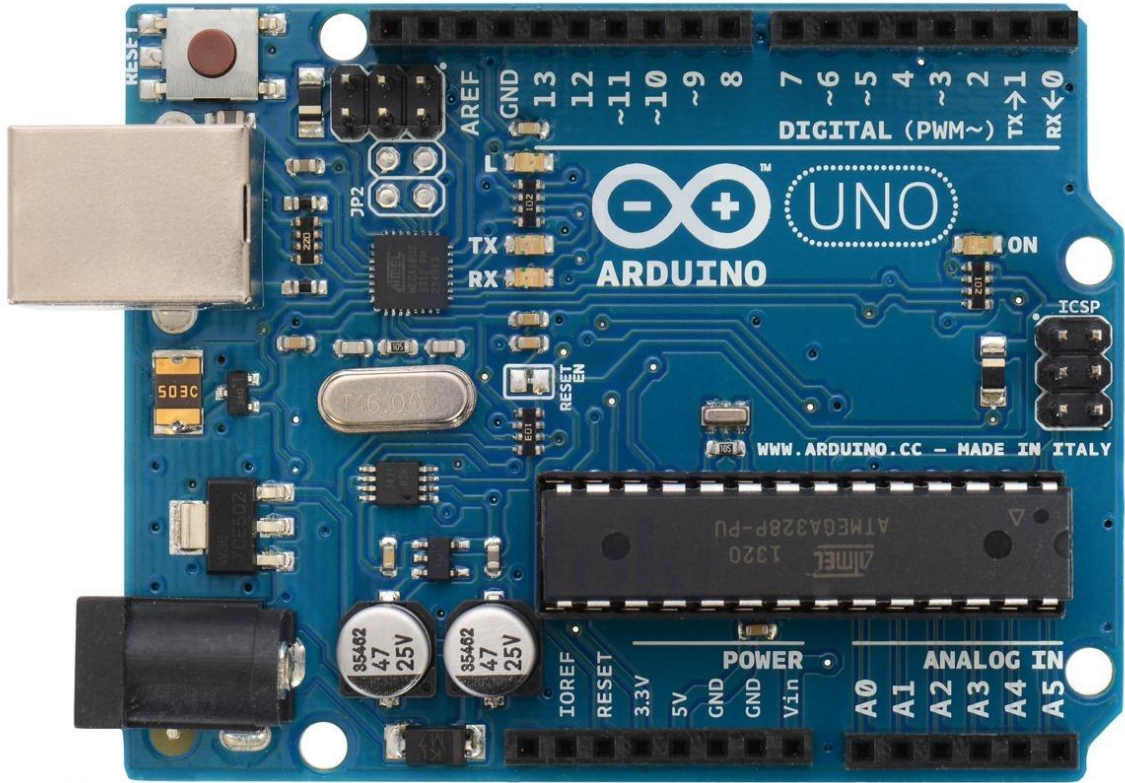
2 adet H köprüsü bulunur. H köprüsü DC motorun iki yönde de hareket etmesini sağlayan faydalı bir yöntemdir. Bu yöntem için 4 adet transistör kullanılır. Bu entegre içerisinde toplam 15 adet bacak bulunmaktadır. Bunlardan IN1, IN2, OUT1, OUT2, ENA, SENA A köprüsü için IN3, IN4, OUT3, OUT4, ENB, SENSB B köprüsü içindir. IN1 ve IN2 girişleri 5V'a duyarlı girişlerdir. OUT1 ve OUT2 isiminden de anlaşılacağı gibi çıkış işlemleri içindir.

### Devre Şeması





### 3.Arduino Uno



#### Özellikler

<b>Mikro denetleyici</b>	ATMega 2560
<b>Çalışma Gerilimi</b>	5V
<b>Giriş Gerilimi(önerilen)</b>	7-12V
<b>Dijital I/O Pinleri</b>	54 (15 tanesi PWM çıkışı)
<b>Analog Giriş Pinleri</b>	16
<b>Her I/O için Akım</b>	40mA
<b>3.3V Çıkış için Akım</b>	50 mA
<b>EEPROM</b>	4 KB (ATmega2560)
<b>SRAM</b>	8KB(ATMega 2560)
<b>Flash Hafıza</b>	256 KB (Atmega2560) 8 KB kadarı bootloader tarafından kullanılmaktadır.
<b>Uzunluk</b>	101.6 mm
<b>Genişlik</b>	53.4mm
<b>Ağırlık</b>	36gr

