

YAPAY ZEKA İLE MOBİL ROBOT KONTROLU

**YÜKSEK LİSANS TEZİ
Müh. Okan OKUTKAN
(514031102)**

**Tezin Enstitüye Verildiği Tarih : 9 Mayıs 2006
Tezin Savunulduğu Tarih : 15 Haziran 2006**

Tez Danışmanı : Doç.Dr. Şakir KOCABAŞ

Diğer Jüri Üyeleri Doç.Dr. Metin Orhan KAYA (İ.T.Ü.)

Doç.Dr.Şeniz ERTUĞRUL (İ.T.Ü.)

HAZİRAN 2006

Önsöz

Gelişen savunma sanayi ve insansız muharebe araçlarına doğan ihtiyaç neticesinde harita üzerinde hedefine ulaşmak için gerekli yolu bulabilen ve gerekirse çevresinin haritasını çıkarabilen, barındırdığı algılayıcılar ile kontrol merkezine geri bildirimde bulunabilen, gerektiğinde kontrol merkezinden uzaktan komuta edilebilen bir mobil robotun tasarımına 2003 yılında başladım. İstanbul Teknik Üniversitesi Makina Fakültesi Makina Mühendisliği bölümündeki bitirme tezim sırasında üzerinde çalışmaya başladığım “İnternet Üzerinden Kontrollü Robot Kamera” projesi daha da gelişerek yüksek lisans tezim olan “Yapay Zeka Kontrollü Mobil Robot” projesinin temellerini oluşturdu.

Tasarım sürecinde gerek donanım gerekse yazılım olarak mümkün olan en genişlemeye müsait çözümleri tercih etmeye özen gösterdim. Bazı donanım ve yazılım kısıtlamaları neticesinde ITU-Bot bugünkü halini aldı.

Bu süreç içerisinde bana öneri ve görüşleri ile ışık tutan, tez danışmanım Sayın Doç. Dr. Şakir Kocabaş’a teşekkürlerimi sunarım.

Ayrıca donanım tasarımı esnasında yardımlarından dolayı, Mikroelektronika Yugoslavya’dan Zoran Ristic ve Warren Schroeder’e, algılayıcı modüllerinin teminindeki ve projeye adaptasyonundaki yardımlarından dolayı Acroname Inc. USA ve Devantech UK çalışanlarına, Toothpic modülünün temininde ve yaşanan sorunları gidermemde yardımcı olan Flexipanel Ltd. UK’ den Richard Hoptroff’a ve Parallax Inc. USA çalışanlarına, ana kontrol kartının temininde yardımlarından dolayı Beti Bilişim’den Emrah İşsever’e ve projem sırasında gerekli olan maddi ve manevi desteğini esirgemeyen aileme teşekkür ederim.

Projem ve eğitimim sırasında gösterdikleri sabır, maddi ve manevi destek nedeniyle bu çalışma aileme adanmıştır.

Mayıs 2006

Okan Okutkan

İÇİNDEKİLER

KISALTMALAR	VI
TABLO LİSTESİ	VII
ŞEKİL LİSTESİ	VIII
ÖZET	X
SUMMARY	XI
1 GİRİŞ	1
1.1 Konu ile İlgili Çalışmalar	2
1.2 Projenin Amacı ve Yapısı	5
2 ROBOTİK	7
2.1 Robot Tanımı	7
2.2 Robotların Kullanım Alanları:	9
2.2.1 Endüstriyel Robotik	9
2.2.2 Operasyonel Robotik	9
2.2.3 Tıp ve Sağlıkta Robotik	9
2.2.4 Askeri Robotik	10
2.2.5 Bilimsel Robotik	10
2.2.6 Sinemada Robotik	10
2.2.7 Hobi ve Eğitim Amaçlı Robotlar ve Robot Oyuncaklar	11
2.3 Robotların Gelişimi	11
3 YAPAY ZEKA	21
3.1 Yapay Zekanın Tanımı	21
3.2 Yapay Zeka Tarihçesi	23
4 ROBOTLAR VE YAPAY ZEKA	29
4.1 Mobil Robotların Gereksinimleri	29
4.2 Navigasyon Sistemi	30
4.2.1 Yol Bulma	30
4.2.2 Rota Planlama	34
4.2.3 Engel Algılama	34
4.2.4 Engelden Sakınma	35
4.2.5 Çarpışma Algılama	35
4.3 Kontrol Sistemi ve Mobil Robot Kontrol Yaklaşımları	35
5 YAPAY ZEKA İLE MOBİL ROBOT KONTROLÜ PROJESİ	39
5.1 Hedefler	39
5.2 Robotun Görevleri	39
5.3 Robotun Kontrol Yapısı	40
5.3.1 Strateji Katmanı	42
5.3.2 Taktik Katmanı	43
5.3.3 Uygulama Katmanı	45

5.4	Yol Bulma Algoritması	48
5.5	Robotun Genel Yapısı	50
6	DONANIM	52
6.1	Kullanılan Teknolojiler ve Çalışma Prensipleri	52
6.1.1	PIC Mikro İşlemci	52
6.1.2	IEEE 802.11 B ve G standartları	54
6.1.3	RS232 Standardı	55
6.1.4	I2C Veri Yolu:	56
6.2	Donanım Yapısı	57
6.3	Kontrol Sistemi	58
6.3.1	Ana Kontrol Kartı:	59
6.3.2	Mikro işlemci	60
6.4	Hareket Sistemi	63
6.4.1	Diferansiyel Sürüş Sistemi	63
6.4.2	DC Motorlar	66
6.4.3	DC Motor Kontrol Kartı	67
6.4.4	Algılayıcı ve Kamera Hareket Sistemi	70
6.4.5	Algılayıcı Tareti	70
6.4.6	Kamera Tareti	72
6.4.7	Servo Motorlar	73
6.4.8	Servo Motor Kontrol Kartı	77
6.5	Haberleşme Sistemi	79
6.5.1	Kablosuz Yönlendirici	80
6.5.2	Wibox Kablosuz Seri Port Yönlendirici	81
6.6	Algılayıcılar	82
6.6.1	Kızılötesi Mesafe Algılayıcılar	82
6.6.2	Ultrasound Mesafe Algılayıcılar (Sonar)	86
6.6.3	Elektronik Pusula	90
6.6.4	İvme Ölçer	92
6.6.5	Diğer Algılayıcılar	93
6.7	Görüntü ve Ses Sistemleri	95
6.7.1	Kablosuz IP Kamera	95
6.7.2	Konuşma Sentezleyici	96
6.8	Diğer Donanımlar	98
7	YAZILIM	100
7.1	Ana Bilgisayar Yazılımı	100
7.1.1	Strateji ve Taktik Katmanı	102
7.1.2	Uygulama Katmanı	107
7.2	Mikroişlemci Yazılımı	113
7.2.1	Ana Modül	113

7.2.2	Kızıl Ötesi Algılayıcı Yordamı	114
7.2.3	Ultrasonik Mesafe Algılayıcı Yordamı	115
7.2.4	Elektronik Pusula Yordamı	116
7.2.5	İvme Ölçer Yordamı	117
7.2.6	Hareket Algılayıcı Yordamı	117
7.2.7	Çarpma Algılayıcı Yordamı	119
7.2.8	DC Motor Kontrol Yordamları	120
7.2.9	Servo Motor Komutları	121
7.2.10	Konuşma Sentezleyici Yordamı	124
8	SONUÇ	126
	KAYNAKLAR	131
	EKLER	136
	EK-A Bilgisayar Programı	136
	EK-B Mikro İşlemci programı	136
	EK-C PIC16F877A Pin Tanımları	137
	EK-D Robotun Resmi	141
	ÖZGEÇMİŞ	142

KISALTMALAR

ms	: Mikro Saniye
adc	: Analog Digital Converter - Analog Dijital Çevirici
PIC	: Programmable Interface Controllers
MIT	: Massachusetts Institute of Technology
CMU	: Carnegie Melon University
I2C	: Inter-Integrated Circuit
DIP	: Dual In Line Package
PDIP	: Plastic Dual In Line Package
SPDIP	: Skinny PDIP
DSSS	: Direct-Sequence Spread Spectrum
CCK	: Complementary Code Keying
OFDM	: Orthogonal Frequency Division Multiplexing
CSMA/CA	: Carrier Sense Multiple Access/Collision Avoidance
Mbps	: Megabits per Second
DTE	: Data Terminal Equipment
DCE	: Data Communications Equipment
RS232	: Recommended Standard 232
IEEE	: Institute of Electrical And Electronics Engineers
SDA	: Serial Data Line
SCL	: Serial Clock Line
ICD	: In Circuit Debugger
PWM	: Pulse Width Modulation- Darbe Genişliği Modülasyonu
SPI	: Serial Peripheral Interface
USART	: Universal Asynchronous Receiver Transmitter:

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1 : Kullanılan Temel Davranışlar.....	37
Tablo 5.1 : Senaryoların Özet Gösterimi	40
Tablo 5.2 : Uygulama Katmanı Makro Komutları ve İşlevleri	46
Tablo 5.3 : Uygulama Katmanı Motor Komutları.....	46
Tablo 5.4 : Uygulama Katmanı Algılayıcı ve Konumlandırma Komutları.....	47
Tablo 5.5 : Yol Bulma Algoritması Pseudo Kodu	50
Tablo 6.1 : Motor Kontrol Kartı Yazmaç Adresleri.....	68
Tablo 6.2 : Kızılötesi Mesafe Algılayıcılar Özellik Tablosu	85
Tablo 6.3 : SRF10 Yazmaç Adresleri	88
Tablo 6.4 : SRF08 Yazmaç Adresleri	88
Tablo 6.5 : Sp03 Yazmaç Komutları.....	98
Tablo 7.1 : Yol Bulma Algoritması.....	104
Tablo 7.2 : Yol Koordinatlarının Bulunması	105
Tablo 7.3 : Çevre Algıla Komutu Pseudo Kodu	109
Tablo 7.4 : Dönüş Komutu Pseudo Kodu	110
Tablo 7.5 : İlerle Komutu Pseudo Kodu	111
Tablo 7.6 : Komut Gönderme Yordamı	112
Tablo 7.7 : Kızıl Ötesi Algılayıcı Yordamı.....	115
Tablo 7.8 : Ultrasonik Mesafe Algılayıcı Yordamı	116
Tablo 7.9 : Elektronik Pusula Yordamı.....	118
Tablo 7.10 : İvme Ölçer Yordamı	118
Tablo 7.11 : Hareket Algılayıcı Yordamı.....	119
Tablo 7.12 : Kesme Yordamı	120
Tablo 7.13 : Dc Motor Hız Yordamı.....	121
Tablo 7.14 : Motor Kontrol Komutları.....	122
Tablo 7.15 : Robot Hareket Komutları.....	123
Tablo 7.16 : Servo Konumlandırma Komutu.....	124
Tablo 7.17 : Konuşma Sentezleyici Yordamı	125

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 : 3 Katmanlı Hiyerarşik Mimari	3
Şekil 4.1 : Hareket Metodları.....	31
Şekil 4.2 : Dijkstra Algoritması	32
Şekil 4.3 : BFS Algoritması.....	32
Şekil 4.4 : Karmaşık Ortamda Dijkstra Algoritmasının Çalışması	33
Şekil 4.5 : Karmaşık Ortamda BFS Algoritmasının Çalışması	33
Şekil 4.6 : Hiyerarşik Kontrol Yaklaşımı	36
Şekil 4.7 : Davranış Temelli Yaklaşım.....	37
Şekil 5.1 : Robotun Kontrol Yapısı	41
Şekil 5.2 : Strateji Katmanının Senaryo 1 de Çalışması.....	42
Şekil 5.3 : Strateji Katmanının Senaryo 2 de Çalışması.....	43
Şekil 5.4 : Strateji Katmanının Senaryo 3 de Çalışması.....	43
Şekil 5.5 : Taktik Katmanının Senaryo 1 de Çalışması	44
Şekil 5.6 : Taktik Katmanının Senaryo 2 de Çalışması	44
Şekil 5.7 : Taktik Katmanının Senaryo 3 de Çalışması	45
Şekil 5.8 : Robotun Genel Yapısı	51
Şekil 6.1 : Genel Donanım Yapısı	58
Şekil 6.2 : Kontrol Sisteminin Yapısı	59
Şekil 6.3 : Easypic3 Kontrol Kartı.....	60
Şekil 6.4 : PIC 16F877A Pin İşlevleri	61
Şekil 6.5 : PIC 16F877A Blok Diyagramı.....	62
Şekil 6.6 : Hareket Sisteminin Yapısı.....	63
Şekil 6.7 : Diferansiyel Sürüş	64
Şekil 6.8 : Teker Hızlarının Hareket Yönüne Etkisi.....	64
Şekil 6.9 : Diferansiyel Sürüşte Keskin Dönüş	64
Şekil 6.10 : Diferansiyel Sürüşte Sabit Dönüş	65
Şekil 6.11 : Diferansiyel Sürüş Mekanizması Üstten Görünüş	65
Şekil 6.12 : Diferansiyel Sürüş Mekanizması Alttan Görünüş.....	66
Şekil 6.13 : Dc Motorlar ve Redüktör Dişli Kutusu	66
Şekil 6.14 : Motor Kontrol Kartı	67
Şekil 6.15 : Algılayıcı Tareti Önden Görünüş	71
Şekil 6.16 : Algılayıcı Tareti Yandan Görünüş	71
Şekil 6.17 : Algılayıcı Tareti Düşey Hareket Aralığı	72
Şekil 6.18 : Arka Algılayıcı Tareti	72
Şekil 6.19 : Kamera Tareti Üstten Görünüş.....	73
Şekil 6.20 : Kamera Tareti Alttan Görünüş	73
Şekil 6.21 : Kamera Tareti ve Kamera	73
Şekil 6.22 : Servo Motorda Darbe-Konum İlişkisi.....	75
Şekil 6.23 : Hitec 322HD Servo Motor	75
Şekil 6.24 : MPI MX-35 Mini Servo Motor	76
Şekil 6.25 : Futaba S3001 Servo Motor.....	77
Şekil 6.26 : Futaba Servo Motor Modifikasyonu	77

Şekil 6.27 : SD21 Servo Motor Kontrol Kartı	77
Şekil 6.28 : Haberleşme Sistemi Genel Yapısı	80
Şekil 6.29 : Wibox Kablosuz Seri Port Yönlendirici	80
Şekil 6.30 : WAG54G Kablosuz Yönlendirici	80
Şekil 6.31 : Seri Tünelleme İşlemi	82
Şekil 6.32 : Kızılötesi Algılayıcıların Çalışma Prensibi	83
Şekil 6.33 : GP2D12 Algılayıcısı için Mesafe Voltaj Grafiği	84
Şekil 6.34 : Kızılötesi Algılayıcıların Montajı	85
Şekil 6.35 : Ultrasonik Mesafe Algılayıcılar	87
Şekil 6.36 : SRF08 Dalga Deseni	90
Şekil 6.37 : SRF10 Dalga Deseni	90
Şekil 6.38 : CMPS03 Pin Bağlantı Şeması	91
Şekil 6.39 : KMZ51 Basit Devre Şeması	91
Şekil 6.40 : AccelBoard İvme Ölçer	93
Şekil 6.41 : Pasif Kızıl Ötesi Hareket Algılayıcı	94
Şekil 6.42 : DS18B20 Dijital Termometre	95
Şekil 6.43 : Dokunma Algılayıcı Mikro Anahtar	95
Şekil 6.44 : Kablosuz IP Kamera	96
Şekil 6.45 : SP03 Bağlantı Şeması	97
Şekil 6.46 : SP03 Yonga Yüzü	97
Şekil 6.47 : SP03 Hoparlör Yüzü	97
Şekil 6.48 : Güç Sistemlerinin Yapısı	99
Şekil 7.1 : Pathfinder Formu	101
Şekil 7.2 : Robot Kontrol Terminali	101
Şekil 8.1 : Yol Bulma Simülasyonu	127

YAPAY ZEKA İLE MOBİL ROBOT KONTROLU

Özet

Dünya’da hızla gelişen bir araştırma alanı ve sanayi sektörü olan robotik alanının günümüzde en büyük araştırma alanlarından biri otonom olarak çalışabilen robotların tasarlanmasıdır. Verilen görevleri insan müdahalesini en aza indirerek gerçekleştirebilen robotlar, sanayide ve bilimsel araştırmalarda maliyetleri ve verimliliği artırmak için günümüzde en çok tercih edilen yöntemdir. Bu gelişmelerin ışığı altında bu çalışmada ön tanımlı bir ortamda ızgara tabanlı bir harita kullanarak kendisine verilen görevleri yerine getirebilen otonom bir mobil robot için donanım ve yazılım alt yapısının tasarlanması ve robotun prototip üretimi amaçlanmıştır. Geliştirilecek olan robotun yapay zeka tekniklerini kullanması ve ileride eklenebilecek yazılım ve donanımlar için esnek bir yapı barındırması hedeflenmiştir.

Çalışmanın ilk bölümlerinde robot ve yapay zeka kavramlarına değinilmiş ardından yapay zekanın robotik alanında kullanımı incelenmiştir. Mobil bir robotun gereksinimleri, yol bulma, rota planlama, engel algılama engelden sakınma ve çarpışma algılama ilgili önemli noktalar vurgulandıktan sonra otonom mobil bir robot için davranış temelli ve hiyerarşik kontrol yaklaşımlarının ortak özelliklerini barındıran hibrid bir kontrol yaklaşımı geliştirilmiştir. Bu kontrol yaklaşımının donanım ve yazılım ile ilgili boyutları irdelenerek mobil robotun görevleri, özellikleri ve genel yapısı belirlenmiştir.

Robotun yol bulma gereksinimi için mevcut yol bulma algoritmaları incelenerek A* ve Dijkstra algoritmalarının özelliklerini bir arada barındıran bir yol bulma algoritması geliştirilerek yazılımsal olarak gerçekleştirilmiştir.

Yapılan teorik çalışmaların ardından diferansiyel sürüş tekniğini kullanan ve üzerinde algılayıcılar, kontrol kartları, haberleşme, görüntüleme ve elektro mekanik sistemleri barındıran robotun donanım projesi ve prototip üretimi gerçekleştirilmiştir. Robotun üzerinde 3 ayrı kontrol kartı, çeşitli amaçlar için kullanılan algılayıcılar, haberleşme sistemi, lcd ekran, konuşma sentezleyici, kablosuz kamera ve çeşitli ebat ve özelliklerde motorlar ve mekanik sistemler bulunmaktadır.

Geliştirilen prototip üzerinde tasarlanan kontrol yapısının oluşturulması için ana bilgisayarda ve robot üzerindeki ana kontrol kartında çalışacak programlar yazılarak, robotun tasarlanan senaryolar üzerinde çalışması sağlanmıştır.

MOBILE ROBOT CONTROL WITH ARTIFICIAL INTELLIGENCE

SUMMARY

One of the largest fields of research areas of robotics, which is a world-wide, rapidly improving research area and manufacturing sector, is the projection of robots that are able to operate autonomously. The robots, which are able to complete assigned tasks with the least amount of human intervention, are today's mostly preferred methods in order to be able to increase productivity and decrease costs in manufacturing and scientific research. In the light of these developments, projection of hardware and software infrastructure and prototype production of a robot have been aimed for the autonomous, mobile robot that is able to complete assigned tasks, using a grid-based map, in a pre-defined and structured working area. Usage of artificial intelligence techniques and a flexible structure for software and hardware have been aimed.

During the first sections of the study, robot and artificial intelligence concepts are explained; afterwards, use of artificial intelligence in robotics has been examined. After emphasizing the important issues, regarding a mobile robot's needs, its path finding, route planning, obstacle detection, obstacle avoidance and collision detection requirements, a hybrid control approach, that has common features of behavior-based and hierarchic control approaches has been developed. Having explicated software and hardware dimensions of this control approach, tasks, features and general structure of the mobile robot have been determined.

Having examined the current path finding algorithms for the robot's path finding needs, path finding algorithm which has A* and Dijkstra algorithms features, has been developed and implemented as software.

After the theoretical studies, robot's software project and prototype robot production, with differential drive, sensors, control cards, communication, imaging and electro-mechanic systems have been implemented. On the robot there are 3 separate control cards, sensors used for various purposes, communication systems, LCD display, speech synthesis equipment, wireless camera, engines and mechanical systems with various features and sizes.

In order to be able to develop a control structure that is designed on the prototype, programs that will work in the main control card on the robot and main computer have been developed and the robot was ensured to be working on the defined scenarios.

1 GİRİŞ

Robot kavramının ilk ortaya atıldığı günden bu yana robotlar kendi başlarına hareket edebilen ve düşünebilen makineler olarak vurgulanmıştır. İlk uygulamalar programlanabilir endüstriyel robotların üretimi ve manipülatörler şeklinde olmuşsa da günümüzde robotik bilimi robotların kendi başlarına (otonom) çalışmaları konusu üzerine çalışmaya devam etmektedir.

Robot manipülatörler günümüzde endüstride insanların çalışmasına uygun olmayan ve çalışmasının tercih edilmediği ortamlarda hizmet vermektedir. Endüstride montaj, taşıma gibi alanlarda kullanılan robotların bir bölümü sabit iken bir bölümü yer değiştirebilen (mobil) bir yapıya sahiptir. Ancak bu robot manipülatörler programlandıkları hareketler dizisi dışında hareket etme yeteneğine sahip değildirler. Her bir görev için yeniden programlanma ihtiyacı robotik bilimini verilen göreve göre kendi hareket kararlarını üretebilen robotlar üzerinde çalışmaya yöneltmiştir. Mobil robotlar olarak tanımlanan bu alt robot grubun üyeleri yapısal ve yapısal olmayan ortamlarda otonom olarak hareket ederek verilen görevi yerine getirmeyi amaçlamaktadır.

Özellikle uzay araştırmaları, askeri alanlar, gözetleme, arama kurtarma çalışmaları vb konularda hizmet vermesi planlanan mobil robotlar robotik biliminin son yıllardaki en önemli araştırma alanlarından biridir.

Bir robotun otonom olarak hareket edebilmesi için gerekli olan şartlardan en önemlisi robotun çevresini algılama, engelleri tanıma ve kendisine bir yol belirleyebilme özelliklerine sahip olmasıdır. Bu işlevselliğin sağlanabilmesi için robotun çevresini algılamasına yeterli algılayıcıları barındırması gerekmektedir. Algılayıcı verileri işlenerek çevre haritası oluşturulmakta ve ardından engellerden sakınarak hedefe ulaşılması hedeflenmektedir.

Bu işlem süreci sırasında tüm bu işlemlerin koordinasyonu ve yönetimi için gerekli kontrol altyapısının oluşturulması önemli gereksinimlerden biridir. Bu kontrol altyapısı robotun karar verme işlemlerini yerine getireceği gibi aynı zamanda robotun donanım işlevlerini de kontrol etmekle yükümlüdür.

Çoğu mobil robot uygulamasında insanların gözetimi ve uzaktan yönetimi de uygulamaya dahil edilmiş durumdadır. Bazı kritik uygulamalarda ise tüm kontrol insan operatörler tarafından gerçekleştirilmektedir. Günümüzdeki çalışmalar ise insan müdahalesini minimuma indirmek veya tamamen ortadan kaldırmak üzerine yoğunlaşmıştır. Bunun için yapay zeka metodları kullanılmaktadır. Robotun kontrol altyapısı yapay zeka yöntemleri kullanarak geliştirilmekte ve gerekli durumlarda operatörlerin müdahalesine izin verilmektedir.

Günümüzde endüstriyel ve bilimsel uygulamaların yanı sıra ticari olarak hizmete girmiş kendi başına hareket edebilen robotlar bulunmaktadır. Honda'nın geliştirmiş olduğu insansı robot asimo, irobot'un geliştirdiği temizlik robotları roomba ve scooba ve sony'nin geliştirdiği aibo ve quiro bu robotların örneklerindendir.

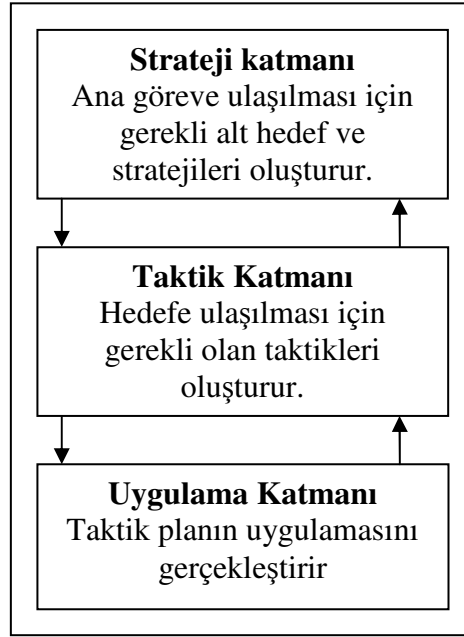
1.1 Konu ile İlgili Çalışmalar

Günümüzde ileri robotik kavramı yüksek seviyeli ve görev bazlı komutları alıp, navigasyonu ve kompleks görevleri en az insan müdahalesi ile yerine getiren makina veya sistem olarak tanımlanmaktadır. İleri robotik bilimi genellikle algılayıcı verilerini baz alan kontrol mekanizmaları, otomatlar ve bir önceki tanımda verilen görevlerin yerine getirilmesi için gerekli fonksiyonelliğin sağlanması amaçlı uygun sistem mimarileri üzerine çalışmaktadır.

Mobil robotların kontrol yöntemleri ve sistem mimarilerinin geliştirilmesi ile ilgili çok sayıda çalışma yapılmıştır [1-14]. Mevcut kontrol yöntemleri hiyerarşik ve davranış temelli olmak üzere ikiye ayrılmaktadır [2, 5].

Geleneksel hiyerarşik yaklaşım, mevcut standart hesaplama yöntemleri ve arayüzleri, sinyal senkronizasyon protokolleri ve sonlu durum tekniklerinin sağladığı saydamlık nedeniyle endüstriyel alanda büyük oranda kabul görmüş bir yaklaşımdır. Gerçek zamanlı zeki kontrol için hiyerarşik mimarinin tanımı Albus [2] tarafından yapılmıştır. Bu yaklaşımın tipik bir uygulaması Şekil 1.1'de gösterilmiştir. Bu yapıda en alt veya uygulama katmanı gerçek zamanlı kontrol işlevlerini yerine getirmekte ve taktik katmanının en üst katmandan aldığı komutları yorumlayarak oluşturduğu makro komutları işlemektedir. En üst katman olan strateji katmanı durum analizi, planlama ve görev yorumlaması fonksiyonları nedeniyle en karmaşık yapıyı barındıran katmandır. Bu fonksiyonların yerine getirilmesi az veya çok gerçek

çevrenin modelinin bilinmesini gerektirmektedir. Bu model algılayıcılar tarafından oluşturulabileceği gibi operatörler tarafından da tanımlanabilmektedir.



Şekil 1.1 : 3 Katmanlı Hiyerarşik Mimari

Endüstride yaygın olarak kullanılmasına rağmen, hesaplama gereksinimleri ve katmanlar arasındaki haberleşme yükü oldukça büyük olabilmektedir. Çevre modelinin güncel tutulması, genellikle yapısal olmayan ve önceden kestirilemeyen çalışma alanının gösterim metodları bu tür kontrol yaklaşımının en önemli araştırma alanlarını oluşturmaktadır.

Son yıllarda ise alternatif olarak geliştirilen davranış temelli veya reaktif yaklaşım giderek ivme kazanan bir kontrol yaklaşımıdır [5]. Bu yaklaşımda ana görev, görevin yerine getirilmesinde kullanılacak davranış ve tavırlara indirgenmektedir. Bu tavır ve davranışlar temel fonksiyonları yerine getirme veya diğer davranışların çalışmasını destekleme gibi görevler üstlenebilmektedir. Oluşturulan her bir davranış genellikle bir katman olarak mimariye dahil edilmektedir. Her bir katman barındırdığı öncelik seviyesi ile kendinden önceki veya sonraki katmanların davranışlarını etkileyebilmekte veya bastırabilmektedir. Bu yaklaşımın sağlamış olduğu mimariye sonradan davranışların eklenebilmesi esnekliği kabul görmesinde büyük rol oynamaktadır. Robotun genel davranışı her bir katmanın birbiri ile etkileşimi sonucu belirlenmektedir. Çevre modelinin katmanlar arasında paylaşımına ihtiyaç duyulmamakta; katmanlar arası iletişim düşük bir bant genişliği ve basit bir protokol ile gerçekleştirilebilmektedir. Her bir katman görece basit bir hesaplama

gücü ile çalışabilmektedir. Bu yapı paralel çalışmaya izin vermekte ve hesaplama yükü genellikle düşük seviyelerde seyretmektedir. Mimarının modülerliği, genişleyebilir bir sistem yapısının oluşturulmasına olanak tanımakta; ön tanımlı olmayan ve genellikle önceden kestirilemeyen ortam koşullarında sistemin çalışabilmesini sağlamaktadır.

Basit davranış modüllerinin bileşiminden karmaşık davranışlar oluşturulabilmesine [6] rağmen modülerin veya katmanların sayısı arttıkça birbirini baskılayan veya etkileyen düğüm noktalarının da sayısı artmakta; karmaşıklığa ve çakışmaların çözümünün zorlaşmasına neden olmaktadır. Genel davranış yapıları çoğunlukla elle tasarlanmakta ve karmaşıklık arttıkça öngörülemeyen ve analiz edilemeyen kombine davranışların oluşması kaçınılmaz olmaktadır.

Hiyerarşik yaklaşımın mantıksal fonksiyon ayrıştırması, veri akışı ve fonksiyonların katmanlar arasında paylaşımı özellikleri ile davranış temelli yaklaşımın çevre ile aktif etkileşim sonucu oluşan davranışlarının kompleks görevleri yerine getirmesi özelliği robot kontrol dizaynı paradigmalarının iki zıt kutbunu oluşturmaktadır.

Konu ile ilgili literatür farklı yaklaşımların [7] yanı sıra her iki yaklaşımın avantajlarını tek bir çatı altında toplamaya çalışan [8, 9] bir çok örneği barındırmaktadır. Bu çalışmada ise hiyerarşik ve davranış temelli yaklaşımların ortak özelliklerini barındıran hibrid bir kontrol mimarisi oluşturulmuştur.

Her iki yaklaşımında çözümlemeye çalıştığı ana problemlerden biri çevre modelinin oluşturulması ve navigasyondur. Bu alan ile ilgili birçok araştırma yapılmış ve günümüzde de yapılmaya devam etmektedir [1, 11-16]. Bu alanda en çok kullanılan yöntemler ızgara tabanlı [11, 16] ve özellik tabanlı [12, 14] haritalamadır.

İzgara tabanlı haritama yöntemi ilk olarak Moravec ve Elfes [17, 18, 19] tarafından kullanılmıştır. Bu yöntem çalışma ortamının, 2 boyutlu hücre adı verilen karelere bölmektedir. Hücreler robot navigasyonu açısından çok önemli olan işgal (doluluk, boşluk), tehlike, ulaşılabilirlik [18] gibi bilgileri içermektedir. Hücrelerin işgal bilgilerini kullanarak robotlar, bulundukları ortamın haritasını çıkartabilmekte veya engellere çarpmadan hedefe gidebilen navigasyon algoritmalarını gerçekleyebilmektedir. Bilgisayar üzerinde gerçekleştirilebilmelerinin ve işlenmelerinin görece daha basit olmasından dolayı bu çalışmada ızgara tabanlı harita sistemi kullanılmaktadır.

Çevre haritasının oluşturulması veya robota verilmesinin ardından ikinci önemli sorun olan yol bulma sorunu ile karşılaşilmektedir. Engellere çarpmadan hedefe ulaşmak için çeşitli yöntemler öne sürülmüştür [20, 21, 22, 23]. Bu yöntemlerden günümüzde en çok kullanılanları Edsger Dijkstra'nın geliştirdiği Dijkstra algoritması [22] ile Hart, Nilsson ve Raphael'in geliştirdiği A*¹ algoritmasıdır [20, 21]. Her iki algoritma da matematikte kullanılan graf teorisinin “tek kaynaklı en kısa yol”² problemini çözmek amacı ile geliştirilmişlerdir. Bu problemin robotik alanına uygulanması ise başlangıç noktasından bitiş noktasına en kısa yolu bularak gitmek olarak özetlenmektedir. Önerilen her bir yöntem işlem zamanı, hafıza gereksinimleri ve en kısa yolun bulunması oranı ile birbirine göre avantaj ve dezavantajlara sahiptir. Robot navigasyonunda ızgara tabanlı haritalarda kullanıldıklarında özellikle büyük haritalarda hafıza gereksinimi ve işlem süresi problemlere neden olabilmektedir. Bu çalışmada A* ve Dijkstra algoritmalarının avantajlarını barındıran bir yöntem kullanılmıştır.

1.2 Projenin Amacı ve Yapısı

“Yapay zeka ile mobil robot kontrolü” projesinde ızgara tabanlı bir haritada çevresini algılayarak otonom olarak veya uzaktan kontrol ile hareket edebilen bir mobil robotun tasarım ve prototip üretimi amaçlanmaktadır.

Mobil robotlar konusunda yapılan çalışmalar göz önüne alınarak bu projede insan müdahalesi olmadan ön tanımlı bir ortamda otonom olarak hareket eden bir robotun geliştirilmesi planlanmıştır. Kendisine verilen görevleri yerine getirebilen ve taktiksel kararlar alabilen zeki bir kontrol mekanizmasını barındıracak olan bu robotun aynı zamanda insan operatörler tarafından kontrol edilebilecek şekilde tasarlanması kararlaştırılmıştır. Robotun taktiksel kararlar alabilen ve zeki davranışlar sergileyebilen bir yapıda olması nedeniyle İngilizce Intelligent Tactical Unit - Robot kelimeleri kısaltılarak robota ITUbot ismi verilmiştir.

Proje sonunda, belirlenen senaryolar dahilinde çalışacak, bir harita üzerinde verilen hedefe ulaşmak için çevresini algılayıp gerekli olan yolu hesaplayacak bir robotun prototip üretimi gerçekleştirilecektir.

¹ A Star olarak okunur.

² İng: single-source shortest path problem

Hedeflenen otonom hareket kabiliyetinin operatör tarafından en az müdahaleyi gerektirecek şekilde tasarlanması ve uygulanması proje sırasında gerçekleştirilmeye çalışılacaktır.

Tasarım sırasında mümkün olan en ölçeklenebilir ve genişlemeye müsait yapı sağlanacak, ileride eklenebilecek algılayıcı ve donanımlar için uygun alt yapı oluşturulacak, belirlenen senaryolar dışında yeni senaryolara adapte edilebilen donanım ve yazılım bir yapısı sağlanarak robotun başka görevleri de yerine getirebilmesi için gerekli hazırlıklar yapılacaktır.

Prototipin tamamlanmasının ardından belirlenen senaryolar üzerinde çalıştırılması ve performans değerlendirilmesi gerçekleştirilecektir.

Tüm bu amaçlar ışığı altında başlayan proje aşağıdaki şekilde yapılandırılmıştır.

Projenin ilk bölümlerinde mobil robotlar ve yapay zeka ile ilgili çalışmalar hakkında genel bilgi verilmektedir. Bu çalışmanın ikinci bölümünde robot kavramı ve robotik bilimin tarihi ile ilgili bilgi verilmekte ardından üçüncü bölümde yapay zeka kavramı ele alınmaktadır. Dördüncü bölümde ise yapay zeka ve robot ilişkisi, mobil bir robotun gereksinimleri ve mobil robotların yönetiminde kullanılan yaklaşımlara değinilmektedir. Beşinci bölümde ise “Yapay zeka ile mobil robot kontrolü” projesinin genel yapısı, hedefler ve görevler detaylı bir şekilde incelenmektedir. Takip eden bölümde ise robotun uygulamasında kullanılan temel teknolojiler hakkında kısa bir bilgi verildikten sonra robotun donanım yapısı detaylı olarak anlatılmaktadır. Yedinci bölümde robotun yazılımsal yönleri incelenmekte ana bilgisayar ve robot üzerinde çalışan kontrol programlarının önemli yordamlarına değinilmektedir. Son bölümde ise yapılan simülasyon ve test sonuçlarına yer verilecek; proje süresinde ulaşılan hedeflere ve yapılabilecek geliştirme çalışmalarına değinilecektir.

2 ROBOTİK

2.1 Robot Tanımı

Robot teknolojisi, çağımız gelişim süreci içinde gelişen birçok bilimsel ve teknolojik olguların, robot adını verdiğimiz teknolojik ürünler üzerinde bütünleşmesi ve uygulamasını içerir.

Robotik, Makine Mühendisliği, Elektrik ve Elektronik Mühendisliği ve Bilgisayar Mühendisliği disiplinlerinin ortak çalışması sonucu doğmuş bilim dalıdır

Robot kelimesi Çekçe'deki köle, hizmetçi veya zorunlu işçi anlamına gelen robota veya robotnik kelimelerinden türemiştir.

Robot kelimesi ilk olarak 1920'lerin başında yazılmış bir oyunda karşımıza çıkar. Karel Čapek'in R. U. R. adlı oyununda mekanik ve otonom, ama insanca duygulardan yoksun yaratılar olarak kullanılan robot, daha sonra bir çok bilim kurgu romanına konu olmuştur. Bilimkurgu yazarı Isaac Asimov, ünlü robot serisiyle teknolojik açıdan tutarlı bir robot kavramı yaratır ve robotların amacının insana hizmet olduğunu, bir robotun kendi amaçlarını insanların amaçlarına hiçbir zaman tercih edemeyeceğini ortaya koyduğu üç Robot Yasası ile belirler. Bu robot yasaları şu anda insan ile robot arasındaki ahlaksal ve hukuksal ilişkinin temelini oluşturmaktadır ve robot biliminin gelişiminde önemli bir nokta olarak kabul edilmektedir. Üç robot yasasını kısaca özetlemek gerekirse;

- Bir robot hiçbir şekilde insanoğluna zarar veremez; veya pasif kalmak suretiyle zarar görmesine izin veremez.
- Bir robot kendisine insanlar tarafından verilen emirlere 1. Kural ile çelişmediği sürece itaat etmek zorundadır.
- Bir robot 1. ve 2. Kurallar ile çelişmediği sürece kendi varlığını korumak zorundadır.

Asimov ardından listeleme sıfırıncı kuralı eklemiştir. Bu kural “bir robot insanlığa zarar veremez ve hareketsiz kalarak insanlığın zarar görmesine izin veremez”

şeklindedir. Ancak günümüzde 3 robot yasasının tamamen uygulandığı bir robot bulunmamaktadır.

Robot deyince ilk akla gelen insan benzeri makinelerdir. Oysa dünyadaki robotların çok azı insana benzer. Çoğu robot tasarlandığı görevin gereksinimleri karşılayacak şekilde sahiptir. Kartezyen, Scara, küresel, silindirik yapıya sahip robot kolların yanı sıra insan benzeri Humanoid³ robotlar da bulunmaktadır. Tüm bu robotlar ortak olarak 4 ana öğeyi barındırırlar

Efektörler: Kol, bacak, el, tekerlekler vb gibi hareket mekanizmaları

Algılayıcılar: Robotların duyu organları olarak çalışan, nesneler, ısı, ışık vb nesnel bilgileri robotların anlayabileceği nümerik verilere çeviren parçalar

Kontrolör: Robotun kontrolünü sağlayan algoritmaları barındıran elektronik devreler veya bilgisayarlar

Ekipmanlar: Robota bağlı olan araçlar ve mekanik aksam

Robotları diğer makinelerden ayıran karakteristik özellik ise robotların genellikle kendi başına çalışmaları, bir görevi yerine getirebilmek için programlanabilmeleridir.

Günümüzde de robotların en büyük kullanım alanı endüstriyel üretimdir. Özellikle otomotiv endüstrisinde çok sayıda robot kullanılır. Bunların çoğu insan hareketlerini taklit eden manipülatör robotlardır. Bunlara örnek olarak parçaları monte eden, birleştiren, kaynak ve boya yapan robotlar verilebilir.

Gelişen teknoloji ile birlikte evlerde de robot kullanımı giderek artmaktadır. Evlere giren ilk robotlar Furby, AIBO, Robosapien gibi oyuncaklar olmakla beraber başta ABD'de olmak üzere dünya genelinde ev işlerine yardımcı olan robotların kullanımı giderek yaygınlaşmaktadır. Örneğin Irobot firmasının ürettiği yerleri kendi kendine süpüren robot elektrik süpürgeleri büyük talep görmektedir.

Robotların gelişim sürecinde insanlardan ve diğer canlılardan etkilenecek sistemler geliştirilmiş ve insana benzemeleri veya insanların yaptıklarını yapabilmeleri hedeflenmiştir. Günümüzde robotlar çok farklı alanlarda hizmet vermektedir.

³ İnsansı robot

2.2 Robotların Kullanım Alanları:

Robotlar ilk olarak endüstride kullanılmaya başladıkları günden bu yana kendilerine çok farklı çalışma ortamları bulmuşlardır. Otomotiv fabrikalarının montaj hatları ile başlayan robotların gelişimi günümüzde sayısız uygulama ile çok geniş bir yelpazeye ulaşmıştır.

Endüstriyel uygulamalardan, operasyonel uygulamalara, tıbbi uygulamalardan ev işlerine kadar kullanım alanı olan robotlar, gelişen teknoloji ile her geçen gün daha gelişmiş yeteneklere sahip olmaktadır.

2.2.1 Endüstriyel Robotik

Sanayide montaj, boyama kaynak vb işlemlerde kullanılırlar; genellikle hantal ve sabit konumludurlar. Farklı işler için programlanabilmeleri üretim akışının hızlanması ve montaj bantlarının yeni ürünlere adaptasyon kolaylığını sağlar. Genellikle insan kolunun hareketlerini taklit ederek bir işçinin yerine getireceği görevleri yerine getirirler. Sahip oldukları mekanik güç ve hareketlerinin yinelenebilirliği en büyük avantajlarıdır. Genellikle hareket mekanizmaları hassasiyeti sağlamak için step motor veya servo motorlara dayanır ve her bir eklemin konum kontrolü genellikle kapalı kontrol çevrimleri ile denetlenir.

2.2.2 Operasyonel Robotik

Genellikle insanların çalışmasına elverişli olmayan veya çalışmasının tercih edilmediği ortamlarda çalışmak üzere tasarlanmışlardır. Özellikle radyoaktif ortamlar, uzay araştırmaları, deniz dibi araştırmaları, boru hatlarının incelenmesi, arama kurtarma çalışmaları gibi konularda hizmet verirler. Genellikle otonom kontrol yerine uzaktan tele operasyon ile yönetilirler. Hareket mekanizmaları servo veya dc motorların yanı sıra hidrolik ve pnömatik unsurları da içerir. Özellikle son yıllarda uzay çalışmalarında kullanılan robotlar daha tanınır hale gelmiştir. Mars yüzeyinin incelenmesi için kullanılan Pathfinder, Sojourner, Spirit ve Opportunity robotik biliminin en gelişmiş ürünlerini ve teknolojilerini barındırmaktadır.

2.2.3 Tıp ve Sağlıkta Robotik

Sağlık alanında robotlar iki büyük uygulama alanı bulurlar. Bunlardan birincisi ortopedi diğeri ise uzaktan ameliyat uygulamalarıdır. Ortopedide uzuvları eksik olan hastaların bu uzuvlarının işlevlerini yeniden kazanmalarında robotik protezler

kullanılır. Bu protezler piezo-elektrik algılayıcıları ile sinirlerdeki veya tendonlardaki komutları algılayarak bu komutları harekete dönüştürmektedirler. Robot kol, bacak ve eller en yaygın biçimleridir. Ancak maliyetleri nedeniyle şu anda sınırlı bir kullanıma sahiptirler. Diğer bir uygulama alanı ise uzaktan ameliyat metodudur. Çok hassas step motorlar ve konumlandırma metodları ile çalışan bu robot cerrahlar kıtalararası iletişim ile cerrahların başka bir kıtadaki ameliyata katılabilmelerini sağlamaktadırlar.

2.2.4 Askeri Robotik

Robotlar insanlar için tehlikeli ortamlar olan savaş alanlarında da boy göstermektedirler. İnsansız keşif uçakları ve keşif araçları bu konuda en büyük araştırmaların yürütüldüğü alanlardan biridir. Özellikle Amerikan ordusu tarafından desteklenen bu çalışmaların sonucu olarak predator casus uçakları günümüzde haber alma ve keşif amaçlı olarak sahada kullanılmaktadır. Aynı zamanda insansız saldırı araçları üzerine de çalışmalar devam etmektedir. Çoğu askeri robot varolan sistemlerin kontrolünün bilgisayar ve yapay zeka metodları kullanılarak insansız yönetilmesi esasına dayanmaktadır [26-27].

2.2.5 Bilimsel Robotik

Genellikle operasyonel alanlarda ve robotik araştırmalarında yer alan robotlar bilim adamlarının ve mühendislerin yeni teknikler geliştirmelerine yardımcı olmaktadır. Başlıca çalışma alanları sensör verilerinin değerlendirilmesi, haritalama ve yol planlama gibi robotların çalışmalarını daha efektif hale getirecek konulardır. Günümüzde birçok üniversite ve araştırma merkezi bilimsel robotlar kullanmakta ve tasarlamaktadır.

2.2.6 Sinemada Robotik

Çoğu konuda olduğu gibi sinemada da robotik yerini almıştır. Filmlerde kullanılan efsane robot karakterlerinin yanı sıra çekimleri çok zor olan sahnelerde robot aktörlerin kullanımı yaygın bir durumdur. Genellikle efekt şirketlerinin teknik bilgilerini ticari sır olarak görmeleri nedeniyle bu konuda kullanılan robotlar hakkındaki bilgi çok kısıtlıdır.

2.2.7 Hobi ve Eğitim Amaçlı Robotlar ve Robot Oyuncaklar

Robotik bilimin gelişmesi ile oyuncak robotların üretilmesi de gündeme gelmiştir. Karmaşık elektronik ve mekanik sistemlerin düşük maliyetle üretilebilmesi oyuncak robotların üretilebilmesini mümkün kılmıştır. Sibernetiğin teorik araştırmaları, ilk ticari ürünlerini bu alanda vermiştir. Furby adlı oyuncakın ardından, Sony' nin Aibo robot köpeğini [28], ünlü robot araştırmacısı Mark Tilden' in Robosapien [29] ve roboraptor robotları takip etmiştir. Canlıların hareketlerini ve tepkilerini taklit eden bu robotlar aynı zamanda üniversitelerde robotik eğimi için de kullanılmaktadır. Dünya genelinde birçok organizasyon tarafından robot yarışmaları düzenlenmektedir. Bunlardan en önemlileri Sony Aibo robotlar ile futbol turnuvaları, Lego Mindstorm robot tasarım yarışmaları, labirent çözme, yangın söndürücü (firefighter) yarışmalarıdır. Aynı zamanda iki robotun karşılıklı yarıştığı robot sumo güreşleri ve robot savaşı (battlebot) gibi organizasyonlarda bulunmaktadır. ABD Savunma Bakanlığı İleri Araştırma Projeleri Ajansı DARPA' nın düzenlediği ve otonom robotların çölü aşmaya çalıştıkları DARPA Challenge adlı dayanıklılık yarışı da önemli organizasyonlardan biridir. Ülkemizde ise ODTÜ Robot Günleri, “First Lego League” ve “Robocon” üniversite öğrencileri tarafından büyük ilgi görmektedir.

