

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GEZGİN ROBOT İÇİN
YÖN BULMA UYGULAMASI**

154459

Elektronik Müh. Murat YALÇIN

**FBE Elektronik ve Haberleşme Anabilim Dalı Elektronik Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ



Tez Danışmanı : Yrd. Doç. Dr. Tuncay UZUN



Prof. Dr. Oruç Bilgiç



Y. Doç. Dr. Ferit ATTAR

İSTANBUL, 2004

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER.....	ii
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ.....	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ.....	1
1.1 Robotlar Hakkında.....	1
1.2 Yön Bulma.....	5
1.3 Algılayıcılar	7
1.3.1 Işığa Bağımlı Direnç (LDR)	7
1.3.2 Diğer Algılayıcılar	9
1.4 Karar Algoritmaları	14
1.4.1 Yapay Sinir Ağı Algoritmaları	14
1.4.2 Klasik Algoritmalar	18
1.5 Haberleşme Teknikleri.....	18
1.5.1 Seri (RS232) Haberleşme	18
1.5.2 Diğer Haberleşme Teknikleri	27
2. YSA ile YÖN BULMA ALGORİTMALARININ SİMULASYONU.....	30
2.1 MATLAB Hakkında.....	30
2.2 MATLAB’da YSA	31
2.3 YSA Programları	31
2.3.1 Resim Bilgisi için YSA Programı.....	32
2.3.2 Resim, Aydınlık ve Mesafe Bilgileri için YSA Programı	35
2.4 Sonuçlar	37
3. YÖN BULMA UYGULAMASININ DONANIM VE YAZILIMI	38
3.1 Yön Bulma Uygulamasının Donanım Sistemi	38
3.2 Algılayıcı Kartı	41
3.3 Kontrol Kartı.....	42
3.3.1 Güç Devresi	44
3.3.2 Mikrodenetleyici Devresi	44
3.3.3 Sesli Uyarı Devresi.....	45

3.3.4	Seri Port Sürücü Devresi	45
3.3.5	Yön Gösterge Devresi.....	46
3.4	Yön Bulma Uygulamasının Yazılımı	47
3.4.1	Mikrodenetleyici Yazılımı.....	47
3.4.2	Bilgisayar Yazılımı.....	49
3.4.3	Haberleşme Protokolü	52
4.	SONUÇ.....	54
KAYNAKLAR.....		55
EKLER		56
Ek 1 PC Seri Haberleşmesi için Kullanılan Çekirdek Programlar		57
Ek 2 MATLAB Çalışmalarında Oluşturulan Program Dosyaları		60
(Sadece Resim Bilgisi Giriş Olarak Kullanıldığında)		60
Ek 3 MATLAB Çalışmalarında Oluşturulan Program Dosyaları		62
(Resim, Aydınlık ve Mesafe Bilgileri Giriş Olarak Kullanıldığında)		62
Ek 4 Donanımda Kullanılan Özel Malzemelerin Veri Dosyaları (Datasheets).....		66
ÖZGEÇMİŞ.....		71

SİMGE LİSTESİ

Ω	Direnç değeri
α	LDR için kullanılan bir sabit katsayı
μsn	mikro saniye



KISALTMA LİSTESİ

GPS	Ayrıntılı pozisyon sistemi (Global Position System)
LDR	Işığa bağımlı direnç (Light Dependent Resistor)
NTC	Negatif sıcaklık iletkeni (Negative Temperature Conductance)
PTC	Pozitif sıcaklık iletkeni (Positive Temperature Conductance)
DSP	Sayısal sinyal işleme (Digital Signal Processing)
CCD	Sayısal kamera (Charge-Coupled Devices)
PDA	El bilgisayarı (Personal Digital Assistants)
RGB	Kırmızı, yeşil, mavi (Red-Green-Blue)
SNR	Sinyal parazit oranı (Signal-to-Noise Ratio)
YSA	Yapay Sinir Ağları (Artificial Neural Network)
ANN	Yapay Sinir Ağları (Artificial Neural Network)
MLP	Çok Katmanlı Perceptron (Multi Layer Perceptron)
PNN	Olasılık Temelli Sinir Ağı (Probabilistic Neural Network)
GRNN	Genelleştirilmiş Sinir Ağı (Generalized Regression Neural Network)
RBF	Dairesel Temelli Fonksiyon (Radial Basis Function)
PCB	Baskılı Devre (Printed Circuit Board)
TTL	Transistor Transistor Logic
LED	Işık Yayan Diyot (Light Emitter Diode)
TxD	Veri Gönderici (Transmit Data)
RxD	Veri Alıcı (Receive Data)

ŞEKİL LİSTESİ

Şekil 1.1 Müşteri odaklı bir oyuncak robot (Aibo).....	3
Şekil 1.2 Araştırma amaçlı gezgin robot.....	3
Şekil 1.3 Devreler ve Sistemler Bölümündeki Robot Mekaniği	4
Şekil 1.4 Işığın fonksiyonu olarak LDR direncinin değişimi	8
Şekil 1.5 LDR için fiziksel boyutlar	8
Şekil 1.6 Ortam sıcaklığına bağlı güç değişimi	9
Şekil 1.7 Ultrasonik algılayıcı örneği	10
Şekil 1.8 Sıcaklık algılayıcı örneği	10
Şekil 1.9 Kuvvet, Basınç ve İvme Algılayıcıları	11
Şekil 1.10 Endüstriyel tip bir kamera.....	12
Şekil 1.11 CCD kameraların kullanım alanları	13
Şekil 1.12 Biyolojik sinir hücresi ve bölümleri	14
Şekil 1.13 YSA hücresi	15
Şekil 1.14 MLP ağ yapısı örneği	17
Şekil 1.15 Gezgin robot için YSA ile koridorda yön bulma	17
Şekil 1.16 Gezgin robot için YSA ile labirentte yön bulma.....	18
Şekil 1.17 Seri ve Paralel veri aktarımları.....	19
Şekil 1.18 Seri veri aktarım yöntemleri	20
Şekil 1.19 ASCII 'A' (41H) karakterinin çerçevesi.....	21
Şekil 1.20 DTE-DCE ve DTE-DTE bağlantıları	23
Şekil 1.21 RS232 DB-25P erkek konnektör.....	23
Şekil 2.1 Sadece resim içeren Learn_bw.m programına ait algoritma	32
Şekil 2.2 Sadece resim içeren Test_bw.m programına ait algoritma.....	33
Şekil 2.3 1x2 nokta siyah-siyah	34
Şekil 2.4 1x2 nokta beyaz-siyah	34
Şekil 2.5 1x2 nokta beyaz-beyaz	34
Şekil 2.6 1x2 nokta siyah-beyaz	34
Şekil 2.7 2x2 nokta beyaz-beyaz beyaz-beyaz	34
Şekil 2.8 2x2 nokta siyah-beyaz beyaz-beyaz	34
Şekil 2.9 2x2 nokta siyah - siyah beyaz-beyaz	34
Şekil 2.10 2x2 nokta siyah - siyah siyah -beyaz	34
Şekil 2.11 2x2 nokta siyah - siyah siyah - siyah.....	34
Şekil 2.12 2x2 nokta siyah -beyaz beyaz- siyah	34
Şekil 2.13 10x10 nokta boyutunda görüntü.....	34
Şekil 2.14 Resim, Aydınlık ve Mesafe içeren Test_bw.m programına ait algoritma.....	35
Şekil 2.15 1x2 nokta beyaz-siyah	37
Şekil 2.16 4x4 nokta LDR bilgisi	37
Şekil 3.1 Yön Bulma Uygulaması Donanım Sisteminin Genel Blok Diyagramı.....	38
Şekil 3.2 Yön Bulma Kontrol Modülünün detaylı blok diyagramı	39
Şekil 3.3 Yön Bulma Kontrol Modülünün çalışır haldeki görüntüsü	40
Şekil 3.4 Yön Bulma Kontrol Modülünün farklı açılardan iki görüntüsü	41
Şekil 3.5 Farklı yüksekliklerde monte edilmiş algılayıcı kartları.....	41
Şekil 3.6 Her bir LDR için kullanılan transistör devresi.....	42
Şekil 3.7 Kontrol kartının şematik çizimi	43
Şekil 3.8 Kontrol kartının üstten görünüşü.....	44
Şekil 3.9 Gerilim doğrultucu ve düzenleyici güç devresinin şeması	44
Şekil 3.10 Sesli uyarı devresinin şeması	45
Şekil 3.11 Seri port sürücü devresinin şeması	46
Şekil 3.12 Yön gösterge devresinin şeması.....	47

Şekil 3.13 Mikrodenetleyici yazılımının akış diyagramı	48
Şekil 3.14 Bilgisayar yazılımının akış diyagramı	50
Şekil 3.15 Bilgisayar yazılımının görüntüsü	50
Şekil 3.16 Program hakkında bilgi penceresi	51
Şekil 3.17 Bilgisayar seri port veri akışını gözlemek için yardımcı pencere	51
Şekil 3.18 Protokol yapısı	52



ÇİZELGE LİSTESİ

Çizelge 1.1 Lisans Bitirme Listesi	5
Çizelge 1.2 Lisans Proje Listesi.....	5
Çizelge 1.3 Yüksek Lisans Tez Listesi	5
Çizelge 1.4 Bilgisayar ve Beynin İşlevsel Farklılıkları	15
Çizelge 1.5. IBM PC 9-uçlu ve 25-uçlu konnektör	24
Çizelge 1.6 RS232'nin RS422 ve RS423 ile karşılaştırılması	25
Çizelge 1.7 PC Seri Port adres atamaları	26
Çizelge 1.8 PC Seri Port I/O adres atamaları	26
Çizelge 2.1 MATLAB 6.5'deki YSA algoritmaları.....	32
Çizelge 2.2 Kullanılan Bilgisayar Özellikleri.....	37
Çizelge 3.1 Seri Port Ayarları.....	48
Çizelge 3.2 Protokol bileşenleri listesi.....	53



ÖNSÖZ

Bu çalışmanın hayata geçirilmesinde yardımlarını esirgemeyen, büyük bir sabır ve özveri ile çalışmalarına yön veren, Yıldız Teknik Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümünde, kendisinden aldığım yüksek lisans derslerinde mikrodenetleyiciler konusundaki uygulama ödevleri ile bu çalışmamda altyapı oluşturmamda bana destek olan, bundan sonraki çalışmalarım ile bu konuyu daha ileriye taşımada yanımda olacağına inandığım çok değerli hocam ve tez danışmanım Yrd. Doç. Dr. Tuncay Uzun'a sonsuz teşekkürlerimi sunarım.

Çalışmamın karar algoritmaları kısmında kullandığım yapay sinir ağları konusunda makale yayınlamamda öncülük eden yapay sinir ağları ile ilgili olarak kendisinden aldığım derslerde gerek içerik konusunda doyurucu bilgiler veren gerekse öğretici ödevleri ile yol gösteren sayın Doç. Dr. Tülay Yıldırım'a teşekkür ederim.

Uygulama aşamasında firması ile sağladığı imkanlardan dolayı değerli büyüğüm sayın Zihni Liman'a teşekkürlerimi sunarım.



ÖZET

Bu çalışmada; özgün bir karar algoritması denenmiş ve uygulaması yapılarak sonuçları incelenmiştir. Karar aşamasında temel olarak Yapay Sinir Ağlarına dayalı bir sistem tasarlanmıştır. Bu sistemde ışığa bağımlı dirençler (LDR) temel algılayıcılardır. Algılayıcılardan alınan sinyaller yapay sinir ağlarıyla tasarlanmış karar algoritmasına giriş bilgisi olarak aktarılmıştır. Algoritma çıkışındaki yön bilgisi gezgin robotu yönlendirmek için kullanılmaktadır. Sonuçta ışığa doğru ilerleyebilecek bir gezgin robotun ihtiyaç duyduğu yön bilgisini yapay sinir ağlarıyla tasarlanmış karar algoritmasından elde eden bir sistem tasarlanmış ve gerçekleştirilmiştir.

Gezgin robotun uygulama aşamasında ışığa bağımlı dirençler kare matris şeklinde yerleştirilerek 4x4 noktadan boyutunda çekirdek resim görüntüsü oluşturulmaya çalışılmıştır. Böylece hazırlanan karar algoritmasının bundan sonraki daha büyük resimlerle gerçekleştirilecek çalışmalar için temel oluşturması amaçlanmıştır.

Bütün bu çalışmalara ek olarak bilgisayar ortamında hazırlanan görsel bir program ile robotun takibi sağlanmıştır. Program aracılığıyla gezgin robotun yön bilgisinin yanında ışığa bağımlı dirençlerin de durumu gözlemlenebilmektedir.

Anahtar kelimeler: Gezgin robot, gezgin robot için yön bulma, karar algoritması, yapay sinir ağı, ışığa bağımlı algılayıcı.

ABSTRACT

In this working; a decision algorithm, that is completely different to other older working, created to navigation of mobile robots. It applied in real application and examined its results. At the decision step a new system designed which is based on Artificial Neural Network (ANN). Light Dependent Resistors are basic sensor in this system. The input signals which are obtained LDR transferred to ANN decision algorithm. The direction information at the output of ANN algorithm is used on navigation of mobile robot. As a result of this working a mobile robot which have this control system goes towards to light source by itself.

At the application step of mobile robot, light dependent resistors put into a cell place of two dimensions (4x4) matrix. So it is aimed to create a miniature image view. In this way the decision algorithm has extendable architecture to develop the system in the future.

In other hand, a visual computer program which is developed on C++ Builder environment added to the working. The program observes a mobile robot and its sensor status.

Keywords: Mobile robot, navigation of mobile robot, decision algorithm, artificial neural network (ANN), light dependent resistor (LDR).

1. GİRİŞ

Günümüze kadar gerçekleştirilen robot geliştirme çalışmalarında yön bulma işlemi için bir çok algılayıcı çeşidi kullanılmıştır. Robotun hangi yöne gitmesi gerektiği bu algılayıcılardan alınan sinyallerin çeşitli algoritmalarından geçirilmesiyle karara bağlanmıştır. Ama insan gözünün performansı ile karşılaştırıldığında çok fazla yol alınabildiği iddia edilemez.

Bu bölüm girişinde robotlar ve tarihçesi hakkında genel bilgiler sunulmaktadır. Daha sonra robotların kullanım alanlarına kısaca değinilmiştir. Ardından üniversitemizde hazırlanan robot çalışmaları listelenmiştir.

Tez çalışmamın temelini oluşturan yön bulma (*navigation*) işlemi bölüm ortasında yer almaktadır. Bunu takip eden konularda yön bulma işleminde kullanılan başlıca algılayıcılar açıklanmıştır. Özellikle uygulama aşamasında kullanılan ışığa bağımlı dirençler hakkında detaylı bilgilere yer verilmiştir. Daha sonra karar algoritmalarına değinilmiş ve yapay sinir ağlarının karar aşamasındaki önemi vurgulanmıştır.

Bölüm sonunda ise haberleşme (özellikle seri haberleşme - RS232) konusunda ayrıntılı bilgiler yer almaktadır.

1.1 Robotlar Hakkında

Robot nedir?

“*Robot*” kelimesi Çekçe bir terim olup ilk kez Karel Capek tarafından kullanılmıştır. Karel Capek kendi yazdığı oyundaki “akıllı işçileri” başlangıçta “*labori*” olarak adlandırmayı düşünmüş fakat kardeşinin de önerisiyle “*robot*” kelimesini kullanmaya karar vermiştir. Böylece “*robot*” terimi ilk kez ortaya atılmıştır.

Günümüzde robot, programlandıktan sonra kendi başına çalışabilen makine olarak tanımlanmaktadır.

Robot Tarihi

1818’de Mary Shelly *Frankenstein*’ı yazmıştır. *Frankenstein*, görünüş olarak insana benzeyen bir yaratığın öyküsüdür. Shelly için robot, insana benzer ama davranış ve fonksiyon olarak o bir makinedir. O insan parçalarından ortaya çıkmıştır ama bir insan değildir.

Karel Capek’in popüler oyununun adı “Rossum’s Universal Robots” 1921’de yazılmıştır. Çek yazar Karel Capek, bu oyunda akıllı işçilerden “robot” olarak söz etmektedir. Robot kelimesi

Çekçe’de işçi anlamında kullanılmaktadır ama İngilizceye çevrilmeden olduğu gibi alınmıştır.

1940’lara kadar robot kelimesi birçok eserde işlenmiştir. Ama 1940’larda bilgisayarın üretilmesiyle gerçeğe dönüşmesi yolunda ilk adımlar atılmaya başlanmıştır.

1941’de, bilim kurgu yazarı Isaac Asimov ilk kez “*robotics*” kelimesini kullanmıştır. Bu şekilde robot teknolojisinden söz açmış ve geleceğin güçlü robot endüstrisinin yükselişini öngörmüştür. Onun bu öngörüsü çok kısa zamanda gerçekleşmiştir. Son zamanlarda yaşanan endüstriyel robotların üretimi ve gelişimindeki büyük patlama bulunduğumuz çağa “robot devrimi”, “robot çağı” gibi isimler verilmesine yol açmıştır.

1956’da, George Devil ve Joseph Engelberger dünyanın ilk robot şirketini kurmuştur.

1960’ların başında ilk endüstriyel robotlar Güney Amerika’da Kitchener, Ontario kentindeki şeker fabrikasında kullanılmaya başlanmıştır.

1961’de ise New Jersey’deki otomobil fabrikası General Motors, ilk robot işçisini işe almıştır.

1980’den günümüze, otomotiv endüstrisinde robot kullanımı konusunda tam anlamıyla bir patlama gerçekleşmiştir. Buna sebep olarak bilgisayar ve mikro elektronikteki ilerleme gösterilmektedir. Robotların üretime getirdiği hız, kalite ve dolayısıyla verimlilik onları vazgeçilmezler arasına katmaktadır.

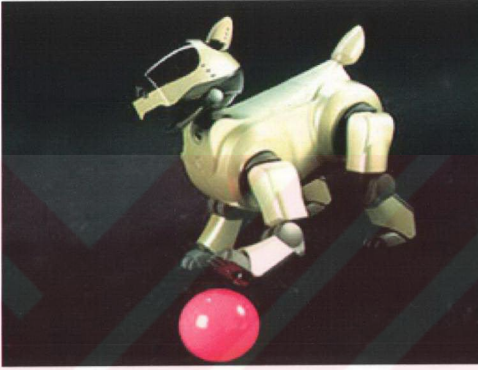
Robot Kuralları

- **Kural Sıfır:** Bir robot insanlığa zarar verecek herhangi bir davranışta bulunmamalıdır.
- **Kural Bir:** Bir robot insanı incitecek herhangi bir davranışta veya eylemde bulunmamalıdır.
- **Kural İki:** Bir robot ilk kuralla çakışmadığı sürece insanın verdiği emirlere uymalıdır.
- **Kural Üç:** Bir robot kendi varlığını korumaya çalışmalıdır. Fakat bu koruma eylemi birinci ve ikinci kuralla çatışmamalıdır.

Robotlar Nereelerde Kullanılıyor?

- **Müşteri Odaklı Robotlar:** Ticari amaçla tasarlanmış robotlardır. Bunların arasında hizmetçi robotlar, temizlik robotları, oyuncak robotlar vb... sayılabilir. Şekil 1.1’de Aibo ismiyle piyasaya sürülen bir ticari robot görülmektedir.
- **Keşif ve Araştırma Amaçlı Robotları:** Uzay çalışmaları, coğrafik amaçlar kullanılan robotlardır. Şekil 1.2’de Coğrafik araştırmalarda kullanılan bir robot yer almaktadır.

- Gömülü Sistemler: Arabalar, mikrodalga fırınlar, kablosuz telefonlar içindeki gömülü sistemler de (mikro işlemcili kontrol kartları) robot olarak tanımlanabilir.
- Gerçek Zamanlı Sistemler: Uzak bir noktadan gerçek zamanlı bilgi aktarımı, konum belirleme (GPS: Global Position System) gibi uygulamalarda tercih edilen robotlardır.
- Çok Görevli Sistemler: Üretim sahalarında bazı işlemleri yüksek kalite ile sürekli ve otomatik olarak gerçekleştiren robotlar bu sınıfta kabul edilebilir.