Arduino Mega 2560 gücünü USB üzerinden veya harici bir güç kaynağından alabilir. Kartın çalışması için USB kablusunun sürekli bağlı olması gerekli değildir. Kart sadece adaptör veya batarya ile de çalıştırılabilir. Bu da kartın bilgisayardan bağımsız da çalışabileceğini gösterir. Harici güç kaynağı olarak 6-20V arası kullanılabilir. Bu değerler limit değerleridir. Önerilen harici besleme 7-12V arasındadır. 12V üzerindeki değerlerde kart aşırı ısınabilir. Mega kartının üzerindeki mikro denetleyicinin çalışma gerilimi 5V'dur.

Güç pinleri:

- **VIN:** Harici güç kaynağı kullanırken 7-12V arası gerilim giriş pini.
- **5V:** Regülatörden çıkan 5V çıkışı verir. Eğer kart sadece USB (5V) üzerinden çalışıyor ise USB üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Eğer karta güç Vin (7-12V) veya güç soketi (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
- **3V:** Kart üzerinde bulunan 3.3V regülatörlü çıkış pinidir. Maks. 50mA çıkış verebilir.
- **GND:** Toprak pinleridir.

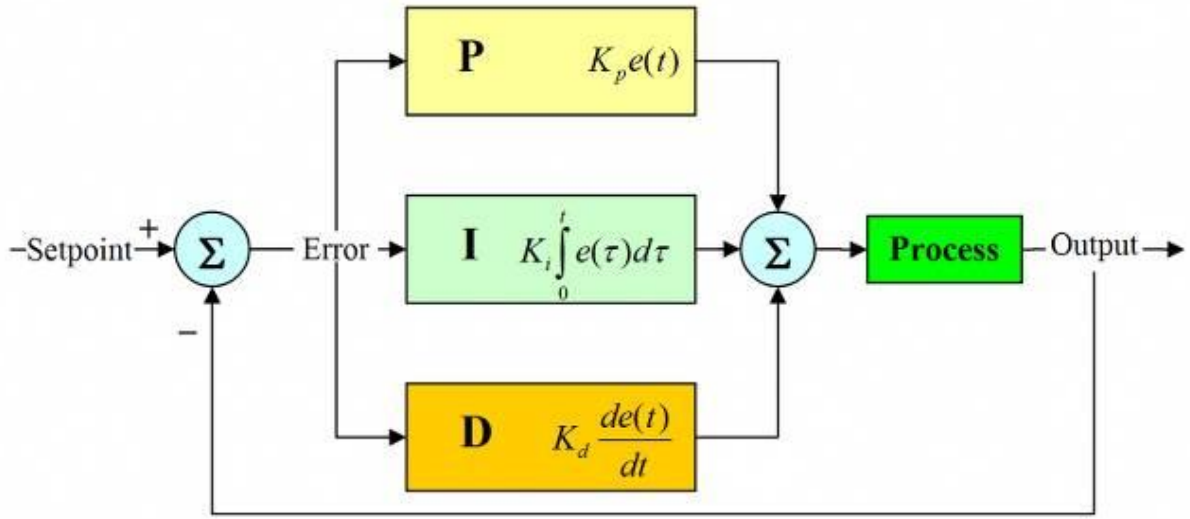
#### 4. DC Motor 6V



##### Özellikler:

- |                          |          |                              |           |
|--------------------------|----------|------------------------------|-----------|
| • Çalışma Voltajı        | : 6V     | • Zorlanma Akımı(12V)        | : 2.9A    |
| • Azami Çalışma Voltajı  | : 12V    | • Motor Çapı                 | : 12-12mm |
| • Devir                  | : 600Rpm | • Redüktör Çapı- Kare        | : 13x12mm |
| • Boşta Çektiği Akım(6V) | : 50mA   | • Mil Ortadan Çıkışlı D Şaft | : 3mm     |
| • Zorlanma Akımı(6V)     | : 1.6A   | • Mil Uzunluğu               | : 10mm    |
| • Ağırlık                | : 16 gr  | • Uçtan Uca Uzunluk          | : 42mm    |

