

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



LABİRENT ÇÖZEN ROBOT

TASARIM PROJESİ

**Fatih ŞEFTALİ
Saim KURBAN**

**2015-2016 GÜZ DÖNEMİ
KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

LABİRENT ÇÖZEN ROBOT

TASARIM PROJESİ

**Fatih ŞEFTALİ
Saim KURBAN**

Bu projenin teslim edilmesi ve sunulması tarafımda uygundur.

Danışman : Doç. Dr. Mustafa ULUTAŞ

.....

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, eleştiriyi kabul etmek ve eleştiriyi 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

Bu tez çalışmasında algoritmik bir problem olan labirent çözümleme işlemi donanım ve yazılım bir arada kullanılarak gerçekleştirilmiştir.

Öncelikle tez konusunu seçerken isteklerimizi göz önünde bulundurup bize yardımcı olan tez danışmanımız Doç. Dr. Mustafa ULUTAŞ hocamıza teşekkürlerimizi sunarız. Bu zorlu tez sürecinde bizden desteğini bir an için bile esirgemeyen değerli sınıf arkadaşımız, Sefa DURMUŞ ve ev arkadaşımız Sinan İMAMOĞLU' na, tüm eğitim hayatımız boyunca bizden maddi ve manevi desteklerini esirgemeyen her zaman yanımızda olan sevgili ailelerimize teşekkürlerimizi bir borç biliriz.

Fatih ŞEFTALİ
Saim KURBAN
Trabzon 2015

İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	II
ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
1. GENEL BİLGİLER	1
1.1.Giriş	1
2. YAPILAN ÇALIŞMALAR	2
2.1.LABİRENT ROBOTU TASARIMI	2
2.2.LABİRENTİN TASARIMI	3
2.3.ROBOTUN MEKANİK YAPISI	3
2.4.MOTORLAR VE MOTOR SÜRÜCÜ DEVRESİ	4
2.5.ROBOTTA KULLANILAN SENSÖRLER	5
2.6.MİKRODENETLEYİCİ KATMANI (ARDUNİO MEGA)	6
2.7.ROBOTUN ALGORİTMİK ŞEMASI(AKIŞ ŞEMASI)	8
2.8.AKIŞ ŞEMASINDA GÖSTERİLEN İLGİLİ BLOKLARA DENK GELEN KOD PARÇALARI VE AÇIKLAMALARI	9
2.9.ROBOTUN PROGRAMINDA KULLANILAN ÖNEMLİ FONKSİYONLAR	13
3. SONUÇLAR	16
4. ÖNERİLER	17
5. KAYNAKLAR	18
STANDARTLAR ve KISITLAR FORMU	19

ÖZET

Bu çalışmada labirent çözen robot gerçekleştirilmiştir. Robot mikrodenetleyici (Arduino Mega) ile kontrol edilmektedir. Robotun amacı girilen labirentten çıkmaktır. Labirenti çözen robot ilerleme sırasında doğrultusu boyunca meydana gelen bozulmaları otomatik düzeltebilmektedir. Tasarlanan robotun boyutları, şekli, hareket organları ve kullanılan sensörler değiştirilerek, insanlar için uygun olmayan fiziksel ortamlarda veya daha farklı alanlarda kullanılabilir.

1. GENEL BİLGİLER

1.1.Giriş

Robotlar ile labirent çözüme 1970'li yıllara dayanmaktadır. Hatta 1970'lerin ortalarından bu yana her yıl IEEE tarafından Micro Mouse Contest isimli labirent çözüme yarışmaları düzenlenmektedir. Yarışmalardaki amaç, labirentin orta noktasını en kısa sürede bulan bir robot tasarlamak ve gerçekleştirmektir. Gerçekleştirilen robotlarda, labirent çözüme programlarında 3 temel algoritma kullanılmaktadır. Bunlar rasgele arama, labirenti haritalama ve sağ veya sol duvar takibi metotlarıdır. Rasgele arama metodu bu üç metot içerisinde en iyisi, en kısası veya en hızlısı değildir ama mantıksal olarak en basit olan metottur. Bu metotta robot, labirent içerisinde rasgele ilerlemektedir. Robot sağında veya solunda bir boşluk olduğunda sağa veya sola dönme kararını rasgele vermektedir. Rastgele arama metodunun tek dezavantajı labirent çözüme işleminin daha uzun süre alması ve muhtemelen istenen süreden daha fazla zaman gerektirmesidir. Ayrıca robot dönüşler sırasında yapacağı olası yanlış kararlarla saatlerce labirent içerisinde dolaşabilir. Dolayısıyla programlama konusunda biraz daha zaman harcayarak farklı algoritmalar kullanmak daha akılcıdır.

Bu yöntemler içerisinde en işlevsel olanı duvar izleme yöntemidir. Bu yöntemde robot labirent içerisinde sağ veya sol yanındaki duvarı izlemektedir. Bu metodu, en iyi kendimizi bir labirentin içerisinde gözleri kapalı olarak hayal ederek açıklayabiliriz. Eğer bir elimizle duvarı tutabilirsek ve çıkış da bir "ada" değilse, eninde sonunda çıkışı bulabiliriz. Ama çıkışa ulaşmaya kadar mutlaka bir duvar takip edilmelidir [1,2,3].