Şekil 1.1 Müşteri odaklı bir oyuncak robot (Aibo)



Şekil 1.2 Araştırma amaçlı gezgin robot

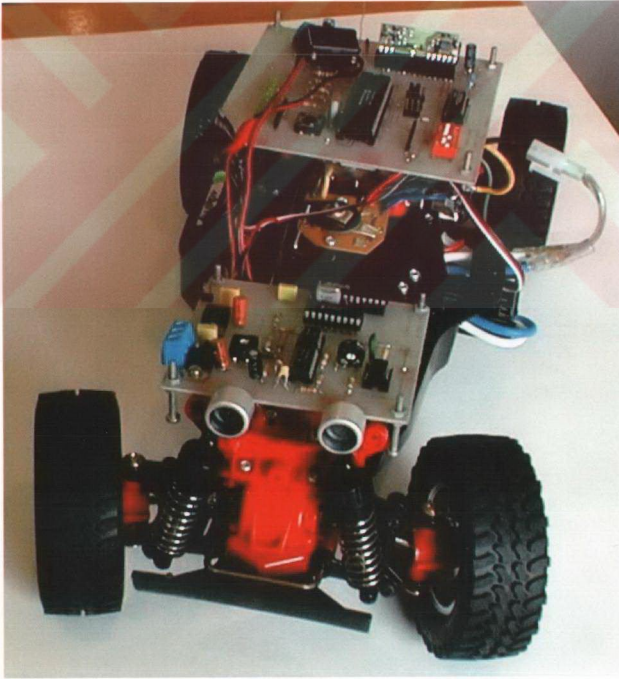
Günümüzde her 10 robottan 9'u endüstride kullanılmaktadır. Bunun anlamı robotlar birçok ortamda fabrikalar, laboratuvarlar, enerji sahaları, hastaneler vb... insanlar için çalışmaktadır.

Birkaç yıl öncesine kadar robotların yaklaşık %90'ı sadece araba fabrikalarında kullanılıyordu. Bunlar üretim hattında farklı görevlerde çalışıyorlardı. Şimdi ise dünyadaki robotların sadece yarısı araba üretmekle meşgul. Diğer yarısı yukarıda adı geçen endüstrilerde kullanılmaktadır.

Üniversitemizdeki Robot Çalışmaları

Üniversitemizde daha önceki yıllarda kısıtlı olanaklara rağmen gerçekleştirilen ve uygulama noktasına getirilen gezgin robot çalışmalarından bazıları Çizelge 1.1...3 ile aşağıda listelenmiştir.

Şekil 1.3'de görülen gezgin robot mekaniği ise bu çalışmalarda kullanılan bir düzenektir.



Şekil 1.3 Devreler ve Sistemler Bölümündeki Robot Mekaniği

Çizelge 1.1 Lisans Bitirme Listesi

Gezgin Robot İçin Sayısal Kamera Uygulaması	M. Alper SOYASLAN	2001
Gezgin Robot İçin Radyo Frekans Uzaktan Kumanda Donanımı	A. İnanç KÖROĞLU	2000
Gezgin Robot İçin Ultrasonik Uzaklık Algılayıcı Devresi	Afşin ÖZPINAR	2000
Gezgin Robot İçin Elektronik Denetim Sisteminin Yazılımı	Burak BORHAN	2000
Gezgin Robot Projesi Kontrol Donanımı Tasarımı	Güner Tunca ERDOĞAN	2000

Çizelge 1.2 Lisans Proje Listesi

Gezgin Robot İçin Radyo MODEM Devresinin Tasarımı ve Uygulaması	A. İnanç KÖROĞLU	1999
Gezgin Robot için Ultrasonik Uzaklık Ölçme Sisteminin Tasarımı ve Uygulaması	Afşin ÖZPINAR	1999
Gezgin Robot için Uzaktan Denetim Yazılımının Tasarımı ve Uygulaması	Burak BORHAN	1999
Gezgin Robot İçin Elektronik Denetim Donanımının Tasarımı ve Uygulaması	Güner Tunca ERDOĞAN	1999
Gezgin Robot İçin Güç Yönetim Donanımının Tasarımı ve Uygulaması	Tamer ÖZDEMİR	1999

Çizelge 1.3 Yüksek Lisans Tez Listesi

Bir Gezgin Robot İçin Elektronik Denetim Donanımının Tasarımı ve Uygulaması	Seçil ÖZEN	2000
Bir Gezgin Robot İçin Elektronik Denetim Yazılımının Tasarımı ve Uygulaması	Ekin YILDIZ	2000

1.2 Yön Bulma

Robotun programlanan rotadan yararlanarak şu andaki pozisyonundan hedef pozisyona doğru hareket etmesi yön bulma olarak tanımlanır. Yön bulma işlemi için anahtar sorular aşağıda sıralanmıştır.

- Neyi hatırlamalı yada referans almalıyım?
 - Harita (Cognitive Mapping)
- Şu anda neredeyim?
 - Yerleşim (Localization)
- Nereye gitmeliyim?
 - Yol Planı (Path Planning)
- Oraya nasıl ulaşırım?
 - Hareket Kontrolü (Motion Control)

Yukarıdaki sorular göz önüne alındığında robotlar için tasarlanacak yön bulma sisteminin bilgiyi işleyebilen bir yapısı olması gerektiği ortadır. Dolayısıyla bu sisteme sahip bir robot

çevresinden aldığı bilgilerin açılımını yapabilen ve edindiği bilgileri kullanarak anlamlı davranışlarda bulunabilen akıllı bir robot olacaktır. Hatta bir adım daha ilerisi düşünüldüğünde “ORAYA NEDEN GİTMELİYİM ?” sorusuna da cevap verebilecek robotlar tasarlanmış olacaktır. Yön bulma işlemi üç temel başlık altında toplanabilir.

- Genel Yön Bulma
- Yerel Yön Bulma
- Kişiyeye Dayalı Yön Bulma

Genel Yön Bulma

Bir harita yardımıyla dünya üzerinde koordinatları bilinen herhangi bir noktadan koordinatları verilen yeni bir noktaya ulaşma işlemidir. Bu yön bulma işlemine en güzel örnek Ayrıntılı Pozisyon Sistemi (GPS – Global Position System)’dir. Özellikle ulaşım araçlarında çok sık kullanım alanı bulan bu sistem tüm dünya ülkeleri tarafından kabul görmüştür.

Genel yön bulma harita temelli bir sistemdir. Dünya üzerindeki enlem ve boylam çizgileri referans alınarak belirlenen koordinat noktaları, hedef ve kaynak pozisyon bilgileri olarak alınır.

Genel yön bulma, bilerek yani bilinçli olarak yapılır. Robotun hedef ve kaynak pozisyonları hakkında kesin bilgisi vardır.

Genel yön bulma işleminde çok uzak mesafeler söz konusu olduğundan yavaş gerçekleştirilen bir eylemdir.

Yerel Yön Bulma

Algılayıcılar ve karar algoritmaları yardımıyla bir alan içinde belirlenen hedefe ulaşma çabasıdır. Örneğin masanın etrafından dolaşarak duvara ulaşmak yada labirent şeklindeki bir koridordan çıkış yolunu bulmak gibi.

Yerel yön bulma, algılayıcı temellidir. Robot, sahip olduğu algılayıcıların verdiği bilgiler ölçüsünde çevresini tanır. Bu sebeple algılayıcılardan gelen bilgilerin hem güvenilir olması hem de amaca uygun olması gerekir. Amaca uygunluktan kasıt ışığa yönelecek robotun algılayıcılarının ışığa duyarlı olmasıdır.

Yerel yön bulmada ortamdaki diğer nesnelerle etkileşim vardır. Robot algıladığı nesnelere bir tepki verir. Hedefe uygun olarak bazen yönünü değiştirir bazen de nesnenin üzerinden veya altından geçer.

Yerel yön bulma işlemi mesafelerin kısa olması itibariyle hızlı gerçekleşen bir eylemdir.

Bu tezde gerçekleştirilen uygulama çalışması bir yerel yön bulma eylemidir.

Kişiyeye Dayalı Yön Bulma

Kişinin uzaktan kumanda cihazı veya bilgisayardan vereceği komutlar aracılığıyla bir alan içinde belirlenen bir hedefe ulaşma çabasıdır. Algılayıcılardan gelen sinyallere göre harekete engel yoksa verilen komutlar işletilir veya komut değiştirilerek uygulanır. Yandaki odaya ulaşmak yada bir robot araba yarışında görev almak gibi.

Kişisel yön bulma işlemi, yerel yön bulmada bulunan parametreleri içerir. Fakat robota yardımcı olarak insanın algılayıcı ve karar yeteneği de devrededir. Dolayısıyla kişiyeye dayalı yön bulma işlemi, algılayıcıların da devrede olmasına rağmen komut temellidir.

Robotun hedefe ulaşma hızı onu kontrol eden kişiyeye bağımlı olacağı için kişiyeye dayalı yön bulma göreceli hızda gerçekleştirilir.

Sonuç olarak eğer yön bulma işlemi tümüyle robota bırakılacaksa (yerel yön bulma) çok önemli iki parametre ön plana çıkmaktadır. Bunlardan biri algılayıcı diğeri ise karar algoritmasıdır.

1.3 Algılayıcılar

Yerel yön bulmada bahsedildiği üzere yön bulma işleminde algılayıcılar robotun dış dünya ile olan ilişkisinde baş rolü oynar. Hatasız bir yön bulma işlemi için doğru bir algılayıcı seçimi zorunludur. Robotlarda kullanılan yön bulma amaçlı algılayıcılardan bazıları hakkında genel bilgiler aşağıda sunulmuştur. Bu çalışmada kullanılan LDR hakkında ise detaylı bilgilere yer verilmiştir.

1.3.1 Işığa Bağımlı Direnç (LDR)

LDR (Light Dependent Resistor), direnci ışığa bağımlı olarak değişen bir malzemedir. LDR hiç serbest elektron içermeyen veya çok az serbest elektron içeren *cadmium sulphide* maddesinden yapılır. Direnci oldukça yüksektir. Işığı algıladığı anda serbest elektronlar harekete geçer ve iletkenliği artar. Direncin ışığa bağımlı değişimi Denklem 1.1 ile verilmiştir.

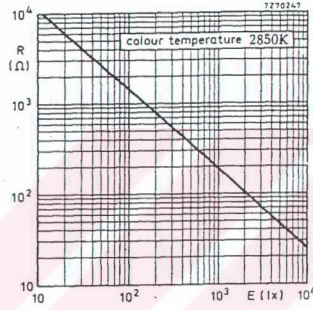
$$R = AL^{-\alpha} \quad (1-1)$$

$R = \text{Direnç } (\Omega)$

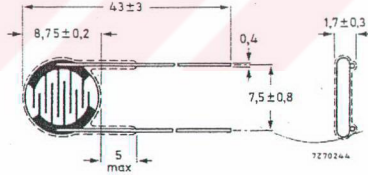
$L = \text{Işık değeri (lux)}$

A ve α ise sabit katsayılardır.

Katsayı α 'nın değeri LDR'nin üretiminde kullanılan cadmium sulphide maddesine bağlıdır. Genellikle 0.7...0.9 arasında bir değer alır. Direnç ve ışık arasındaki ilişki Şekil 1.4'de gösterilmiştir. LDR için fiziksel boyutları gösteren bir çizim Şekil 1.5'de yer almaktadır.



Şekil 1.4 Işığın fonksiyonu olarak LDR direncinin değişimi (Philips, 1983)



Şekil 1.5 LDR için fiziksel boyutlar (Philips, 1983)

Spektral Cevap (spectral response)

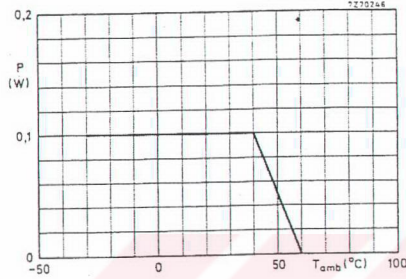
Direncin ışığa bağımlı değişimi sınırlı dalga boyları için mümkündür. LDR'nin en iyi cevap verebildiği dalga boyu 680nm olarak tespit edilmiştir. (Philips, 1983)

Sıcaklığa Bağımlılık

Serbest elektronlar sadece ışıktan değil sıcaklıktan da etkilenmektedirler. Karanlık direnci normal sıcaklıkta sınırsız değildir. Ortamın ısınmasıyla artar ve bir soğutucu yardımıyla da

azaltılabilir.

Sıcaklık, direnci ışık altında da etkileyebilir. Pratikteki ışıklandırma seviyeleri ve normal ortam sıcaklıklarında sıcaklık sabiti çok küçük olduğu için ihmal edilebilir. Sıcaklık bağımlılığına ait fonksiyon grafiği Şekil 1.6'da yer almaktadır.



Şekil 1.6 Ortam sıcaklığına bağlı güç değişimi (Philips, 1983)

Toparlanma Hızı (recovery rate)

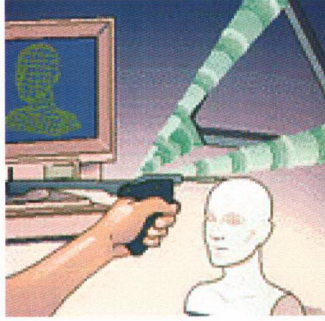
LDR, ışıklı ortamdan karanlık ortama geçirildiği anda olması gereken direnç değerine hızlı bir şekilde ulaşamaz yani toparlanma hızı düşüktür. LDR için doğrulama hızı birim zamanda değişen direnç değeri olarak hesaplanır. En sık kullanılan LDR tipleri için 1000lux değerindeki bir ışık ortamından karanlık ortama girildiği ilk 20sn için ortalama toparlanma hızı 200K Ω /s olarak ölçülmüştür. (Philips, 1983)

Toparlanma hızı diğer yönde yani karanlık ortamdan aydınlık ortama girildiğinde daha hızlıdır.

1.3.2 Diğer Algılayıcılar

Sonar (Ultrasonik Algılayıcı)

Ultrasonik algılayıcılar mesafe tespitinde kullanılır. Verici ve alıcı çiftinden oluşur. Verici tarafından ortama yüksek frekanslı ses dalgaları gönderilir. Alıcı tarafından seviyesi izlenen bu dalgaların yoğunluğu değiştiğinde bir engelin varlığı söz konusudur. Ultrasonik algılayıcı ile engel arasındaki mesafe alıcı çıkışından okunan gerilim değerlerinden yararlanarak hesaplanabilir. Şekil 1.7'de ultrasonik algılayıcı yardımıyla bir portrenin bilgisayara aktarımını gösteren resim yer almaktadır. (Erdoğan, 2000)



Şekil 1.7 Ultrasonik algılayıcı örneği (Erdoğan, 2000)

Ultrasonik algılayıcıların hassasiyeti vericiden gönderilen sinyalin gücüne bağlı olarak değişim gösterir. Ölçülebilen mesafeler birkaç santimetreden denizaltında kullanıldığı üzere metrelerce mesafelere kadar uzanabilir.

Günümüzde bu tip algılayıcılar arabaların park etmede kolaylık sağlaması amacıyla bir çok araba firması tarafından kullanılmaya başlanmıştır.

Sıcaklık Algılayıcıları

Robotun bulunduğu ortam sıcaklığını ölçmek robot mekanizmasını korumak için gerekebilir. Birçok uygulamada sıcaklık ölçümü NTC, PTC gibi direnci sıcaklıkla değişen elemanlarla ölçülebileceği gibi P-N birleşiminin sıcaklıkla değişiminden yararlanılarak yapılmış yarıiletken algılayıcıların kullanılması da mümkündür. Şekil 1.8’de bir sıcaklık algılayıcısının fiziksel yapısı örnek olarak verilmiştir. (Erdoğan, 2000)



Şekil 1.8 Sıcaklık algılayıcı örneği

Sıcaklık değişimleri çok ani gerçekleşmediği için bu tip algılayıcıların toparlanma hızı (recovery rate) genellikle düşük olur. Sıcaklık algılayıcılarından alınan değerler analog bir gerilim üzerinden mikrodenetleyiciye aktarılır. (Erdoğan, 2000)

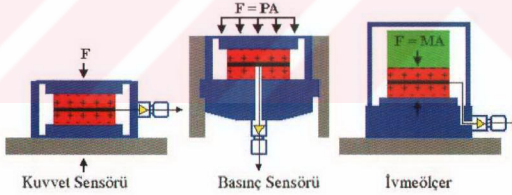
Basınç, Kuvvet, İvme Algılayıcıları

Rezistif elemanlardan oluşturulmuş bir Wheatstone köprüsü yardımı ile ölçülür, köprünün dört yanı da değişken dirençle oluşturulabileceği gibi sadece bir yanı da değişken direnç ile oluşturulmuş olabilir. Özellikle ağırlık ölçümünde kullanılan basınç algılayıcıları yapı olarak ince bir şerit biçiminde kıvrılmış ve boyu uzatılmış bir iletkenin sağlam bir yüzeye sıkıca tutturulmasıyla gerçekleştirilmiştir. Bu yüzeyin üzerine yük konulduğunda kıvrılmış telin boyu ve dolayısıyla direnci değişecektir. Bu esnada Wheatstone köprüsünün dengesi de değişeceğinden bu değişim miktarı tespit edilerek yükün ağırlığı hesaplanabilir.

Bu yolla ani darbelerin algılanması (ivme ölçer) da mümkündür.

Bunların ötesinde örneğin deniz altında gerçekleştirilecek yön bulma uygulamaları gibi bazı robot uygulamalarında ortam basıncının da algılanması gerekebilir. Bu ölçümlerin piezo elektrik esasına dayanarak çalışan algılayıcılarla da yapılması mümkündür.

Şekil 1.9'da kuvvet, basınç ve ivme algılayıcılarına ait bir çizim yer almaktadır.



Şekil 1.9 Kuvvet, Basınç ve İvme Algılayıcıları

Nem Algılayıcıları

Bazı robot yön bulma uygulamalarında ortam neminin ölçülmesi de yararlı olabilmektedir. Nem ölçümü genellikle "higrometre" ile gerçekleştirilir. Higrometreler kapasitif etkiyle çalışırlar. Yapı olarak ortada yalıtkan malzeme ve bunun iki yanında iletken malzeme olmak üzere üç katmandan oluşur. Katmanlar üst üste yerleştirilmişlerdir. Ortadaki katman su geçirmez özelliktedir. Üsteki katman ise suyu sızdıracak bir yapıya sahiptir. Bu katmandaki nem oranına göre sistemin kapasitesi değişeceğinden ortam nemi ölçülebilmektedir.

Kameralar

Günümüzde gömülü sistemlerin yüksek hızlarda çalışabilmesi ve kapasite probleminin büyük ölçüde aşılmış olması robotlarda algılayıcı olarak kameraların kullanımını arttırmıştır.

Kameralardan alınan görüntüler sayısal sinyal işleme (DSP – Digital Signal Processing) teknikleri ile irdelenerek robotun önündeki yolun incelemesi yapılır. Kamera görüntüleri kenar algılayan, kontrast farklılıklarını ayırt eden ve renk çözümlemeleri yapan vb... yazılımsal filtrelerle incelendiğinde yol üzerindeki nesneler hakkında bazı açıklayıcı bilgiler edinilebilir. Diğer algılayıcılardan gelen bilgiler de göz önünde bulundurulduğunda karar algoritmasından gidilecek yön hakkında daha doğru çıkışlar alınabilir.

Kameralar gerek görüntüyü algılama yöntemleri (örn. *Linescan*), gerekse bilgiyi aktarma yolları bakımından çok çeşitlilik gösterirler. Kameralarda kullanılan yardımcı elemanlar da (örn. *lens*) uygulamaya göre çeşitlilik gösterir. Ayrıca ortamın ışıklandırılması kameradan alınan görüntünün kalitesinde doğrudan etkili olmaktadır.

Kamera görüntülerinden yararlanarak ürün kalite testlerinde kullanılmak üzere geliştirilen robotlar söz konusudur. Bu tarz sistemler yapay göz (*machine vision*) sistemleri olarak bilinir. Ülkemizde endüstri alanında yapay göz ile ürün testleri yapan firmaların sayısı giderek artmaktadır. Şekil 1.10'da bu tarz bir kameranın görüntüsü yer almaktadır.



Şekil 1.10 Endüstriyel tip bir kamera

Aşağıda sıkça kullanılan CCD kamera hakkında genel bilgiler sunulmuştur.

CCD Kameralar

CCD kameralar yapay göz sistemleri, denetim uygulamaları, kalite testleri, güvenlik gibi uygulamalarda endüstriyel ve ticari amaçlarla kullanılır. Şekil 1.11'de CCD kameraların

kullanım alanları hakkında bazı resimler yer almaktadır.

CCD kameralar renkli, gri veya siyah-beyaz görüntü aktarabilir. Renkli görüntünün oluşturulmasında üç ana renk olan kırmızı, yeşil, mavinin (RGB) farklı karışımları kullanılır. Gerektiğinde kamera içindeki renk ayrıştırıcı filtrelerden yararlanarak tipik renk ayrımları gerçekleştirilebilir.



Şekil 1.11 CCD kameraların kullanım alanları

CCD kameralar için önemli özellikler yatay çözünürlük (*horizontal resolution*), görüntü okuma hızı (*maximum frame rate*), pozlama hızı (*shutter speed*), duyarlılık (*sensitivity*) ve sinyal parazit oranı (*SNR - signal-to-noise ratio*) olarak sıralanabilir.

