

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**ARDUINO İLE ENGELDEN KAÇAN ROBOT**

**TASARIM PROJESİ**

**Yaşar ÖZMEN**

**2015-2016 GÜZ DÖNEMİ**

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**ARDUINO İLE ENGELDEN KAÇAN ROBOT**

**TASARIM PROJESİ**

**Yaşar ÖZMEN  
259176**

**Bu projenin teslim edilmesi ve sunulması tarafımda uygundur.**

**Danışman : Yrd. Doç. Dr. Murat AYKUT .....**

**2015-2016 GÜZ DÖNEMİ**



## **IEEE Etik Kuralları IEEE Code of Ethics**



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

## ÖNSÖZ

Üniversite hayatımızda öğrendiğimiz teorik bilgileri gerçek hayatta uygulayabilmek en önemli hususlardandır. Bundan dolayı üniversite hayatımızda yaptığımız projeler bizi gerçek hayata en iyi hazırlayacağını düşündüğüm fırsatlardır.

Tasarım projesinin hem donanımı hem de yazılımı bir arada bulundurması bakımından oldukça önemli bir deneyim olduğunun düşüncesindeyim.

Projenin başından sonuna kadar hep yanımda olup bana her konuda yardımcı olan Aycan ALTINDAL'a, elinden geldiğince yardım eden Fatih Emre ERDEM ve diğer arkadaşlarıma, donanım kısmı konusunda fikirler verip ilgilenen Laboratuvar sorumlusu İsmail Taşkına teşekkür ediyorum.

Çalışmam sırasında danışmanlığımı yapan Sayın Yar. Doç. Dr. Murat AYKUT'a teşekkür eder saygılarımı sunarım.

Yaşar ÖZMEN  
Trabzon, 2015

# İÇİNDEKİLER

Sayfa No

IEEE ETİK KURALLARI .....	II
ÖNSÖZ .....	III
İÇİNDEKİLER .....	IV
ÖZET .....	V
1.GİRİŞ .....	1
1.1.Arduino Uno .....	1
1.2.Ultrasonik Mesafe Sensörü HC-SR04.....	2
1.3. Motor Sürücü Entegresi L293B .....	3
2.YAPILAN ÇALIŞMALAR .....	5
2.1.DONANIM .....	5
2.2.YAZILIM .....	7
2.2.1. Arduino Programlama Komutları .....	7
2.2.2. Projenin Kodlarının Yazılması .....	8
3. SONUÇLAR .....	15
4. ÖNERİLER .....	16
5.KAYNAKLAR .....	17
STANDARTLAR VE KISITLAR FORMU .....	18

## ÖZET

Robotlar günlük yaşamda ve endüstriyel otomasyon uygulamalarında gün geçtikçe daha yaygın bir biçimde yer almaya başlamıştır. İnsana özgü eksiklerden arındırılmışlardır ve görevlerini eksiksiz bir biçimde gerçekleştirmektedirler. Dış dünya ile sensörleri aracılığıyla haberleşirler ve gelen bilgileri mantıksal işlemlerle değerlendirip karar verirler.

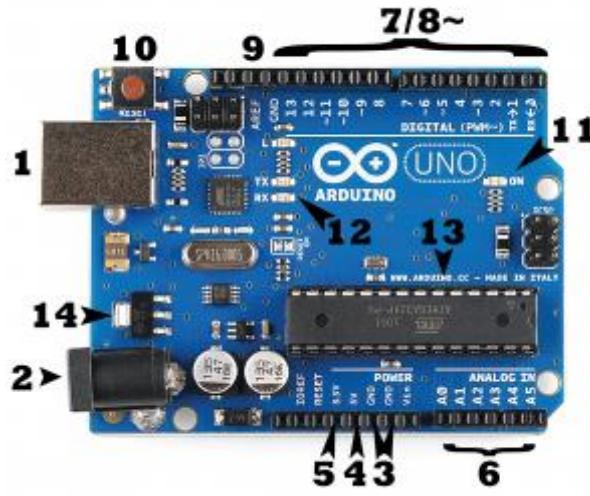
Bu proje kapsamında ise engelden kaçarak yolunu bulan türde bir robot tasarlamak amaçlanmıştır. Robot; DC motorlar, DC motor sürücü devresi(L293B), mikrodenetleyici yazılımı, ultrasonik algılama devresi gibi alt sistemlerden oluşmaktadır. Tasarlanan robotun ana işlevi şöyle açıklanabilir; engelden kaçan robot sahip olduğu ultrasonik algılayıcı ile çevresindeki cisimleri algılayıp hareketini bu cisimlerin konumuna göre düzenleyecektir.

## 1. GİRİŞ

### 1.1. Arduino Uno

Arduino, elektronik projeler oluşturmak için kullanılan açık kaynak kodlu bir platformdur. Temel olarak devre kartı, işlemci, derleyici ve derlenen programları işlemciye yükleyen araçlardan meydana gelir.