## 5. PID



Şekil 9: PID kontrol Diyagramı

PID, Oransal İntegral Türev için kullanılan bir kısaltmadır. En genel tanımıyla bir kontrol geri bildirim mekanizmasıdır. PID yönteminin en temel amacı hatayı minimize etmek, en aza indirmektir. PID kontrolünü uygulamak ve kavramak oldukça zordur. PID kontrolünde öncelikle hata tanımlaması yapılmalıdır. Hata ise referans değere olan uzaklık olarak tanımlanabilir. Açıklamak gerekirse;

Referans=İstenilen değer

Gelen=Şuan ki Konum

HATA=Referans-Gelen

### Oransal Terim(P):

Oransal terim, sistemden gelen hatayı bir katsayı ile çarparak hatayı küçültmeyi hedefler. Bozucu etkileri de mevcuttur. Projemizin yazılım kısmında oldukça ağırlık verdiğimiz PID kontrolünde deneme ve test aşamasında bu katsayıya büyük bir değer vermemiz gerektiğini anladık.

$$P=K_p \cdot \text{HATA}$$

### İntegral Terimi(I):

İntegral, hatanın alanını bulmak anlamına gelir. İntegralin çok yükselmesini önlemek için sınırlandırmak gereklidir. Sürekli toplandığı için integral çok artarsa tekrar azalmasını beklemek zaman alır. Bu yüzden integrali sınırlamak sistemin çabuk toparlamasını sağlayacaktır.

$$I = I + (K_i * HATA * dt)$$

dt: PID fonksiyonuna her girdiğinde geçen zaman.

### **Türev Terimi(D):**

Türev sistemdeki iki örnek arasındaki zamanı hesaplar. Eğer hatada bir değişim olmadıysa türev sıfır olur.

EHata=Bir önceki hatanın değeri

$$HD = HATA - EHata$$

$$D = (K_d * HD) / dt$$

### **PID Algoritması**

$K_p$  ,  $K_i$  ve  $K_d$  katsayılarından oluşur. Bu katsayılar deneme yanılma yöntemiyle bulunur. Yapacağınız sistemde optimum katsayıları bulmak için değerde değişiklik yapıp sistemi gözlemlemeniz gerekmektedir.