Yatay çözünürlük, bir satırda bulunan en fazla nokta (*en küçük resim elemanı - pixel*) sayısıdır. Birçok kaynakta ortak gösterim ise bütün görüntü içindeki toplam nokta sayısını ifade edecek biçimde satır ve sütun nokta sayısı çarpımıdır. Örneğin 640x480.

Görüntü okuma hızı, kameranın bir saniye içinde yakaladığı görüntü sayısıdır.

Pozlama hızı veya diğer deyişle pozlama zamanı iki pozlama arasında geçen zamandır. Genellikle en yüksek seviyede iken resim kalitesi daha iyi olmaktadır. Sürenin kısaltılması kaliteden ödün vermek anlamına gelir.

Sinyal parazit oranı, kameranın çıkışına aktarılan görüntünün gerçek görüntüye göre ne kadar bozucu işaret içerdiğinin bir göstergesidir.

CCD kameralarla birlikte kullanılan analog sinyal formatları NTSC, PAL, SECAM, RS-170, RS-330 ve CCIR 'dır. En çok kullanılan sayısal çıkış formatları ise RS232, RS422, RS485, paralel arabirimler, Ethernet, DeviceNet, ARCNET, PROFIBUS, CANbus, Foundation Fieldbus, IEEE 1394 (Firewire), Modem, SCSI, TTL, USB ve RF olarak sıralanır. CCD kameralarda sinyal sayısal olarak aktarılıyorsa 8, 10, 12, 14 veya 16 bitlik paketlere bölünür. Renkli görüntü aktarımlarında resim sinyali RGB, S-video yada birleşik (*composite*) olarak aktarılır.

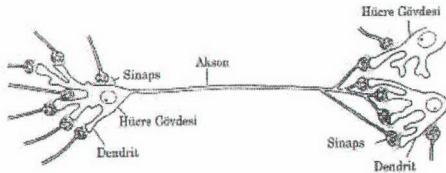
1.4 Karar Algoritmaları

1.4.1 Yapay Sinir Ağı Algoritmaları

Yapay Sinir Ağı Nedir?

Yapay sinir ağı açıklanmadan önce biyolojik sinir hücreleri hakkında genel bazı bilgiler sunulmuştur. Ardında YSA hücresi, YSA tarihçesi ve YSA ağlarına yer verilmiştir.

Ortalama bir beyinde 10 milyar kadar sinir hücresi vardır. Şekil 1.12'de biyolojik bir sinir hücresinin (neuron) yapısı ve bölümleri görülmektedir. Şekilde belirtildiği üzere sinir hücresi dendrit, hücre gövdesi, akson ve akson uçlarından (*sinaps*) oluşur. Dendritler sinaptik sinyalleri girdi olarak almakta, hücre gövdesi bu sinyalleri etkilerini göz önünde bulundurarak toplamakta ve üretilen denetim sinyali yada sinyalleri aksonlar aracılığı ile denetlenecek hedef hücrelere iletilmektedir. Tipik bir sinir hücresi, hücre gövdesi ve dendritleri üzerine dış kaynaklardan gelen elektrik darbelerinden (kimyasal etki ile oluşturulan mV mertebesindeki sinyaller) üç şekilde etkilenir. Gelen darbelerden bazıları nöronu uyarır, bazıları bastırır, geri kalanı da davranışında değişikliğe yol açar. Nöron yeterince uyarıldığında bu etki akson üzerinden sinaps boşluklarına iletilir. Aksonun sinaps boşluklarına açılan yaklaşık 10^4 civarında çıkıntısı vardır. Dolayısıyla bilgi 10^4 bit genişliğindeki paralel bir veri yolunu kullanarak beyne ulaşır. Fakat çoğu zaman bu yolun tamamı aynı anda kullanılmaz.



Şekil 1.12 Biyolojik sinir hücresi ve bölümleri

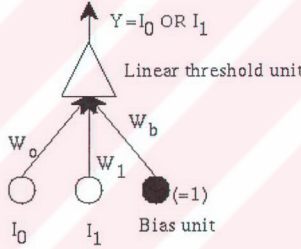
Çizelge 1.4’de beyin ile bilgisayar arasındaki işlevsel farklılıklar yer almaktadır.

Çizelge 1.4 Bilgisayar ve Beynin İşlevsel Farklılıkları

Beynin İşleyiş Biçimi	Bilgisayarın İşleyiş Biçimi
Bilgi analog olarak işlenir	Bütün işlemler sayısaldır
Veriler sinir hücrelerinden paralel olarak çevre hücrelere iletilir (10000 bitlik gruplar).	Bilgi aktarımı seridir. Paralel olduğu yerlerde bile 8 bitlik gruplardan oluşur.
Veri aktarım hızı yavaş ama etkisi büyüktür.	Hızlı veri aktarımları söz konusudur.
Eksik veya bozuk bilgi anımsanabilir.	Komut seti tam olarak tanımlanmış olmalıdır.
Giriş verilerine çok duyarlı değildir.	Giriş verilerine çok duyarlı olarak çalışır.

YSA Hücresi

YSA hücresinde de benzer bir yapı oluşturulmuştur. Şekil 1.13’de görülen bu yapıda I_0 ve I_1 giriş bilgilerini oluşturur. W_0 ve W_1 her bir girişin etkisini ayarlamak için kullanılan ağırlık katsayılarıdır. Çıkış bilgisi ise bir karar devresinden geçirildikten sonra elde edilir.



Şekil 1.13 YSA hücresi

Karar devresine gelen giriş bilgisi;

$$X = W_0 \cdot I_0 + W_1 \cdot I_1 \quad (1-2)$$

olarak tanımlanır. Çıkış bilgisi ise karar devresinin transfer fonksiyonuna bağlı olarak oluşur. (Schalkoff, 1997)

YSA Tarihçesi

Beyin ile ilgili çalışmalar YSA’nın temelini oluşturur. Bu çalışmalar Aristo zamanlarına (M.Ö. 4. yy) dayanır. İlk bilimsel çalışmalar 1940’larda gerçekleştirilmiştir. İlk YSA hücre modeli ise 1943’de Mc Culloch Pitts tarafından tasarlanmıştır. Bundan sonraki gelişmeler kronolojik sırasıyla şöyledir:

1949: Biyolojik sinaps göz önüne alınarak Hebbian öğrenme sistemi geliştirilmiştir.

1950: İlk nörobilgisayar tasarlanmıştır.

1954: İlk nörobilgisayar test edilmiştir.

1956: Taylor tarafından çağrışimli bellek yapıları ortaya konmuştur.

1958: Rosenblatt perceptron isminde çok girişli bir YSA hücresi kullanmıştır. Bu hücre tek katmanlı olup lineer bir ayırım gerçekleştirebilecek yapıdadır.

1960: Widrow-Hoff öğrenme kuralı açıklanmıştır.

1961: Taylor'un ortaya attığı çağrışım yapıları Steinbunch tarafından geliştirilmiştir.

1969: Minsky-Papert tek katmanlı yapıdan çok katmanlı yapıya geçişte öncülük etmişlerdir.

1972: Sun Khi Amari öğrenme işareti içeren bir YSA hücresi kullanmıştır.

1973: Vander Malsberg kendi kendini düzenleyen sistemler üzerinde çalışmalar yapmıştır.

1974: Grossberg ve Carpenter yarışmalı öğrenme yöntemini içeren ağlar geliştirmişlerdir.

1974-82: Kohonen eğiticişiz öğrenme üzerine çalışmalar yapmıştır. (SOM ve SOFM ağları)

1982: Hopfield enerji fonksiyonu (karar devresi, aktivasyon fonksiyonu) kavramını ortaya atmıştır.

1986: McClelland-Rumelhart Geriye yayımlı öğrenme yöntemini oluşturmuşlardır.

Günümüzde bu çalışmalara çok ilgi vardır. Her geçen gün farklı bir ağ modeli bu yöntemler çeşitli uygulamalarda denenmektedir.

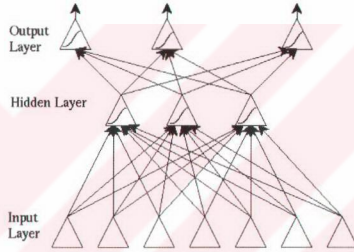
YSA'nın Özellikleri

Genel itibariyle YSA yöntemleri eğitilebilirlik, uyum sağlama ve kendi kendine organize olma gibi etkili özelliklere sahiptir. Aşağıda sıkça kullanılan YSA yöntemlerine değinilmiştir.

Radial Basis Function (RBF): Eksensel (radial) olarak simetrik fonksiyonların lineer kombinasyonlarını içeren ve daha çok fonksiyon yaklaştırma uygulamalarında kullanılan bir algoritmadır. Kullanımı oldukça kolay ve esnek. Yapı olarak giriş katmanından saklı katmana doğru lineer olmayan işlem yapabilir ama gizli katmandan çıkış katmanına doğru yaptığı işlemler lineerdir.

Generalized Regression Neural Network (GRNN): Yeni geliştirilen algoritmalarından biridir. MLP gibi iter aktif bir eğitim gerektirmez. Bu sebeple çok hızlı sonuca ulaşabilir. Eğitime datasına bağlı olarak girişle çıkış arasında isteğe göre rasgele bir fonksiyon kullanılır. Eğitim datasının boyutu arttıkça hata sıfıra doğru azalma gösterir. GRNN, standart regresyon tekniklerinde olduğu gibi sürekli değişken verilerin tahmini için yada sınıflandırma işlemleri için kullanılabilir.

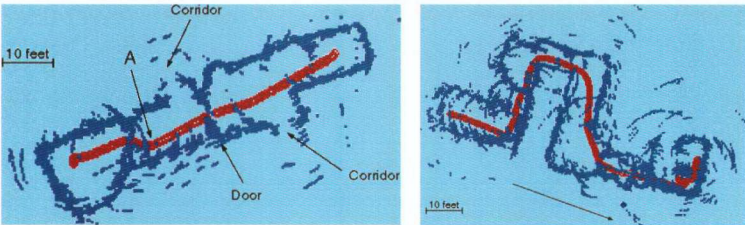
Multi Layer Perceptron (MLP): Lineer dönüşümleri içeren ama doğrusal olmayan aktivasyon fonksiyonlarıyla da kullanılabilen çok katmanlı bir algoritmadır. Yapı olarak giriş katmanı, saklı katmanlar ve çıkış katmanından oluşur. Şekil 1.14’de MLP yapısına bir örnek verilmiştir.



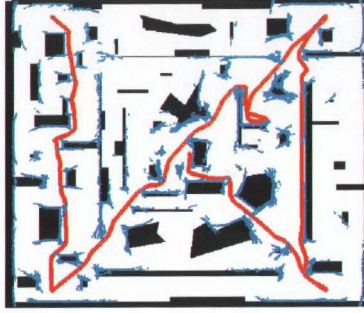
Şekil 1.14 MLP ağ yapısı örneği

Probabilistic Neural Network (PNN): PNN bir Bayes-Parzen sınıflandırmasıdır. 1960’larda geliştirilmiş olmasına rağmen geçen on yıla kadar pek sık kullanılmamıştır. Bayes-Parzen sınıflandırıcı, yapılacak işlemi kolay çözülebilecek birçok basit parçaya böler. Bundan sonra çok katmanlı bir ağ yapısında bu parçalar paralel olarak işlenir. Rasgele değişkenlerin olasılık yoğunluk fonksiyonunun tahmini PNN’in temelini oluşturur.

YSA ile gerçekleştirilen bazı uygulamalara örnekler Şekil 1.15 ve Şekil 1.16’da görülmektedir.



Şekil 1.15 Gezgin robot için YSA ile koridorda yön bulma



Şekil 1.16 Gezgin robot için YSA ile labirentte yön bulma

1.4.2 Klasik Algoritmalar

Klasik algoritmalarda bir referans noktası söz konusudur. Bu referans noktası ve hedef göz önünde bulundurularak bir akış diyagramı oluşturulur ve program hazırlanır.

Kesin Koordinatlar: Genel yön bulmada (global navigation) daha sık kullanılır. Yön bulma sahasındaki bütün noktalar için aynı olan bir sabit nokta referans olarak seçilir. Örn. GPS dünyadaki bütün alanlar için referans olarak tek bir nokta kullanır.

Göreceli Koordinatlar: Yerel yön bulmada (local navigation) robotun bulunduğu ortamda referans olarak bir nokta seçilebileceği gibi bazı uygulamalarda robotun geçmişteki kendi pozisyonu da referans olarak alınmaktadır.

Kişisel yön bulmada (personal navigation) ise referans daima robotun kendisidir.

1.5 Haberleşme Teknikleri

Bu çalışmada kullanılan haberleşme yöntemi RS232 standardındaki Seri Haberleşmedir. Bu nedenle RS232 haberleşme yöntemi detaylı olarak açıklanacaktır. Diğer yöntemler hakkında kısa bilgiler sunulacaktır.

1.5.1 Seri (RS232) Haberleşme

Seri port küçük çaplı robot uygulamalarında daha çok tercih edilen bir haberleşme yöntemidir.

Bunun nedenleri aşağıda sıralanmıştır:

- Seri kablolar paralel kablolarla göre daha uzun olur. Seri Port sayısal '1' seviyesini -3 ile -25 volt ve sayısal '0' seviyesini +3 ile +25 volt arasında iletir. Bu nedenle, seri

port, 50V maksimum voltaj değişim aralığına sahiptir. Böylece kabloda oluşan kayıp çok önemli boyutlara ulaşmaz.

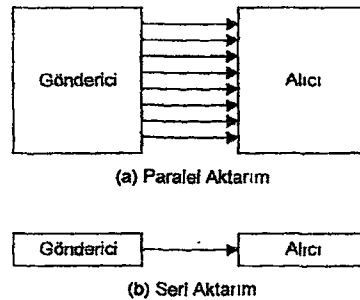
- Seri iletişimde kullanılan kablodaki damar sayısı azdır. Eğer bir robot, bilgisayardan uzak bir yerde çalıştırılacak ise, bu cihaza giden kablo sadece 3 damarlıdır ve bu durum maliyeti azaltır.
- Seri haberleşmede aynı anda iki yönlü veri iletimi gerçekleştirilebilir.
- Günümüzde, tüm devresi üzerinde dış dünya ile iletişimde kullanılan, seri haberleşme arabirimi bulunan mikro denetleyiciler çok yaygınlaşmıştır. Seri haberleşme uçları, mikro denetleyicilerin tüm devre uç sayısını azaltmaktadır. Genelde, TxD (*Transmit Data*) ve RxD (*Receive Data*) olarak adlandırılan 2 uç kullanılmaktadır. Bu durum donanımın robot tarafındaki kısmın tasarımını kolaylaştırmaktadır.

Seri Haberleşmenin Temelleri

Bir mikrodenetleyici, dış dünya (hafıza ve I/O birimleri) ile genelde 8-bit'lik parçalarla (8, 16, 32 ve 64 gibi) haberleşir. Şekil 1.17 (a)'da görüldüğü gibi, bu şekilde yapılan veri aktarımı *paralel* veri aktarımı (*transfer*) olarak adlandırılır. Bazı durumlarda, örneğin PC'nin yazıcı ile haberleşmesinde, veri yolundan 8-bit veri ile paralel haberleşme kullanılarak yapılır.

Eğer mesafe uzunsa paralel veri aktarımı pek uygun değildir. Ayrıca, 8-bit veri yolu kullanılması pahalıdır. Bu gibi durumlarda *seri* haberleşme daha uygun olur. Şekil 1.17 (b)'de görüldüğü gibi yapılan bir veri aktarımı, *seri* veri aktarımı olarak adlandırılır. Tek bir veri hattının kullanıldığı bu tür haberleşmenin ucuz olmasının yanında, iki farklı şehirde bulunan iki bilgisayarın telefon hattı üzerinden, bu yöntemi kullanarak, haberleşmesi de mümkün olur.

Seri haberleşmede, gönderici kısmında 8-bit veri, paralelden seriye çevrilir ve daha sonra tek bir hattın karşısına gönderilir. Alıcı, seri veriyi paralele çevirerek 8-bit veriyi oluşturur. Eğer veri telefon hattından iletilecekse, 0 ve 1'ler ses tonlarına dönüştürülür. Bu dönüştürme işlemi *modem* (*MOdulator/DEModulator*) olarak adlandırılan bir cihaz tarafından yapılır.



Şekil 1.17 Seri ve Paralel veri aktarımları

Eğer mesafe kısa ise, sayısal sinyal modülasyonuna gerek duyulmadan, basit bir hat üzerinden

veri iletilir. IBM PC'de klavye ile ana kart haberleşmesi bu şekilde yapılır. Telefon gibi, uzun mesafe veri transferleri, haberleşme hatlarını kullanır. Bu tür haberleşmede, göndericide, 1 ve 0'lar modülatör ile ses tonlarına çevrilir; alıcıda ise, bu ses tonları demodülatör ile 0 ve 1'lere dönüştürülür. (Gümüşkaya, 1999)

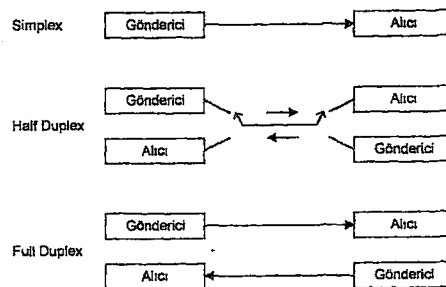
Senkron ve Asenkron Haberleşme

Seri veri Haberleşmesinde *senkron* ve *asenkron* olmak üzere iki yöntem kullanılır. Senkron haberleşmede, bir anda bir veri bloğu (karakterler dizisi) aktarımı yapılırken, asenkron haberleşmede bir anda sadece bir karakter iletilir.

Her iki haberleşme yöntemi yazılım ile gerçekleştirilebilir. Bununla beraber, bu şekildeki haberleşmede programlar uzun ve zor olacağı için, genelde seri veri iletiminde özel tüm devreler kullanılır. Bu tüm devreler yaygın şekilde, UART (*Universal Synchronous Receiver Transmitter*) veya USART (*Universal Synchronous Asynchronous Receiver Transmitter*) olarak adlandırılır. IBM PC'de kullanılan COM port'u, daha ileriki bölümlerde detaylı olarak sunulacak olan, 8250 UART'ını veya günümüzde daha gelişmiş sürümlerini kullanmaktadır. 8251 ise bir USART tüm devresidir.

Seri Veri Aktarım Kanalları

Seri veri iletimi tek yönde oluyorsa, PC'den yazıcıya olduğu gibi, bu veri iletimi *simplex* olarak adlandırılır. Veri, hem gönderilip hem alınabiliyorsa bu yöntem *duplex* denir. Bu iletişimde eğer veri bir anda sadece bir yönde aktarılabilirse *half duplex*, aynı anda her iki yönde aktarılıyorsa *full duplex* olarak adlandırılır. Full duplex haberleşmede, bir hat, göndermek ve bir hat da almak için olmak üzere, toplam iki tane hat kullanılır. Seri veri aktarımındaki kullanılan kanal durumuna göre yapılan haberleşme çeşitleri Şekil 1.18'dir.



Şekil 1.18 Seri veri aktarım yöntemleri

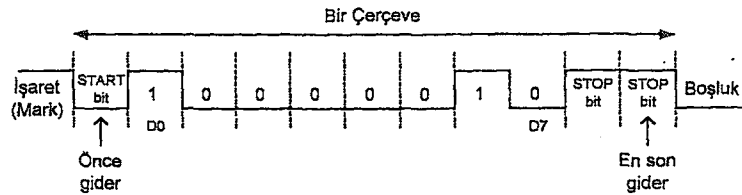
Asenkron Seri Haberleşme

Seri veri haberleşmede, veri gönderen ve alan uçların belli kurallara göre haberleşmesi gerekir. *Protokol* olarak adlandırılan bu kurallar, verinin nasıl paketleneyeceği, bir karakterdeki bit sayısı ve verinin ne zaman başlayıp biteceği gibi bilgileri belirler.

Asenkron seri veri haberleşmesi, karakter-tabanlı iletişimlerde yaygın olarak kullanılır. Buna karşın blok-tabanlı veri aktarımları, senkron yöntemi kullanır. Asenkron yöntemde, her gönderilecek karakter, başlama (START) ve bitme (STOP) bit'lerinin arasına yerleştirilir. Bu işlem çerçeveleme (*framing*) olarak adlandırılır. Asenkron haberleşmedeki veri çerçeveleme işleminde, ASCII karakterler bir START ile bir STOP bit'i arasında paketlenir. START bit'i her zaman bir bit'tir, fakat STOP bit'i bir veya iki bit'tir. Şekil 1.5'te görüldüğü gibi, START bit her zaman sayısal 0 (düşük), STOP bit sayısal 1'dir (yüksek). Verilen örnekte ASCII 'A', ikili olarak 01000001, START bit ile 2 STOP bit arasında çerçevenmiştir. Seri iletişimde önce en düşük değerli bit D0 (LSB) dışarı gönderilir.