Robotların günümüzde gelmiş olduğu bu noktanın daha iyi anlaşılabilmesi için robotların gelişiminin incelenmesinde fayda vardır.

2.3 Robotların Gelişimi

Robot bilimi eski çağlara kadar dayanmaktadır. Günümüz robot biliminin temelleri ise 20. yüzyılın başlarında atılmıştır. Makine , Elektrik, Elektronik ve Bilgisayar bilimlerinin bir birleşimi haline gelen günümüz robot biliminin tarihini incelerken bu alanlardaki önemli olayları da dikkate almak faydalı olacaktır.

Batı literatürüne bakıldığında robotlar ile ilgili en eski bilgi M.Ö. 300'lü yıllara dayanır. Yunan matematikçi Archytas'ın buharla çalışan bir güvercin yaptığı söylenmektedir. Ancak robotik ile ilgili bilinen en eski yazılı eser Anadolu'ya aittir. 1136-1206 yıllarında yaşayan ve Artuklu sarayında mühendislik yapan El-Cezeri zamanının çok ötesinde teknoloji içeren ve otomatik olarak çalışan düzenekleri hayata geçirmiştir. Aynı zamanda mühendislik üzerine yazdığı 6 bölümden oluşan Kitab-ül Cami Beyn-el -İlmi ve'l-Ameli en Nafi Fi Sinaati-l Hiyel (mekanik

hareketlerden mühendislikte faydalanmayı içeren kitap) mühendislik açısından çok önemli bilgiler içermektedir. Bu kitapta otomatik kuşlar, filli saat, otomatik yüzen kayık ve çalgıcılar, birbirine şerbet ikram eden iki şeyh, dört çıkışlı iki şamandıralı otomatik sistem, iki bölümlü testi (termos), otomatik su akıtma, ikramda bulunma ve kurulama makinesi, su çarkı kepçe mekanizması, motor-kompresör mekanizması, su çarkı su dolabı gibi mekanizmalara yer verilmiştir

1400'lü yıllarda robotik ve otomatik makineler üzerine çalışmalarda ise Leonardo Da Vinci'nin eserleri öne çıkar. Tasarladığı birçok otomatik makine arasında savaşçı robot'ta bulunmaktadır. Leonardo'nun "Savaşçı Robotu" kollarını çenesini ve başını hareket ettirebilmektedir.

18. ve 19. yüzyıllarda ise Avrupalı zenginler için üretilmiş olan birçok robot/otomat bulunmaktadır. Bu yıllarda Osmanlı sarayı için 1769 yılında Baron Von Kempelen tarafından geliştirilen "satranç oynayan adam"da bu robot/otomatlara güzel örneklerden biridir. Bu otomat viyana ve Moskova'daki fuarlarda sergilenmiş ancak içinde insan gizlendiği iddia edilmiştir. Yapısı ise bir zemberek'ten güç alan metal silindir ve üzerindeki kamlar aracılığıyla karmaşık olasılıkları hesaplayabilen bir mekanizmadan oluşmaktadır. Bu silindir ve kam mekanizması ileride Edison'a esin kaynağı olacak ve gramofonu icat etmesini sağlayacaktır.

1785 Yılında Pierre Kintzing tarafından yapılan ve bir tür vurmali akustik çalgı olan Harpsicord çalan kadın döneminin androidi sayılabilir. Bu gün Fransa'da müzede bulunan bu örnek de kurulan bir zemberekten güç almaktaydı. Bu otomatlar gerçekte bir çok müzik parçasını çalabilen karmaşık makinelerdi. Avrupa'nın bilgi birikimi, o çağda doruğa çıkmış olan saat yapıcılığı ve mekanik ustalığından ileri gelmekteydi. Çok küçük parçalar yapmakta ustalaşmış saat yapıcıları ve mekanik ustaları için otomat yapımı sarayda ve soylu çevrelerde kendilerini gösterebilecekleri eşsiz fırsatlardı.

Bu otomatların ardından 20 yüzyıla gelindiğinde günümüz robotiğinin temelleri atılmaya başlandı. Özellikle 1940 ve 1950'li yıllarda günümüzde dahi pratiğe geçirilememiş olan robot fikirleri ortaya atıldı. 1948'de transistörün bulunması ise elektronik ve bilgisayar biliminin çok hızlı gelişmesine olanak tanıdı. Bilgisayar ve yapay zeka biliminin gelişmesine paralel olarak öncüllerine nazaran çok üstün yeteneklere sahip robotlar geliştirilmiş, robotik bir bilim dalı ve popüler bir araştırma alanı haline gelmiştir.

Robotların gelişiminde önemli rol oynayan tarihi olaylar aşağıda listelenmiştir.

1801 Joseph Jacquard, punch kart⁴ sistemi ile kontrol edilen ilk otomatik dokuma makinesini yaptı. Punch kartlar 20. yy'da üretilen ilk bilgisayarların programlanmasında kullanılan önemli metodlardan biri haline gelmiştir.

1822 Charles Babbage, günümüz bilgisayarlarının temeli olan ikili sayı tabanı fikrini ortaya attı".

1847 George Boole, önerdiği mantıksal cebir ile mantığın matematiksel bir formda gösterimini yaptı. Günümüzde programlama dillerinde kullanılan mantık değişkenleri onun anısına boolean olarak isimlendirilmektedir.

1898 Nikola Tesla, Madison Square Garden'da uzaktan kontrol edilen bir teknenin tanıtımını yaptı.

1921 Robot terimi çek yazar Karel Capek'in "R. U. R. " (Rossum's Universal Robots) isimindeki oyununda kullanıldı. Oyunun senaryosu insan robotu üretir robot insanı öldürür şeklide özetlenebilir.

1926 Fritz Lang'in "Metropolis" adlı filmi vizyona girdi. Filmdeki "Maria" adlı bayan robot beyaz perdeye yansıyan ilk robottur.

1936 Alan Turing, Turing makinesi adı verilen teorik bilgisayar tasarımını öne sürdü.

1938 DeVilbiss firmasında çalışan Willard Pollard ve Harold Roselund tarafından ilk programlanabilir spreyci boyama mekanizması geliştirildi.

1940 Issac Asimov, "Super Science" dergisi için robotlar hakkında kısa öyküler yazmaya başladı.

1941 Bilimkurgu yazarı Asimov, robotik kelimesini robot teknolojisini tanımlamak için kullandı ve robot endüstrisinin hızlı gelişimini öngördü-

1942 – Asimov, "Runaround" adlı öyküsünde ünlü 3 robot kuralını tanımladı.

1946 George Devol, genel amaçlı makinelerin kontrolü için geliştirdiği manyetik kayıt sisteminin patentini aldı.

1946 J. Presper Eckert ve John Mauchly Pennsylvania üniversitesinde ENIAC adı verilen ilk elektronik bilgisayar ürettir.

⁴ Delikli kart

1946 MIT⁵'de Whirlwind adındaki ilk dijital genel amaçlı bilgisayar ilk problemini çözdü.

1948 Transistor icat edildi.

1950 Alan Turing, "Computing Machinery and Intelligence" adlı eserinde bir makinenin kendi başına düşünme yeteneğinin olup olmadığının anlaşılmasına yarayan düşünsel bir test önerdi. Bu test "Turing Test"i olarak bilinir.

1951 Fransa'da Raymond Goertz, atom enerjisi kurumu için ilk uzaktan kontrol edilen mafsallı kolu geliştirdi. Bu robot kol kaplinler ve makaralar ile kontrol edilmekteydi. Günümüzde dahi bu mekanizmanın türevleri nükleer araştırma laboratuvarlarında kullanılmaktadır. Goertz' in bu çalışması güç geri besleme⁶ teknolojisinin en önemli adımlarından biri kabul edilir.

1954 George Devol, ilk programlanabilir robotu geliştirdi ve evrensel otomasyon⁷ terimini kullandı. Bu terim ileride kuracağı Unimation adlı firmanın isminin temelini oluşturdu.

1956 - George Devol ve Joseph Engelberger Unimation isimli ilk robot şirketini kurdular.

1956 Rockefeller vakfı tarafından desteklenen John McCarthy, Marvin Minsky, Nat Rochester ve Claude Shannon " Dartmouth Summer Research Project on Artificial Intelligence" adı verilen konferansı gerçekleştirdiler ve "artificial intelligence"⁸ terimi ortaya çıktı.

1956 Alan Newell ve Herbert Simon ilk uzman sistem⁹ olan ve zor matematik problemlerinin çözülmesinde yardımcı olan "Logic Theorist" i geliştirdiler

1959 MIT Servomechanisms laboratuvarında ilk bilgisayar destekli üretim tanıtıldı

1959 John McCarthy ve Marvin Minsky MIT'de yapay zeka araştırma laboratuvarını kurdu.

⁵ Massachusetts Institute of Technology

⁶ İng: Force feedback

⁷ İng: Universal Automation

⁸ Yapay zeka

⁹ İng: Expert System

1960 Sonraları AMF olarak bilinen “American Machine and Foundry” firması Harry Johnson ve Veljko Milenkovic tarafından geliştirilen Versatran adındaki silindirik robotu piyasaya sürdü

1960 Unimation firması is Condec satın aldı ve Unimate robot sistemlerinin tasarımına başladı.

1961 UNIMATE adı verilen ilk endüstriyel robot, General Motors’un New Jersey’deki otomobil fabrikasında hizmete girdi. Bu robot kol üretim bandındaki tekrarlanan veya tehlikeli olan görevleri yerine getirmek için tasarlanmıştı.

1961 Heinrich Ernst MIT’de bilgisayar kontrollü robot bir el olan MH-1’i geliştirdi.

1963 Bilgisayar kontrollü ilk yapay robot kol tasarlandı. “Rancho Arm” adı verilen bu robot kol özürlü insanlar için bir araç olarak geliştirilmişti ve bir insan kolunun serbestlik derecesine sahipti.

1963 John McCarthy Stanford Üniversitesinde yapay zeka laboratuvarını kurmak üzere MIT’den ayrıldı

1964 M. I. T. , Stanford Araştırma enstitüsü, Stanford Üniversitesi, Edinburgh üniversitesi gibi önemli üniversitelerde yapa zeka laboratuvarları kuruldu

1964 C&D Robotics firması kuruldu.

1965 Uzman kişilerin bilgi birikiminden faydalanan ilk uzman sistem olan DENDRAL geliştirildi.

1965 Robot kinematiğine homojen dönüşümler uygulandı ve bu günümüz robot teorisinin temelini oluşturdu.

1966 ELIZA adı verilen yapay zeka programı MIT’de Joseph Weizenbaum tarafından geliştirildi. Eliza kullanıcının cümlelerinden sorular üreten bir bilgisayar psikolog olarak çalışmaktaydı.

1966 Stanford Araştırma enstitüsü (ileride SRI Technology olarak ismi değişmiştir) kendi hareketlerini bilen ve tepki verebilen Shakey isimli ilk mobil robotu geliştirdi.

1967 Japonya AMF’den ilk Versatran robotunu ithal etti. (bu robot Japonya’nın ithal ettiği ilk robottur)

1967 Richard Greenblatt, MacHack adında satranç oynayan ilk programı yazdı. Bu program ileride Santanç ustası Kasparov'u yenen BigBlue gibi programların yolunu açtı.

1968 Kawasaki Unimation firmasının hidrolik robot kolunun lisansını aldı ve Japonya'da üretimine başladı

1968 SRI görüş kapasitesine sahip mobil robot Shakey, 'i üretti

1968 Stanley Kubrick, ünlü bilim kurgu yazarı Arthur C. Clarke'ın, 2001: uzay macerası romanını filme aktardı. Bu filmde HAL adındaki bilgisayar çalışmak için insanlara ihtiyaç duymadığına karar vererek mürettebatı etkisiz hale getirir. HAL isminin IBM firmasının harflerinden bir önce gelen harflerden oluşması tartışma konusu olmuştur.

1969 - Stanford yapay zeka laboratuvarında çalışan Victor Scheinman adındaki makine mühendisliği öğrencisi ilk elektrikle çalışan ve bilgisayarla kontrol edilen robot kol Stanford Arm'ı tasarlandı. Bu robot kolun kinematik tasarımı günümüzde standart kol olarak adlandırılır.

1970 Stanford Üniversitesi Stanford Cart'ı üretti. Çizgi izleyen bir robot olarak tasarlanan bu robot aynı zamanda bir bilgisayar tarafından radyo link aracılığıyla kontrol edilebilmekteydi.

1973 Cincinnati Milacron ilk mikro bilgisayar kontrollü endüstriyel robot olan T3'ü piyasaya sürdü.

1974 Stanford Arm'ın tasarımcısı Professor Victor Scheinman tarafından kurulan Vicarm Inc firması, dokunma ve basınç algılayıcılarından gelen verileri kullanarak küçük parçaların montajını yapan Stanford Arm'ın endüstriyel versiyonu Silver Arm adındaki robot kolu tasarlandı ve piyasaya sürdü.

1976 Viking 1 ve 2 uzay sondalarına robot kollar kullanıldı.

1977 Avrupalı robot firması ASEA, iki boyda, programlama ve çalışma için mikro bilgisayarlar kullanan endüstriyel robotları piyasaya sürdü.

1977 George Lucas' Star Wars adlı filmi vizyona girdi. Bu filmde yer alan R2-D2 ve C-3PO robotları en ünlü robotlar olarak kabul edilmekte ve yeni araştırmacı nesillerinin oluşması için ilham kaynağı olmaktadır.

1977 Unimation, Vicarm Inc. firmasını satın aldı

1978 Brooks Automation firması kuruldu

1978 Vicarm'ın teknolojisini kullanarak, Unimation PUMA (Programmable Universal Machine for Assembly) robotunu geliştirdi. Puma robotu günümüzde dahi birçok araştırma laboratuvarında yer almaktadır.

1979 Stanford Cart adlı robot sandalyelerle dolu bir odayı insan yardımı olmadan geçti. Üzerinde bulunan televizyon kamerası ile farklı açılarda resimler çekerek robot ve engel arasındaki mesafeyi hesaplayan bir bilgisayara göndermekteydi. Bu çalışma engelden sakınma ve yol planlama konusundaki en önemli çalışmalardan biridir

1979 Japonya'daki Yamanashi üniversitesi tarafından tasarlanan SCARA (selective compliant articulated robot arm) robot kol Sankyo ve IBM tarafından piyasaya sürüldü.

1979 Carnegie Mellon Üniversitesi robot enstitüsü kuruldu.

1979 Stanford Hans Moravec tarafından yeniden tasarlandı. Daha otonom bir hareket sağlayan bir görüntüleme sistemi geliştirildi. Bu 3 boyutlu çevre haritalaması üzerine yapılan ilk çalışmalardan biridir.

1981 CRS Robotics Corp. kuruldu.

1981 Cognex kuruldu.

1981 Takeo Kanade direk sürürlü ilk robot kolu üretti. Bu robot kol eklemleri üzerinde motorlar barındıran ilk robot koldur ve bu değişiklik önceki robot kollardan daha hızlı ve daha hassas konumlandırma yapmaya olanak tanımıştır.

1982 Japonya'dan Fanuc ve General Motors Kuzey Amerika'da robot satışı için GM-Fanuc adında ortak bir şirket kurdular. .

1983 Adept Technology firması kuruldu

1984 Joseph Engelberger, daha sonra ismi Helpmates olarak değişen Transition Robotics firmasını kurdu. Firma servis robotları üzerine çalışmalar yaptı.

1986 Honda robotların insanların yapamayacağı görevleri yerine getirerek insanlarla birlikte çalışması prensibini benimseyen bir robot araştırma programı başlattı.

1986 LEGO ve MIT Media Lab, lego tabanlı eğitim araçlarının tasarımı için güçlerini birleştirdiler.

1986 Unimation lisansının sona ermesinin ardından, Kawasaki kendi elektronik robot ürün ailesini üretmeye başladı.

1988 Stäubli Grubu, Westinghouse'dan Unimation firmasını satın aldı.

1989 MIT'deki mobil robot araştırma grubu Genghis adındaki yürüyen robotu tanıttı.

1989 Barrett Technology firması kuruldu

1989 Computer Motion firması kuruldu

1992 Dr. John Adler, cyberknife adında hastayı x ışınları ile tarayan ve bir tümör bulduğunda o noktaya radyasyon gönderen bir robot fikrini ortaya attı.

1992 Marc Thorpe uzaktan kumandalı bir elektrikli süpürge üretmek üzerine çalışırken robot savaşı¹⁰ organizasyonu fikrini ortaya attı.

1993 Carnegie Mellon Üniversitesinde geliştirilen Dante adlı 8 bacaklı yürüyen robot Antarktika'daki Erebrus yanardağının kraterinde araştırma yaptı. Ancak robotun zincirlerinden biri kopup kratere yuvarlanması sonucu araştırma başarısızlıkla sonuçlandı.

1993 Sensable Technologies kuruldu.

1994 CMU¹¹ Robot enstitüsünde bir önceki versiyonundan daha sağlam olarak tasarlanan Dante II Alaska'daki Spurr yanardağında başarılı bir araştırma gerçekleştirdi.

1994 Marc Thorpe San Francisco'da ilk robot savaşı etkinliğini gerçekleştirdi.

1996 David Barrett tarafından MIT'de doktora tezi olarak geliştirilen RoboTuna balıkların yüzme şeklini taklit eden ilk robottu.

1996 Chris Campbell ve Stuart Wilkinson, Güney Florida Üniversitesinde organik besinleri sindiren ve çıkan karbondioksiti enerji amaçlı kullanan Gastrobot'u geliştirdiler.

1996 Honda, 1986'da başlayan 10 yıllık bir çalışmanın ürünü olan P3'ü duyurdu. ,

1997 NASA'nın pathfinder adındaki mars uçuşu başladı. Bu uçuşta görev alan Sojourner adındaki mobil robot mars yüzeyinin ilk fotoğraflarını dünyaya gönderdi.

¹⁰ Robot Combat

¹¹ Carnegie Mellon Üniversitesi

1997 Uluslararası Uzay İstasyonunun ilk bölümü yörüngeye yerleşti. 2001’de Kanadalı MD Robotics geliştirdiği robot kol istasyona eklendi.

1998 LEGO ilk robot geliştirme sistemi Robotics Invention System 1.0’ı piyasaya sürdü. Bu ürün günümüzde çoğu üniversite tarafından ve amatör robot meraklıları tarafından kullanılmaktadır.

1998 Tiger Electronics firması Furby adındaki oyuncak robotu piyasaya sürdü. Bu çok çeşitli algılayıcılarla donanmış robot oyuncak çevresine tepki verebiliyor; 800 kelimeden oluşan dağarcığını kullanarak İngilizce ve kendi dili olan furbish ile çevresi ile iletişim kurabiliyordu.

1999 LEGO firması robot geliştirme sisteminin 1. 5 versiyonunu piyasaya sürdü.

1999 SONY AIBO adındaki robot köpeği piyasaya sürdü.

2000 Honda humanoid¹² projesinin son ürünü olan ASIMO’ yu tanıttı.

2000 LEGO, MINDSTORMS Robotics Invention System TM 2.0 versiyonunu piyasaya sürdü.

2000 Sony, Sony Dream Robots (SDR) tarafından geliştirilen humanoid robotu Robodex fuarında tanıttı. .

2001 Kanadalı MD Robotics tarafından geliştirile “Space Station Remote Manipulator System” (SSRMS) yörüngeye yerleşti ve uluslararası uzay istasyonunun diğer bölümlerini inşa etmeye başladı.

2001 Amerikan sağlık bakanlığı FDA CyberKnife robotunun tümörlerin tedavisinde kullanılmasına izin verdi.

2001 Sony AIBO robot köpeğin daha gelişmiş versiyonunu piyasaya sürdü.

2002 Honda' nın ASIMO robotu New York borsasının açılış zilini çaldı.

2003 NASA, 10 Haziranda MER-A "Spirit" ve 7 Temmuzda MER-B "Opportunity". uzay araçlarını marsa gönderdi

2004 4 ocakta kapsülün marsa inişi sırasında yanması ve mars yüzeyinde sekmesinin ardından keşif robotu Spirit marsa indi.

2005 Woowee firması oyuncak insansı robotu Robosapien’i piyasaya sürdü.

¹² İnsansı robot

2005 Irobot firması robot elektrik süpürgesi roomba ve yer silme robotu scooba'yı piyasaya sürdü. Bu robotlar ticari olarak satışa sürülen ilk ev içi hizmet robotları olarak kabul edilmektedir.

2006 Honda, ASIMO robotunun özelliklerine koşma yeteneğini ekledi.

2006 Wowwee firması robosapien'in 2. versiyonunu piyasaya sürdü. Planlanan iki yeni versiyonun tanıtımını yaptı.

Bu uzun gelişim sürecinin ardından günümüzde robotik bilimi bilgisayar ve yapay zeka gibi diğer disiplinler ile ortaklaşa bir çalışma yürütmektedir. Günümüzde sadece mekanik olarak tasarlanmış bir sistem robot oluşturmaya yetmemekte donanım ve yazılım unsurlarıyla desteklenerek daha üst bir işlevsellik ve yetenek seviyesine ulaşmaya çalışılmaktadır.

Robotik biliminin başlıca araştırma konuları, insan robot etkileşim metodları, otonom robotlar, işbirliği yapan robotlar, robot öğrenmesi, robotlar için karar verme metodları, çevre, yüz, ses, sembol tanıma ve işleme, robot algılayıcıları ve algılayıcı verilerinin işlenmesi, insansı robotlar, alternatif hareket metodları, navigasyon metodları, animatronikler¹³ olarak sıralanabilir. Görüldüğü gibi günümüz robotik bilimi robotların daha zeki ve daha işlevsel olmasını amaçlamaktadır. Bu nedenle yapay zeka konularının da robotik biliminin gelişiminde büyük bir önemi vardır. Bu sebeple bir sonraki bölümde yapay zeka konuları irdelenecektir.

¹³ Canlıların jest ve mimiklerini taklit eden robotlar

3 YAPAY ZEKA

3.1 Yapay Zekanın Tanımı

Yapay zeka kavramı çok geniş bir bilim dalını temsil eder ve bilgisayarların insan zekasını gerektiren işlevleri yerine getirmesi ile ilgili araştırmalar bütünüdür.

Geçmişten günümüze gelen birikimle YZ hakkında birçok tanımlamalar yapılmıştır [30, 31, 37].

“Yapay zeka insan tarafından yapıldığında zeki olarak adlandırılan davranışların makine tarafından yapılmasıdır.” tanımı en yaygın kullanılan tanımlardan biridir.

Bu tanımda belirtilen zeki davranışların arasında düşünebileceğimiz bir çok davranış vardır. Karmaşık matematiksel problemlerin çözümü gibi davranışlar günümüzde bilgisayarlar tarafından kolaylıkla sergilenebilmektedir. Oysa insanların genellikle üzerinde düşünmeden yaptığı birçok zeki davranış (örneğin bir kişiyi gördüğünde yüzünü tanıması) bilgisayar ve makineler tarafından yapılması oldukça zor ve çok karmaşık çıkarım işlemleri ve bilgi tabanı gerektiren zeki davranışlardır.

Yapay zeka çalışmalarının yapılmasında farklı nedenler bulunmaktadır. Bunlardan birincisi insan zekasının ve davranışlarının daha iyi anlaşılabilmesidir. Diğer bir neden ise insan zekası benzetimlerini kullanarak daha kullanışlı ve insana yardımcı olan makinelerin üretilmesidir.

Yapay zeka genellikle bilgisayar biliminin bazı kolları ile örtüşen bir araştırma alanıdır. Yapay zekanın bazı alanları ise psikoloji, felsefe ve mantık ve dil bilimi gibi alanlarla ilintilidir. Yapay zekanın detaylarına inmeden önce zekanın yapay olarak oluşturulup oluşturulamayacağına değinmekte fayda vardır.

Yapay zeka araştırmaları insan zekasının karmaşık sembollerin işlenmesi şekline indirgenebileceği ve hangi işleme ortamının kullanıldığının önemi olmadığı (mutlaka biyolojik bir beyine gereksinim olmadığı) varsayımına dayanır. Bu varsayım birçok filozof tarafından eleştirilmekte ve gerçek zekanın bir bilgisayar tarafından sağlanamayacağını mutlaka insan özelliklerini barındırması gerektiği savunulmaktadır.

Yapay zekanın hedeflenen biçimi ile gerçekleşip gerçekleşmeyeceğine ilişkin görüşler karşılıklı söylenmekle birlikte, insanı salt molekül ya da davranış yığını olarak görmeyip, insanı insan yapan bir “öz” ün var olduğu gerçeğini kabul edenler için bilgisayar bir makine olmaktan öte geçemeyecektir. Bu konudaki tartışmalar günümüzde de sürmektedir.

Bu felsefi tartışmaya en büyük katkılar Turing’in “Turing testi” makalesi [32] ve Searle’in Çin odası deneyi [33, 34] tarafından yapılmıştır. Kısaca Turing öne sürdüğü testte bir makinenin gerçekten zeki olup olmadığının nasıl anlaşılacağını incelemektedir. Genel anlamda bu test bir uzmanın, makinenin performansı ile bir insaninkini ayırt edip edemeyeceğini ölçer. Eğer ayırt edemezse, makine insanlar kadar zihinsel yetiye sahip demektir. Bu testte deneyi yapan kişi diğer odalarda bulunan bir insan ve bir bilgisayar ile hangisi ile iletişim içinde olduğunu bilmeden bilgisayar kullanarak haberleşmektedir. Amaç, deneyi yapanın uygun sorgulama ile deneklerden hangisinin insan, hangisinin bilgisayar olduğunu bulmasıdır. Deneyi yapan kişi diğer odada bulunan deneye sorular yöneltir. Bu soru “bu şiir hakkında ne düşünüyorsun?” gibi çok sübjektif bir soru dahi olabilir. Eğer bilgisayar soruyu soran kişinin cevaplardan hangisinin bir insana hangisinin bir bilgisayara ait olduğunu anlayamayacağı kadar iyi bir cevap veriyorsa o zaman bilgisayar Turing testini geçer ve insanlar kadar kavrama yeteneğinin olduğu kabul edilir.

Searle ise sadece zeki gibi davranmanın zeki olmak için yeterli olmadığını öne sürer. Bu fikrini tasarladığı bir düşünce deneyi ile desteklemiştir. Çin odası adı verilen bu deneyde Çince bilmeyen bir kişi kapalı bir odada bulunmaktadır. Kendisine üzerinde Çince mesajlar bulunan kağıtlar verilmekte ve bu kişi elinde bulunan ve hangi Çince sembollere karşılık hangi sembolleri vermesi gerektiğini belirten bir liste yardımıyla odanın dışındakilerle haberleşmektedir. Bu durumda kişi Çince bilmemesine rağmen dışarıdaki kişilerin onu Çince biliyor gibi düşünmesi muhtemeldir. Searle bu durumda bilgisayarlarında zeki gibi davranmalarına rağmen zeki olamayacaklarını sadece kendilerine verilen kuralları (örneğin Çince sembolleri içeren liste) kullanabileceklerini söylemiştir.

Bazı araştırmacılar ise bilgisayarların zeki gibi davranmalarının (Turing testini geçmelerinin) bile mümkün olmadığını iddia etmişlerdir

Bu tartışmaların ardından ortaya çıkan dört farklı görüş bulunmaktadır.

- Bilgisayarlar zeka gerektiren bazı davranışlar sergilese bile hiçbir zaman zeki gibi görünemezler
- Bilgisayar zeki gibi görünebilirler ancak sadece zeki davranışların benzeşimini gerçekleştirebilirler. Hiçbir zaman gerçekten zeki olamazlar
- Bilgisayarlar sonuçta gerçek zekaya sahip olabilirler.
- Bilgisayarlar gerçek zekaya sahip olmanın yanı sıra bilinç ve duygulara da sahip olabilirler.

Bu felsefi tartışmaların yapay zeka araştırmaları ve uygulamalarına çok nadir etkileri olmaktadır. Açıktır ki yapay zeka metodları kullanılarak normalde insan zekası gerektiren işleri yapabilen programlar tasarlanabilmekte ve insan zekasının doğası daha detaylı bir şekilde incelenabilmektedir.

Günümüzde belirli sınırlar içinde de olsa bağımsız ve özgün hareketler sergileyebilen yapay zeka örnekleri üretilmektedir ancak bugünkü teknolojilerin uzantısı olarak bilgisayarların yapabileceği tek şey kendilerine verilen algoritmaları yerine getirmektir [35]. Algoritma her biri açık bir şekilde tanımlanmış işlemler dizisini ifade eder. Öyleyse bir bilgisayarın bir şeyi yapıp yapamayacağı, o şeyin algoritmaya indirgenip indirgenemeyeceğine bağlıdır denebilir. Dolayısıyla bilgisayar sadece belirlenmiş davranışları sergileyebilir ve sadece davranışı sergilemek gerçekten öğrenme olduğu anlamına gelmeyebilir.

Bilgisayarların formel (kural tabanlı) ve deterministik sistemler olduğu göz önüne alındığında yapay zekanın elde edilebilmesi için farklı yöntemlerin geliştirilmesi gereksinimi doğmuştur. Bu gereksinim sonucunda genler ve mutasyonu örnek alan genetik algoritmalar ve insan beynindeki nöronları örnek alan yapay sinir ağları gibi canlıların yapısını taklit eden yöntemler geliştirilmiştir. Bu yöntemler ve yapay zekanın diğer çalışma alanlarına bir sonraki bölümde değinilecektir.

3.2 Yapay Zeka Tarihçesi

Yapay Zeka kavramının geçmişi bilgi işlem biliminin ilk günlerine kadar uzanır. Hatta konuya felsefi açıdan yaklaşıldığında, Yapay Zeka'nın temellerini Aristo'nun mantıksal çıkarımlarına dayandığı söylenebilmektedir. Tarihte gollem olarak bilinen robotların yapımı ve bunların insan gibi davranmasına yönelik çalışmalar da yapay zekanın doğuşunda büyük bir önem taşımaktadır.

Yapay zeka isminin oluşması ise 1950'lere dayanır. 1956 yılında çalışmalarını Princeton Üniversitesi'nde sürdüren John McCarthy, Marvin Minsky, Shannon ve Rochester'le birlikte Dartmouth'da bir konferans düzenlemiş ve McCarthy, yeni türettiği Yapay Zeka (Artificial Intelligence) ismini, konferansa ad olarak önermiştir. Bu toplantıda bir çok çalışmanın temelleri atılmakla birlikte, toplantının en önemli özelliği Mc Carthy tarafından önerilen [36] Yapay Zeka adının kabul edilmesidir.

Yapay zekanın gelişimini dönemler halinde incelemekte fayda vardır. Böylece günümüzde yapay zekanın geldiği nokta daha kolay anlaşılabilir. MÖ 5. yüzyıla Aristo sembolik mantığı geliştirerek ilk tümdengelimli çıkarım sistemini oluşturmuştur. 17. yüzyılda ise Dekart hayvanların vücutlarının karmaşık makineler olduğunu öne sürmüştür. Bu çıkarım hayvanlara benzeyen makinelerin yapılabileceğini ve bu makinelerin hayvanların davranışlarını ve zekasını sergileyebileceği sonucunu doğurur. Yine 17. yüzyılda Hobbes madde ve katışımsal düşünce teorisini içeren “The Leviathan” eserini yayınlamıştır. 1642 yılında ise Pascal o ana kadar sadece insan zekasının yapabildiği işlemlerden biri olan hesap yapmayı geliştirdiği ilk rakamsal hesap makinesi ile makinelerinde yapabileceğini göstermiştir. 1673 yılında ise ünlü bilim adamı Leibniz pascal'ın bu tasarımını daha da ilerleterek çarpma ve bölme yapabilen bir hesap makinesi geliştirmiştir. 18. yüzyılda ise genellikle insan ve hayvanları taklit eden otomatlar üzerine çalışılmıştır. Bu çalışmalar günümüzde yapay zekanın alt dallarından biri olan yapay yaşamın¹⁴ temellerini oluşturmuştur. Bir robotik çalışması olan ve Osmanlı sarayı için üretilen satranç oynayan adam adındaki robot aynı zamanda karmaşık hamlelerin hesaplamasını yaparak yapay zekanın önemli adımlarından biri olmuştur.

1800lerde ise George Boole önerdiği mantıksal cebir ile mantığın matematiksel bir formda gösterimini yaparak yapay zeka çalışmalarında kullanılan yapılardan birini oluşturmuştur. Yine bu yüzyılda Charles Babbage ve Ada Byron (Lady Lovelace) programlanabilir mekanik hesap makineleri üzerinde çalışmışlardır.

20 yüzyılda ise yapay zeka çalışmaları büyük bir ivme kazanmıştır. ilk olarak Bertrand Russell ve Alfred North Whitehead “*Principia Mathematica*” [38] adlı eseri yayınlarak formel mantığın oluşmasını sağlamışlardır. Russell, Ludwig Wittgenstein, ve Rudolf Carnap felsefeyi bilginin mantıksal analizi konusuna yönlendirerek yapay zekanın gelişimi için gerekli olan düşünsel temeli

¹⁴İng: Artificial life

oluşturmuşlardır. Warren McCulloch ve Walter Pitts'in yayınladığı "A Logical Calculus of the Ideas Immanent in Nervous Activity" adlı kitap ise yapay sinir ağlarının ilk incelenmesi olarak bilinmektedir. Arturo Rosenblueth, Norbert Wiener ve Julian Bigelow 1943 yılında yayınladıkları makalelerinde sibernetik¹⁵ terimini ilk kullananlar olmuşlardır.

1950 yılında ise Yapay zeka felsefesini ilk ortaya çıkaran kişi olarak bilinen ünlü İngiliz mantıkçı ve matematikçisi Alan Turing yapay zeka isminin Dartmouth konferansında doğmasından 6 yıl önce Mind adlı felsefe dergisinde *Computing Machinery and Intelligence* adlı bir makale [32] yayınlamıştır. Bu makalede Turing "Makineler düşünebilir mi?" sorusunu ortaya atarak bir felsefi tartışmayı açmıştır. Aynı zamanda bu çalışmasında makinelerin zeki olup olmadığının anlaşılması için sonradan turing testi olarak adlandırılacak olan testi öne sürmüştür. Yine aynı yılda Claude Shannon satranç oyunun detaylı analizini içeren eserini yayınlamıştır. Bu çalışmalar Isaac Asimov 'un 3 robot kuralını yayınladığı zamanlara denk gelmektedir.

1956'da McCarthy'nin yapay zeka terimini kullanmasının ardından yapay zekanın modern çağı olarak adlandırılan dönem başlar. Modern yapay zeka çalışmalarının tarihi bu bölümde daha detaylı olarak incelenecektir. Carnegie teknoloji enstitüsü bugünkü adıyla Carnegie Melon Üniversitesi'nden Allen Newell, J. C. Shaw ve Herbert Simon'ın yazdığı bir program olan Logic Theorist bilinen ilk yapay zeka programı olarak adlandırılmaktadır. 1957'de ise Newell, Shaw ve Simon tarafından General Problem Solver (genel problem çözücü) adını taşıyan program geliştirilmiştir. 1952 1960 yılları arasında ise IBM'de çalışan Arthur Samuel ilk oyun oynayan program olan dama programını geliştirmiş ve bu program dünya şampiyonlarına kafa tutmuştur. 1958 yılında yapay zeka isminin babası olan John McCarthy lisp programlama dilini geliştirmiştir. Herb Gelernter ve Nathan Rochester teorem ispatlayan bir yöntem geliştirmişlerdir. İngiltere'de düzenlenen düşünme sürecinin mekanizasyonu konulu Teddington konferansında John McCarthy'nin "Programs with Common Sense, " Oliver Selfridge'in "Pandemonium, " ve Marvin Minsky'nin "Some Methods of Heuristic Programming and Artificial Intelligence. " makaleleri yapay zekanın gelişiminde önemli katkılarda bulunmuştur. 1960'ların

¹⁵ İng:cybernetics

başında ise Margaret Masterman ve ekibi Cambridge üniversitesinde makine tercümesinde kullanılmak üzere semantik ağlar geliştirmişlerdir.

1963'de Thomas Evans'ın MIT'de hazırladığı doktora tezinin bir parçası olarak geliştirdiği ANALOGY adlı program insanların IQ testlerinde kullanılan analoji problemlerinin bilgisayarlar tarafından da çözülebileceğini göstermiştir. 1964 de Danny Bobrow'un MIT'de yaptığı çalışma bilgisayarların sözlü cebir problemleri çözebilecek düzeyde doğal dili anlayabileceğini göstermiştir. Bert Raphael' in çalışması ise bilginin mantıksal gösteriminin soru cevaplayan sistemler için kullanımını sağlamıştır. Joseph Weizenbaum'un geliştirdiği ELIZA adlı program ise İngilizce olarak herhangi bir konu üzerinde konuşabilen ilk program olarak bilinmektedir. 1966 yılında ise Ross Quillian'ın çalışması semantik ağların bir gösterimi niteliğini taşımaktadır. 1967 yılında Stanford üniversitesinden Edward Feigenbaum, Joshua Lederberg, Bruce Buchanan ve Georgia Sutherland tarafından geliştirilen Dendral adlı program ise organik kimyasal bileşimlerin analizini yaparak bilimsel çıkarımlar yapan ilk bilgi tabanlı program olmuştur. Aynı yılda Joel Moses'in sembolik çıkarımların integrasyon problemlerine uygulanmasını sağlayan Macsyma programı, matematik alanındaki ilk başarılı bilgi tabanlı programdır. MIT'de Richard Greenblatt'ın geliştirdiği Machack adındaki program ise bilgi tabanlı bir satranç programı olup satranç turnuvalarında c-sınıfı derece alabilecek yeteneklere sahipti.

Yapan zekanın önemli adımlarından biri olarak kabul edilen ve basit sinir ağlarının bir gösterimi olan perceptrons adlı eser Marvin Minsky ve Seymour Papert tarafından 1968 de yayınlandı. 1969'da Stanford araştırma enstitüsünde geliştirilen Shakey adlı robot hareket, algılama ve problem çözme yetilerinin bir birleşimi olarak yapay zeka ve robotiğin bağlarını güçlendirdi

1970 yılında ise Jaime Carbonell tarafından geliştirilen SCHOLAR adlı interaktif program semantik ağlar ve bilgi gösterimi üzerine kurulmuştu ve bilgisayar destek ders verme amacını taşımaktaydı.

1971 yılında Terry Winograd doktora tezinde bilgisayarların çocuk oyun blokları üzerine yazılmış ingilizce kelimelerle oluşturulan cümleleri anlayabileceğini gösterdi. Bu çalışmada kendi yazdığı shrudlu programı ve İngilizce komutları yerine getiren bir robot kol kullanılmıştı.

1972 yılında ise yapay zeka programlama dili olan prolog geliştirildi. Prolog programlama dili birçok yapay zeka araştırmasında kullanılan önemli bir araç haline geldi. 1974'te Ted Shortliffe ise ilk uzman sistem olarak kabul edilen ve tıbbi teşhis alanında çalışan kural tabanlı bilgi gösterimine dayalı bir sistem olan MYCIN'i geliştirdi. 1975'te Marvin Minsky bilgi gösteriminde çerçeve kullanımını öne süren ve şemalar ile semantik linkler hakkında birçok fikri bir araya getiren makalesini yayınladı.

1980 yıllarda ise lisp tabanlı bilgisayarlar üretilerek piyasaya sürüldü ve ilk uzman sistem geliştirme sistemleri ve ticari yapay zeka uygulamaları ortaya çıktı. Kazanılan ticari başarı yapay zeka üzerine olan ilginin ve araştırmaların artmasını sağladı. Bu yıllarda sinir ağlarının backpropagation algoritmalarında kullanımı yaygınlaştı.

1989 yılında CMU'da geliştirilen ALVINN (An Autonomous Land Vehicle in a Neural Network) adlı yapay zeka kontrollü araç Amerika kıtasını bir uçtan bir uca geçerek bir ilke imza attı. 1980lerin sonunda MIT'de Rod Brooks tarafından yönetilen ve birçok bilim adamının katkıda bulunduğu COG projesi hümanoid robot geliştirme alanında önemli adımlar kat etti. 90'ların başında ise Gery Tesauro'nun geliştirdiği td-gammon adlı tavla programı destekli öğrenme metodunu kullanarak dünya çâğındaki oyuncularla yarıştı. 1997 yılında ise IBM tarafından geliştirilen deep blue adlı satranç programı zamanın dünya şampiyonu Garry Kasparov yenerek büyük bir sansasyon yarattı. 1990'ların sonlarına doğru internetin kullanımının yaygınlaşması ile arama motorları ve diğer yapay zeka tabanlı bilgi çıkarım programları geliştirildi. 2000lerin başında ise yapay zeka kullanan oyuncaklar diğer bir adla akıllı oyuncaklar piyasaya sürüldü. Yine aynı yıllarda Cynthia Breazeal toplumcul makineler üzerine hazırladığı ve yüzü olan ve duygularını aktarabilen KISMET adlı robotu tanımladığı çalışmasını yayınladı. 2005 yılında ise ünlü internet arama motoru google yapay zeka metodlarını kullanarak verilen kelimelerle ilintili setler oluşturabilen google sets programının deneme sürümünü yayınladı.

Yapay zekanın yüz yıla yakın gelişme süreci sonrasında günümüzde yapay zeka bilimi bir çok alt bilim dalını ve disiplinler arası konuyu barındıran çok geniş bir araştırma alanı halini almıştır. Günümüzde zeki etmenler, bilgi temsili, bulanık mantık, çıkarsama, dağıtık zeka, doğal dil işleme, doğal dil anlama, konuşma sentezleme, genetik algoritmalar, görüntü işleme, örüntü tanıma, yapay görme, makine öğrenmesi, robotik, uzman sistemler, veri madenciliği, yapay sinir ağları,

yapay yaşam, yapay zeka felsefesi ve zeki oyunlar gibi birçok alanda yapay zeka çalışmaları devam etmektedir.

Yapay zeka ile mobil robot kontrolü projesinde ise yapay zekanın robotikle olan ilişkisi ve çıkarım yapma metodları büyük önem taşımaktadır. Takip eden bölümde yapay zeka ve mobil robot ilişkisi detaylı olarak incelenecek ve mobil robotların yapay zeka ile yönetim metodları detaylı olarak anlatılacaktır.

4 ROBOTLAR VE YAPAY ZEKA

Robotiğin ilk ürünleri olan endüstriyel robotlar genellikle ön tanımlı ve yinelenen işleri yapmak için geliştirildiklerinden, karmaşık kontrol algoritmaları ve programlama metodları kullanmaktaydılar. İnsanlar tarafından programlandıktan sonra programlarında bir değişiklik yapılmıyca kadar aynı görevi tekrar tekrar yerine getiren bu robotların en gelişmiş örnekleri yapay zekanın genellikle bulanık mantık ve örüntü tanıma yöntemlerinden faydalanıyordular.

Robotik biliminin hümanoidlere ve otonom robotlara yönelmesinin ardından robotların görme, işitme, tanıma ve hatta konuşma gibi özellikleri barındırması ve bu yetenekleri kullanarak kararlar vermesi, çevreleri ile etkileşmesi üzerine çalışmalar başlamıştır. Bu aşamadan sonra yapay zekanın doğal dil anlama, doğal dil işleme, yapay görme, örüntü tanıma, zeki etmenler, makine öğrenmesi, genetik algoritmalar ve yapay sinir ağları gibi konuları robotik çalışmalar için büyük bir araştırma alanı haline gelmiştir.

4.1 Mobil Robotların Gereksinimleri

Günümüzde tasarlanan mobil robotların büyük bir çoğunluğu insan müdahalesini en aza indirebilmek için otonom veya yarı otonom özellikleri barındıracak şekilde tasarlanmaktadır. Bir robotun otonom davranışlar sergileyebilmesi için kendi kararlarını alabilmesi gerekmektedir. Robotun kendi kararlarını alması zeki bir davranıştır ve işte tam bu noktada yapay zeka devreye girer.

Zeki bir robotun belirsizliklerin bulunduğu bir ortamda hareket edebilmesi, algılayıcılarından gelen verilere göre hareketlerini planlayabilmesi gibi davranışlar, zeki davranış olarak nitelendirilirler [30-32, 37]. Benzer bir şekilde bir robotun yaptığı davranışlar aynı durumda bulunan bir insanın yapabileceği davranışlara uyuyorsa; robot zeki davranışlar sergiliyor demektir.

Kısaca otonom olarak hareket edebilen bir robot barındırdığı yapay zeka ile bir sonraki hamlesini planlayabilmelidir. Bunu sağlamak için günümüzde farklı yöntemler uygulanmaktadır. Yapay sinir ağlarının robot beyni olarak kullanımı ve

kural tabanlı sistemler bunların en önemlileridir. Yapay sinir ağlarının kullanımı uzun bir eğitim süreci gerektirirken kural tabanlı kontrol sisteminin uygulanması görece daha kolaydır. Ancak kural tabanlı kontrol sistemlerinde kuralların her türlü olasılığı kapsayacak şekilde tanımlanması oldukça zordur. Bu nedenle sistem tasarımcısı tarafından önceden kestirilememiş bir durum oluştuğunda robotun beklenmeyen hareketler sergilemesi muhtemeldir.

Kontrol tekniğinin yanı sıra robotun çevresi ile etkileşirken kullandığı mekanizmalarında zeki bir şekilde kontrol edilebilmesi gerekmektedir. Örneğin ön tanımlı fonksiyonları gerçekleştirirken robot, değişen çevre koşullarını algılayabilmeli veya doğabilecek hataları düzeltebilmelidir.

Tüm bu işlemler günümüzde yapay zeka teknikleri kullanılarak gerçekleştirilmektedir. Hangi tekniklerin kullanılacağına belirlenmesi için bir mobil robotun gereksinimlerinin belirlenmesi büyük önem taşır. Bu nedenle mobil bir robotun gereksinimleri ve bu gereksinimlerin yapay zeka ile nasıl sağlanabileceği incelenmelidir.

Otonom mobil bir robot hareket ederken 2 önemli unsura ihtiyaç duyar. Bunlar navigasyon sistemi ve kontrol sistemidir. Navigasyon sistemi robotun hangi yöne hangi şekilde ilerleyeceğini belirlerken kontrol sistemi donanım ve yazılım faaliyetlerinin kontrolünü sağlar.

4.2 Navigasyon Sistemi

Navigasyon sisteminin ana görevleri yol bulma¹⁶, rota planlama¹⁷, engel algılama¹⁸, engelden sakınma¹⁹ ve çarpışma algılama²⁰ şeklinde sıralanabilir. Burada belirtilen görevlerin bir bölümü kontrol sisteminin yardımıyla gerçekleştirilmektedir. Bu sistemlerin detaylı olarak bilinmesi ilerideki kontrol yaklaşımlarının anlaşılmasında büyük önem taşımaktadır.