HATA = Referans - Gelen

HD = HATA - EHata

P =  $K_p * Hata$

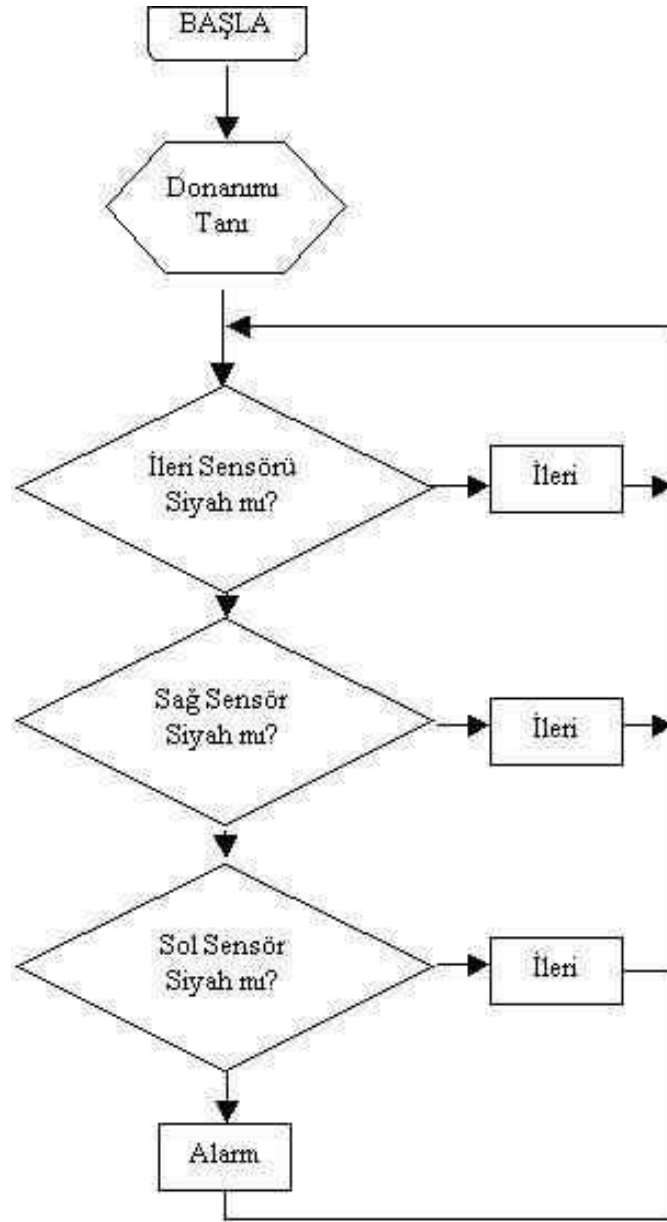
I =  $I + (K_i * HATA * dt)$

D =  $(K_d * HD) / dt$

PID = P + I + D

EHata = Hata

## 2. Yazılım Yapısı



### a. Algoritma

- Çizgi izleyen robotun orta sensörü çizgi üzerindeyse iki motor da ileri doğru sürülmelidir.
- Çizgi izleyen robotun en sol sensörü çizgi üzerinde ise çizgi izleyen robotun piste doğru yönelmesi için sağ motorun ileriye doğru çalışması sol motorun ise bu esnada geriye doğru çalışması gerekmektedir.
- Çizgi izleyen robotun en sağ sensörü çizgi üzerinde ise çizgi izleyen robotun piste doğru yönelmesi için sol motorun ileriye doğru çalışması sağ motorun ise bu esnada geriye doğru çalışması gerekmektedir.

- Çizgi izleyen robotun sağ sensörü çizgi üzerinde ise çizgi izleyen robotun piste doğru yönelmesi için sol motorun ileriye doğru çalışması sağ motorun ise bu esnada sol motordan daha yavaş bir şekilde ileriye doğru çalışması gerekmektedir.
- Çizgi izleyen robotun sol sensörü çizgi üzerinde ise çizgi izleyen robotun piste doğru yönelmesi için sağ motorun ileriye doğru çalışması sol motorun ise bu esnada sağ motordan daha yavaş bir şekilde ileriye doğru çalışması gerekmektedir.

## **b. Kodlama**

```
#include <QTRSensors.h>

#define NUM_SENSORS 6

#define EMITTER_PIN 8
```

```
#define PWMA 10

#define a1 7

#define a2 8
```

```
#define PWMb 9

#define b1 6

#define b2 5
```

```
QTRSensorsAnalog qtrrc((unsigned char[]) { A0, A1, A2, A3, A4, A5 }, NUM_SENSORS,
EMITTER_PIN);
```

```
unsigned int sensorValues[NUM_SENSORS];
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  pinMode(a1, OUTPUT);
```

```
  pinMode(a2, OUTPUT);
```

```
pinMode(PWMA, OUTPUT);

pinMode(b1, OUTPUT);

pinMode(b2, OUTPUT);

pinMode(PWMB, OUTPUT);
```

```
int i;

for (int i = 0; i < 250; i++)
{
    qtrrc.calibrate();

    delay(10);
}

delay(1000);
}
```

```
void loop()
{
    unsigned int sensors[6];

    int position = qtrrc.readLine(sensors);

    Serial.println(position);

    if(position<=500)
    {
        digitalWrite(a1, HIGH);

        digitalWrite(a2, LOW);

        analogWrite(PWMA,250);

        digitalWrite(b1, LOW);

        digitalWrite(b2, HIGH);

        analogWrite(PWMB,100);
    }
}
```

```

    }
else if(position<=1000)
{
    digitalWrite(a1, HIGH);
    digitalWrite(a2, LOW);
    analogWrite(PWMA,240);
    digitalWrite(b1, HIGH);
    digitalWrite(b2, LOW);
    analogWrite(PWMB,40);
}

else if(position<=1500)
{
    digitalWrite(a1, HIGH);
    digitalWrite(a2, LOW);
    analogWrite(PWMA,220);
    digitalWrite(b1, HIGH);
    digitalWrite(b2, LOW);
    analogWrite(PWMB,60);

}

else if(position<=2000)
{
    digitalWrite(a1, HIGH);
    digitalWrite(a2, LOW);
    analogWrite(PWMA,200);
    digitalWrite(b1, HIGH);
    digitalWrite(b2, LOW);
    analogWrite(PWMB,80);
}