Şekil 1.19'da solda görüldüğü gibi, seri veri aktarımının yapılmadığı durumda, sinyal seviyesi 1'dir (yüksek) ve bu durum *işaret (mark)* olarak adlandırılır. Sayısal 0 (düşük) ise *boşluktur (space)*. Veri iletimi bir START bit'i ile başlar ve bunu takiben D0 (LSB) daha sonra D7'ye (MSB) kadar geri kalan bit'ler ve son olarak 'A' karakterinin sonunu belirten 2 STOP bit'i gelir. (Gümüşkaya, 1999)

Asenkron seri haberleşmede kullanılan tüm devreler ve modemler 5, 6, 7 ve 8 bit veri uzunlukları için programlanabilir. Veri uzunluğuna ek olarak, 1 veya 2 tane STOP bit'i kullanılır. Eski sistemlerde ASCII karakterler 7-bit idi. Yeni uzatılmış ASCII karakterler yüzünden, bir ASCII karakter için 8-bit gerekir. Bazı ASCII olmayan klavyeler 5 veya 6 bit karakterleri kullanır. Bazı eski sistemlerde, veri alan cihazın yavaşlığından dolayı, cihaza, kendini ayarlamaya yeterli zaman vermek için 2 STOP bit kullanıldı. Günümüzde modern PC'lerde genellikle 1 STOP bit kullanılır.



Şekil 1.19 ASCII 'A' (41H) karakterinin çerçevesi

Bir seri veri haberleşmesinde, iletilen verinin dışında fazladan kullanılan bit'ler bir fazladan zaman (*overhead*) ve yük oluşturur. Örneğin, ASCII karakterin iletiminde, eğer 2 STOP bit'i kullanılıyorsa, her bir 8-bit ASCII kodu için toplam 11 bit iletilir. Yani her 8-bit için, iletişim hattında fazladan 3 bit veya %30 ek yük bulunur.

Bazı durumlarda, veri bütünlüğünü korumak için karakterin eşlik (*parity*) bit'i veri çerçevesine eklenir. Yani her karakter için (sisteme bağlı olarak 7-bit veya 8-bit) START ve STOP bit'lerine ek olarak bir tek eşlik bit'i eklenir. Eşlik bit'i tek (*odd*) veya çift (*even*) olur. Tek-eşlik bit durumunda, eşlik bit'i dahil, veri bit'lerinin sayısı bir tek sayıdır. Benzeri şekilde, çift-eşlik bit durumunda, eşlik bit'i dahil veri bit'lerinin sayısı bir çift sayıdır. Örneğin, ASCII 'A' karakteri ikili 0100 0001 olarak 0 çift-eşlik bit'ine sahiptir. UART tüm devreleri, daha sonra görüleceği gibi tek, çift veya eşiksiz (*no-parity*) olarak programlanabilir. Eğer bir sistem eşlik bit'i gerektiriyorsa, eşlik bit'i verideki en değerli bit'ten (MSB) sonra gönderilir. Bunu STOP bit'i takip eder. (Gümüşkaya, 1999)

Veri Aktarım Hızı

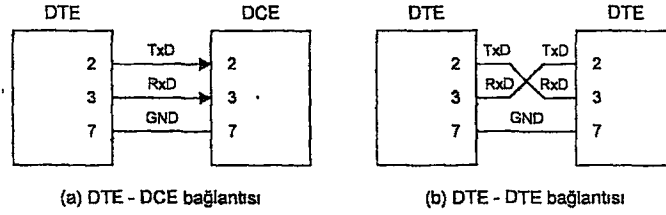
Seri veri haberleşmesinde veri aktarım hızı *saniyedeki bit sayısı (bps-bit per second)* olarak belirtilir. Veri aktarım hızını belirtmede diğer yaygın olarak kullanılan İngilizce terim *baud rate* 'dir. Bununla beraber, bu iki ifadenin birbirine eşit olması gerekmez. Çünkü, *baud rate* bir modem terminolojisidir ve saniyede sinyaldeki değişim sayısı olarak tanımlanır. Modemlerde bazı durumlarda bir sinyal değişimi ile birçok veri bit'i transfer edilir. İletişim teli düşünüldüğünde, bps ve *baud rate* aynıdır. Bu yüzden kitapta *bps* ve *baud rate* karşılıklı değişmeli olarak kullanılmaktadır.

Bir bilgisayarın seri veri transfer hızı haberleşme port'larına bağlıdır. Örneğin, eski IBM PC/XT 100 ile 9600 bps hızlarında veri transfer edebilmekteydi. Bununla beraber, yeni PC'ler 19200 bps gibi yüksek hızlara çıkabilmektedir. Asenkron veri haberleşmesinde, baud rate genellikle 100 000 bps ile sınırlıdır.

Veri Haberleşme Sınıfları

Seri haberleşmeyi kullanan cihazlar iki sınıfa ayrılmaktadır. Bunlar *DTE (Data Terminal Equipment)* ve *DCE (Data Communication Equipment)* olarak adlandırılır. DTE, bilgisayar veya terminal gibi veri gönderen veya alan cihazlardır. Buna karşın DCE, modem ve yazıcı gibi veri aktaran cihazlardır. Daha sonraki bölümde Tablo 19.1'de gösterilen sinyaller DTE yönünden tanımlanmıştır.

DTE ve DCE arasındaki en basit bağlantı, Şekil 1.20 (a)'da görüldüğü gibi, en az üç tane uç, TxD, RxD ve toprak gerektirir. İki PC gibi, iki tane DTE cihazı arasındaki minimum bağlantıda, 2 ve 3 numaralı uçlar, Şekil 1.20 (b)'deki gibi, karşılıklı çapraz bağlanır.



Şekil 1.20 DTE-DCE ve DTE-DTE bağlantıları (Gümüşkaya, 1999)

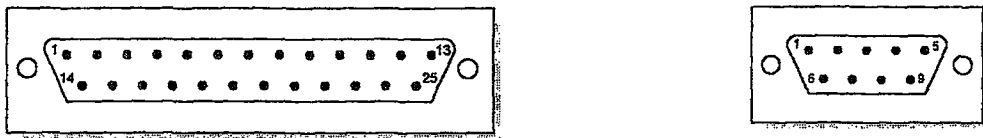
RS232 Standardı

Değişik üreticiler tarafından yapılmış veri haberleşme cihazlarının uyumluluğunu sağlamak amacıyla, EIA (*Electronics Industries Association*) tarafından RS232 olarak adlandırılan standart 1960 yılında belirlendi. 1963 yılında bu ilk standart değiştirildi ve RS232A olarak adlandırıldı. Daha sonraları 1965'te RS232B ve 1969'da RS232C standartları ilan edildi. Kitapta bu standart için kısaca RS232 kullanılacaktır.

Günümüzde RS232 en yaygın kullanılan seri I/O arabirim standardıdır. Bu standart TTL sayısal ailesinden çok önceleri belirlendiği için, giriş ve çıkış voltaj seviyeleri TTL uyumlu değildir. RS232'de sayısal '1' -3V ile -25V arasında, sayısal '0' +3V ile +25V arasında tanımlanır. -3V ile +3V arası tanımsızdır. Bu yüzden herhangi RS232 cihazını, bir mikrogenetleyici-tabanlı sisteme bağlamak için, MC1488, MC1489, TSC232 veya ST232 gibi voltaj çeviriciler kullanılır. Bu tüm devreler hat sürücüler/alıcıları (*line driver/receiver*) olarak adlandırılır.

RS232 Sinyalleri

RS232 için, Şekil 1.21'de görülen, bir DB-25 konnektörü ile erişilen toplam 25 uç tanımlanmıştır. Bu uçların hepsinde sinyal bulunmaz. Modern bilgisayarlarda bütün 25 uca gerek olmadığı için, IBM, yine Şekil 1.21'de görülen, DB-9 seri I/O standardını tanımlamıştır. Çizelge 1.5'de IBM PC 9-uçlu ve 25-uçlu konnektör'de bulunan RS232 sinyalleri verilmiştir.



Şekil 1.21 RS232 DB-25P erkek konnektör

Çizelge 1.5. IBM PC 9-uçlu ve 25-uçlu konnektör

Uç No(9-uçlu)	Uç No(25-uçlu)	Sinyal Adı
1	8	Data Carrier Detect (DCD)
2	3	Receive Data (RxD)
3	2	Transmit Data (TxD)
4	20	Data Terminal Ready (DTR)
5	7	Signal Ground (GND)
6	6	Data Set Ready (DSR)
7	4	Request To Send (RTS)
8	5	Clear To Send (CTS)
9	22	Ring Indicator (RI)

İki cihaz arasında güvenli veri iletişiminin olabilmesi için, veri aktarımının düzenlenmesi gerekir. Bazı amaçla seri veri haberleşmesinde senkronizasyon sinyalleri kullanılır. Bu sinyaller çok yaygın olarak kullanıldıkları için, bilinmeleri, herhangi bir UART tüm devresi veya modem cihazının çalışmasını anlamada çok önemlidir. Bu senkronizasyon sinyalleri aşağıda tanıtılmaktadır.

1. DTR (Data Terminal Ready): DTE (PC COM port'u) çıkış, modem giriş ucudur. Bu hat, modem'e, terminalin (bir PC COM port'u, dolayısıyla UART) veri iletişimi için hazır olduğunu belirtir. COM port'unda bir problem var ise, bu sinyal aktif yapılmaz.
2. DSR (Data Set Ready): DTE (PC COM port'u) giriş, modem çıkış ucudur. Bu hat, terminale (bir PC COM port'u, UART), modem'in veri iletişimi için hazır olduğunu belirtir. Eğer modem herhangi bir nedenden dolayı telefon hattına bağlanamıyorsa, bu sinyal pasif yapılarak, veri göndermek veya almak için modem hazır olmadığı PC'ye belirtilir.
3. RTS (Request To Send): DTE (PC gibi) göndereceği verisi olduğunda, bu uçla modeme haber verir.
4. CTS (Clear To Send): RTS sinyaline cevap olarak, modem alacağı veri için yeri olduğunda, bu sinyali DTE'ye (PC UART'ına) göndererek veri almaya hazır olduğunu belirtir.
5. DCD (Data Carrier Detect): Modem, telefon hattından bir taşıyıcıyı (*carrier*) belirlediğinde, bu hattı aktif yaparak DTE'ye (PC) bu durumu haber verir.
6. RI (Ring indicator): Telefon çaldığında, modem, DTE'ye (PC) bu sinyalle haber verir. Altı senkronizasyon sinyalinden en az kullanılabilecek bu sinyaldir. Çünkü telefona modem cevap verir. Eğer telefona PC'nin cevap vermesi isteniyorsa bu sinyal kullanılabilir.

Yukarıdaki tanımlamalardan sonra, PC ve modem haberleşmesi şu şekilde özetlenebilir: DTR ve DSR sinyalleri, PC ve modem tarafından çalışmakta ve hazır olma durumlarını belirtmede kullanılır. Buna karşın, RTS ve CTS veri akışını kontrol etmede kullanılır. PC veri göndermek istediğinde, RTS hattını aktif yapmakta, buna cevaben, modem veri kabul etmek için hazır ise (veri için yeri var ise), CTS sinyalini gönderir. Eğer yer yok ise, modem CTS'yi aktif yapmaz.

Bu durumda PC'nin tekrar aynı şekilde denemesi gerekir.

Diğer Standartlar

Bir RS232 kablosunun uzunluğu arttığında, sinyalde kapasitif yük artmaktadır. Bunun sonucu, yüksek veri hızları güvensiz olmaktadır. Eğer kablo uzunluğu 5 feet veya daha az olursa, RS232 ile 100 000 bsp veri aktarım hızı elde edilebilir. Veri hızını ve kablo uzunluğunu arttırmak için RS232'nin elektriksel özellikleri yeniden tanımlanmalıdır. Bunun sonucu, RS422 ve RS423 gibi yeni standartlar çıkmıştır. Çizelge 1.6, RS232 ile bu standartların karşılaştırmasını göstermektedir.

Çizelge 1.6 RS232'nin RS422 ve RS423 ile karşılaştırılması

	RS232	RS422	RS423
Maksimum Kablo Uzunluğu (ft)	50	4000	4000
Maksimum Hız (baud)	20K	10M / 40ft	100K / 30ft
		1 M / 100ft	10K / 300ft
		100K/ 4000ft	1K/ 4000ft

PC Seri Portu

Bir PC Seri Port'u, daha sonra anlatılacak olan, bir UART tüm devresi kullanır. DOS ve BIOS, 8250 gibi bir UART tüm devresinin detaylarından programcılar kurtararak, IBM PC'nin Seri Port'una kolay erişim sağlar. PC'deki INT 14 kesmesi seri veri iletişimde kullanılabilir. İleride Bölüm 19.6.2'de INT 14 kesmesi tanıtılacak ve bu kesme ile iki PC'yi haberleştiren bir Assembly program örneği verilecektir. DOS kesmesinin yanında, Seri Port'ların veri uzunluklarını belirleme ve hızlarını ayarlama gibi fonksiyonları, DOS'ta MODE komutuyla programlanabilir. (Gümüşkaya, 1999)

Seri Port Adresleri

Bir IBM PC, COM olarak belirtilen 4 taneye kadar Seri Port'a sahiptir. Bu port'lar 1, 2, 3 ve 4 (BIOS numaraları 0, 1, 2 ve 3) olarak numaralandırılmıştır. PC açıldığında POST (*Power-On Self-Test*) esnasında, 4 COM port'undan her biri için UART tüm devresi test edilir. Eğer COM port'ları bulunur ise, I/O port adresleri hafızada 0040:0000 ile 0040:0007 arasına yazılır. Çizelge 1.7'da COM adresleri görülmektedir. Her UART için atanan I/O adresi 16-bit olduğundan, UART başına 2 karakter gerekir.

Çizelge 1.7 PC Seri Port adres atamaları

COM Taban Adresi	İsim
0000:0400	COM1
0000:0402	COM2
0000:0404	COM3
0000:0406	COM4

Çizelge 1.8'de PC Seri Port standart I/O adres atamaları gösterilmektedir. Bu adresler birçok bilgisayar için geçerlidir.

Çizelge 1.8 PC Seri Port I/O adres atamaları

İsim	Adres	IRQ
COM1	3F8h	4
COM2	2F8h	3
COM3	3F8h	4
COM4	2F8h	3

Bir port'a atanmış olan adresi öğrenmek için değişik yöntemler bulunur. Örneğin, sistem ayarlarına bakılarak bu öğrenilebilir. Aşağıda DOS *debug* programı ile 0040:0000 adresine bakılarak COM atamaları öğrenilmektedir.

```
>debug
d 0040:0000 L8
0040:0000 F8 03 F8 02 00 00 00 00
```

Yukarıda verilen örnekte, COM1 3F8h ve COM2 2F8h adreslerine atanmış olup COM3 ve COM4'e bir atama yapılmamıştır. Aşağıda verilen örnek ile, Seri Port'lar için BIOS tarafından atanan adresleri C programı ile nasıl okuyabileceğimize dair bir örnek program EK1'de verilmiştir.

BIOS INT14h Kesmesi

PC'nin seri haberleşme port'ları, BIOS tabanlı INT 14H kesme komutu kullanılarak erişilebilir. INT 14 komutunun çeşitli seçenekleri AH saklayıcısı ile belirlenir. Bu komutu kullanarak iki PC birbirine veri gönderebilir. PC'ler arasında seri veri iletimi şu şekildedir.

- Bir karakter göndermek için, önce AH = 1 ve AL = karakter, yüklenir ve sonra INT 14 komutu yürütülür.
- Bir karakter almak için, önce AH = 3 yapılır ve sonra INT 14 komutu yürütülür. COM port'unun durumu AH saklayıcısında geri döner. Bu okunan COM port'unun durumu olup AL saklayıcısında bulunan modem'in durumu değildir. Daha sonra durum

port'unun DO bit'i (*received data ready*) kontrol edilir. Eğer bu bit yüksek ise, COM port'u üzerinden bir karakter alınmıştır ve UART içinde bulunmaktadır. Alınan karakteri okumak için AH = 2 yapılarak tekrar INT 14 komutu yürütülür. Sonuçta AL saklayıcısında veri bulunur.

İki PC arasında COM2 üzerinden, INT 14 komutu kullanılarak, seri veri haberleşmesi yapan bir program yine EK 1'de sunulmuştur. İki bilgisayarın seri bağlantısı Şekil 1.9'daki gibi yapılmalıdır. Program derlenip her iki makinede çalıştırıldıktan sonra, klavyeden girilen karakterler diğer makinenin ekranında görüntülenir.

1.5.2 Diğer Haberleşme Teknikleri

Paralel Haberleşme

Gezgin robotun bilgisayarla haberleşmesinde paralel port kullanımı çok nadir görülür. Bunun sebepleri şöyle sıralanabilir:

- Paralel portta sayısal '1' sinyali 5V, sayısal '0' sinyali 0V olarak belirlenmiştir. Bu gerilim seviyeleri, uzak mesafelere veri taşınmasını engeller.
- Bir kablodaki damar sayısı sekiz veri ucuna eklenen kontrol uçlarıyla birlikte kablo maliyetini oldukça arttırmaktadır.
- Robot tarafında sinyallerin alınması ve gönderilmesi için daha çok giriş/çıkış bağlantısı gerektireceği için uygulamada zorluklarla karşılaşılır.

Diğer taraftan bu haberleşme yönteminin hız bakımından avantajları olduğu söylenebilir. Ama günümüzde paralel porttan çok daha hızlı veri aktarımlarına imkan veren haberleşme portları birbiri ardına uygulamaya girmektedir.

RF (Radio Frequency) Haberleşme

Kablosuz bir haberleşme olmasından ve uzak mesafelere veri aktarımlarına olanak vermesinden dolayı Robot haberleşmelerinde en çok tercih edilen yöntemdir. Alıcı ve verici modüller halinde sisteme adapte edilir. Haberleşme frekansı olarak genellikle 433MHz seçilir. Verici den alıcıya olmak üzere tek yönlü bir haberleşme söz konusudur. Fakat hem alıcı hem de verici içeren "*trans receiver*" modülleri ile iki yönlü veri akışı sağlanabilir. Mikrodenetleyici açısından bakıldığında sadece bir veya iki giriş/çıkış ucu gereklidir. Haberleşme protokolü oluştururken darbe genişlikleri 300-500 μ sn olarak seçilebilir. Böylece çok daha sağlıklı ve hızlı bir haberleşme gerçekleştirilebilir.

RF haberleşmede en önemli konulardan biri güvenlidir. Verici modülden gönderilen sinyallerin herkese açık olması şifreleme işlemini ön plana çıkarmaktadır. Gönderilen verilerin başkaları tarafından öğrenilmesi istenmiyorsa protokol oluşturulurken kullanılacak

olan şifre özenle belirlenmelidir. Bunun yanında modüllerin üretimini yapan firmalar da birçok şifreleme seçeneği sunmaktadırlar. Ayrıca sadece şifreleme ve şifre çözme “*encode*” ve “*decode*” işlemleri için özel tüm devreler üreten firmalar da mevcuttur.

IR (Infra-RED) Haberleşme

IR haberleşme kablosuz gerçekleştirilen bir haberleşme yöntemidir. Bu haberleşme yönteminde bir çeşit Infra-RED sinyali kullanılır. Verici ve alıcı çiftleri birbirlerini görecektir şekilde yerleştirilir. Aralarındaki mesafe en fazla 5-6 metre kadardır. Ayrıca verici, belli bir açı üçgeninde (yaklaşık 60 derece) alıcıyı etkileyebilir. Vericiden gönderilen sinyal 30-40kHz aralığında darbeler ile modüle edilmiş bir sinyal olmalıdır.

Robot haberleşmelerinde bu yöntemi kullanmanın olumlu sonuçlar vermeyeceği ortadadır.

USB (Universal Serial Bus) Haberleşme

USB, seri haberleşmeyi hızlandırmak ve geliştirmek amacıyla yapılan çalışmaların bir sonucudur. RS232 standardındaki seri haberleşmeden çok daha hızlı çalışan özel bir seri haberleşme yöntemidir. Bilgisayarlardaki seri haberleşme portları yavaş yavaş yerini USB haberleşmeye bırakmaya başlamışlardır. Hatta günümüzde piyasaya sürülen diz üstü bilgisayarlar seri haberleşme adına sadece USB portları yer almaktadır.