4.2.1 Yol Bulma

İlk olarak neden yol bulmaya ihtiyaç duyulduğunun anlaşılması gerekmektedir. Bunu gözleri görmeyen bir kişi ile yol bulma metodu kullanmayan bir robotun analogisi

¹⁶ Path finding

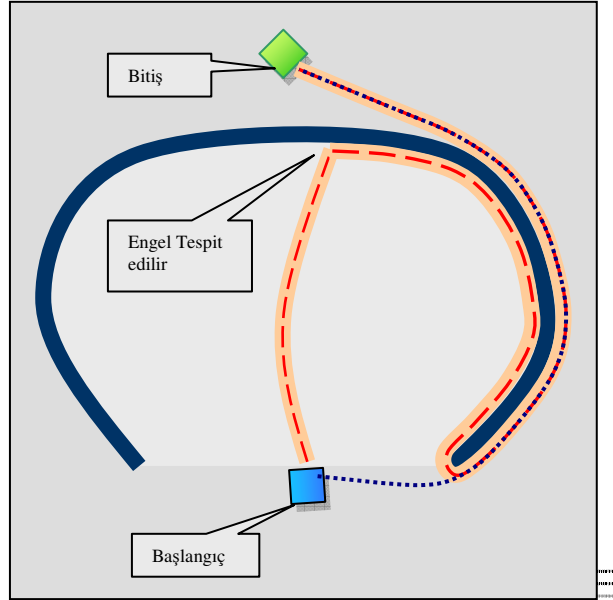
¹⁷ Route planning

¹⁸ Obstacle detection

¹⁹ Obstacle avoidance

²⁰ Collision detection

üzerinden açıklamak mümkündür. Görmeyen bir kişi veya yol planlaması yapmayan bir robot bir ortamda gitmek istediği yönde ilerleyecek ve eğer bir engelle karşılaşrsa bu engelin çevresini dolaşmaya çalışacaktır. (Şekil 4.1 kırmızı yol) Eğer önceden bir yol planlaması yapılacak olursa (Şekil 4.1’de mavi yol) bu gibi durumlardan sakınılabilmektedir.



Şekil 4.1:Hareket Metodları

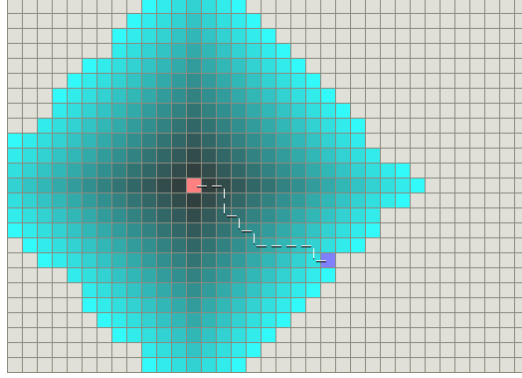
Bu durumda bir robotun gideceği yolu önceden planlaması hareketlerinin daha efektif olması açısından tercih edilmesi gereken bir yöntemdir.

Mobil robotun belirtilen bir noktadan diğer noktaya ulaşabilmesi için verilen veya kendisinin oluşturduğu haritada iki nokta arasında bir yol bulabilmesi gerekmektedir. Günümüzde en yaygın kullanılan haritalama yöntemleri ızgara tabanlı ve özellik tabanlı haritalama yöntemleridir. Izgara tabanlı haritalama metodunda çalışılan ortam hücre adı verilen 2 boyutlu geometrik alanlara bölünmüştür. Bu alanlar genellikle kare şeklindedir. Her bir hücre doluluk, tehlike, erişilebilirlik, geçiş maliyeti gibi navigasyon açısından önemli olan bilgileri içermektedir. Özellik tabanlı haritalamada ise çalışılan çevrenin geometrik özellikleri baz alınmaktadır. Örneğin duvar, köşe, kapı gibi özellikler haritada belirtilmektedir. Izgara tabanlı harita kullanılan navigasyon sistemlerinde robotun harita üzerinde bulunduğu konum bilindiği takdirde çevredeki engellerin uzaklığı yaklaşık olarak hesaplanabilmektedir. Ayrıca ızgara tabanlı sistemler matrisler olarak tanımlanabildiği için bilgisayar tarafından işlenmeleri çok daha hızlı ve kolay olmaktadır. Özellik tabanlı sistemlerin raster

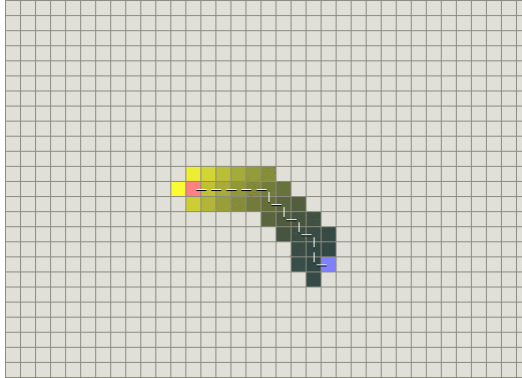
grafikler²¹ olarak gösterilmeleri durumunda bilgisayarın onları yine bir matris olarak algılaması mümkündür.

Her iki durumda da robotun bu haritayı yorumlaması ve başlangıç noktasından bitiş noktasına giden bir yol bulması gerekmektedir. Bu çalışmada ızgara tabanlı haritalama kullanıldığı için bu yöntem üzerinde kullanılan yol bulma teknikleri incelenecektir.

Yol bulma için günümüzde 3 ana yöntem kullanılmaktadır; A*[21, 22] , Dijkstra [20], BFS. Her bir yöntemin bir diğerine göre üstün ve zayıf yönleri bulunmaktadır. Bu yöntemler robotun bulunduğu hücreden başlayıp çevre hücrelere doğru yayılarak hedefe ulaşmaya çalışır. Dijkstra hedefe her yöne yayılma yöntemini kullanarak ulaşmaya çalışırken BFS heuristic adı verilen fonksiyonlar kullanarak hedefin olduğu yöne doğru yayılmayı tercih eder. Bu davranışlar Şekil 4.2e Şekil 4.3'te detaylı olarak görülebilmektedir.



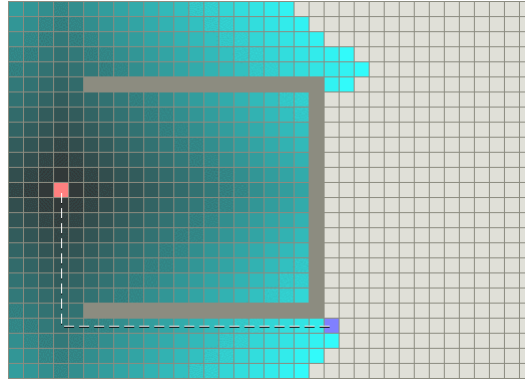
Şekil 4.2 : Dijkstra Algoritması



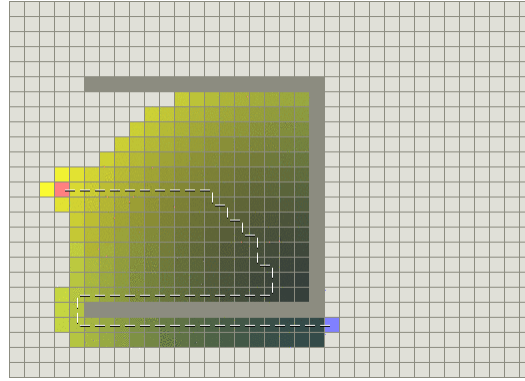
Şekil 4.3 : BFS Algoritması

²¹ Piksel tabanlı grafik

Dijkstra algoritması mümkün olan en kısa yolun bulunmasını garanti ederken; BFS algoritması en kısa zamanda yol bulmanın gerçekleşmesini sağlar. Bu algoritmaların zayıf tarafları ise ; Dijkstra'nın daha fazla hücreyi taramasından dolayı daha fazla zaman gerektirmesi (bkz. Şekil 4.2), BFS'nin her zaman en kısa yolu bulamaması şeklinde ifade edilebilir. Şekil 4.2 ve Şekil 4.3 en basit durum gösterilmiştir. Durum karmaşıktıkça iki algoritma arasındaki farklar daha belirgin hale gelir. Şekil 4.4'te görüldüğü gibi Dijkstra algoritması daha fazla hücreyi taramasına rağmen en kısa yolu hesaplayabilmektedir. BFS algoritması ise Dijkstra ya yakın sayıda hücreyi taramasına ve yaklaşık olarak eşit zaman harcamasına rağmen optimum yoldan çok daha uzun bir yol belirlemiştir (Şekil 4.5).



Şekil 4.4 : Karmaşık Ortamda Dijkstra Algoritmasının Çalışması



Şekil 4.5: Karmaşık Ortamda BFS Algoritmasının Çalışması

Günümüzde en yaygın olarak kullanılan A* algoritması ise kullandığı heuristic fonksiyonu ile bu iki yönteme nazaran büyük bir avantaj sağlamaktadır. En kısa yolu en kısa zamanda bulmayı garantilemektedir. A* algoritmasında heuristic fonksiyonu değiştirilerek Dijkstra veya BFS gibi davranması sağlanabilmektedir. Bu nedenle heuristic fonksiyonun seçimi büyük önem taşımaktadır. Bu fonksiyon seçilirken zaman ve en optimum yol arasında bir ters orantı olduğu göz önüne alınmalıdır.

A* algoritması karar verirken her bir hücre için 3 adet ağırlık fonksiyonundan faydalanır: $f(n)$, $g(n)$, $h(n)$. $g(n)$ fonksiyonu başlangıç noktasından n . Hücreye olan yolun maliyetini barındırır, $h(n)$ ise n . Hücresinin bitiş noktasına olan tahmini maliyetidir. Bu iki fonksiyonun toplamı olan $f(n)$ ise toplam maliyeti temsil eder.

A* algoritmasının çalışmasında iki hücre kümesi bulunur bunlar açık ve kapalı hücre kümeleridir. Başlangıçta robotun bulunduğu hücre açık kümesinde yer alır kapalı küme ise boşdur. İncelenen her bir hücre kapalı kümeye aktarılır. Ana döngü içerisinde ilk olarak açık kümeden maliyeti en düşük olan hücre seçilir eğer bu hücre bitiş hücresi ise döngü sonlandırılır. Eğer bitişe ulaşamamış ise n . hücre açık kümeden çıkarılıp kapalı kümeye aktarılır. Ve bu hücrenin komşuları olan hücreler taranır. Eğer n^1 hücresi kapalı kümede ise önceden işleme alındığı için bir işlem yapılmaz. Diğer durumda hücre açık kümesine eklenerek n . hücrenin alt hücresi olarak işaretlenir. Maliyeti ise $g(n)$ maliyeti ile n^1 hücresine geçmenin maliyetinin toplamı şeklinde kaydedilir.

Bu çalışmada ise üç yöntemin bir bileşimi olan bir yol bulma yöntemi izlenmiştir. Bu yöntemin teorik detayları bölüm 5.4'te; yazılım uygulaması ise bölüm 7 de detaylı olarak incelenmektedir.

4.2.2 Rota Planlama

Rota planlama işlemi yol bulma işlemi tarafından hesaplanan yolun nasıl izleneceğinin belirlenmesinde kullanılır. Robotun başlangıç anındaki yönü, hızı gibi parametreler göz önüne alınarak planlanan yoldaki bir sonraki adıma ulaşılabilmesi için gerekli olan hareketler belirlenir. Rota planlama işlemi sırasında hız, ivmelenme, dönüşler gibi gerekli olan hesaplamalar yapılarak bulunan yolun en az hata ile izlenebilmesi amaçlanmaktadır.

4.2.3 Engel Algılama

Robotun izleyeceği yol üzerinde önceden bilinen veya bilinmeyen engelleri algılayabilmesi hedefe ulaşabilmek için hayati önem taşır. Engel algılama işlemi yol planlama işlemi sırasında harita üzerinde gerçekleştirilir. Ancak haritanın tam doğru olmaması ve robotun hareketinde olabilecek konum hataları nedeniyle robotun işletimi sırasında da gerçek zamanlı olarak yürütülmesi gereken bir işlemdir.

4.2.4 Engelden Sakınma

Mobil robotun bir engeli algıladıktan sonra bu engelden sakınabilmek için bazı kararlar alması gerekmektedir. Bu karar engelin etrafını dolaşmak şeklinde olabileceği gibi yol bulma mekanizmasını tekrar çalıştırıp yeni bir yol bulmak şeklinde de olabilir.

4.2.5 Çarpışma Algılama

Alınan tüm önlemlere rağmen robot bir engeli algılayamamış ise bir çarpışmanın yaşanması kaçınılmazdır. Bu durumda robotun kendisine ve engele (engel bir insan olabilir) zarar vermemesi için gerekli işlemleri yapması gerekmektedir. Örneğin hareketini durdurup geri çekilmesi en uygun işlemlerden biridir. Bu işlem belirlenirken aynı durumda kalan bir insanın yapacakları dikkate alınarak zeki bir davranış sağlanabilir.

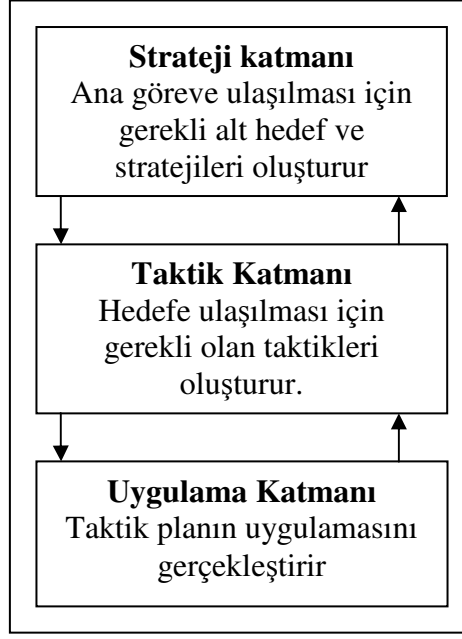
Tüm bu navigasyonel faaliyetlerin hayata geçirilmesi ve koordineli bir şekilde çalışabilmesi için robotlar bir kontrol sistemine ihtiyaç duyarlar.

4.3 Kontrol Sistemi ve Mobil Robot Kontrol Yaklaşımları

Kontrol sistemi navigasyonel faaliyetlerin düzenlenmesinin yanı sıra donanım ve yazılımın zeki bir şekilde yönetilmesinden de sorumludur. Algılayıcı verilerinin okunması ve değerlendirilmesi, hareket mekanizmalarının çalıştırılması ve navigasyon sisteminin ürettiği sonuçların işleme koyulması yine kontrol sisteminin sorumluluğundadır.

Kontrol sistemi, navigasyon sistemi ve diğer alt sistemler, robotun kontrol yapısını oluştururlar. Günümüzde kullanılan iki genel kontrol yaklaşımı bulunmaktadır. Bunlar hiyerarşik yaklaşım ve davranış temelli yaklaşımdır.

Hiyerarşik yaklaşımda tüm alt sistemler biri birine seri olarak bağlanmışlardır ve üst ilişkisini barındırırlar Genellikle 3 katman barındıran bu yaklaşım var olan bilgisayar prosedür ve ara yüzlerine kolaylıkla adaptasyonu deneniyle endüstriyel uygulamalarda en çok tercih edilen yaklaşımdır. Bu mimarinin 3 katmanlı hali Şekil 4.6'da görülmektedir.



Şekil 4.6: Hiyerarşik Kontrol Yaklaşımı

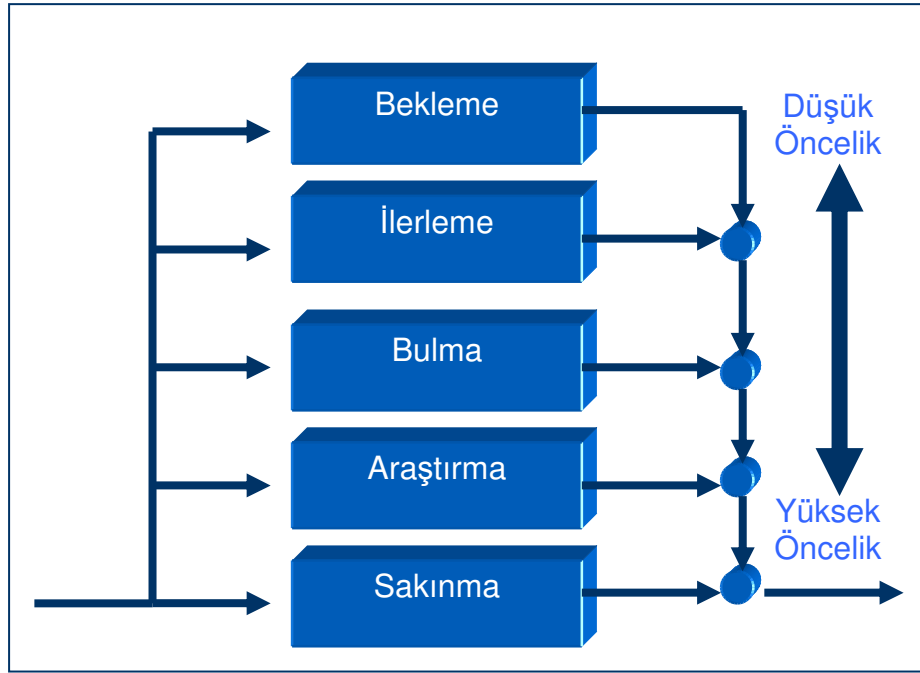
En alt katman olan uygulama katmanı gerçek zamanlı kontrol fonksiyonlarını yerine getirir ve taktik katmanından gelen makro komutların uygulanmasını sağlar. Taktik katmanı ise strateji katmanından gelen komutları (örneğin harita üzerinde bulunmuş olan yol) yorumlayarak bu komutun gerçekleştirilebilmesi için gerekli makro komutları (örneğin robotun hangi yöne ilerleyeceği, hangi algılayıcıları okuyacağı) oluşturur. En üst katman olan strateji katmanı ise durum analizi yapma, hedefe ulaşılması için gereksinimleri belirleme ve strateji hazırlama görevleri nedeniyle yapay zeka yöntemlerinin en sık kullanıldığı katmandır. Hiyerarşik kontrol metodu çok yaygın olarak kullanılmasına rağmen hesaplama yükü ve katmanlar arasındaki haberleşme gereksinimleri bazı durumlarda aşırı olabilmektedir.

Günümüzde çevre haritasının tutarlılığının ve hassasiyetinin sağlanması, yapısal olmayan ve genellikle önceden kestirilemeyen alanların genel gösterimi gibi alanlar hiyerarşik kontrol ile ilgili en çok araştırmanın yapıldığı konulardır.

Hiyerarşik metoda alternatif olarak gelişen ve günümüzde büyük bir ivme kazanmış olan metod, davranış temelli veya reaktif olarak adlandırılan kontrol metodudur. Davranış temelli yöntemde genel görev her biri çeşitli davranış fonksiyonlarını kontrol eden veya etkileyen alt davranış ve yeteneklere ayrılır. Bu davranış ve yetenekler genellikle birbirine paralel olarak çalışan katmanlı bir yapı şeklinde tasarlanmışlardır. En alt katman olarak engellerden sakınma davranışı kullanılır. Bu

davranış katmanı tasarlanıp test edildikten sonra diğer davranışlar (gezinme, harita çıkarma vb) üst katmanlara eklenerek robotun genel yeteneği geliştirilir.

Her bir davranışın önceliği tasarım aşamasında belirlenerek yüksek öncelikli davranışların daha az önceliğe sahip davranışları baskılaması sağlanır. Şekil 4.7’de robotun genel davranışının farklı çıktıların nasıl bir birleşimi olduğu görünmektedir. Tablo 4.1’de ise davranış temelli kontrol uygulanan robotlarda ve strateji türündeki bilgisayar oyunlarında sıkça kullanılan bazı temel davranışlar listelenmektedir.



Şekil 4.7: Davranış Temelli Yaklaşım

Tablo 4.1 : Kullanılan Temel Davranışlar

Stop	Bulunduğun konumda dur
Stay	Bulunduğun alanda bekle
Flee	Güvenli bir alana kaç
Retreat	Güvenli bir alana düşmanlarla savaşarak kaç
Explore	Hakkında az bilgi bulunan alanları keşfet
Wander	Amaçsızca gezin
Search	Çevrede belirli bir objeyi ara
Spy	Bilinmeyen bir objenin yanına görünmeden git ve bilgi topla
Patrol	Belirtilen bölgede devriye gez
Defend	Belirtilen bir objenin yanında durarak onu koru
Guard	Bir alanın girişinde bekleyerek düşmanın girmesini engelle
Attack	Bir düşmana esir almak veya yok etmek için saldır
Avoid	Bir objeden uzak dur
Follow	Hareketli bir objenin yakınında kalarak onu takip et
Work	Bir görevi yerine getir

Davranış temelli yaklaşımda genel bir çevre modeli kullanılmadığı gibi katmanlar arasında da az bir bant genişliği kullanan basit bir haberleşme protokolü yeterli olmaktadır. Her bir katmanın görece basit bir hesaplama sistemi bulunduğu için paralel işlem yapılması mümkün olmakta ve genellikle çok az bir işlemci gücü gerekmektedir. Sistemin modüler yapısı, genişletilebilir bir sistem yaklaşımını ve önceden kestirilemeyen ortamlarda iyi sonuçlar vermesini sağlamaktadır.

Çok karmaşık davranışların daha basit davranışların bir kombinasyonu olarak gerçekleştirilebilmesine rağmen katmanların sayısı arttıkça baskılayan etmenlerin sayısı da artmakta ve bu durum çok karmaşık bir altyapı ve çoğu zaman çözümlenemeyen karışıklıklara sebep olabilmektedir. Tüm davranış örüntüsü elle tasarlandığı için önceden tahmin edilemeyen ve analizi çok zor olan yanlış davranışların oluşması ihtimali çok yüksektir.

Görüldüğü gibi her iki kontrol yaklaşımının bir birine üstün ve zayıf yönleri bulunmaktadır. Bu iki araştırma alanı da olgun olmaktan çok uzaktır ve yapay zeka ve robotik camiası tarafından üzerinde çalışılmaya devam edilmektedir.

Bu proje’de iki kontrol yaklaşımının da özelliklerinden faydalanan hibrid bir kontrol yaklaşımı tasarlanmıştır. Bu yaklaşımla ilgili detaylar 5. bölümde ele alınacaktır.

5 YAPAY ZEKA İLE MOBİL ROBOT KONTROLÜ PROJESİ

5.1 Hedefler

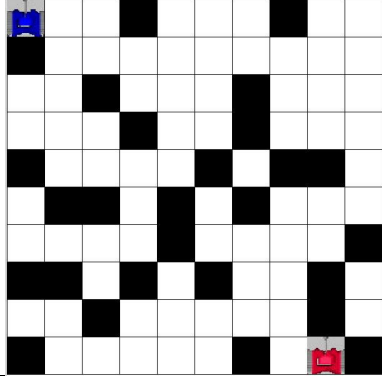
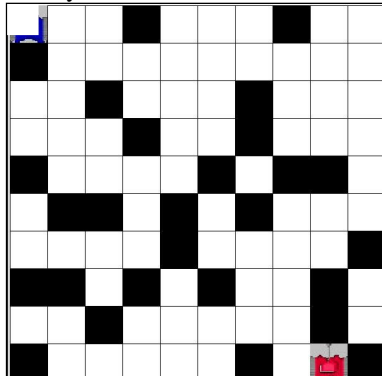
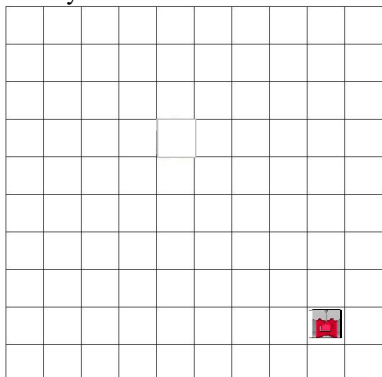
Bu projede ön-tanımlı bir ortamda ızgara tabanlı bir harita kullanarak otonom olarak hareket edebilen, verilen bir haritada hedefe ulaşmak için kendi yolunu bulabilen, gerektiği durumlarda operatör tarafından uzaktan kumanda edilebilen ve zeki davranışlar sergileyebilen mobil bir robotun tasarlanması ve prototip üretimi hedeflenmektedir. Robotun önceden tasarlanmış senaryoların yanı sıra uzaktan kumanda edilme durumlarında da etkin bir şekilde çalışabilmesi ulaşılmayı planlanan ilk hedeftir.

Geliştirilecek mobil robotun mümkün olan en ölçeklenebilir yapıya sahip olması, eklenecek donanımlar ile yeni özellikler kazandırılabilmesi diğer hedeflerdir.

5.2 Robotun Görevleri

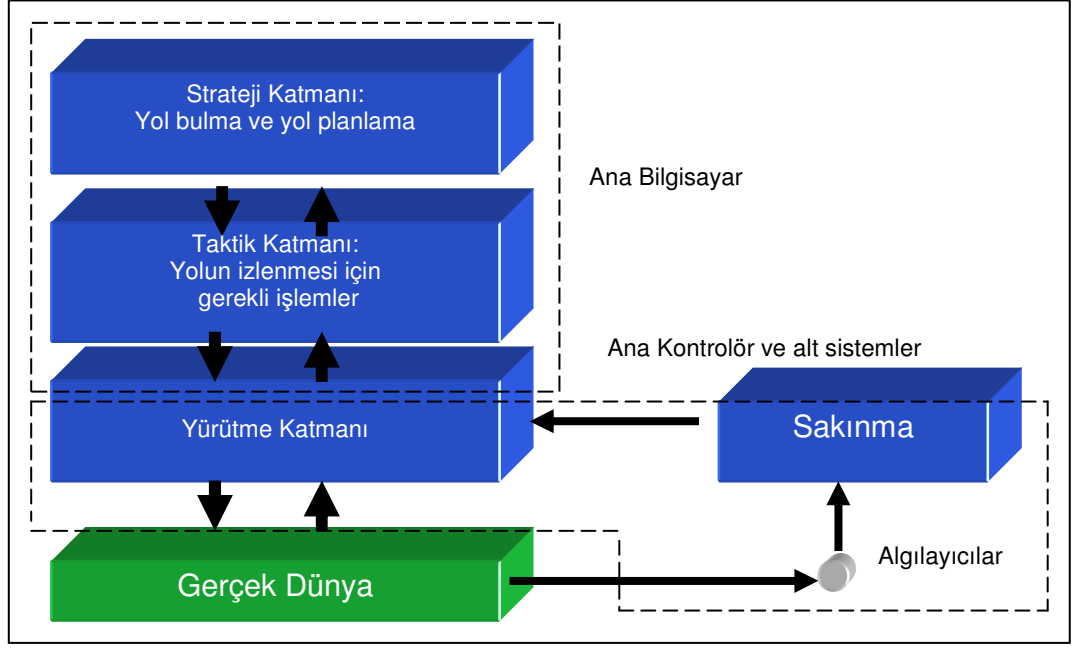
Robot verilen harita üzerinde başlangıç noktasından bitiş noktasına ulaşmak için gerekli çıkarımların yapılması ve uygulanması ile yükümlüdür. Bu özelliklerin sınanabilmesi için 3 farklı senaryo tasarlanmıştır. Bu senaryolar Tablo 5.1'deki şekilde özetlenebilir. Proje sonunda tasarlanan robotun bu senaryolar üzerinde başarımı değerlendirilecektir.

Tablo 5.1 : Senaryoların Özet Gösterimi

Senaryo	Verilenler	İstenenler
Senaryo 1 	Harita veriliyor Başlangıç noktası koordinatları belirli Bitiş Noktası koordinatları belirli	• Harita üzerinde yolunu hesaplamak (strateji) • Yolu izlemesi için gerekli hareketleri belirlemek (Taktik) • Uygulama
Senaryo 2 	Harita veriliyor Başlangıç noktası koordinatları belirli değil Bitiş Noktası koordinatları belirli	• Harita üzerinde bulunduğu koordinatı bulmak (strateji) • Harita üzerinde yolunu hesaplamak (strateji) • Yolu izlemesi için gerekli hareketleri belirlemek (Taktik) • Uygulama
Senaryo 3 	Harita verilmiyor Başlangıç noktası koordinatları belirli değil Bitiş Noktası koordinatları belirli	• Haritayı oluşturmak (strateji) • Harita üzerinde bulunduğu koordinatı bulmak (strateji) • Harita üzerinde yolunu hesaplamak (strateji) • Yolu izlemesi için gerekli hareketleri belirlemek (Taktik) • Uygulama

5.3 Robotun Kontrol Yapısı

Robotun otonom hareketi için bölüm 4.3'te tanımlanan kontrol yapılarından hiyerarşik yaklaşım tercih edilmiştir. Robotun tüm karar verme ve hareketleri hiyerarşik yaklaşım kullanılarak tanımlanmıştır. Aynı zamanda davranış temelli yaklaşımda olduğu gibi bazı davranışların diğerlerini baskılaması da sağlanarak hibrid bir kontrol yapısı oluşturulmuştur. Bu yapının kavramsal açıklaması bu bölümde yapılırken donanım ve yazılım detaylarına ilerleyen bölümlerde değinilecektir.



Şekil 5.1 : Robotun Kontrol Yapısı

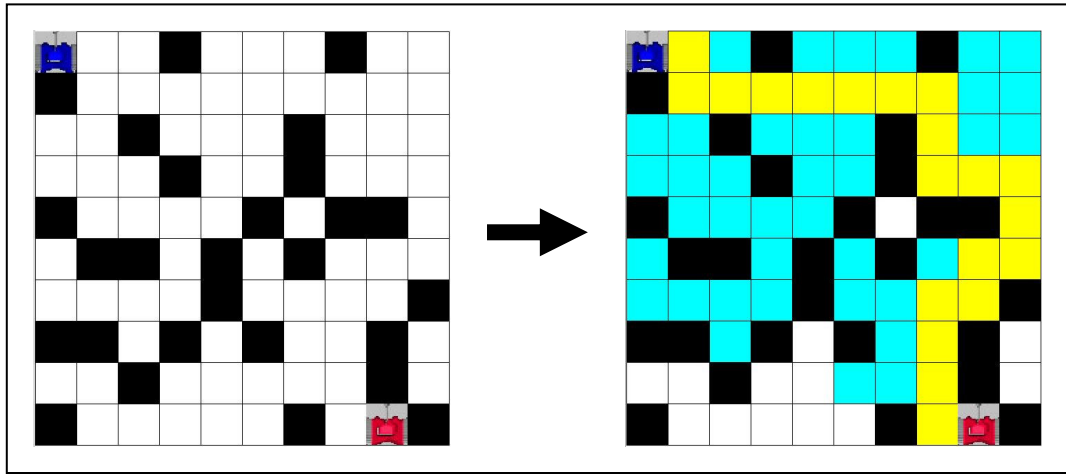
Şekil 5.1’de robotun kontrol yapısı şematik olarak gösterilmiştir. Bu yapı 4 yazılım katmanı ve 2 donanım bölümünden oluşmaktadır. En üst seviyede bulunan strateji katmanı verilen hedefe ulaşılması için yapılması gerekenler ve izlenilmesi gereken yola karar verir. Verilen kararların nasıl uygulanacağı ise taktik katmanı tarafından belirlenmektedir. Taktik katmanı oluşturduğu makro komutları, bu komutların donanım üzerinde uygulanmasını sağlayacak olan uygulama katmanına aktarır. Uygulama katmanı tüm donanım işlevlerinin kontrolünden ve taktik katmanı tarafından verilen makro komutların uygulanmasından sorumludur. Uygulama katmanına paralel olarak çalışan sakınma katmanı robotun engellerden sakınma ve çarpışma algılama özelliklerini barındırır ve birincil öncelik seviyesine sahiptir. Diğer katmanlardan gelen komutlar eğer bir engelle çarpışmaya neden olacak şekilde ise bu sakınma katmanı devreye girerek robotun zarar görmesini veya çevresine zarar vermesini engeller. Strateji ve taktik katmanı tamamıyla ana bilgisayar üzerinde çalışırken, uygulama ve sakınma katmanları ana bilgisayar ve ana kontrolör üzerinde çalışacak şekilde tasarlanmıştır.

Bu katmanların çalışması senaryoların daha detaylı incelenmesi ile çok daha iyi anlaşılabilecektir.

5.3.1 Strateji Katmanı

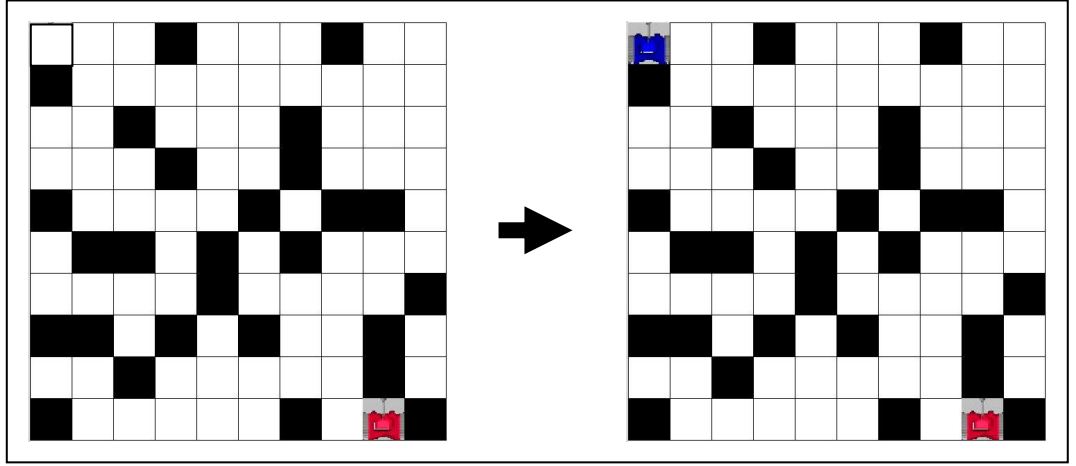
Strateji katmanının görevi verilen hedefe ulaşabilmek için gerekli olan stratejik kararların verilmesidir. Bu katman harita üzerinde yolun bulunması, eğer harita üzerindeki konum veya harita bilinmiyorsa gerekli taktiksel kararların alınması için taktik katmanının etkinleştirilmesi gibi kararlardır. Yol bulma algoritmasının detayları bölüm 5.4’te detaylı olarak ele alınacaktır

Senaryo 1 de olduğu gibi harita, başlangıç ve bitiş konumları biliniyorsa strateji katmanı hedefe ulaşmak için gerekli yolun hesaplamasını yapar (Şekil 5.2). Bu yol bulma işleminin ardından bulunan yolun geçtiği her bir hücrenin koordinatları taktik katmanına aktarılarak kontrol taktik katmanına bırakılır.



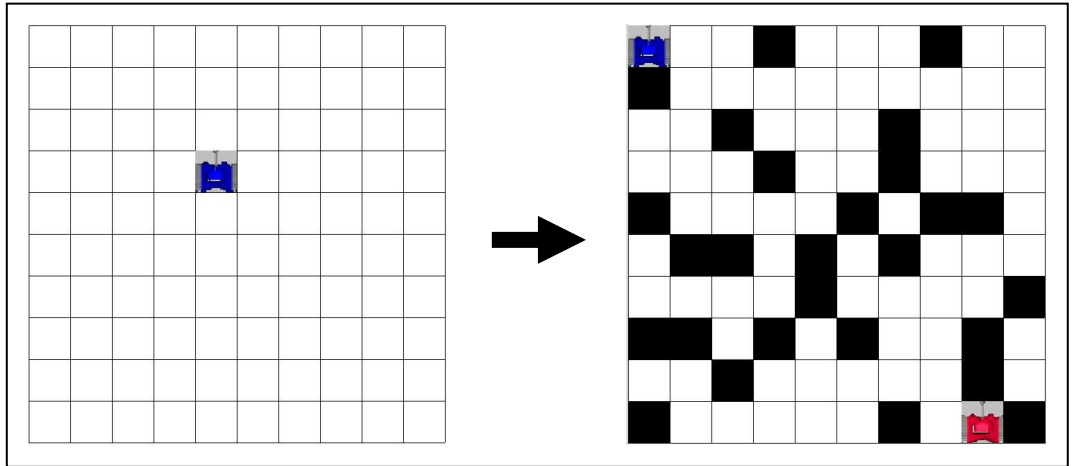
Şekil 5.2 : Strateji Katmanının Senaryo 1 de Çalışması

Senaryo 2 ‘de ise harita bilinmesine rağmen robotun haritadaki başlangıç komutu bilinmemektedir. Yolun hesaplanması için başlangıç konumunun algılayıcılar yardımıyla öğrenilmesi gerekir. Bu nedenle strateji katmanı taktik katmanına “konum belirle” komutunu gönderir. Taktik katmanının konumunu belirlemesinin ardından durum senaryo 1’in başlangıç konumuna indirgenmiş olur (Şekil 5.3) ve strateji katmanı senaryo 1 de olduğu gibi çalışmaya devam eder (Şekil 5.2).



Şekil 5.3 : Strateji Katmanının Senaryo 2 de Çalışması

Senaryo 3’te ise robot haritayı ve harita üzerinde bulunduğu konumu bilmemektedir (Şekil 5.4). Böyle bir durumda yapması gereken ilk olarak haritayı öğrenmesidir. Bunun için taktik katmanı etkinleştirilerek haritanın çıkarılması sağlanır. Haritanın çıkarılması sırasında robotun harita üzerindeki konumu da belirlenmiş olacağı için durum senaryo 1 e indirgenmiş olur. Strateji katmanı hedefe ulaşmak için yolunu hesaplar ve kontrolü taktik katmanına aktarır.

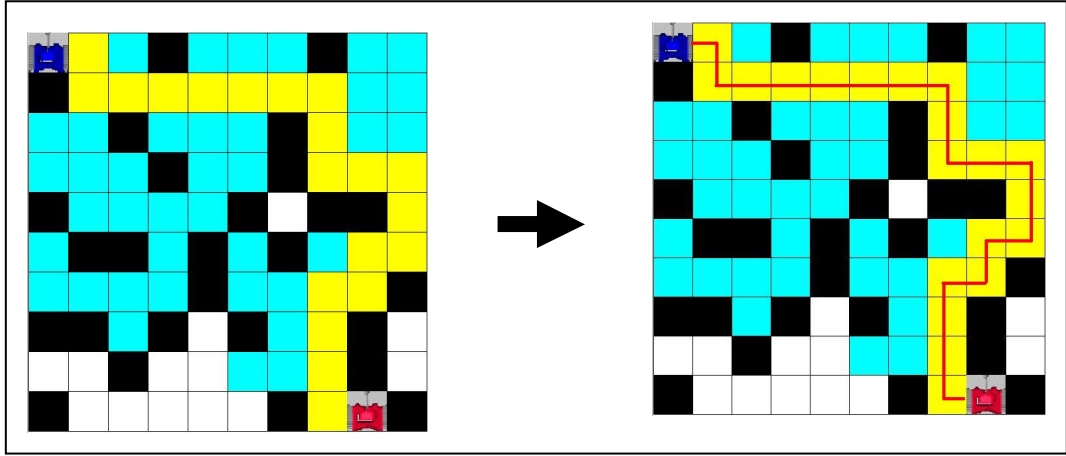


Şekil 5.4 : Strateji Katmanının Senaryo 3 de Çalışması

5.3.2 Taktik Katmanı

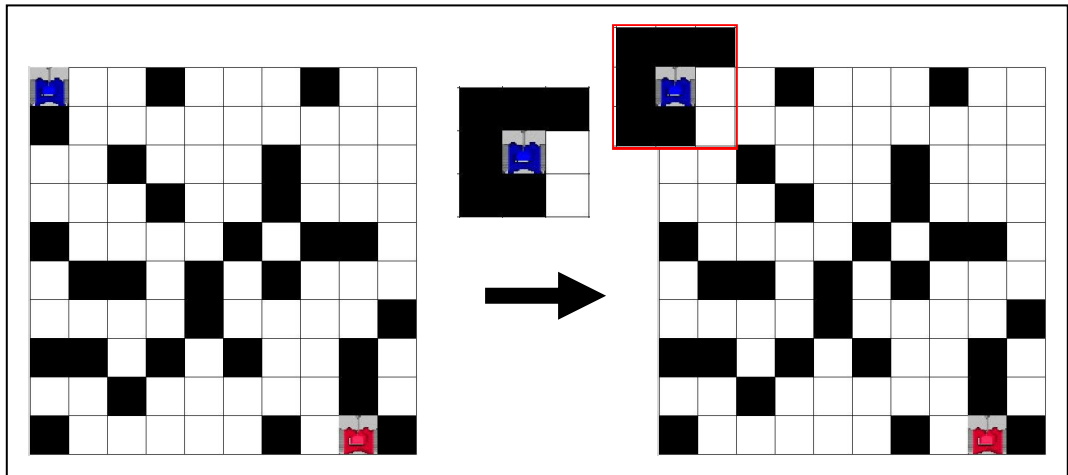
Taktik katmanı strateji katmanının verdiği kararların yorumlanmasından sorumludur. Strateji katmanının kararlarının nasıl uygulanacağı belirlenir ve uygulama katmanının bu kararları nasıl gerçekleştireceği hesaplanır. Taktik katmanı aynı zamanda robotun harita üzerindeki yerinin belirlenmesi ve haritanın bilinmediği durumlarda haritanın çıkarılması görevlerini yerine getirerek strateji katmanının karar almasına yardımcı olur.

Senaryo 1’de strateji katmanının bulduğu yolun koordinatları ilk olarak rotaya ardından da motor komutlarına çevrilmesi yapılır. (Şekil 5.5). Oluşturulan makro hareket komutları uygulama katmanına aktarılarak uygulanması sağlanır.



Şekil 5.5 : Taktik Katmanının Senaryo 1 de Çalışması

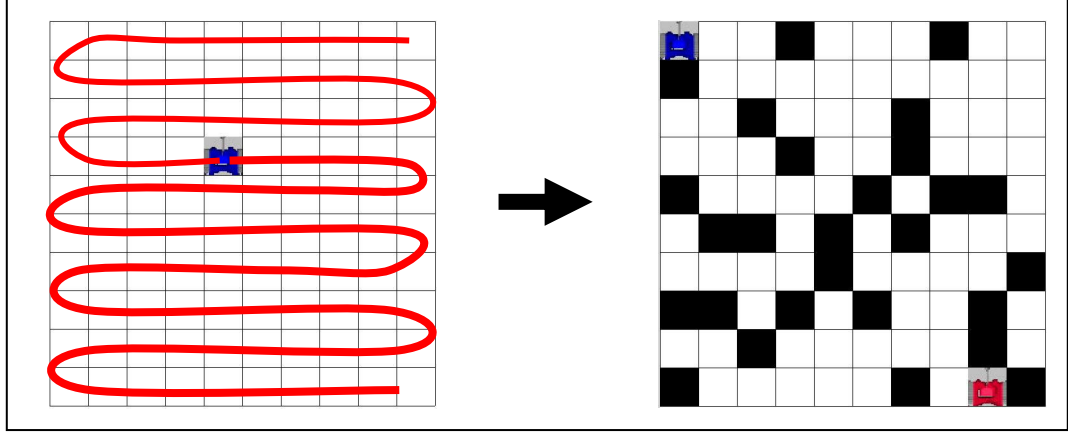
Senaryo 2 ‘de ise strateji katmanından “konum bul” komutunun ardından uygulama katmanına konumun bulunması için gerekli algılayıcı komutları gönderilir. Uygulama katmanından alınan verilere dayanılarak robotun harita üzerindeki yeri belirlenmeye çalışılır. Ardından belirlenen bu konum strateji katmanına geri bildirilerek gerekli kararların alınması sağlanır. Şekil 5.6’da bu işlem şematik olarak gösterilmiştir.



Şekil 5.6 : Taktik Katmanının Senaryo 2 de Çalışması

Senaryo 3’te olduğu gibi haritanın bilinmediği durumlarda ise strateji katmanı taktik katmanına haritayı oluşturması için komut gönderir. Bunun ardından taktik katmanı haritanın oluşturulması için gerekli olan uygulama katmanı komutlarını oluşturur. Uygulama katmanı ile eş zamanlı yürütülen bu çalışmanın ardından oluşturulan

harita ve konum bilgisi strateji katmanına aktarılarak gerekli işlemin yapılması sağlanır. Bu işlemin en basit hali Şekil 5.7’de şematik olarak gösterilmiştir.



Şekil 5.7 : Taktik Katmanının Senaryo 3 de Çalışması

5.3.3 Uygulama Katmanı

Taktik katmanından gelen makro komutların donanım komutlarına çevrilmesi, tüm haberleşme işlemlerini gerçekleştirilmesi, alt seviye karar ve hata düzeltme işlemlerini gerçekleştirilmesi, algılayıcı verilerini okunup, değerlendirilip üst katmanlara iletilmesi uygulama katmanının görevleridir.

Uygulama katmanı 3 ana bölümden oluşmaktadır. Bunlardan birincisi ana bilgisayar üzerinde çalışan ve taktik katmanının oluşturduğu makro komutları donanım komutlarına çeviren komut işleme bölümüdür. Ana kontrolör üzerinde çalışan “komut yürütme bölümü” ise ana bilgisayardan gelen alt seviye donanım komutlarının yürütülmesinden sorumludur.

3. bölüm olan haberleşme bölümü ise ana bilgisayar ve ana kontrolör arasındaki iletişimi sağlamaktan sorumludur. Haberleşme görevi ana bilgisayar ve ana kontrolör üzerinde çalışan alt programlar tarafından gerçekleştirilmektedir.

Ana bilgisayar tarafında yer alan her bir komut için ana kontrolörde karşılık olarak komutlar bulunur. Bu komutları hareket komutları, motor komutları, algılayıcı konumlandırma ve algılayıcı komutları olarak 4 grupta incelemek mümkündür.

Hareket komutları taktik katmanının oluşturduğu makro komutların donanım karşılığıdır. Bu komutlar ileri, sağa dön ve sola dön komutlarıdır. Hareket komutları kendi içerisinde algılayıcı verilerine göre hata düzeltme ve konum kontrolü gibi özellikleri içerirler ve üst seviye uygulama komutlarıdır. Aynı zamanda uygulama

katmanının oluşturacağı “yeniden başla” ve “tüm sistemleri başlat” komutları da sisteme entegre edilmiştir. Bu komutlar ve işlevleri **Tablo 5.2**’de gösterilmiştir.

Tablo 5.2 : Uygulama Katmanı Makro Komutları ve İşlevleri

Komut	Seviyesi	İşlevi
forward	Üst seviye makro komut	Robotun haritada bir hücre ilerlemesini sağlar
right	Üst seviye makro komut	Robotun bulunduğu hücre içerisinde 90 derece sağa dönmesini sağlar.
left	Üst seviye makro komut	Robotun bulunduğu hücre içerisinde 90 derece sola dönmesini sağlar
reset	Üst seviye makro komut	Robotun ana kontrolör üzerinde bulunan tüm alt sistemlerinin yeniden başlatılmasını sağlar.
init	Üst seviye makro komut	Robotun başlangıç için gerekli tüm algılayıcılarının okunması ve alt sistemlerinin hazırlanmasını sağlar

Motor komutları ise donanımsal olarak ana kontrolör kartına gönderilen alt seviye komutlardır. Bu komutlar bir dizi şeklinde kullanılarak karmaşık hareketlerin elde edilmesi mümkündür. Algılayıcı komutları ile birlikte kullanılarak üst seviye komutlar olan makro hareket komutlarının gerçekleştirilmesinde kullanılırlar. Bu komutların detayları **Error! Not a valid bookmark self-reference.**’de gösterilmiştir.