```



```

else if(position>=2500 && position<=3000)
{
digitalWrite(a1, HIGH);
digitalWrite(a2, LOW);
analogWrite(PWMA,250);
digitalWrite(b1, HIGH);
digitalWrite(b2, LOW);
analogWrite(PWMB,250);
}
else if(position<=3500)
{
digitalWrite(a1, HIGH);
digitalWrite(a2, LOW);
analogWrite(PWMA,80);
digitalWrite(b1, HIGH);
digitalWrite(b2, LOW);
analogWrite(PWMB,200);

}
else if(position<=4000)
{
digitalWrite(a1, HIGH);
digitalWrite(a2, LOW);
analogWrite(PWMA,60);
digitalWrite(b1, HIGH);
digitalWrite(b2, LOW);
analogWrite(PWMB,220);
}

```

```

else if(position<=4500)
{
digitalWrite(a1, HIGH);
digitalWrite(a2, LOW);
analogWrite(PWMA,40);
digitalWrite(b1, HIGH);
digitalWrite(b2, LOW);
analogWrite(PWMB,240);
}
else if(position<=5000 )
{
digitalWrite(a1, LOW);
digitalWrite(a2, HIGH);
analogWrite(PWMA,100);
digitalWrite(b1, HIGH);
digitalWrite(b2, LOW);
analogWrite(PWMB,250);
}
else
{
digitalWrite(a1, LOW);
digitalWrite(a2, LOW);
analogWrite(PWMA,0);
digitalWrite(b1, LOW);
digitalWrite(b2, LOW);
analogWrite(PWMB,0);
}
}

```

## SONUÇ

### 1. Bilgi Düzeyine Katkıları

Her şeyden önce proje yönetiminin nasıl olması gerektiğini ve projenin sunumunun nasıl yapılması gerektiğini öğrendik. 4 Haftalık süreçte birçok devre elemanının kullanımıyla ilgili tecrübeler edindik. Devrenin kurulumu, gerekli kodlamanın yapılması, devrelerin şematik olarak gösterilmesi, fritzing, proteus kullanımı ve daha birçok konuda bilgi edindik. Bir robotun elektronik ve mekanik tasarımının nasıl olması gerektiğini öğrendik ve yaptığımız yanlışlar ile tasarım aşamasında yapılmaması gerekenleri fark ettik.

PID kontrolü konusuna yoğunlaştık. Hata oranını minimize ederek, robotun sorunsuz bir şekilde çizgiyi tam anlamıyla takip etmesi için araştırmalar yapıp  $K_p$ ,  $K_d$  ve  $K_i$  değerlerine deneme ve yanılma yöntemiyle uygun katsayıları bulmaya çalıştık. Bizi en zorlayan kısım PID kontrolü ve robotun çizgi takip işlemini gerçekleştirirken hız ve açısal konumunun ayarlanması oldu. Test aşamasında gördüğümüz hatalarımızı düzelttik. 1.Proje sayesinde devre tasarımında devre elemanlarının zarar görmemesi için yapılması gerekenleri devre tasarımının nasıl olması gerektiğini ve bağlantıları öğrendik.

### 2. Teknolojik Katkıları

Çizgi izleyen robotlar günümüzde gelişen teknolojiyle birlikte endüstriyel alanda bir süredir kullanılmaktadır. Genellikle lojistik ve otomasyon bölümleri içerisinde oldukça gereksinim duyulan niteliksiz insan gücü ile yapılan taşıma işlemlerini bir süredir çizgi izleyen robotlar yapmaya başlamışlardır. Günümüzde bu kadar yaygın olarak kullanılan çizgi takip eden robot tasarımını gerçekleştirmiş olmamız bize iş hayatımızda kazanım olarak döneceğini ve bizim için önemli bir tecrübe, deneyim olduğunu düşünüyoruz.

### 3. Ekip Çalışmasının Katkıları

Ekip çalışmasının bize kattıkları:

- Her grup elemanının teknik bilgisi arttı.
- Konuya bütünlük açısından bakmamıza yardımcı oldu.
- Proje, çalışanların sorun çözme becerisini geliştirdi.
- Çalışanların birbiriyle iletişim alışkanlıklarını geliştirdi.
- Ekip, bir kişinin tek başına üretebileceğinden daha fazla fikir üretebilir ve çözüm önerisi geliştirebileceğinden ekip olarak çalışmanın faydalı olduğunu gördük.
- Yanlış karar verme ve yanlış uygulama riski ekip çalışması ile en aza indirgenmiş oldu.