Fakat mikro işlemci tüm devrelerinde modül olarak henüz yaygınlaşmamışlardır. Bu nedenle küçük çaplı robot uygulamalarında tasarımı hızlandırmak ve ucuza getirmek için seri haberleşme yöntemi olarak USB yerine RS232 standardı tercih edilmektedir.

Bluetooth Haberleşme

Özel bir kablosuz haberleşme yöntemidir. İlk ortaya atılmasındaki amaç PC çevre elemanlarının (yazıcı, fare, klavye, kamera vb...) bağlantısında kablo karmaşasını ortadan kaldırmaktır. Fakat ürünlere entegre edilmesi zaman alacak gibi görünüyor.

USB'ye benzer şekilde Bluetooth haberleşme modülleri de henüz mikrodenetleyici tüm devrelerinde entegre olarak yaygınlaşmamıştır.

Firewire Haberleşme

Çok hızlı veri aktarımlarında kullanılan kablolu bir haberleşme yöntemidir. Daha çok kamera sistemlerinde ve görüntü aktarımlarında tercih edilirler. Yüksek hızının verdiği avantajdan dolayı özel robot uygulamalarında kullanılan bir haberleşme tekniğidir.

GSM / GPRS Haberleşmeleri

Uydu haberleşmeleri veya ülkeler arası mesafelerde haberleşme ihtiyacı duyulan çok özel robot uygulamalarında tercih edilen haberleşme yöntemleridir. Bu uygulamalardan bazıları konum belirleme, uzay araştırma çalışmaları olarak belirtilebilir.



2. YSA ile YÖN BULMA ALGORİTMALARININ SİMULASYONU

YSA ile yön bulma algoritmaları ilk olarak MATLAB ortamında yazılarak simülasyon denemeleri gerçekleştirilmiştir.

Bu bölümde öncelikle MATLAB hakkında genel bazı bilgiler verilmiştir. Ardından MATLAB’da gerçekleştirilen algoritmalar hakkında detaylar anlatılmıştır.

2.1 MATLAB Hakkında

MATLAB; (**MAT**rix **LAB**oratory); ilk defa 1985’de C.B. Moler tarafından geliştirilmiş ve özellikle de matris esaslı matematik ortamında kullanılabilen etkileşimli bir paket programlama dili olarak tanımlanmıştır. Başlangıçta MATLAB özellikle mühendislik alanında, iyi grafik özelliklere sahip daha çok sayısal hesaplamalarda kullanılmak amacı ile geliştirilmiş bir paket programlama dili olarak ortaya çıkmıştır. MATLAB, orijinal olarak matris yolu ile matris hesaplamalarının öncüleri olarak bilinen LINPACK ve EISPACK projeleri yolu ile geliştirilen matris yazılım programlarına kolay erişim sağlamak amacı ile yazılmıştır. O zaman için özellikle FORTRAN dilinde uzun zaman alan programlama işlemlerine bir alternatif olarak ortaya çıkmıştır. İlk sürümleri FORTRAN diliyle yazılmış olmakla beraber son sürümleri C dilinde hazırlanmıştır. (Yüksel, 2000)

Bugün için farklı alanlarda kullanılabilen çok geniş bir ürün yelpazesine sahip MATLAB, teknik hesaplamalarda kullanılan yüksek başarılı bir dil olarak tanımlanmaktadır. MATLAB’ın belli başlı kullanım alanları:

- Matematik ve hesaplama işlemleri, algoritma geliştirme.
- Modelleme, benzetim ve prototip geliştirme.
- Verilen analizi, incelenmesi ve görüntülenmesi.
- Bilimsel ve mühendislik alanında grafik işlemleri.
- Grafikselle kullanıcı ara yüz yapısını da içine alan uygulama geliştirme.

MATLAB sistemi 5 ana kısımdan ibarettir.

MATLAB Dili: Bu şartlı deyimler, fonksiyonlar, veri yapıları, giriş / çıkış ve nesneye yönelik programlama özellikleri içeren yüksek seviyeli bir matris / dizim dilidir.

MATLAB Çalışma Ortamı: Bu, MATLAB kullanıcısı veya programcısı olarak çalışma imkanı sağlayan bir araç ve gereçler takımıdır. Çalışma ortamındaki değişkenleri kontrol etmeyi ve verileri çalışma ortamından başka bir ortama aktarmayı ve başka bir ortamdaki verileri çalışma ortamına aktarmayı kolaylaştırır. Ayrıca MATLAB uygulamalarının

geliştirilmesi, kontrol edilmesi, hatta bulunması ve M- dosyalarının biçimlendirilmesi için gerekli araçları da kapsar. Çalışma ortamı özellikleri “*general*” adı altında tek bir klasör içinde yer almaktadır.

MATLAB Grafik Sistemi: Bu, MATLAB ta Handle Graphics olarak bilinir. 2-boyutlu ve 3-boyutlu veri görüntülemesi, görüntü işleme, canlandırma (animation) ve grafik sunumu için gerekli yüksek seviyeli komutları kapsar. Ayrıca MATLAB uygulamaları için tamamen kullanıcı arabirimi (GUI) oluşturmak kadar, grafiklerin görünüşünü elle düzenlemeye olanak veren düşük seviye komutları da içerir. (Yüksel, 2000)

MATLAB Matematiksel Fonksiyon Kütüphanesi: Bu, *sum*, *sine*, *cosine* gibi elemansal fonksiyonlardan ve karmaşık aritmetikten, matris tersi, matris öz değerleri, *Bessel* fonksiyonları ve hızlı Fourier dönüşümlerine kadar değişen geniş bir hesaplama algoritmaları toplamıdır.

MATLAB Uygulama Programı Arabirimi (API): MATLAB ile etkileşimli çalışan C ve Fortran programları yazılmasına olanak tanıyan bir kütüphanedir. MATLAB’ tan rutinleri çağırma (dinamik ilişki), MATLAB’ ı bir hesaplama motoru gibi çağırma ve MAT (M-file) dosyalarını yazma ve okuma için gerekli gereçleri içerir. (Yüksel, 2000)

2.2 MATLAB’da YSA

MATLAB, YSA algoritmaları konusunda program geliştirmeye açık bir ortamdır. Yapısında bulunan “Neural Networks Toolbox” ile birçok YSA algoritmasını ve karar fonksiyonları (aktivasyon fonksiyonları) desteklemektedir. İstenildiğinde bu YSA algoritmaları kullanılarak simülasyonlarını da yine MATLAB ortamında yapmak mümkün olmaktadır. MATLAB’da kullanılabilecek YSA algoritmaları ve kısa açıklamaları Çizelge 2.1 ile aşağıda sunulmuştur.

2.3 YSA Programları

Mikrodenetleyici kartına yazılım hazırlamadan önce YSA kullanılarak MATLAB ortamında gösteri amaçlı programlar hazırlanmıştır. Her iki programda da GRNN algoritması seçilmiştir. Bunun nedeni GRNN algoritmasının iter aktif bir yapıya sahip olmamasından dolayı diğerlerine oranla daha hızlı sonuç vermesidir. Sözü geçen programlarla ilgili detaylar giriş bilgilerinin farklılıklarına göre bir sonraki konuda anlatılmıştır. (MarhWorks, 2002)

Çizelge 2.1 MATLAB 6.5'deki YSA algoritmaları

Komut	Açıklama
newc	Create a competitive layer.
newcf	Create a cascade-forward backpropagation network.
newelm	Create an Elman backpropagation network.
newff	Create a feed-forward backpropagation network.
newfftd	Create a feed-forward input-delay backprop network.
newgmnn	Design a generalized regression neural network.
newhop	Create a Hopfield recurrent network.
newlin	Create a linear layer.
newlind	Design a linear layer.
newlvq	Create a learning vector quantization network
newp	Create a perceptron.
newpnn	Design a probabilistic neural network.
newrb	Design a radial basis network.
newrbe	Design an exact radial basis network.
newsom	Create a self-organizing map.

Programın karar aşamasında YSA'nın GRNN algoritmasından yararlanılmıştır. Matrisel olarak noktaların siyah/beyaz değerlerinden oluşan giriş bilgilerine karşılık algoritma sonucunda yine matrisel bir çıkış elde edilir.

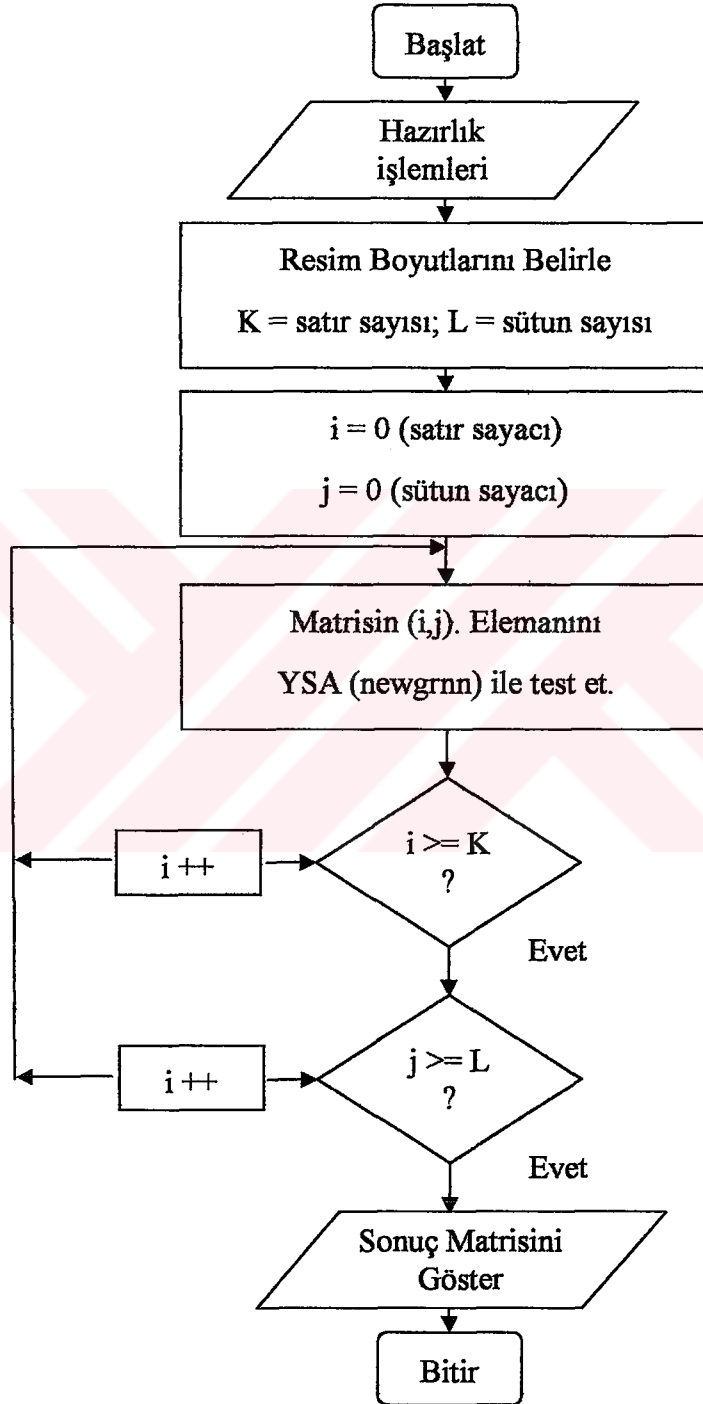
2.3.1 Resim Bilgisi için YSA Programı

Bu program ile farklı nokta boyutlarındaki resimler giriş verileri olarak kullanılıp algoritma sonucunda yönleri açıklayan matrisel çıkışlar üretilmektedir. Birinci aşamada eğitme işlemi gerçekleştirilir. Bunun için hazırlanmış olan "*learn_bw.m*" dosyasındaki program koduna ait algoritma Şekil 2.1'de gösterilmiştir.



Şekil 2.1 Sadece resim içeren Learn_bw.m programına ait algoritma

İkinci aşamada ise giriş bilgisi olarak alınan resim matrisleri birinci aşamada eğitilmiş olan YSA algoritmasından geçirilir. Bu aşamaya ait program kodu “*test_bw.m*” dosyasında yer almaktadır. Bu dosyadaki program kodunun algoritması Şekil 2.2 ile aşağıda verilmiştir.



Şekil 2.2 Sadece resim içeren Test_bw.m programına ait algoritma

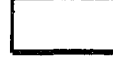
Yukarıda adı geçen “*learn_bw.m*” ve “*test_bw.m*” komut dosyaları EK 2 bölümünde vardır.



Şekil 2.3 1x2 nokta
siyah-siyah



Şekil 2.4 1x2 nokta
beyaz-siyah



Şekil 2.5 1x2 nokta
beyaz-beyaz



Şekil 2.6 1x2 nokta
siyah-beyaz



Şekil 2.7
2x2 nokta
beyaz-beyaz
beyaz-beyaz



Şekil 2.8
2x2 nokta
siyah-beyaz
beyaz-beyaz



Şekil 2.9
2x2 nokta
siyah - siyah
beyaz-beyaz



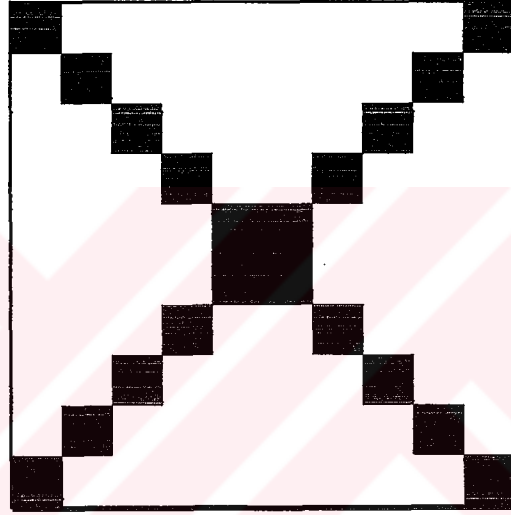
Şekil 2.10
2x2 nokta
siyah - siyah
siyah - beyaz



Şekil 2.11
2x2 nokta
siyah - siyah
siyah - siyah



Şekil 2.12
2x2 nokta
siyah -beyaz
beyaz- siyah



Şekil 2.13 10x10 nokta boyutunda görüntü

black_white.m

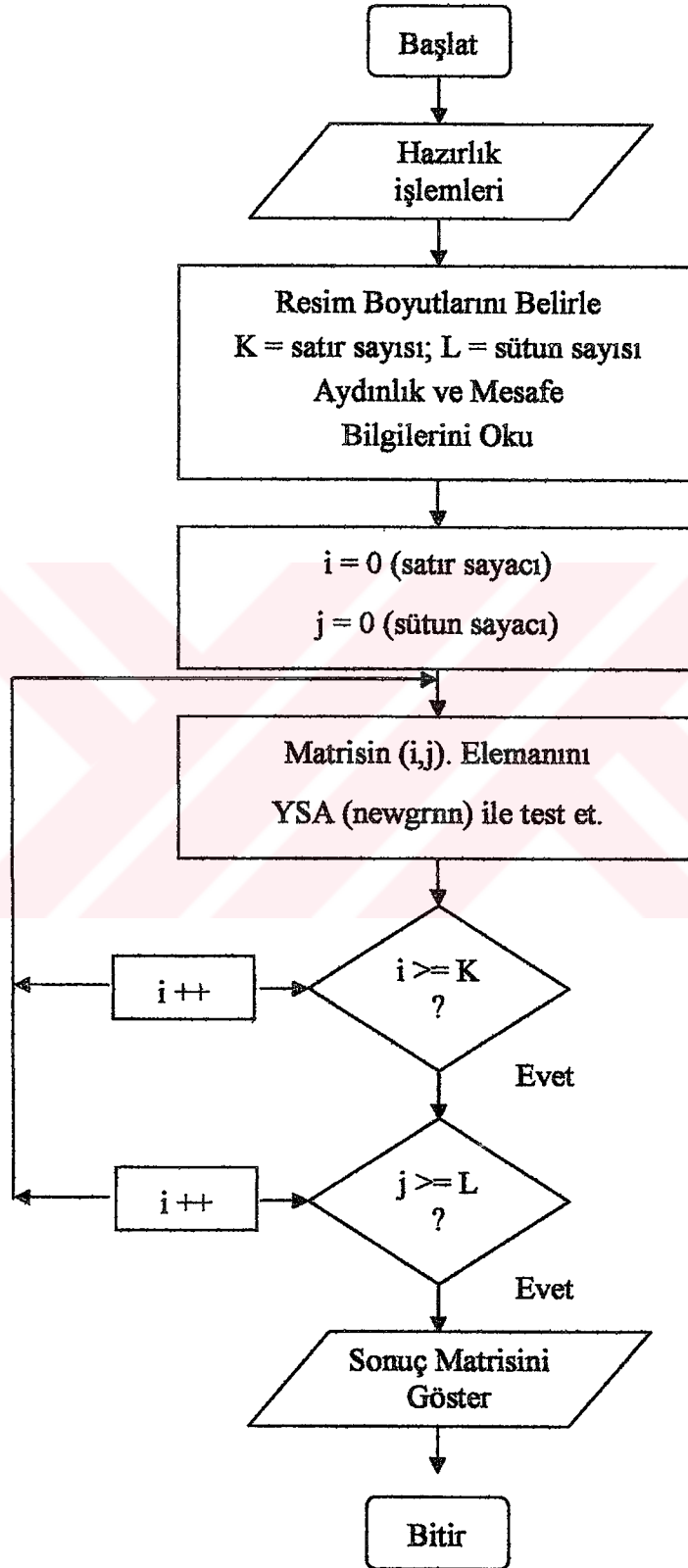
```

one = learn_bw;
test_bw('one_a.bmp',one);           %1x2 nokta boyutunda
test_bw('one_b.bmp',one);           %1x2 nokta boyutunda
test_bw('one_c.bmp',one);           %1x2 nokta boyutunda
test_bw('one_d.bmp',one);           %1x2 nokta boyutunda
test_bw('two_a.bmp',one);           %2x2 nokta boyutunda
test_bw('two_b.bmp',one);           %2x2 nokta boyutunda
test_bw('two_c.bmp',one);           %2x2 nokta boyutunda
test_bw('two_d.bmp',one);           %2x2 nokta boyutunda
test_bw('two_e.bmp',one);           %2x2 nokta boyutunda
test_bw('two_f.bmp',one);           %2x2 nokta boyutunda
test_bw('two_x.bmp',one);           %10x10 nokta boyutunda
%test_bw('two_z.bmp',one);          %320x240 nokta boyutunda

```

2.3.2 Resim, Aydınlık ve Mesafe Bilgileri için YSA Programı

Bu programda farklı nokta boyutlarındaki çekirdek resimler yanında ortamın aydınlık durumu ve mesafe bilgisi giriş verilerine eklenmiştir.



Şekil 2.14 Resim, Aydınlık ve Mesafe içeren Test_bw.m programına ait algoritma

Programın karar aşamasında yine YSA'nın GRNN algoritmasından yararlanılmıştır. Matrisel olarak noktaların siyah/beyaz değerleri, mesafe bilgileri ve ortamın aydınlık bilgilerinden oluşan giriş bilgilerine karşılık algoritma sonucunda yine matrisel bir çıkış elde edilir.

Aşağıdaki program YSA algoritmasının gömülü olduğu asıl programı çağıran bir toplu işlem dosyasıdır. Bu dosyada YSA algoritmanın nasıl kullanılacağına dair detaylı açıklamalar ile bir örnek yer almaktadır. Asıl programlar uzun oldukları için EK 2 bölümünde sunulmuştur.

Şekil 2.12 ve Şekil 2.13'deki resimler bu algoritmalarla denenmiştir. Alınan sonuçlar bölüm sonundaki Konu 2.4'de yorumlanmıştır.

using.m

```
% Neural Network (NN) karar algoritmasının (network) Eğitilmesi
%-----
% Kullanım:
% Network = learn_bw;
% Network => Neural Network karar algoritması
N = learn_bw;

% Eğitilmiş olan NN algoritması ile verilerin test edilmesi
%-----
% Kullanım:
% test_bw('dosya_adi', Network , Lighting, Distance);
% Network => Neural Network karar algoritması
% Lighting => Ortamın aydınlık olup olmadığı (aydınlık:1 karanlık:0)
% Distance => Engelle olan mesafe (Aralık:0-10, Yakın:D<5 Uzak:D>5)

IMG = imread('one_two.bmp');      %1x2 nokta boyutundaki resim
figure; imshow(IMG); shg;

L = 0;      %Karanlık
D = 3;      %Yakın
test_bw('one_two.bmp',N,L,D);
L = 1;      %Aydınlık
D = 3;      %Yakın
test_bw('one_two.bmp',N,L,D);
L = 1;      %Aydınlık
D = 7;      %Yakın
test_bw('one_two.bmp',N,L,D);
L = 1;      %Aydınlık
```

```
D = 7;      %Uzak
test_bw('one_two.bmp',N,L,D);
```

% Aynı komutla boyutu 1x2 noktadan büyük resimler de test edilebilir.

