

O-bOt



ÖRNEK UYGULAMALAR

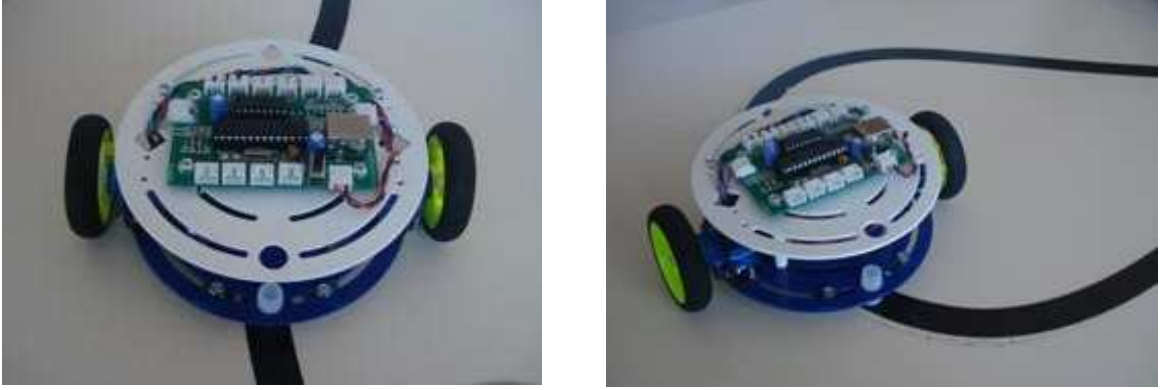
İÇİNDEKİLER

1	ÇİZGİ İZLEYEN ROBOT	3
1.1	Genel Bilgiler	3
1.2	O-bOt'un Çizgi İzleyen Robot Haline Getirilmesi	3
1.3	Algoritma ve Programlama	6
2	ENGELDEN KAÇINAN ROBOT	11
2.1	Genel Bilgiler	11
2.2	O-bOt'un Engelden Kaçınan Robot Haline Getirilmesi	11
2.3	Algoritma ve Programlama	13
2.3.1	Uzaklık Algılayıcısı Okuma Deneyi	14
3	DUVAR TAKİP EDEN ROBOT	20
3.1	Genel Bilgiler	20
3.2	O-bOt'un Duvar Takip Eden Robot Haline Getirilmesi	20
3.3	Algoritma ve Programlama	22
4	İŞIK TAKİP EDEN ROBOT	25
4.1	Genel Bilgiler	25
4.2	O-bOt'un Işık Takip Eden Robot Haline Getirilmesi	25
4.3	Algoritma ve Programlama	27

1 ÇİZGİ İZLEYEN ROBOT

1.1 Genel Bilgiler

Çizgi izleyen robotlar, beyaz zemin üzerine çizilen siyah veya siyah zemin üzerine çizilen beyaz renkteki çizgiyi CNY70, QRD1114 gibi optik algılayıcılar yardımıyla görüp takip edebilen robotlardır. Bu robotlar, eğitim ve hobi amaçlı kullanılabilmesi gibi endüstride malzeme taşınması gibi birçok alanda kullanılmaya elverişlidir. Çizgi izleyen robotlar çizilen çizgiden yardım alarak (çizgiyi takip ederek) verilen iki nokta arasında hareket etmeyi amaçlarlar.



Şekil 1 Çizgi izleyen O-bOt.

1.2 O-bOt'un Çizgi İzleyen Robot Haline Getirilmesi

Elimizde mevcut olan bir O-bOt'u çizgi izleyen robot haline getirebilmek için tek yapmamız gereken iki adet RS-AK01/01-CNY Optik Algılayıcı Kiti'ni robotumuza monte etmek.



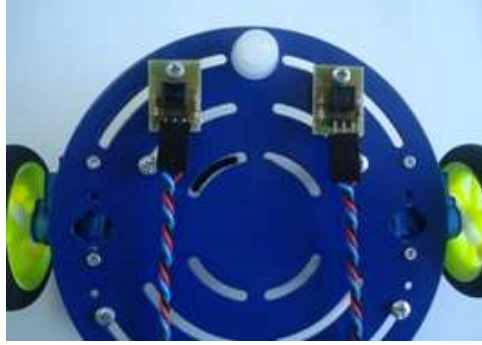
Şekil 2 O-bOt ve optik algılayıcı kitleri (CNY70).

1. Bunun için öncelikle kitle beraber gelen vida ve somunu resimdeki gibi birleştiriyoruz. Bu işlem, algılayıcının yere daha yakın olmasını sağlar:



Şekil 3 Çizgi algılayıcıya vida ve somunun bağlantısı.

2. Daha sonra bağlantısı yapılmış olan kitleri resimdeki gibi O-bOt şasesi üzerinde yerlerini belirliyoruz:

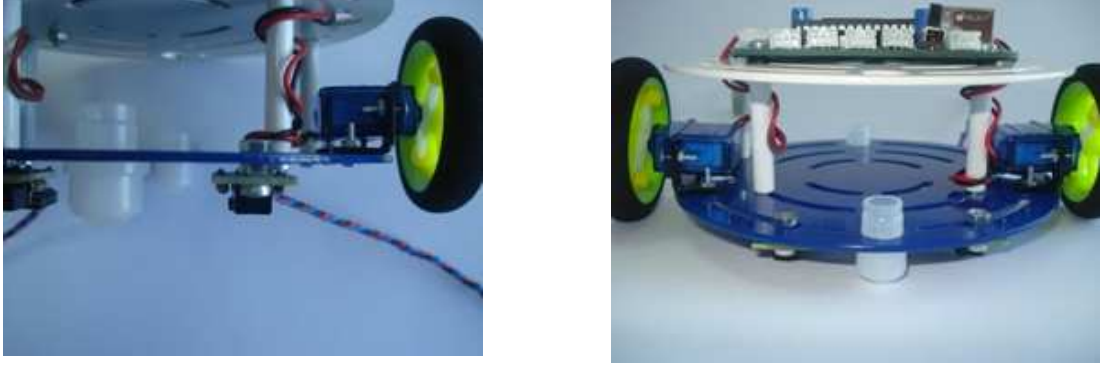


Şekil 4 Kitlerin O-bOt şasesindeki yerlerinin belirlenmesi.

ÖNEMLİ NOT

Algılayıcı kitleleri arasındaki uzaklık, çizgi izleyen robotlar için en önemli unsurlardan biridir. Bu uzaklığın takip edilen çizginin kalınlığına göre ayarlanması gerekir. O-bOt şasesi, üzerindeki açıklıklar sayesinde sizlere bu mesafeyi ayarlayabilme imkanı sunmaktadır. Böylelikle izleyeceğiniz çizginin kalınlığına göre algılayıcılar arasındaki mesafeyi dilediğiniz gibi ayarlayabilirsiniz.

Algılayıcılar arasındaki mesafeyi ayarladıktan sonra, bir somunla yerlerine sabitleyoruz.

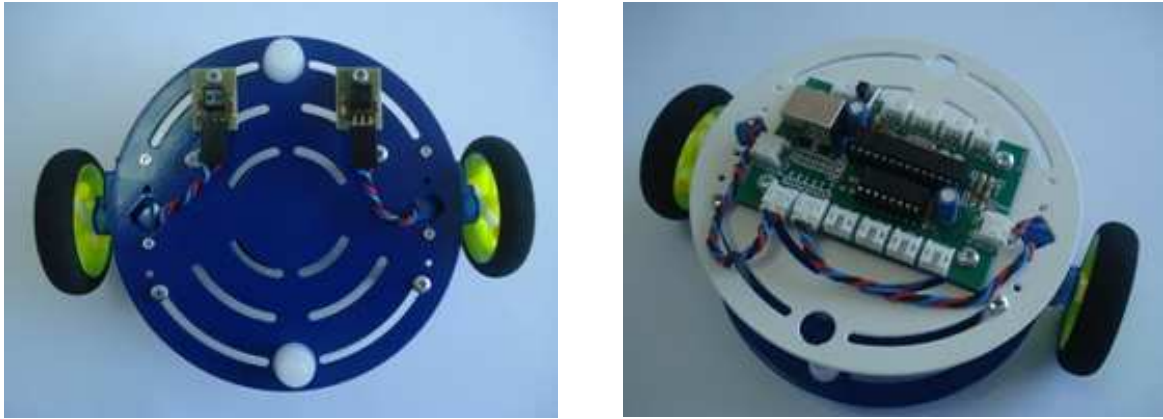


Şekil 5 Algılayıcıların O-bOt şasesine sabitlenmesi.

3. Son olarak, algılayıcıların kablolarını şasedeki deliklerden geçirerek O-bOt Kontrol Kartı ile bağlantılarını yapıyoruz. CNY70 sayısal çıkış veren bir algılayıcı olduğu için kontrol kartının sayısal giriş/çıkış pinlerine bağlanması gerekir. Bu programda “P5” ve “P6” pinleri tercih edilmiştir. Dilerseniz, program içerisinde gerekli değişiklikler yapmak koşulu ile bu bağlantıları başka sayısal giriş/çıkış pinlerine yapabilirsiniz.

ÖNEMLİ NOT

O-bOt Kontrol Kartı ve sayısal giriş/çıkışların kullanımını ile ilgili detaylı bilgiye “O-bOt ile Mekatronik Yapı Taşları” isimli dokümandan ulaşabilirsiniz.



Şekil 6 Algılayıcıların kablo bağlantılarının yapılması.

Artık O-bOt’unuz çizgi takip etmek için hazır. Sizde kendi programınızı geliştirebilir, daha yetenekli robotlar ortaya koyabilirsiniz!



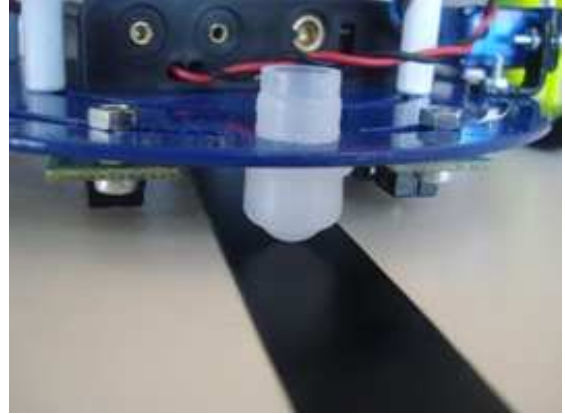
Şekil 7 Çizgi izleme yeteneğine sahip O-bOt.

1.3 Algoritma ve Programlama

Bu bölümde robotumuzu çizgi izleyerek bir noktadan diğerine götürecek algoritmayı oluşturacak ve iDea'da yazılıma dönüştüreceğiz. Yazılımımızın çalışmasını robotumuz üzerinde deneyerek göreceğiz.

Burada verilen örnek çizgi izleyen robot programı, beyaz zemin üzerinde siyah çizgiyi takip etmek üzere hazırlanmıştır. Ancak derseniz, programda yapacağınız değişikliklerle siyah zeminde beyaz çizgiyi de takip ettirebilirsiniz.