Arduino ailesi; Uno, Leonardo, Due gibi birçok çeşitte board ve Arduino proto shield gibi yardımcı elemanlardan meydana gelmektedir. Arduino ailesinin popüler board' u olan Arduino Uno R3' ün genel yapısı şu şekildedir;



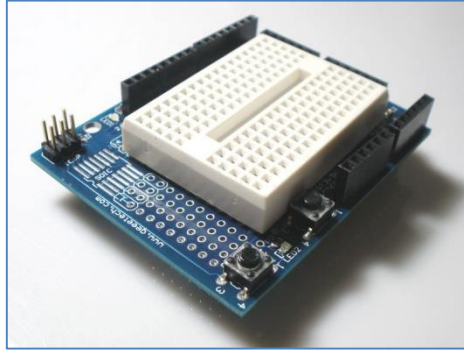
Şekil 1. Arduino Uno R3

- 1) USB jakı
- 2) Power jakı(7-12V DC)
- 3) GND- toprak pini
- 4) 5V power pini
- 5) 3.3V power pini
- 6) Analog giriş pinleri
- 7) Dijital giriş / çıkış pinleri
- 8) Tilda(~) işaretli pinler PWM çıkışı sağlar
- 9) AREF- analog referans pini
- 10) Reset butonu
- 11) Power ledi
- 12) TX / RX ledleri
- 13) Mikrodenetleyici ATmega328
- 14) Regülatör

Arduino Uno'nun teknik özelliklerine bakılırsa; mikrodenetleyici olarak ATmega328 kullanılmıştır. 32 KB belleğe sahiptir. Çalışma gerilimi de +5V DC'dir. Board'un USB jakından veya power jakından Arduino'ya güç vererek 5V çıkış veren power pininden düzenli

olarak çıkış alınabilir. Board'un sağ tarafında 14 adet dijital giriş / çıkış pini bulunur. Bu pinlerden yanında tilda (~) işareti olan 6 tane pin PWM çıkışı destekler. Board'un sol tarafında ise 6 adet analog giriş pini bulunur. Giriş / çıkış pini başına düşen DC akım 40 mA kadardır.

Arduinonun yardımcı elemanlarından olan Arduino Uno Proto Shield ise Arduino'nun üzerine takılarak kullanılır. Arduino projelerini ayrı bir breadboard kullanmaksızın Proto Shield üzerine koyulan küçük breadboard üzerinde devre kurulumuna ve özel olarak tasarlanmış devrelerin Arduino ile beraber kullanılmasına imkan sağlamaktadır. Yani orta bölümde bulunan mini breadboard ile lehimlemeye gerek kalmadan devreler rahatlıkla kurulabilir.



Şekil 2. Arduino Uno Proto Shield

## 1.2. Ultrasonik Mesafe Sensörü HC-SR04

Ultrasonik sensörler ses dalgalarının yardımıyla nesnelere temas etmeksizin tespit edebilen sensörlerdir. Ses dalgaları sınıflandırılmasında 20Khz-1Ghz aralığındaki ses sinyalleri ultrasonic ses olarak tanımlanmıştır. Ultrasonic sensörler de bu insan kulağının algılayamayacağı frekanslardaki ultrasonik ses dalgalarını kullanarak işlem yaparlar.

HC-SR04 olarak adlandırılan ultrasonik mesafe sensörü basitçe tanımlanırsa bir hoparlör ve bir mikrofondan oluşur. 40Khz frekansında ultrasonik ses dalgaları üretmektedir. Sensör bu ürettiği ses dalgalarını hoparlörden yayar. Eğer ses dalgaları bir cisim ile çarpışırsa yansiyarak sensörün mikrofoni tarafından algılanırlar. Burada önemli olan dalgaların gidiş geliş süresidir. Bu süre kullanarak da cismin sensöre olan uzaklığı hesaplanır.



Şekil 3. Ultrasonik Mesafe Sensörü HC-SR04



HC-SR04 mesafe sensöründe 4 adet pin mevcuttur. Bunlar Vcc, GND, Trig ve Echo pinleridir. Vcc ve GND pinlerinden sensöre 5V ve toprak verilirken Trig( Trigger ) ve Echo pinleri de Arduino üzerindeki ikisine bağlanır. Trigger pini output iken echo pini inputtur. Trigger pinine güç verince sensör ses dalgasını yaymaya başlar. Cismin sensörden uzaklığı ile doğru orantılı olarak echo pini belli bir süre lojik 1 seviyesinde kalır ve tekrar lojik 0 olur. Yani cismin sensöre uzaklığını ölçmek için echo pininin ne kadar süreyle lojik 1 seviyesinde kaldığına bakılır.

HC-SR04 mesafe sensörünün temel özellikleri şu şekildedir;

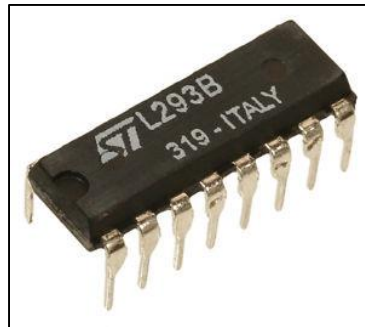
- Çalışma Voltajı — 5V DC
- Çektiği Akım — 15 mA
- Çalışma Frekansı — 40 Hz
- Maksimum Görme Menzili — 4 m
- Minimum Görme Menzili — 2 cm
- Görme Açısı — 15 derece
- Boyutları — 45mm x 20mm x 15mm

### 1.3. Motor Sürücü Entegresi L293B

Mikrodenetleyicilerin çıkışları DC motorları veya step motorları direkt olarak kontrol etmek için yetersiz olduğundan motor sürücü devreler kullanılır. Yani Arduino'nun pinleri 50mA çıkış verirken motorlar için 1000mA(1A ) gerekmektedir.