% örnek:

```
IMG = imread('four_four.bmp');      %4x4 nokta boyutundaki resim
figure; imshow(IMG); shg;
```

```
L = 1;      %Aydınlık
D = 4;      %Uzak
test_bw('four_four.bmp',N,L,D);
```



Şekil 2.15 1x2 nokta beyaz-siyah



Şekil 2.16 4x4 nokta LDR bilgisi

2.4 Sonuçlar

Gösteri amaçlı programlar hazırlanırken MATLAB'ın 6.5 versiyonu ile “Neural Network Toolbox”ın 4.0 versiyonu kullanılmıştır. Ayrıca testlerin yapıldığı bilgisayarın özellikleri de Çizelge 2.2’de yer almaktadır.

Çizelge 2.2 Kullanılan Bilgisayar Özellikleri

Celeron 1700MHZ
Gigabyte Tümleşik Ana kart
256 MB SDRAM
40 GB HDD
Onboard Ekran Kartı (16 MB)

Bu koşullar altında alınan sonuçlar %100 doğruluğa sahiptir. Karar algoritması çıkışında her bir noktaya ait yön bilgisi hatasız olarak tespit edilmiştir.

Alınan başarılı sonuçların ardından uygulaması gerçekleştirilen mikrodenetleyicili kartın programının YSA algoritmasına göre hazırlanmasına başlanmıştır.

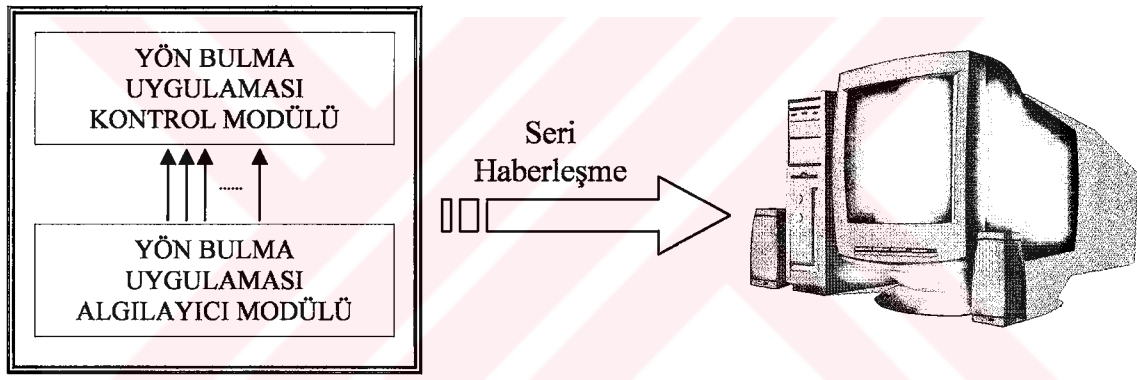
3. YÖN BULMA UYGULAMASININ DONANIM VE YAZILIMI

Bu bölümde önce uygulama donanımı hakkında detaylı bilgi aktarılmıştır. Ardından mikrodnetleyici ve bilgisayar yazılımlarından bahsedilmiştir. Son olarak da bilgisayar ile donanım arasında kullanılan haberleşme protokolünün detaylarına yer verilmiştir.

Uygulama projesi iki aşamada oluşturulmuştur. Birinci aşamada mikrodnetleyicili donanım tasarlanmış ve hayata geçirilmiştir. İkinci aşamada ise seri haberleşme portu üzerinden bilgisayarla bağlantı kurulmuş ve ardından bilgisayar ortamında görsel öğeler içeren bir gözlemleyici program yazılmıştır.

3.1 Yön Bulma Uygulamasının Donanım Sistemi

Yön bulma uygulamasına ait donanım sistemi genel bir blok diyagram halinde Şekil 3.1’de sunulmuştur.



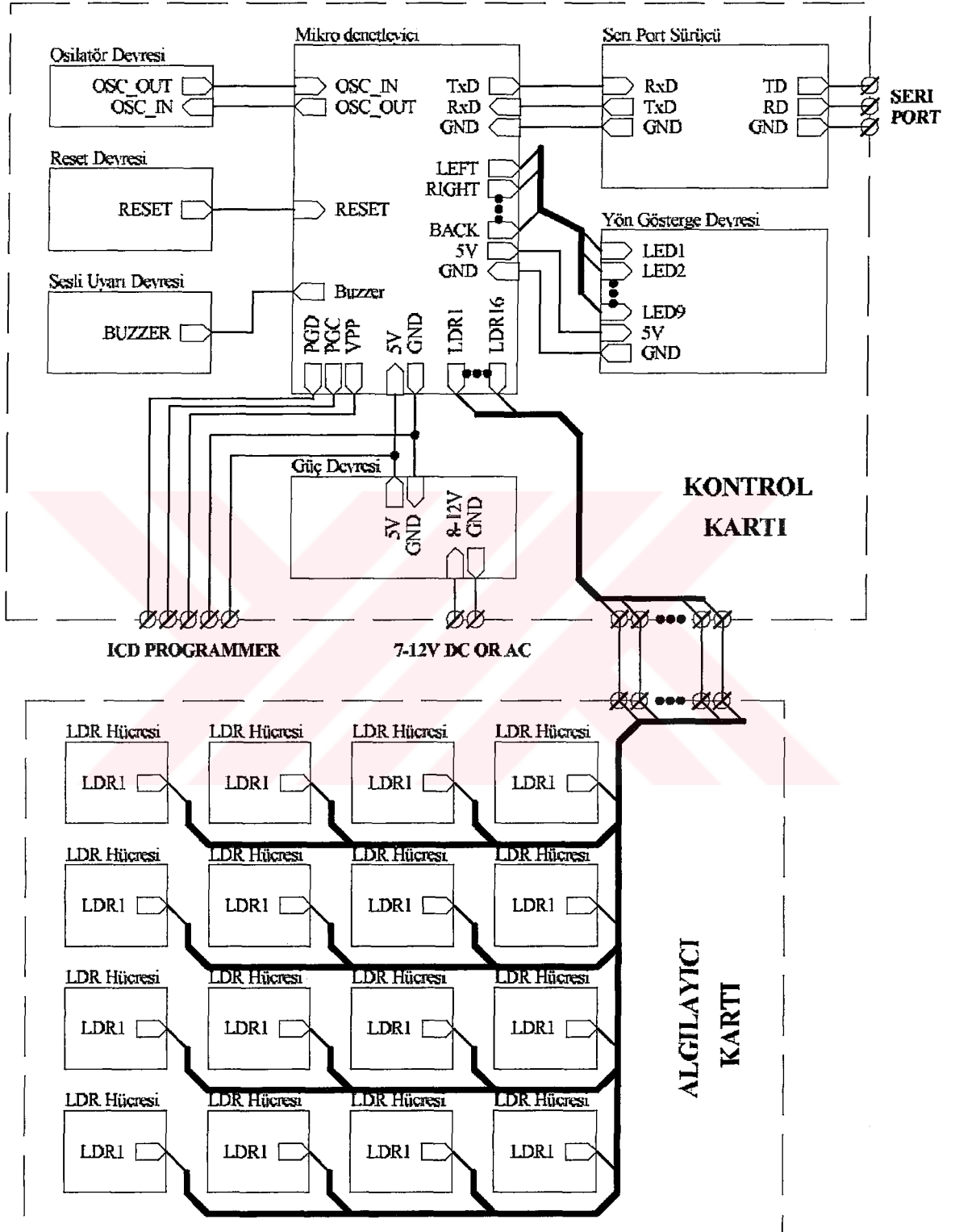
Şekil 3.1 Yön Bulma Uygulaması Donanım Sisteminin Genel Blok Diyagramı

Şekil 3.1’de gösterilen yön bulma uygulaması kontrol modülü iki ayrı elektronik karttan oluşmaktadır. Bunlardan ilki mikrodnetleyiciyi içinde barındıran kontrol kartıdır. Karar algoritmasına ait program ve haberleşme ile ilgili buffer devreleri de bu kartta yer almaktadır. İkinci kart ise ışığı algılamada kullanılan LDR’lerin bulunduğu algılayıcı kartıdır.

Bilgisayara bilgi aktarmak için kullanılan seri haberleşme RS-232 standardında gerçekleştirilmektedir. Bilgi akışı yön bulma sisteminden bilgisayara doğrudur.

Yön bulma sisteminde bilgisayar sadece gözlemleyici amacıyla yer almaktadır. Bütün algılama ve karar işlemleri mikrodnetleyici tarafından kontrol edilmektedir. Bilgisayarda hazırlanmış bulunan görsel bir program aracılığıyla algılayıcıların durumları ve mikrodnetleyicinin karar verdiği yön bilgisi bilgisayar ekranından gözlenebilmektedir.

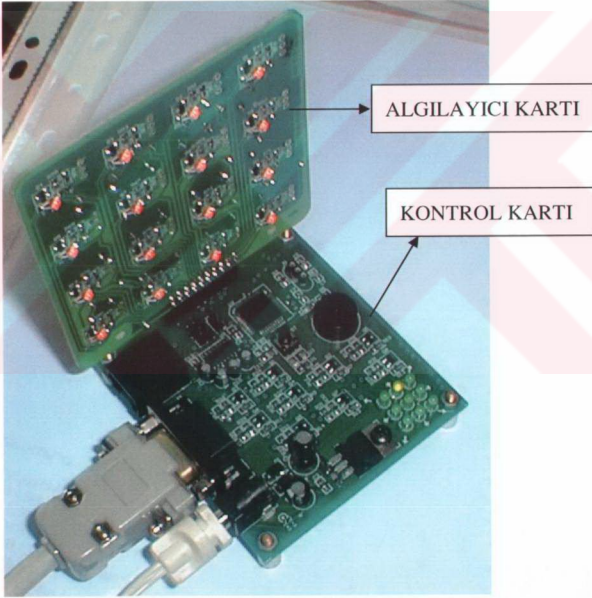
Yön bulma uygulaması kontrol modülünde yer alan elektronik kartların detayları Şekil 3.2’de verilmiştir.



Şekil 3.2 Yön Bulma Kontrol Modülünün detaylı blok diyagramı

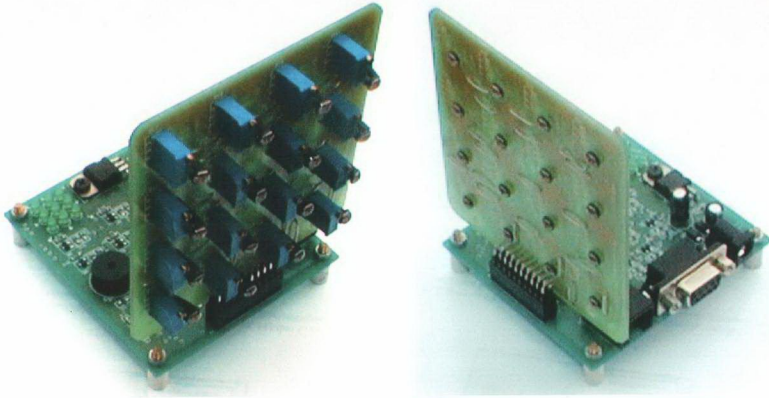
Şekil 3.2’de belirtilen algılayıcı kartı ve kontrol kartı birbirine iki sıralı (2x10 pin) bir konnektör ile bağlanmıştır. Konnektör üzerinde bulunan sinyaller algılayıcı kartı için gereken besleme sinyalleri ve algılayıcılardan toplanarak mikrodenetleyiciye aktarılan giriş sinyalleridir.

Şekil 3.3’de yön bulma kontrol modülünün fiziksel görüntüsü, modül çalışır durumdayken görülmektedir. Algılayıcı kartı ve kontrol kartı birbirine dik olarak monte edilmektedir. Böylece ışık kaynağı yön bulma kontrol modülünün karşında kalmaktadır ve böylece ışığın algılanması daha kolay olmaktadır. Çünkü algılayıcı kartında kullanılan ışığa bağımlı dirençler karşıdan gelen ışığa daha duyarlı tepki vermektedir.



Şekil 3.3 Yön Bulma Kontrol Modülünün çalışır haldeki görüntüsü

Şekil 3.4’de yön bulma kontrol modülünün fiziksel görüntüsü farklı açılardan iki görüntüsüne yer verilmiştir. Soldaki görüntüde algılayıcı kartındaki ışığa duyarlı dirençleri algılama noktalarını ayarlayabilmek için çok turlu potlar (mavi) yer almaktadır.

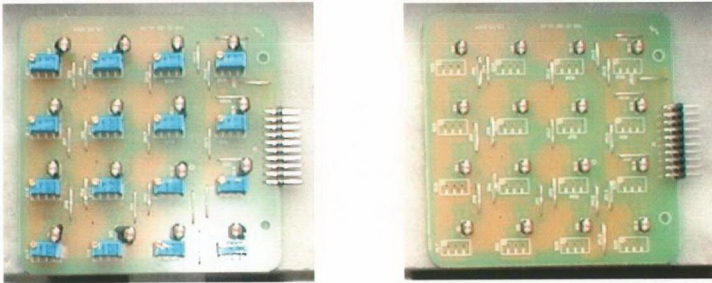


Şekil 3.4 Yön Bulma Kontrol Modülünün farklı açılardan iki görüntüsü

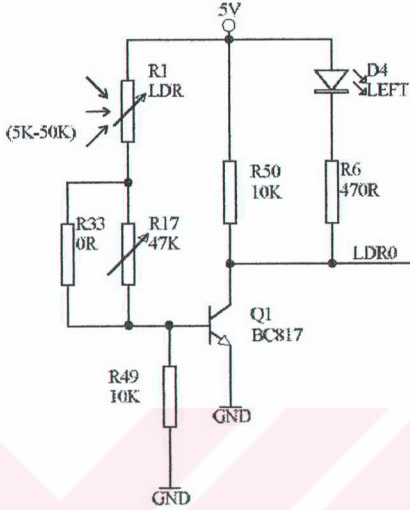
3.2 Algılayıcı Kartı

Algılayıcı kartı bu uygulama için kullanılmış olan tek algılayıcı devresidir. Kartın üzerinde matris şeklinde eşit aralıklarla yerleştirilmiş 16 adet (4x4) ışığa bağımlı dirençler (LDR) vardır. Bu kartın amacı 4x4 nokta boyundaki bir çekirdek görüntü bilgisi oluşturmak ve bunu mikrodenetleyiciye aktarmaktır.

Her bir LDR sinyali, transistorlu bir devre ile işlendikten sonra mikrodenetleyiciye aktarılır. Transistorların kolektörlerine eklenmiş olan yüzey montajlı LED diyotlar ile LDR'nin o anki ışığı algılama durumu gözlemlenebilmektedir. Şekil 3.5'de farklı yüksekliklerde monte edilmiş LDR'lerden oluşan algılayıcı kartlarının fiziksel görüntüsü verilmiştir.



Şekil 3.5 Farklı yüksekliklerde monte edilmiş algılayıcı kartları



Şekil 3.6 Her bir LDR için kullanılan transistor devresi

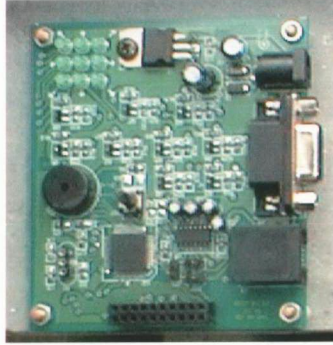
Şekil 3.6 ile sunulan devre şeması 16 adet LDR'nin her biri için kullanılan transistor hücresidir. Transistor ortak emiter tipinde bağlanmıştır. LDR ışık almadığı durumda direnci çok büyüktür. Transistorun kollektör gerilimi 5V olarak okunur ve kollektöre bağlı bulunan LED ışık vermektedir. Bu durumdayken mikrodnetleyiciye lojik 1 sinyali iletilmektedir. Transistorun bazında bağlı bulunan LDR yeterli miktarda ışık aldığıında direnci küçülür. LDR'nin direnci yeterince küçüldüğünde transistor iletime geçer ve kolektör gerilimi sıfıra düşer, kollektördeki LED söner. Bu durumda da mikrodnetleyiciye lojik 0 sinyali iletilmektedir.

3.3 Kontrol Kartı

Kontrol kartı beş bölümden oluşmaktadır. Bunlar güç devresi, mikrodnetleyici devresi, sesli uyarı devresi, seri port sürücü devresi ve yön gösterge devreleridir.

Kontrol kartının şematik çizimi Şekil 3.7 ile bir sonraki sayfada sunulmuştur. Kontrol kartı, algılayıcı kartından alınan giriş sinyallerini işleyerek gidilecek olan yöne karar verir. Bu yön bilgisi kontrol kartı üzerinde yer alan gösterge LED'lerine aktarılır. Aynı zamanda seri port üzerinden bilgisayara gönderilir.

Kontrol kartının fiziksel görüntüsü Şekil 3.8’de yer almaktadır.

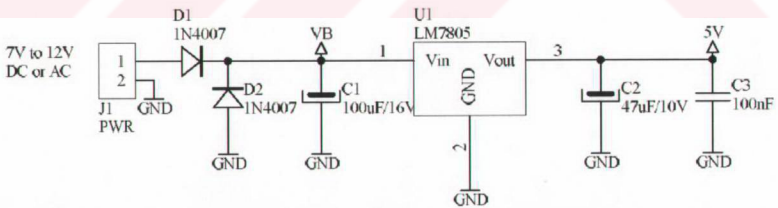


Şekil 3.8 Kontrol kartının üstten görüntüsü

3.3.1 Güç Devresi

Güç devresi yön bulma kontrol modülünün enerji girişini düzenleyen bir devredir. Devre girişine uygulanan 7-12V arasındaki AC veya DC gerilimlere karşılık devre çıkışında sabit 5V DC gerilim oluşmaktadır. Bu gerilim yön bulma kontrol modülünün bütün besleme gerilimlerini sağlamaktadır.

Şekil 3.9’da devreye ait şematik çizim yer almaktadır.



Şekil 3.9 Gerilim doğrultucu ve düzenleyici güç devresinin şeması

3.3.2 Mikrodenetleyici Devresi

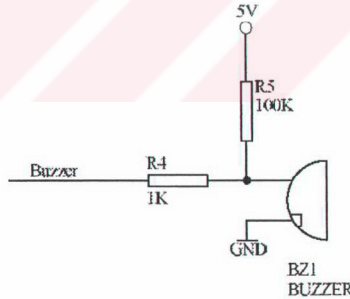
Mikrodenetleyici devresi olarak belirtilen kısmın merkezinde Microchip firmasının PIC184320 kodlu entegresi yer almaktadır. TQFP-44 paket yapısındaki entegre mikro tipte

SMD bir malzemedir. Bu tüm devrenin seçilmesindeki nedenler devre içi programlama ve hata takibine imkan veren yapısı, hafıza yapısının büyük olması ve komut setinin az olması, C diline uyumluluğu, dahili osilatöre sahip olması, temin edilebilirliği ve maliyetinin diğerlerine oranla daha kabul edilebilir olmasıdır. Bu tüm devre hakkındaki detaylı bilgiler için malzemenin veri dosyasının (*datasheet*) ilk birkaç sayfası EK 4’de sunulmuştur. Daha detaylı bilgi için Microchip firmasının web sitesi ziyaret edilebilir.

LDR kartından alınan giriş bilgileri, gösterge devresine aktarılan yön bilgileri, seri port haberleşmesinin kontrolü ve karar algoritması bu tüm devre üzerindeki program aracılığıyla sağlanmaktadır.

3.3.3 Sesli Uyarı Devresi

Sesli uyarı devresi engelle gezgin robot ışığa doğru ilerlerken engelle karşılaştığında kullanılabilecek bir buzzer devresidir. Buzzer kapasitif yapıdadır. Birbirine yakın iki metal yapraktan alınan bağlantıları çıkış uçları olarak kullanılır. Uçlara uygulanan darbe treni şeklindeki gerilim ile yapraklar titreştirilerek tiz bir ikaz sesinin çıkması sağlanır. Darbe trenin frekansı ve darbe genişliği sesin tonu belirler. Buzzer için kullanılan devre Şekil 3.10’da sunulmuştur.