Çizgi algılayan robot algoritmasında üç temel durum vardır. Bunlardan birincisi, robotun çizginin üzerinde düz şekilde durmasıdır. Bu durumda, robotun altındaki her iki çizgi algılayıcı beyaz zemini görür (çizgiyi gören herhangi bir algılayıcı yoktur) ve robot çizginin üzerinde olduğu için düz gitmesi gerekir.

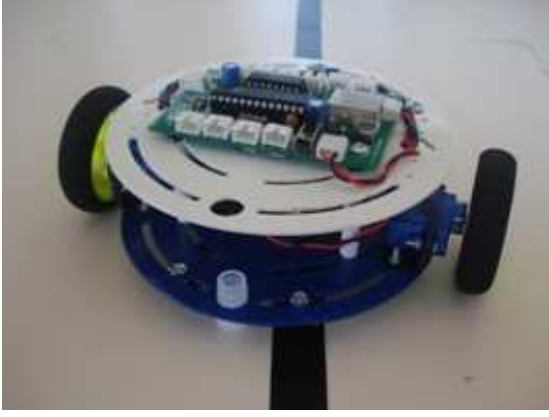


Şekil 8 Çizgi izleyen robotun düz durumu: çizginin sağ ve solunda yer alan algılayıcılar çizgiyi algılamazlar.

İkinci durum ise, robotun çizginin sağına kaymasıdır. Bu durumda, sol algılayıcı siyah çizginin üzerinde, sağ algılayıcı ise beyaz zeminin üzerindedir. Bu durumda robotun sola dönmesi gerekir: bunun için sol motor (motor-a) geri, sağ motor (motor-b) ileri yönde harekete geçirilmelidir.

ÖNEMLİ NOT

Motor-a ile aynı yöndeki algılayıcının sol, Motor-b ile aynı yöndeki algılayıcının ise sağ algılayıcı olduğunu unutmayınız!



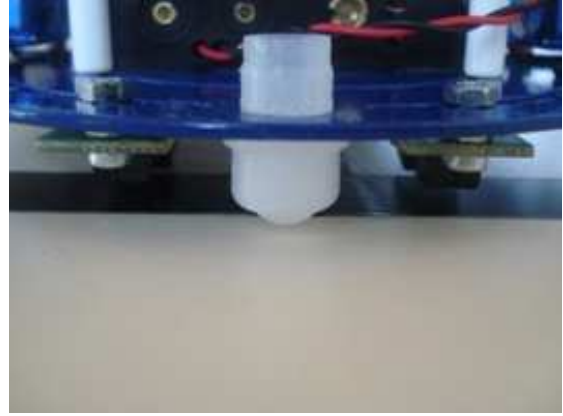
Şekil 9 Çizgi izleyen robotun çizginin sağına kaymış durumu: sol taraftaki algılayıcı çizginin üzerindedir.

Üçüncü durum ise, robotun çizginin soluna kaymasıdır. Bu durumda, sağ algılayıcı siyah çizginin üzerinde, sol algılayıcı ise beyaz zemin üzerindedir. Bu koşulda robotun sağa dönmesi gerekir: bunun için Motor-a ileri, Motor-b geri yönde harekete geçirilmelidir.



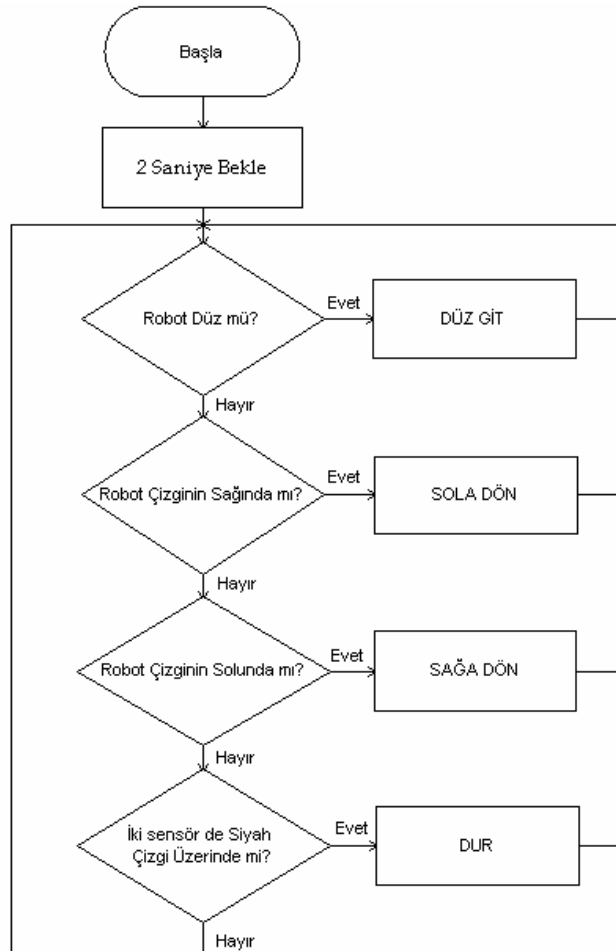
Şekil 10 Çizgi izleyen robotun çizginin soluna kaymış durumu: sağ taraftaki algılayıcı çizginin üzerindedir.

Çizgiyi takip ederek, gitmek istediğimiz noktaya geldiğimizi göstermek için bir işaret daha ekleyerek parkurun sonunda, normal çizgiye dik olarak çizilecek bir çizgiyle robotumuzu durdurmak mümkündür. Robot bu çizgi üzerine geldiğinde, her iki algılayıcı siyah rengi görerek programımıza ekleyeceğimiz komutlar ile durdurulabilir.



Şekil 11 Çizgi izleyen robotun çizginin sonunda durdurulması: sol ve sağ taraftaki algılayıcılar siyah çizgi üzerindedir.

Programlamaya geçmeden önce, söylediğimiz tüm bu durumları akış şeması yöntemi ile çizmek, programı yazmamız aşamasında çok yardımcı olacaktır.



Şekil 12 Çizgi izleyen robotun algoritması.

Çizgi izleyen robot algoritmasına göre yazılmış rsBasic programına bir örnek aşağıda gösterilmiştir;

```
pause 2000 'Robot harekete geçmeden önce 2 saniye bekle.

sensor: 'Çizgi algılayıcıları kontrol et
  if pin2=0 and pin3=0 then ileri 'Sağ ve sol algılayıcılar beyaz zeminde
mi? Öyleyse 'ileri' programına git.
  if pin2=1 and pin3=0 then soladon 'Sol algılayıcı siyah, sağ algılayıcı
beyaz zeminde mi? Eğer Öyleyse 'soladon' programına git.
  if pin2=0 and pin3=1 then sagadon 'Sağ algılayıcı siyah, sol algılayıcı
beyaz zeminde mi? Eğer Öyleyse 'sagadon' programına git.
  if pin2=1 and pin3=1 then dur 'Her iki algılayıcı siyah zeminde mi? Eğer
Öyleyse 'dur' programına git.
  goto sensor 'Algılayıcıları tekrar kontrol et

ileri: 'Her iki algılayıcı beyaz zeminde ise robot çizgiden çıkmamıştır.
Bu yüzden düz gitmesi gerekir.
forward a 'Motor a'yı ileri yönde harekete geçir.
forward b 'Motor b'yi ileri yönde harekete geçir.
pause 10 '10 ms bekle
goto sensor 'Yeni durum için algılayıcıları kontrol et.

soladon: 'Sol algılayıcı çizginin üzerinde, sağ algılayıcı beyaz zeminde
ise robot çizginin sağındadır. Sola dönmesi gerekir.
backward a 'Motor a'yı geri yönde harekete geçir.
forward b 'motor b'yi ileri yönde harekete geçir.
pause 10 '10 ms bekle
goto sensor 'Yeni durum için algılayıcıları kontrol et.

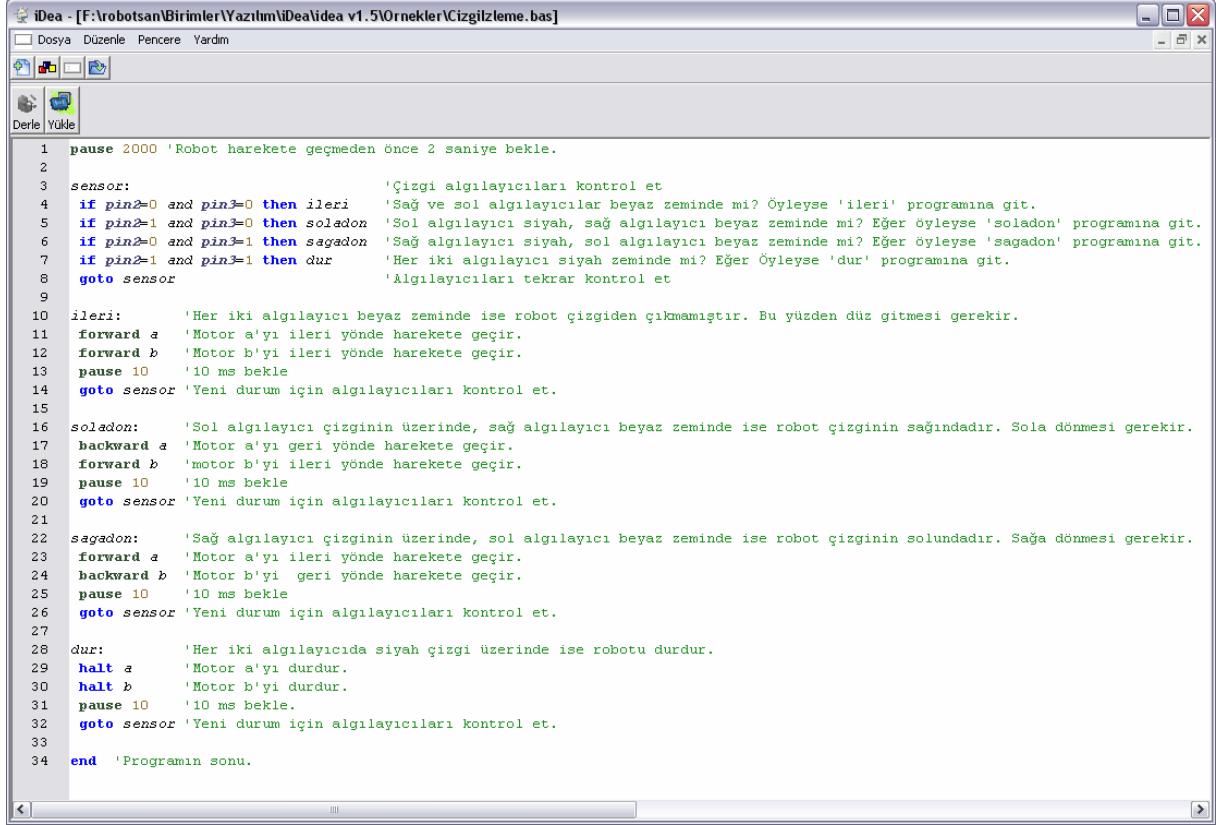
sagadon: 'Sağ algılayıcı çizginin üzerinde, sol algılayıcı beyaz zeminde
ise robot çizginin solundadır. Sağa dönmesi gerekir.
forward a 'Motor a'yı ileri yönde harekete geçir.
backward b 'Motor b'yi geri yönde harekete geçir.
pause 10 '10 ms bekle
goto sensor 'Yeni durum için algılayıcıları kontrol et.

dur: 'Her iki algılayıcıda siyah çizgi üzerinde ise robotu durdur.
halt a 'Motor a'yı durdur.
halt b 'Motor b'yi durdur.
pause 10 '10 ms bekle.
goto sensor 'Yeni durum için algılayıcıları kontrol et.

end 'Programın sonu.
```

Şekil 13 Çizgi İzleyen Robot Programı.