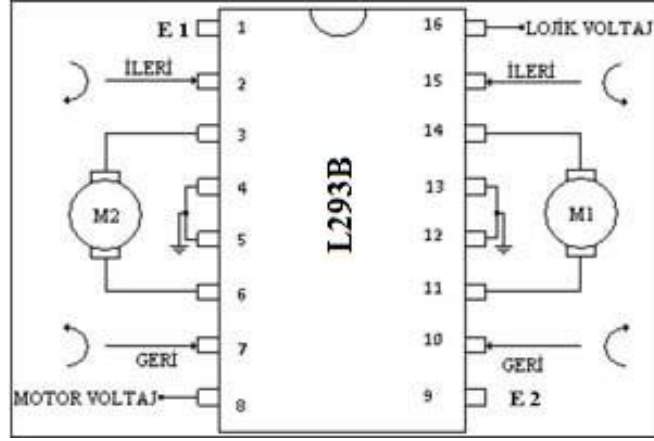
Motor sürücü devreler ile mikrodenetleyicilerin çıkışlarından alınan sinyaller yükseltilerek motorların kontrolü sağlanır. Motor sürücü devreler transistörler kullanılarak H köprüsü ve benzeri şekillerde hazırlanabilir. Ancak genellikle kolaylık açısından motor sürücü entegre devreler tercih edilmektedir.

Robotikte en sık kullanılan motor sürücü entegre devreler; DC motor kontrolleri için L293D, L293B, L298 motor sürücü entegrelerdir. Motor sürücü entegre seçiminde temel özellik entegrenin kullanım voltajı ve akım sınırı gibi özellikleridir.



Şekil 4. L293B Entegresi

L293D ve L293B motor sürücü entegreleri içlerinde iki adet H köprüsü barındıran 16 bacaklı motor sürücü entegrelerdir. Genellikle DC motor kontrolünde tercih edilen motor sürücü entegreler olan L293D ve L293B ile iki motor birbirinden bağımsız olarak çift yönlü kontrol edilebilmektedir.



Şekil 5. L293B Datasheet

L293B motor sürücü entegrenin ise 4,5 V ile 36 V aralığında, maksimum 1,2 A akım sınırına kadar kullanılması mümkündür.

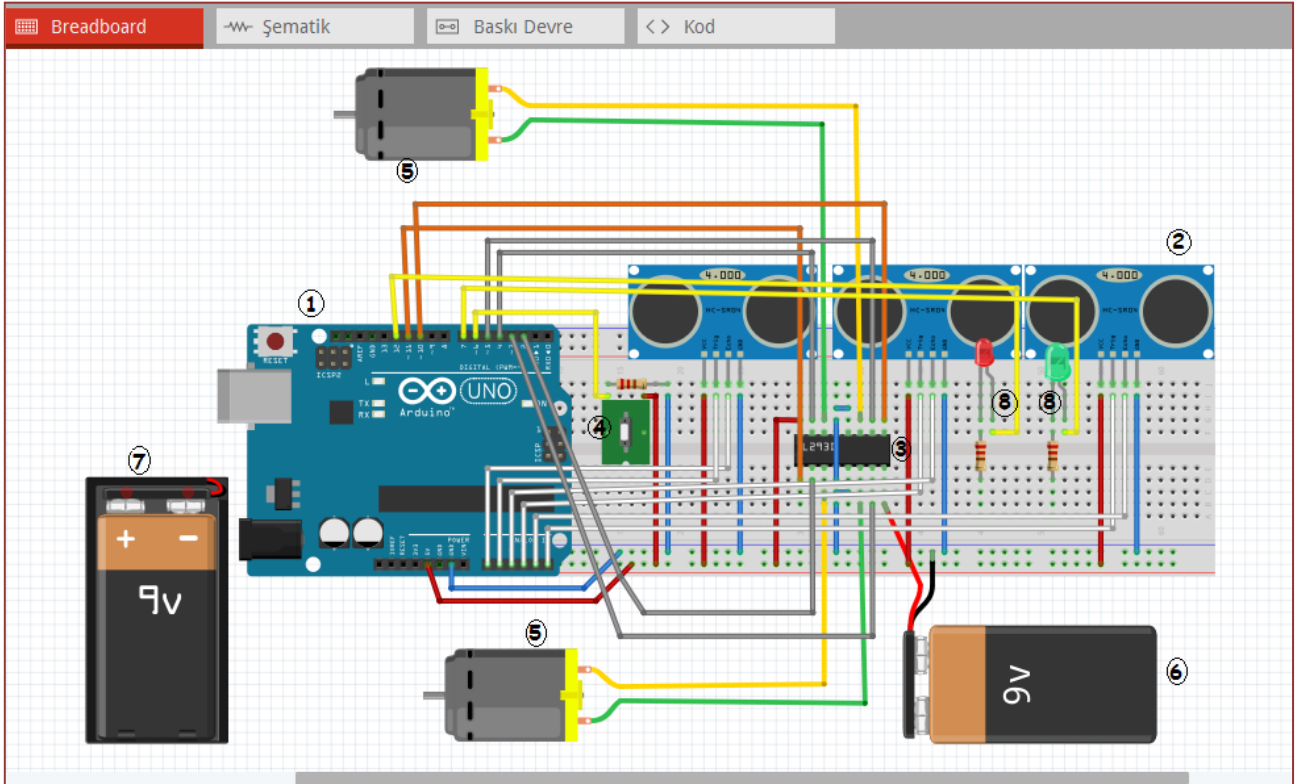
Ayrıca L293 motor sürücü entegrelerin enable bacaklarının (1. ve 9. bacaklar) kullanılmasıyla PWM kontrolü de yapılabilmektedir. PWM kontrol sayesinde motorun hızı istenen seviyede ayarlanabilir.