Tablo 5.3 : Uygulama Katmanı Motor Komutları

Komut	Seviyesi	İşlevi
Go Forward	Üst seviye motor komutu	Motorların ileri gidecek şekilde çalışmasını sağlar. forward komutun işletilmesinde algılayıcı komutları ile birlikte kullanılır
Go Backward	Üst seviye motor komutu	Motorların geri gidecek şekilde çalışmasını sağlar.
turnright	Üst seviye motor komutu	Motorların sağa dönecek şekilde çalışmasını sağlar. Right komutun işletilmesinde algılayıcı komutları ile birlikte kullanılır
turnleft	Üst seviye motor komutu	Motorların sola dönecek şekilde çalışmasını sağlar. Left komutun işletilmesinde algılayıcı komutları ile birlikte kullanılır
Dc_left	Alt seviye motor komutu	Motorların sola dönecek şekilde kısa bir süre çalışmasını sağlar. Üst seviye hareket komutlarının uygulanması sırasında hata düzeltme amaçlı kullanılır.
Dc_right	Alt seviye motor komutu	Motorların sağa dönecek şekilde kısa bir süre çalışmasını sağlar. Üst seviye hareket komutlarının uygulanması sırasında hata düzeltme amaçlı kullanılır
Dc_fwd1	Alt seviye motor komutu	1 numaralı dc motorun ileri yönde kısa bir süre dönmesini sağlar.
Dc_rew1	Alt seviye motor komutu	1 numaralı dc motorun geri yönde kısa bir süre dönmesini sağlar
Dc_fwd2	Alt seviye motor komutu	2 numaralı dc motorun ileri yönde kısa bir süre dönmesini sağlar
Dc_rew2	Alt seviye motor komutu	2 numaralı dc motorun geri yönde kısa bir süre dönmesini sağlar

Algılayıcı komutları robot üzerinde bulunan algılayıcıların okunması ve konumlandırılması ile ilgili komutlardır. Hareket komutlarının uygulanmasında ve hata düzeltmede bu komutlar sonucu okunan veriler kullanılmaktadır. Bu komutlar arasında alt seviye algılayıcı ve konumlandırma komutlarının birleşiminden oluşan

üst seviye komutlarda bulunmaktadır. Algılayıcı ve konumlandırma komutlarının listesi **Tablo 5.4**'te verilmiştir.

Tablo 5.4 : Uygulama Katmanı Algılayıcı ve Konumlandırma Komutları

Komut	Seviyesi	İşlevi
SonarScan	Üst seviye algılayıcı komutu	Sonarın 180 derecelik bir tarama yapmasını sağlar
IRScan	Üst seviye algılayıcı komutu	Kızılötesi uzaklık ölçerin 180 derecelik tarama yapmasını sağlar
Servo LeftSweep1	Üst seviye algılayıcı komutu	1 numaralı servo motorun sola doğru tarama yapmasını sağlar.
Servo RightSweep1	Üst seviye algılayıcı komutu	1 numaralı servo motorun sağa doğru tarama yapmasını sağlar.
Servo LeftSweep2	Üst seviye algılayıcı komutu	2 numaralı servo motorun sola doğru tarama yapmasını sağlar.
Servo RightSweep2	Üst seviye algılayıcı komutu	2 numaralı servo motorun sağa doğru tarama yapmasını sağlar.
Servo LeftSweep3	Üst seviye algılayıcı komutu	3 numaralı servo motorun sola doğru tarama yapmasını sağlar.
Servo RightSweep3	Üst seviye algılayıcı komutu	3 numaralı servo motorun sağa doğru tarama yapmasını sağlar.
Servo LeftSweep4	Üst seviye algılayıcı komutu	4 numaralı servo motorun sola doğru tarama yapmasını sağlar.
Servo RightSweep4	Üst seviye algılayıcı komutu	4 numaralı servo motorun sağa doğru tarama yapmasını sağlar.
Servo LeftSweep5	Üst seviye algılayıcı komutu	5 numaralı servo motorun sola doğru tarama yapmasını sağlar.
ServoRightSweep5	Üst seviye algılayıcı komutu	5 numaralı servo motorun sağa doğru tarama yapmasını sağlar.
Servo Left1	Alt seviye algılayıcı komutu	1 numaralı servonun sola 1 adım dönmesini sağlar
Servo Right1	Alt seviye algılayıcı komutu	1 numaralı servonun sağa 1 adım dönmesini sağlar
Servo Left2	Alt seviye algılayıcı komutu	2 numaralı servonun sola 1 adım dönmesini sağlar
Servo Right2	Alt seviye algılayıcı komutu	2 numaralı servonun sağa 1 adım dönmesini sağlar
Servo Left3	Alt seviye algılayıcı komutu	3 numaralı servonun sola 1 adım dönmesini sağlar
Servo Right3	Alt seviye algılayıcı komutu	3 numaralı servonun sağa 1 adım dönmesini sağlar
Servo Left4	Alt seviye algılayıcı komutu	4 numaralı servonun sola 1 adım dönmesini sağlar
Servo Right4	Alt seviye algılayıcı komutu	4 numaralı servonun sağa 1 adım dönmesini sağlar
Servo Left5	Alt seviye algılayıcı komutu	5 numaralı servonun sola 1 adım dönmesini sağlar
Servo Right5	Alt seviye algılayıcı komutu	5 numaralı servonun sağa 1 adım dönmesini sağlar
Center1	Alt seviye algılayıcı komutu	1 numaralı servonun merkez konuma dönmesini sağlar
Center2	Alt seviye algılayıcı komutu	2 numaralı servonun merkez konuma dönmesini sağlar
Center3	Alt seviye algılayıcı komutu	3 numaralı servonun merkez konuma dönmesini sağlar
Center4	Alt seviye algılayıcı komutu	4 numaralı servonun merkez konuma dönmesini sağlar
Center5	Alt seviye algılayıcı komutu	5 numaralı servonun merkez konuma dönmesini sağlar
Read Sonar1	Alt seviye algılayıcı komutu	1 numaralı sonarın uzaklık ölçümü için kullanılır
Read Sonar2	Alt seviye algılayıcı komutu	2 numaralı sonarın uzaklık ölçümü için kullanılır
Read IR1	Alt seviye algılayıcı komutu	1 numaralı kızıl ötesi algılayıcının uzaklık ölçümü için kullanılır
Read IR2	Alt seviye algılayıcı komutu	2 numaralı kızıl ötesi algılayıcının uzaklık ölçümü için kullanılır
Read IR3	Alt seviye algılayıcı komutu	3 numaralı kızıl ötesi algılayıcının uzaklık ölçümü için kullanılır
Read IR4	Alt seviye algılayıcı komutu	4 numaralı kızıl ötesi algılayıcının uzaklık ölçümü için kullanılır
Read Compass	Alt seviye algılayıcı komutu	Elektronik pusuladan yönelim bilgisinin okunması için kullanılır
Read Light	Alt seviye algılayıcı komutu	Işık şiddeti algılayıcısının okunması için kullanılır
Read Batery	Alt seviye algılayıcı komutu	Akülerin durumunu ölçmek için kullanılır.
Read PIR	Alt seviye algılayıcı komutu	Pasif infrared hareket algılayıcının okunması için kullanılır.
Read Temp	Alt seviye algılayıcı komutu	Sıcaklı algılayıcısının okunması için kullanılır.
ReadTouch	Alt seviye algılayıcı komutu	Dokunma algılayıcıların okunmasında kullanılır
Read AccelX	Alt seviye algılayıcı komutu	İvme ölçerin X eksenini üzerindeki değerinin okunmasında kullanılır
Read AccelY	Alt seviye algılayıcı komutu	İvme ölçerin Y eksenini üzerindeki değerinin okunmasında kullanılır

5.4 Yol Bulma Algoritması

Projenin strateji katmanında yer alan yol bulma algoritması Dijkstra, BFS ve A* algoritmaları baz alınarak geliştirilmiştir. Dijkstra ve A* algoritmalarının en iyi yolu bulma garantisi ve BFS algoritmasının hızı bu algoritma altında sağlanmaya çalışılmıştır.

Robotun operasyonel başarısı için en kısa yolun garantilenmesi en önemli kriter olarak alınmıştır. Bu nedenle BFS’de kullanılan yönlü yayılma yerine her yöne yayılma tercih edilmiştir. Kullanılan haritaların karmaşıklık gösterebilmesi nedeniyle BFS’nin oluşturacağı uzun yollar operasyonel nedenle tercih edilmemiştir.

A* algoritmasında kullanılan açık ve kapalı kümeler yerine bu algoritma tek bir küme üzerinde çalışmaktadır. Bu küme hem haritayı hem de incelenmiş her bir hücrenin erişim maliyetini barındırmaktadır. Böylece işlemci yükü ve hafıza gereksinimleri en aza indirgenmeye çalışılmıştır. Aynı zamanda A* algoritmasında kullanılan ikili maliyet fonksiyonu yerine tek kademeli maliyet fonksiyonu kullanılmıştır. Bu maliyet fonksiyonu her bir hücrenin başlangıç hücresine olan uzaklığını belirtmektedir.

Yol bulma algoritması 2 ana döngüden oluşmaktadır. Birinci döngü hücrelerin taranmasını ve maliyet hesaplarını gerçekleştirirken 2. döngü en düşük maliyetli yolun seçimini yapmaktadır.

Harita büyüklüğünün $I \times J$ şeklinde olduğu bir harita, 2 boyutlu bir dizi olarak tanımlanmıştır. Bu dizinin indisleri her bir hücrenin harita üzerindeki x ve y koordinatlarını belirtmektedir. Başlangıç hücresi bu dizide “1”, bitiş hücresi ise “2” ile belirtilmektedir. Harita üzerinde geçilebilir hücreler “0” geçilemeyen hücreler ise “9” değeri ile matris üzerinde işaretlenmektedir.

İlk olarak başlangıç değeri 0 olan toplam indekisleme maliyeti “tim” tanımlanmaktadır. Bu değer geçilebilir her bir hücre için maliyet değerini barındırmanın yanı sıra bir sonraki adımda taranacak hücreleri de belirlemektedir. Matris içerisindeki “0”, “1”, “2” ve “9” değerleri ile çakışmaması için işlenecek her bir hücreye 100+tim değeri eklenmektedir. Tanımlanan diğer bir değişken ise hedefe ulaşıp ulaşılmadığının kontrolünü yapan “reachable” mantıksal değişkenidir.

Algoritma çalışırken ilk olarak başlangıç hücresinin maliyetini sıfır ($100+tim$) olarak işaretler ve birinci ana döngüye başlar. Maliyeti sıfır olan başlangıç hücresinin komşu hücrelerine doğru yayılmaya başlar. Taranan komşu hücre eğer boş ise bir sonraki adımda işlenmek üzere $100+tim+1$ değeri ile işaretlenir. Aynı zamanda her bir komşu hücrenin hedef hücre olup olmadığı da test edilir. Eğer taranan komşu hücre hedef hücre ise reachable değeri true olarak atanarak birinci ana döngünün sonuçlanması sağlanır. Eğer ilk adımda hedefe ulaşamaz ise ilk olarak tüm hücrelerin değerleri toplanır ve kaydedilir ardından tim değeri bir artırılarak bir sonraki çevrime geçilir. Bir sonraki çevrimde değeri tim değerine eşit olan hücreler taranır ve bu hücrelerin komşularına doğru yayılma gerçekleştirilir. Aynı zamanda her bir komşu hücrenin hedef hücre olup olmadığı da test edilir. Eğer taranan komşu hücre hedef hücre ise reachable değeri true olarak atanarak birinci ana döngünün sonuçlanması sağlanır. Çevrim sonunda hedefe ulaşamaz ise tüm hücrelerin değerleri toplanarak bir önceki çevrimde elde edilen toplam değer ile karşılaştırılır. Bu iki değer aynı olması artık yayılmaya müsait hücre bulunmadığının göstergesidir. Böyle bir durumda hedefe ulaşamayacağı anlaşılarak algoritma sonlandırılır. İki toplam birbirinden farklı ise bir sonraki adımda işlenecek hücrelerin bulunduğu anlaşılır ve döngüye devam edilir.

Birinci ana döngünün başarı ile sonuçlanması döngü içerisinde hedefe ulaşıldığının göstergesidir.²² Birinci döngü sonucunda elde edilen tim değeri ise hedefe ulaşmak için gereken toplam maliyeti belirtmektedir. Bu durumda ikinci döngüye geçilir. Bu döngünün görevi en uygun yolun seçilmesidir. Diğer algoritmalarından farklı olarak burada hedeften başlangıca doğru bir tarama yapılmaktadır. İlk adımda hedef hücrenin koordinatları yolun son hücresi olarak kaydedilir. Ardından komşu hücreleri taranarak en düşük maliyetli olan yol dizisine eklenir. Diziye eklenen hücre başlangıç hücresinden farklı olduğu sürece bu işleme devam edilir. İkinci döngünün sonucunda başlangıç hücresinden hedefe ulaşan en kısa yol hesaplanmış olur.

Bu yol bulma algoritması ile ilgili pseudo kodu Tablo 5.5'te verilmiştir. Bu pseudo kodun yazılım uygulaması 7. bölümde detaylı olarak incelenecektir.

²² Aksi bir durumda reachable değeri hiçbir zaman true değerine ulaşmayacağı için algoritma 1. döngü başarıyla tamamlanmadan sonlandırılacaktır.

Tablo 5.5 : Yol Bulma Algoritması Pseudo Kodu

```
Tim=0
erişilebilir değeri= doğru değil
Do (erişilebilir değeri doğru değil)
  For i=0 to mapsize x
    For j=0 to mapsize y
      eğer hücre değeri = 100+ tim ise{
        hücrenin komşu hücrelerini tara
        eğer Komşu hücre = boş ise {
          komşu hücreyi bir sonraki adımda işlenmek üzere işaretle
        }
        eğer Komşu hücre= hedef ise{
          hedefe ulaşıldığını belirlemek üzere erişilebilir değerini doğru olarak ata
        }
      }
    Next
  Next

Eğer eğer erişilebilir değeri doğru değil ise{
  Tüm hücrelerin değerlerini topla
  Eğer tüm hücrelerin değerleri toplamı = son toplam ise{
    Hedefe ulaşılamıyor
    İşlemi sonlandır
  }
  Eşit değil ise {
    Son toplam = tüm hücrelerin değerleri toplamı
  }
  Tim değişkenini 1 artır
  Loop

Yol uzunluğu =tim
Yol dizisinin boyutu= Yol uzunluğu
Son eklenen hücre =hedef hücre

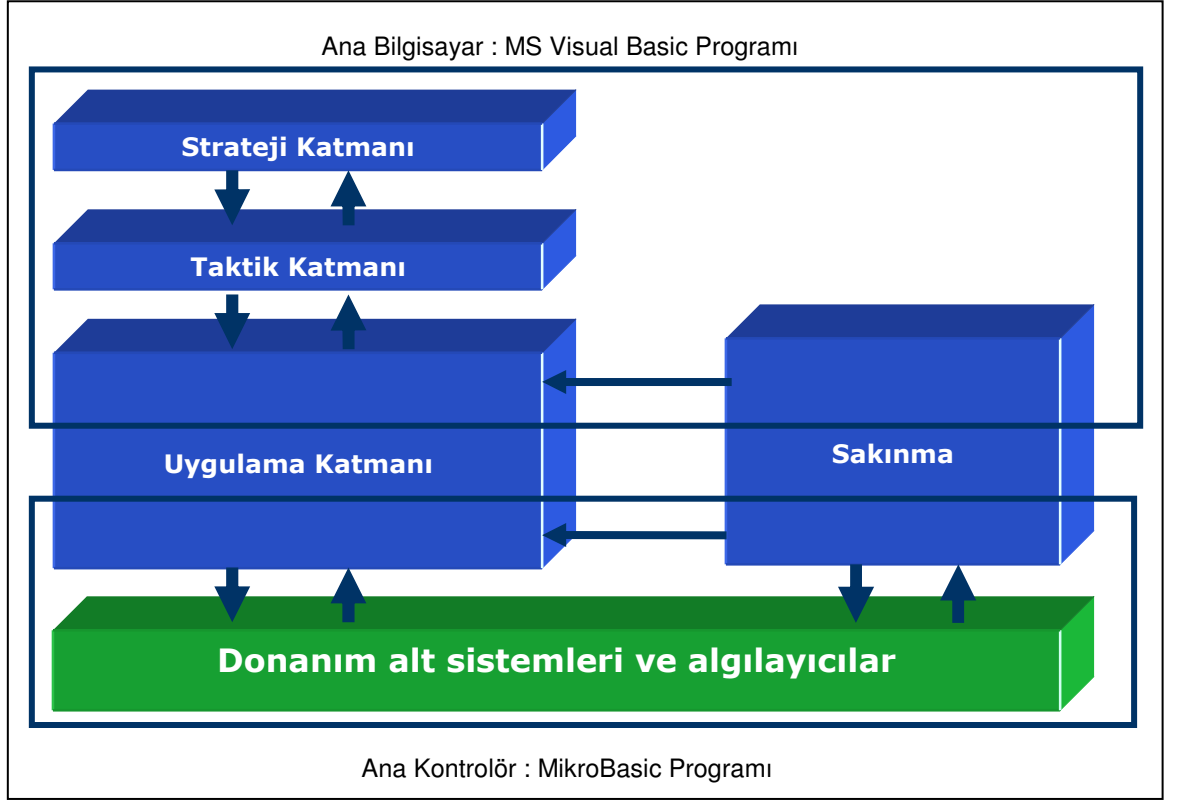
do (son eklenen hücre, başlangıç hücresinden farklı olduğu sürece)
  tim değerini bir azalt
  son eklenen hücrenin komşu hücrelerini tara
  en düşük tim değerine sahip olanı yol dizisine ekle
loop
```

5.5 Robotun Genel Yapısı

Robotun kontrol yapısının yanı sıra donanım ve yazılım yapısı da robotun işlevselliği açısından büyük önem taşımaktadır. Robot ana kontrol bilgisayarı ve ana kontrol kartına bağlı alt donanımlardan oluşmaktadır. Bu iki alt donanım sistemi arasındaki haberleşme kablosuz olarak gerçekleştirilmektedir.

Robotun yazılım alt yapısı ise yine ana bilgisayar ve ana kontrolör üzerinde çalışan programlardan oluşmaktadır. Ana bilgisayar üzerinde Ms Visual Basic üzerinde çalışan program kontrol yapısının strateji, taktik ve uygulama katmanlarını barındırmaktadır. Ana kontrolör üzerinde çalışan mikrobasic programı ise uygulama katmanının bir bölümünü oluşturmaktadır. Sakınma katmanı ise hem ana kontrolör

hem de ana bilgisayar üzerinde çalışan alt programlar ile gerçekleştirilmiştir. Tüm bu sistemlerin genel yapısı ve ilişkileri Şekil 5.8’de şematik olarak gösterilmiştir. Donanım ve yazılım detayları ilerleyen bölümlerde irdelenecektir.



Şekil 5.8 : Robotun Genel Yapısı

6 DONANIM

Bu bölümde robotun donanım yapısı ve detayları ele alınacaktır. İlk olarak alt sistemler ve algılayıcıların çalışmasının daha iyi anlaşılabilmesi için takip eden bölümde kullanılan teknolojiler hakkında kısa teknik bilgi verilecektir.

Ardından robotu oluşturan donanım alt sistemleri, bu alt sistemlerin birbiri ile ilişkileri ve bu sistemlerin çalışma prensipleri incelenecektir. Robotu oluşturan alt donanım sistemleri; ana kontrolör, dc motor kontrolörü, servo motor kontrolörü, hareket sistemleri, algılayıcılar, haberleşme sistemleri, görüntü ve ses sistemleri ve güç sistemleri olarak sıralanabilir. Her bir sistem ilgili alt başlıkta detaylı olarak incelenecek ve diğer sistemler ile etkileşimi anlatılacaktır.

6.1 Kullanılan Teknolojiler ve Çalışma Prensipleri

6.1.1 PIC Mikro İşlemci

Robotun donanım tasarımına başlanırken piyasada bulunan farklı işlemci türleri (pic, avr, basic stamp, picaxe vb.) incelenmiş, maliyeti kolay temini ve çok değişik fonksiyonel gereksinimlere cevap verebilmesi nedeniyle Microchip firmasının PIC²³ mikro işlemci ailesinin kullanılmasına karar verilmiştir.

PIC mikro işlemci ailesi günümüzde endüstriyel uygulamalarda en çok tercih edilen mikro işlemcilerden biridir. Bu ürün ailesinde çok kısıtlı giriş çıkış pin²⁴ sayısına sahip alt modeller olduğu gibi; üzerinde analog²⁵ dijital çeviriciler, USART, I2C, USB vb gibi arabirimlere sahip çok pin sayılı üst modellerde bulunmaktadır. Ürün ailesi farklı boyutlarda hafıza ve donanım yeteneklerine sahip birçok modeli barındırmaktadır.

PIC mikro işlemciler aynı zamanda farklı fiziksel yapılarda piyasaya sunulmuştur. Breadboard²⁶ ve manuel uygulamalar için uygun olan 0,1 inç pin aralığına sahip olan DIP²⁷, PDIP²⁸ ve SPDIP²⁹ şekillerinin yanı sıra elektronik devrelere yüzey montajı

²³ Programmable interface controllers

²⁴ Mikroişlemcinin giriş-çıkış işlemlerini gerçekleştirdiği her bir fiziksel bağlantı, metal ayak

²⁵ Örnekse

²⁶ Elektronik devrelerin prototiplerinin tasarlanmasında kullanılan delikli devre kartı

²⁷ Dual in line package

iin kullanılan fiziksel formlarda bulunmaktadır. PIC'ler programlanma metodları ve tekrar kullanılabilirlikleri aısından da farklılık gösterirler. Bazı PIC mikro ilemciler tek sefer programlanabilir yapıya sahip olup yüksek miktarda üretimi yapılan endüstriyel ürünlerde tercih edilmektedirler. Diğer bir ürün türü ise üzerinde morötesi ışınları geçiren bir penceresi olan ürünlerdir. Bu mikro ilemciler programlandıktan sonra morötesi lambalar kullanılarak silinebilirler. Son geliştirilen PIC mikro ilemciler ise elektriksel olarak programlanabilen ve silinebilen (flash) yapıya sahiptirler. Bu projede flash yapıya sahip mikro ilemciler kullanılmıştır.

PIC mikro ilemciler giriş ıkış ve haberleşme işlemleri iin port adı verilen yapılar kullanılır. Genellikler her bir port 8 pinden oluşmaktadır. Bu portlar sadece giriş ıkış özelliğine sahip olabildiği gibi bazıları analog dijital çevrim ve haberleşme arabirimlerini de barındırır. Mikro ilemci her bir portun özelliğini ve değerini belirleyen ve yazma³⁰ adı verilen 2 fiziksel hafıza birimi barındırır. Bu hafıza adreslerinin içeriği değıştirilerek portun giriş veya ıkış iin kullanımı ve değeri değıştirilebilir. Giriş ıkış dışında başka işlevleri bulunan portlar iin farklı yazma değerleri de tanımlanmıştır. Bu değerler her bir ilemci iin farklılık göstermektedir. Bir PIC mikro ilemci port pinlerinin yanı sıra 5 önemli pin daha barındırır. Bunlar VDD, VSS, MCLR, CLKIN ve CLKOUT pinleridir. VSS ve VDD pinleri ilemciyi beslemek iin kullanılırken MCLR pini ilemcinin resetlenmesi iin kullanılmaktadır. PIC mikro ilemciler dahili kristallere sahip olmalarına rağmen CLKIN ve CLKOUT pinlerine harici bir kristal bağlanarak ilemci alışma frekansı belirlenebilmektedir. Bu alışma frekansı 16F ve 18F PIC mikro ilemci serileri iin 4Mhz ile 40Mhz arasında değışmektedir.

PIC mikro ilemciler Microchip firmasının geliştirdiği ilemci komut setini kullanırlar ve assembler dili ile programlanırlar. Derlenen assembler programı intel hex formatına uygun bir biçim alır ve mikro ilemciye programlayıcı devreler yardımıyla aktarılır. Günümüzde C, pascal, basic gibi daha üst seviye dilleri kullanan derleyiciler de geliştirilmiştir. Bu derleyicilerin kullanımı, tasarım ve programlama aşamalarının daha kısa sürede tamamlanmasını sağlamaktadır. Bu proje kapsamında Mikroelektronika firmasının geliştirdiği mikrobasic programlama dili kullanılmıştır.

²⁸ Plastic Dual in line package

²⁹ Skinny PDIP

³⁰ İng:Yazma

6.1.2 IEEE 802.11 B ve G standartları

Günümüzde yaygın olarak kullanılan bu standartlar IEEE tarafından geliştirilen 802.11 standardının [39] türevleridir ve OSI katmanlarından datalink katmanında çalışan protokollerdir. Kablosuz ağ bağlantılarını mümkün kılan bu protokoller 2.4 Ghz frekansında çalışırlar ve 11Mbps ve 54Mbps hızlarında iletişime olanak sağlarlar. Günümüzde bazı donanım üreticileri farklı sıkıştırma teknikleri kullanarak 112 MBps hızında iletişim sağlayan cihazlar geliştirmişleridir. 802.11b standardı DSSS³¹ ve CCK³² kodlamalarını kullanırken 802.11g standardı 20MBps'nin altındaki hızlarda CCK ve DSSS, 20Mbps'nin üstündeki hızlarda OFDM³³ kodlamasını kullanır.

802.11b standardının ham veri iletişim hızı 11Mbps olmasına rağmen kullanılan CSMA/CA erişim kontrolünün getirdiği yük nedeniyle TCP protokolünde 5.9 Mbit/s UDP protokolünde 7.1 Mbit/s hızlarına erişilebilmektedir. Kapalı mekanlarda kapsama alanı 90 metreyi bulurken açık havada yüksek kazançlı antenler kullanılarak birbirini gören iki nokta arasında 80-120 km mesafeye kadar çıkılabilmektedir.

802.11G standardı ise aynı frekansı kullanmasına rağmen kullandığı kodlama yöntemi nedeniyle daha yüksek hızlara ulaşabilmektedir.

Her iki standardında diğer kablosuz iletişim standartlarına göre birçok avantajı bulunmaktadır:

- Lisans gerektirmeyen frekanslarda çalışırlar
- Ağ için kablolu gereksinimi yoktur, böylece kablo çekilemeyecek binalarda veya binalar arası bağlantılarda kolaylık sağlarlar.
- Diğer kablosuz çözümlere göre çok daha ucuz ve kolay alınıp kurulabilirler.
- Birden çok kablosuz erişim noktası kullanılan ağlarda, kablosuz dolaşım ile kablosuz iletişim kesilmeden bir erişim noktasından diğerine geçiş yapılabilir.
- WEP, WPA ve benzeri kablosuz şifreleme yöntemleri veya IEEE 802.1x gibi yetkilendirme yöntemleriyle çeşitli güvenlik seçenekleri sunarlar.

Bu avantajlarının yanı sıra bu standartların dezavantajları da bulunmaktadır.

³¹ direct-sequence spread spectrum

³² Complementary Code Keying

³³ Orthogonal Frequency Division Multiplexing

Lisans gerektirmeyen frekans aralıklarında çalıştığı için, Wi-Fi cihazlar diğer kablosuz cihazlarla çakışabilir veya birbirlerinin iletişimini engelleyebilirler.

2.4 GHz frekans aralığında çalışan 802.11b ve 802.11g uyumlu cihazların iletişim kalitesi ve hızı, diğer Wi-Fi cihazlar dışında, Bluetooth, mikrodalga fırın, telsiz telefon, bazı telsizler ve benzeri radyo sinyalleriyle çalışan cihazlar tarafından düşürülebilir veya tamamen engellenebilir.

Wi-Fi için yapılan uluslararası düzenlemelerin tümü aynı olmadığı için değişik ülkeler için üretilen cihazların bazı kanallarda uyumsuzluk yaşanması olasıdır. Diğer standartlara göre güç tüketimi oldukça yüksektir

Kablosuz iletişimde radyo sinyalleri kullanıldığı için, kullanıcı ile kablosuz erişim noktası arasındaki iletişim dinlenebilmektedir. Bunu engellemek için WEP, WPA ve WPA2 gibi şifreleme yöntemleri kullanılsa da, bu şifreleme yöntemleri halen yeterince güvenli görülmemektedir. Yeterli sayıda şifrelenmiş paket toplandıktan sonra bir çok şifreli kablosuz iletişim çözülebilmektedir. Bu nedenle daha üst katmanlarda çalışan şifreli iletişim yöntemleri kullanılmaktadır.

Bu projede robotun üzerinde bulunan ana kontrolör ile ana bilgisayarın iletişiminde ve kamera görüntülerinin ana bilgisayara aktarılmasında WEP şifrelemesi ile 802.11B ve G standartları kullanılmaktadır. Kullanılan donanım ile ilgili detaylar ilerleyen bölümlerde irdelenecektir.

6.1.3 RS232 Standardı

1969'da tanımlanan RS232 standardı [40] DTE ve DCE cihazlar arasında binary³⁴ seri iletişim için kullanılan bir standarttır. Bu standartta gönderilecek olan veri bitleri ardı ardına iletişim kanalına aktarılır. RS232 standardı hem senkron hem de asenkron haberleşmeyi desteklemektedir. Gönderim ve alım için farklı fiziksel devreler kullanıldığı için full-dupleks³⁵ iletişim sağlanabilmektedir. RS 232 standardı ± 5 V, ± 10 V, ± 12 V, ve ± 15 V voltaj seviyelerinde çalışabilmektedir.

40 yılı aşkın süredir bilgisayarlar ve gömülü sistemlerin haberleşmesinde kullanılan bu standart günümüzde yerini USB ve Bluetooth gibi standartlara bırakmaktadır. Ancak sağlamış olduğu iletişim kolaylığı, uygulama maliyeti ve yaygın kullanımı nedeniyle günümüzde birçok gömülü sistemde halen RS232 standardı

³⁴ İkili sistem

³⁵ Çift taraflı, eş zamanlı

kullanılmaktadır. PIC mikro işlemcilerin donanıma dahil edilmiş RS232 desteği bu standardın projede kullanımının ana sebeplerinden biridir. Günümüzde RS232 iletişiminin kablosuz olarak kullanımını sağlayan cihazlar geliştirilmiş olması RS232 standardının popülerliğini devam ettirmektedir. Robotun mobil olma özelliğini sağlanabilmesi için kablosuz bir RS232 cihazı sisteme entegre edilmiştir. Bu cihaz ile ilgili detaylar haberleşme sistemleri bölümünde aktarılacaktır.

6.1.4 I2C Veri Yolu:

Gömülü sistemlerin günümüzde yaygın kullanımı bu sistemler arasında hızlı ve maliyeti düşük bir veri yolu ihtiyacını doğurmuştur. Özellikle sınırlı giriş çıkış arabirimine sahip mikro işlemciler için geliştirilen I2C veri yolu 2 kablo kullanır. Bunlardan birincisi SDA (Serial Data line) ikincisi ise SCL (Serial Clock line) adını alır. SCL iki uç arasındaki senkronizasyonu sağlarken SDA veri iletimi için kullanılır. Bu veriyolunun çalışabilmesi için iki ucunda ortak toprak hattını kullanması gerekmektedir.

I2C veri yolunun kullanımını yaygınlaştıran nedenlerden bir diğeri ise aynı veri yolu üzerinde 128 cihazın haberleşebilmesidir. Her bir cihaz barındırdığı adres üzerinden erişilebilmektedir. Cihazların adresleri 8 bitten oluşmaktadır. İlk 7 bit cihazın adresini belirlerken 8. bit cihaza okuma veya yazma için erişildiğini belirler

İletişim protokolünün işleyişi aşağıdaki şekilde özetlenebilir.

- Ana işlemci START komutunu gönderir
- Start komutunu alan diğer işlemciler dinleme moduna geçerler.
- Ana işlemci haberleşmek istediği cihazın adres bilgisini gönderir. Bu bilgi gönderilirken yazma veya okuma için olduğu belirtilir
- Veri yoluna bağlı olan tüm cihazlar gönderilen adresi kendi adresleri ile karşılaştırırlar. Eğer adres kendi adresleri ise ACK mesajı ile cevap verirler. Diğer durumda STOP sinyali gelinceye kadar hiçbir iletişim yapmazlar
- ACK mesajını alan ana işlemci verileri göndermeye başlar
- Veri gönderimi tamamlandığında STOP sinyalini göndererek veri yolunun diğer işlemciler tarafından kullanılabilmesini sağlar.

Günümüzde I2C veri yolunu kullanan hafıza modülleri, algılayıcılar, görüntü arabirimleri, kontrolörler, haberleşme modülleri gibi birçok donanım alternatifi bulunmaktadır. Bu projede çeşitli algılayıcılar ve kontrol alt sistemlerinin haberleşmesinde I2C veri yolu kullanılmıştır. Her bir cihazın detayları ilgili bölümde incelenecektir.

6.2 Donanım Yapısı

Robotun donanım yapısı kontrol kartları, algılayıcılar ve elektro-mekanik sistemlerden oluşmaktadır. Bu sistemler aralarında bir önceki bölümde değinilen 802.11B ve G, I2C, RS232 standartlarının yanı sıra analog sinyaller üzerinden de haberleşmektedirler. Robotun donanım yapısı 5 ayrı alt sistemden oluşur. Bunlar kontrol sistemi, hareket sistemi, haberleşme sistemi, algılayıcılar ve görüntü-ses sistemidir.

Kontrol sistemi 2 alt sistemden oluşmaktadır. Ana kontrol kartı tüm donanım sistemlerinin kontrolünü sağlarken ana bilgisayar tüm yazılım işlemlerinin kontrolünü yapmaktadır. Hareket sistemi ise ana bilgisayardan ana kontrolöre gelen hareket komutlarının yerine getirilmesinden sorumludur. Motor kontrol kartlarından ve motorlardan oluşmaktadır.

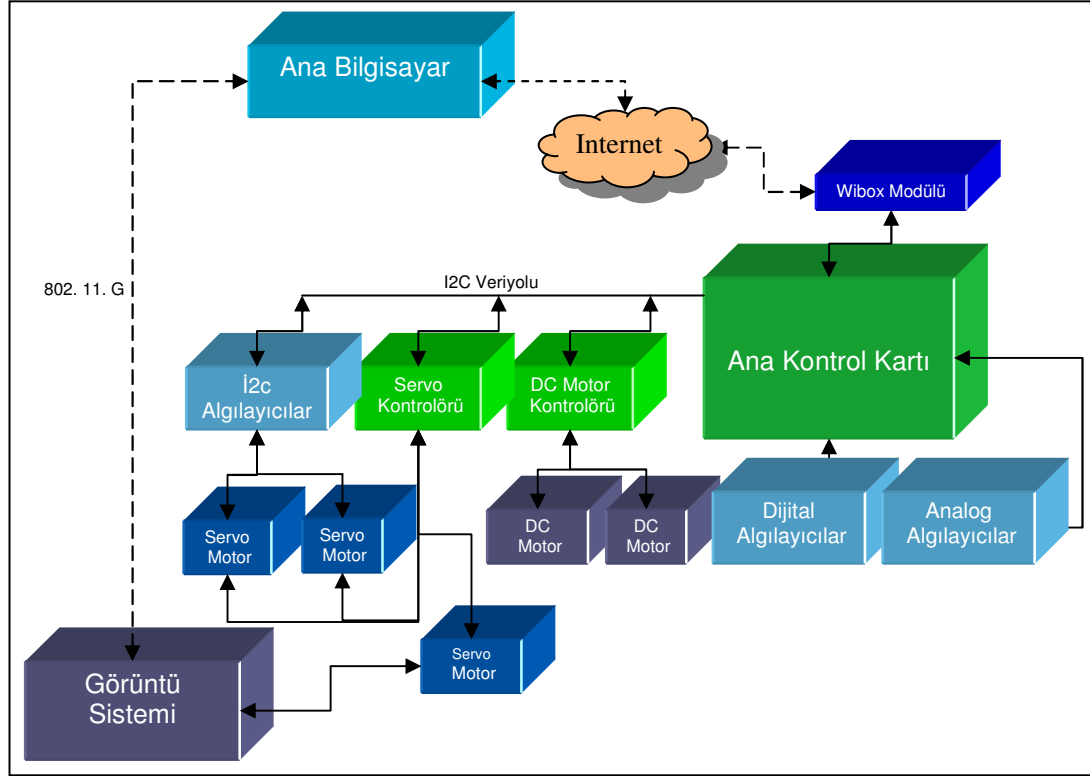
Tüm alt sistemlerin birbiri arasında haberleşmesi ise haberleşme sistemi tarafından gerçekleştirilir. Haberleşme sisteminin en önemli elemanı ana bilgisayar ile ana kontrolörün haberleşmesini sağlayan wibox modülüdür. Bu modül Ana bilgisayar ile IEEE 802.11.b standardı ile haberleşirken ana kontrolör ile RS232 standardını kullanarak haberleşir. Haberleşme alt sisteminin diğer önemli bölümü ise ana kontrol kartının motor kontrol kartları, konuşma sentezleyicisi ve algılayıcılarla haberleşmesini sağlayan I2C arabirimidir.

Robotun çevresi ile etkileşimini sağlayan algılayıcılar ise 4. alt sistemi oluşturur. Bu algılayıcılar dijital ve analog sinyaller ile ana kontrol kartı ile haberleşerek çevre ile ilgili verilerin elde edilmesini sağlar. Uzaklık, ivme, ışık şiddeti, yönelim, batarya durumu, hareket algılama gibi veriler okunarak robotun karar verme algoritmalarında kullanılırlar.

5. alt sistem ise ses ve görüntü alt sistemidir. Bu sistem 2 önemli donanımdan oluşmaktadır. Robotun üzerinde bulunan kablosuz IP kamera ile operatöre ve ana

bilgisayara görüntü geri beslemesi yapılırken; konuşma sentezleyici ile uygulanan komutlar ve algılayıcı değerleri ile ilgili sesli geri bildirim sağlanır.

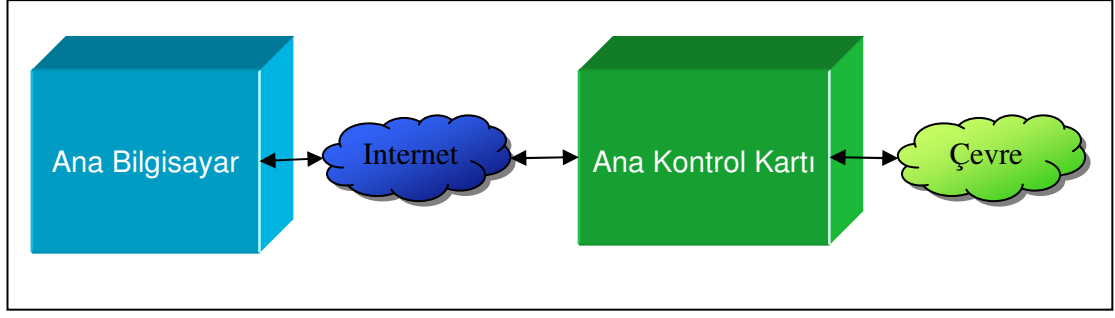
Tüm alt sistemlerin ve donanımların birbiri ile olan ilişkisi Şekil 6.1’de detaylı olarak gösterilmiştir.



Şekil 6.1 : Genel Donanım Yapısı

6.3 Kontrol Sistemi

Kontrol sistemi robotun tüm donanım işlevlerinin yönetilmesi ve yazılım bölümünün verdiği karar ve komutların yerine getirilmesi görevini üstlenir. Ana bilgisayar ve ana kontrol kartından oluşur. Ana bilgisayar üzerinde çalışan yazılımın verdiği tüm komutlar haberleşme sistemi ile ana kontrolöre aktarılmakta ve ana kontrol kartı üzerinde çalışan yazılım tarafından işlenerek fiziksel etkilere dönüştürülmektedir. Kontrol sisteminin yapısı Şekil 6.2’de gösterilmiştir.



Şekil 6.2 : Kontrol Sisteminin Yapısı

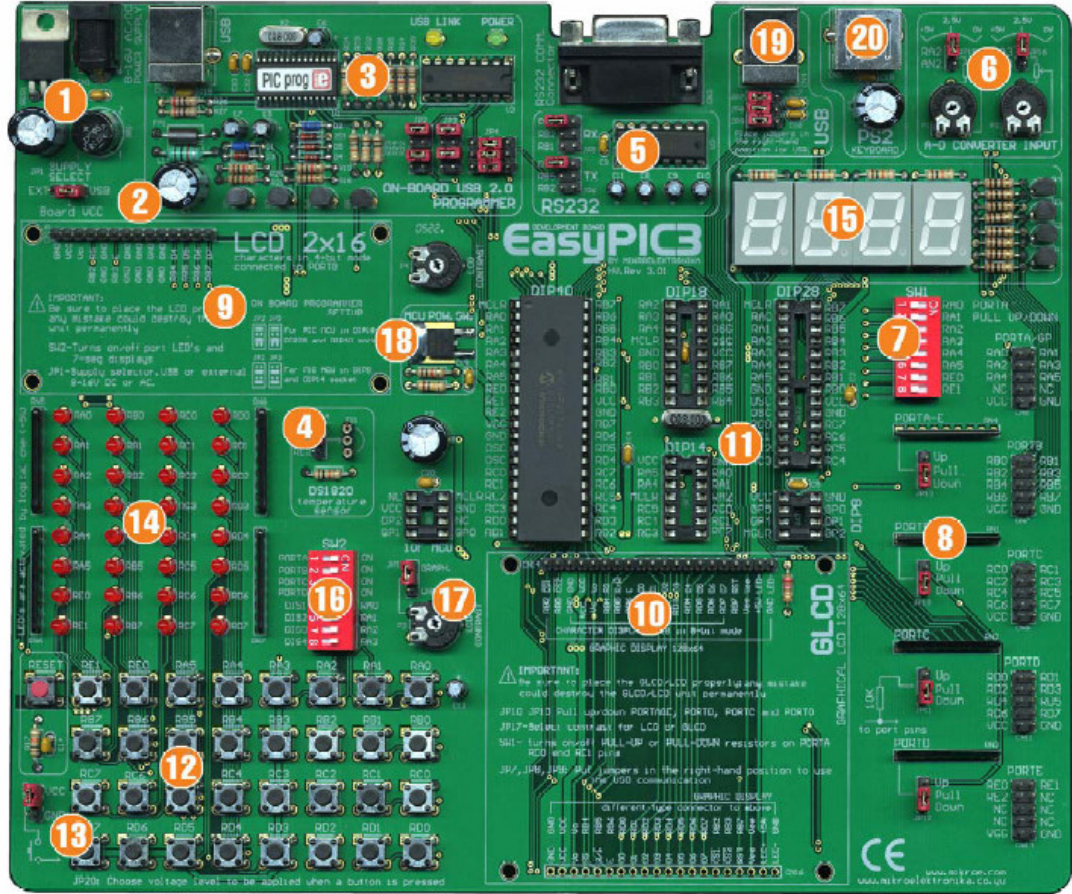
6.3.1 Ana Kontrol Kartı:

Ana bilgisayardan gelen tüm komutların işlenmesi ve fiziksel etkilere dönüştürülmesi, alt donanım sistemlerinin kontrolü ana kontrol kartı tarafından gerçekleştirilmektedir. Sistemde ana kontrol kartı olarak Mikroelektronika firmasının geliştirmiş olduğu Easypic3 geliştirme kartı kullanılmıştır. Bu kart 8, 14, 18, 28 ve 40 pinli PIC mikro işlemcileri desteklemektedir.

Ana kontrol kartı, 7-15 volt arası doğru akım veya alternatif akım harici güç girişi (Şekil 6.3 'te 1 ile gösterilmiştir), USB veya harici güç girişi seçim atacı³⁶ (Şekil 6.3'te 2), USB arabirimini kullanan programlayıcı (3), DS1820 sayısal termometre yuvası (4), RX ve TX pinleri ataçlarla seçilebilen RS232 arabirimi (5), analog sayısal dönüştürücünün denenmesi için kullanılabilen ve ataçlarla ayarlanan 2 potansiyometre (6), PortA pinleri direnç ağına bağlamak için mikro anahtar (7), Her bir port için ataçlarla pull-up veya pull-down olarak ayarlanabilen çoklu dirençler (8), 4bit bağlantı için karakter LCD konektörü (9), 8-bit bağlantı ile karakter LCD veya grafik LCD bağlanabilen konektörler (10), 8, 14, 18, 28 ve 40 pin mikro işlemciler için yuvalar ve harici osilatör yuvası (11), Mikro işlemcinin her bir pinine bağlı 32 adet düğme (12), düğmelere basıldığında 5 volt veya topraklama yapılabilmesini sağlayan ataç (13), tüm port pinlerinin durumlarının görülebildiği 32 adet LED (14), çoklanmış 4 rakamlı 7-bölmeli LED (15), her bir porta bağlı ledlerin açılıp kapatılmasını sağlayan mikro anahtar (16), LCD karışıklığını ayarlamak için potansiyometre (17), Mikroişlemci güç regülatörü (18), USB cihazları kontrol etmek için USB arabirimi (19), Klavye bağlantısı için PS2 arabirimi (20) ve her bir port pininin harici cihazlara bağlanmasını sağlayan 10'lu bağlantı arayüzü barındırmaktadır.

³⁶ İng: Jumper

Bu projede ana işlemci olarak microchip firmasının 40 pinli PIC 16F877A [42] mikro işlemcisi kullanılmıştır. Ayrıca görsel geri bildirim amaçlı olarak 4 bit arabirim kullanan 2 satır 16 sütun karakter gösterebilen LCD ekran kullanılmıştır.