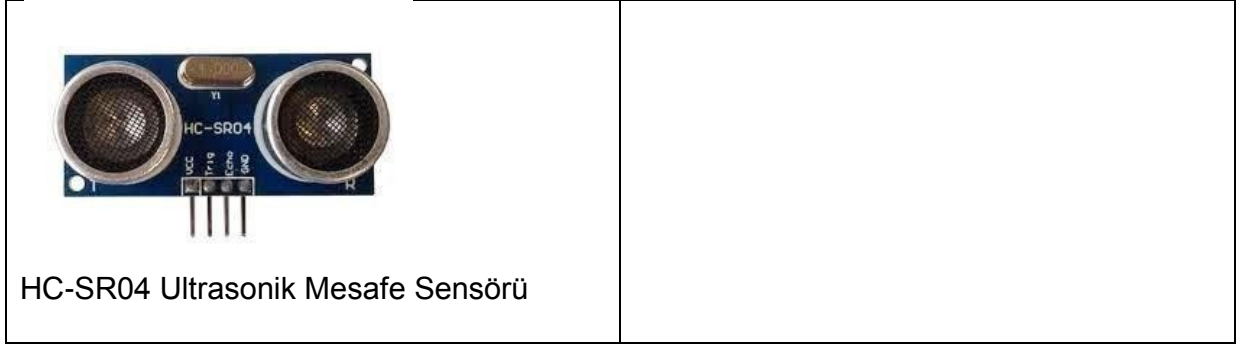
2. YAPILAN ÇALIŞMALAR

2.1. Labirent Robotu Tasarımı

Robotun mekanik yapısı; Fotoblok malzeme, 6.5cm çapında 2 adet plastik tekerlekler ve 2 adet DC motordan oluşmaktadır. Elektronik yapısında ise 4 adet ultrasonik mesafe sensörü, L293B motor sürücü entegresi ve işlemci devresi(Ardunio Mega), 1 adet RGB LED ve iki adet dijital buton bulunmaktadır.

Tablo 1.Labirent robotu tasarımında kullanılan malzemeler.

	
Ardunio Mega	Fotoblok malzeme
	
L293B motor sürücü entegresi	RGB LED
	
Plastik tekerlekler ve DC Motorlar	Dijital Buton



2.2. Labirentin Tasarımı

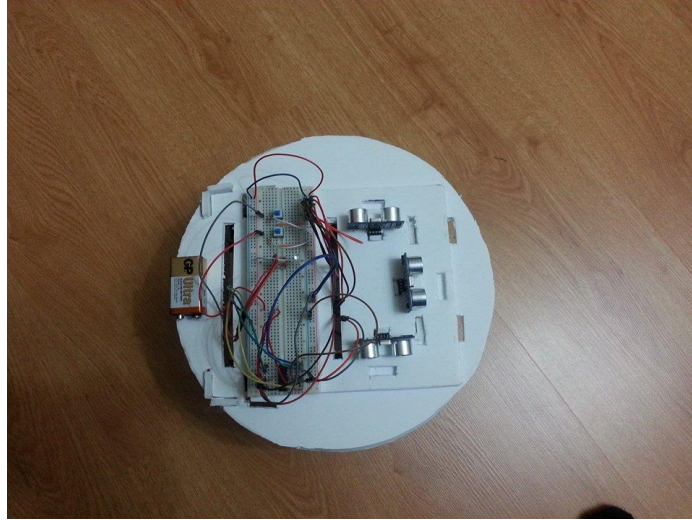
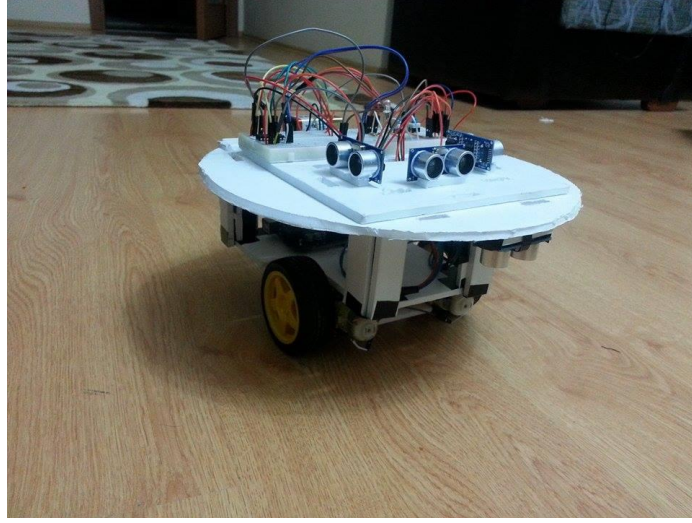
Labirent, 200cm eninde, 200cm uzunluğunda karton malzemenen yapılmıştır. Bu zemin üzerine duvarlar 40cm aralıklarla yerleştirilmiştir. Duvarlar 40cm eninde, 20cm yüksekliğindedir.

2.3. Robotun Mekanik Yapısı

Robotun gövdesi 17cm çapında, 15cm yüksekliğinde fotoblok malzemenen yapılmıştır.