Programı iDea ile yazdıktan ve O-bOt ile USB bağlantısını yaptıktan sonra “YÜKLE” simgesine tıklayarak programı O-bOt’a yükleyebilirsiniz. O-bOt artık çizgi izlemek için hazır!



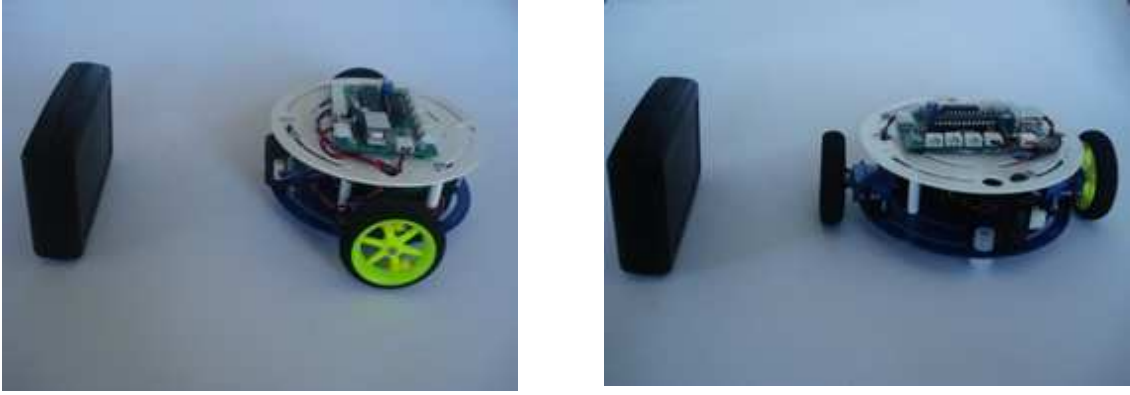
```
1 pause 2000 'Robot harekete geçmeden önce 2 saniye bekle.
2
3 sensor: 'Çizgi algılayıcıları kontrol et
4 if pin2=0 and pin3=0 then ileri 'Sağ ve sol algılayıcılar beyaz zeminde mi? Öyleyse 'ileri' programına git.
5 if pin2=1 and pin3=0 then soladon 'Sol algılayıcı siyah, sağ algılayıcı beyaz zeminde mi? Eğer öyleyse 'soladon' programına git.
6 if pin2=0 and pin3=1 then sagadon 'Sağ algılayıcı siyah, sol algılayıcı beyaz zeminde mi? Eğer öyleyse 'sagadon' programına git.
7 if pin2=1 and pin3=1 then dur 'Her iki algılayıcı siyah zeminde mi? Eğer Öyleyse 'dur' programına git.
8 goto sensor 'Algılayıcıları tekrar kontrol et
9
10 ileri: 'Her iki algılayıcı beyaz zeminde ise robot çizgiden çıkmamıştır. Bu yüzden düz gitmesi gerekir.
11 forward a 'Motor a'yı ileri yönde harekete geçir.
12 forward b 'Motor b'yi ileri yönde harekete geçir.
13 pause 10 '10 ms bekle
14 goto sensor 'Yeni durum için algılayıcıları kontrol et.
15
16 soladon: 'Sol algılayıcı çizginin üzerinde, sağ algılayıcı beyaz zeminde ise robot çizginin sağındadır. Sola dönmesi gerekir.
17 backward a 'Motor a'yı geri yönde harekete geçir.
18 forward b 'motor b'yi ileri yönde harekete geçir.
19 pause 10 '10 ms bekle
20 goto sensor 'Yeni durum için algılayıcıları kontrol et.
21
22 sagadon: 'Sağ algılayıcı çizginin üzerinde, sol algılayıcı beyaz zeminde ise robot çizginin solundadır. Sağa dönmesi gerekir.
23 forward a 'Motor a'yı ileri yönde harekete geçir.
24 backward b 'Motor b'yi geri yönde harekete geçir.
25 pause 10 '10 ms bekle
26 goto sensor 'Yeni durum için algılayıcıları kontrol et.
27
28 dur: 'Her iki algılayıcıda siyah çizgi üzerinde ise robotu durdur.
29 halt a 'Motor a'yı durdur.
30 halt b 'Motor b'yi durdur.
31 pause 10 '10 ms bekle.
32 goto sensor 'Yeni durum için algılayıcıları kontrol et.
33
34 end 'Programın sonu.
```

Şekil 14 Programın iDea ile yazılması.

2 ENGELDEN KAÇINAN ROBOT

2.1 Genel Bilgiler

Doğada hemen hemen her canlıda görülen engellerden sakınma işlemi, robotlarda da en çok kullanılan ve uygulaması bulunan özelliklerden biridir. Engeli tespit etme işi yine algılayıcılar (Sharp mesafe algılayıcısı gibi) aracılığıyla yapılır. Engelden kaçınma işlemi diğer programlara da dahil edilebilir. Örneğin; çizgi izleyen bir robotun önüne herhangi bir engel çıkması halinde durması işlemi çizgi izleyen robot programına engelden kaçınan robot programının eklenmesi ile oluşturulabilir. Engelden kaçınan robotların genel amacı, belirlenen bir alan içerisinde hiçbir yere çarpmadan hareket etmektir.



Şekil 1 O-bOt'un önündeki engelleri algılaması ve engellerden kaçması: önündeki engeli görünce düz gidiyorken engelden kaçınacak şekilde yön değiştirmesi.

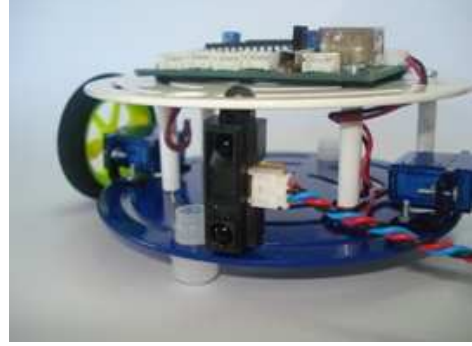
2.2 O-bOt'un Engelden Kaçınan Robot Haline Getirilmesi

Elimizdeki bir O-bOt'u engelden sakınan robot haline dönüştürebilmemiz için, O-bOt'a bir adet SHARP GP2Y0A21 uzaklık algılayıcısı eklememiz yeterli olacak.



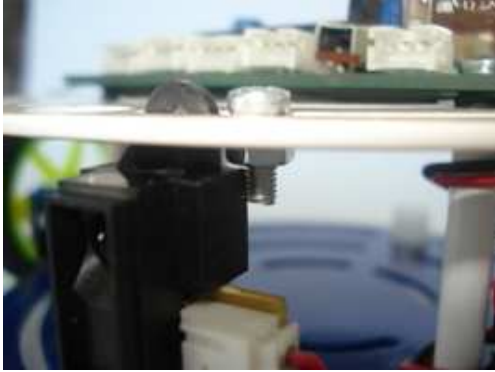
Şekil 2 O-bOt ve uzaklık algılayıcı.

1. Öncelikle, algılayıcıyı O-bOt şasesi üzerindeki aralıklara şekildeki gibi dik konumda yerleştiriyoruz:



Şekil 3 Uzaklık algılayıcının O-bOt'a yerleştirilmesi.

2. Algılayıcıyı yerleştirdikten sonra, hareket esnasında yerinden oynamaması için resimdeki gibi iki adet vida ve somun ile sabitleyoruz:



Şekil 4 Algılayıcıların sabitlemesi.

3. Algılayıcıyı sabitledikten sonra artık tek yapmamız gereken, algılayıcının kablosunu şase üzerindeki deliklerden geçirerek O-bOt Kontrol Kartına bağlantısını yapmak. Uzaklık algılayıcısı, analog çıkış verdiği için O-bOt Kontrol Kartı'nın analog girişlerinden birine bağlanması gerekir. Bu programda, "P11" pini tercih edilmiştir. Ancak derseniz programda gerekli değişiklikleri yaptıktan sonra istediğiniz bir başka analog girişe bağlayabilirsiniz.

ÖNEMLİ NOT

O-bOt Kontrol Kartı ve Analog Girişlerin Kullanımı ile ilgili detaylı bilgiye "Mekatronik Yapı Taşları" isimli dokümandan ulaşabilirsiniz.



Şekil 5 Algılayıcının O-bOt kontrol kartına bağlantısı.

Ve robotunuz artık engelden kaçınmak için hazır. Sizde bu programı geliştirebilir veya kendi programınızı yazarak daha yetenekli robotlar yapabilirsiniz!



Şekil 6 Uzaklık algılayıcı takılmış O-bOt.

2.3 Algoritma ve Programlama

Burada verilen program, bir yönde düz olarak ilerleyen robotun önüne bir engel çıkması halinde 90 derece sola dönerek yönünü değiştirmesi işlemi yapmaktadır. Robot, yönünü değiştirdikten sonra hareketine kaldığı yerden devam eder.

Engelden kaçınan robotun temel işlevi, önünde engel olup olmadığını kontrol etmek ve elde edilen bilgiler doğrultusunda yapacağı harekete karar vermektir. Önünde engel yok ise ileri yönde düz olarak hareket eder. Eğer önünde engel varsa Motor-a geri, Motor-b ileri yönde hareket ederek robotu yaklaşık olarak 90 derece sola döndürür. Sonrasında eğer engel yok ise robot ileri yöndeki hareketine devam eder. Robot yine bir engelle karşılaşır, engelden kurtuluncaya kadar sola dönmeye devam eder.

Buradaki en önemli nokta, robotun engele ne kadar yaklaşacağıdır. Yani, robot engele kaç santimetre yaklaştıktan sonra önündekinin bir engel olduğunu anlayacaktır. Bu değerler program içerisinde belirtilmiştir ancak herhangi bir uzunluk biriminde değildir. Bu değerlerin nasıl elde edildiğini anlamak için uzaklık algılayıcısı okuma deneyinin uygulanması gerekir.