## 2. YAPILAN ÇALIŞMALAR

Yapılan tez çalışmasında engelden kaçan robot tasarımı gerçekleştirilmiştir. Tasarımda donanım aşaması ve yazılım aşaması olmak üzere iki kısım mevcuttur. Tezin bu bölümünde donanım ve yazılım aşamalarının nasıl gerçekleştirildiği adım adım anlatılmıştır.

### 2.1. DONANIM

Projenin donanım kısmında kullanılan devre elemanları ve board üzerindeki bağlantıları Fritzing programı ile çizilerek gösterilmiştir.



Şekil 6. Robotun devre şeması

1 numaralı kısımda kullanılan Arduino Uno görülmektedir. Öncelikle Arduino Uno'nun 5V ve GND pinlerinden board'a bağlantı yapılarak 5V hattı ve toprak hattı oluşturuldu.

2 numaralı kısımda görünen 3 adet HC-SR04 kullanılmıştır. Bunlardan biri robotun ön kısmında yer alırken diğer ikisi biri sağda biri solda olmak üzere yan kısımlara yerleştirilmiştir. Her birinin Vcc pinleri board üzerindeki gerilim hattına, GND pinleri ise toprak hattına bağlanmıştır. Trigger ve Echo pinleri içinse Arduino üzerinde bulunan 6 tane analog giriş pini kullanılmıştır. Yalnız bu pinler kod içerisinde dijital pin olarak kullanılarak dijital çıkış değerleri okunmuştur.

3 numaralı kısımda bulunan L293B entegresinin de gerekli bağlantıları board ve Arduino üzerine yapılmıştır. Burada önemli olan nokta entegrenin 1 ve 9 numaralı enable bacakları motor hızının kontrolünü sağlayabilmek için Arduino'nun dijital pinlerinden tilda(~) işaretli olan ve PWM çıkışı destekleyen iki pinine bağlanmıştır.

4 numaralı kısımda görülen ve aracın çalışmasını başlatıp durdurmaya yarayan switch anahtarının da gerekli bağlantıları yapılmıştır.

5 numara ile gösterilen sađ ve sol motorların ileri ve geri gitmesini sađlayan ikişer kablosu ise L293B entegresinin ilgili bacaklarına bađlanmıştır.

6 numaralı kısımda görölen 9Vluk pil ile de L293B entegresinin 8 numaralı bacağından entegreye gerekli voltaj verilmiştir.

7 numaralı kısımdaki pil yatađı aracılıđı ile Arduino'yu besleyecek olan gerilim kaynak bađlantısı Arduino'nun power yakından verilmiştir.

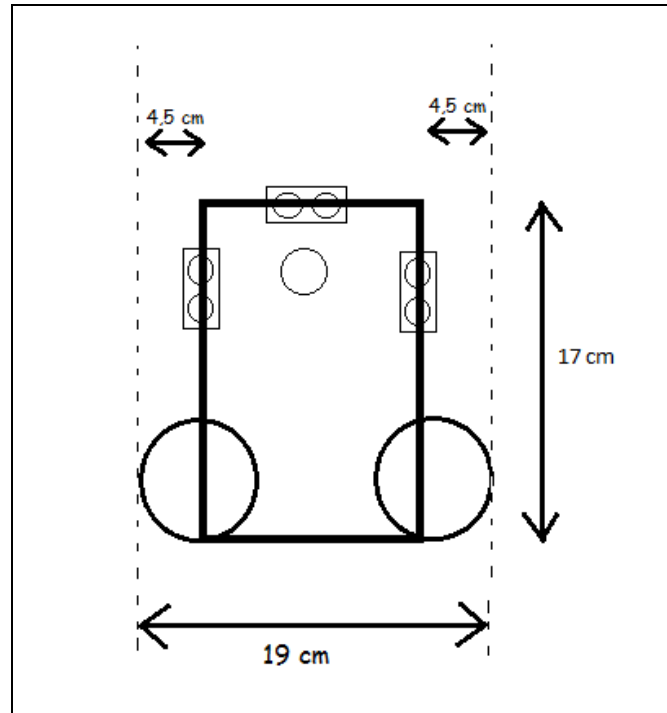
8 numaralı kısımlardaki ledlerin de gerekli bađlantıları yapılarak arabanın donanım kısmı tamamlanmış olmaktadır. Burada araç durur vaziyette olduđu sürece kırmızı led yanar iken çalışır vaziyete geçince kırmızı led sönüp yeşil led yanmaya başlayacaktır.

Motor sürücü entegresi üzerinde bulunan pinlerden sađ ve sol motor için enable, ileri ve geri pinlerinin Arduino pinleri ile olan bađlantıları řu şekildedir;

Tablo 1. L293B ile Arduino arasındaki sađ ve sol motor ile ilgili pin bađlantıları

	L293B pinleri	Arduino pinleri
Sađ Motor Enable	1	11
Sađ Motor İleri	2	2
Sađ Motor Geri	7	3
Sol Motor Enable	9	10
Sol Motor İleri	15	4
Sol Motor Geri	10	5

Proje için oluşturulan aracın řase kısmı arkada iki tane tekerlek ön orta kısımda bir sarhoş tekerlek kullanılarak tasarlanmıştır. Mesafe ölçümü yapacak olan sensörler ise biri aracın tam önüne diđer ikisi ise sađda ve solda olmak üzere iki yana yerleştirilmiştir.