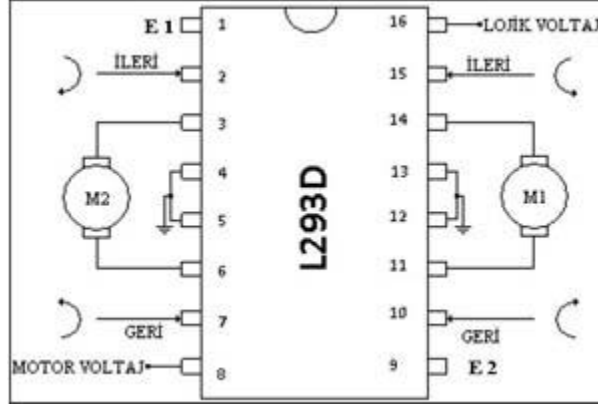
Step motorlar alt katman üzerine sırt sırta yerleştirilmişler ve plastik tekerlekler de step motorlara geçirilmişlerdir. Labirent üzerindeki duvarları algılamak üzere 3 adet sensör üst katmana duvarlara bakacak şekilde ve labirent bitişini anlamak için yine bir adet sensör üst katmana zemine bakacak şekilde yerleştirilmiştir. Robotun üst katmanında o anki çalışma durumu veya sonlanma durumunu gösteren bir adet RGB LED ve robotun kumanda(Başlama/Bitiş ve Algoritma seçimi) butonları yer almaktadır.

Tablo 2.Labirent robotunun perspektif ve üstten görünümü.



2.4. Motorlar ve Motor Sürücü Devresi

Robotun hareketini 2 adet adım motoru (step motor) sağlamaktadır. Kullanılan adım motoru hızı 250 RPM' dir. Motorlar; L293B motor sürücü devresi tarafından sürülmektedirler. Şekil 1'de motorları kontrol eden bu sürücü devre ve bağlantı noktaları görülmektedir. Bu devre için gerekli olan motor kontrol sinyalleri mikrodenetleyici(Arduino Mega) tarafından sağlanmaktadır. Bu motorların beslemesi 9 volt 'tur. Devre tam yük altında iken 160mA akım çekmektedir [4, 7].



Şekil 1.Motor sürücü devresi ve bağlantıları.

2.5. Robotta Kullanılan Sensörler

Hc-sr04 ultrasonik sensör sonar(Sound Navigation and Ranging) kullanarak karşısındaki nesneye olan mesafesini hesaplayan bir giriş kaynağıdır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığını boyutunu elde etmemizi sağlar. Bu tür sensörlerin esin kaynağı yunuslar ve yarasalardır. Onlarda sonar ile iletişim kurar ve hareket eder.

2cm ile 400cm arası mesafe en sağlıklı okuma yaptığı aralıktır. Üzerinde bir alıcı ve bir verici modül bulunur.

Hc-sr04 ultrasonik mesafe sensörü zaman diyagramı aşağıda gösterilmektedir. Ölçümün başlaması için Trig ucu en az 10 µs de +5V almalıdır. Böylelikle sensörden 40 KHz' de 8 devir ses dalgası dışarıya iletilir ve yansıması beklenir. Hc-sr04 alıcıdan ses dalgasını aldığıında Echo ucu 0V dan 5V a geçer ve mesafeyle orantılı bir süre bekler. Echo ucundaki genişliği ölçerek, aradaki mesafeyi elde edebilir [5].

Zaman = Echo uç genişliği, [µs]