Şekil 6.3 : Easypic3 Kontrol Kartı

6.3.2 Mikro işlemci

PIC16F877A işlemcisi [42] CMOS-Flash tabanlı 200 nanosaniye komut işlem hızında ve toplamda 35 farklı komut barındıran bir işlemcidir. 14 Kilobyte program hafızası, 368 Byte RAM ve 256 Byte eeprom barındırmaktadır. 40 pinden oluşan işlemci 33 I/O (giriş/çıkış) pinine sahiptir. B, C ve D portları 8 er pine sahipken PortA ve PortE birlikte tasarlanmış olup toplamda 9 pin barındırırlar. Mikro işlemci aynı zamanda 8 kanal (10 bit) analog sayısal dönüştürücü, 2 comparator³⁷, ICD³⁸, 2 kanal 10 bit capture³⁹ / compare⁴⁰ / PWM⁴¹ giriş çıkışı, 3 kablo ile SPI⁴² veya 2 kablo

³⁷ Karşılaştırıcı

³⁸ In Circuit Debugger: Devre içi hata ayıklayıcı

³⁹ Yakalama

⁴⁰ Karşılaştırma

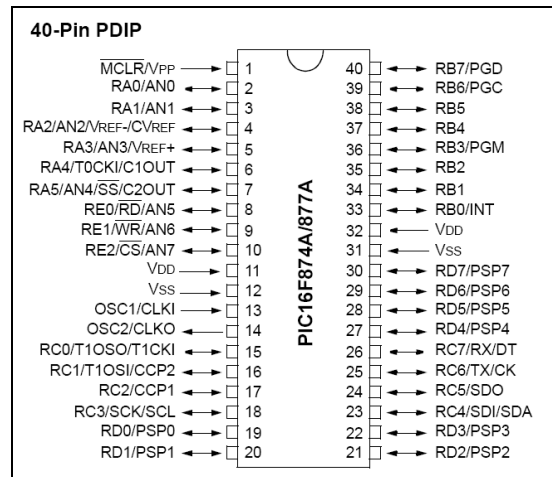
⁴¹ Pulse width modulation: Darbe genişliği modülasyonu

⁴² Serial Peripheral Interface: Seri birim arayüzü

ile I2C⁴³ arabirimi olarak kullanılabilen senkron seri port ve USART⁴⁴ arabirimlerine sahiptir. Ayrıca 15 kesme kaynağı ve 3 zamanlayıcı bulunmaktadır. Bu özelliklerin tamamı 33 giriş-çıkış pini ile sağlanmaktadır. Her bir pin birden fazla özelliği barındırabilmektedir. Bir pin'in hangi işlevi yerine getireceği programlama aşamasında konfigürasyon belleklerinin değerlerinin değiştirilmesi ile belirlenmektedir. Bir pin'in alternatif fonksiyonu kullanıldığında o pin genel amaçlı giriş çıkış pini olarak kullanılamamaktadır. Örneğin A portunun pinleri analog sayısal dönüştürücü için giriş olarak kullanıldığında sayısal giriş-çıkış için kullanılamamaktadır. Mikro işlemcinin pinlerinin işlevleri Şekil 6.4'te, blok diyagramı ise Şekil 6.5'te görülmektedir. Tüm pinlerin detaylı açıklamaları Ek-C'de detaylı olarak listelenmiştir.

Bu proje'de 20 Mhz hızında bir işlemci osilatörü kullanılmıştır. İstenildiğinde bu hız osilatör değiştirilerek 4 Mhz'ye kadar düşürülebilmektedir.

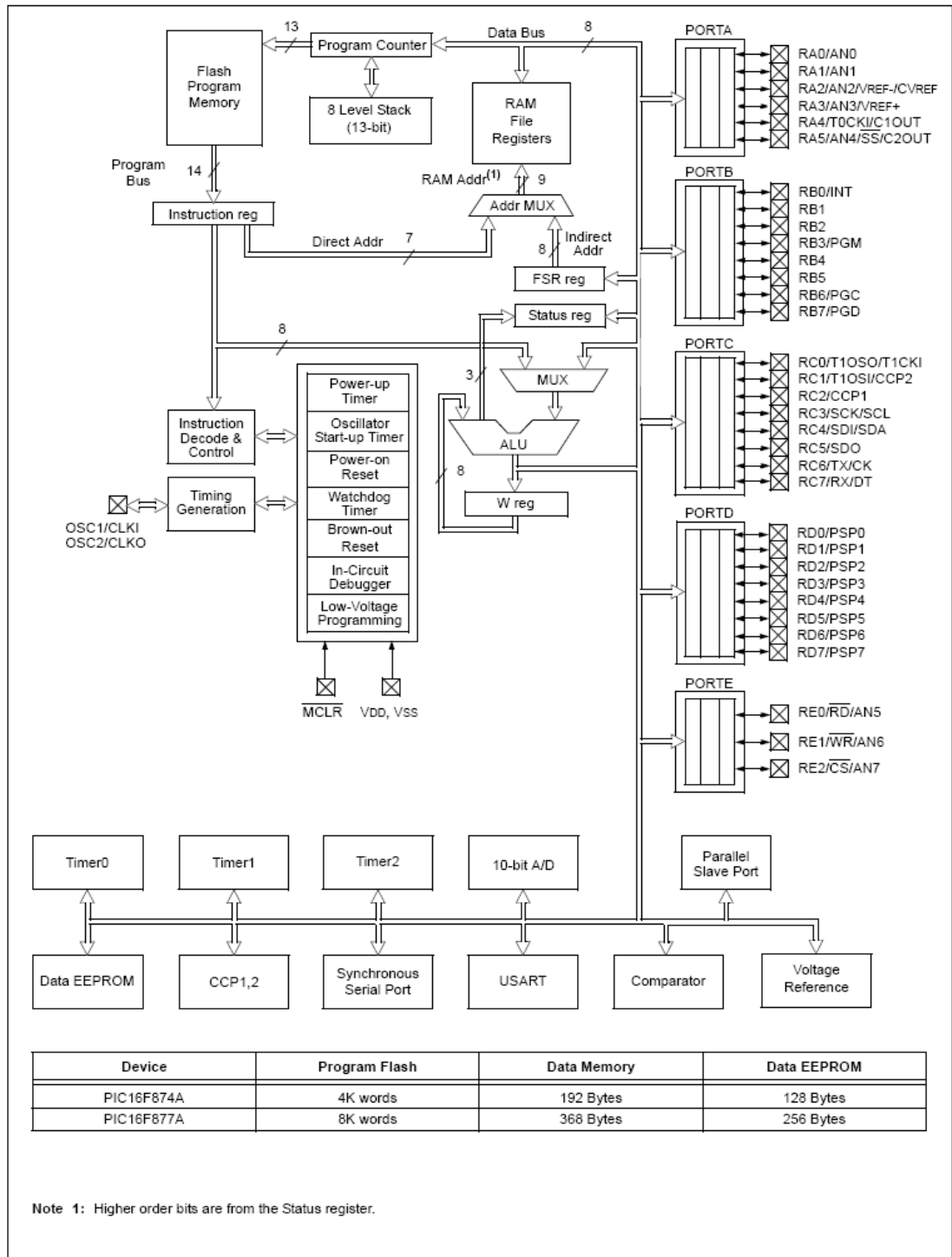
Ana bilgisayarla haberleşme için RS232 arabirimi (Şekil 6.3'te 5) kullanılarak wiport modülüne bağlantı sağlanmıştır. PortA ve PortE üzerinde bulunan analog sayısal çeviriciler ise uzaklık, ivme ve hareket algılayıcıların değerlerini okumak için kullanılmaktadır. I2C pinleri kullanılarak dijital algılayıcılar ve motor kontrol kartları kumanda edilmektedir. PortB üzerinde 4 bit arabirim kullanılarak karakter LCD sürülmektedir. Mikro işlemci üzerinde bulunan ve donanım kontrolünü sağlayan mikroişlemci kontrol yazılımı bölüm 7 de detaylı olarak incelenecektir.



Şekil 6.4: PIC 16F877A Pin İşlevleri

⁴³ Inter-Integrated Circuit: Entegre devreler arası arayüz

⁴⁴ Universal Asynchronous Receiver Transmitter:Evrensel asenkron alıcı verici

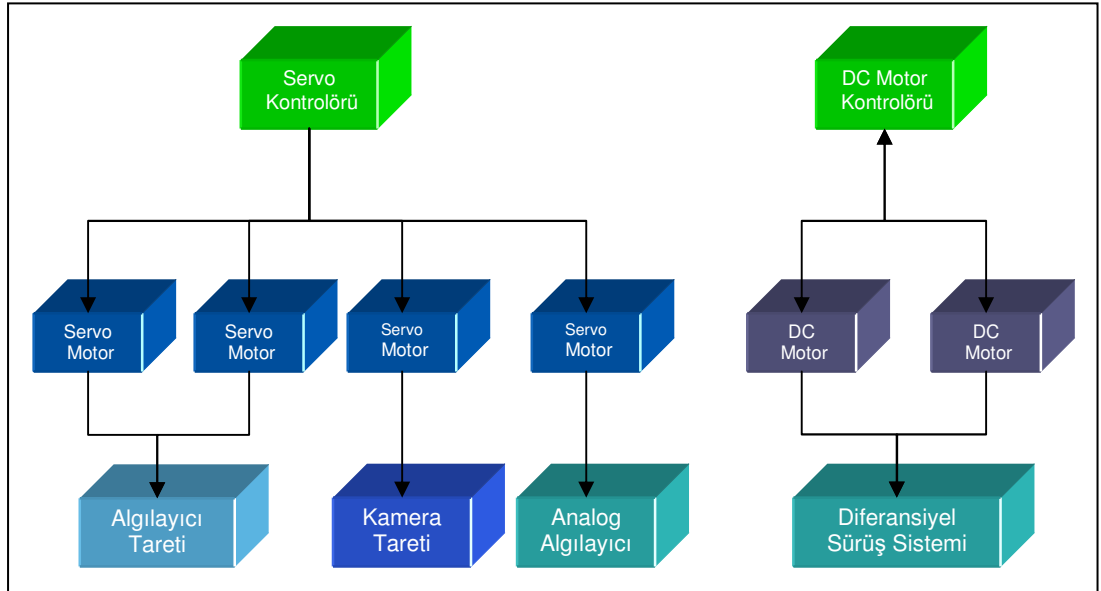


Şekil 6.5: PIC 16F877A Blok Diyagramı

6.4 Hareket Sistemi

Hareket sistemi robotun, algılayıcıların ve kameranın hareketinden sorumlu sistemdir. Kontrol kartları, motorlar ve mekanik sistemlerden oluşmaktadır. Bu sistemi robotun hareketini sağlayan ve algılayıcıların hareketini sağlayan sistemler olarak iki ana başlık altında incelemek mümkündür.

Robotun hareketi dc motor kontrol kartı, motorlar ve diferansiyel sürüş sistemi tarafından sağlanmaktadır. Algılayıcılar ve kameranın hareket kontrolünde ise servo motor kontrol kartı, servo motorlar, algılayıcı taretleri ve kamera taretleri görev almaktadır. Hareket sistemi ve alt sistemlerin birbiri ile olan ilişkisi Şekil 6.6'da gösterilmiştir. Tüm bu donanımlar ilgili bölümlerde detaylı olarak ele alınacaktır.

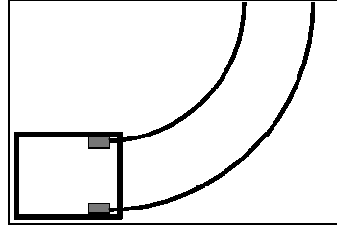


Şekil 6.6 : Hareket Sisteminin Yapısı

6.4.1 Diferansiyel Sürüş Sistemi

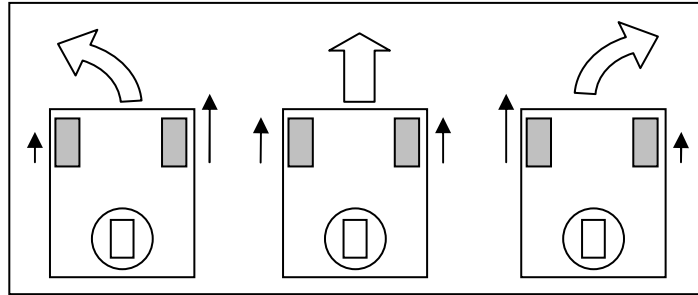
Diferansiyel sürüş ismi (mekanik diferansiyeller içinde geçerli olan) hareket vektörünün her bir tekerin hareket vektörlerinin toplamı olması durumundan gelmektedir. Ancak çoğu robot sisteminde mekanik bir diferansiyel kullanılmamaktadır. Robotlarda kullanılan diferansiyel sürüş sistemi genellikle ayrı ayrı tahrik edilen sol ve sağ tekerlerden oluşur. Ancak tanklarda olduğu gibi paletli tahrik mekanizmaları da kullanılabilir. En yaygın kullanılan teker dizilimi sağ ve solda birer tahrikli teker, arka ortada ise 3 serbestlik dereceli kastor teker şeklindedir.

Aralıklı iki teker barındıran bir taşıt bir köşeyi dönmeye çalıştığında dıştaki tekerin daha fazla yol kat ettiği bilinmektedir. Bu otomobillerin diferansiyel dişlilerine sahip olmasının nedenidir.



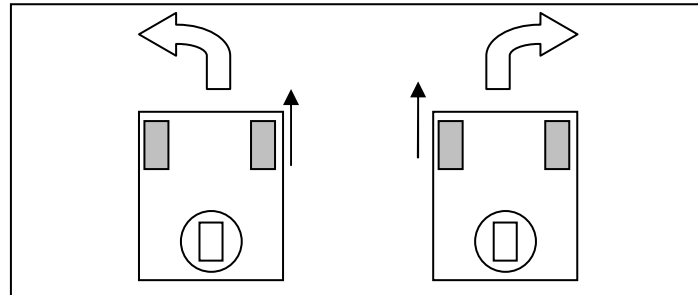
Şekil 6.7 : Diferansiyel Sürüş

Bir dönüş esnasında dışta bulunan tekerin içteki tekerden daha fazla yol kat etmesi dolayısıyla daha hızlı dönme gereksinimi diferansiyel sürüşün çıkış noktasıdır (Şekil 6.7). Bu prensipten yararlanarak tekerlerden birini diğerinden daha hızlı döndürdüğümüzde bu teker daha fazla yol kat edecek ve robot, yavaş olan tekerin bulunduğu tarafa doğru dönecektir (Şekil 6.8).



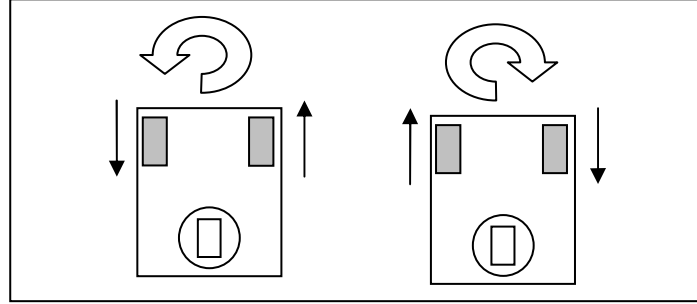
Şekil 6.8 : Teker Hızlarının Hareket Yönüne Etkisi

Bu dönüşün çapı her bir tekerin hızına bağlıdır. Dönüş yönüne göre içte kalan tekerin hızı arttıkça dönüş çapı da artmaktadır. Keskin bir dönüş yapılmak istendiğinde ise Şekil 6.9’da görüldüğü gibi iç taraftaki tekerin hızı sıfırlanıp dış teker maksimum hızda döndürülerek istenen dönüş sağlanabilir.



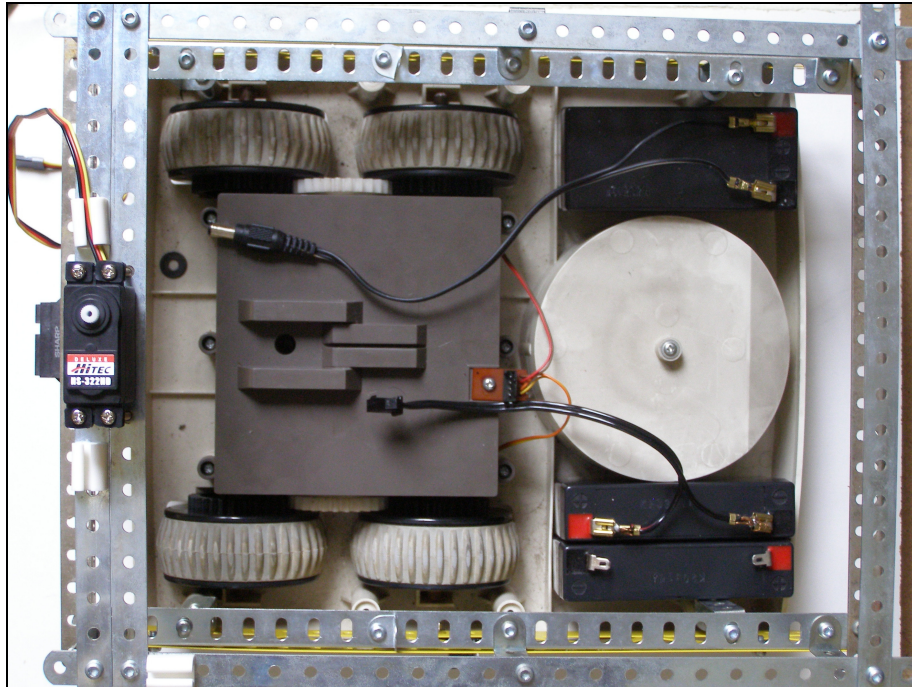
Şekil 6.9 : Diferansiyel Sürüşte Keskin Dönüş

Diferansiyel sürüş sisteminin bir diğer avantajı ise iç ve dış tekerlerin zıt yönde döndürülerek (Şekil 6.10) robotun bulunduğu konumu değiştirmeden yönünü değiştirmesinin sağlanabilmesidir. Bu manevra kolaylığı ızgara tabanlı bir haritada hareket ederken büyük bir esneklik sağlamaktadır.

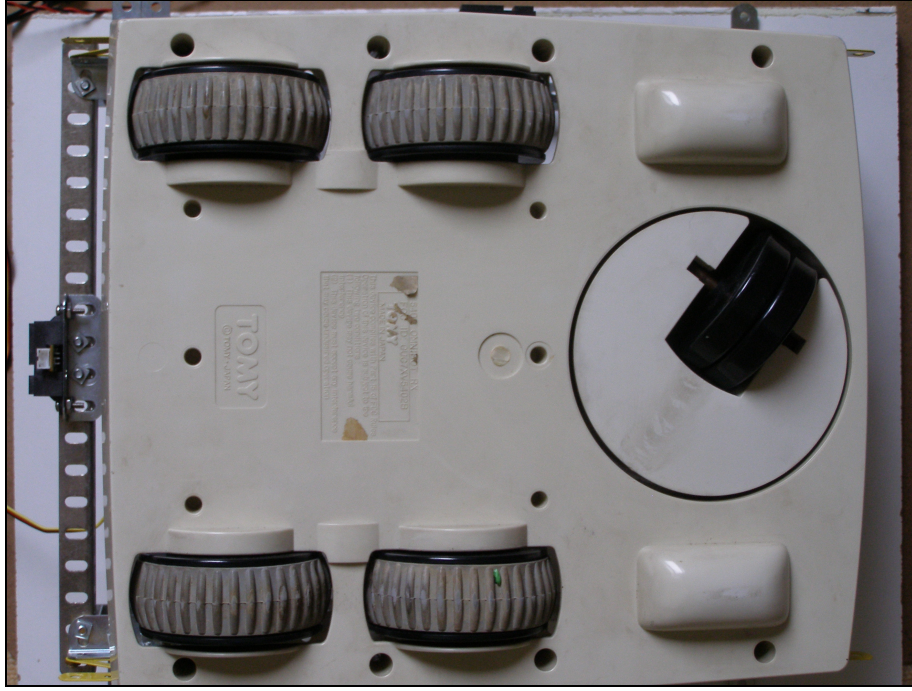


Şekil 6.10 : Diferansiyel Sürüşte Sabit Dönüş

Bu projede sol ve sağda ikişer teker barındıran bir diferansiyel sürüş sistemi kullanılmıştır. 73mm çapındaki bu tekerler yol tutuşunun artırılması için kauçuk bantlarla kaplıdır. Tekerler Şekil 6.11’de görülen dişli kutusundan çıkan 27 diş barındıran birer evolvent dişli çark ile tahrik edilmektedir. Sürüş esnasında ve dönüşlerde dengenin sağlanabilmesi için arka ortada bir adet kastor tekerlek kullanılmıştır (Şekil 6.12).



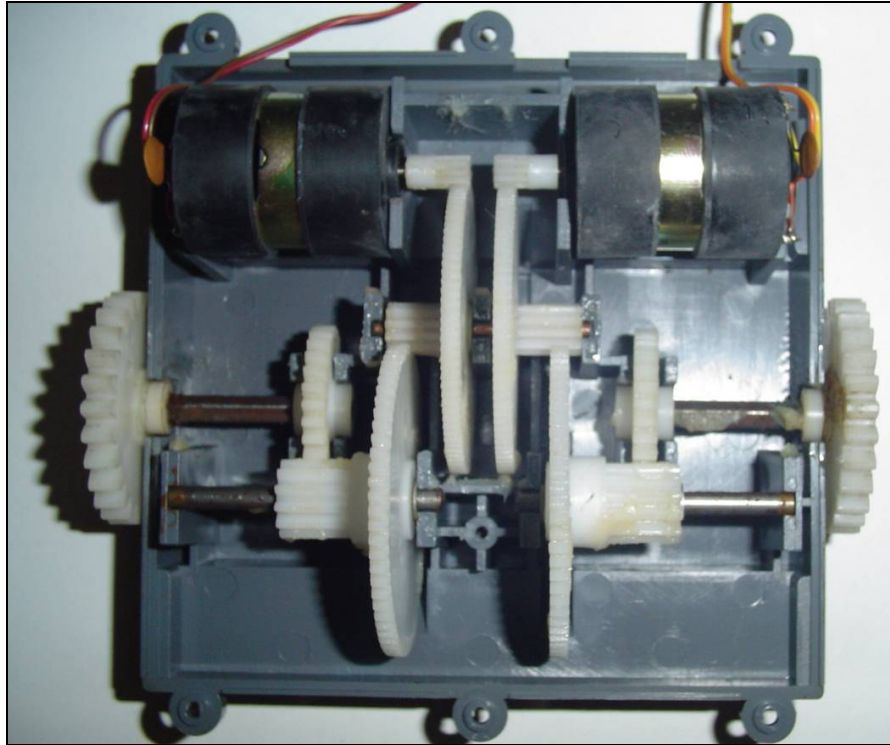
Şekil 6.11 : Diferansiyel Sürüş Mekanizması Üstten Görünüş



Şekil 6.12 : Diferansiyel Sürüş Mekanizması Alttan Görünüş

6.4.2 DC Motorlar

Robotun hareketini sağlamak için 2 adet 12 voltluk dc motor kullanılmaktadır. Motorlardan biri diferansiyel sürüş sistemindeki sol tekerleri kontrol ederken diğeri sağ tekerleri kontrol etmektedir.



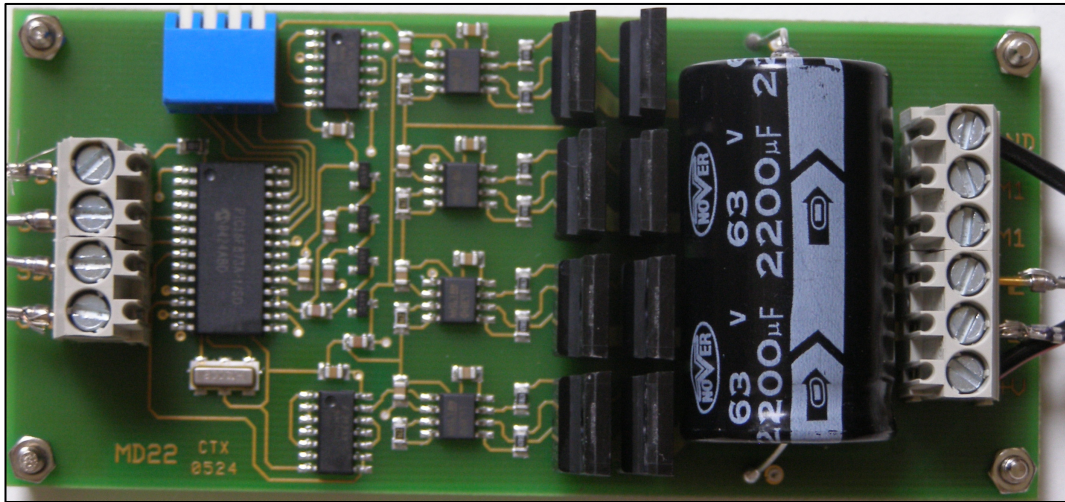
Şekil 6.13 : Dc Motorlar ve Redüktör Dişli Kutusu

Motor sarımlarında oluşan indüksiyon akımının kontrol kartını etkilememesi için her bir motorun bacakları kondansatörler ile birbirine bağlanmıştır. Böylece oluşabilecek geri beslemenin önüne geçilmiştir. Yüksek devirde dönen motorların robotun ağırlığını hareket ettirecek bir tork sağlaması mümkün olmadığından motorlar bir redüksiyon dişli kutusuna bağlanmış; çıkış devir sayısı düşürülerek elde edilen tork artırılmıştır. Dc motorlar ve redüktörün detaylı görünümü Şekil 6.13 'de verilmiştir.

6.4.3 DC Motor Kontrol Kartı

Robotun hareketinin sağlanması için kullanılan dc motorların kontrolü Şekil 6.14'te görülen Devantech firmasının “MD22 - 50V 5A Dual H-Bridge Motor Driver” [43] ürünü kullanılmıştır. Kartın görevi sağ ve sol dc motorları kontrol ederek robotun hareketini sağlamaktır.

Bu kontrol kartı 2 motoru aynı anda sürebilmektedir. Çalışma için 5 volt besleme voltajına ihtiyaç duyan MD22 kartı 5-50V aralığında her bir kanal için 5 ampere kadar akım sağlayabilmektedir.



Şekil 6.14: Motor Kontrol Kartı

Devrenin çalışması ve motorların beslenmesi için gerekli olan akım ayrı konektörler aracılığıyla sağlanmaktadır (Şekil 6.14'de soldaki bağlantı konektörleri). PIC16F873A işlemci barındıran kontrol kartı, 5 farklı kontrol modunda çalıştırılabilmektedir. Bu modların seçimi kart üzerinde bulunan mikro anahtar yardımıyla yapılmaktadır. Bu modlar analog mod, diferansiyel sürüslü analog mod, RC servo modu, diferansiyel sürüslü RC servo modu ve I2C modudur.

Analog modda her bir motor kontrol arabiriminin SDA (motor1) ve SCL (motor2) girişlerine uygulanan 0-5 volt arasındaki akımla kontrol edilir. 0 volt maksimum hızla geri, 5 volt ise maksimum hızla ileri komutuna karşılık gelmektedir. 2. 5 volt ise motorun durmasını sağlamaktadır. Durma konumunun stabilizasyonu için %2. 7lik bir güvenlik bandı kullanılmıştır. Bu mod durma anında enerji tükettiği için bu proje için uygun değildir.

Diferansiyel sürürlü analog modda ise iki motorun hızı da SCL girişine uygulanan voltaj ile kontrol edilir. SDA girişine uygulanan voltaj ise dönüş yönünü kontrol etmekte kullanılır. 0 volt tamamen sola dönüş, 5volt tamamen sağa dönüş, 2, 5volt ise düz gidiş anlamına gelmektedir. SDA kontrol voltajı 0-5 volt aralığında değiştirilerek robotun yönü ayarlanabilmektedir.

RC servo kontrol modu ise radyo kontrollü uzaktan kumanda alıcılarının md22'ye bağlanabilmesi için kullanılan bir moddur. Her bir rc kanalı bir motorun hareketini kontrol etmektedir. Bu modda kontrol kartı 1ms ile 2ms aralığında kontrol sinyalleri üreterek servo motor kontrolü yapabilmektedir. Diferansiyel sürürlü rc servo modunda ise kanallardan biri ileri geri hareketi kontrol ederken diğer kanal dönüş işlemini kontrol eder.

I2C modunda ise kontrol kartı I2C veri yolundan gelen komutları kullanarak motor kontrolünü gerçekleştirir. Kontrol kartının I2C modunda erişilebilen 8 bit genişliğinde 8 ayrı yazmaçı bulunmaktadır. Bu yazmaçların işlevleri Tablo 6.1'de açıklanmıştır.

Tablo 6.1 : Motor Kontrol Kartı Yazmaç Adresleri

Yazmaç Adresi	İsim	Okuma (R) Yazma (W)	İşlevi
0x00	Mod	R/W	Çalışma modu
0x01	Hız	R/W	Mod 0 ve 1 de sol motor hızı, mod 2ve 3 te ise hız bilgisi
0x02	Hız /dönüş	R/W	Mod 0 ve 1 de sağ motor hızı, mod 2ve 3 te ise dönüş bilgisi
0x03	ivmelenme	R/W	Mod 0 ve 1 de ivmelenme bilgisi
0x04	Kullanım dışı	Salt okunur	Sıfır olarak okunur
0x05	Kullanım dışı	Salt okunur	Sıfır olarak okunur
0x06	Kullanım dışı	Salt okunur	Sıfır olarak okunur
0x07	Yazılım versiyonu	Salt okunur	Yazılım revizyon numarası

I2C modunda kullanılan protokol popüler 24c04 serisi gibi eepromlarda kullanılan protokol ile aynıdır. Herhangi bir yazmaçın değerini okumak için ilk olarak okuma yazma biti 0 olarak kontrol kartının I2C adresi (0xb0) ana kontrolör tarafından gönderilir. Ardından okunmak istenen yazmaçın adresi gönderilir. Bu işlemden sonra I2C haberleşmesi yeniden başlatılarak kontrol kartının I2C adresi okuma yazma biti 1 olarak (0XB1) gönderilir. Bu komut kontrol kartı tarafından alındıktan sonra kontrol kartı ilgili yazmaçın değerini ana kontrol kartına gönderir. Eğer ana kontrol kartı ACK sinyali gönderirse bir sonraki yazmaçın değeri I2C veri yoluna aktarılır. Böyle ardışık yazmaçların tek bir işlem dizisi ile okunması sağlanmış olur. Bir yazmaçın değeri değiştirilmek istenildiğinde ise okuma yazma biti 0 olarak kontrol kartının I2C adresi (0xb0) gönderilir. Bunu takiben değiştirilecek yazmaçın adresi ve istenilen değer gönderilerek işlem tamamlanır.

Motor kontrol kartı I2C ile kumanda edilirken 6 ayrı mod kullanılabilir. Bu modların seçimi 0x00 adresindeki mod yazmaçının değeri değiştirilerek yapılmaktadır.

Mod 0 ve 1’de 0x01 ve 0x02 yazmaçları sağ ve sol motor için ayrı ayrı hız bilgisini tutmaktadır. Mod 0 ‘da 0 değeri “tam yol geri”, 128 değeri “dur”, 255 değeri ise “tam yol ileri” anlamını taşımaktadır. Mod 1 ise -128 “tam yol geri”, 0 değeri “dur”, +127 “tam yol ileri” anlamındadır.

Mod 2 ve 3 te 0x01 yazmaç hız, 0x02 yazmaç ise dönüş yönünü belirlemek için kullanılır. Mod2’de kullanılan hız değerleri mod 0 da olduğu gibi 0-255 arasındadır. Mod 3 ise -128 +127 aralığını kullanır.

$$Dönüş_faktörü = dönüş_register * \frac{Hız_registeri}{Maksimum_Hız} \quad (6.1)$$

$$\begin{aligned} Motor1_hızı &= Hız_registeri + DönüşFaktörü \\ Motor2_hızı &= Hız_registeri - DönüşFaktörü \end{aligned} \quad (6.2a)$$

$$\begin{aligned} Motor1_hızı &= Hız_registeri - DönüşFaktörü \\ Motor2_hızı &= Hız_registeri + DönüşFaktörü \end{aligned} \quad (6.2b)$$

Eğer ileri doğru gidiliyorsa motorların hızı (6. 2a) denklemi ile geri doğru gidiliyorsa (6. 2b) denklemi ile kontrol kartı tarafından hesaplanarak uygulanır. Eğer motorlardan biri hesaplanan hızı sağlayamayacak durumda ise (örn hesaplanan hız

255'in üzerinde) kontrol kartı diğer motorun hızını ayarlayarak istenen dönüş değerine ulaşılmasını sağlar.

$$\begin{aligned} Motor1_hızı &= Hız_registeri - Dönüş_registeri \\ Motor2_hızı &= Hız_registeri + Dönüş_registeri \end{aligned} \quad (6.3)$$

Mod 4 ve 5 ise mod 3, 4 te olduğu gibi işlev yürütür. Ancak motor hızlarının hesaplanmasında farklı bir yöntem izlenir. Mod 3 ve 4'ün aksine bir dönüş faktörü hesaplanmaz. Motor hızları hesaplanırken (6.3) denklemi kullanılır. Eğer motorlardan biri için hesaplanan hız değeri maksimum veya minimum hız değeri sınırlarını aşıyorsa diğer motorun hızı ayarlanarak istenilen dönüş kontrol kartı tarafından sağlanır.

Bu projede md22 kontrol kartı I2C veri yolu kullanılarak ana kontrol kartı ile haberleşmektedir. Kartın beslemesi easypic3 kartından gelen +5V ile yapılırken motorlar 12 volt 1.2mAh değerindeki akü üzerinde beslenmektedir. Motorlarda meydana gelebilecek bir sorunun diğer devrelere ve kontrol kartına zarar vermesini engellemek için 10 amperlik bir sigorta sisteme dahil edilmiştir.

MD22 motor kontrol kartı mod-0'da kullanılarak sağ ve sol motorların 0-255 arasında hız kontrolü yapılmaktadır. Böylece her bir motor birbirinden bağımsız olarak kontrol edilebilmektedir. Dönüş için gerekli işlemler ise ana kontrol kartı üzerinde bulunan mikro işlemci yazılımı tarafından kontrol edilmektedir. Yazılım detayları bölüm 7'de ele alınacaktır.

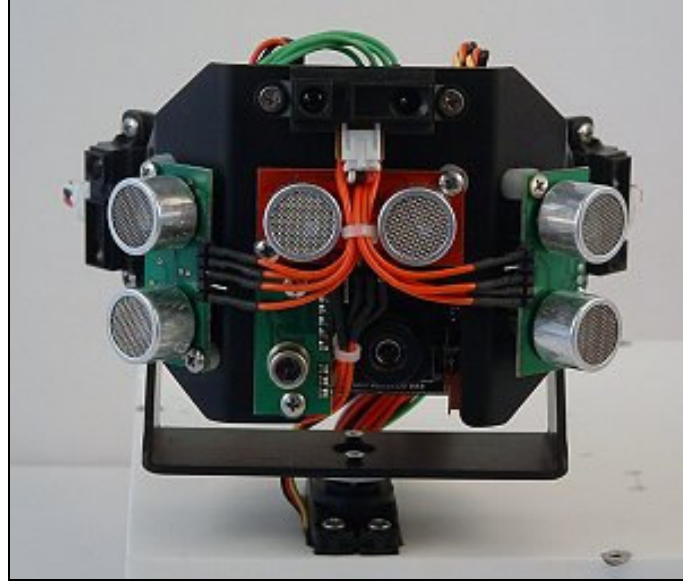
6.4.4 Algılayıcı ve Kamera Hareket Sistemi

Robotun barındırdığı algılayıcılardan bir bölümü çeşitli açılardan ölçüm yapılabilmesi amacıyla hareket edebilir şekilde tasarlanmıştır. Bunların başında kızılötesi ve sonar uzaklık algılayıcılar ve kablosuz kamera sistemi gelmektedir. Her bir sistem taretler ve servo motorlar ile hareket etmektedir. Bu servo motorların kontrolü ise Devantech sd21 servo motor kontrol kartı ile gerçekleştirilmektedir.

6.4.5 Algılayıcı Tareti

Sistemde 2 adet algılayıcı taretleri bulunmaktadır; bunlardan biri robotun ön bölümünde diğeri ise arka bölümde yer almaktadır. "CMUcam2+ Sensor shell Turret" algılayıcı taretleri, 3 kızılötesi algılayıcı, 3 sonar, 1 thermopile array, 1 cmucam2+ kamera olmak üzere toplamda 8 adet algılayıcı taşıyabilmektedir. Şekil 6.15 ve Şekil 6.16'da taret

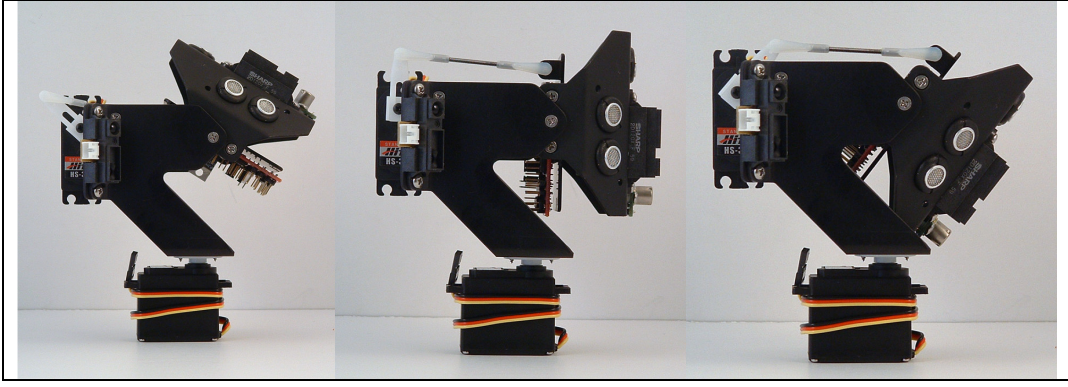
tüm bu algılayıcılar monte edilmiş olarak görülmektedir. Robotun ön tarafına yerleştirilen taret sonar mesafe algılayıcısı ve kızıl ötesi mesafe algılayıcısı barındırmaktadır. Yatayda 180, düşeyde ise 90 derecelik hareket serbestliğine sahip olan taretin düşey hareket aralığı Şekil 6.17’de gösterilmiştir.



Şekil 6.15 : Algılayıcı Tareti Önden Görünüş



Şekil 6.16 : Algılayıcı Tareti Yandan Görünüş



Şekil 6.17 : Algılayıcı Tareti Düşey Hareket Aralığı

Robotun arka bölümüne monte edilmiş diğer taret ise yatay da 180 derecelik bir tarama yapabilmektedir. Bu tarete sonar veya kızıl ötesi mesafe algılayıcılar monte edilebilmektedir. Şekil 6.18’de bu taret devantech srf10 ultrason mesafe algılayıcı monte edilmiş olarak görülmektedir.



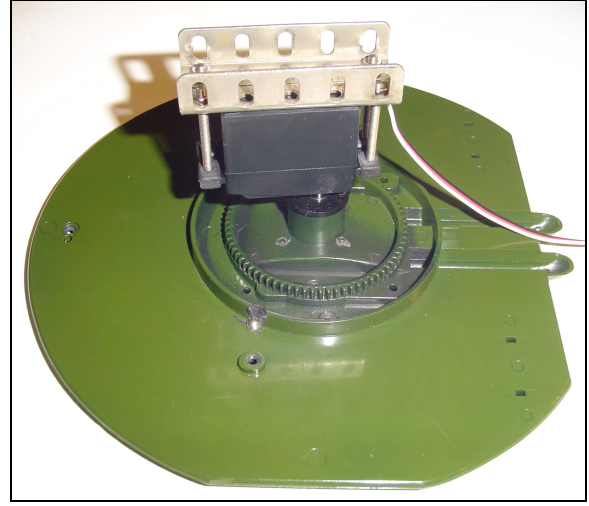
Şekil 6.18 : Arka Algılayıcı Tareti

6.4.6 Kamera Tareti

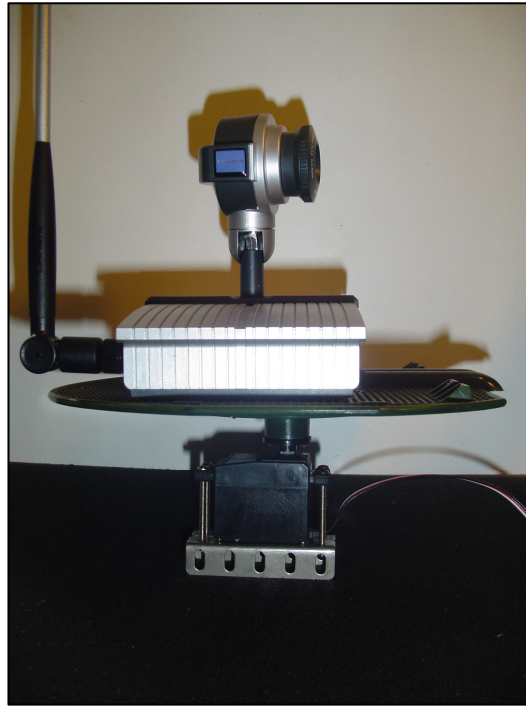
Robot üzerinde bulunan ve operatöre görsel geri bildirim amacıyla kullanılan kablosuz kamera taret yardımıyla 300 derece dönebilmektedir. Böylece operatör robotun çevresindeki tüm engelleri görme şansına sahip olmaktadır. Kamera tareti 200mm çapında bir platform ve sürekli dönüş için modifiye edilmiş bir servodan oluşmaktadır (Şekil 6.19, Şekil 6.20 ve Şekil 6.21).



Şekil 6.19 : Kamera Tareti Üstten Görünüş



Şekil 6.20 : Kamera Tareti Altan Görünüş



Şekil 6.21 : Kamera Tareti ve Kamera

6.4.7 Servo Motorlar

Servo motorlar barındırdıkları dc motorların torkunu redüktörler yardımıyla artırıp çıkış şaftı üzerinden dışarıya iletebilen özel motorlardır. Standart redüktörlü motorlardan farkı şaftın konumunun servonun hareket aralığındaki istenen bir konuma kodlanmış bir sinyal yardımıyla getirilebilmesidir. Bu özellik servo motorların endüstride, robotlarda ve uzaktan kumandalı uçak, araba vs gibi hobi araçlarının yönlendirme mekanizmalarının harekete geçirilmesinde çok yaygın olarak kullanılmasını sağlamaktadır. Kodlu sinyal servonun giriş ucunda bulunduğu sürece

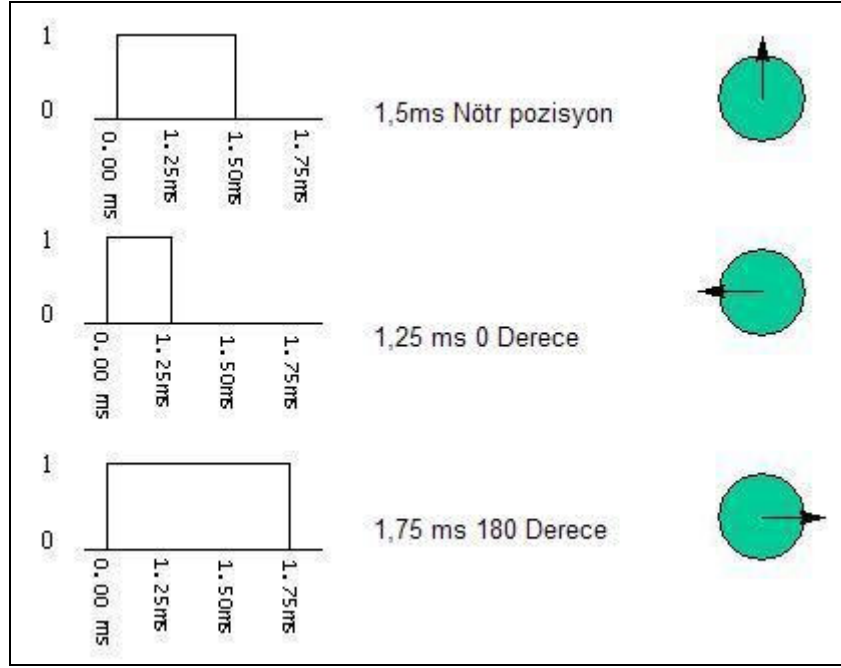
şaftın konumu servonun dahili elektronik kontrolörü tarafından aynen korunacaktır. Kodlu sinyaldeki değişimlere göre şaftın açısı da değişecektir.

Oldukça küçük boyutlu olmalarına rağmen servo motorlar, bu boyut içerisinde hem motoru, hem redüksiyon dişlilerini hem de gereken kontrol ve sürücü devrelerini içerirler. Kapladıkları alana oranla çok güçlüdürler ve çok az akım çekerler. Standart dc motorların aksine servo motorlar 3 kablolu bir ara yüz kullanırlar. Bu kablolar güç bağlantısı + 5V (Vcc - KIRMIZI), toprak (GND - SİYAH), kontrol sinyali girişi (Vss - SARI veya BEYAZ) şeklindedir.

Servolar bir kontrol devresi ve servonun çıkış şaftına bağlı bir potansiyometre yardımıyla çalışırlar. Bu potansiyometre, kontrol devresinin şaftın o anki konumunu anlayabilmesini sağlamaktadır. Kontrol devresinin görevi potansiyometreden okunan değer istenilen açiya karşılık geliyorsa motoru durdurmak; eğer okunan değer farklı ise motoru şaftı istenen açiya getirecek yönde çalıştırmaktır. Servo motorların çıkış şaftının dönme miktarı yaklaşık 0-180° aralığındadır. Bu açı üreticiye ve modele bağlı olarak 210° ye kadar artabilmektedir. Normal bir servo bu aralığın ötesinde bir açiya dönemez. Bu, dişli grubuna eklenen, mekanik bir sınırlama sayesinde sağlanmaktadır.

Kontrol devresince motora uygulanan güç, kat edilmesi gereken mesafe ile doğru orantılıdır. Eğer şaftın geniş bir açiyı kat etmesi gerekiyorsa motor tam hızda çalıştırılır. Eğer küçük bir açıda dönme gerekiyorsa motor hızı kontrol devresince düşürülür. Bu, oransal kontrol olarak adlandırılmaktadır.

Şaftın konumlanacağı açı, kontrol sinyal girişi üzerinden gönderilen bir sinyalin darbe boyu tarafından belirlenmektedir. PWM (darbe kodlu modülasyon) adı verilen bu teknikte her 20 milisaniyede (0.02saniye) bir darbe gönderilmektedir. Darbenin uzunluğu şaftın hangi açiya konumlanacağını bildirir. Örneğin 1.5 milisaniyelik bir darbe şaftın yaklaşık 90° ye konumlandırılmasını sağlar. Bu pozisyon genelde nötr pozisyon olarak adlandırılır. Eğer darbe 1.5 ms den kısa süreli ise şaft konumu 0° ye doğru, 1.5ms den uzun ise şaft 180° ye doğru ilerleyecektir.



Şekil 6.22 : Servo Motorda Darbe-Konum İlişkisi

Bu projede robotun algılayıcılarının ve kameranın hareket sistemlerinde 4 adet servo motor kullanılmıştır. 2 Adet Hitec-322HD servo motor (Şekil 6.23) ön algılayıcı taretinin yatay ve düşeydeki hareketlerini sağlamaktadır. Bu servolar daha uzun ömür ve daha hassas hareket için ağır yüke dayanıklı dişliler içermektedir. 4,8 V - 6V aralığında çalışabilen 322HD servolar 4,8 voltta 60° lik bir açığı 0,19 saniyede kat edebilmektedirler. 43 gram ağırlığındaki bu servolar 4,8 voltta 3kg-cm’lik tork üretebilmektedirler.



Şekil 6.23 : Hitec 322HD Servo Motor

Arka Sensör taretı için MPI MX-35 mini servo kullanılmıřtır (řekil 6.24). Standart servolardan daha k (21 x 11 x 22 mm) ve daha hafif (5,9 gram) olması ve yksek hızı (60°/ 0,09sn @ 4,8V) bu servoyu ideal bir algılayıcı hareket servosu haline getirmektedir. MPI MX-35 mini servo motor 4,8 voltta 1kg-cm tork retebilmektedir.