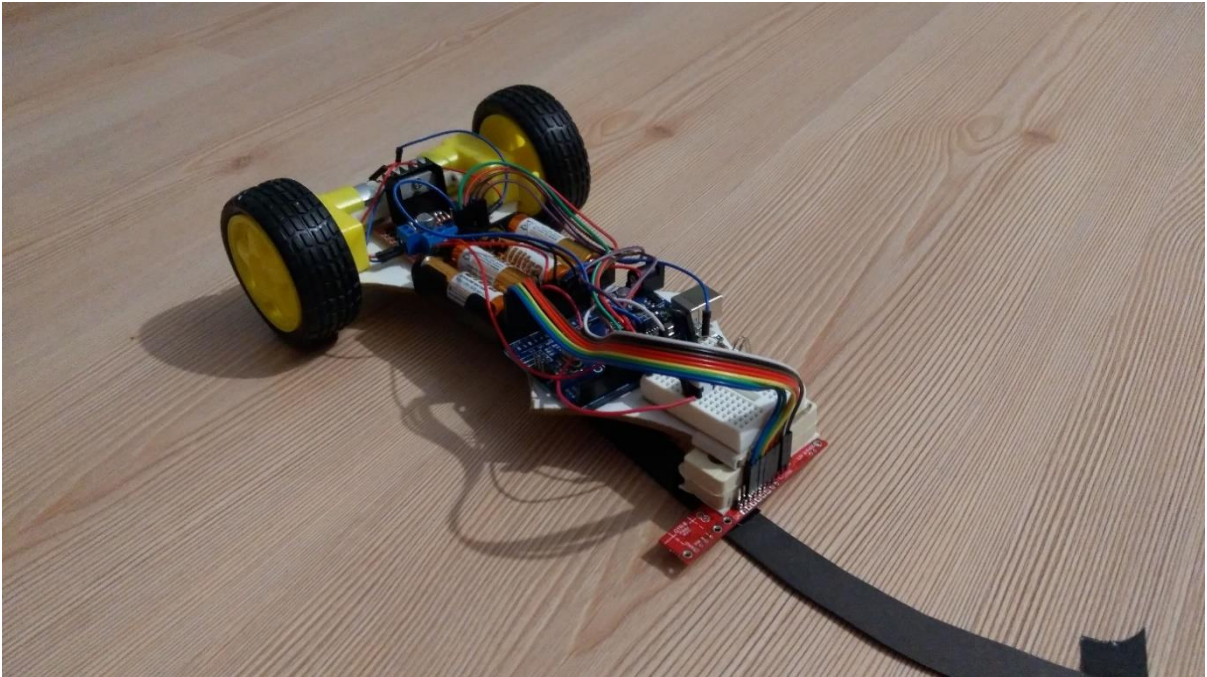
#### 4. Aksayan Yönler

Projeyi gerçekleştirirken karşılaştığımız sorunlar:

- PID Kontrolü:  $K_p$ ,  $K_d$ ,  $K_i$  katsayılarına uygun değerleri bulmakta zorluk çektik. Verilen yanlış değerlerden dolayı robotun dönme açısının doğru olmadığını fark ettik. Bu konuyla ilgili birçok araştırma yaparak sorunu çözmeye çalıştık.
- Devre Tasarımının Çizimi (Fritzing, Proteus, Circuits ): Devre tasarımının çiziminde kullandığımız devre elemanlarının fritzing, proteus gibi devre tasarımı programlarında bulunmaması karşımıza bir sorun olarak çıktı. Çözüm olarak 'Paint' çizim programını kullanarak devre tasarımını gerçekleştirdik.

#### 5. Görüş ve Öneriler

Bu projede elektronik ve mekanik iki kısım bulunmakta olması hem elektronik ve hem de mekanik becerilerimizi artırmamızı sağladı. Bu iki kısmın birbirleriyle uyumu açısından irdelendiğinde hem teoride hem uygulamada proje grubumuza birçok bilgi birikimi sağlamıştır. Edindiğimiz bu bilgiler sayesinde yapacağımız diğer projelerde daha başarılı olacağımıza inanıyoruz. İlerleyen aşamalarda yapacağımız yeni çalışmalarla çizgi takip eden robotumuza ultrasonik mesafe sensörü ile mesafe algılamayı sağlamak, bluetooth ile kablosuz haberleşmeyi sağlamak gibi özellikler ekleyerek daha fazla geliştirebileceğimizi düşünüyoruz.



## **6. GitHub Adresi**

<https://github.com/KullanAt/9cGrup1>