Şekil 7. Aracın řase tasarımı ve ölçüleri

## 2.2. YAZILIM

### 2.2.1. Arduino Programlama Komutları

Arduino ile üzerine yüklenmiş olan aygıt yazılımı sayesinde Arduino programlama dilinde program geliştirilebilir. Arduino Programlama dili basitleştirilmiş C++ kullanır. Genel olarak üç bölümden oluşur: Tanımlamalar, kurulum ve ana program bloğu. Arduino’da yazılan programlara sketch adı verilir. Genel yapısı aşağıdaki şekildedir;

Tablo 2. Bir Arduino programının genel yapısı

```
//Tanımlamalar
void setup() {
// Kurulum kodları buraya yazılır. (Bir defa çalışır)
}
void loop() {
// Ana program kodları buraya yazılır. (Sürekli çalışır)
}
```

#### setup() Fonksiyonu

Arduino’ya enerji verildiğinde veya yeniden başlatıldığında setup() fonksiyonu bir defa çalışır. Bu fonksiyon değişkenler, pin modları, seri iletişim, kütüphaneler için kullanılır.

#### loop() Fonksiyonu

Döngü kısmı setup() fonksiyonundan sonra döngü şeklinde sürekli çalışır. Ana program kodları bu fonksiyon içine yazılır.

Arduino Programlama dilinin temel özellikleri;

- Program yazımı belirli kalıpta, bloklar halinde olur.
- Bloklar, { } parantezleri ile oluşturulur.
- Komutlar aynı satıra veya alt alta satırlara yazılabilirler.
- Tüm komutlar, noktalı virgül (;) ile biter. Yalnız blok başlatan ifadelerden sonra noktalı virgül kullanılmaz.
- Programda kullanılan tüm değişkenler ve bilgi tipleri bildirilir.
- Programın başında kütüphaneler aktifleştirilir, çağrılır. Kütüphaneleri çağırmak için #include ifadesi kullanılır.
- Açıklamalar tek satır ise “/ ... /” şeklinde ve birden fazla satır ise “/\* ... \*/” şeklinde yazılır.
- #define ile eşdeğer ifade atanır.

#### Fonksiyonlar

- pinMode() fonksiyonu pinleri giriş veya çıkış olarak yapılandırma işlemi yapar. Bu yapılandırma işlemi setup() kısmında gerçekleştirilir. pinMode(pin numarası, INPUT) veya pinMode(pin numarası, OUTPUT) şeklinde kullanılır.
- digitalWrite() fonksiyonu çıkış olarak ayarlanan pinlerin değerlerini, HIGH veya LOW olarak ayarlar. digitalWrite(pin numarası, HIGH) veya digitalWrite(pin numarası, LOW) şeklinde kullanılır.

- digitalRead() fonksiyonu belirtilen digital pin deęerini okur. digitalRead(pin numarası) şeklinde kullanılır.
- analogReference() fonksiyonu analog giriş için referans gerilimini ayarlar.
- analogRead() fonksiyonu belirtilen analog pin deęerini okur.
- analogWrite() fonksiyonu ayarlanan pinden analog çıkış almayı sağlar. LED parlaklığı, motor hızı ayarlama gibi işlemlerde kullanılır.
- delay() fonksiyonu içerisine yazılan deęeri milisaniye cinsinden kabul ederek o süre boyunca gecikme sağlar.

## Seri Port Üzerinden Haberleşme

Seri iletişim en basit anlamıyla dijital bilginin yani 1 ve 0'ların tek bir hat üzerinden peşi sıra iletilmesi anlamına gelmektedir.

Arduino'nun üzerinde seri iletişim birimi bulunmaktadır. Bu birimler vasıtasıyla seri iletişim yürütülür. Arduino'da seri iletişim için Serial adında bir kütüphane bulunur. Bu kütüphanenin mevcut fonksiyonlarından en sık kullanılanları begin(), print() ve println() fonksiyonlarıdır. Arduino programının setup() kısmında Serial.begin(9600); komut satırı ile seri haberleşme başlatılmış olur. Programın loop() kısmında ise Serial.print(); komut satırı kullanılarak fonksiyon içerisine ekranda deęerini görmek istediğimiz deęişkeni veya çift tırnak içerisinde ("...") ekranda görölmesini istediğimiz metni yazarak seri haberleşmeyi gerçekleştirmiş oluruz.

### 2.2.2. Projenin Kodlarının Yazılması

Projenin kodlama kısmında öncelikle programın en üst kısmına setup() fonksiyonunun üzerine donanım kısmında oluşturulan devre elemanlarının Arduino pinleri ile olan bağlantılarının tanımlamaları ve program içerisinde kullanılacak deęişken ve sabitlerin tanımlamaları ile gerekli olanlara ilk deęer atamaları yapıldı.

Tablo 3. Pin bağlantıları, deęişkenler ve sabitlerin tanımlanması

```
//mesafe sensörünün pinleri
const int trigPinFront = A0;
const int echoPinFront = A1;
const int trigPinLeft = A2;

const int echoPinLeft = A3;
const int trigPinRight = A4;
const int echoPinRight = A5;

// ledler
const int ledGreen = 7 ;
const int ledRed = 12;

// güç butonu
const int powerButton = 6;
```

```

unsigned long timeUltrasonicDelay = 0;

//Motor pinleri
const int leftMotorE = 10 ; //Sol motor için enable // hız kontrolu
const int rightMotorE = 11 ; // Sag motor için enable // hız kontrolu

const int right_i = 2; // sag motor için 1. control pin
const int right_g = 3; // sag motor için 2. control pin
const int left_i = 4; // sol motor için 1. control pin
const int left_g = 5; // sol motor için 2. control pin

//mesafe sensörleri için değişenler
int timeFront = 0;
int distanceFront = 0;
int timeLeft = 0;
int distanceLeft = 0;
int timeRight = 0;
int distanceRight = 0;

int buttonState ; //butonun durumunu tutan değişken

```

Arduino programlamada analog pinler A0, A1 şeklinde adlandırılırlar. Bu yüzden mesafe sensörlerinin analog pinlere bağlanan trigger ve echo pinlerinin tanımlanması A0, A1 şeklinde ilgili pinlerle eşleştirilmiştir.