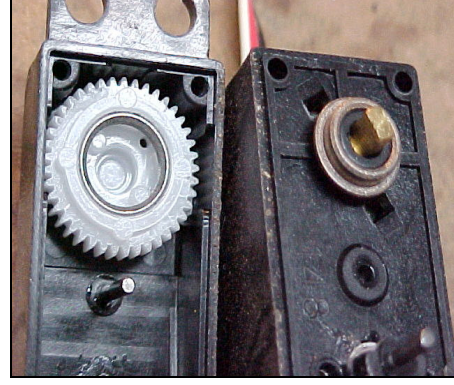


řekil 6.24 : MPI MX-35 Mini Servo Motor

Kamera taretı iin ise Futaba s3001 servo motor kullanılmıřtır (řekil 6.25). Bu motor 4,8 voltta 60°yi 0,23 saniyede kat edebilmektedir. 3.2kg-cm tork reten bu motor srekli dnř iin modifiye edilerek standart servolardaki 180° dnř aıřı sınırlaması kaldırılmıřtır. Bu modifiye iřlemi sırasında servonun kontrol potansiyometresinin alıřmasını saėlayan kontrol halkası (řekil 6.26) kaldırılarak servonun konum geri beslemesi iptal edilmiřtir.



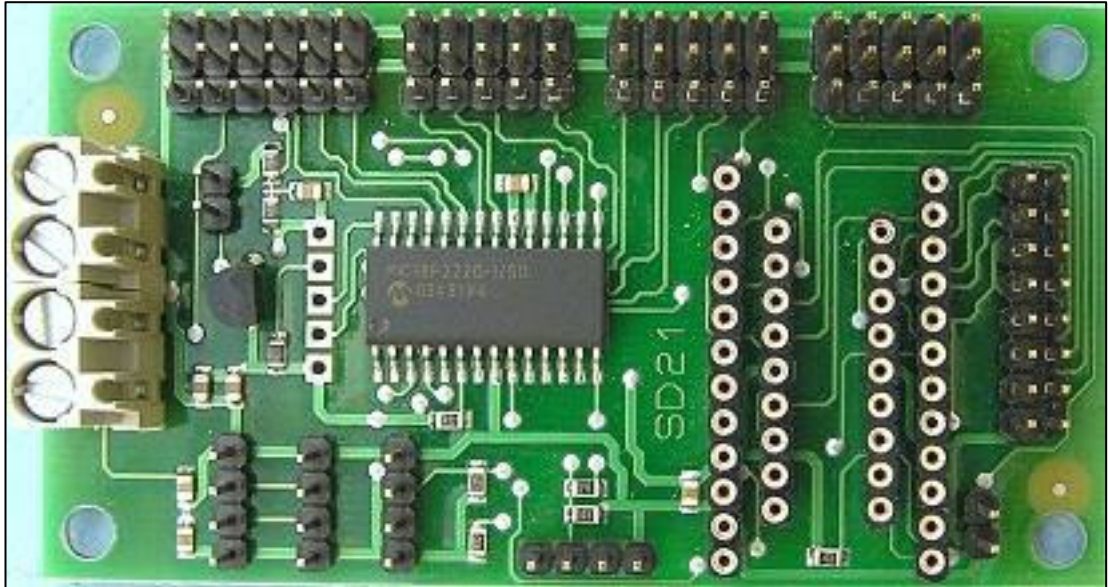
Şekil 6.25 : Futaba S3001 Servo Motor



Şekil 6.26 : Futaba Servo Motor Modifikasyonu

6.4.8 Servo Motor Kontrol Kartı

Robot üzerinde kullanılan 4 adet servonun kontrolü için Devantech firmasının “SD21 - 21 Servo Control Module” kontrol kartı[44] kullanılmıştır. Kart aynı anda 21 servo motoru kontrol edebilmekte ve servo sayısı ve servoların konumlarından (darbe genişliği) etkilenmeden 20ms tazeleme hızını koruyabilmekte; I2C veri yolundan gelen komutlarla kart üzerinde bulunan PIC18F2220 mikro işlemciyi kullanarak servo motorların hem konumunu hem de hızını kontrol edebilmektedir.



Şekil 6.27 : SD21 Servo Motor Kontrol Kartı

Kartın üzerinde 3 adet I2C portu bulunmaktadır. Bu portlardan herhangi biri karta komut göndermek için kullanılabilir. Diğer 2 porta I2C veriyolunu kullanan diğer donanımlar bağlanabilmektedir. Bunun yanı sıra Picaxe, BS2p, Atom, BX-24

vb. mikro işlemciler kartın üzerindeki yuvaya bağlanarak direk kontrol sağlanabilmektedir.

Kart, üzerinde bulunan 2 ayrı güç girişinden beslenmektedir. Birinci güç girişi +5v ile kart üzerindeki devreleri beslerken diğer güç girişi servo motorlara güç sağlar. Servo ve entegre güç devreleri baskı devre içinde ortak bir toprak hattına sahiptir. Servo motorlar için 4,8-6 volt aralığında bir voltaj gerekmektedir. İki ayrı güç kaynağı kullanımından kaçınmak için kart 7,2 voltluk bir batarya ile servo güç girişinden beslenebilmektedir. Kart üzerindeki ataç kapalı konumuna getirildiğinde dahili güç regülatörü devreye girerek hem entegreleri hem de servo motorları beslemektedir.

SD21 kontrol kartının kalbini ön-programlı PIC18F2220 mikro işlemci oluşturmaktadır. Bu mikro işlemciye I2C veri yolu ile 0xC2 adresi üzerinden erişilerek 21 ayrı servo motorun konum ve hızları ayarlanabilmektedir. Kart, her bir servo motor için 3 ayrı yazmaç barındırmaktadır. Bu yazmaçlar ilgili motorun hızı ve konum bilgisini tutarlar. Konum yazmaç 8 er bitlik 2 yazmaçın birleşimidir. 16 bitlik bu değer direk olarak mikro saniye cinsinden darbelere çevrilerek servo motorlara gönderilmektedir. Örneğin konum yazmaçlarına 1500 değerinin girilmesi ($1500\mu s=1,5ms$) servo motoru nötr konuma getirmektedir. Genellikle $1000\mu s$ ile $2000\mu s$ arasında bir darbe genişliği tüm servolarda kullanılabilir. Hitec servolarda ise $800-2200\mu s$ arasında değerler kullanılarak daha geniş bir açının taranması sağlanabilmektedir. Konum yazmaçları aynı zamanda okunabilen yazmaçlardır. Hız kontrollü konumlandırmalarda servo hareket ederken okuma yapılarak işlemin ne kadarının tamamlandığı okunabilmektedir.

Hız yazmaç ı ise servonun yeni konuma hangi hızda geleceğini kontrol kartına belirlemek için kullanılır. Eğer hız yazmaç ı 0x00 ise servo maksimum hızla konum yazmaç ındaki konuma yönlendirilir. Her güç çevriminde hız yazmaç ı varsayılan 0x00 değerine sabitlenir. Eğer servonun hareket hızı düşürölmek istenmiyorsa bu değer varsayılan değer olarak bırakılabilir. Hız değeri 0x00'da farklı bir değeri olarak atandığında ise her 20mikro saniyede bir bu değeri o anki konuma eklenir ve servoya gönderilir. Böylece adım adım istenen değeri ulaşılmış olur. Örneğin $1000\mu s$ konumundan $2000\mu s$ konuma giderken hız yazmaç ı 10 olarak atanmış ise işlem süresi 2 saniye olacaktır. Toplam işlem süresini hesaplamak için (6. 4) formülü kullanılmaktadır.

$$Süre = \frac{Bitiş_konumu - Başlangıç_konumu}{Hız_registeri} * 20ms \quad (6.4)$$

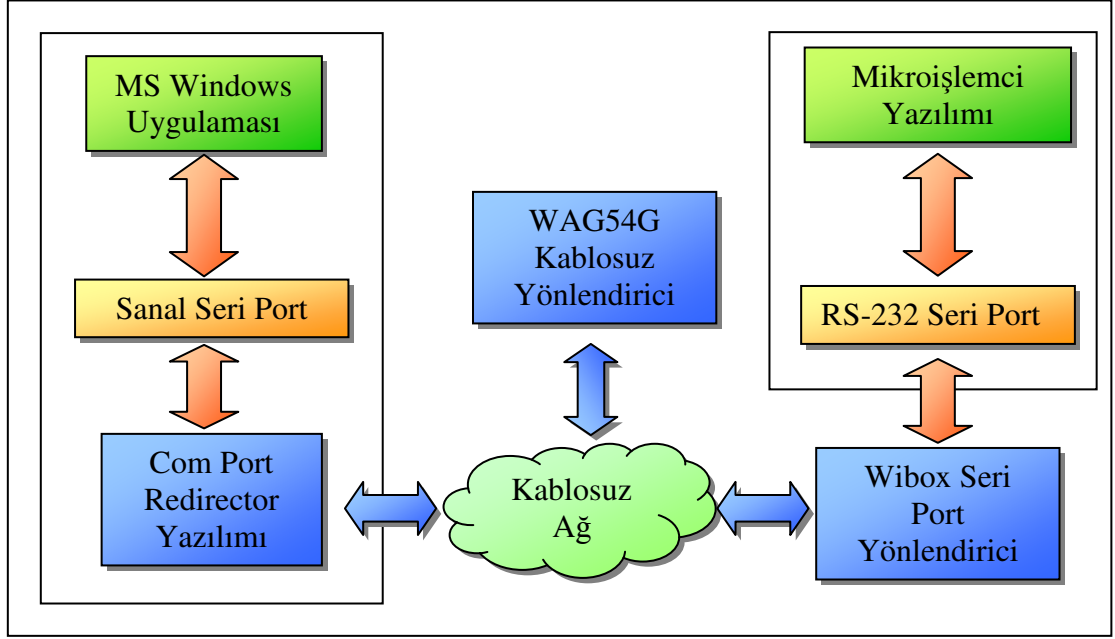
Her servo motor için bulunan bu 3 yazmaçın yanı sıra 0x40 ve 0x41 adreslerinde yazılım versiyon numarası ve güç kaynağı voltajı değerlerinin okunabildiği 2 ayrı yazmaç daha bulunmaktadır. Servo motor kontrol kartı ve dc motor kontrol kartı ortak akü kullandıklarından bu yazmaç yardımıyla motor besleme voltajı yazılım tarafından okunabilmektedir.

Algılayıcı ve kamera hareket sistemlerinde yer alan servolar kontrol kartının 0, 1, 2, 3 numaralı servo çıkışlarına bağlanmıştır. Bu servoların kontrolü için kullanılan yazılım metodları bölüm 7 de detaylı olarak incelenecektir.

6.5 Haberleşme Sistemi

Robotun kontrol yapısı ana bilgisayar ve ana kontrol kartı üzerinde çalışan programlardan oluşmaktadır. Bu programların birbiri ile sürekli bir iletişim içinde olmaları robotun üst seviye görevleri yerine getirebilmesi açısından büyük bir önem taşımaktadır. Ayrıca robotun mobil yapısı nedeniyle haberleşmenin kablosuz olması zorunluluğu bulunmaktadır. Bu nedenle ana bilgisayar ve robot arasındaki haberleşme 802.11B standardı üzerinden TCP/IP protokolü kullanılarak yapılmaktadır. TCP/IP protokolünün kullanılması ana bilgisayar ve robotun aynı ortamda bulunma zorunluluğunu ortadan kaldırmakta İnternet üzerinden robotun kontrolünün yapılabilmesini sağlamaktadır. Haberleşme sistemi ana bilgisayar üzerindeki kablosuz ethernet kartı, linksys wag54g (Şekil 6.30) kablosuz yönlendirici⁴⁵ ve robot üzerinde bulunan wibox (Şekil 6.29) kablosuz seri port yönlendiricisi olmak üzere 3 ana bileşenden oluşmaktadır. Bu bileşenlerin birbiri ile olan iletişimleri ve haberleşme sisteminin genel yapısı Şekil 6.28’de şematik olarak gösterilmiştir.

⁴⁵ İng: Router



Şekil 6.28 : Haberleşme Sistemi Genel Yapısı

6.5.1 Kablosuz Yönlendirici

Ana bilgisayar ve robotun haberleşmesinde kullanılan ağın yönetilmesi için Linksys wag54g kablosuz yönlendirici kullanılmaktadır. Bu yönlendirici IEEE 802.11 B ve G standartlarını desteklemektedir. Ayrıca barındırdığı dinamik dns desteği ile otomatik olarak bir internet alanına bağlanarak barındırdığı network'ün internet üzerinden erişilebilmesini sağlamaktadır. Bu yönlendirici proje kapsamında anabilgisayar, wibox kablosuz seri port yönlendirici ve kablosuz IP kameranın aralarındaki iletişimi sağlamaktadır.



Şekil 6.29 : Wibox Kablosuz Seri Port Yönlendirici



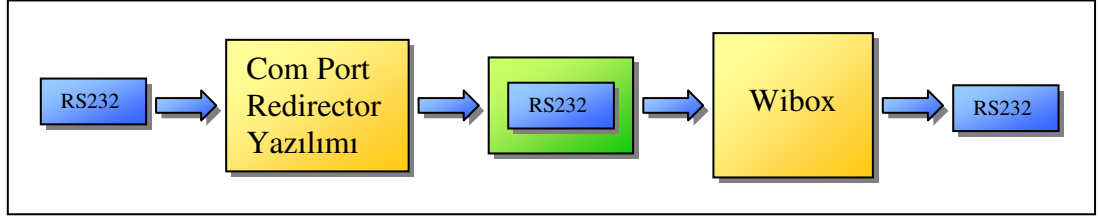
Şekil 6.30 : WAG54G Kablosuz Yönlendirici

6.5.2 Wibox Kablosuz Seri Port Yönlendirici

Ana kontrol kartının ana bilgisayar ile haberleşmesinde en büyük rolü wibox seri port yönlendiricisi oynamaktadır. Ana kontrol kartı üzerinde bulunan mikro işlemci RS-232 haberleşmesini desteklemektedir. Ancak kablo tabanlı bir iletişim sistemi olan RS-232'nin mobil bir robot üzerinde kullanımı mümkün olmamaktadır. Bu nedenle ana kontrol kartının gönderdiği RS-232 sinyallerini yorumlayıp kablosuz olarak ana bilgisayara iletecek ara bir donanıma ihtiyaç vardır.

Lantronix Wibox seri port yönlendirici, seri porta (RS232/422/485) sahip cihazların Ethernet ortamı üzerinden iletişimini sağlayan bir cihazdır. Ethernet arayüzüne sahip olmayan ancak iletişim için seri port protokollerini (RS-232, RS-422, RS-485) kullanan herhangi bir cihaz wibox ile ethernet ağına bağlanabilmektedir. Lantronix WiBox hareket kabiliyeti gerektiren ve kablolanmanın pratik olmadığı uygulamalarda cihazlara esnek bir bağlantı olanağı sağlamaktadır. Mevcut olan yazılımlar ve donanımlar üzerinde herhangi bir değişiklik yapmadan iki uç arasında transparan bir seri iletişim tüneli yaratmaktadır. Seri tünelleme olarak bilinen bu metod ile wibox tünelin iki ucundan aldığı RS-232 verilerini enkapsüle edip 802.11B kablosuz ağ standardını kullanarak aktarır. Noktadan noktaya seri iletişim sağlayabildiği gibi noktadan çok noktaya iletişim kurulması da mümkündür. WiBox ağ içinde konumlandırılıp, seri tüneller yaratılarak, sanal seri bağlantıları lokal ağ üzerinde veya Internet üzerinde herhangi bir noktaya erişebilecek şekilde oluşturmak mümkündür. WiBox ile birlikte gelen Com Port Redirector yazılımı Com port tabanlı Windows uygulamalarının herhangi bir değişiklik yapılmadan sanal com portlar üzerinden seri haberleşme yapmasını sağlamaktadır. Windows tabanlı program tarafından com porta gönderilen veriler enkapsüle edilerek kablosuz ağ üzerinden wibox'a aktarılmakta ve burada paket açılarak tekrar RS-233 verisine çevrilmektedir. Bu işlem Şekil 6.31'de şematik olarak gösterilmiştir.

Üzerinde tam tcp/ip yığını ve bir işletim sistemi barındıran cihaz eklendiği cihazın uzaktan konfigürasyonu, gözlem ve sorun gidermesini sağlayan dahili web sunucusuna sahiptir. WiBox ayrıca, WEP, WPA, TKIP kablosuz ağ şifreleme teknolojilerini de desteklemektedir.



Şekil 6.31 : Seri Tünelleme İşlemi

Wibox üzerinde 2 adet seri port barındırmaktadır. Bunlardan birincisi sadece RS-232 standardını desteklerken diğer port yazılım ile RS-232, RS-422, RS-485 standartlarına göre ayarlanabilmektedir. 300 - 921,600 bps hızlarında iletişimi destekleyen her bir fiziksel seri port'a wibox'ın IP numarası üzerinde yaratılan TCP veya UDP portu üzerinden ulaşılabilir. Robot üzerindeki 12v akü ile beslenen cihaz, üzerinde güç, kablosuz bağlantı ve seri portlar için led göstergeler barındırmaktadır. Wibox tamamen transparan olarak çalıştığı için ana bilgisayar ve ana kontrolör üzerinde çalışan yazımlar üzerinde herhangi bir değişiklik yapılmasına gerek bulunmamaktadır.

6.6 Algılayıcılar

Robotun çevresini algılaması ve bu algılara göre kararlar verebilmesi için farklı görevler yürüten birçok algılayıcı robota entegre edilmiştir. Bunların arasında mesafe, ivme, yönelim, ışık şiddeti, hareket ve sıcaklık gibi birçok fiziksel parametreyi ölçen algılayıcılar bulunmaktadır.

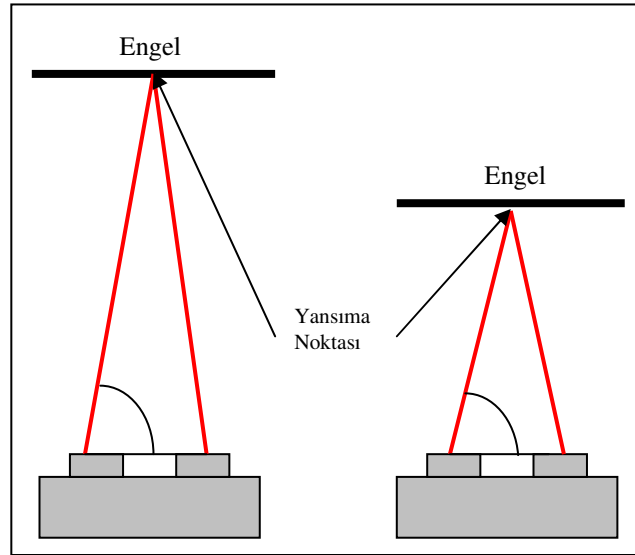
6.6.1 Kızılötesi Mesafe Algılayıcılar

Geçmişte sharp kızılötesi alıcıları, uygun sürme devreleri ve bir veya birden fazla kızılötesi led kullanılarak robotlarda engel algılama amaçlı kullanılmaktaydılar. Bu yaklaşım tetiklenen her bir led için yansımanın olup olmasına göre mantıksal doğru veya yanlış değerleri olarak okuma yapmaya elvermekteydi. Ancak bu yaklaşımın sınırlı ölçüm mesafesi, çevresel ışığın yarattığı girişime duyarlılığı ve led, sürme devresi, kızıl ötesi gibi donanımlara olan ihtiyacı kullanımında sorunlar doğurmaktaydı [45].

Sharp firmasının GP2DXX serisi algılayıcılarını piyasaya sürmesi ile birlikte yeni bir yaklaşım kullanılmaya başlandı. Bu yaklaşım eski yaklaşıma göre daha uzun bir menzil sunmasının yanı sıra GP2D02 ve GP2D12 algılayıcılarının kullanılması durumunda mesafe bilgisinin okunmasına da olanak tanımaktadır. Yeni algılayıcı

serisi kullanılan yöntem nedeniyle aynı zamanda çevresel ışığın etkilerine daha iyi bir tolerans göstermektedir.

Bu algılayıcıların tamamı üçgenleme⁴⁶ metodu ve küçük lineer CCD algılayıcı dizileri kullanarak mesafe ve/veya görüş alanında bir nesnenin olup olmadığı hesabını yapmaktadırlar. Temel düşünce yayılan kızıl ötesi ışığın görüş alanında bir nesne varsa bu nesneye çarpıp algılayıcıya geri yansıtacağı prensibine dayanmaktadır. Çalışma esnasında ilk olarak kızıl ötesi bir darbe yayınlanır. Bu darbe görüş alanında yayılmaya başlar. Bu yayılma sırasında ya bir engelle çarpıp geri yansıtacak ya da yayılmaya devam edecektir. Kızıl ötesi ışının bir engelden yansımaması durumunda hiç bir nesne algılanmayacaktır. Bir engelle karşılaşılması durumunda ise yansıyan ışık algılayıcının üzerinde bulunan CCD dizisine geri dönecektir. Bu durumda kızıl ötesi verici, nesne ve CCD dizisinin, köşelerini oluşturduğu bir üçgen meydana gelecektir.

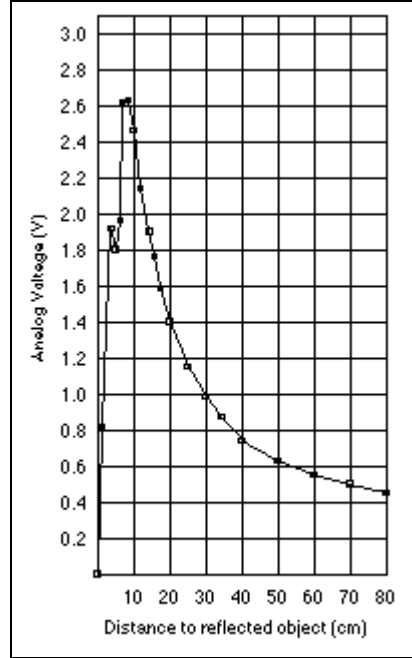


Şekil 6.32 : Kızılötesi Algılayıcıların Çalışma Prensibi

Oluşan bu üçgenin taban açıları yansımanın olduğu engelin mesafesine göre değişiklik göstermektedir. Yeni algılayıcıların alıcı kısmında bulunan hassas lensler, yansıyan kızıl ötesi ışının geldiği açıya göre algılayıcının içinde bulunan lineer CCD dizisine yansıtır. Ardından algılayıcı CCD dizisini okuyarak ışının hangi açı ile geldiğini hesaplar. Açı ve kızılötesi verici alıcı arasındaki uzaklık bilindiği için mesafe trigonometrik formüllerle hesaplanır.

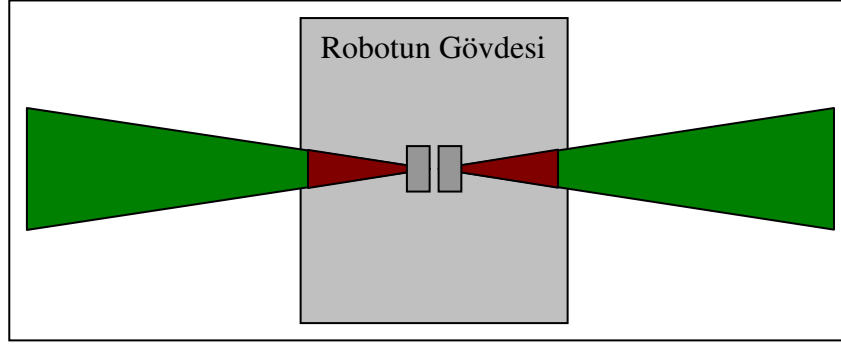
⁴⁶ İng: triangulation

Bu yöntem çevresel ışığa karşı çok az duyarlılık göstermekte ve bu nedenle çok farklı ortamlarda doğru okumalar elde edilmesini sağlamaktadır. Bu aynı zamanda engelin renginin de önemli olmadığı anlamına gelmektedir. Güneş ışığı altında siyah bir engel bile bu yöntemle algılanabilmektedir.



Şekil 6.33 : GP2D12 Algılayıcısı için Mesafe Voltaj Grafiği

Kullanılan trigonometrik hesapların doğası nedeniyle bu algılayıcıların sinyal çıkışlarındaki voltaj, kızıl ötesi ışının yansıdığı mesafe ile lineer olmayan bir ilişki göstermektedir. Şekil 6.33'da görülen iki önemli unsur bulunmaktadır. Grafik 10-80 cm aralığında logaritmik bir eğriye yakın bir eğilim göstermektedir. Algılayıcının minimum ölçüm mesafesi olan 10 cm'in altına düşüldüğünde ise çıkış voltajı ani bir düşüş göstermekte ve okunan değerler engelin olduğundan daha uzakta olduğu izlenimini yaratmaktadır. Bu robotlar için büyük bir sorun oluşturabilecek bir durumdur. Robotun önünde bir engel olmasına rağmen ilerlemeye çalışması hasarlara neden olabilmektedir. Bu sorunun çözümü için bu algılayıcıların, ölü bölgeleri (0-10cm) robotun gövdesi içinde kalacak şekilde monte edilmeleri gerekmektedir (Şekil 6.34).



Şekil 6.34 : Kızılötesi Algılayıcıların Montajı

Sharp kızılötesi algılayıcıların ışın demeti yapısının genişliği ise yaklaşık 16 cm'dir. Bu görece düşük demet genişliği, Sharp algılayıcılarını özellikle servo motorlarla yapılan açısal süpürme işlemleri için ideal kılmaktadır. Aynı zamanda sunulan geniş ürün yelpazesi her türlü mikro işlemci için uygun bağlantı seçenekleri sağlamaktadır. Tablo 6.2'de en yaygın olarak kullanılan Sharp kızıl ötesi algılayıcıların özellikleri verilmiştir. Görüldüğü gibi GP2D02 ve GP2D12 gerçek mesafe değerlerinin okunmasına izin verirken GP2D05 ve Gp2D15 sadece engel olup olmadığının algılanmasını sağlayabilmektedir. Algılayıcılardan bazıları dijital okumaya izin verirken bazıları ise analog voltaj çıkışı vererek mikro işlemcinin analog dijital çevirici barındırmasını zorunlu kılmaktadır.

Tablo 6.2 : Kızılötesi Mesafe Algılayıcılar Özellik Tablosu

Algılayıcı	Çıkış Türü	Menzil	Çalışma Metodu	Çalışma Akımı	Bekleme Akımı
GP2D02	Seri arabirim	10cm - 80cm	Harici olarak okuma işlemi tetiklenir.	~25 mA	~2 uA
GP2D05	Sabit bir mesafeye göre Boolean (1 veya 0)	10cm - 80cm (dahili bir potansiyometre ile ayarlanabilir)	Harici olarak okuma işlemi tetiklenir.	~25 mA	~2 uA
GP2D12	Ölçülen mesafeye göre değişen Analog değer (0V to ~3V)	10cm - 80cm	Sürekli okuma (her okuma~38ms)	~25 mA	*
GP2D15	Dijital çıkış (0 veya 1)	Fabrika çıkışı 24cm sabitlenmiş	Sürekli okuma (her okuma~38ms)	~25 mA	*

Bu projede yakın mesafe algılamaları için robot üzerinde 4 adet Sharp GP2D12 kızılötesi mesafe algılayıcı [46] bulunmaktadır. Bu algılayıcılardan ikisi robotun yanlarında gövdeye, diğer ikisi ise ön ve arkada bulunan algılayıcı taretlerine monte

edilmiş durumdadır. Her bir algılayıcının analog çıkışı PIC mikro işlemcinin PortA portunda bulunan analog dijital çeviricinin girişlerine bağlanmış durumdadır. Okunan voltaj değerlerinin mesafeye çevrilme işlemi ile ilgili detaylar bölüm 7 de incelenecektir.

6.6.2 Ultrasound Mesafe Algılayıcılar (Sonar)

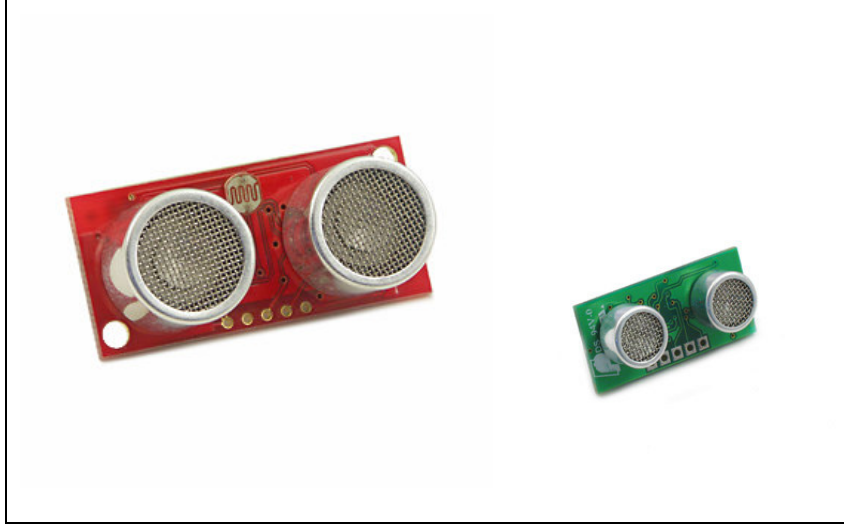
Sonarlar yayılan ses dalgalarının yansımalarının ölçülmesi ile çalışan algılayıcılardır. Sonar günümüzde deniz taşıtlarında (özellikle denizaltı ve savaş gemilerinde), balık sürülerinin tespit vb uygulamalarda yaygın olarak kullanılan bir teknolojidir. Doğa'da ise yarasalar benzer bir sistemi kullanarak yollarını ve hatta avlarını bulurlar. Robotik uygulamalarında ise engellerin algılanmasında sonarlar yaygın olarak kullanılmaktadır [45].

Sonar ilk olarak ultrasonik bir ses dalgası yayar ve ardından yankıları dinlemeye başlar. Ses dalgasının yayılmasından yankının algılanmasına kadar geçen süre ölçülerek yankının gerçekleştiği nesnenin uzaklığı hesaplanabilir. Sesin havada yaklaşık 340,29 metre/saniye hızla ilerlediği düşünülürse oldukça hızlı bir ölçüm yapılabildiği görülmektedir.

Sonarların bu avantajlarının yanı sıra bazı dezavantajları da bulunmaktadır. Ultrasonla mesafe ölçümündeki ana dezavantaj, nesnelerin yüzeyinden gerçekleşen yansıma ile ilgili problemlerdir. Buna aynasal yansıma adı da verilmektedir. Yansıma yönü gelen ses dalgasının yüzeyle yaptığı açıyla ve yüzeyin şekliyle ilgilidir. Geliş açısı ne kadar küçük olursa, sesin yansıma yapmadan yüzeyi sıyırması ihtimali o kadar yükselir ve hatalı bir uzak mesafe ölçümü yapılır. Bu duruma aynasal denmesinin sebebiyse, kaygan yüzeylerin, yansıtıcı özellikleri ile bu sorunun büyümesine yol açmalarıdır. Kaba yüzeylerde ise düzensiz yansılardan birinin geri dönme ihtimali daha yüksektir.

Uzak mesafelerde ise ölçümlerin kesinliği büyük oranda düşecektir, bunun sebebi yanlış ölçümlerin dönmesi ihtimalinin yüksek olmasıdır. Özellikle sesin ışıktan daha yavaş hareketi göz önüne alındığında yansımaların oluşturduğu girişim yanlış okumalara sebep olabilmektedir. Aynı zamanda ses dalgalarının kızılötesi ışınlarda olduğu gibi bir demet halinde odaklanamaması çok geniş bir yayılma alanı oluşturmakta ve bazı durumlarda zeminden yansıyan ses dalgalarının yanlış okumalara sebep olmasına rastlanmaktadır. Tüm bu dezavantajlarına rağmen ultrason

ölçümleri hareketli robot uygulamalarında sıklıkla uygulanmaktadır. Bu uygulamalar arasında iç mekan ve dış mekan haritalarının çıkarılması da yer almaktadır.



Şekil 6.35 : Ultrasonik Mesafe Algılayıcılar

Bu projede devantech firmasının diğer ultrasonik algılayıcılara göre daha ekonomik⁴⁷ olarak piyasaya sürdüğü SRF10 (sağda) [47] ve SRF08 (solda) [48] ultrasonik mesafe algılayıcıları kullanılmıştır (Şekil 6.35). Her iki algılayıcı 5v besleme gerilimine ihtiyaç duymakta, çalışma anında 15mA, bekleme konumunda ise yaklaşık 3mA güç tüketmektedirler. 40KHz frekansında ultrasonik dalgalar kullanan bu sonarlar 3cm-6m arasında engelleri algılayabilmektedirler.

SRF10 PIC16F87 mikroişlemci SRF08 ise PIC16F872 mikro işlemci barındırmaktadır. Her iki algılayıcının tetiklenmesi ve okunması I2C veri yolu üzerinden yapılmaktadır. SRF10 ve SRF08'in varsayılan fabrika çıkışı I2C adresi 0xE0'dır [47,48]. Ancak I2C veri yolu üzerinden gönderilecek komutlarla E0, E2, E4, E6, E8, EA, EC, EE, F0, F2, F4, F6, F8, FA, FC veya FE şeklinde değiştirilebilmekte aynı anda 16 sonarın aynı veri yolu üzerinde kullanılabilmesi sağlanmaktadır. SRF10 I2C üzerinden erişilebilen 4 adet yazmaç (Tablo 6.3) barındırırken SRF08 36 adet yazmaç (Tablo 6.4) barındırmaktadır. Sadece 0, 1, 2 numaralı yazmaçlara yazılabilmektedir. 0x00 yazmaç komut yazmaçıdır. Buraya yazılan değere göre algılayıcı üzerindeki mikro işlemci ölçüm yapmakta 16 bitlik ölçüm değerinin ilk 8 bitini 2 numaralı ikinci 8 bitini 3 numaralı yazmaça yazmaktadır. 1 numaralı yazmaç ise ana mikro işlemci tarafından okunduğunda 0x80

⁴⁷ Yaklaşık 25 GBP

değerini vermektedir. Bu değer test edilerek cihazın çalışıp çalışmadığı kontrol edilebilmektedir.

Tablo 6.3 : SRF10 Yazmaç Adresleri

Yazmaç Adresi	Okuma	Yazma
0x00	Yazılım versiyonu	Komut Yazmaç
0x01	Kullanılmıyor (0x80 okunur)	Sinyal Kazancı Yazmaç (varsayılan 16)
0x02	Mesafe ilk Byte	Menzil Yazmaç (varsayılan t 255)
0x03	Mesafe ikinci Byte	Salt okunur