2.3.1 Uzaklık Algılayıcısı Okuma Deneyi

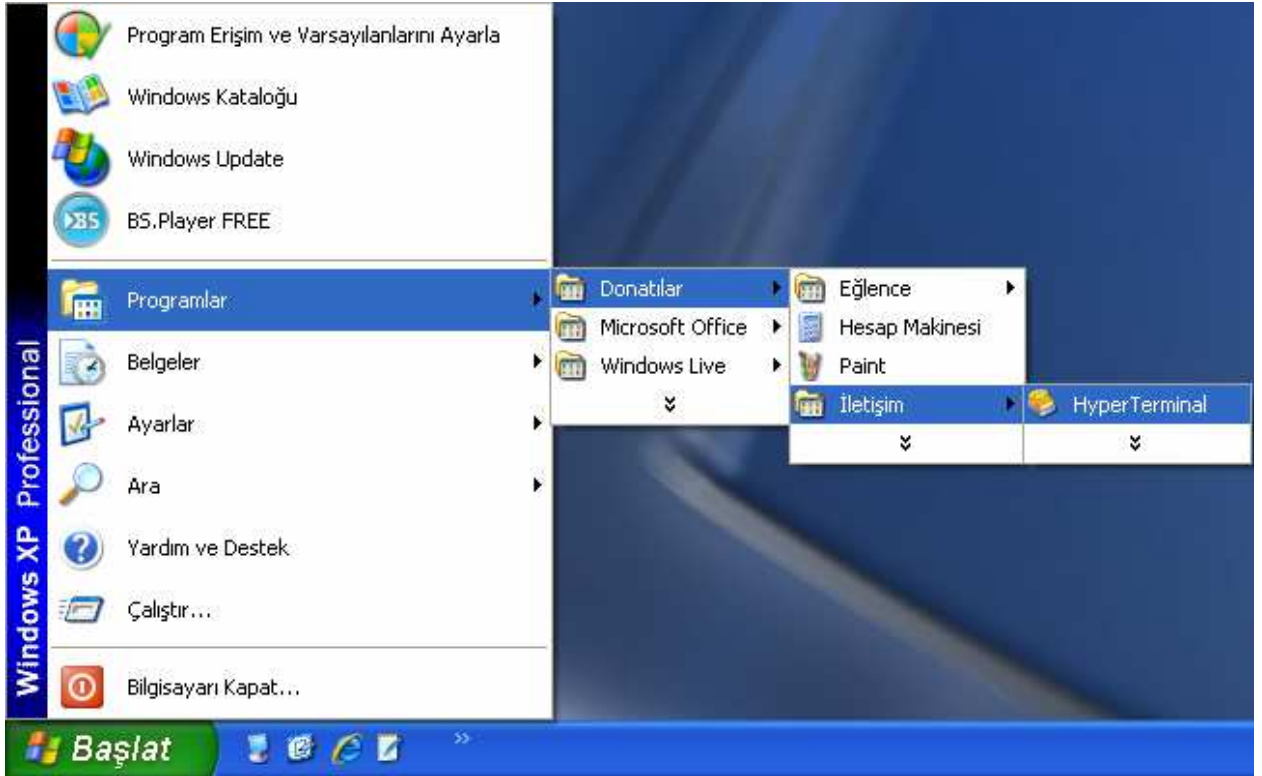
Bu deney uzaklık algılayıcıların nasıl çalıştığını ve nasıl değerler ürettiğini anlamak için yapılmıştır. Uzaklık algılayıcısı değerleri “debug” komutu ile bilgisayara aktarılmakta ve değerlerini görebilmemiz sağlanmaktadır. Bu deneyin aşamaları aşağıda yer almaktadır:

1. Algılayıcıyı, O-bOt Kontrol Kartının analog girişlerinden birincisine (P11) bağlandıktan sonra aşağıdaki programı yüklüyoruz:

```
symbol mesafe=w0 'mesafe değişkeni tanımlandı,  
kontrol: 'kontrol döngüsü,  
readadc 0,mesafe 'mesafeden bilgileri oku,  
pause 1 'değerlerin okunması için gerekli geçikme,  
debug 'bilgileri bilgisayara aktar,  
pause 500 'artarma işlemini her 500ms'de bir yap,  
goto kontrol 'algılayıcıyı tekrar kontrol et
```

2. Programı yükledikten sonra, O-bOt ile USB bağlantısını kesmeden, bilgisayarınızdaki “HYPER TERMINAL” programı ile algılayıcının ürettiği değerleri görmek artık mümkün. Bunun için öncelikle Hyper Terminal programını başlatıyoruz;

BAŞLAT>PROGRAMLAR>DONATILAR>İLETİŞİM>HYPER TERMINAL



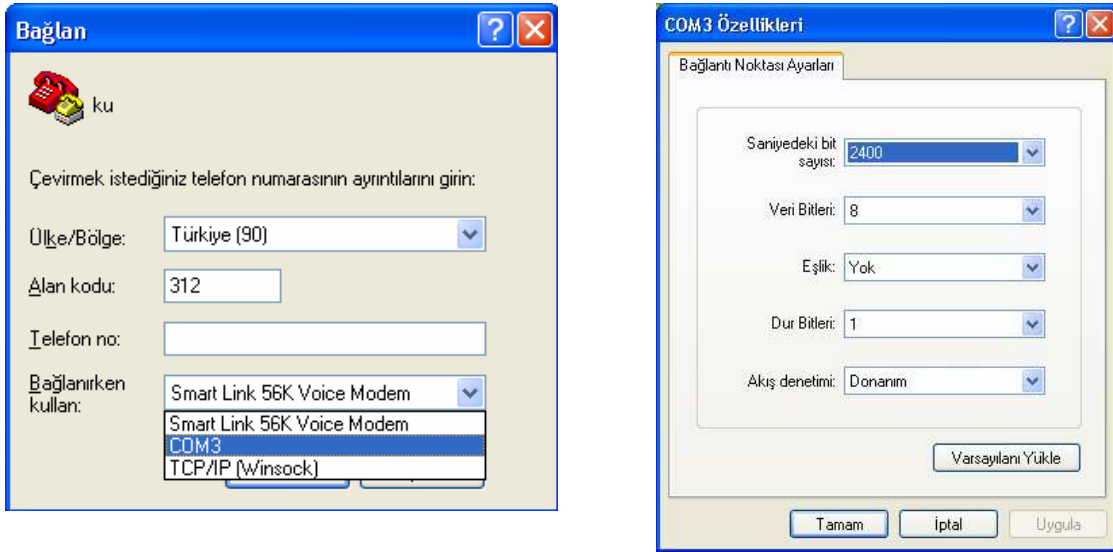
Şekil 7 Programın başlatılması.

3. Program başlatıldıktan sonra gerekli ayarlamalar yapılmalıdır. Şekil 8’deki gibi açılan pencere ile dokümanın adını belirliyoruz.



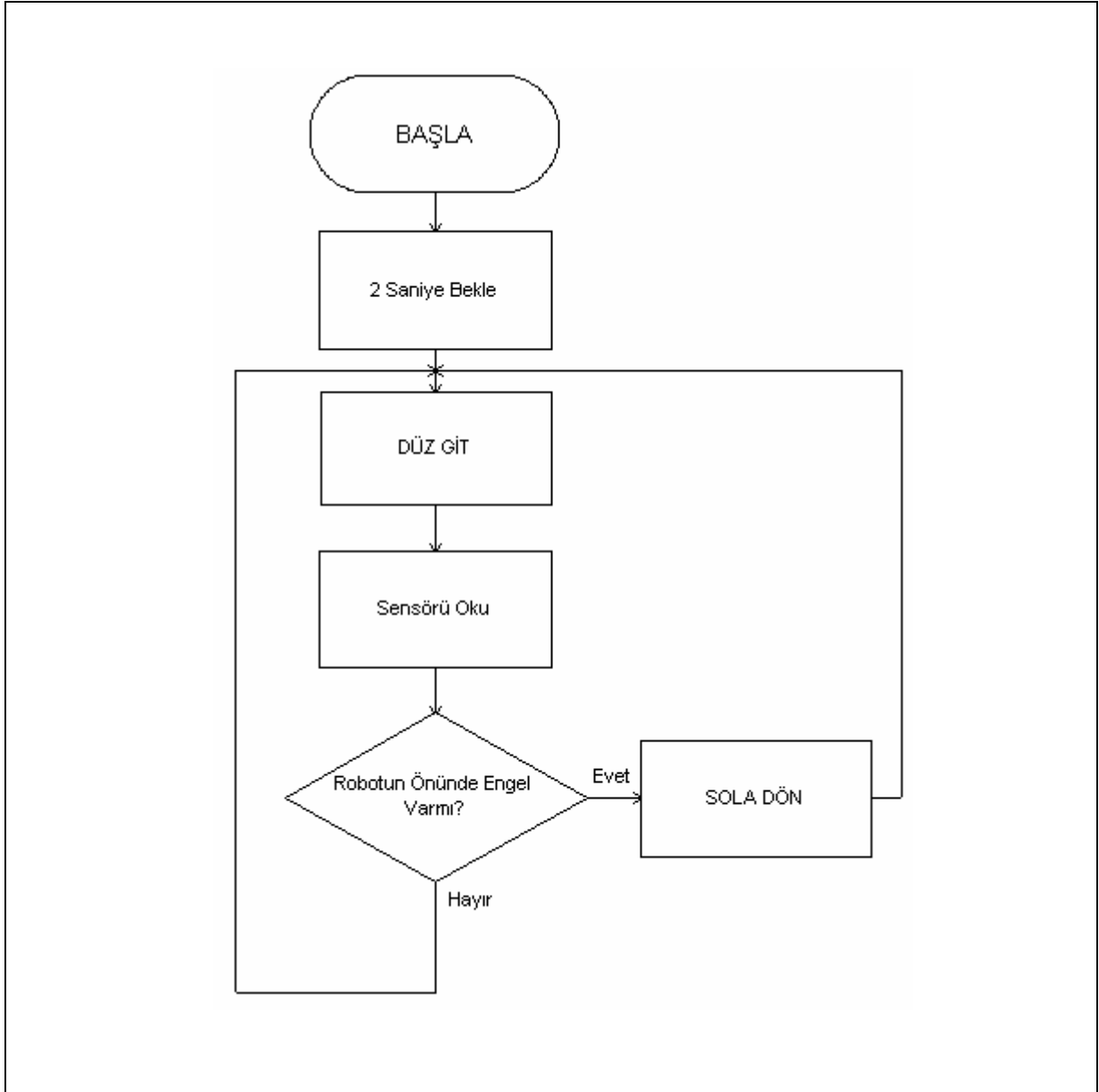
Şekil 8 Dokümana isim verilmesi.

4. Daha sonra açılan pencerede, O-bOt Kontrol Kartının hangi port'a bağlı olduğunu bildirmemiz gereklidir. Bu pencereden sonra açılan yeni pencerede iletişim özellikleri görülmektedir. "Tamam" butonunu tıklayarak ayarlama işlemlerini bitiriyoruz.



Şekil 9 Port belirleme işlemleri.

Bu basamaklardan sonra, ekranda sürekli olarak bazı değerler görünecektir. Bu değerler mesafe algılayıcısından alınan bilgilerin işlemci tarafından dönüştürülmesiyle elde edilen sayılardır. Yani bu sayılar doğrudan algılayıcı tarafından üretilmez. Algılayıcı, önündeki engelin uzaklığına göre çıkışından 0 ile 5 volt arasında değişik gerilim değerleri üretir. İşlemci ise analog girişlerine uygulanan bu farklı değerlerdeki voltaj seviyelerine göre 0 ile 1023 arasında sayılar üretir, bu sayılar programda kullanılarak mesafe değerleri belirlenir.