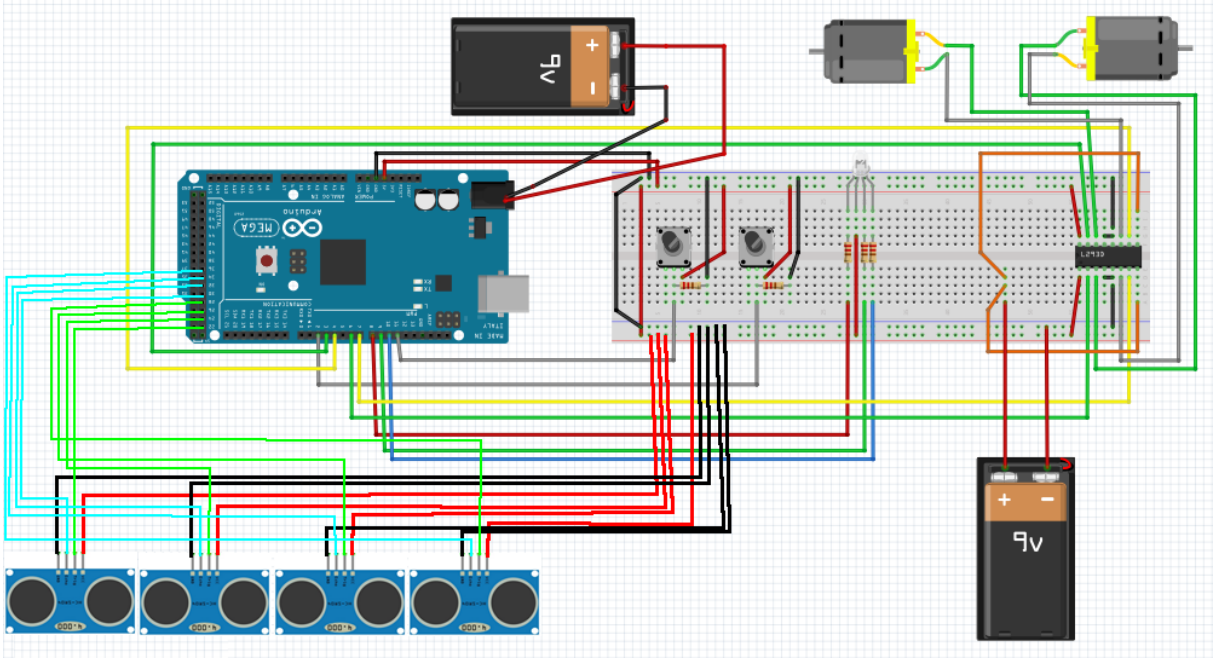
- Cm cinsinden mesafe = Zaman / 58

2.6. Mikrodenetleyici Katmanı (Arduino Mega)

Arduino Mega 2560 tabanlı bir Arduino kartıdır. 54 dijital I/O ucu vardır. Bunların 14 tanesi PWM çıkışı olarak kullanılabilir. 16 analog girişi, 4 UART (serial port), 16 MHz kristal osilatörü, USB bağlantısı, adaptör girişi, ICSP çıkışı ve bir reset butonu vardır. Arduino Duemilanove ve Diecimila için tasarlanmış tüm eklentilere (shield) uyumludur [6].

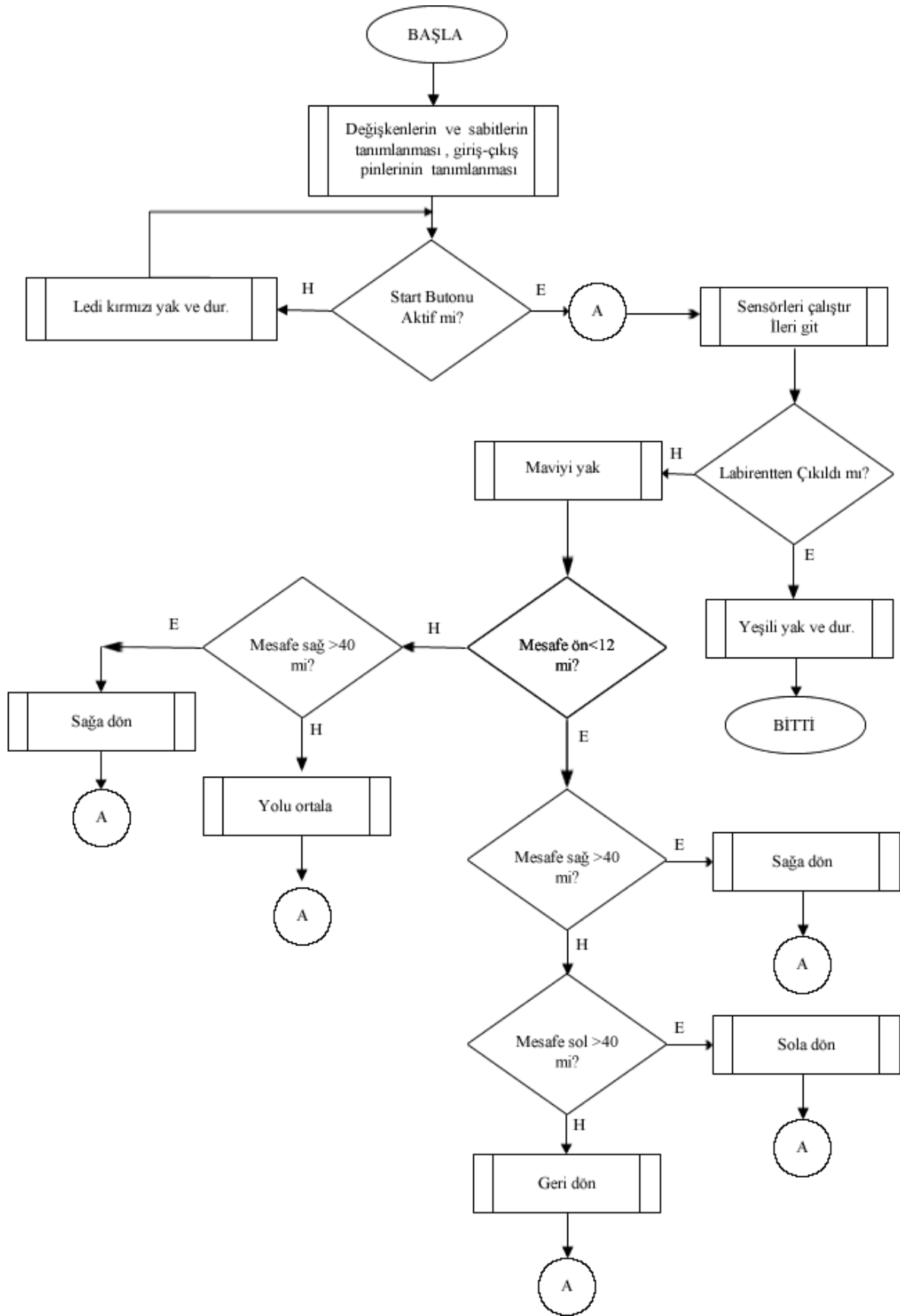
Özellikler

Mikrodenetleyici	ATmega2560
Çalışma Gerilimi	5V
Besleme Voltajı (Önerilen)	7-12V
Besleme Voltajı (Limit)	6-20V
Dijital I/O uçları	54 (14ü PWM çıkışı)
Analog Giriş Uçları	16
I/O Uçlarının Akımı	40 mA
3.3V Uçlarının Akımı	50 mA
Flash Bellek	256 KB (8KB'ını bootloader kullanıyor)
SRAM	8 KB
EEPROM	4 KB
Saat Frekansı	16 MHz [6].