Digital pinler ise arduino programlamada direk pin numarası olan 0, 1, 2 şeklinde adlandırılır. Bu yüzden de diğer devre elemanlarının arduino ile olan ilgili bağlantıları dijital pinler ile sağlandığı için tanımlamalarda ilgili pinlerin pin numarası kullanılmıştır.

Daha sonra setup() kısmında pin modları belirtilmiş ve gerekli yerlerde değişkenlerin değerlerini takip ederek programın doğruluğunu test etmek amaçlı kullanılacak olan seri iletişim başlatılmıştır.

Tablo 4. Arduino programının setup() kısmı

```

void setup() {
  //arduinonun pinlerinin giriş mi yoksa çıkışını olacağını burda belirtiyoruz.
  pinMode ( trigPinFront, OUTPUT );
  pinMode ( trigPinRight, OUTPUT );
  pinMode ( trigPinLeft, OUTPUT );

  pinMode ( echoPinFront, INPUT );
  pinMode ( echoPinLeft, INPUT );
  pinMode ( echoPinRight, INPUT );

  pinMode ( ledGreen, OUTPUT );

```

```

pinMode ( ledRed, OUTPUT );
pinMode ( powerButton, INPUT );

pinMode ( rightMotorE, OUTPUT );
pinMode ( leftMotorE, OUTPUT );

pinMode ( left_i, OUTPUT );
pinMode ( left_g, OUTPUT );
pinMode ( right_i, OUTPUT );
pinMode ( right_g, OUTPUT );

Serial.begin(9600); //Seri haberleşme
delay(3000); //araç başlamadan kendine gelmesi için verilen süre
}

```

Son olarak da loop() kısmında programın yapması istenilen işlevlerini gerçekleştirecek olan kodlar belirlenen algoritmaya göre düzenlenerek her bir işlev ayrı bir fonksiyon şeklinde yazılıp koda eklenmiştir. Yazılan bu fonksiyon tanımlamaları ve işlevleri de loop() kısmından önce koda eklenmiştir.

Tablo 5. Sensörlere ölçüm yaptıran fonksiyon

```

void mesafeolc()
{
    if((millis()- timeUltrasonicDelay) > 50) //only get the distance every 50ms
    {
        // ön sensör
        Serial.println("mesafe olctuk ");
        digitalWrite(trigPinFront , HIGH); //triggera işaret verme
        delayMicroseconds(10);
        digitalWrite(trigPinFront,LOW);

        timeFront = pulseIn(echoPinFront, HIGH); //sensör için mesafe hesaplama
        distanceFront = (timeFront/2) / 29.1; //29.1 sayısı sıcaklık ile değişir

        Serial.print("distanceFront = ");
        Serial.println(distanceFront);

        // sol senör
        digitalWrite(trigPinLeft , HIGH); //triggera işaret verme
        delayMicroseconds(10);
        digitalWrite(trigPinLeft,LOW);

        timeLeft = pulseIn(echoPinLeft, HIGH);
        distanceLeft = (timeLeft/2) / 29.1;
        Serial.print("distanceLeft = ");
    }
}

```



```

Serial.println(distanceLeft);

//sağ sensör
digitalWrite(trigPinRight , HIGH); //triggera işaret verme
delayMicroseconds(10);
digitalWrite(trigPinRight,LOW);

timeRight = pulseIn(echoPinRight, HIGH);
distanceRight = (timeRight/2) / 29.1;
Serial.print("distanceRight = ");
Serial.println(distanceRight);

timeUltrasonicDelay = millis();
}
}

```

Tablo 6. Belirlenen algoritmaya göre yol seçimi yapacak olan fonksiyon

```

void chooseDirection()
{
  if ( distanceFront > 20 || distanceFront < 0 ) // döndürme yapıyor
  {
    if ( distanceRight > 7 && distanceRight < 11 )
      ileri(200);
    else if ( distanceRight >= 11 )
    {
      analogWrite ( leftMotorE, 180 ); //sol motor için hız kontrolü
      analogWrite ( rightMotorE, 120 ); //sağ motor için hız kontrolü
      Serial.println ("ileri gidiyoruz sol daha hızlı");

      digitalWrite ( left_i, HIGH );
      digitalWrite ( left_g, LOW );
      digitalWrite ( right_i, HIGH );
      digitalWrite ( right_g, LOW );
      delay(20);
    }
    else if( distanceRight <=7)
    {
      analogWrite (leftMotorE, 130); //sol motor için hız kontrolü
      analogWrite (rightMotorE, 200); //sağ motor için hız kontrolü

      Serial.println ("ileri gidiyoruz sağ daha hızlı");
      digitalWrite ( left_i, HIGH ); //araba geri gidiyor
      digitalWrite ( left_g, LOW );
      digitalWrite ( right_i, HIGH );
      digitalWrite ( right_g, LOW );
      delay (50);
    }
    else if(distanceLeft <= 8)
    {
      analogWrite (leftMotorE, 180); // sol motor için hız kontrolü
      analogWrite (rightMotorE,130); // sağ motor için hız kontrolü
    }
  }
}