Tablo 6.4 : SRF08 Yazmaç Adresleri

Yazmaç Adresi	Okuma	Yazma
0	Yazılım versiyonu	Komut Yazmaç
1	Işık Şiddeti Algılayıcı	Sinyal Kazancı Yazmaç (varsayılan 31)
2	1. yankı ilk Byte	Menzil Yazmaç (varsayılan t 255)
3	1. yankı ikinci Byte	Salt okunur
~~~~	~~~~	~~~~
34	17. yankı ilk Byte	Salt okunur
35	17. yankı ikinci Byte	Salt okunur

SRF10 ve SRF08 ana mikro işlemci tarafından kontrol edilirken komut yazmaçına yazılan değer algılayıcının mesafeyi nasıl bildireceğini belirlemektedir. 0x50 inç, 0x51 santimetre, 0x52 değeri ise mikro saniye cinsinden uzaklığın okunmasını sağlamaktadır. Her ölçüm başladığında algılayıcı mesafe yazmaçlarını silmekte ve ölçüm tamamlandığında komut yazmaçında verilen değere göre mesafeyi hesaplayıp bu yazmaçlara işlemektedir. Ölçümün tamamlanması için varsayılan süre (algılayıcının yankıları beklediği süre) 65ms'dir. Ancak bu süre istenirse ölçüm komutu verilmeden önce menzil yazmaçının değeri değiştirilerek kısaltılabilmektedir.

Algılayıcının dahili zamanlayıcısı ile ölçülen bekleme süresi normalde 65ms, yaklaşık 11metredir. Ancak bu rakam srf10 ve srf08'in ölçüm yeteneği olan 6 metrenin çok üstündedir. Bu nedenle menzil yazmaçının değeri değiştirilerek srf10'un yankıyı beklediği süre dolayısıyla menzil, 43mm'lik adımlarla 43mm-11m arasında değiştirilebilmektedir. Menzilin değiştirilmek istenmesinin iki önemli

nedeni bulunmaktadır. Bunlardan biri okumanın daha hızlı olarak alınması diğeri ise sonarın daha sık aralıklarla tetiklenmek istenmesidir.

Sonarlar daha sıklıkla ateşlendiklerinde (bekleme süresi ve mesafe kapasitesi azaltıldığında) bir önceki ateşlemenin yankıları ölçüm sırasında alınabilmekte ve yanlış okumalar oluşabilmektedir. Bu nedenle SRF10 ve SRF08'un mesafe ölçümünün analog safhalarında kullandıkları analog sinyal kazancı 1 numaralı kazanç yazmaçının değeri ile oynanarak değiştirilebilmektedir. Bu sayede algılayıcıların hassasiyeti ayarlanmakta, uzak bir nesneden gelen yankı daha zayıf olacağı için bu yankının göz ardı edilmesi sağlanmaktadır.

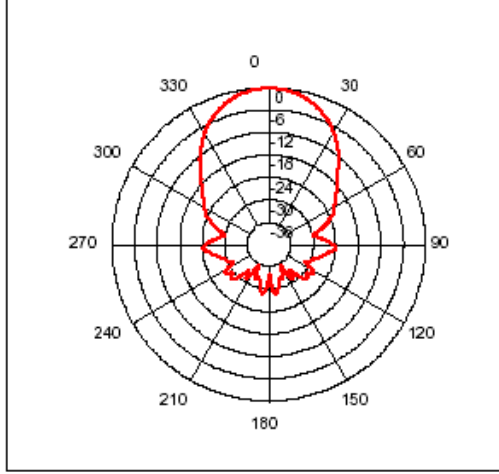
Her güç çevriminde mesafe ve kazanç yazmaçları varsayılan değerlere dönmektedir. Bu nedenle her seferinde bu yazmaçların değerlerinin atanması gerekmektedir.

SRF10 ve SRF08 çalışma prensibi olarak benzer olmalarına rağmen teknik özellikleri açısından farklılıklar göstermektedirler. SRF10 32mm x 15mm x 10mm boyutlarında iken SRF08 43mm x 20mm x 17mm boyutlarındadır. SRF10'un küçük boyutları daha küçük robotlarda kullanılmasına ve daha küçük motorlarla hareket ettirilebilmesine olanak sağlamaktadır.

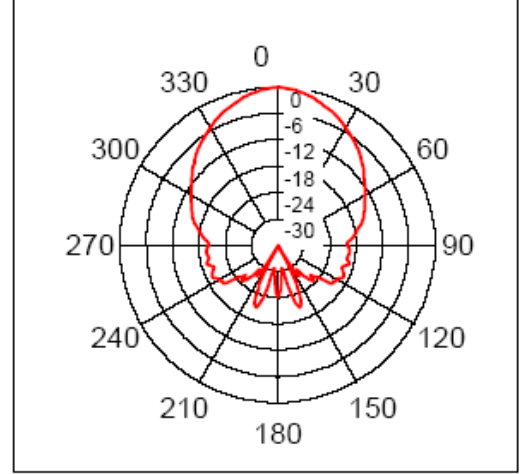
SRF10 algıladığı ilk yankı için ölçüm değerlerini bildirebilirken SRF08 ilk 17 yankı için mesafe ölçümü yapabilmektedir. Ayrıca SRF08 üzerinde barındırdığı ışık şiddeti algılayıcı sayesinde değişik uygulamalara adapte edilebilmektedir. SRF08 desteklediği komutlar açısından da farklılık göstermektedir. SRF10'un desteklediği komutların yanı sıra 0x53, 0x54, 0x55 komutları ile yapay sinir ağlarında kullanılmak üzere sonuçlar üretebilmektedir. Yapay sinir ağı modunda çalışırken mesafe yazmaçları (4-35 nolu yazmaçlar) 32 byte uzunluğunda bir tampon bellek görevi yapmaktadır. Bu modda 65ms'lik (65536uS) toplam bekleme süresi 2048uS'lik 32 eşit parçaya bölünmektedir. 2048uS'lik her bir parça 352mm'lik bir mesafeyi kapsamaktadır. Eğer ilgili yazmaçın zaman dilimi içerisinde bir yankı algılanırsa o yazmaçın değeri 1 olarak atanmaktadır. Böylece 32 nörona sahip bir yapay sinir ağının eğitilmesi sağlanabilmektedir.

İki algılayıcı yaydıkları ultrasonik dalga yayım deseni açısından da farklılık göstermektedir. SRF08 55°'lik konik bir yayım desenine (Şekil 6.36) sahipken srf10

72°lik bir desene (Şekil 6.37) sahiptir.⁴⁸ Robotta biri (SRF08) ön algılayıcı taretı diğeri (SRF10) arka algılayıcı taretı üzerinde olmak üzere 2 adet ultrasonik mesafe algılayıcı yer almaktadır. Bu algılayıcıların ana mikro işlemci tarafından sürülmeleri ile ilgili yazılım detayları 7. bölümde incelenecektir.



Şekil 6.36 : SRF08 Dalga Deseni



Şekil 6.37 : SRF10 Dalga Deseni

### 6.6.3 Elektronik Pusula

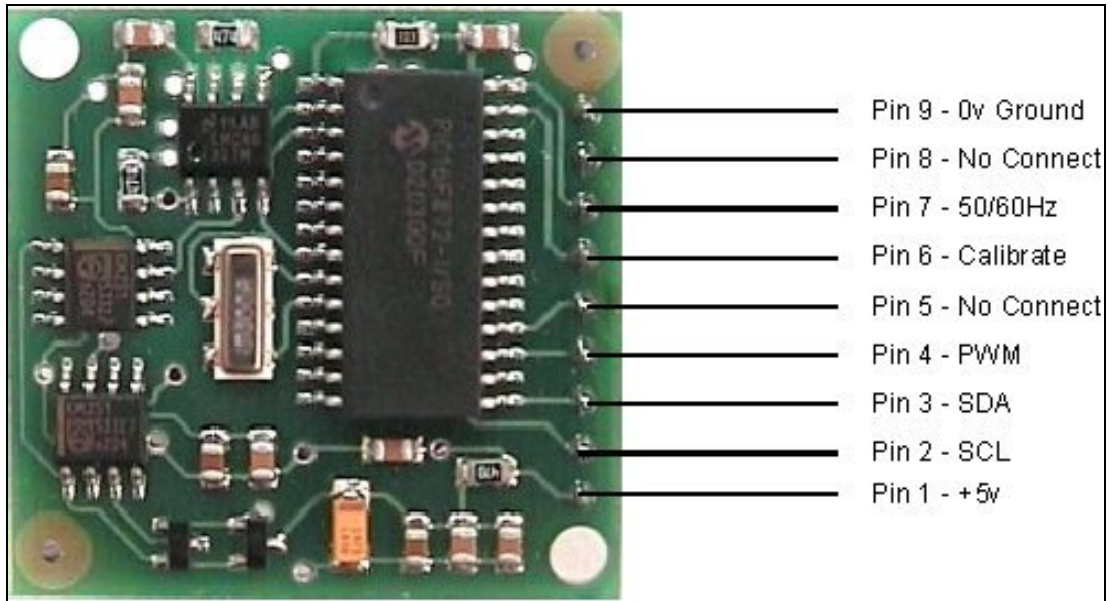
Robotun hareketi esnasında hangi yöne doğru gittiğinin bilinmesi karar verme aşaması için oldukça büyük bir önem taşımaktadır. Ayrıca dönüşler ve ilerleme sırasında oluşabilecek açısal sapmaların düzeltilebilmesi için robotun yönelim bilgisine sahip olması gerekmektedir. Bu nedenle çoğu mobil robot uygulamasında elektronik pusulalar kullanılmaktadır [45].

Elektronik pusulalar dünyanın manyetik alanı algılayarak çalışırlar. Bilindiği gibi Dünya'nın manyetik alanı, merkezden 440 km dışıdır ve dönüş eksenini ile 11°lik bir açı yapmaktadır. Manyetik kuzey ile coğrafik kuzey arasındaki bu fark, bir robotun navigasyonunda ancak çok uzun mesafeler kat edilecek ise bir hataya sebep olmaktadır. Bu nedenle manyetik alanın algılanarak buna göre işlem yapılması çoğu durumda herhangi bir sorun teşkil etmemektedir. Ancak 0,5 Gaussluk bir manyetik alanı algılayabilecek bir algılayıcının üretimi oldukça zordur. Bu kadar küçük güçteki bir manyetik alanı okuyabilecek bir kaç türde algılayıcı mevcuttur. Bunlar

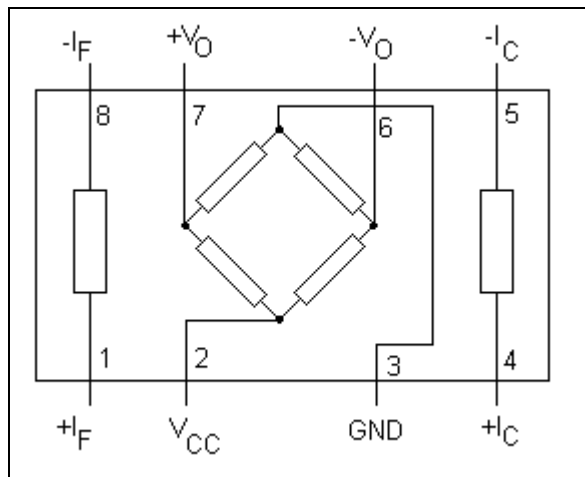
⁴⁸ Devantech firmasının çok daha dar bir desene sahip olan SRF235 algılayıcısı bulunmasına rağmen maliyeti (55 GBP) nedeniyle tercih edilmemiştir.

mekanik manyetik pusulalar, fluxgate pusulalar, hall-effect pusulalar, manyeto-elastik pusulalar ve manyeto-direnç pusulalar olarak sıralanabilir.

Bu projede kullanılan Devantech CMPS03 elektronik pusula manyeto-direnç esasına göre çalışan Philips'in KMZ51 algılayıcısını kullanmaktadır [49]. Bu algılayıcı manyeto-direnç özelliğine sahip çok ince permalloy⁴⁹ tabaka barındırmaktadır. Manyeto-direnç tabakanın oluşturduğu bir Wheatstone köprüsü ile entegre kompanzasyon ve yükleme /boşaltma bobinlerinin oluşturduğu bu algılayıcının basitleştirilmiş devre şeması Şekil 6.39'de bulunmaktadır.



Şekil 6.38 : CMPS03 Pin Bağlantı Şeması



Şekil 6.39 : KMZ51 Basit Devre Şeması

⁴⁹ Kolaylıkla manyetize ve demanyetize olabilen patentli bir nikel-çelik alaşımı.

Algılayıcı yönelim doğrultusu ile azimut doğrultusu arasındaki açıyı (Azimut açısı⁵⁰) ve yönelim doğrultusunun yatayla yaptığı açıyı ölçerek doğrultusunu hesaplamaktadır. Devantech CMPS03, birbirine dik açıyla yerleştirilmiş 2 adet KMZ51 entegresinin yanı sıra tüm kontrol ve haberleşme işlemlerini yürütmek üzere PIC16F872 mikro işlemci barındırmaktadır.

Yönelim verileri darbe genişlik modülasyonlu (pwm) sinyal veya I2C veriyolu üzerinden okunabilmektedir. PWM modunda her bir darbenin genişliği 1mS'den (0°) 36,99mS ye (359,9°) kadar değişen bir yapı sergilemektedir. Darbe uzunluğu ölçülerek 0.1 derecelik hassasiyetle yönelim bilgisi elde edilebilmektedir.

I2C modunda ise okuma ana mikro işlemci tarafından algılayıcıya gönderilen komutlar ile gerçekleştirilir. Bu amaçla cmcs03 16 yazmaç barındırır. Yazmaç 0 yazılım versiyonunu, yazmaç 1 ise 0-255 aralığında yönelim bilgisi kaydını tutar. Yazmaç 2 ve 3 toplamda 16 bitlik 0-3599 aralığında yönelim bilgisi tutar. Bu iki yazmaç okunarak 0.1 derecelik hassasiyetle yönelim bilgisi okunabilmektedir. 4, 5, 6 ve 7. yazmaçlar dahili test amacı için kullanılmaktadır. 8, 9, 10, 11 ise kalibrasyon işlemi için ayrılmıştır. 12-13-14 numaralı yazmaçlar okunduğunda ise 0 değeri elde edilir. Bu yazmaçlar okunarak haberleşmenin doğru çalışıp çalışmadığı test edilebilmektedir. 15 numaralı yazmaç ise kalibrasyon komut yazmaçıdır. Bu yazmaça 255 değerinin yazılması kalibrasyon işlemi başlatmaktadır.

Projede I2C modu kullanılarak yönelim bilgisi okunmaktadır. Okunan bu değer sağ ve sol dönüşlerin açısal hata düzeltilmesinde ve ilerleme sırasında motorlardan kaynaklanan açısal sapmaların en aza indirgenmesinde kullanılmaktadır. Elektronik pusulanın okunması ve navigasyonda kullanılması ile ilgili yazılım detayları bölüm 7'de incelenecektir.

#### **6.6.4 İvme Ölçer**

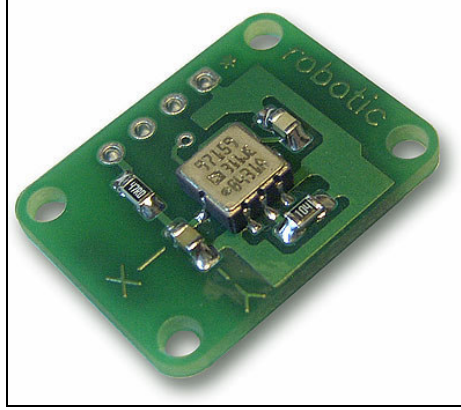
İvme ölçerler (accelerometer) konum, hareket, eğilme, şok vibrasyon ve ivme ölçümünde kullanılan mikro elektro-mekanik sistemlerdir. 1, 2 ve 3 eksenli ölçüm yapabilen modeller bulunmaktadır. İvme ölçerler hava taşıtlarının stabilite ve kontrol sistemleri, güdümlü füzeler, yük dengelemesi yapan çamaşır makineleri, araba alarmlarının arabanın çekici tarafından çekildiğini anlaması gibi çok farklı alanlarda

---

⁵⁰ Yönelim ile manyetik kuzey arasındaki açı.

kullanılmaktadır. Günümüzde cep telefonu ve video oyun konsollarının kumandaları gibi cihazlarda veri girişi için dahi ivme ölçerler kullanılmaya başlamıştır.

Mobil bir robotun hareketinin kinematik olarak incelenebilmesi, hızın ve kat edilen mesafenin hesaplanabilmesi için anlık ivmenin bilinmesi büyük önem taşımaktadır. Bu nedenle çoğu mobil robot uygulamasında çeşitli ivme ölçerler kullanılmaktadır.



Şekil 6.40 : AccelBoard İvme Ölçer

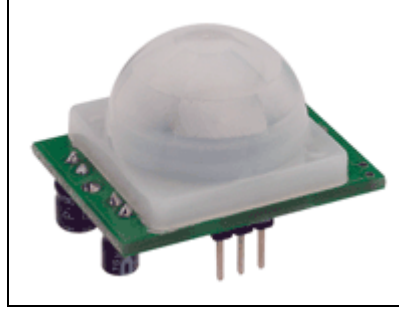
ITUbot üzerinde Mikroelektronika Accel Board ivme ölçer kullanılmıştır (Şekil 6.40). Üzerinde ADXL311 ivme algılayıcı [50] barındıran accel board 2 eksen üzerinde -2g +2g aralığında ölçüm yapabilmektedir. Her iki eksen için analog çıkış sağlayan kart ana kontrol kartı üzerinde A portunun analog dijital çeviricileri ile sisteme dahil edilmiştir. İvme değerlerinin okunması ve kalibrasyon ile ilgili işlemler ana kontrol kartı üzerinde çalışan yazılım tarafından yürütülmektedir.

### 6.6.5 Diğer Algılayıcılar

Önceki bölümlerde sayılan algılayıcıların yanı sıra robot üzerinde 1 adet pasif kızılötesi hareket algılayıcı, 1 adet ds18b20 elektronik termometre [51], srf08 üzerinde bulunan ışık şiddeti algılayıcı ve robotun engel sakınma davranışının başarılı olamaması durumunda çarpışma algılamaya yarayan dokunma algılayıcılar bulunmaktadır.

Pasif kızıl ötesi hareket algılayıcılar kızıl ötesi radyasyona maruz kaldığında elektrik üreten elamanlar barındırırlar. Kart üzerinde bulunan fresnel lens ile çevrede bulunan pasif kızıl ötesi ışınları pyroelektrik algılayıcının üzerine odaklanmaktadır. Algılayıcının üzerine gelen kızılötesi radyasyon miktarı değiştiğinde üretilen voltaj da değişmektedir. Bu voltaj baskı devre üzerindeki güçlendirici ile yükseltilerek sürekli olarak ölçülmekte ve önceki değerlerle karşılaştırılmaktadır. İnsan vb gibi

canlılar hareket ettiklerinde yaydıkları radyasyon oranı dolayısıyla çevresel kızılötesi radyasyon da değişecektir.



**Şekil 6.41 : Pasif Kızıl Ötesi Hareket Algılayıcı**

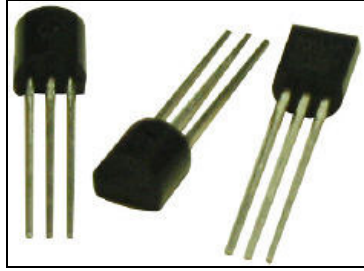
Çevresel kızıl ötesi radyasyonda ani bir değişiklik olması durumunda algılayıcı bunu hareket olarak algılamaktadır. Günümüzde alarm sistemlerinde kullanılan hareket algılayıcılar da aynı sistemle çalışmaktadır. Robot üzerinde bulunan parallax PIR algılayıcısı yaklaşık 6 metrelik bir alandaki hareketi algılayabilmektedir. Bu algılayıcı var olan senaryolarda bir işlev yürütmese de ileride geliştirilecek senaryolar için sisteme dahil edilmiştir.

Robot üzerinde bulunan diğer bir algılayıcı ise srf08 üzerinde bulunan ışık şiddeti algılayıcısıdır. Bu algılayıcı I2C veriyolu üzerinden okunabilmektedir. SRF08, 0-255 aralığında ışık şiddeti ölçümü yapabilmektedir. Üzerinde bulunduğu algılayıcı taretinin hareketi ile 180 derecelik bir açıyı tarayabilen algılayıcı sayesinde ileride eklenebilecek senaryolarda ışığa yönelme ve ışıktan kaçma gibi davranış temelli kontrol uygulamaları yapılabilecektir.

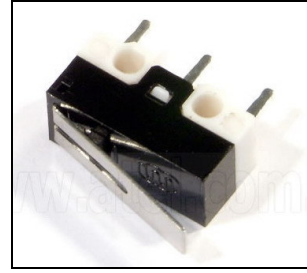
Ana kontrol kartı üzerinde bulunan diğer bir algılayıcı ise dallas one-wire arabirimini kullanan maxim ds18b20 elektronik termometredir (Şekil 6.42). Bu algılayıcı 0.5° hassasiyetle -10° +85° aralığında sıcaklık ölçümü yapabilmektedir. Sıcaklık verileri 9 veya 12 bit olarak one-wire arabirimi aracılığıyla okunabilmektedir. Algılayıcı üzerinde 64bitlik ve değiştirilemeyen bir seri numarası bulunmaktadır. Bu seri numarası sayesinde one-wire arabirimi üzerindeki her bir cihaza ayrı ayrı ulaşılabilir.

Robot üzerinde çarpma algılama amaçlı olarak kullanılan 2 adet mikro dokunma algılayıcısı bulunmaktadır (Şekil 6.43). Bu algılayıcılar robotun ön bölümüne yerleştirilerek herhangi bir çarpma durumunda robotun hareketini durdurmasını

sağlamaktadır. Her bir algılayıcı ana mikroişlemcinin dijital giriş/çıkış portları üzerinden okunmaktadır.



Şekil 6.42 : DS18B20 Dijital Termometre



Şekil 6.43 : Dokunma Algılayıcı Mikro Anahtar

## 6.7 Görüntü ve Ses Sistemleri

Robotun çevresi ile etkileşimi görüntü ve ses sistemleri aracılığıyla sağlanmaktadır. Ana kontrol kartı üzerinde bulunan LCD ekran ile robotun aldığı en son komut ve komut sonucunda oluşan yanıt gösterilmektedir. Ayrıca ana sisteme gönderilen yanıtlar konuşma sentezleyicisi aracılığıyla seslendirilmektedir. Bunların yanı sıra robot üzerinde bulunan kablosuz IP kamera ile ana bilgisayara görüntü geri beslemesi yapılmaktadır.

### 6.7.1 Kablosuz IP Kamera

Ana bilgisayara görüntü aktarımı için Linksys WVC54G kablosuz IP video kamera kullanılmıştır (Şekil 6.44). Bu kamera 802.11g standardını kullanarak 54Mbps hızında 640x480 çözünürlükte mpeg4 görüntü ve ses aktarımı yapabilmektedir. 5v ile çalışan kamera üzerinde ethernet RJ-45 yuvası, harici mikrofon girişi, gösterge ledleri (Ready, Act/Link, Ethernet, Wireless) ve LCD IP adresi göstergesi barındırmaktadır. Statik IP üzerinden çalışan kamera dinamik IP kullanacak şekilde de ayarlanabilmektedir. Hem yapısal hem de ad-hoc ağlara bağlanabilen kamera harici anteni ile geniş bir kapsama alanı sunmaktadır. Desteklediği dinamik DNS sistemi ile otomatik olarak bir internet alanına bağlanarak uzaktan erişime ve web üzerinden yapılandırmaya olanak sağlamaktadır.

Üzerinde bulunan 4.48mm. x F1.5 lens ile 57 derecelik bir görüş açısı sunmaktadır. Robotun üzerindeki tarete monte edilmiş olan kamera, ana bilgisayarda çalışan yazılım aracılığıyla 270°'lik bir açıyı tarayacak şekilde hareket ettirilebilmektedir.

Kameranın görüş açısı ile birleştirildiğinde bu 360 derecelik bir alanın operatör tarafından izlenebilmesini sağlamaktadır.



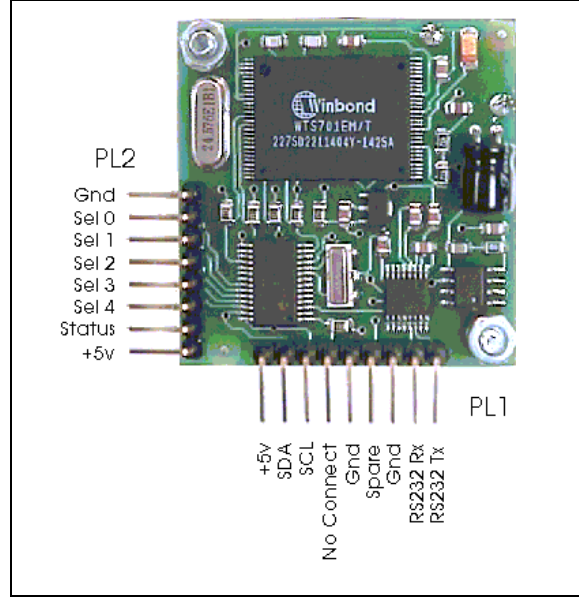
**Şekil 6.44 :** Kablosuz IP Kamera

### **6.7.2 Konuşma Sentezleyici**

Yapay zekaya sahip bir sistemin çevre ile iletişim kurabilmesi önemli gereksinimlerden biridir. Bu nedenle ana bilgisayardan gelen komutlar, ana kontrol kartının yanıtları ve okunan algılayıcı verileri robotun üzerine monte edilmiş olan Devantech “SP03 - Text to Speech Synthesizer” (Şekil 6.45) aracılığıyla konuşmaya çevrilerek çevre ile sesli etkileşim sağlanmaktadır. SP03, ön tanımlı 30 ayrı mesajı konuşabilmekte ayrıca salt metin halinde gönderilen mesajları konuşmaya çevirebilmektedir [52]. Ön tanımlı metinler bilgisayar üzerinden seri port kullanılarak cihaza yüklenmektedir.

SP03 konuşma sentezleyici, üzerinde Winbond WTS701EM/T sentezleyici yongası, PIC 16F872 mikro işlemci ve dahili 40mm’lik hoparlör barındırmaktadır (Şekil 6.46 ve Şekil 6.47). Winbond sentezleyicinin ses çıkışı 325 miliwatt’lık LM386 amplifier ile güçlendirilerek hoparlöre aktarılmaktadır [52].

Cihaz I2C veya RS232 veri yolu üzerinden kontrol edilebilmektedir. Ayrıca üzerindeki sel0-sel4 paralel bağlantı noktasına (Şekil 6.45) gönderilen 5 bitlik veri ile ön tanımlı 30 mesajın seslendirilmesi sağlanabilmektedir. Bu özellik RS232 ve I2C haberleşmesini desteklemeyen işlemciler için büyük kolaylık sağlamaktadır.

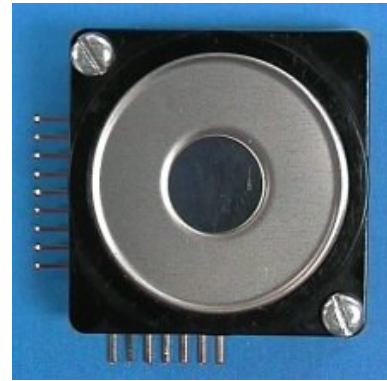


**Şekil 6.45 : SP03 Bağlantı Şeması**

RS232 bağlantısı kullanıldığında 38400 baud⁵¹ hızında iletişim yapabilmektedir. Seri iletişim protokolü 8N2-34000 (8bit, paritesiz, 2 stop bit, 38400 baud) yapılandırmasında çalışmaktadır. Seri iletişimde konuşulacak metin gönderilmeden önce 0x80 komutu ve 3 kontrol byte'ı gönderilmektedir. Bu kontrol byte'ları ses yüksekliği, karakter yoğunluğu ve konuşma hızlarını belirtmektedir.



**Şekil 6.46 : SP03 Yonga Yüzü**



**Şekil 6.47 : SP03 Hoparlör Yüzü**

I2C modunda çalışırken cihaz 2 yazmaç aracılığıyla kontrol edilmektedir. 0x00 adresinde komut yazmaç 0x01adresinde ise yazılım versiyon numarası bulunmaktadır. Sp03'ün I2C adresi ise 0xC4 şeklindedir.

Komut yazmaçının kabul ettiği 32 ayrı komut bulunmaktadır. Bunlar Tablo 6.5'de listelenmiştir. NOP komutu sentezleyicini tampon belleğine konuşulacak metin kayıt

⁵¹ Bilgi iletim hız birimi, baud

edilirken kullanılmaktadır. İlk olarak sp03'un I2C adresi veriyoluna yazılmakta ardından komut yazmacının adresi gönderilmektedir. Yazmaç adresini takiben NOP komutu, kontrol byte'ları ve konuşulacak metin sp03'e gönderilir. Gönderilen metnin tamamlandığını belirtmek için sonuna NOP komutu eklenmektedir. SP03'un tampon belleği 85 byte uzunluğundadır. Bu 85 byte'ın 4 byte'ı ses yüksekliği, karakter yoğunluğu ve konuşma hızı kontrol byteları tarafından kullanılmaktadır. Geri kalan 81 bytelık tampon bellek konuşulacak metin için ayrılmıştır.

Tampon belleğe metin aktarıldıktan sonra ikinci bir I2C haberleşmesi başlatılarak I2C adresi, yazmaç adresi ve SPKBUF komutu gönderilir. SPKBUF komutunun ardından SP03 tampon bellekteki metni konuşmaya çevirir. Her bir konuşmanın ardından tampon bellek mikroişlemci tarafından temizlenmektedir. SPKPRE komutları ise ön tanımlı metinlerin konuşulmasını sağlamak için kullanılmaktadır. SP03'un mikro işlemci tarafından kontrolü ile ilgili yazılım detaylarına bölüm 7'de değinilecektir.

**Tablo 6.5 : Sp03 Yazmaç Komutları**

Komut	İşlev
NOP (0x00)	Başlangıç bitiş
SPKPRE 1 - 30, veya 0x01 - 0x1E	Ön tanımlı metni konuş
SPKBUF 64 veya 0x40	Tampon bellekteki metni konu.

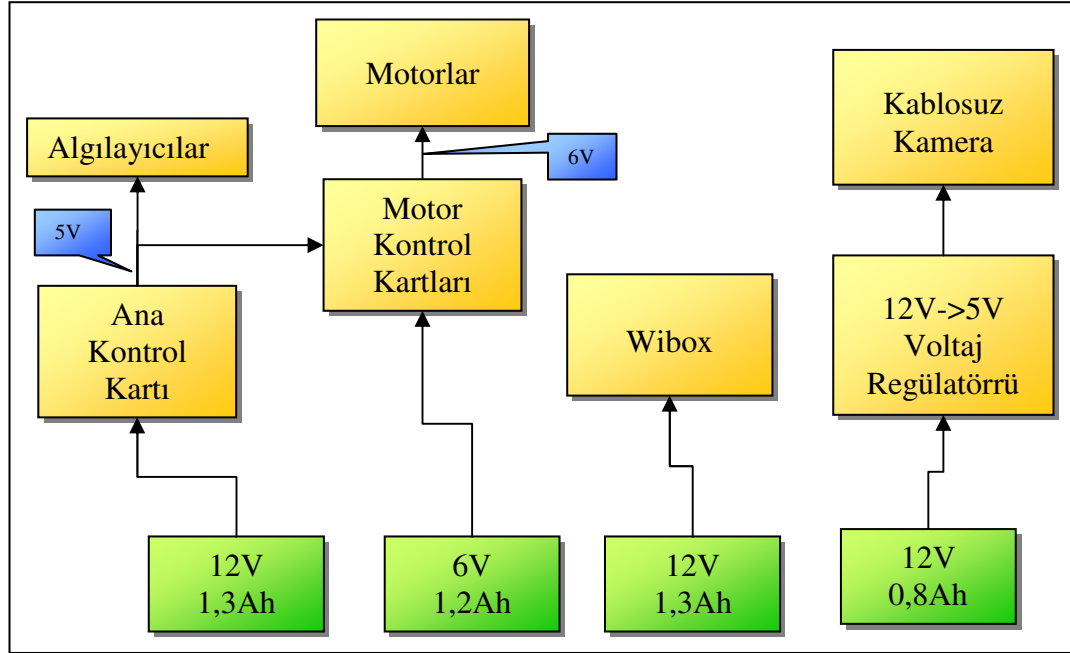
## 6.8 Diğer Donanımlar

Önceki bölümlerde değinilen alt sistemlerin donanımları yanı sıra bu sistemleri destekleyen veya kullanımı isteğe bağlı olan elektronik ve mekanik donanımlar bulunmaktadır.

Bu donanımların başında güç kontrol ünitesi, voltaj regülatörleri ve aküler gelmektedir. Güç kontrol ünitesi üzerinde barındırdığı anahtarlar yardımıyla her bir donanıma giden akımın açılıp kapatılabilmesini sağlamaktadır.

Ana kontrol kartı 12 voltluk 1,3 amper/saatlik bir akü ile beslenmektedir. Tüm algılayıcıların ve motor kontrol kartlarının entegre bölümlerinin beslenmesi ana kontrol kartı üzerinden yapılmaktadır. Motorlar ise 6 volt 1,2 Amper/saatlik bir aküden güç almaktadır. Wiport modülü ve kablosuz IP kameraya ise 2 ayrı 12 voltluk akü ile güç sağlanmaktadır. Kamera 5v ile çalıştığı için harici bir güç

regülatörü robota dahil edilmiştir. Robotun barındırdığı güç sistemlerinin ilişkileri Şekil 6.48’de şematik olarak gösterilmiştir.



Şekil 6.48 : Güç Sistemlerinin Yapısı

Güç kontrol kartının yanı sıra robot üzerinde bulunan I2C veri yolunu kullanan 6 ayrı cihazın bağlandığı bir I2C bağlantı kartı yapılarak sisteme dahil edilmiştir. Mevcut donanımlara ek olarak irda protokolünü kullanarak haberleşme sağlayan Mikroelektronika IrDA Board kartı, 220 volta kadar anahtarlama yapabilen 3 kanallı röle kartı, I2C üzerinden kontrol edilebilen 20x4 LCD ekran ve 3x4 nümerik tuş takımı sisteme istenildiğinde eklenebilmektedir.

Üzerinde diferansiyel sürüş sistemi ve redüktör dişli kutusunu barındıran robot bazasına diğer elektronik ve elektro mekanik donanımlar Meccano metal konstrüksiyon sisteminin parçaları kullanarak monte edilmiştir. Kullanılan hafif metal alaşım, tüm donanımın ağırlığını taşımakta ve robotun iskeletini oluşturmaktadır.

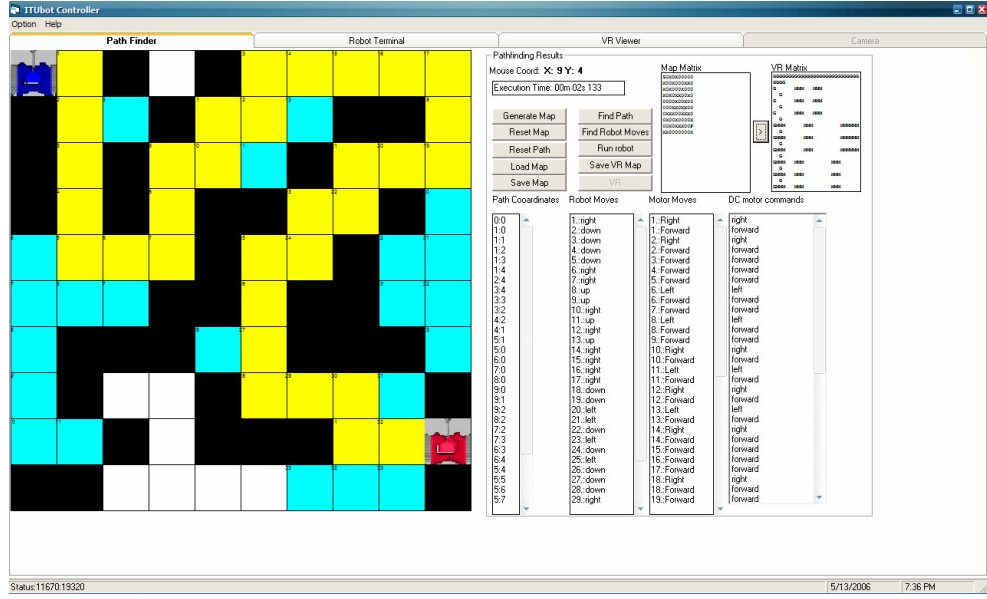
## 7 YAZILIM

Robotun navigasyonel faaliyetlerinin düzenlenmesi, algılayıcı verilerinin okunması ve değerlendirilmesi, hareket mekanizmalarının çalıştırılması ve navigasyon sisteminin ürettiği sonuçların işleme koyulması gibi tüm işlevler ana bilgisayar ve ana kontrol kartı üzerinde çalışan yazılımlar aracılığıyla gerçekleştirilmektedir. Ana bilgisayar üzerinde çalışan Microsoft Visual Basic uygulaması robotun strateji, taktik ve uygulama katmanlarını barındırmaktadır. Uygulama katmanının bir bölümü ise mikroişlemci üzerinde çalışan MikroBasic programı tarafından yürütülmektedir. Uygulama katmanının bu iki bölümü haberleşme sistemi üzerinden komutları paylaşmaktadır. Ana bilgisayar üzerinde çalışan uygulamanın detayları bölüm 7.1 mikro işlemci yazılımının detayları ise bölüm 7.2’de incelenecektir.

### 7.1 Ana Bilgisayar Yazılımı

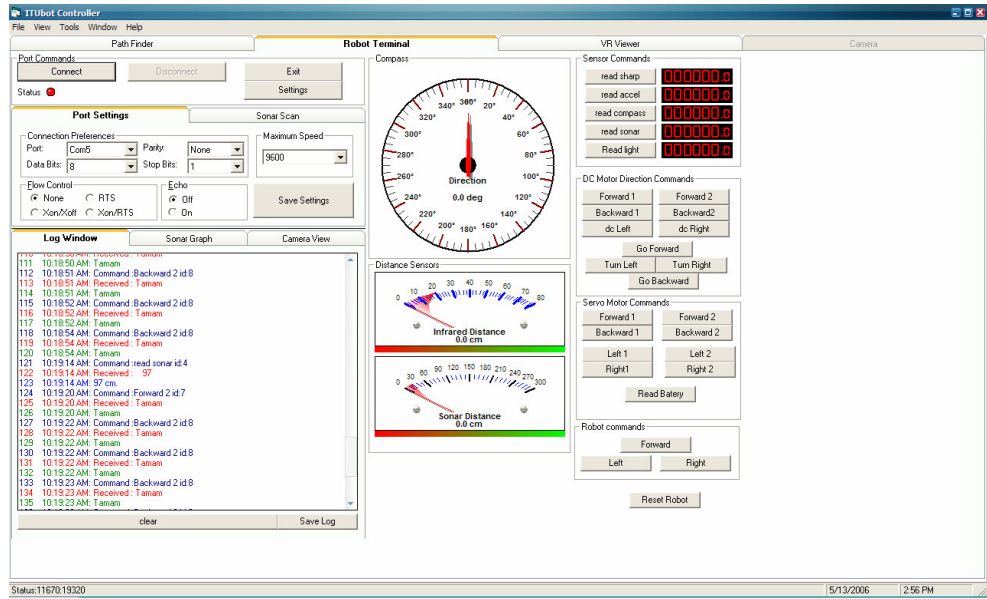
Microsoft Visual Basic 6.0 programlama dili kullanılarak geliştirilen bu yazılım mantıksal 3 katmanı barındırmaktadır. Yol bulma, yol planlama, yolun izlenmesi için gerekli işlemlerin hesaplanması ve bu işlemlerin uygulanması bu programın sorumluluğundadır. Programın başka projelerde de kullanılabilmesi amacıyla program arayüzü İngilizce olarak tasarlanmıştır. Program 3 ana form ve yardımcı form ve modüllerden oluşmaktadır. Programın kaynak kodu Ek-A’da verilmiştir

Pathfinder Ekranı (Şekil 7.1) robotun strateji ve taktik katmanlarını barındırmaktadır. Operatör tarafından haritanın oluşturulması, kaydedilmesi ve geri yüklenmesi, harita üzerinde yol bulma, rota planlama, engel algılama, engelden sakınma ve çarpışma algılama işlemleri bu form üzerinde yapılmaktadır. Taktik katmanında oluşturulan komutlar uygulama katmanını barındıran Robot terminal ekranı üzerinden yürütülmektedir. Bu ekran üzerinde sol tarafta oluşturulan harita sağ tarafta ise harita üzerinde bulunan yolun koordinatları, rota bilgileri ve rotanın izlenmesi için gerekli olan motor ve uygulama komutları listelenmektedir. Programın modüler yapısı ileride eklenebilecek senaryo ve kontrol yöntemleri için esneklik sağlamaktadır.



Şekil 7.1 : Pathfinder Formu

Robot terminal ekranı (Şekil 7.2) robot ile olan haberleşmenin kontrolü ve kaydı işlevlerinin yanı sıra mikro işlemci tarafında çalışan uygulama katmanı komutlarının tetiklenebildiği arabirimi içerir. Ayrıca her bir algılayıcının en son okunan değerlerinin bulunduğu göstergeler, sonar ve kızılötesi tarama sonuçları ile kamerada bilgisayara iletilen görüntüler bu ekran üzerinden takip edilmektedir. Robot terminal formunun kod kısmı projenin uygulama katmanı için kullanılan kodun büyük bir bölümünü içermektedir.



Şekil 7.2 : Robot Kontrol Terminali

VR Viewer Ekranı pathfinder ekranında oluşturulan haritanın 3 boyutlu gösterimini barındırmaktadır. Kullanıcı bu ekran içerisinde oluşturulan 3 boyutlu gösterim içerisinde robotun sanal olarak hareket etmesini sağlayabilmektedir.

### 7.1.1 Strateji ve Taktik Katmanı

Pathfinder formu tüm strateji ve taktik katmanı işlevlerinin gerçekleştirildiği formdur. Bu işlevlerin yanı sıra kullanıcı görsel işlemler ve harita tanımlamalarının yapılması ile ilgili prosedürleri taşımaktadır. Form yüklendiğinde ilk olarak harita boyutu ve hücre büyüklük bilgileri windows registry kaydından okunur. Ekran harita bu bilgiler doğrultusunda çizilmektedir. Menüler yardımıyla haritanın ebadı ve hücre büyüklüğü değiştirilebilmektedir. Ardından kullanıcının haritayı çizmesi veya önceden kaydedilmiş bir haritayı yüklemesi beklenir. Programın rastlantısal bir harita hazırlama özelliği de bulunmaktadır. (Program senaryo 3⁵² için çalıştırıldığında ise belleğe boş bir harita yüklenmektedir. ) Senaryo 1’de kullanıcının başlangıç ve bitiş hücrelerini belirlemesi gerekmektedir. Senaryo 2 için ise uygulama katmanı çalıştırılarak robotun harita üzerindeki yeri belirlenmeye çalışılır. Bu işlem için robotun çevresindeki 4 uzaklık algılayıcı okunarak 3x3 boyutlarında bir matris oluşturulur. Ardından bu matris harita matrisi taranarak eşleştirilmeye çalışılır. Matris eşleşmesi gerçekleştiğinde harita içerisindeki başlangıç koordinatı da belirlenmiş olacaktır. Böylece yolun hesaplanması için gerekli olan 3 bilgi (başlangıç noktası, bitiş noktası, harita matrisi) elde edilmiş olmaktadır.

Ardından harita üzerinde yolun hesaplanma işlemine geçilir. Yolun hesaplanmasında kullanılan harita matrisi  $A \times B$  ebadındaki bir harita için  $A \times B$  büyüklüğünde bir matristen oluşmaktadır. Bu matrisin indisleri her bir hücrenin harita üzerindeki x ve y koordinatlarını belirtmektedir. Başlangıç hücresi bu dizide “1”, bitiş hücresi ise “2” ile belirtilmektedir. Harita üzerinde geçilebilir hücreler “0” geçilemeyen hücreler ise “9” değeri ile matris üzerinde işaretlenmektedir. Yolun hesaplanması sırasında elde edilen maliyet değerleri geçilebilir hücrelerin değerlerine eklenmektedir. Ön tanımlı değerler ile çakışma olmaması için maliyet değerler “100+maliyet” şeklinde matrisin elemanlarına aktarılmaktadır. Harita matrisi aynı zamanda kullanıcının daha rahat okuyabilmesi için pathfinder ekranında farklı bir gösterimle belirtilmektedir. Bu gösterimde boş hücreler 0 dolu hücreler X, başlangıç hücresi S, bitiş hücresi F harfleri ile ilişkilendirilmiştir.

---

⁵² Haritanın bilinmediği senaryo.

Yol bulma işleminin başında toplam maliyet değişkeni “Tim” sıfıra eşitlenir bu değişken her bir hücrenin maliyet değerinin hesaplanmasının yanı sıra bir sonraki adımda işleme alınacak hücrelerin belirlenmesi için de kullanılmaktadır. Yol bulma algoritmasında kullanılan diğer önemli bir değişken ise “true” veya “false” değerini alabilen mantıksal “reachable” değişkenidir. Yol bulma algoritma iki ana döngüden oluşmaktadır. Birinci döngüde yayılma işlemi gerçekleşmekte ve hedefe ulaşıp ulaşılamayacağı anlaşılmaktadır. İkinci döngü ise hedefe ulaşılabilirliği anlaşıldıktan sonra yol koordinatlarının belirlenmesi için kullanılmaktadır.

Birinci döngüde her bir hücrenin maliyet değeri hesaplanmaktadır. Bu döngü “Do While reachable = False” yapısı kullanılarak gerçekleştirilmiştir. Bu döngü içerisinde ilk olarak başlangıç hücresinin değeri “100+tim” olarak atanır ve bu hücrenin komşu hücreleri (alt, üst, sağ ve sol hücreler) taranır. Bu hücrelerin hedef hücre olup olmadığı kontrol edilir. Eğer hedef hücreye ulaşılmış ise “reachable” değerine true atanarak döngü sonlandırılır. Hedefe ulaşılmamış ise döngüye devam edilir. Döngünün bir sonraki adımına geçilmeden önce matrisin tüm elemanlarının değerleri toplanarak “sum” değişkenine yazılır. Bu değişken her adımda bir önceki adımın toplamı ile karşılaştırılmaktadır. Yol bulma algoritması “waterfall⁵³” yaklaşımını kullandığı için bir önceki adıma göre yayılma olup olmadığının kontrolü büyük önem taşımaktadır. Eğer bir önceki adımda işlenen hücrelerden daha fazla yayılma yapılamamış ise robotun geçebileceği tüm hücrelerin taranmış olduğu anlaşılır ve yol bulma yordamından çıkılır. Böyle bir durum ancak hedefe ulaşmanın imkansız olduğu (örneğin robotun çevresinin tamamen geçilemez hücrelerle çevrili olduğu durum) bir haritada meydana gelmektedir. İşlenen hücrelerin kullanıcı tarafından izlenebilmesi için bu hücreler mavi renk ile işaretlenmektedir (bkz. Şekil 7.1).

Döngünün her adımında tim değeri 1 artırılmakta ardından değeri “tim” değişkenine eşit olan hücrelerin komşu hücreleri için kontrol ve işaretleme işlemi gerçekleştirilmektedir. Bu işlemler hedef hücreye ulaşılan (reachable=true) veya yayılacak başka hücre kalmayınca (lastsum=sum) kadar devam eder. Birinci döngünün başarı ile sonlanması (Reachable=true) hedefe ulaşıldığının göstergesidir. Döngü sonlandığında ızgara ekrana yeniden çizilir ve hedefe ulaşıldığı bir mesajla kullanıcıya bildirilir. İlk döngünün Visual Basic’te gerçekleşmesi **Tablo 7.1**’de gösterilmiştir.

---

⁵³ İng:Su akışı

**Tablo 7.1 : Yol Bulma Algoritması**

```
tim = 0
pos (sx, sy) = 100 + tim
reachable = False
Do While reachable = False
For j = 0 To Up - 1
  For i = 0 To Across - 1
    DoEvents
    If pos (i, j) = 100 + tim Then
      If i < Across - 1 Then
        If pos (i + 1, j) = 0 Then
          pos (i + 1, j) = 100 + tim + 1
          frmPathFinder. Line ( (i + 1) * cz, j * cz) - ( (i + 1) * cz + cz, j * cz + cz), vbCyan, BF
          ElseIf pos (i + 1, j) = 2 Then
            'If the position is no 0 but its 2 (FINISH) then Reachable = true!!! We found end
            reachable = True
            End If
          End If
        If i > 0 Then
          If pos ( (i - 1), j) = 0 Then
            pos (i - 1, j) = 100 + tim + 1
            frmPathFinder. Line ( (i - 1) * cz, j * cz) - ( (i - 1) * cz + cz, j * cz + cz), vbCyan, BF
            ElseIf pos (i - 1, j) = 2 Then
              reachable = True
              End If
            End If
          If j < Up - 1 Then
            If pos (i, j + 1) = 0 Then
              pos (i, j + 1) = 100 + tim + 1
              frmPathFinder. Line (i * cz, (j + 1) * cz) - (i * cz + cz, (j + 1) * cz + cz), vbCyan, BF
              ElseIf pos (i, j + 1) = 2 Then
                reachable = True
                End If
              End If
            If j > 0 Then
              If pos (i, j - 1) = 0 Then
                pos (i, j - 1) = 100 + tim + 1
                frmPathFinder. Line (i * cz, (j - 1) * cz) - (i * cz + cz, (j - 1) * cz + cz), vbCyan, BF
                Load lb (lb. UBound + 1)
                ElseIf pos (i, j - 1) = 2 Then
                  reachable = True
                  End If
                End If
              End If
            End If
          Next i
        Next j
      If reachable = False Then
        Sum = 0
        For j = 0 To Up - 1
          For i = 0 To Across - 1
            Sum = Sum + pos (i, j)
          Next i
        Next j
        If Sum = LastSum Then
          DrawGrid
          MsgBox "Destination not reachable"
          Exit Sub
        Else
          LastSum = Sum
        End If
      End If
      tim = tim + 1
      If ShowMsgBox = True Then
        MsgBox "Destination not found yet" & vbCrLf & "Expand it to each direction if possible. " & vbCrLf & " Then change the counter tim to point to all next squares. They are to be expanded next"
      End If
    Loop
```

Bu döngünün ardından bitiş noktasından başlangıç noktasına doğru çalışan ikinci döngü devreye girer. İlk olarak bitiş noktasının komşu hücreleri taranır ve en düşük maliyete sahip olan hücre yol koordinatlarını tutan 2 boyutlu “pathcoord” dizisine

eklenir. Eklenen her hücre ekrandaki haritada sarı renk ile işaretlenmektedir (bkz. Şekil 7.1). Ardından bu hücrenin komşularının taranmasına geçilir. Bu döngü başlangıç hücresine ulaşıncaya kadar devam eder. Her bir adımda en düşük maliyetli hücrenin seçilmesi en kısa yolun bulunmasını garantilemiş olmaktadır. İkinci döngünün kaynak kodu Tablo 7.2’de verilmiştir.

**Tablo 7.2 : Yol Koordinatlarının Bulunması**

```

Do While LastX <> sx Or LastY <> sy
  DoEvents
  tim = tim - 1
  did = False
  If LastX < Across Then
    If pos (LastX + 1, LastY) = 100 + tim Then
      LastX = LastX + 1
      frmPathFinder. Line (LastX * cz, LastY * cz)- (LastX * cz + cz, LastY * cz + cz), vbYellow, BF
      did = True
      path = "(" & LastX & ", " & LastY & "):" & path
      pathcord (tim, 0) = LastX
      pathcord (tim, 1) = LastY

    End If
  End If

  If did = False Then
    If LastX > 0 Then
      If pos (LastX - 1, LastY) = 100 + tim Then
        LastX = LastX - 1
        frmPathFinder. Line (LastX * cz, LastY * cz)- (LastX * cz + cz, LastY * cz + cz), vbYellow, BF
        did = True
        path = "(" & LastX & ", " & LastY & "):" & path
        pathcord (tim, 0) = LastX
        pathcord (tim, 1) = LastY
      End If
    End If

    If did = False Then
      If LastY < Up Then
        If pos (LastX, LastY + 1) = 100 + tim Then
          LastY = LastY + 1
          frmPathFinder. Line (LastX * cz, LastY * cz)- (LastX * cz + cz, LastY * cz + cz), vbYellow, BF
          did = True
          path = "(" & LastX & ", " & LastY & "):" & path
          pathcord (tim, 0) = LastX
          pathcord (tim, 1) = LastY
        End If
      End If

      If did = False Then
        If LastY > 0 Then
          If pos (LastX, LastY - 1) = 100 + tim Then
            LastY = LastY - 1
            frmPathFinder. Line (LastX * cz, LastY * cz)- (LastX * cz + cz, LastY * cz + cz), vbYellow, BF
            path = "(" & LastX & ", " & LastY & "):" & path
            pathcord (tim, 0) = LastX
            pathcord (tim, 1) = LastY
          End If
        End If
      End If
    End If
  End If
Loop

```

Yol bulma işlemi tamamlandıktan sonra yol koordinatları pathfinder ekranındaki “path coordinates” kutusunda listelenir.

Yol koordinatlarının bulunmasının ardından taktik katmanı devreye girer. Bulunan yol koordinatlarının uygulama katmanı tarafından işlenebilmesi için alt seviye komutlara çevrilmesi gerekmektedir. İlk olarak yol üzerindeki her bir hücre başlangıçtan bitişe doğru taranarak bir önceki hücreye göre konumu belirlenir. (alt, üst, sağ, sol). Yol üzerindeki hücrelerin birbirine göre konumları “robot moves” listesi kullanılarak ekranda görüntülenmektedir. Bu liste robotun harita üzerinde ilerlerken bulunduğu hücreden diğer hücreye giderken hangi doğrultuda hareket edeceğini belirtmektedir. Örneğin “pathcoord” matrisindeki bir sonraki hücre robotun bulunduğu hücrenin sağında ise robotun sağ hücreye geçmesi gerekecektir. Ancak hücrelerin birbirine göre konumlarının bilinmesi robotun bir hücreden diğerine geçmesi için yeterli değildir. Robotun yönü de hareketin hesaplanması için önemlidir. Örneğin robotun önü haritanın kuzeyine (üst) doğru iken alttaki hücreye geçmesi için ilk olarak yönünü güneye çevirmesi gerekmektedir. Bu aşamada taktik katmanının “find motor moves” yordamı devreye girer. Bu yordam “robotmoves” listesini adım adım tarayarak her bir işlemin gerçekleşmesi için gerekli olan hareketleri hesaplar. Bir önceki örnekte verilen işlemin gerçekleşmesi için robot iki kez sola dönmeli ardından bir kare ilerlemelidir. “find motor moves” yordamı oluşturduğu bu hareket taktiğini ekranda “motor moves” listesinde görüntülemektedir. Bu listenin uzunluğu “robot moves” listesinin uzunluğundan büyük veya eşittir. Bu işlemin ardından uygulama katmanına aktarılacak olan “dc motor commands” listesi oluşturulur. Bu liste uygulama katmanı tarafından adım adım okunacak ve alt donanım komutlarına çevrilerek uygulanacaktır.

Robotun 3. senaryo için çalışması ise senaryo 1 ve senaryo 2 den farklılık göstermektedir. Senaryo 3’te harita bilinmediği için robotun kesin bir yol hesaplaması mümkün değildir. Bu zorluğun aşılması için iki yöntem öngörülmüştür. Bunlardan birincisi robotun tüm haritayı öğrenmesine dayanırken diğeri hedefe en kısa sürede ulaşmasını amaçlamaktadır. Birinci yöntemde başlangıç ve bitiş hücreleri dışında tüm hücreler geçilemez olarak kabul edilmektedir. Robot üzerindeki 4 uzaklık algılayıcısını çalıştırarak komşu hücrelerinin dolu veya boş olduğunu algılamakta ve bu bilgiyi harita üzerinde işaretlemektedir. Ardından boş hücrelerden birini seçip bu hücreye ilerlemekte ve ilk adımdaki şekilde komşu hücrelerin değerlerini haritaya işlemektedir. Bu işlem hedef hücreye ulaşılan veya robotun ziyaret etmediği boş hücre kalmayınca kadar devam etmektedir. Ancak bu işlem

harita üzerinde hücrelerin çoğunun robot tarafından ziyaret edilmesini gerektirdiği için uzun zaman almakta ve hedefe ulaşılmasını zorlaştırmaktadır.

İkinci yöntem ise harita üzerindeki hücrelerin tamamının geçilebilir (boş) olduğu varsayımına dayanmaktadır. Robot ilk olarak boş harita üzerinde başlangıç noktasından bitiş noktasına bir yol hesaplamaktadır. Ardından her bir adımda çevresindeki hücreleri algılayıcıları ile tarayarak harita üzerinde işaretlemektedir. Eğer ilk belirlediği yol üzerinde geçilemeyen bir hücreye rastlarsa, robot güncellenmiş harita üzerinde ikinci bir yol hesaplamaktadır. Böylece önüne çıkan bir engelden sakınarak en kısa sürede hedefe ulaşması sağlanmaktadır. Bu yöntem haritanın tamamının öğrenilmesine gerek duymadığı için birinci yöneme göre daha avantajlı olmaktadır.

### **7.1.2 Uygulama Katmanı**

Bu katman taktik katmanının belirlediği komutları yerine getirmekle yükümlüdür. Bu komutlar bir yolun izlenmesi veya algılayıcı verilerinin okunarak robotun komşu hücrelerinin taranması şeklinde olabilmektedir.

Robot Terminal formu uygulama katmanının işlevlerinin gerçekleştirildiği formdur. Bu form aynı zamanda robotun operatör tarafından yönetilmesini de sağlamaktadır. Operatör kontrolünde stratejik ve taktiksel kararların operatör tarafından alındığı kabulü yapılmaktadır. Robot terminal formu aynı zamanda haberleşme yordamlarını ve algılayıcı verilerinin görüntülenmesi işlevini de yerine getirmektedir. Form üzerinde mikro işlemci üzerinde çalışan her bir alt seviye komutun manuel olarak çalıştırılmasını sağlayan komut butonları ve robot üzerindeki algılayıcıların her birinin değerlerinin gösterildiği göstergeler bulunmaktadır. Ayrıca robot ile haberleşmede kullanılan com portun ayarlarının yapılması, sonar ve kızıl ötesi taramaların gerçekleştirilmesi ve kamera görüntüsünü izlenmesi için arabirimler barındırmaktadır.

Ana bilgisayar ve ana kontrol kartının haberleşmesini sağlayan haberleşme sistemi robotun çalışmasında en büyük önemi taşımaktadır. Haberleşme Lantronics firmasının “com port redirector” yazılımının oluşturduğu sanal haberleşme portu üzerinden gerçekleştirilmektedir. Uygulama katmanı bu sanal com portu fiziksel bir port olarak görmektedir. Bu porta erişim Microsoft firmasının mscomm.ocx nesnesi

üzerinden yapılmaktadır. Mscomm.ocx nesnesi bir porta bağlanma, haberleşme ve bağlantıyı kesme ile ilgili alt yordamları barındırmaktadır.

Robot terminal ekranında bulunan “port settings” sekmesi bu ayarlarının yapılması için gerekli olan arabirimi sağlamaktadır. “Port Commands” kutusu ise bağlantının başlatılması ve kesilmesi ile ilgili komut butonlarını içermektedir. Haberleşme sırasında robota gönderilen komutlar ve robotun bu komutlara verdiği yanıtlar “log window” penceresinde listelenmekte istenirse kayıt edilmesi sağlanmaktadır. Bu kayıt özellikle robotun hareketlerinde oluşabilecek hataların giderilmesi aşamasında büyük bir kolaylık sağlamaktadır.

Uygulama katmanını 3 başlık altında incelemek mümkündür. Bunlardan birincisi taktik katmanının oluşturduğu makro donanım komutlarının uygulanması için donanım komutlarının belirlenmesi, donanım komutlarının ana kontrol kartına gönderilmesi ve hata düzeltme işlemleridir.

Taktik katmanının oluşturduğu 4 adet makro komut bulunmaktadır. Bunlar çevre hücreleri algıla ve sağa dön, sola dön ve ilerle şeklindedir. Bu komutlar robot terminal ekranında bulunan “robot commands” bölümünde manuel olarak çalıştırılabilmektedir. Uygulama katmanının makro komutları birçok donanım komutunun birleşiminden oluşmaktadır. Makro komutların her birinin barındırdığı donanım işlevlerinin incelenmesi uygulama katmanının çalışmasının anlaşılması için önem taşımaktadır.

“Çevre hücreleri algıla” komutu robotun üzerindeki uzaklık algılayıcıların okunması ve bu veriler ile 3x3 boyutunda bir çevre haritasının oluşturulması işlemlerinden oluşmaktadır. Algılayıcıların yapısından kaynaklanan hataların giderilebilmesi için kızılötesi ve sonar uzaklık algılayıcıları birlikte kullanılmaktadır. İlk olarak 4 kızıl ötesi algılayıcı okunmakta ardından ön algılayıcı taretleri 0, 90 ve 180 derece konumlarına getirilerek sonar okumaları yapılmaktadır. Tüm okumalar birleştirilip çevre hücrelerin dolu veya boş olduğuna karar verilmektedir. Bu kararların ardından 3x3'lük çevre haritası oluşturularak taktik katmanına aktarılmaktadır. Bu komutu oluşturan donanım komutlarının işlem sırası Tablo 7.3’de listelenmiştir.

**Tablo 7.3 : Çevre Algıla Komutu Pseudo Kodu**