Şekil 11 Engelden kaçınma algoritması - Akış Şeması.

Engelden kaçınan robot programı;

```
pause 2000 '2 saniye bekle

symbol mesafe=w0 'mesafe değişkeni tanımlandı

goto duzgit 'robot öncelikle düz gidecek, 'duzgit' programına git

kontrol: 'mesafe algılayıcı kontrol programı
  readadc 0,mesafe 'Mesafe algılayıcıyı oku ve değeri "mesafe" isimli
  değişkenin içerisine yaz
  pause 1 'değerin okunabilmesi için gerekli süreyi bekle

  if mesafe>512 then 'Eğer engele yaklaşık 12cm kadar yaklaşılmışsa;
  goto soladon 'Robotu sola döndür
  else
    'Eğer engel yoksa veya uzaktaysa
  goto duzgit 'Robotu düz hareket ettir
  endif
  'if bloğunun sonu
  goto kontrol 'Algılayıcıyı tekrar kontrol et

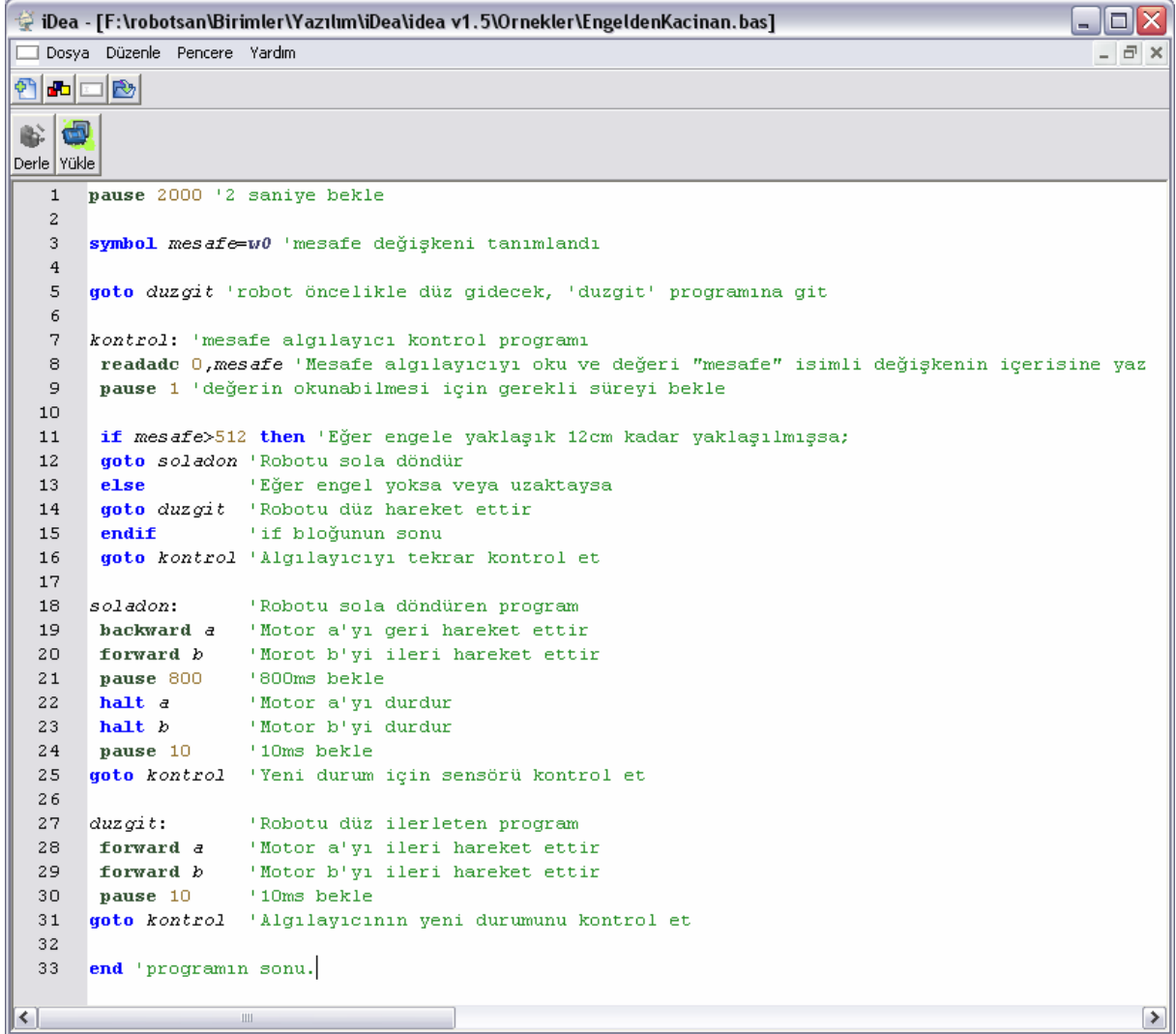
soladon: 'Robotu sola döndüren program
  backward a 'Motor a'yı geri hareket ettir
  forward b 'Motor b'yi ileri hareket ettir
  pause 800 '800ms bekle
  halt a 'Motor a'yı durdur
  halt b 'Motor b'yi durdur
  pause 10 '10ms bekle
  goto kontrol 'Yeni durum için sensörü kontrol et

duzgit: 'Robotu düz ilerleten program
  forward a 'Motor a'yı ileri hareket ettir
  forward b 'Motor b'yı ileri hareket ettir
  pause 10 '10ms bekle
  goto kontrol 'Algılayıcının yeni durumunu kontrol et

end 'programın sonu.
```

Şekil 12 Engelden Kaçınan Robot Programı.

Programı iDea ile yazdıktan ve O-bOt ile USB bağlantısını yaptıktan sonra “YÜKLE” simgesine tıklayarak program O-bOt’a yükleyebilirsiniz. O-bOt artık engelden kaçınmak için hazır!



```
iDea - [F:\robotan\Birimler\Yazılım\iDea\idea v1.5\Ornekler\EngeldenKacinan.bas]
Dosya Düzenle Pencere Yardım
Derle Yükle
1  pause 2000 '2 saniye bekle
2
3  symbol mesafe=w0 'mesafe değişkeni tanımlandı
4
5  goto duzgit 'robot öncelikle düz gidecek, 'duzgit' programına git
6
7  kontrol: 'mesafe algılayıcı kontrol programı
8  readadc 0,mesafe 'Mesafe algılayıcıyı oku ve değeri "mesafe" isimli değişkenin içerisine yaz
9  pause 1 'değerin okunabilmesi için gerekli süreyi bekle
10
11  if mesafe>512 then 'Eğer engelle yaklaşık 12cm kadar yaklaşmışsa;
12  goto soladon 'Robotu sola döndür
13  else 'Eğer engel yoksa veya uzaktaysa
14  goto duzgit 'Robotu düz hareket ettir
15  endif 'if bloğunun sonu
16  goto kontrol 'Algılayıcıyı tekrar kontrol et
17
18  soladon: 'Robotu sola döndüren program
19  backward a 'Motor a'yı geri hareket ettir
20  forward b 'Motor b'yi ileri hareket ettir
21  pause 800 '800ms bekle
22  halt a 'Motor a'yı durdur
23  halt b 'Motor b'yi durdur
24  pause 10 '10ms bekle
25  goto kontrol 'Yeni durum için sensörü kontrol et
26
27  duzgit: 'Robotu düz ilerleten program
28  forward a 'Motor a'yı ileri hareket ettir
29  forward b 'Motor b'yi ileri hareket ettir
30  pause 10 '10ms bekle
31  goto kontrol 'Algılayıcının yeni durumunu kontrol et
32
33  end 'programın sonu.
```

Şekil 13 Programın iDEA ile yazılması.

3 DUVAR TAKİP EDEN ROBOT

3.1 Genel Bilgiler

Duvar takip eden robotun amacı, sol tarafındaki (Algılayıcı sol tarafına takıldığı için) duvarı belli bir mesafeden takip etmektir. Algılayıcının yerini değiştirerek sağ tarafındaki duvarı da takip ettirmek mümkündür. Takip sırasında robotun duvardan aşırı derecede uzaklaşmaması veya aşırı derecede yaklaşmaması gerekir. Robotun, duvarı izlemesi istenilen mesafe değerleri programda belirtilmiştir. Bu değerleri değiştirerek mesafeyi ayarlamak mümkündür. Ayrıca bu programda ek olarak engelden kaçınma işlemi de dahil edilmiştir. Robot, önüne takılan algılayıcı aracılığıyla engelleri algılar. Eğer önünde engel varsa robot durur. Engel kaldırıldığında ise işlevine kaldığı yerden devam eder.



Şekil 1 O-bOt'un duvarı takip etmesi ve engellerden kaçınması.

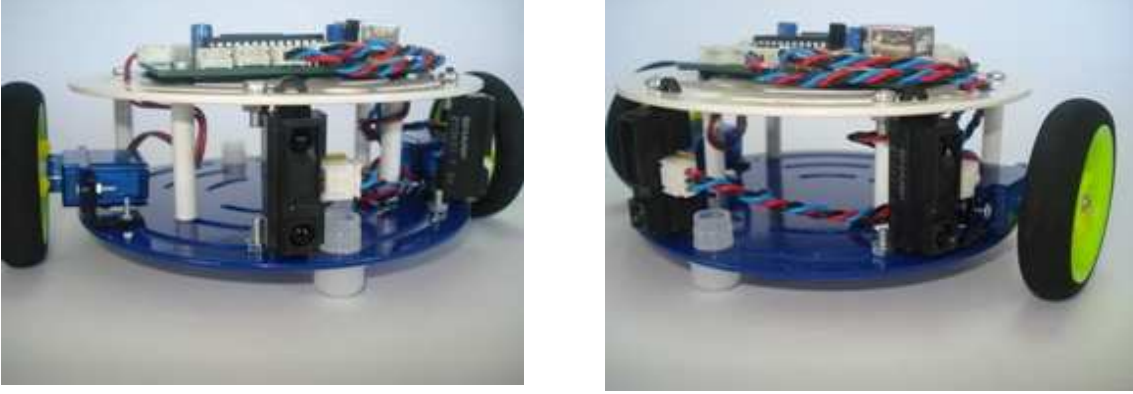
3.2 O-bOt'un Duvar Takip Eden Robot Haline Getirilmesi

Elimizdeki O-bOt'u duvar takip eden robot haline getirmek için sol tarafına bir SHARP GP2Y0A21 uzaklık algılayıcısı takmak yeterlidir ancak buradaki programa aynı zamanda engelden kaçınma özelliği eklendiği için, aynı algılayıcıdan robotun ön tarafına da takmak gerekmektedir.



Şekil 2 O-bOt ve uzaklık algılayıcısı.

1. O-bOt'u duvar takip eden robot haline getirme işlemi “Engelden Kaçınan Robot” ta yapılan işlemlerle tamamen aynıdır. Öncelikle uzaklık algılayıcılarının yerlerini belirliyoruz. Daha sonra somun ve vidalarla yerlerine sabitliyoruz. Yalnız burada yapılacak olan tek değişiklik; resimde görüldüğü gibi, öndeki algılayıcının, destek tekerleğinin diğer tarafına takılmasıdır.