```

```

        Serial.println ("ileri gidiyoruz sol daha hızlı");
        digitalWrite (left_i, HIGH ); //araba geri gidiyor
        digitalWrite(left_g, LOW);
        digitalWrite(right_i, HIGH);
        digitalWrite(right_g, LOW);
        delay(50);
    }
}

else // önünde engel varsa
{
    if(distanceLeft <= 20 && distanceRight>20) {
        saga90(350);
        Serial.println("saga90");
    }
    if(distanceLeft >20 && distanceRight>20) {
        saga90(350);
        Serial.println("saga90");
    }
    if(distanceRight <=20 && distanceLeft>20) {
        sola90(350);
        Serial.println("sola90");
    }
    if(distanceRight<=20 && distanceLeft<=20){
        saga180(550);
        Serial.println("saga180");}
}
}

```

Tablo 7. Aracın hareketlerini belirleyecek olan fonksiyonlar

```

void ileri(int t)
{
    Serial.println("ileri gidiyoruz aynı hızlarda mesafe sağ 13-10 arasında ");
    analogWrite(leftMotorE, 135); // tam aracın düz gitmesi için 135 verildi
    analogWrite(rightMotorE, 120); // sağ motor için hız kontrolü

    digitalWrite(left_i,HIGH); //araba ileri gidiyor
    digitalWrite(left_g,LOW);
    digitalWrite(right_i,HIGH);
    digitalWrite(right_g,LOW);
    delay(t);

    dur(1);
}

void geri(int t)
{
    analogWrite(leftMotorE,120); // sol motor için hız kontrolü
    analogWrite(rightMotorE, 120); // sağ motor için hız kontrolü

    //Serial.println("Geri gidiyoruz");
    digitalWrite(left_i,LOW); //araba geri gidiyor
    digitalWrite(left_g,HIGH);
    digitalWrite(right_i,LOW);
}

```

```

    digitalWrite(right_g,HIGH);
    delay(t);
}

void dur( int t)
{
    Serial.println("duruyoruz");
    digitalWrite(left_i,LOW); //araba duruyor
    digitalWrite(left_g,LOW);
    digitalWrite(right_i,LOW);
    digitalWrite(right_g,LOW);
    delay(t);
    return;
}

void sola90(int t)
{
    Serial.println("Sola 90 derece donduk");
    analogWrite(leftMotorE, 120); // sol motor için hız kontrolu
    analogWrite(rightMotorE, 120); // sag motor için hız kontrolu

    digitalWrite(left_i,LOW);
    digitalWrite(left_g,HIGH);
    digitalWrite(right_i,HIGH);
    digitalWrite(right_g,LOW);
    delay(t);
    dur(1);
}

void saga90(int t)
{
    Serial.println("Saga 90 derece donduk");
    analogWrite(leftMotorE, 120); // sol motor için hız kontrolu
    analogWrite(rightMotorE, 120); // sag motor için hız kontrolu

    digitalWrite(left_i,HIGH);
    digitalWrite(left_g,LOW);
    digitalWrite(right_i,LOW);
    digitalWrite(right_g,HIGH);
    delay(t);
    dur(1);
}

void saga180(int t)
{
    Serial.println("Saga 180 derece donduk");

    analogWrite(leftMotorE, 120); // sol motor için hız kontrolu
    analogWrite(rightMotorE, 120); // sag motor için hız kontrolu

    digitalWrite(left_i,HIGH);
    digitalWrite(left_g,LOW);
    digitalWrite(right_i,LOW);
    digitalWrite(right_g,HIGH);
    delay(t);
    dur(1);
}

```

Kullanılacak bütün fonksiyon tanımlamaları loop() kısmının üzerinde yapıldıktan sonra son olarak güç butonunun ve aracın çalışır olma durumuna göre yanacak olan ledlerin durumunu belirleyecek kodlar ile birlikte gerekli fonksiyon çağırımları yapılarak loop() kısmı da tamamlanmıştır.