Şekil 2.Devrede kullanılan elemanlar ve bağlantıları ile devre şeması gösterilmiştir.

2.7. Robotun Algoritmik Şeması(Akış Şeması)



Şekil 3. Labirent robotunun akış şeması.

2.8. Akış Şemasında Gösterilen İlgili Bloklara Denk Gelen Kod Parçaları Ve Açıklamaları



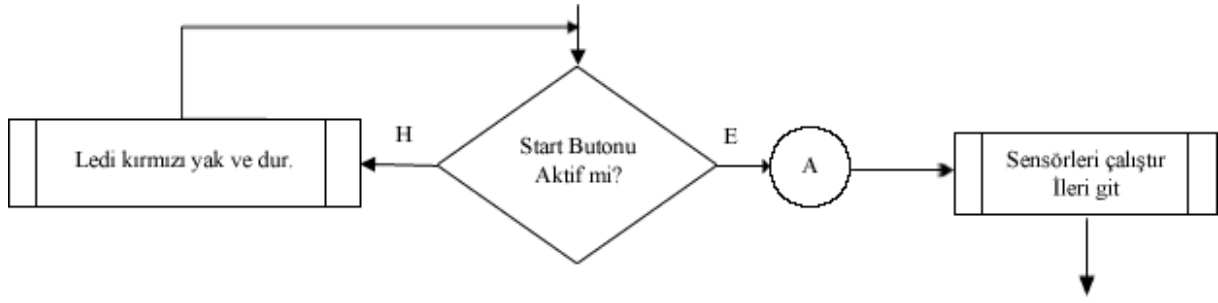
Şekil 3.1. Akış şemasında gerekli tanımlamaların yapıldığı kısım.

Değişken tanımlaması ile ilgili kısım aşağıdaki gibidir.

```
...  
#define BaslangicBtn 2  
#define SolMotorileri 3  
#define SolMotorGeri 4  
#define SagMotorileri 6  
#define SagMotorGeri 7  
...
```

Ardunio' da setup() fonksiyonu içinde ilk kurulum setlemeleri yapılır ve bir defaya mahsus programın çalışması ile birlikte koşar. Giriş çıkış uçlarının tanımlanması setup() fonksiyonu içinde aşağıdaki gibi yapıldı.

```
void setup() {  
...  
//motor çıkışları  
pinMode(SolMotorileri, OUTPUT);  
pinMode(SolMotorGeri, OUTPUT);  
pinMode(SagMotorileri, OUTPUT);  
pinMode(SagMotorGeri, OUTPUT);  
  
// Başlatma ve Algoritmalar  
pinMode(BaslangicBtn, INPUT);  
pinMode(AlgoritmaBtn, INPUT);  
  
////Sensor giriş ve çıkışları  
pinMode(trigSag, OUTPUT);  
pinMode(echoSag, INPUT);  
pinMode(trigOn, OUTPUT);  
pinMode(echoOn, INPUT);  
pinMode(trigSol, OUTPUT);  
pinMode(echoSol, INPUT);  
pinMode(trigAlt, OUTPUT);  
pinMode(echoAlt, INPUT);  
...  
}
```



Şekil 3.2. Akış şemasında algoritmanın başlatılmasıyla ilgili kısım.

Ardunio programlama da loop() fonksiyonu içerisine sürekli yapılması istenen kod parçaları yazılır.

```

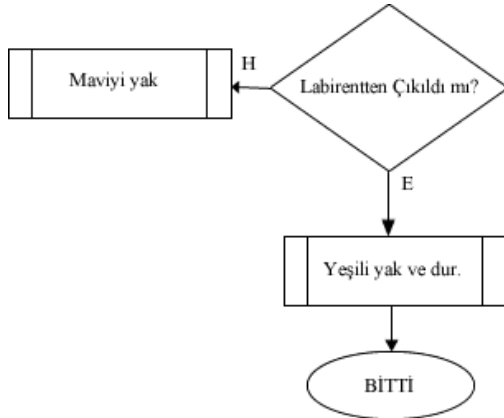
...
void loop() {

    BaslangicDurumu = digitalRead(BaslangicBtn);
    AlgoritmaDurumu = digitalRead(AlgoritmaBtn);

    if(BaslangicDurumu == HIGH){

        SensorleriCalistir();
    }
}

```



Şekil 3.3. Akış şemasında labirentten çıkılmasının kontrolünün yapıldığı kısım.