Şekil 3 O-bOt üzerine uzaklık algılayıcılarının takılması.

2. Bu deneyde kullandığımız uzaklık algılayıcısı analog çıkış veren algılayıcı olduğu için O-bOt Kontrol Kartı'nın analog girişine bağlanmalıdır. Bu programda “P11” ve “P12” bağlantıları tercih edilmiştir. Ancak derseniz programda gerekli değişiklikleri yaptıktan sonra algılayıcıları istediğiniz bir başka analog girişe bağlayabilirsiniz.

ÖNEMLİ NOT

O-bOt Kontrol Kartı ve analog pinlerin kullanımı ile ilgili detaylı bilgiye “Mekatronik Yapı Taşları” isimli dokümandan ulaşabilirsiniz.

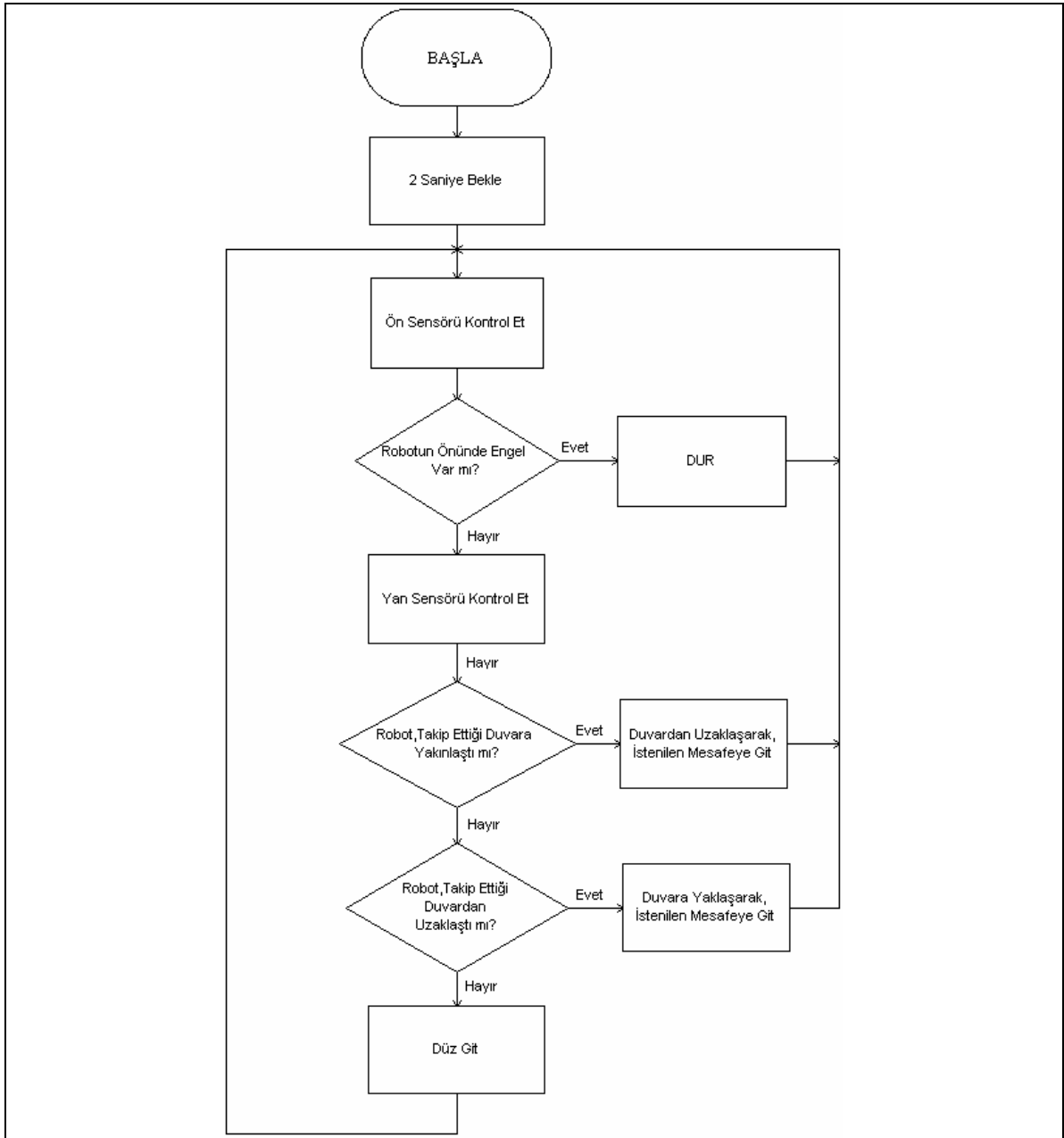
Bu işlemlerden sonra, artık duvar takip eden robotumuz hazır. Sizde bu programı geliştirebilir veya daha kapsamlı algoritmalar oluşturabilirsiniz.



Şekil 4 Duvar takip eden O-bOt.

3.3 Algoritma ve Programlama

Burada örnek olarak verilen algoritmada duvar takip eden robot, öncelikle önündeki algılayıcı vasıtasıyla herhangi bir engelle karşı karşıya olup olmadığını kontrol eder. Eğer engel varsa durur, yoksa duvarı takip etmeye başlar. Robot, önünde herhangi bir engel olmadığı bilgisini aldıktan sonra yan algılayıcıyı kontrol etmeye başlar ve belirtilen mesafe aralığında duvarı takip eder. Eğer robot duvara aşırı şekilde yaklaşmışsa Motor-a ileri, Motor-b geri yönde hareket ederek robotu duvardan uzaklaştırır. Robotun duvardan aşırı derecede uzaklaşması durumunda ise Motor-a geri, Motor-b ileri yönde çalışarak robotu duvara yaklaştırır. Takip hareketi esnasında bir engelle karşılaşırsa robot hemen durur, engel kaldırıldığında ise hareketine kaldığı yerden devam eder. Programın akış şeması aşağıdaki gibidir;



Şekil 5 Programın Akış Şeması.

Duvar Takip Eden Robot Programı;

```
pause 2000 'Robot hareket etmeden önce 2 saniye bekle

symbol yansensor=w0 '"yansensor" değişkeni tanımlandı (Yandaki uzaklık algılayıcı için)
symbol onsensor=w1 '"onsensor" değişkeni tanımlandı. (Öndeki uzaklık algılayıcı için)

anakontrol: 'Algılayıcıları kontrol et
  readadc 0,onsensor 'Öndeki algılayıcıdan bilgileri oku
  pause 1 'Okuma işleminin gerçekleşmesi için 1 ms bekle
  if onsensor>512 then 'Robotun önünde engel var mı? Varsa;
  goto dur ' "dur" programına git(Robotu durdur)
  else 'Yoksa;
  goto kontrol ' "kontrol" programına git(diğer algılayıcıyı kontrol etmek için)
  endif 'if bloğunu sonlandır

kontrol: 'Yan algılayıcının kontrol işlemi
  readadc 1,yansensor 'Yan algılayıcıdan bilgileri oku
  pause 1 'Okuma işleminin gerçekleşmesi için 1 ms bekle
  if yansensor<300 then soladon 'Eğer robot duvardan uzaklaşmışsa "soladon" programına git
  if yansensor>400 then sagadon 'Eğer robot duvara yaklaşmışsa "sagadon" programına git
  if yansensor>300 and yansensor<400 then duzgit 'Eğer robot duvara paralel ve istenen mesafedeyse "duzgit" programına git
  goto anakontrol 'Algılayıcıları tekrar kontrol et.

dur: 'Robotu durduran program,
  halt a 'Motor a'yı durdur
  halt b 'Motor b'yi durdur
  pause 10 '10 ms bekle
  goto anakontrol 'Yeni durum için algılayıcıları tekrar kontrol et.

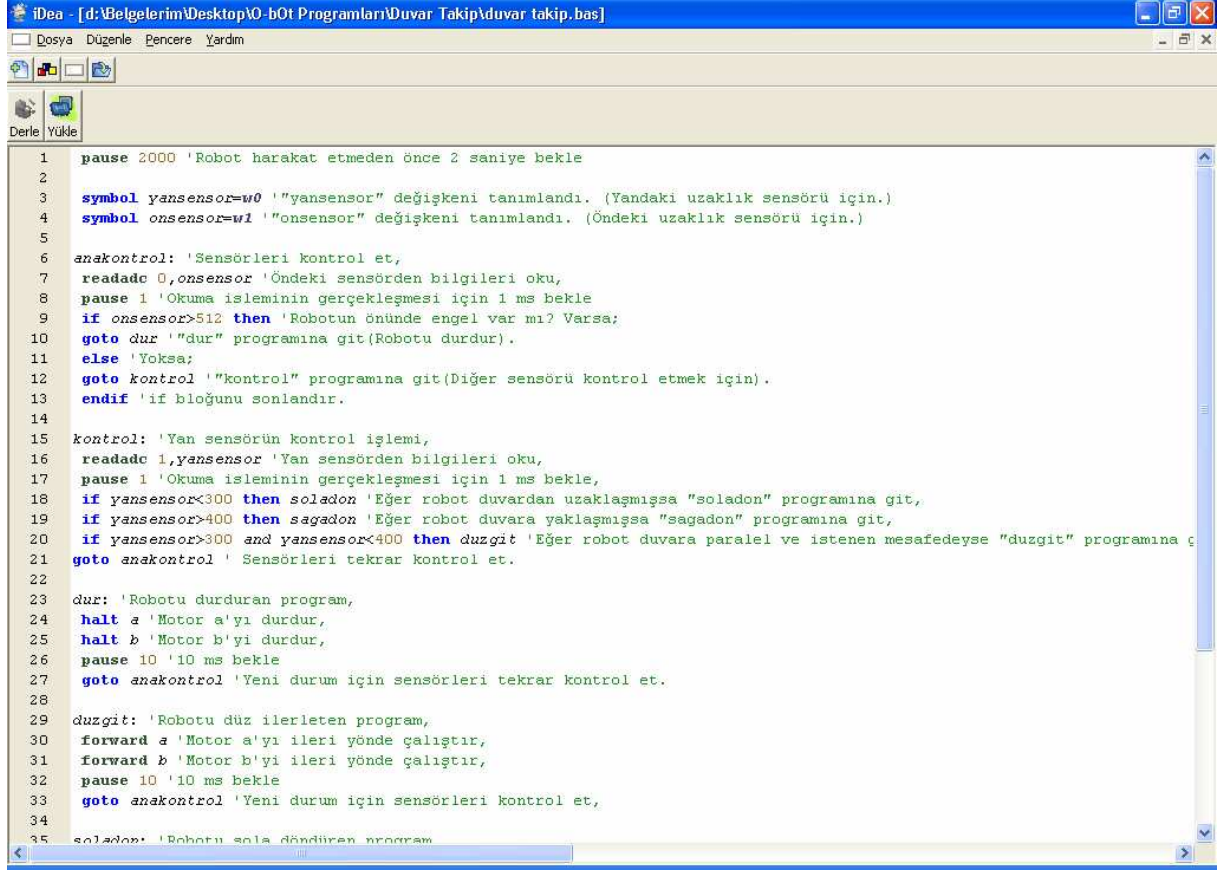
duzgit: 'Robotu düz ilerleten program,
  forward a 'Motor a'yı ileri yönde çalıştır
  forward b 'Motor b'yi ileri yönde çalıştır
  pause 10 '10 ms bekle
  goto anakontrol 'Yeni durum için algılayıcıları kontrol et

soladon: 'Robotu sola döndüren program
  backward a 'Motor a'yı geri yönde çalıştır
  forward b 'Motor b'yi ileriyönde çalıştır
  pause 10 '10 ms bekle
  goto anakontrol 'Yeni durum için algılayıcıları kontrol et

sagadon: 'Robotu sağa döndüren program
  forward a 'Motor a'yı ileri yönde çalıştır
  backward b 'Motor b'yi geri yönde çalıştır
  pause 10 '10 ms bekle
  goto anakontrol 'Yeni durum için algılayıcıları kontrol et

end 'programın sonu.
```

Programı iDea ile yazdıktan ve O-bOt ile USB bağlantısını yaptıktan sonra “YÜKLE” simgesine tıklayarak programı O-bOt’a yükleyebilirsiniz. O-bOt artık duvar takip etmek için hazır!