Tablo 8. Programın döngü halinde sürekli çalışan loop() kısmı

```
void loop()
{
  buttonState = digitalRead(powerButton);
  while(buttonState == HIGH) //güç butonum açık mı
  {
    digitalWrite(ledRed, LOW);
    buttonState = digitalRead(powerButton);

    if(buttonState == LOW)
    {
      digitalWrite(ledGreen, LOW);
      digitalWrite(ledRed, HIGH);
      Serial.println("led sondu");
      break;
    }
    digitalWrite(ledRed, LOW);
    digitalWrite(ledGreen, HIGH); // araba çalışıyor yeşil led yanıyor

    mesafeolc(); //mesafe sensörüne işaret gönderip mesafeyi ölçüyoruz
    dur(50);
    chooseDirection();

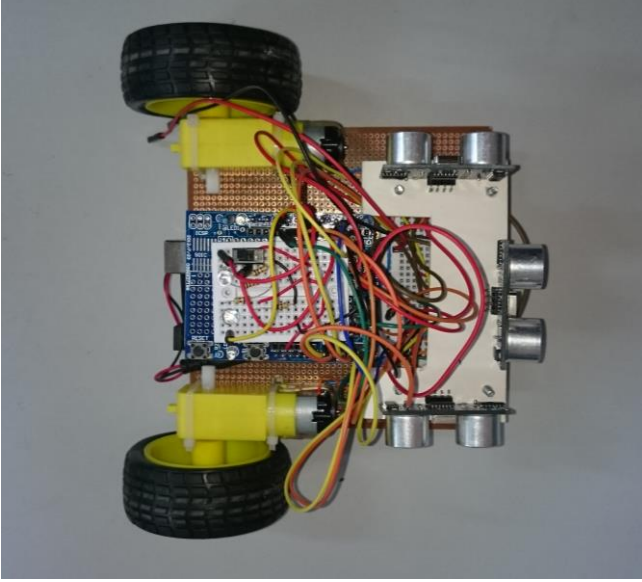
  }

  if(buttonState == LOW)
  {
    digitalWrite(ledGreen, LOW);
    digitalWrite(ledRed, HIGH);
    Serial.println("Kirmizi led yanıyor arac calismiyor");
    dur(50);

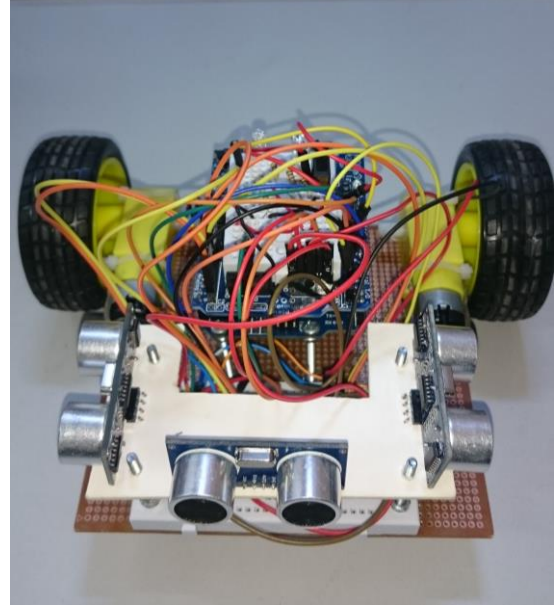
  }
}
```

### 3. SONUÇLAR

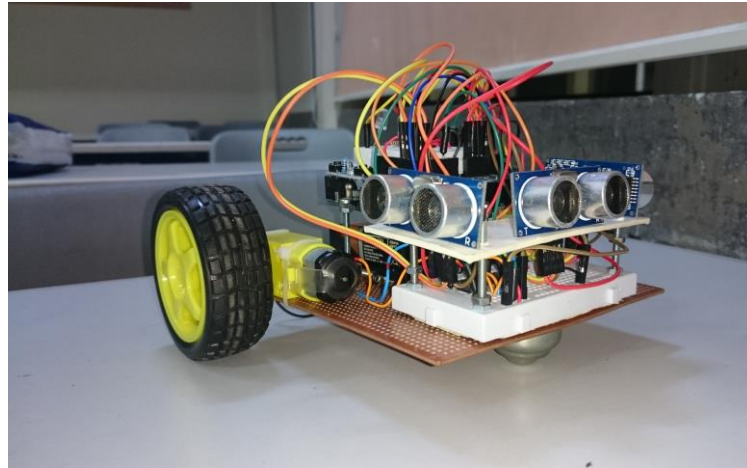
Projenin sonucunda arduino ile engelden kaçan bir aracın yapılması tam olarak kavranmıştır.



Şekil 8. Aracın üstten görünüşü



Şekil 9. Aracın önden görünüşü



Şekil 8. Aracın genel görünüşü

#### 4. ÖNERİLER

Arduino ile proje yapılırken öncelikle Arduino ile programlamanın temellerini iyi kavramak gerekmektedir. Bunun için Arduino programlama öğrenirken ledler, dirençler switch butonlar gibi temel devre elemanları kullanılarak basit örnek projeler oluşturularak işe başlanmalıdır. Bu temel devre elemanları ile oluşturulan basit devreler yardımıyla arduino programlamanın mantığı kavrandıktan sonra yapılacak olan asıl proje için gerekli olan elemanlar örneğin sensörler, sürücüler, kullanılacak olan motorlar tespit edilmelidir. Projede kullanılacak olma tüm elemanlar da projeye başlamamadan önce test edilmeli ve nasıl kullanıldıkları en iyi şekilde öğrenilmelidir. Daha sonra projenin amacına uygun bir tasarım yapılarak elemanların montesi gerçekleştirilmelidir. Sonra da projede gerçekleştirilmek istenen amaca uygun bir algoritma geliştirilerek Arduino programı yazılmalıdır. Son olarak testler yapılarak düzeltilmesi gereken kodlar veya donanımsal aksaklıklar varsa bunlar düzeltilmesi ve son kez testleri yapılarak proje bitirilmelidir.

## 5. KAYNAKLAR

### Kitaplar:

1. Robotik Mekatronik ve Yapay Zeka
2. Projeler ile Arduino Erdal Delebe

### İnternet:

3. <http://www.instructables.com/>
4. <http://letsmakerobots.com/>
5. <https://www.arduino.cc/>
6. <https://www.youtube.com/>

**STANDARTLAR ve KISITLAR FORMU**

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Daha önce benzer projeler yapılmıştır. Yaptığım projenin kod kısmı tamamen bana aitken görünümünün bir kısmı benim tasarımımdır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Bir robotun nasıl engeli algılayıp ona göre yolunu belirleme problemini çözdüm

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Algoritma, donanım ve yazılım konularında edinmiş olduğum bilgileri kullandım.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

Sadece planlı bir şekilde ilerlemeye çalıştım.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

- a) Ekonomi

Mali durumun daha iyi olması durumunda proje daha fazla geliştirilebilir.

- b) Çevre sorunları:

Projenin yapılması ve denenmesi yurt imkânları kısıtlı olduğu için çok zor oldu.