Şekil 3.10 Sesli uyarı devresinin şeması

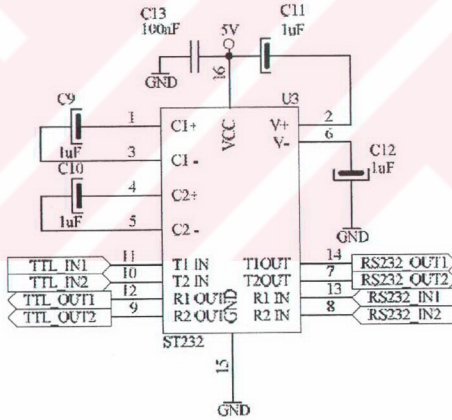
3.3.4 Seri Port Sürücü Devresi

Gezgin robotun gözlem amaçlı verilerini bilgisayara aktarmak için kullandığı seri haberleşme portu bu devrenin özünü oluşturmaktadır. Bilindiği üzere genelde bütün sayısal devreler TTL veya CMOS sayısal seviyelerinde çalışır. Buna karşın Bölüm 1’de anlatıldığı gibi RS232

sinyalleri -3V ile -25V ve +3V ile +25V arası için tanımlanmıştır. Bu yüzden, RS232 haberleşmesinde voltaj seviye uyumluluğunu sağlamak amacıyla, RS232 çeviricileri olarak adlandırılan, sayısal seviye çevirimi yapan özel tüm devreler kullanılır.

Eskiden yaygın olarak kullanılan RS232 seviye çeviricileri, 1488 RS232 sürücüsü (*driver*) ile 1489 RS232 alıcısıdır (*receiver*). Her bir tüm devre 4 tane bir çeşit dönüştürücü (*inverter*) içerir. Bir sürücü, +7.5V ile +15V ve -7.5V ile -15V arasında değişen iki besleme gerektirir. Bu durum, birçok kere genelde bir tek +5V beslemesi bulunan devrelerde bir problem oluşturur. Bu tüm devrelerin tek bir avantajı ucuz olmalarıdır.

Bu tüm devreler yerine, günümüzde RS232 haberleşmelerinde, genelde tek bir besleme ile çalışan tüm devreler kullanılır. Örneğin, ST 232 içindeki *charge pump* olarak adlandırılan bir donanım özelliği ile, tek +5V beslemeden +10V ve -10V üretir. Bu tüm devre ayrıca aynı kılıf içinde iki alıcı ve iki gönderici (sürücü) içerir.



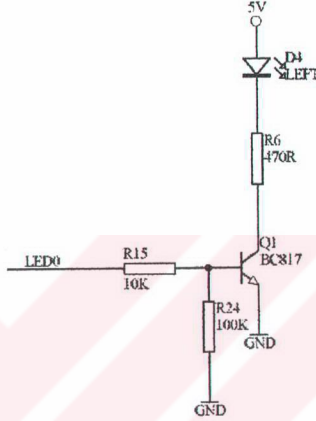
Şekil 3.11 Seri port sürücü devresinin şeması

Bu uygulamada seri port sürücü devresi olarak ST 232 tüm devresi kullanılmıştır. Devrenin şematik çizimi Şekil 3.11'de yer almaktadır. ST 232 tüm devresi hakkındaki detaylı bilgiler için malzemenin veri dosyasının (*datasheet*) ilk birkaç sayfası yine EK 4 sunulmuştur.

3.3.5 Yön Gösterge Devresi

Bu devre karar algoritması çıkışında oluşan yön bilgisini gösteren bir devredir. Devre matris

şeklinde dizilmiş yeşil LED'ler kullanılarak hazırlanmıştır. Her bir LED'in yakınına yön bilgisini göstermek üzere PCB maskesine ok işaretleri eklenmiştir. LED'ler ortak emetörlü olarak bağlanmış bulunan transistörlerin kollektör yükleri olarak yerleştirilmişlerdir. Her bir LED'i süren transistör devresi Şekil 3.12 ile aynı yapıdadır.



Şekil 3.12 Yön gösterge devresinin şeması

3.4 Yön Bulma Uygulamasının Yazılımı

Yön bulma uygulamasında iki yazılım söz konusudur. Bunlardan birincisi kontrol kartı üzerindeki mikrodenetleyici yazılımı ikincisi ise bilgisayar ortamındaki gözlemci yazılımdır.

3.4.1 Mikrodenetleyici Yazılımı

Yön bulma kontrol modülünün uygulama aşamasında kullanılacak YSA algoritmasının mikrodenetleyici ortamında programlanabilir olması en önemli kriterdir. Çünkü kullanılan ağırlıkların sayısı ve ondalık sayı olarak hassasiyetleri işlemcinin hafıza kapasitesi ile sınırlıdır. Ayrıca YSA algoritmasında yer alan bütün karar fonksiyonlarının mikrodenetleyici ortamında yazılacak program ile gerçekleşmesi mümkün olmayabilir.

Bütün bu kriterler göz önüne alınarak hafıza yapısı büyük olan ve C diline (MATLAB'da hazırlanan programla paralellik göstermesi için) uyumlu olan bir işlemci seçilmesi daha uygun olacaktır.

Bu nedenlerden dolayı Microchip firması tarafından üretilen ve temin edilmesi mümkün olan bir işlemci (PIC18F4320) seçilmiştir.

Mikrodenetleyici yazılımının akış diyagramı Şekil 3.13’de sunulmuştur.



Şekil 3.13 Mikrodenetleyici yazılımının akış diyagramı

Diyagramdan da anlaşılacağı üzere başlangıç işlemlerinden sonra sürekli olarak LDR’ler gözlemlenerek buna bağlı yön bilgisi YSA ile oluşturulur. YSA ile elde edilen yön bilgisi LED’lere ve aynı zamanda seri port üzerinden bilgisayara aktarılır.

Mikrodenetleyici yazılımında seri port için haberleşme hızı olarak 19200 bps seçilmiştir. Seri portunun diğer ayarları ise Çizelge 3.1’de belirtildiği gibi seçilmiştir.

Çizelge 3.1 Seri Port Ayarları

veri uzunluğu	sekiz bit
stop bit sayısı	bir
parity test	yok
veri akış kontrolü	yok

Bu bilgiler uygulamanın bilgisayar programındaki default port ayarlarına da aynı şekilde yansıtılmıştır.

3.4.2 Bilgisayar Yazılımı

C++ Builder nedir?

Windows altında görsel (visual) olmayan programlama dilleri ile program yazmak gerçekten zordur. Windows'un çıktığı ilk zamanlar Windows altında programlamayı destekleyen Pascal ve C derleyicileri de çıktı. Ancak bunlarla program yazmak çok zor olduğu için uzun yıllar insanlar DOS altında program geliştirmeye devam ettiler. Ancak çok az sayıda insan bu dilleri öğrenerek Windows altında programlamaya başladı.

Visual Basic'in piyasaya çıkması ile hem programlama dünyası yepyeni ve çok iyi bir programlama diliyle karşılaştı hem de DOS altında program yapan insanlar hızla Visual Basic kullanarak Windows altında programlamaya başladılar.

Bu sıralar Microsoft Visual C derleyicileri de üretmeye başladı. Ama bu çalışmalar kullanıcının işini çok da kolaylaştırmıyordu. İnsanlar programın yapması gereken işe odaklanmak yerine uğraşılması gereken ayrıntılarla zaman harcamak zorundaydılar.

Borland firması Visual Pascal diyebileceğimiz Delphi'yi ürettiğinde ise C'de program yapan insanlar bile Delphi'yi öğrenerek Windows altında programlamaya başladılar. Çünkü Delphi gerçek bir görsel dildi ve Visual Basic'ten daha gelişmiş, Visual C'den de daha kolaydı.

Borland firması C bilenleri Delphi'ye zorlamak yerine C tabanlı bir programlayıcı olan C++ Builder'ı piyasaya sürdü. Aslında bu çok geç kalmış bir programlama diliydi. Delphi'nin bu kadar yaygınlaşmasının en büyük nedeni C++ Builder'in geç kalmasıydı.

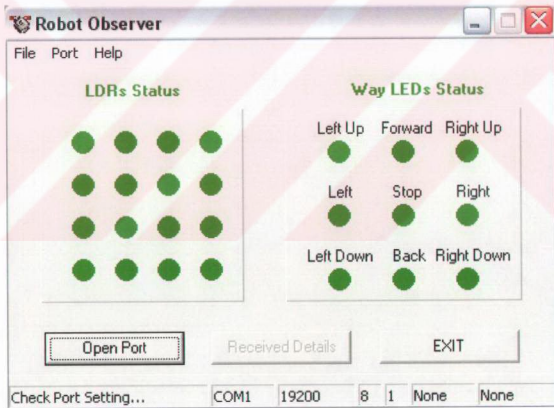
Tez uygulamasında bilgisayar kısmında kullanım kolaylığı ve modülerlik avantajlarından dolayı C++ Builder tercih edilmiştir.

Bilgisayar yazılımında kullanılan akış diyagramı Şekil 3.14'de yer almaktadır. Bu akış diyagramı mikrodenetleyici yazılımı ile paralellik göstermektedir. Tek fark yön bulma kararının mikrodenetleyici tarafından oluşturulmasıdır. Kişisel bilgisayar üzerinde çalışan bu yazılım ileride yapılacak çalışmalara alt yapı oluşturmak amacıyla hazırlanmıştır.

Şekil 3.15'de bilgisayar yazılımının ana penceresi görülmektedir. Bu pencerede LDR durumlarını veya gezgin robotun gittiği yönü gösteren aktif LED göstergeler yer almaktadır. Bu göstergeler gezgin robot üzerindeki LED göstergelerle eş zamanlı olarak çalışmaktadır. Ayrıca kullanıcıya yardımcı bilgiler verebilmek için programa durum çubuğu eklenmiştir. Durum çubuğu ile port ayarları ve açıklayıcı metinlerle kullanıcı yönlendirilmektedir.



Şekil 3.14 Bilgisayar yazılımının akış diyagramı



Şekil 3.15 Bilgisayar yazılımının görüntüsü

Programın komutları:

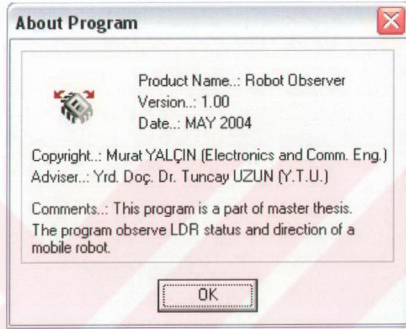
File→Exit komutuyla veya pencere üzerindeki “Exit” butonuyla programdan çıkılarak Windows ortamına dönülebilir.

Port→Open Port komutu veya pencere üzerindeki “Open Port” butonu haberleşme için seri

portun aktif edilmesini sağlar. Port aktifken bu komut etiketleri “Close Port” şeklinde değişmektedir. Yani portun kapatılması yine aynı butonla gerçekleşir. Port aktifken programdan çıkılırsa port otomatik olarak kapatılır.

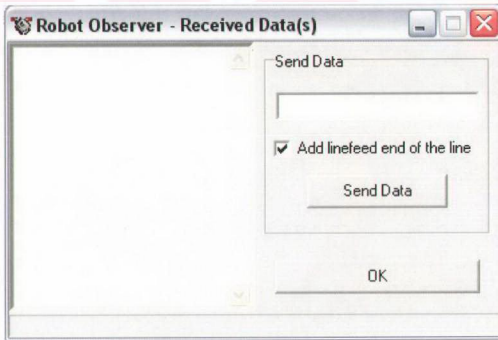
Port→Setting komutuyla istenirse port kapalıyken port ayarları değiştirilebilir.

Help→About komutu kullanılarak program hakkında bazı açıklayıcı bilgiler listelenebilir. Bu komutla açılan pencere Şekil 3.16’da yer almaktadır.



Şekil 3.16 Program hakkında bilgi penceresi

Received Details port aktifken kullanıma açıktır. Received Details komutu çalıştırıldığında Şekil 3.17’deki pencere açılacaktır. OK butonu sadece pencereyi kapatmak için kullanılabilir.



Şekil 3.17 Bilgisayar seri port veri akışını gözlemek için yardımcı pencere

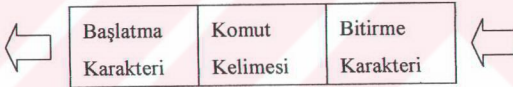
Bu komutla açılan pencerede seri porta gelen verileri gözlemlemek mümkündür. Hatta bazı komutlar göndererek gezgin robottan farklı bilgilerin istenmesi de söz konusu olabilir. Bu kısım ileriki çalışmalarda gerekebileceğinden programa şimdiden ilave edilmiştir.

3.4.3 Haberleşme Protokolü

Mikrodenetleyici ve bilgisayar yazılımları arasındaki haberleşme tek yönlü olarak gerçekleştirilmiştir. Burada mikrodenetleyici yazılımı sunucu (*master*) ve bilgisayar yazılımı istemci (*server*) pozisyonundadır. Yani veri akışı her zaman mikrodenetleyiciden bilgisayara doğru gerçekleştirilir. Sonuç olarak bilgisayar programı sadece bir gözlemci pozisyonundadır.

Mikrodenetleyici yazılımında haberleşme hızı olarak 19200 bps sabittir. Seri portunun diğer özellikleri ise şu şekilde belirlenmiştir: veri uzunluğu → sekiz bit, stop bit sayısı → bir, parite test → yok, veri akış kontrolü → yok. Bu bilgiler uygulamanın bilgisayar programındaki default port ayarlarına da aynı şekilde yansıtılmıştır.

Mikrodenetleyici tarafından bilgisayara aktarılabilecek bilgiler azlığı göz önüne alındığında haberleşmede kullanılacak protokolün sade olacağı açıktır. Çünkü aktarılabilecek bilgiler sadece LDR ve yön göstergelerinin durumlarıdır. Şekil 3.18’de protokol yapısı verilmiştir.



Şekil 3.18 Protokol yapısı

Protokol yapısında yer alan başlatma ve bitirme karakterleri ile komut listesi Çizelge 3.2 ile bir sonraki sayfada yer almaktadır.

Çizelge 3.2 Protokol bileşenleri listesi

Başlatma Karakteri	'X'	
Bitirme Karakteri	'Y'	
Komut Kelimesi	_STOP	10
	_FORWARD	15
	_BACK	20
	_LEFT	25
	_LEFT_UP	30
	_LEFT_DOWN	35
	_RIGHT_UP	40
	_RIGHT_DOWN	45
	_RIGHT	50
	_LDR_1_1_H	60
	_LDR_1_1_L	61
	_LDR_1_2_H	62
	_LDR_1_2_L	63
	_LDR_1_3_H	64
	_LDR_1_3_L	65
	_LDR_1_4_H	66
	_LDR_1_4_L	67
	_LDR_2_1_H	68
	_LDR_2_1_L	69
	_LDR_2_2_H	70
	_LDR_2_2_L	71
	_LDR_2_3_H	72
	_LDR_2_3_L	73
	_LDR_2_4_H	74
	_LDR_2_4_L	75
	_LDR_3_1_H	76
	_LDR_3_1_L	77
	_LDR_3_2_H	78
	_LDR_3_2_L	79
	_LDR_3_3_H	80
	_LDR_3_3_L	81
	_LDR_3_4_H	82
	_LDR_3_4_L	83
	_LDR_4_1_H	84
	_LDR_4_1_L	85
	_LDR_4_2_H	86
	_LDR_4_2_L	87
	_LDR_4_3_H	88
	_LDR_4_3_L	89
	_LDR_4_4_H	90
	_LDR_4_4_L	91

4. SONUÇ

Bu çalışmada YSA algoritmalarından yararlanarak gezgin robot için yön bulma problemine çözüm bulunmaya çalışılmıştır. Çalışmalar sonucunda başarılı sonuçlar elde edilmiştir. LDR kartından alınan hücresel aydınlatma bilgilerinden yararlanarak gezgin robotun önüne gelen engellerin (LDR'nin aldığı ışığı kesecek şekilde olan engeller) robotun karar algoritması tarafından algılanabildiği gözlenmiştir ve robotun ışık kaynağına doğru hareket edecek biçimde yönünü değiştirebilmesi için gerekli altyapı sağlanmıştır.

Algılayıcıdan alınan değişimlerin ve çıkış değeri olarak mikrodnetleyici kartı üzerindeki LED göstergelerine yansıtılan yön bilgilerinin mikrodnetleyici kartı ile eşzamanlı olarak bilgisayar programı tarafından da gösterilebildiği gözlenmiştir.

Çalışma gerek mikrodnetleyici donanımı gerekse mikrodnetleyici ve bilgisayar yazılımları itibariyle modüler bir yapıda tasarlandığından geliştirilmeye açıktır. Algılayıcı sayısının artırılması hatta LDR yerine mümkünse düşük çözünürlüğe sahip bir kameranın ilave edilmesi, donanım ile bilgisayar arasındaki haberleşmenin kablosuz gerçekleştirilmesi ve mikrodnetleyici üzerindeki yazılımın bu değişiklere göre yapılandırılması çalışmanın daha ileri noktalara taşınmasında etkili olacaktır.

KAYNAKLAR

- Alexe, G., Alexe, S., Hammer, P. L., and Kogan, A., "Comprehensive vs. Comprehensive classifiers in Logical Analysis of Data", March 2002
- Cheung, N., "Machine Learning Techniques for Medical Analysis", School of Information Technology and Electrical Engineering, BsC thesis, University of Queensland, 19 Oct. 2001
- Erdoğan, G. T., (2000), "Gezgin Robot Projesi Denetim Donanımı Tasarımı", Yıldız Teknik Üniversitesi, İstanbul.
- Gümüşkaya, H., (1999), "Mikrodenetleyiciler ve Bilgisayarlar Intel Ailesi ve IBM PC", Alfa Yayınevi, Bursa.
- Hagan, M.T., Demuth, H.B., and Beale, M., "Neural network design." PWS Publishing Company, Boston, 1996.
- Karagülle, İ. - Pala, Z., (1998), "Borland C++ Builder Visual Programlama", Türkmen Kitabevi, 16-54, İstanbul.
- MarhWorks, (2002), "MATLAB 6.5 Help Files", <http://www.mathworks.com/>
- Masters, T., "Advanced algorithms for neural networks", Wiley, New York, 1995.
- Philips., (1983) "Sensors", Philips, December 1981 – September 1983, 370-377.
- Schalkoff, R. J., (1997) "Artificial Neural Networks", McGraw-Hill Inc., Singapore, 1997.
- Schildt, H., – Akın, C., (1999), "C++ Temel Öğrenim Kılavuzu", Alfa Yayınevi, Bursa.
- Specht, D.F. , "Probabilistic neural networks", Neural networks, vol.3, pp. 109-118, 1990.
- Uzun, T., (2001), "Endüstriyel Otomatik Kontrol Sistemleri Ders Notları", Yıldız Teknik Üniversitesi, İstanbul.
- Uzun, T., (2001), "Programlanabilir Denetleyiciler Ders Notları", Yıldız Teknik Üniversitesi, İstanbul.
- Yıldırım, T., (2002), "Nöron Ağlarına Giriş Ders Notları", Yıldız Teknik Üniversitesi, İstanbul.
- Yıldırım, T., (2003), "Yapay Sinir Ağları ve Uygulamaları Ders Notları", Yıldız Teknik Üniversitesi, İstanbul.
- Yüksel, İ., (2000), "MATLAB ile Mühendislik Sistemlerinin Analizi ve Çözümü", Vipaş A.Ş., 1-68, Bursa.

EKLER

- Ek 1 PC Seri Haberleşmesi için Kullanılan Çekirdek Programlar
 - Ek 2 MATLAB Çalışmalarında Oluşturulan Program Dosyaları (Sadece Resim Bilgisi Giriş Olarak Kullanıldığında)
 - Ek 3 MATLAB Çalışmalarında Oluşturulan Program Dosyaları (Resim, Aydınlık ve Mesafe Bilgileri Giriş Olarak Kullanıldığında)
 - Ek 4 Donanımda Kullanılan Özel Malzemelerin Veri Dosyaları (Datasheets)
- 

Ek 1 PC Seri Haberleşmesi için Kullanılan Çekirdek Programlar

PC Seri Portundaki adres atamalarını tespit eden C programı

```
#include <stdio.h>

#include <dos.h>

void main(void)

{

    unsigned int far *ptr_addr;

    unsigned int address;

    int i;

    ptr_addr = (unsigned int far *) 0x00000400;

    for (i=0; i<4; i++)

    {

        address = *ptr_addr;

        if (address == 0)

            printf("COM$d için Port bulunamadi \n", i+1);

        else

            printf("COM$d iciri atanan adres $Xh\n", i+1, address);

        *ptr_addr++;

    }

}
```

Yukarıda verilen C programı Seri Port adresleri COM1, 3F8h; COM2, 2F8h; COM3, tanımsız; COM4, tanımsız olan bir bilgisayarda çalıştırıldığında aşağıdaki çıkış elde edilir.