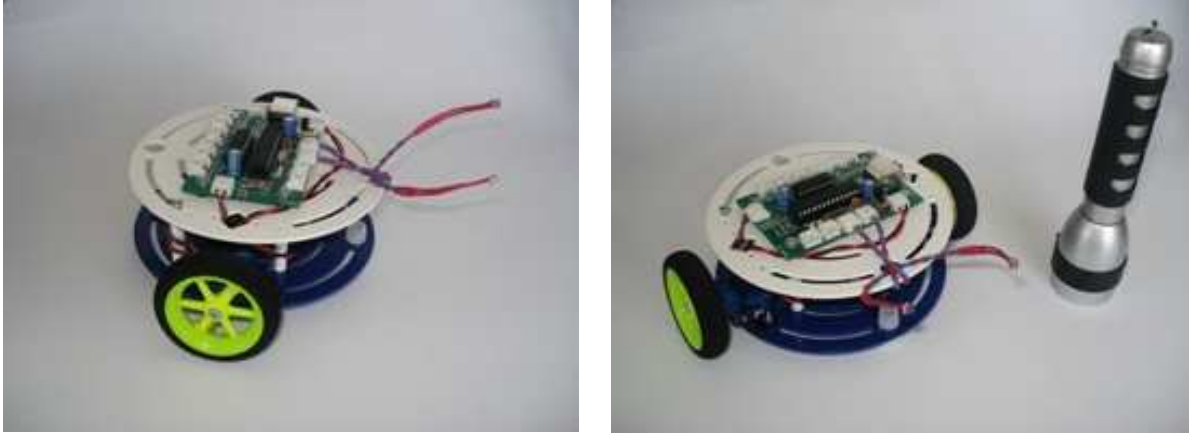
```
1 pause 2000 'Robot hareket etmeden önce 2 saniye bekle
2
3 symbol yansensor=w0 "yansensor" değişkeni tanımlandı. (Yandaki uzaklık sensörü için.)
4 symbol onsensor=w1 "onsensor" değişkeni tanımlandı. (Öndeki uzaklık sensörü için.)
5
6 anakontrol: 'Sensörleri kontrol et,
7 readadc 0,onsensor 'Öndeki sensörden bilgileri oku,
8 pause 1 'Okuma işleminin gerçekleşmesi için 1 ms bekle
9 if onsensor>512 then 'Robotun önünde engel var mı? Varsa;
10 goto dur "dur" programına git(Robotu durdur).
11 else 'Yoksa;
12 goto kontrol "'kontrol" programına git(Diğer sensörü kontrol etmek için).
13 endif 'if bloğunu sonlandır.
14
15 kontrol: 'Yan sensörün kontrol işlemi,
16 readadc 1,yansensor 'Yan sensörden bilgileri oku,
17 pause 1 'Okuma işleminin gerçekleşmesi için 1 ms bekle,
18 if yansensor<300 then soladon 'Eğer robot duvardan uzaklaşmışsa "soladon" programına git,
19 if yansensor>400 then sagadon 'Eğer robot duvara yaklaşmışsa "sagadon" programına git,
20 if yansensor>300 and yansensor<400 then duzgit 'Eğer robot duvara paralel ve istenen mesafedeyse "duzgit" programına git,
21 goto anakontrol ' Sensörleri tekrar kontrol et.
22
23 dur: 'Robotu durduran program,
24 halt a 'Motor a'yi durdur,
25 halt b 'Motor b'yi durdur,
26 pause 10 '10 ms bekle
27 goto anakontrol 'Yeni durum için sensörleri tekrar kontrol et.
28
29 duzgit: 'Robotu düz ilerleten program,
30 forward a 'Motor a'yi ileri yönde çalıştır,
31 forward b 'Motor b'yi ileri yönde çalıştır,
32 pause 10 '10 ms bekle
33 goto anakontrol 'Yeni durum için sensörleri kontrol et,
34
35 soladon: 'Robotu sola döndüren program
```

Şekil 6 Programın O-bOt ile yazılması.

4 IŞIK TAKİP EDEN ROBOT

4.1 Genel Bilgiler

Sağda ve solda olmak üzere iki adet ışık algılayıcısı (LDR) bulunan bu robotun amacı, ışığın geldiği yöne doğru hareket etmektir. Işık sağ algılayıcıya düşüyorsa sağa, sol algılayıcıya düşüyorsa sola, her ikisine de düşüyorsa ileri yönde hareket eder. Eğer hiçbir algılayıcıya ışık düşmüyorsa, robot durur. Bu şekilde robotu karanlık bir ortamda bir el feneri ile kontrol etmek mümkündür.



Şekil 1 Işık takip eden O-bOt.

4.2 O-bOt'un Işık Takip Eden Robot Haline Getirilmesi

Elimizde mevcut olan bir O-bOt'u ışık izleyen robot haline getirmek için yapmamız gereken tek şey, O-bOt'a iki adet ışık algılayıcısı (LDR) eklemek.



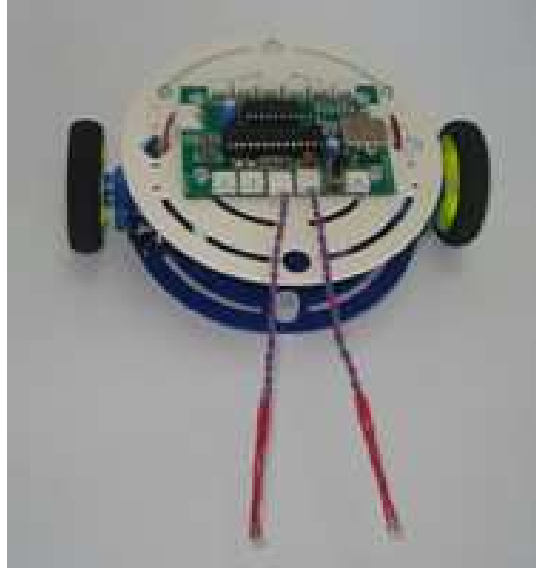
Şekil 2 O-bOt ve Işık Algılayıcılar.

1. Öncelikle, ışık algılayıcılarını resimdeki gibi O-bOt'a bağlıyoruz. Işık algılayıcıları analog çıkış veren algılayıcılardır ve O-bOt Kontrol Kartı'nın analog girişlerine bağlanmalıdırlar.

Bu programda “P11” ve “P12” tercih edilmiştir. Ancak derseniz programda gerekli deęişiklikleri yaptıktan sonra istediđiniz başka bir analog girişe bağlayabilirsiniz.

ÖNEMLİ NOT

O-bOt Kontrol Kartı ve analog pinlerin kullanımı ile ilgili detaylı bilgiye “Mekatronik Yapı Taşları” isimli dokümandan ulaşabilirsiniz.



Şekil 3 O-bOt Kontrol Kartına ışık algılayıcıların bağlantısı.

2. Daha sonra algılayıcının kablolarını O-bOt şasesinin önündeki delikten geçirerek O-bOt’a bir böcek görüntüsü verilebilir. Bu işlem hem algılayıcıları havada tutmamızı, hem de daha şık bir robota sahip olmamızı sağlar. Tüm bu işlemlerin ardından O-bOt, artık ışık takip etmeye hazır!



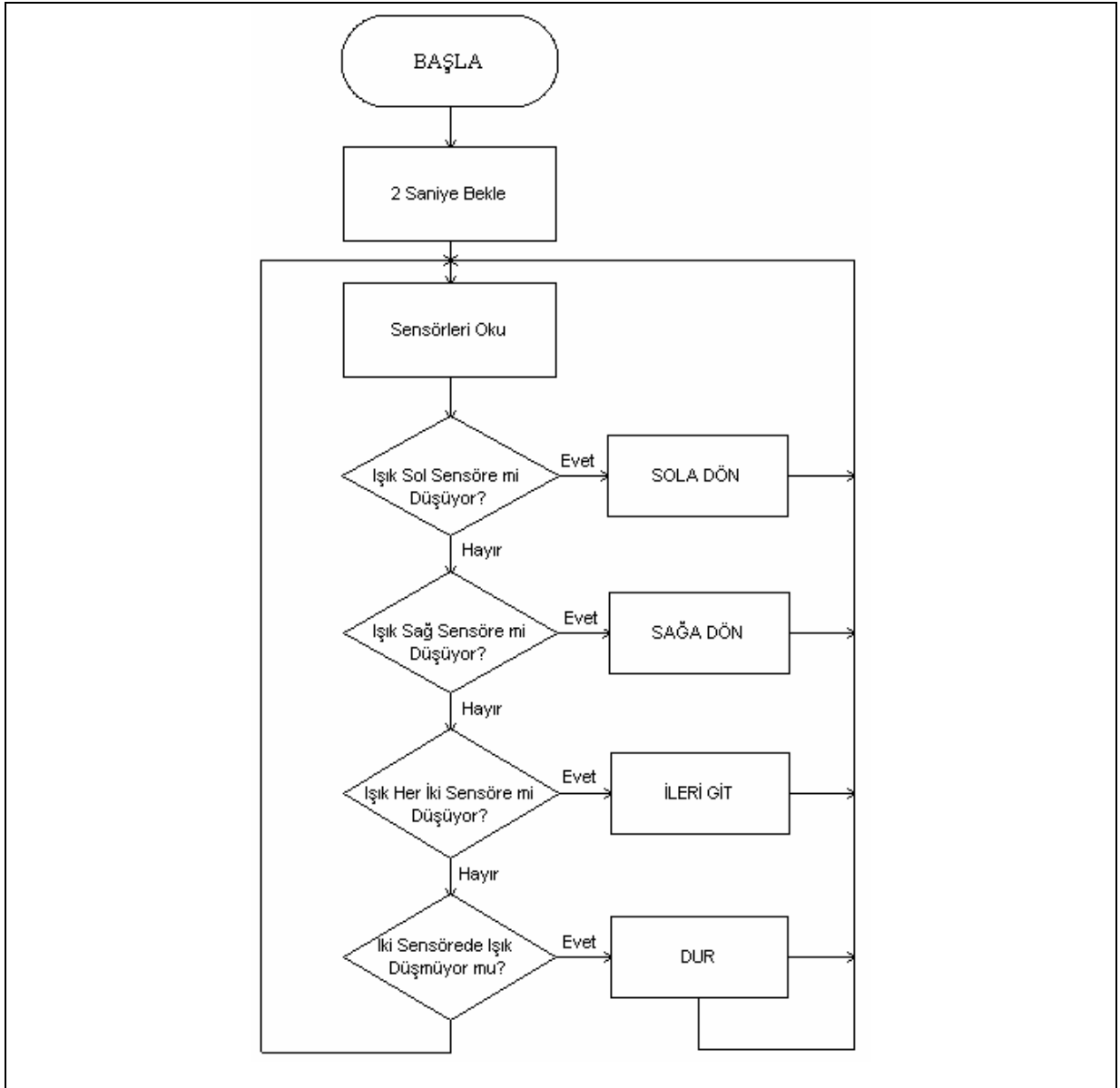
Şekil 3 Işık izleyen O-bOt.