Labirent çözen robotun labirentin bitimini anlaması işlemi labirent sonuna konan 5cm yüksekliğinde engele gelindiğinde bunun mesafe sensörü ile algılanması mantığına dayanmaktadır. Bunun için farklı yöntemlerde uygulanabilir. Örneğin renk sensörü kullanılarak zemine çizilen çizginin algılanması ile de labirent çıkışı algılanabilir.

```

void loop() {
    ...
    if(mesafeAlt <= 5 )// Labirentin sonunda konulan çıkış engelinin kontrolü
    {
        YesiliYak();
    }
}

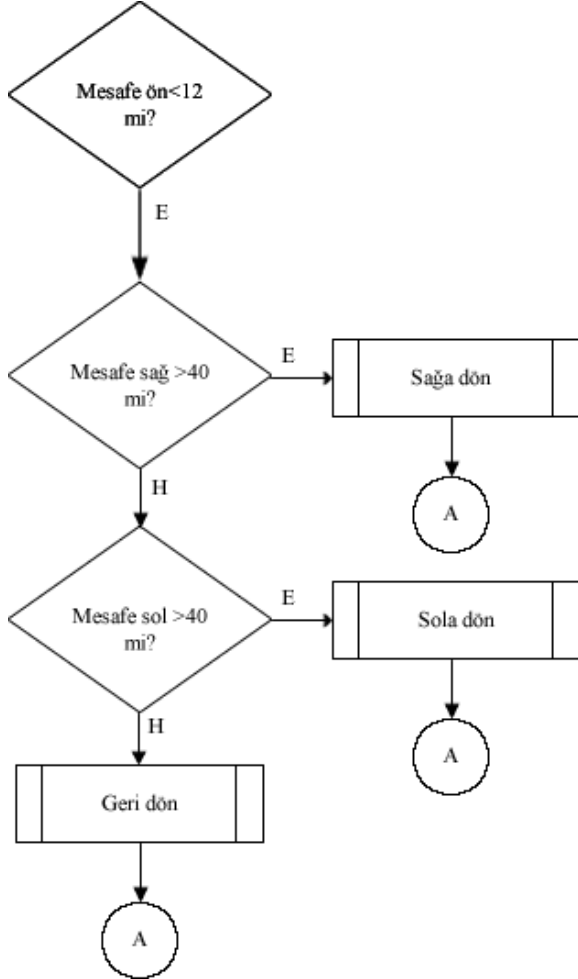
```



```

dur();
}
else{
MaviyiYak();//Mavi ışık robotun labirenti çözmeye devam ettiği
anlamına geliyor
... }

```

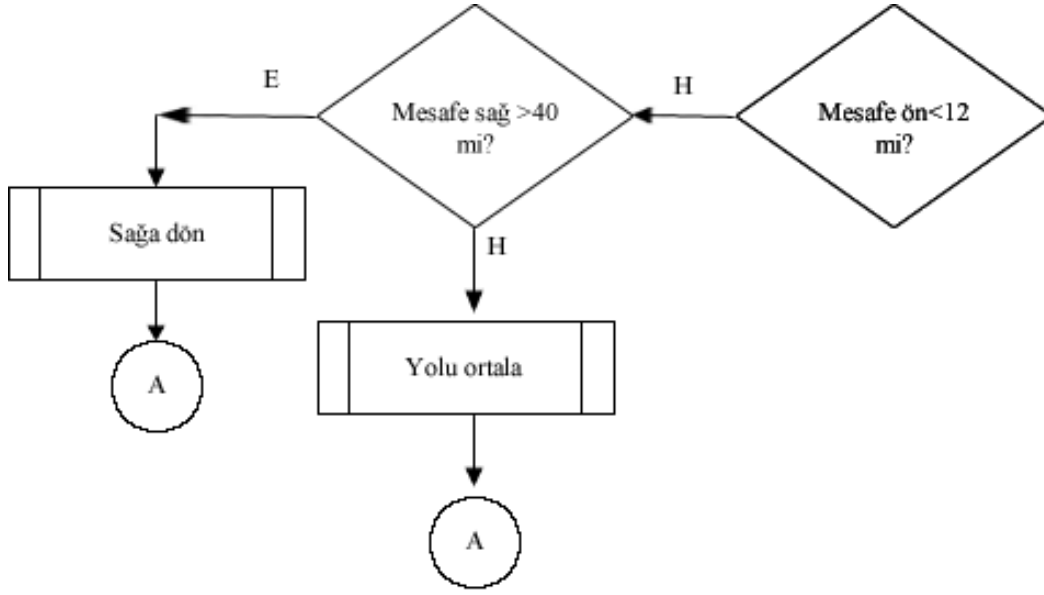


```

if(mesafeOn <= 12.0 ){ //17.5
dur();
if(mesafeSag > 40.0){ //30
dur();
//sagaDonmeDurumu =
true;
sagaDon();
}else if(mesafeSol >
40.0){
dur();
//solaDonmeDurumu =
true;
solaDon();
}else{
dur();
delay(1000);
geriDon();
}
}

```

Şekil 3.4. Akış şemasında ön mesafe<12 olduğunda sağa, sola, geriye dönme kontrolünün yapıldığı kısım.