COM1 için atanan adres 3F8h

COM2 için atanan adres 2F8h

COM3 için Port bulunamadi

COM4 için Port bulunamadi

İki PC'yi COM2 üzerinden haberleştiren Assembler programı

Program başında PC ekranına bazı mesajlar basıldıktan sonra, COM2 port'u, 4800 baud hızında, eşlik bit'siz (No Parity), 1 STOP bit'i kullanarak ve 8-bit veri gönderecek şekilde programlanır.

Programda daha sonra bir tuşa basılması kontrol edilir. Eğer basılan tuş ESC karakteri ise programdan çıkılır. Bu tuşun haricindeki bir karakter, COM2 üzerinden karşı makineye gönderilir. Tuşa basılmamış ise, COM2 port'u kontrol edilerek karşı makineden veri gelmediği anlaşılır. Program bu şekilde sonsuz bir döngü içinde, tuş gözlemeye devam eder ve ESC tuşu ile program bitirilir. Programın diğer açıklamaları aşağıda açıklamalar sütununda verilmektedir.

```
;seriport.asm .MODEL SMALL DATA SEGMENT
MSG DB 'Seri Haberlesme Programi', 0AH, 0DH
DB `COM2, 4800, No Parity, 1 STOP, 8-bit veri', 0AH, 0DH
DB 'Basilan tus diger PCye gonderilir...', 0AH, 0DH
DB 'Cikmak ioin ESC tusuna basiniz...', 0AH, 0DH, '$'
DATA ENDS

CODE SEGMENT
ASSUME CS:CODE, DS:DATA
MAIN:
MOV AX, DATA
MOV DS, AX          ; DS'i belirle
MOV AH, 09H
MOV DX, OFFSET MSG
INT 21H              ; Ekrana mesaj yaz

; COM2 programlanmasi
MOV AH, 0            ; COM port programlama
MOV DX, 1             ; COM2 secilir
MOV AL, 0C3H          ; 4800, No Parity, 1 STOP, 8-bit veri INT 14H

;Tus kontrolu ve basilan tusun COM2'ye gonderilmesi
TEKRAR:
MOV AH, 01H          ; INT 16H ve AH = 01H ile tus kontrolu
INT 16H              ; Eger Z = 1 ise, tusa basilmamis
JZ NEXT              ; Basilmamis ise, COM port kontrol
MOV AH, 0            ; Tusa basildi, onu al
INT 16H              ; AL = basilan ASCII karakter
CMP AL, 1BH          ; Basilan tus ESC mi?
JE CIKIS              ; Evet ise, programdan cik, Hayir ise
                     ; karakteri COM2'ye gonder...
MOV AH, 1
MOV DX, 01
INT 14H              ; COM2 kontrolu, karakter var ise oku ve ekranda goster

NEXT:
MOV AH, 03H          ; COM2'nin durumunu ogren
MOV DX, 01H
INT 14H; AH          ; COM port'unun durumu
AND AH, 01H          ; DO haricini temizle
CMP AH, 01H          ; DO kontrolu, veri yok ise
```

```
JNE TEKRAR          ; klavyeyi gozlemeye devam...
MOV AH, 02H          ; COM2'de veri var! Onu al.
MOV DX, 01H
INT 14H              ; Karakteri al, AL = Karakter
MOV DL,AL             ; ve INT 21 ile goster...
MOV AH, 02H
INT 21H,JMP

TEKRAR CIKIS:
MOV AH, 4CH          ; DOS'a cikis...
INT 21H
CODE ENDS
END MAIN
```



Ek 2 MATLAB Çalışmalarında Oluşturulan Program Dosyaları

(Sadece Resim Bilgisi Giriş Olarak Kullanıldığında)

learn_bw.m

```
function [network]=learn_bw
black = 0;
white = 1;
go = black; %black color allow to GO
% create source input vector
source = [black white];
[row column] = size(source);
% create target input vector
for i=1:row
    for j=1:column
        if source(i,j) == go
            target(i,j) = 1;
        else
            target(i,j) = 0;
        end
    end
end
% create decision algorithm (with GRNN from NN)
network = newgrnn(source,target);
```

test_bw.m

```
function test_bw(picture,network)
%image informations
%imfinfo(picture);
source = imread(picture);
%show image in new figure windows
figure;
imshow(source);
shg;

%test image
tic          %start timer
source = double(source);
[row column] = size(source);
```

```
for i=1:row
    for j=1:column
        if sim(network,source(i,j)) > 0.5
            result(i,j) = 1;
        else
            result(i,j) = 0;
        end
    end
end
result
toc    %stop timer
```

Ek 3 MATLAB Çalışmalarında Oluşturulan Program Dosyaları
(Resim, Aydınlık ve Mesafe Bilgileri Giriş Olarak Kullanıldığında)

learn_bw.m

```
function [network]=learn_bw

% Yakın/Uzak cismin alıcısındaki görüntüsü
black = 0;
white = 1;
pixel = black;

% Ortamdaki aydınlık
dark = 0;
light = 1;
lighting = light;

% Hedefle araç arasındaki mesafe (cm)
near = 3;
far = 8;
distance = near;
tolerance = 3;

% Aracın yapması gereken işlem (sonuc)
run = 1;
stop = 0;
directive = run;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pixel  lighting  distance
% -----
% black dark near
% black dark near+1      Pixel and
% black dark near+2      Lighting
% .
% .
% .
% black dark far-2      send RUN directive to
% black dark far-1      the vehicle.
% black dark far
% -----
```

```

% black light near
% black light near+1
% black light near+2
% .
% .
% .
% black light far-2
% black light far-1
% black light far
% -----
% white dark near
% white dark near+1
% white dark near+2
% .
% .
% .
% white dark far-2
% white dark far-1
% white dark far
% -----
% white light near
% white light near+1      Pixel and
% white light near+2      Lighting
% .
% .
% .
% white light far-2      send RUN directive to
% white light far-1      the vehicle.
% white light far
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% create source input vector
P = [black white];      % Pixel vector
[row_P col_P] = size(P);

L = [dark light];      % Lighting vector
[row_L col_L] = size(L);

for i=near:far          % Distance vector
    D(1,(i-near)+1) = i;
end
[row_D col_D] = size(D);

```

```

S = zeros(3, col_P * col_L * col_D);
[row_S col_S] = size(S);

x = 1;
while x <= col_S
    for i=1:col_P
        for j=1:col_L

            for k=1:col_D
                S(1,x) = P(i);
                S(2,x) = L(j);
                S(3,x) = D(k);
                x = x+1;
            end
        end
    end
    % create target input vector
    for i=1:col_S
        T(i) = stop;
    end

    for i=1:col_S
        if abs(S(3,i)-far) < tolerance
            T(i) = run;
        end
        if S(1,i) == S(2,i)
            T(i) = run;
        end
    end

    %P
    %L
    %D
    %S
    %T

    % create decision algorithm (with GRNN from NN)
    network = newgrnn(S,T);

```

test_bw.m

```

function test_bw(picture, network, L, D)

%image informations
%imfinfo(picture);
source = imread(picture);

%show image in new figure windows
%figure;
%imshow(source);
%shg;

%test image
%tic      %start timer
source = double(source);
[row column] = size(source);
for i=1:row
    for j=1:column
        if sim(network, [source(i,j), L, D]') > 0.5
            result(i,j) = 1;
        else
            result(i,j) = 0;
        end
    end
end
result
%toc      %stop timer

```

Ek 4 Donanımda Kullanılan Özel Malzemelerin Veri Dosyaları (Datasheets)



MICROCHIP PIC18F2220/2320/4220/4320

28/40/44-Pin High-Performance, Enhanced Flash MCUs with 10-bit A/D and nanoWatt Technology

Low-Power Features:

- Power Managed modes:
 - Run: CPU on, peripherals on
 - Idle: CPU off, peripherals on
 - Sleep: CPU off, peripherals off
- Power Consumption modes:
 - PRI_RUN: 150 μ A, 1 MHz, 2V
 - PRI_IDLE: 37 μ A, 1 MHz, 2V
 - SEC_RUN: 14 μ A, 32 kHz, 2V
 - SEC_IDLE: 5.8 μ A, 32 kHz, 2V
 - RC_RUN: 110 μ A, 1 MHz, 2V
 - RC_IDLE: 52 μ A, 1 MHz, 2V
 - Sleep: 0.1 μ A, 1 MHz, 2V
- Timer1 Oscillator: 1.1 μ A, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A
- Two-Speed Oscillator Start-up

Oscillators:

- Four Crystal modes:
 - LP, XT, HS: up to 25 MHz
 - HSPLL: 4-10 MHz (16-40 MHz internal)
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal oscillator block:
 - 8 user selectable frequencies: 31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz
 - 125 kHz-8 MHz calibrated to 1%
 - Two modes select one or two I/O pins
 - OSCTUNE – Allows user to shift frequency
- Secondary oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor
 - Allows for safe shutdown if peripheral clock stops

Peripheral Highlights:

- High current sink/source 25 mA/25 mA
- Three external interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution is 6.25 ns ($T_{CY}/16$)
 - Compare is 16-bit, max. resolution is 100 ns (T_{CY})
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead-time
 - Auto-Shutdown and Auto-Restart
- Compatible 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Dual analog comparators
- Addressable USART module:
 - RS-232 operation using internal oscillator block (no external crystal required)

Special Microcontroller Features:

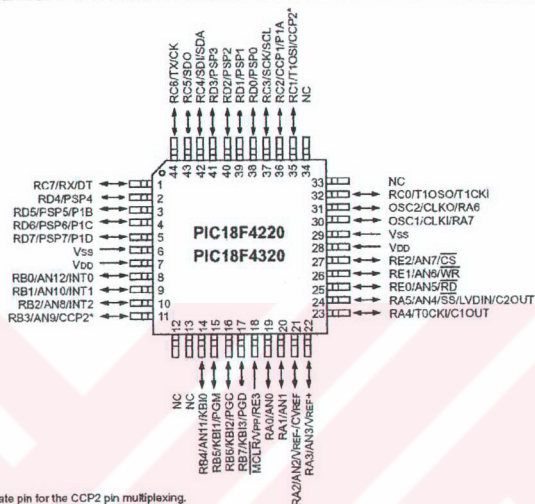
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
 - 2% stability over V_{DD} and Temperature
- Single-supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range: 2.0V to 5.5V

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		USART	Comparators	Timers 8/16-bit
	Flash (bytes)	# Single Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI™	Master I ² C™			
PIC18F2220	4096	2048	512	256	25	10	2/0	Y	Y	Y	2	2/3
PIC18F2320	8192	4096	512	256	25	10	2/0	Y	Y	Y	2	2/3
PIC18F4220	4096	2048	512	256	36	13	1/1	Y	Y	Y	2	2/3
PIC18F4320	8192	4096	512	256	36	13	1/1	Y	Y	Y	2	2/3

PIC18F2220/2320/4220/4320

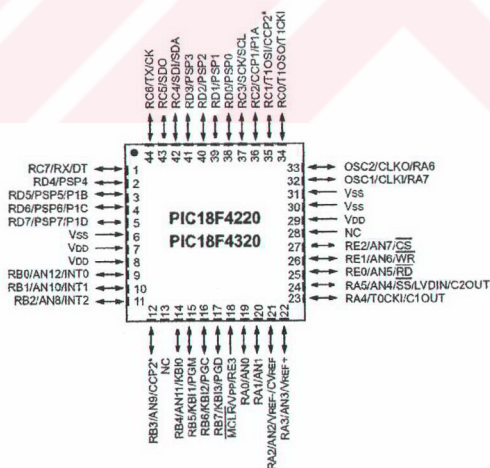
Pin Diagrams (Cont.'d)

TQFP



* RB3 is the alternate pin for the CCP2 pin multiplexing.

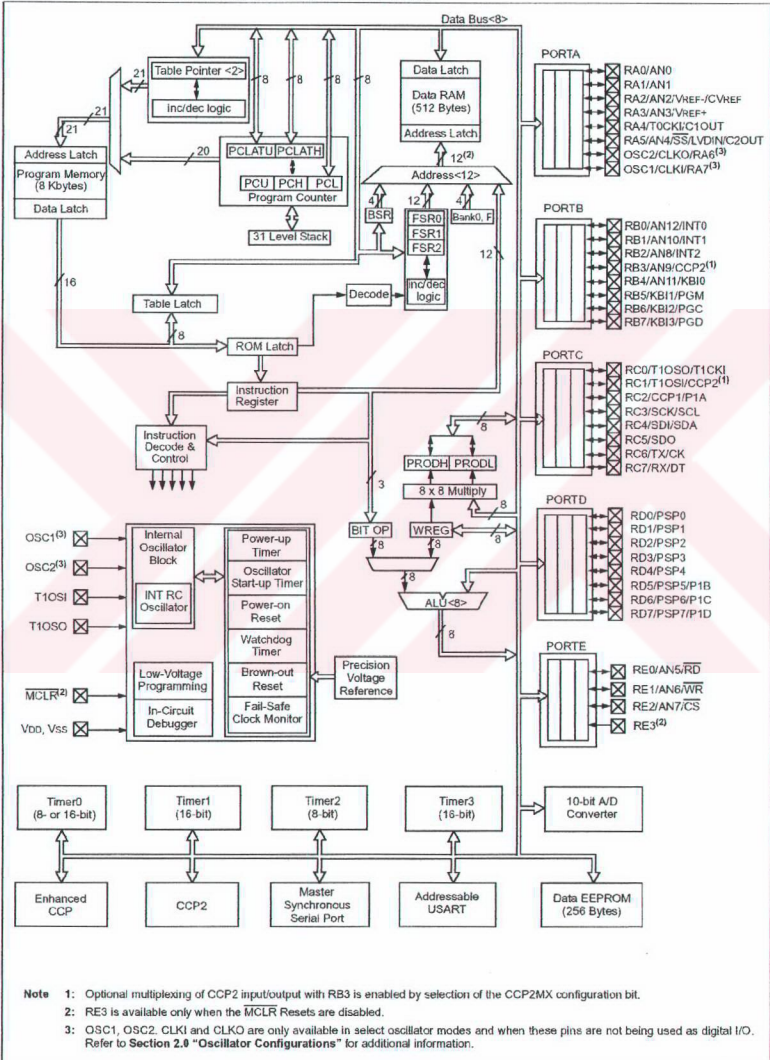
QFN



* RB3 is the alternate pin for the CCP2 pin multiplexing.

PIC18F2220/2320/4220/4320

FIGURE 1-2: PIC18F4220/4320 BLOCK DIAGRAM



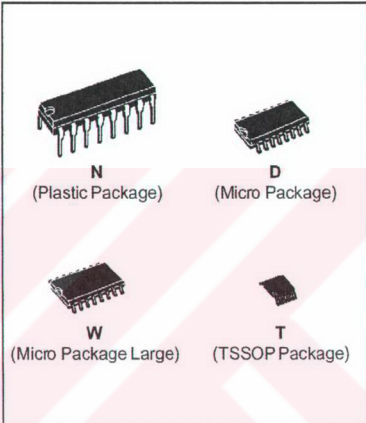

ST232

5V POWERED MULTI-CHANNEL RS-232 DRIVERS AND RECEIVERS

- SUPPLY VOLTAGE RANGE: 4.5 TO 5.5V
- SUPPLY CURRENT NO LOAD (TYP): 5mA
- TRANSMITTER OUTPUT VOLTAGE SWING (TYP): $\pm 7.8V$
- CONTROLLED OUTPUT SLEW RATE
- RECEIVER INPUT VOLTAGE RANGE: $\pm 30V$
- DATA RATE (TYP): 220Kbps
- OPERATING TEMPERATURE RANGE: -40 TO 85 °C, 0 TO 70 °C
- COMPATIBLE WITH MAX232 AND MAX202

DESCRIPTION

The ST232 is a 2 driver, 2 receiver device following EIA/TIA-232 and V.28 communication standard. It is particularly suitable for applications where $\pm 12V$ is not available. The ST232 uses a single 5V power supply and only four external capacitors (0.1 μF). Typical applications are in: Portable Computers, Low Power Modems, Interfaces Translation, Battery Powered RS-232 System, Multi-Drop RS-232 Networks.

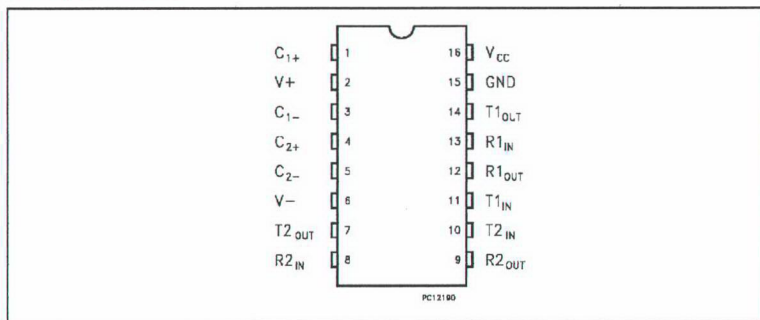


ORDER CODES

Type	Temperature Range	Package	Comments
ST232CN	0 to 70 °C	DIP-16	25 parts per tube / 40 tube per box
ST232BN	-40 to 85 °C	DIP-16	25 parts per tube / 40 tube per box
ST232CD	0 to 70 °C	SO-16 (Tube)	50 parts per tube / 20 tube per box
ST232BD	-40 to 85 °C	SO-16 (Tube)	50 parts per tube / 20 tube per box
ST232CDR	0 to 70 °C	SO-16 (Tape & Reel)	2500 parts per reel
ST232BDR	-40 to 85 °C	SO-16 (Tape & Reel)	2500 parts per reel
ST232CW	0 to 70 °C	SO-16 Large (Tube)	49 parts per tube / 25 tube per box
ST232BW	-40 to 85 °C	SO-16 Large (Tube)	49 parts per tube / 25 tube per box
ST232CWR	0 to 70 °C	SO-16 Large (Tape & Reel)	1000 parts per reel
ST232BWR	-40 to 85 °C	SO-16 Large (Tape & Reel)	1000 parts per reel
ST232CT	0 to 70 °C	TSSOP16 (Tube)	only for samples
ST232BT	-40 to 85 °C	TSSOP16 (Tube)	only for samples
ST232CTR	0 to 70 °C	TSSOP16 (Tape & Reel)	2500 parts per reel
ST232BTR	-40 to 85 °C	TSSOP16 (Tape & Reel)	2500 parts per reel

ST232

PIN CONFIGURATION



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1	C ₁₊	Positive Terminal for the first Charge Pump Capacitor
2	V+	Doubled Voltage Terminal
3	C ₁₋	Negative Terminal for the first Charge Pump Capacitor
4	C ₂₊	Positive Terminal for the second Charge Pump Capacitor
5	C ₂₋	Negative Terminal for the second Charge Pump Capacitor
6	V-	Inverted Voltage Terminal
7	T2 _{OUT}	Second Transmitter Output Voltage
8	R2 _{IN}	Second Receiver Input Voltage
9	R2 _{OUT}	Second Receiver Output Voltage
10	T2 _{IN}	Second Transmitter Input Voltage
11	T1 _{IN}	First Transmitter Input Voltage
12	R1 _{OUT}	First Receiver Output Voltage
13	R1 _{IN}	First Receiver Input Voltage
14	T1 _{OUT}	First Transmitter Output Voltage
15	GND	Ground
16	V _{CC}	Supply Voltage

ABSOLUTE MAXIMUM RATINGS (Note 1)

Symbol	Parameter	Value	Unit
V _{CC}	Supply Voltage	-0.3 to 6	V
T _{IN}	Transmitter Input Voltage Range	-0.3 to (V _{CC} + 0.3)	V
R _{IN}	Receiver Input Voltage Range	±30	V
T _{OUT}	Transmitter Output Voltage Range	(V _{CC} + 0.3) to (V _{CC} - 0.3)	V
R _{OUT}	Receiver Output Voltage Range	-0.3 to (V _{CC} + 0.3)	V
T _{SCTOUT}	Short Circuit Duration on T _{OUT}	infinite	
T _{slg}	Storage Temperature Range	-65 to +150	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.
 Note1: No external supply can be applied to V+ terminal and V- terminal.

ÖZGEÇMİŞ

Doğum tarihi 10.05.1978

Doğum yeri BURSA

Lise 1992-1996

Bursa Demirtaşpaşa Teknik Lisesi
Elektronik Bölümü

Lisans 1996-2000

Uludağ Üniversitesi
Mühendislik ve Mimarlık Fakültesi
Elektronik Mühendisliği

Yüksek Lisans 2000-...

Yıldız Teknik Üniversitesi
Elektronik ve Haberleşme Anabilim Dalı
Elektronik Bölümü

Çalıştığı kurumlar

1998-2000

KINAY Mühendislik, Endüstriyel Otomasyon

2000-2001

İSNET, NETAŞ Yetkili Teknik Servisi

2002-2002

E3TAM, Endüstriyel Tasarım ve Ar-Ge

2003-2004

TESLA, Elektronik Kart Tasarım ve Programlama