4.3 Algoritma ve Programlama

Işık takip eden robotun üç temel durumu vardır. Bunlardan birincisi; iki algılayıcıya da ışığın düşmemesidir ve bu durumda robot duracaktır. İkinci durum ise ışığın iki algılayıcıdan birine düşmesidir. Işık sağ algılayıcı üzerine düşüyorsa robot sağa, sol algılayıcı üzerine düşüyorsa robot sola döner. Üçüncü ve son durum ise ışığın her iki algılayıcı üzerine de düşmesidir. Bu durumda ise robot düz olarak ilerler.

ÖNEMLİ NOT

Motor-a ile aynı yöndeki ışık algılayıcısının sol, Motor-b ile aynı yöndeki ışık algılayıcısının sağ algılayıcı olduğunu unutmayınız. Programlama sırasında oluşabilecek hataları en aza indirmek ve karmaşıklığı önlemek için yine programı yazmaya geçmeden önce programın akış şeması çıkarılmalıdır.



Şekil 4 Programın Akış Şeması.

Işık Takip Eden Robot Programı;

```
pause 2000 '2 sn bekle

symbol solsensor=w0 'Sol algılayıcı tanımlandı
symbol sagsensor=w1 'Sağ algılayıcı tanımlandı

kontrol: 'Algılayıcıları kontrol et
  readadc 0,sagsensor 'Sağ algılayıcıyı oku, değeri 'sagsensor' isimli
  değişkenin içerisine kaydet
  pause 1
  readadc 1,solsensor 'Sol algılayıcıyı oku, değeri 'solsensor' isimli
  değişkenin içerisine kaydet
  pause 1

  if solsensor>250 and sagsensor<250 then soladon 'Eğer sola ışık düşüyorsa
  ve sağa ışık düşmüyorsa,'soladon' programına git
  if sagsensor>250 and solsensor<250 then sagadon 'Eğer sağa ışık düşüyorsa
  ve sola ışık düşmüyorsa,'sagadon' programına git
  if sagsensor>250 and solsensor>250 then duzgit 'Eğer iki algılayıcıya ışık
  düşüyorsa 'duzgit' programına git
  if sagsensor<250 and solsensor<250 then dur 'Eğer iki algılayıcıya ışık
  düşmüyorsa 'dur' programına git
  goto kontrol 'Algılayıcıları tekrar kontrol et

soladon:      'Robotu sola döndüren program
backward a    'Motor a'yı geriye hareket ettir
forward b     'Motor b'yi ileriye hareket ettir
pause 10      '10 ms bekle
goto kontrol  'Yeni durum için algılayıcıları kontrol et

sagadon:      'Robotu sağa döndüren program
forward a     'Motor a' yı ileri hareket ettir
backward b    'motor b'yi geri hareket ettir
pause 10      '10 ms bekle
goto kontrol  'Yeni durun için algılayıcıları kontrol et

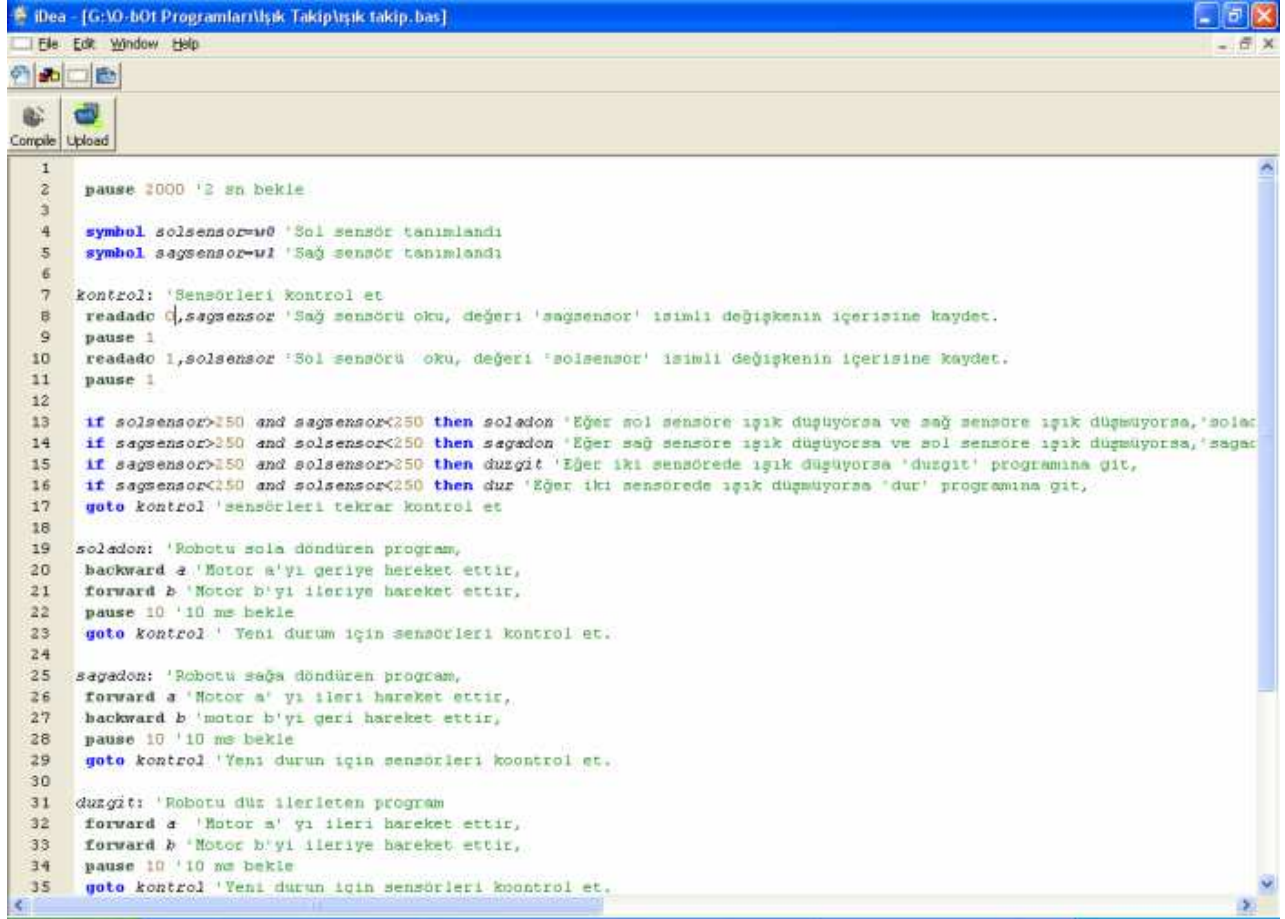
duzgit:       'Robotu düz ilerleten program
forward a     'Motor a' yı ileri hareket ettir
forward b     'Motor b'yi ileriye hareket ettir
pause 10      '10 ms bekle
goto kontrol  'Yeni durun için algılayıcıları kontrol et

dur:          'Robotu durduran program
halt a        'Motor a'yı durdur
halt b        'Motor b'yi durdur
pause 10      '10 ms bekle
goto kontrol  'Yeni durun için algılayıcıları kontrol et

end ' Programın sonu.
```

Şekil 5 Işık takip eden robot programı.

Programı iDea ile yazdıktan ve O-bOt ile USB bağlantısını yaptıktan sonra “YÜKLE” simgesine tıklayarak programı O-bOt’a yükleyebilirsiniz. O-bOt artık ışık takip etmek için hazır!..



```
1
2  pause 2000 '2 sn bekle
3
4  symbol solsensor=w0 'Sol sensör tanımlandı
5  symbol sagsensor=w1 'Sağ sensör tanımlandı
6
7  kontrol: 'Sensörleri kontrol et
8  readadc 0,sagsensor 'Sağ sensörü oku, değeri 'sagsensor' isimli değişkenin içerisine kaydet.
9  pause 1
10 readadc 1,solsensor 'Sol sensörü oku, değeri 'solsensor' isimli değişkenin içerisine kaydet.
11 pause 1
12
13 if solsensor>250 and sagsensor<250 then soladon 'Eğer sol sensöre ışık düşüyorsa ve sağ sensöre ışık düşmüyorsa,'solad
14 if sagsensor>250 and solsensor<250 then sagadon 'Eğer sağ sensöre ışık düşüyorsa ve sol sensöre ışık düşmüyorsa,'saga
15 if sagsensor>250 and solsensor>250 then duzgıt 'Eğer iki sensörde ışık düşüyorsa 'duzgıt' programına git,
16 if sagsensor<250 and solsensor<250 then dur 'Eğer iki sensörde ışık düşmüyorsa 'dur' programına git,
17 goto kontrol 'sensörleri tekrar kontrol et
18
19 soladon: 'Robotu sola döndüren program,
20 backward a 'Motor a'yı geriye hareket ettir,
21 forward b 'Motor b'yi ileriye hareket ettir,
22 pause 10 '10 ms bekle
23 goto kontrol 'Yeni durum için sensörleri kontrol et.
24
25 sagadon: 'Robotu sağa döndüren program,
26 forward a 'Motor a' y'i ileri hareket ettir,
27 backward b 'motor b'yi geri hareket ettir,
28 pause 10 '10 ms bekle
29 goto kontrol 'Yeni durum için sensörleri kontrol et.
30
31 duzgıt: 'Robotu dur iletilen program
32 forward a 'Motor a' y'i ileri hareket ettir,
33 forward b 'Motor b'yi ileriye hareket ettir,
34 pause 10 '10 ms bekle
35 goto kontrol 'Yeni durum için sensörleri kontrol et.
```

Şekil 5 Programın iDea ile yazılması.

iDea Yazılımı ve O-bOt kullanarak, sadece gerekli modüler bileşenleri eklemeniz, robot programlamak için yeterli.

Robot programlamak hiç bu kadar kolay olmamıştı!..



“Herşey Yaratıcılığınızın Sınırlarında...”