Şekil 3.5. Akış şemasında mesafe ön<12 değilse mesafe sağ>40 ise sağa dönme değilse yolu ortalama işleminin yapılması gösterilmiştir.

```

...
else if(mesafeOn > 12.0 && (mesafeSag > 40.0 || mesafeSol > 40 ) ){//12
    sağıTakip = false;
    soluTakip = false;
    dur();
    delay(1000);
    ileri();
    delay(450);
    sagaDon();
    SensorleriCalistir();
    ileri();
    delay(450);
    dur();
}
else{
    YoluOrtala();
}
ileri();
delay(80);
dur();
//delay(5);
}
...

```

2.9. Robotun Programında Kullanılan Önemli Fonksiyonlar

DC motorların ileri, geri, sağ ve sola hamlelerini yapmasını sağlayan fonksiyonlar aşağıdaki gibidir.

```
void ileri ()
{
    digitalWrite(SolMotorGeri, LOW);
    digitalWrite(SolMotorileri, HIGH);
    digitalWrite(SagMotorGeri, LOW);
    digitalWrite(SagMotorileri, HIGH);
}

void geri ()
{
    digitalWrite(SolMotorileri, LOW);
    digitalWrite(SolMotorGeri, HIGH);
    digitalWrite(SagMotorileri, LOW);
    digitalWrite(SagMotorGeri, HIGH);
}

void sag ()
{
    digitalWrite(SolMotorGeri, LOW);
    digitalWrite(SolMotorileri, HIGH);
    digitalWrite(SagMotorileri, LOW);
    digitalWrite(SagMotorGeri, HIGH);
}

void sol ()
{
    digitalWrite(SolMotorGeri, HIGH);
    digitalWrite(SolMotorileri, LOW);
    digitalWrite(SagMotorileri, HIGH);
    digitalWrite(SagMotorGeri, LOW);
}
```

Robotun sağa, sola ve geriye dönmesinde kullanılan fonksiyonlar motor hızları ile tekerlek çaplarının kullanılmasıyla yapılan hesaplamalar sonucunda motorların hangi şekilde ve ne kadar milisaniye dönmesi gerektiğinin belirtildiği fonksiyonlardır. Bu fonksiyonlar aşağıdaki şekildedir.

```
void sagaDon () {
    dur ();
    delay (1000);
    sag ();
    delay (400);
    dur ();
    soluTakip = true;
    sagiTakip = false;
    dur (); }
void solaDon () {
    dur ();
    delay (1000);
```

```

    sol();
    delay(400);
    dur();

    sagiTakip = true;
    soluTakip = false;
    dur();
}
void geriDon(){

    dur();
    delay(1000);
    sag();
    delay(800);
    dur();

    if(sagiTakip == true){
        sagiTakip =false;
        soluTakip = true;
    }else{
        soluTakip =false;
        sagiTakip = true;
    }
}

```

Sensörlerle ölçüm yapılmasını sağlayan fonksiyon aşağıdaki şekildedir. Aşağıdaki fonksiyonda sadece tek sensör için hesaplama gösterilmiştir. Diğer sensörler içinde aynen geçerlidir sadece ilgili echo ve trig uçlarının belirtilmesi gerekmektedir.

```

void SensorleriCalistir(){
    ...
    digitalWrite(trigSag, HIGH);
    delayMicroseconds(100);
    digitalWrite(trigSag, LOW);

    sureSag = pulseIn(echoSag, HIGH);
    mesafeSag = (sureSag/2)/29.1;

    Serial.print("Sağa olan Mesafe : ");
    Serial.print(mesafeSag);
    Serial.println(" cm");
    ...
}

```

Son olarak aşağıda robotun düz gitmesi için gerekli olan en önemli fonksiyon bulunmaktadır. Robotta mevcut olan motorların birbiri ile aynı olmasına rağmen dönüş hızlarında ve buna bağlı olarak robotun düz gitmesi engelleyen vs. gibi mevcut problemler bu fonksiyonla giderildi. Fonksiyon sol ve sağ sensörlerden ölçülen mesafe bilgilerine göre

otomatik olarak kendisini hizalamasını sağlıyor. Bu hizalama sırasında robota 2 cm' lik bir tolerans değeri tanınmıştır.

```
void YoluOrtala(){
    if(mesafeSag < 30  && sagiTakip == true){
        if((mesafeSag-16) >= 2.0 ){
            sagaToparla();
        }
        else if((mesafeSag-16) <= -2.0 ){
            solaToparla();
        }
    }else if(mesafeSol < 30 && soluTakip== true){
        if((mesafeSol-16) >= 2.0 ){
            solaToparla();
        }
        else if((mesafeSol-16) <= -2.0 ){
            sagaToparla();
        }
    }else if(mesafeSag < 30 ){
        if((mesafeSag-16) >= 2.0 ){
            sagaToparla();
        }
        else if((mesafeSag-16) <= -2.0 ){
            solaToparla();
        }
    }else if(mesafeSol < 30 ){
        if((mesafeSol-16) >= 2.0 ){
            solaToparla();
        }
        else if((mesafeSol-16) <= -2.0 ){
            sagaToparla();
        }
    }
}
```

Bu fonksiyonda çağrılan yardımcı fonksiyonlar aşağıdaki gibidir.

```
void sagaToparla(){
    sag();
    delay(32);
    dur();
}
void solaToparla(){
    sol();
    delay(32);
    dur(); }
}
```

3.Sonuçlar

Robot, mekanik sistemleri ve bununla ilişkili kontrol ve algılama sistemleriyle bilgisayar algoritmalarına bağlı olarak akıllı davranan makinelerdir. Amerikan Robot Enstitüsü tarafından ise robot şu şekilde tanımlanmaktadır: ” Robot, yeniden programlanabilen; maddeleri, parçaları, aletleri, programlanmış hareketlerle yapılacak işe göre taşıyan veya işleyen çok fonksiyonlu makinedir”.

Tasarlanan robotun donanımsal olarak kullanılan sensörler ile mesafe ve engel algılaması, arduino ile programlanması ve farklı algoritmalara ve programlamaya göre çalışma ve farklı amaçlarda kullanımı gerçekleştirilebilir. Farklı amaçta kullanıma örnek olarak yapılan değişiklikler sonucu labirentte belirlenen yeri öğrenme ve bulma, yapılan tekrarlar ile daha hızlı hedefe ulaşma gibi işler verilebilir. Ayrıca yapılan projedeki mantıkla bir fabrikada belirli bir alanda bir noktadan başka noktaya taşıma gibi örnekler de verilebilir.

4.Öneriler

Projenin tasarlama aşamasında istenen amaca uygun donanımsal ve yazılımsal adımlar uygulanmadan önce ilgili problem parçalara ayrılarak ayrı ayrı değerlendirilmelidir. Donanımsal araç ve gereçler istenen amaca uygun seçilmeli ve kullanımları araştırılmalıdır. Uygun devre şeması kurulduktan sonra yazılım aşamasında yapılacak adımlar gerçekleştirilmeli ve anlaşılır, en optimize şekilde kod yazılmalıdır. Bu aşamaların gerçekleştirilmesi planlı yapıldığı takdirde gelecekte insanların işlerini kolaylaştırma adına belki de insanların hayatına daha fazla girecek olan robotların projemizde istenildiği şekilde yapılacak değişiklikler ile farklı amaçlarda kullanımı gerçekleştirilmiş olacaktır.

5. Kaynaklar

[1] Yađlı, O. , LABİRENT ÖĐRENEN ROBOT TASARIMI VE GERÇEKLEŐTİRİLMESİ, 5.Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), 13-15 Mayıs 2009, Karabük, Türkiye

[2] <https://tr.wikipedia.org/wiki/İTÜRO> 15 Ekim 2015.

[3] https://en.wikipedia.org/wiki/Maze_solving_algorithm 15 Ekim 2015.

[4]http://www.gorobotics.net/index.php?option=com_content&task=view&id=18&Itemid=46
8 Kasım 2015.

[5] <http://nevit.blogspot.com.tr/2014/01/arduino-hc-sr04-ultrasonic-sensor-ile.html>
10 Kasım 2015.

[6] <http://arduinoturkiye.com/arduino-mega-2560-nedir/> 10 Kasım 2015.

[7] https://en.wikipedia.org/wiki/Motor_driver_L293d 16 Kasım 2015.

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?)

Yapılmış olan bir projeye kendi tasarımımıza uygun değişiklikler yapılarak kendi projemiz oluşturulmuştur.

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

Projenin tasarım aşamasında ilgili labirentten çıkması için robotun uygun şekil ve boyutlarda olmasında, labirent duvarlarının en ve boylarının, iki duvar arasındaki mesafenin doğru bir şekilde seçilmesinde, robot üzerindeki sensör ve tekerlerin doğru yerleştirilmesinde gerekli matematiksel hesaplamalar ayrıca motorların doğru voltta çalışması gerçekleştirilmiştir.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Algoritmalar, c++ programlama, elektrik ve elektronik devreler, mikroişlemciler, matematiksel hesaplamalar gibi derslerdeki bilgi ve beceriler kullanılmıştır.

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

USB 2.0, ICSP (SPI),Arduinio mega, Arduinio ide

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

- a) Ekonomi

Kullanılan araç ve gereçlerin minimum maliyet ile elde edilmesidir.

- b) Çevre sorunları:

Kullanılan labirentin yollarının(iki duvar arası mesafe) standart olması ve robotun gideceği zeminin uygun olmasıdır.

c) Sürdürülebilirlik:

Projenin çözümlenebilmesi, değiştirilebilmesi, sınanabilmesidir.

d) Üretilebilirlik:

Projenin algoritmik ve donanımsal açıdan gerçekleştirilecek şart ve amaca uygun üretimi yapılabilir.

e) Etik:

Projemizde toplum yararı dışında etik açıdan hiçbir sorun yoktur.

f) Sağlık:

Projemizde herhangi bir sağlık sorununa yol açan bir durum yoktur.

g) Güvenlik:

Projemizin güvenlik açısından tehlike oluşturabilecek bir zararı yoktur.

h) Sosyal ve politik sorunlar:

Projemiz farklı alanlarda kullanılarak sosyal veya politik faydalar elde edilebilir ama saptanan herhangi bir sosyal ve politik sorunu bulunmamaktadır.