```
Read sharp1
Read Sharp2
Read Sharp3
Read Sharp 4
ServoLeftSweep1
Read_Sonar1
Center1
ReadSonar1
ServoRightSweep1
ReadSonar1
```

Sağa ve sola dön komutları robotun bulunduğu hücre içerisinde yönü değiştirmesi için kullanılan komutlardır. Bu komutlar robotun yol üzerinde ilerlerken bir sonraki hücrenin robotun doğrultusundan farklı bir doğrultuda olması durumunda taktik katmanı tarafından “dc motor moves” listesine eklenmektedir. Robotun belirlenen yol üzerinde ilerlerken konum hatalarının en aza indirilebilmesi için dönüşlerin hassas bir şekilde gerçekleştirilmesi gerekmektedir. Bu nedenle dönüş komutları hata açısal düzeltme algoritmalarını barındırmaktadır. İlk olarak robota “read_compass” komutu gönderilerek pusuladan harekete başlamadan önceki doğrultusu okunmaktadır. Ardından dönüş komutu çalıştırılmaktadır. Robot dönüş tamamlandı bilgisini gönderdiğinde pusula tekrar okunarak yapılan dönüş hareketinin başarısı ölçülmektedir. Eğer dönüş hareketi 90 derecelik bir dönüş olarak gerçekleşmiş ise işlem sona ermektedir. Ancak robot üzerinde bulunan motorlar, tekerler ve zeminin karakteristiğinden kaynaklanan açısal hatalar sistematik olarak veya rastlantısal olarak oluşabilmektedir. Bu durumda uygulama katmanı açısal hatanın yönüne göre robota “dc_left” ve “dc_right” komutlarını göndererek açısal hatayı düzeltmeye çalışmaktadır. Her bir düzeltme girişiminin ardından pusula tekrar okunarak açısal hatanın kabul edilebilir düzeye indirilmesine çalışılmaktadır. Bu projede açısal hata payı pusulanın en düşük ölçüm seviyesi olan 0.1 derece olarak alınmıştır. Dönüş komutlarının çalışmasını özetleyen pseudo kodu Tablo 7.4’de verilmiştir

**Tablo 7.4 : Dönüş Komutu Pseudo Kodu**

```
İlk açı= Read_compass
Başarı=false
Left/Right
Do while başarı=False
    Son açı= Read_compass
    Eğer (son açı) farklı (ilk açı+90) ise {
        Eğer Son Aç1 >ilk Aç1 +90 ise { DC_Right / Dc_Left }
        Eğer Son Aç1 <ilk Aç1 +90 ise { Dc_Left / DC_Right }
    }
    Eşit ise
    {
        Başarı=True
    }
Loop
```

İlerle komutu robotun bulunduğu hücreden bir sonraki hücreye doğru ilerlemesi için kullanılan komuttur. Bu komut robotun yol üzerinde ilerlerken bir sonraki hücrenin robotun doğrultusuyla aynı doğrultuda olması durumunda veya robotun dönüş komutu ile bu doğrultuya dönmesi sağlandıktan sonra taktik katmanı tarafından “dc motor moves” listesine eklenmektedir. Robotun belirlenen yol üzerinde ilerlerken konum hatalarının en aza indirilebilmesi için ilerleme sırasında açısal bir hatanın oluşmaması gerekmektedir. Bunun yanı sıra robotun ilerlemeden önce önünde yeterli boşluğun bulunduğundan emin olması gerekmektedir. Bu nedenle ilerle komutu uygulama katmanı tarafından uygulanırken ilk olarak ön uzaklık algılayıcıları okunmakta robotun önünde bir engel olup olmadığı kontrol edilmektedir. Engel yoksa robota “read_compass” komutu gönderilerek yönelim bilgisi okunmakta ardından “forward” komutu gönderilerek robotun bir kare ilerlemesi sağlanmaktadır. İlerleme işleminin tamamlanmasından sonra tekrar pusula okunarak açısal bir hatanın varlığı kontrol edilmektedir. Bir hata oluşması durumunda dönüş komutlarında olduğu gibi robotun doğrultusu “dc_left” ve “dc_right” komutları kullanılarak düzeltilmektedir. İlerle komutunun pseudo kodu Tablo 7.5’de verilmiştir.

**Tablo 7.5 : İlerle Komutu Pseudo Kodu**

```
İlk açı= Read_compass
Başarı=false
Ön mesafe=read_sharp1
Eğer ön mesafe>= Hücre boyutu{
    Forward
    Do while başarı=False
        Son açı= Read_compass
        Eğer son açı farklı ilk açı ise {
            Eğer Son Açı >ilk Açı ise { DC_Right / Dc_Left }
            Eğer Son Açı <ilk Açı ise { Dc_Left / DC_Right }
        }
        Eşit ise
        {
            Başarı=True
        }
    Loop
}
```

Makro komutları oluşturan her bir donanım komutu uygulama katmanı tarafından sanal com porta gönderilmekte ve ana kontrol kartı üzerinde çalışan program tarafından işlenmektedir. Bu donanım komutlarının her biri 1 baytlık veri oluşturacak şekilde kodlanmış indekslere sahiptir. Bu indeksler aynı zamanda robot terminal ekranında bulunan komut butonlarının dizi indeksleri ile eşleştirilmiştir. Uygulama katmanı makro komutları uygularken “send_command” yordamını ilgili indeksi parametre olarak göndererek çalıştırmaktadır. Bu yordam ilk olarak “mscomm” nesnesi üzerinden portun açık olup olmadığını kontrol ettikten sonra indeks değerini ASCII koduna çevirerek sanal port üzerinden ana kontrol kartına göndermektedir. Gönderme işleminin ardından “lastcommand” değişkeni yanıtın işlenebilmesi için indeks değerine eşitlenmektedir. Beklenen yanıtın uzunluğu “mscomm” nesnesinin “RTtreshold” ve “Inputlen” özelliklerine aktarılarak 5 baytlık veri alındığında “mscomm” nesnesinin yanıt yorumlayıcı yordamı tetiklemesi sağlanmaktadır. Ana kontrol kartında gönderilen komutun işlenmesi ve yanıt gönderilmesi ile ilgili detaylar bir sonraki bölümde ele alınacaktır. Hata ayıklamayı kolaylaştırmak için gönderilen komut “log window” penceresine yazılmaktadır. Bu işlemlerin detaylı kaynak kodu Tablo 7.6’da verilmiştir. Komutu gönderiminin ardından “mscomm” nesnesi ana kontrol kartından gelecek yanıtı beklemeye başlar.

**Tablo 7.6 : Komut Gönderme Yordamı**

```
Public Sub send_command (Index As Integer)
Dim header As String
If Index = 22 Then
For i = 0 To UBound (Distance)
Distance (i) = 0
Next
distindex = 0
Text2. Text = ""
End If

Start:
If MS. PortOpen Then 'And DevBusy = False And serialconnected = True Then
'If MS. PortOpen Then
    MS. RThreshold = 5
    MS. InputLen = 5
    MS. Output = Chr (Index + 1)
    lastcommand = Index + 1
    DevBusy = True

    Display "Command :" & Command1. Item (Index). Caption & " id:" & CStr (Index + 1) & vbCrLf,
    &H800000
    Command1 (Index). enabled = False
    Timer2. enabled = True
Else
    answer = MsgBox ("Not Connected! Do you want to connect", vbYesNo, "Error")
    If answer = 6 Then
        Command3_Click
    End If
End If

End Sub
```

“Mscomm” nesnesi port üzerinde 5 byte’lık veri aldığıında “MS_OnComm” yordamı çalışmaya başlar. Mscomm nesnesinin iletişim tamponu “sdata” değişkenine aktarılır ve ardından “ProcessResponse” yordamı “sdata” parametresi ile işleme alınır.

“ProcessResponse” yordamı ana kontrol kartından gelen yanıtların yorumlanmasından sorumludur. Bu işlem sırasında en son gönderilen komutun indeks değerini tutan “lastcommand” değişkeninden faydalanılır. Bir “select case” yapısı kullanılarak en son gönderilen komuta göre gerekli işlemler gerçekleştirilir. Ana kontrol kartından gelen yanıtlar gönderilen komuta göre değişiklik göstermektedir. Motorlarla ilgili komutlara “tamam” metni ile cevap verilirken, algılayıcı komutlarına algılayıcıdan okunan rakamsal değer ile yanıt verilmektedir. Gelen tüm yanıtlar ilk olarak “log window” penceresine eklenmektedir. Rakamsal değerler ise ilgili algılayıcı göstergesinin değerlerine yansıtılmaktadır. Burada ayrıcalıklı iki komut bulunmaktadır. Sonar ve kızıl ötesi taraması yapan komutlar ardı ardına okuma değerlerini ana bilgisayara göndermektedirler. Bu nedenle yanıt

yorumlayıcı bu komutlar için farklı bir yöntem izlemektedir. Okunan değerler bir diziye aktarılmakta ve ardından “Excel” tablosuna kayıt edilmektedir. “Excel”in grafik özellikleri kullanılarak bu veriler radar grafiği olarak gösterilmektedir.

Algılayıcıları verilerinin gösterilmesi amaçlı olarak pusula, analog vumetre ve 7 bölmeli led göstergeleri barındıran özel kullanıcı kontrolleri⁵⁴ geliştirilmiş ve projeye dahil edilmiştir.

Robot kontrol terminal ekranı uygulama katmanının makro ve donanım komutlarını barındırmasının yanı sıra robot üzerinde bulunan kablosuz video kameranın görüntülerinin kullanıcıya aktarılması görevini de üstlenmektedir. Kablosuz IP kameranın asf formatın yayınladığı görüntüler projeye dahil edilen “Windows media player control” nesnesi ile kablosuz ağ üzerinde okunmakta ve bu ekranda görüntülenmektedir.

## **7.2 Mikroişlemci Yazılımı**

Anabilgisayar üzerinde çalışan kontrol yazılımının oluşturduğu donanım komutları ana kontrol kartı üzerinde çalışan mikroişlemci yazılımı tarafından işlenmekte ve hayata geçirilmektedir. Mikrobasic programlama dili ile yazılan ve derlenen bu yazılım, robotun ana bilgisayar ile haberleşmesinin, motor kontrol kartlarının, lcd ekran ve konuşma sentezleyicinin kontrolünün yanı sıra analog ve dijital algılayıcıların okunmasından sorumludur. Uygulama katmanının en alt bölümü olarak çalışmaktadır. Program komut yorumlayıcı olarak çalışan “main” yordamı ve alt yordamlardan oluşmaktadır. Bu bölümde ilk olarak “main” yordamının çalışması incelenecek ardından donanım fonksiyonlarını yerine getiren alt yordamların detaylarına değinilecektir. Programın kaynak kodu Ek-B’de verilmiştir.

### **7.2.1 Ana Modül**

Bu modül her mikrobasic programının barındırması gereken ana kod bloğunu oluşturmaktadır. PIC mikroişlemciye güç verildikten sonra ilk olarak konfigürasyon yazmaçlarını okumakta ardından komut imlecini “main” etiketinin bulunduğu adrese getirerek komutları işlemeye başlamaktadır.

Ana kontrol kartında çalışan programın “main” modülü iki ana bölümden oluşmaktadır. İlk bölümde program içerisinde kullanılan değişkenlerin ilk değerleri

---

⁵⁴ Visual Basic User Control

atanmakta ardından işlemci portlarının TRIS yazmaçları ayarlanmaktadır. Ardından LCD, analog dijital çevirici ve işlemcinin USART donanımının öncülleme⁵⁵ işlemi gerçekleştirilmektedir. Bu işlemlerin tamamlanmasından sonra komut yorumlayıcısını barındıran ikinci bölüm devreye girmektedir. Bu bölüm işlemci kapatılana kadar süren sonsuz bir döngüden oluşmaktadır. Döngünün her adımında USART tampon belleği “Usartbyte” değişkenine okunmakta ardından bir “select case” yapısı ile ana bilgisayardan gelen komut işlenmektedir. Genel olarak her bir komut için ilgili yordam çalıştırılmaktadır. Servo motor tarama komutları, dc motor komutları, sonar ve kızıl ötesi tarama işlemleri ise birden fazla yordamın çalıştırılması ile gerçekleştirilmektedir. Her bir yordam işlem sonucunu evrensel “text” değişkenine yazmaktadır. Yordamın tamamlanmasının ardından kontrol ana modüle geri dönmektedir. Ana modül “text” değişkeninin içeriğini donanım USART modülünü kullanarak ana bilgisayara göndermektedir. RS232 9600 8N1 standardında gerçekleşen bu haberleşmede gönderilen sinyaller wibox modülü tarafından kablosuz ağ kullanılarak ana bilgisayara aktarılmaktadır. Bu haberleşme sırasında wibox modülü transparan olarak çalışmakta; mikro işlemci yazılımı ve ana bilgisayar yazılımı donanımsal bir seri port üzerinden çalışıyormuşçasına işlem yapmaktadır.

Mikro işlemci yazılımı, algılayıcıların okunması, dc motor ve servo motorların kontrolü ve konuşma sentezleyicisi ile sesli geri bildirim işlemlerini gerçekleştiren yordamlar barındırmaktadır. Bu yordamların yanı sıra işlemlerin zaman senkronizasyonunu sağlamak için gecikme⁵⁶ yordamları da sisteme eklenmiştir. Her komut ile ilgili yordamların incelenmesi robotun donanımının çalışmasının anlaşılabilmesi için önem taşımaktadır.

### **7.2.2 Kızıl Ötesi Algılayıcı Yordamı**

Bu yordam robot üzerinde bulunan 4 adet kızıl ötesi algılayıcısının analog dijital çevirici kullanılarak okunması ve uzaklık değerlerinin hesaplanmasını gerçekleştirmektedir. Parametre olarak algılayıcı indeksini almaktadır. Mikrobasic’in “Aadc_read (channel)” komutu kullanılarak ilgili analog dijital çevirici kanalının voltaj değeri 10-bit okunmakta ardından bu voltaj değerinden mesafe hesaplanmaktadır. Bölüm 6’da da değinildiği gibi Sharp kızılötesi algılayıcıların voltaj-mesafe ilişkisi doğrusal olmayan bir yapıdadır. Bu nedenle hesaplamada bir

---

⁵⁵ İng Initialization

⁵⁶ İng: Delay

doğrusallaştırma fonksiyonu kullanılmaktadır. Sharp kızıl ötesi algılayıcıların veri kılavuzunda⁵⁷ Voltaj ile  $1 / ( \text{mesafe} + 0.42 )$  arasında doğrusal bir ilişki olduğu belirtilmektedir. Veri kılavuzundaki grafikler incelendiğinde mesafe voltaj arasında denklem 7.1’de gösterilen ilişki olduğu görülmektedir.

$$\text{Mesafe} = \frac{6787}{\text{Voltaj} - 3} - 4 \quad (7.1)$$

Bu formül kullanılarak hesaplanan mesafe değeri word değişken türünden “Wordtostr (Data, Text)” komutu kullanılarak metne çevrilmekte ve ana bilgisayara gönderilecek yanıt bilgisini tutan “text” değişkenine atanmaktadır.

**Tablo 7.7 : Kızıl Ötesi Algılayıcı Yordamı**

```
Sub Procedure Read_sharp (Dim Index As Byte)
T = Adc_read (Index)      ' Get Adc Value From Channel
Data = (6787 / (T - 3)) - 4 ' Calculate Distance From Voltage
'=====format And Display Data=====
'      Lcd_cmd (Lcd_clear)      ' Send Command To Lcd (Clear Lcd)
'      Text = Sname1
'      Lcd_out (1, 1, Text)      ' Print String A On Lcd, 1st Row, 1st Column
'      Text= "Distance:"
'      Lcd_out (2, 1, Text)
'      Text="Cm"
'      Lcd_out (2, 15, Text)
'      Wordtostr (Data, Text)    ' Change Integer To String
'      Lcd_out (2, 10, " ")    ' And Print It On Lcd
'      Lcd_out (2, 9, Text)
Wordtostr (Data, Text)    ' Format Word
End Sub
```

### 7.2.3 Ultrasonik Mesafe Algılayıcı Yordamı

Robot üzerinde bulunan iki ultrasonik mesafe ölçerin değerlerinin okunması bu yordam aracılığıyla yapılmaktadır. I2C veri yolunu kullanan bu algılayıcılar gönderilen komut üzerine mesafe ölçümünü başlatmakta ve 65ms içerisinde ölçümü sonuçlandırmaktadır. Okuma işlemi üç ayrı I2C oturumu kullanılarak gerçekleştirilmektedir. İlk iki oturumda algılayıcı I2C adresinin okuma yazma biti 0 olarak kullanılmakta üçüncü oturumda ise bu bit 1 olarak algılayıcıya gönderilmektedir. Birinci oturumda algılayıcının komut yazmaçı adresine ölçümün santimetre cinsinden yapılacağını belirten 0x51 değeri yazılmaktadır. İşlemin tamamlanması için 75 milisaniye bekledikten sonra ikinci I2C oturumu başlatılmaktadır. Bu oturumda okunmak istenen ölçüm yazmaçının adresi

⁵⁷ İng: datasheet

algılayıcıya gönderilmektedir. Bu adresi alan algılayıcı okumanın yapılacağı 3. oturum için hazır hale gelmektedir. Ölçüm değerinin alt ve üst bytelerini barındıran iki yazmaç ardı ardına okunarak yordam içerisinde birleştirilmektedir. Ardından okunan değer “text” değişkenine aktarılarak yordam sonlandırılmaktadır. Bu yordamın kaynak kodu Tablo 7.8’de verilmiştir.

**Tablo 7.8 : Ultrasonik Mesafe Algılayıcı Yordamı**

```
Sub Procedure Read_sonar
    Soft_I2C_config (Portd, 0, 1)      ' Initialize I2C Module 100 Khz Clk, Full Master Mode
    '=====start Ranging=====
    Soft_I2C_start ()                  ' Issue I2C Signal: Start
    Soft_I2C_write (0xe0)              ' Sonar Address
    Soft_I2C_write (0x00)              ' Command Register Address
    Soft_I2C_write (0x51)              ' Send Data To Be Written Ranging Mode - Result In Centimeters
    Soft_I2C_stop ()                  ' Issue I2C Signal: Stop
    Delay_ms (75)                      ' Pause While Sonar Is Busy
    Soft_I2C_start ()                  ' Issue I2C Start Signal
    Soft_I2C_write (0xe0)              ' Send Sonar Address
    Soft_I2C_write (0x00)              ' Send Byte (Address For Highbyte)
    Soft_I2C_start ()                  ' Issue I2C Signal: Repeated Start
    Soft_I2C_write (0xe1)              ' Send Byte (Request Data From Sonar)
    Sw_revision = Soft_I2C_read (1)     ' Read The Software Revision Number
    Delay_5500us
    Unused = Soft_I2C_read (1)         ' Read register Data 0x80
    Delay_5500us
    Ee_high = Soft_I2C_read (1)        ' Read The High Byte
    Delay_5500us
    Ee_low = Soft_I2C_read (0)         ' read Low Byte
    Soft_I2C_stop ()                  ' Issue I2C Signal: Stop
    '=====Format & Display Data=====
    Data = Word (Ee_high << 8) + Ee_low 'combine Highbyte And Low Byte
    Wordtostr (Data, Text)             ' Format Word
End Sub
```

#### 7.2.4 Elektronik Pusula Yordamı

Robotun gerçekleştirdiği makro komutların ardından oluşabilecek açıl sapmaları düzeltmek için elektronik bir pusula kullanılmaktadır. Bu pusula ile I2C veriyolu kullanılarak haberleşme gerçekleştirilmektedir. Kullanılan ultrasonik mesafe algılayıcıların tersine elektronik pusula sürekli olarak ölçüm yapmakta ve herhangi bir tetiklemeye ihtiyaç duymamaktadır. Bu nedenle yönelim bilgisini 2 ayrı I2C oturumunda okumak mümkün olmaktadır. İlk oturumda pusulanın I2C adresi okuma yazma biti boş olarak gönderilmekte; bu adresi takiben okunmak istenen yazmaçın adresi gönderilmektedir. Ardından I2C oturumu yeniden başlatılıp adres okuma yazma biti set edilmiş olarak gönderilmekte ve yazmaçların okunması gerçekleştirilmektedir. Yönelim bilgisinin ilk baytı pusulanın 0x02 adresinde ikinci baytı ise 0x03 adresinde bulunmaktadır. Okunan bu yazmaçlar birleştirilip metin

haline çevrilerek “text” değişkenine aktarılmaktadır. Okunan yönelim değeri ana modül tarafından bilgisayara gönderilmekte ve konuşma sentezleyicisine aktarılarak sesli geri bildirim sağlanmaktadır. Bu yordamın kaynak kodu Tablo 7.9 ‘da verilmiştir.

#### **7.2.5 İvme Ölçer Yordamı**

Robotun üzerinde -2g +2g aralığında ölçüm yapabilen iki eksenli bir ivme ölçer bulunmaktadır. Analog çıkışa sahip olan bu ivme ölçer mikro işlemcinin analog dijital çeviricisi ile okunmakta ve 10bitlik ham data işlenmek üzere bilgisayara gönderilmektedir. Ardından x ve y eksen verileri kalibrasyon verileri ile karşılaştırılarak algılayıcı üzerinden robotun eğimi hesaplanmaktadır. Hesaplanan bu değer “text” değişkenine atanarak yordam sonlandırılmaktadır. Açının hesaplanması ile ilgili detaylar Tablo 7.10’da verilmiştir.

#### **7.2.6 Hareket Algılayıcı Yordamı**

Robot çevresindeki hareketleri algılamak için pasif kızılötesi hareket algılayıcı barındırmaktadır. Bu algılayıcı analog sayısal çevirici üzerinde mikro işlemciye bağlanmıştır. Okunan 10bitlik değer voltaja çevrilmekte ve belirlenen eşiği aşıyor ise hareket olduğuna karar verilmektedir. Hareket durumunda “text” değişkenine “mvmnt” sabit durumda ise “stabl” değeri atanarak yordam sonlandırılmaktadır. Bu yordamın detayları Tablo 7.11’de verilmiştir.

**Tablo 7.9 : Elektronik Pusula Yordamı**

```

Sub Procedure Read_compass
I2Cport=0
Soft_I2C_config (I2Cport, Sda, Scl)      ' Initialize I2C Module 100 Khz Clk, Full Master Mode
'=====start Reading=====
Soft_I2C_start      ' Issue I2C Start Signal
Soft_I2C_write (0xc0) ' Send Byte Via I2C To Cmps03 Adres 0xc0
Soft_I2C_write (0x00)      ' Send Byte (Register Adres)
Soft_I2C_start      ' Issue I2C Signal Repeated Start
Soft_I2C_write (0xc1) ' Send Byte (Request Data From Eeprom)
Sw_revision = Soft_I2C_read (1) ' Read The Software Version From Register 0 And Send
Acknowledge
Delay_5500us
Ee_byte = Soft_I2C_read (1)      ' Read The Unused Register From Register 1 And Send
Acknowledge (It Should Read 0x80)
Delay_5500us
Ee_high = Soft_I2C_read (1)      ' Read The Highbyte Data And Send Acknowledge
Delay_5500us
Ee_low = Soft_I2C_read (0)      ' Read The Low Byte Data And Do Not Send Acknowledge
Soft_I2C_stop      ' Issue I2C Stop Sinal
'=====Format and Display Data=====
Data = Word (Ee_high << 8) + Ee_low 'combine Highbyte And Low Byte
Wordtostr (Data, Text)      ' Format Word
End Sub

```

**Tablo 7.10 : İvme Ölçer Yordamı**

```

Sub Procedure Read_accel

X = Adc_read (0)      'read X Channel
Y = Adc_read (1)      ' Read Y Channel
Usart_Write_Text (x
Usart_Write_Text (y)
'=====format And Display Data=====
anglex=x-zeroG_x
angley=y-zeroG_y
data = word (anglex << 8) + word (angley)

if anglex<0 then
    anglex=zeroG_x-x      'Find absolute value of difference from zeroG
end if
if angley<0 then
    angley=zeroG_y-y      'Find absolute value of difference from zeroG
end if

if anglex<angley then
    angle=anglex*45      'Use result for X if angle is <45 deg
    angle=angle div oneG_x      'Use linear scaling
else
    angle=angley*45      'Use result for y if angle is >45 deg
    angle=angle div oneG_y      'Use linear scaling
    angle=90-angle
end if
LCD_Cmd (LCD_CLEAR)
text="Angle: "
LCD_Out (2, 1, text)      ' print string a1 on LCD, 2nd row, 1st column
Data = Word (X)
Wordtostr (Data, Text)      ' Format Word
End Sub

```

**Tablo 7.11 : Hareket Algılayıcı Yordamı**

```
sub procedure read_pir
  t = ADC_read (2)      ' get ADC value from 2nd channel
  tlong = t*5000
  t = longint (tlong >> 10)
  ch = t div 1000      ' prepare value for display
  if ch>0 then
    text="mvmnt"
    lcd_out (2, 9, Text)
    sound_init (portd, 0)
    melody
  else
    text="stabl"
    lcd_out (2, 9, Text)
  end if
end sub
```

### 7.2.7 Çarpma Algılayıcı Yordamı

Robotun ilerlerken karşılaşılabileceği engeller mesafe algılayıcılar tarafından algılanmaya çalışılmaktadır. Ancak bu algılayıcıların başarısız olması durumunda robotun bir engele çarpası kaçınılmazdır. Çarpışmanın algılanması ve robotun hareketini durdurması büyük önem taşımaktadır. Bu nedenle çarpışma algılama için kullanılan dokunma algılayıcılar diğer algılayıcılardan farklı bir sistem ile kontrol edilmektedir. PIC mikro işlemcinin barındırdığı timer0 zamanlayıcısı yardımı ile dokunma algılayıcıları her 50 milisaniyede bir okunarak bir çarpışma olup olmadığı kontrol edilmektedir. Timer0 zamanlayıcısı 1:32 ön-oranlayıcı⁵⁸ ayarı ve 100 başlangıç değeri ile çalıştırılmaktadır. Bu 20 mhz çalışma hızında 502.4 mikro saniyelik bir periyoda eşit olmaktadır. Dokunma algılayıcılar ile ilgili okumalar ve çarpışma algılama işlemi Tablo 7.12’de verilen kesme yordamında⁵⁹ gerçekleşmektedir. Timer0 zamanlayıcısında taşma (sayacın 255 değerini aşıp 256 değerini alması) meydana geldiğinde mikroişlemci INTCON yazmaçının 5. biti olan TMR0IF bitini 1 olarak atamaktadır. Bu olay kesme işlemi başlatmaktadır. İşlemci normal program akışını keserek kesme vektörünün bulunduğu adresteki komutları işlemeye başlar. Kesme yordamında ilk olarak dokunma algılayıcılarının bağlı olduğu portd7 pini kontrol edilir. Eğer bu değer 1 ise (dokunma algılanmış) motor kontrol kartına motorları durdurması için komut gönderir. Ardından bilgisayara çarpışma olduğunu belirtmek amacıyla “crash” mesajını seri port üzerinden aktarır. Kesme yordamından çıkmadan önce timer0 başlangıç değeri olan 100 ile

⁵⁸ İng: Prescaler

⁵⁹ İng : Interrupt handler routine

yüklendikten sonra TMR0IF biti temizlenerek bir sonraki kesme işlemi için hazırlanmaktadır.

**Tablo 7.12 :Kesme Yordamı**

```
sub procedure interrupt
  if portd. 7=1 then
    Set_motorspeed (128, 128) 'stop motors
    Usart_send_text ("crash")
  End if
  TMR0 = 0x64
  TMR0IF = 0
end sub
```

### 7.2.8 DC Motor Kontrol Yordamları

Robotun hareketini sağlayan dc motorlar devantech md22 motor kontrol kartı tarafından yönetilmektedir. Bu kart ana kontrol kartı ile I2C veriyolunu kullanarak haberleşmektedir. Diğer I2C cihazlarda olduğu gibi ilk olarak cihazın adresi ardından yazılmak istenen yazmaçın adresi gönderilmekte ardından da yazılmak istenen değer cihaza aktarılmaktadır. MD22 kontrol kartı I2C modunda çalışırken farklı motor kontrol yöntemlerini desteklemektedir. Bu projede mod-0 kontrol yöntemi kullanılmaktadır. Mod-0'da 0x01 ve 0x02 yazmaçları sağ ve sol motorun hız değerlerini tutmaktadır. Bu değer 0-255 arasında değişmektedir. 0 değeri tam hız geri anlamına gelirken 255 değeri tam hız ileri anlamını taşımaktadır. Bir motoru durdurmak için ise ilgili yazmaça 128 değeri yazılmaktadır.

Bilgisayar üzerinden gelen her bir donanım motor kontrol komutu, md22 kontrol kartı ile haberleşmek için kaynak kodu Tablo 7.13'de verilen "set_motorspeed" yordamını kullanmaktadır. Bu yordam ilk olarak kontrol kartının çalışma modunu mod0 olarak ayarlamakta ardından 0x01 ve 0x02 yazmaçlarına kendisine parametre olarak gönderilen "left_motorspeed" ve "right_motorspeed" değişkenlerini yazmaktadır. Bu yordam en ilkel motor komutu olarak düşünülmelidir.

**Tablo 7.13 :Dc Motor Hız Yordamı**

```
Sub Procedure Set_motorspeed (Dim Leftmotor As Byte, Dim Rightmotor As Byte)
Soft_I2C_config (Portd, 0, 1) ' Initialize I2C Module 100 Khz Clk, Full Master Mode
Soft_I2C_start () ' Issue I2C Signal: Start
Soft_I2C_write (0xb0) ' Send md22 Adress
Soft_I2C_write (0x00) ' Send mod register adress
Soft_I2C_write (0x00) ' Set mod to mod-0
Soft_I2C_write (Leftmotor) ' Set left motor speed
Soft_I2C_write (Rightmotor) ' Set right motor speed
Soft_I2C_stop () ' Issue I2C Signal: Stop
End Sub
```

Diğer hareket komutları gecikme yordamları ile birlikte bu yordamı kullanarak istenilen hareketin yapılmasını sağlamaktadır. Örneğin robotun sola dönmesi gerektiğinde set_motorspeed (0, 255) komutu verilerek sol motorun tam yol geri sağ motorun tam yol ileri çalışması sağlanmakta 1000ms (1 saniye) sonra set_motorspeed (128, 128) komutu verilerek motorlar durdurulmaktadır. Kullanılan gecikme süresi dönüşün miktarını belirlemektedir. Böylece gecikme süresi değiştirilerek istenilen açıda dönüş veya istenilen mesafede ilerleme sağlanabilmektedir. Bu özellik bilhassa açısal hataların düzeltilmesi için kullanılan ufak dönüşler için büyük önem taşımaktadır. Projede kullanılan dc motor kontrol yordamlarının yapısı ve kullandıkları gecikme süreleri Tablo 7.14’de bu komutların birleşmesinden oluşan robot hareket komutları Tablo 7.15’de verilmiştir.

### 7.2.9 Servo Motor Komutları

Robotun üzerinde bulunan algılayıcı taratlerinin hareketleri 4 adet servo motor tarafından sağlanmakta ve bu motorlar SD21 servo kontrol kartı aracılığıyla kontrol edilmektedir. SD21 kartı aynı anda 21 servoyu 20ms tazeleme hızını koruyarak kontrol edebilmektedir. Her bir servo için biri hız diğer ikisi konum belirten 3 adet yazmaç bulunmaktadır. Bu yazmaçlar I2C veriyolu ile değiştirilerek servo motorların istenilen konuma gelmeleri sağlanmaktadır. Servo konumlandırma komutu (Tablo 7.16) diğer yordamlar tarafından da kullanılmaktadır. Özellikle tarama yapan algılayıcı komutları servo motorun konumunu adım adım değiştirerek her adımda okuma yapmaktadırlar. Servo motorların konumlarının program içerisindeki tüm yordamlar tarafından erişilebilmesi için “servo motor” adında bir yapı tanımlanmıştır. Ardından bu yapının bir kopyası 4 elemanlı bir dizi olarak oluşturulmuştur. Bir servo motorun konumunu değiştirmek isteyen yordam, ilk olarak “servo[indeks].pos” ve “servo[indeks].speed” değişkenlerinin değerlerini atamakta ardından “set_servopos[indeks]” komutunu çalıştırarak motorların istenen

konuma hareket etmesini sağlamaktadır. Program her bir servo motor için birer adımlik sağa veya sola dönüş yordamlarının yanı sıra motorun tüm hareket açısını taramayı sağlayan yordamları da barındırmaktadır.

**Tablo 7.14 : Motor Kontrol Komutları**

'=====
'dc Motor 1 Forward
'=====
Sub Procedure Dc_fwd1
Set_motorspeed (255, 128)
Mwait '100ms
Set_motorspeed (128, 128)
End Sub
'=====
'dc Motor 1 Backward
'=====
Sub Procedure Dc_rew1
Set_motorspeed (0, 128)
Mwait '100ms
Set_motorspeed (128, 128)
End Sub
'=====
'dc Motor 2 Forward
'=====
Sub Procedure Dc_fwd2
Set_motorspeed (128, 255)
Mwait '100ms
Set_motorspeed (128, 128)
End Sub
'=====
'dc Motor 2 Backward
'=====
Sub Procedure Dc_rew2
Set_motorspeed (128, 0)
Mwait '100ms
Set_motorspeed (128, 128)
End Sub
'=====
'dc Motors Left 1 Step
'=====
Sub Procedure Dc_left
Set_motorspeed (0, 255)
Bekle2 '10ms
Set_motorspeed (128, 128)
End Sub
'=====
'dc Motors Right 1 Step
'=====
Sub Procedure Dc_right
Set_motorspeed (255, 0)
Bekle2 '10ms
Set_motorspeed (128, 128)
End Sub

**Tablo 7.15 : Robot Hareket Komutları**

'=====
'Robot Turn Left
'=====
Sub Procedure Turnleft
Set_motorspeed (0, 255)
Bekle '500 Ms
Bekle '500 Ms
Set_motorspeed (128, 128)
End Sub
'=====
'Robot Turn Right
'=====
Sub Procedure Turnright
Set_motorspeed (255, 0)
Bekle '500 Ms
Bekle '500 Ms
Set_motorspeed (128, 128)
End Sub
'=====
'Robot Go Backward
'=====
Sub Procedure Gobackward
Set_motorspeed (0, 0)
Bekle '500 Ms
Bekle '500 Ms
Set_motorspeed (128, 128)
End Sub
'=====
'Robot Go Forward
'=====
Sub Procedure Goforward
Set_motorspeed (255, 255)
Bekle '500 Ms
Bekle '500 Ms
Set_motorspeed (128, 128)
End Sub

**Tablo 7.16 : Servo Konumlandırma Komutu**

```
Structure Servomotor
  Dim Pos As Word
  Dim Speed As Byte
End Structure
Dim Servo As Servomotor[4]
Sub Procedure Set_servopos (Dim Servoindex As Byte)
  Dim Regaddress As Byte
  Regaddress =3* (Servoindex-1)

  Portd=0 'reset Port
  Soft_I2C_config (Portd, 0, 1) ' Initialize I2C Module 100 Khz Clk, Full Master
  Mode
  '=====start Writing Servo Position=====
  Soft_I2C_start ' Issue I2C Start Signal
  Soft_I2C_write (0xc2) ' Send Byte Via I2C To Servo Controller Addres 0xc02
  Soft_I2C_write (Regaddress) ' Send Byte (Register Adres)
  Soft_I2C_write (Servo[Servoindex]. Speed) ' Send Servo Speed
  Soft_I2C_write (Lo (Servo[Servoindex]. Pos)) ' Send Servo Position Low Byte
  Soft_I2C_write (Hi (Servo[Servoindex]. Pos)) ' Send Servo Position High Byte
  Soft_I2C_stop ' Issue I2C Stop Sinal
End Sub
```

#### 7.2.10 Konuşma Sentezleyici Yordamı

Bu bölümde son olarak robotun aldığı komutlara sesli olarak yanıt vermesini sağlayan konuşma sentezleyicisi incelenecektir. SP03 konuşma sentezleyicisi usart veya I2C veri yolları üzerinden gönderilen salt metin halindeki verileri konuşmaya çevirebilmektedir. Bu projede haberleşme için I2C veriyolu kullanılmıştır. Sentezlenecek konuşma ilk olarak sp03'ün tampon belleğine aktarılmakta ardından SPKBUF komutu gönderilerek tampon bellekteki metnin konuşmaya çevirilmesi sağlanmaktadır. Sp03 bir seferde 81 karaktere kadar metni konuşmaya çevirebilmektedir. Ancak projede mikro işlemci üzerindeki hafıza gereksiniminin en aza indirilmesi amacı ile 40 karakterlik bölümü kullanılmaktadır.

**Tablo 7.17 : Konuşma Sentezleyici Yordamı**

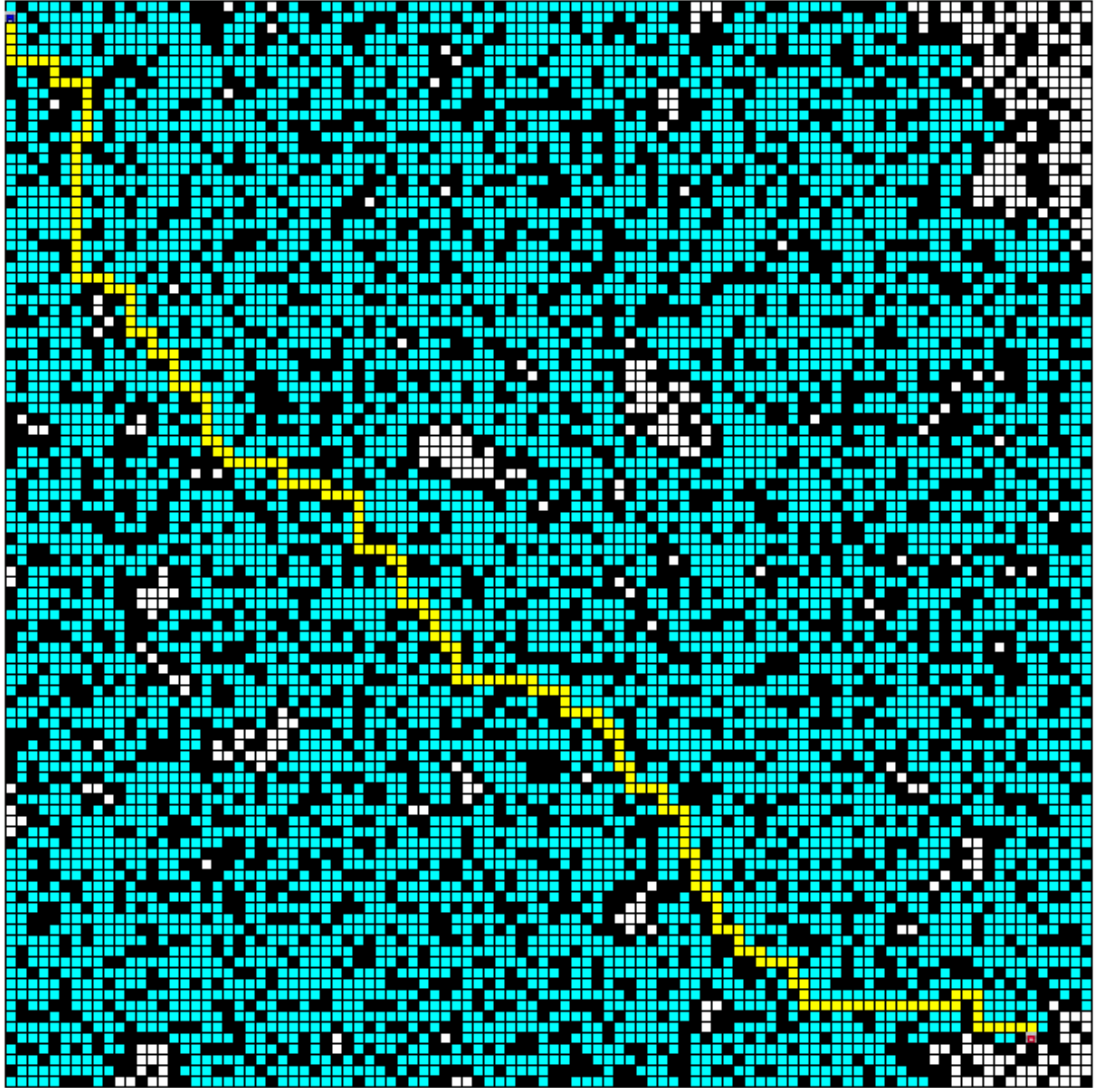
```
Sub Procedure Speaktext (Dim Byref Sptext As String[40])
  Portd=0
  Soft_I2C_config (Portd, 0, 1)      ' Initialize I2C Module 100 Khz Clk, Full Master Mode
  '-----
  ' Buffer Text
  '-----
  Soft_I2C_start ()                  'Issue I2C Signal: Start
    Soft_I2C_write (0xc4)            'address
    Soft_I2C_write (0x00)            'command Register
    Soft_I2C_write (0x00)            'sp03 Nop Command
    Soft_I2C_write (0x02)            'volume (Max. )
    Soft_I2C_write (0x05)            'speech Speed
    Soft_I2C_write (0x03)            'speech Pitch
    For Spindex =0 To Strlen (Sptext)
      Soft_I2C_write (Sptext[Spindex])  'text Char
    Next Spindex
    Soft_I2C_write (0x00)            'Null
  Soft_I2C_stop ()                  'Issue I2C Signal: Stop
  '-----
  'speak Buffered Text
  '-----
  Soft_I2C_start ()                  ' Issue I2C Signal: Start
    Soft_I2C_write (0xc4)            ' Address
    Soft_I2C_write (0x00)            ' Command Register
    Soft_I2C_write (0x40)            ' Speak Text Previously Stored In Buffer
  Soft_I2C_stop ()
End Sub
```

## 8 SONUÇ

Robotların hayatımızda hızla daha büyük bir önem kazanması süregelen araştırma geliştirme faaliyetlerinin bir sonucudur. Günümüzde sanayi ve bilimsel araştırmaların yanı sıra ev içi kullanım için tasarlanmış robotlar da ardı ardına boy göstermektedir. Ülkemizde ise yapılan robot araştırmalarının sınırlı sayıda olması gelişmekte olan bu endüstride söz sahibi olmamızı engellemektedir. Bu nedenle mobil robotlar konusunda yapılan çalışmalar göz önüne alınarak bu projede insan müdahalesi olmadan ön tanımlı bir ortamda otonom olarak hareket eden bir robotun geliştirilmesi planlanmıştır. Kendisine verilen görevleri yerine getirebilen ve taktik kararlar alabilen zeki bir kontrol mekanizmasını barındıran bu robotun aynı zamanda insan operatörler tarafından kontrol edilebilecek şekilde çalışması sağlanmıştır. Robotun taktik kararlar alabilen ve zeki davranışlar sergileyebilen bir yapıda olması nedeniyle İngilizce Intelligent Tactical Unit - Robot kelimeleri kısaltılarak robota ITUbot ismi verilmiştir.

Projenin ilk aşamalarında gerçekleştirilecek senaryolar üzerinde çalışılmış ve bu senaryoların çözümleri üzerine araştırmalar yapılmıştır. Otonom bir mobil robotun en önemli özelliklerinden biri olan yol bulma (pathfinding) üzerine yapılan araştırmalar sonucunda A* ve Dijkstra algoritmaları bilgisayar ortamında gerçekleştirilmiştir. Bu yol bulma algoritmalarından hareketle daha iyi bir işlem performansı sunan bir algoritma MS Visual Basic'te çalışacak şekilde hazırlanmıştır. Bu algoritma hafıza gereksinimi ve simülasyonlardaki başarımı ile öne çıkmaktadır. Algoritmanın çalışması sırasındaki görsel unsurlar (haritanın ekrana çizilmesi vb. ) kaldırılarak başarımlar daha da üst seviyeye çıkartılabilmektedir. 100x100 ebadındaki aşırı karmaşık bir haritada başlangıç ve bitiş noktaları haritanın karşılıklı köşelerine yerleştirilmesi durumunda bile toplam işlem süresi 17 saniyenin altındadır. Bu süre aynı harita üzerinde Ms Visual Basic'te uygulaması yapılmış A* algoritması kullanıldığında 20 saniyenin üzerine çıkmaktadır. 100x100 boyutundaki bir harita üzerinde gerçekleştirilen yol bulma simülasyonu **Şekil 8.1**'de gösterilmektedir. Bu simülasyonda toplamda 192 adımdan oluşan bir yol 14,14 saniyelik bir sürede

hesaplanmıştır. Bu süre görsel animasyonlar kapatılarak 7.92 saniyeye kadar düşürülebilmektedir.



Şekil 8.1 : Yol Bulma Simülasyonu

Projede robot kontrol stratejileri incelenerek çok katmanlı bir model oluşturulmuştur. Bu model hiyerarşik bir yapıya sahip olmanın yanı sıra barındırdığı sakınma davranışı nedeniyle davranış temelli kontrol yaklaşımının etkilerini de taşımaktadır. Yol bulma ve yol planlama işlemleri strateji katmanında, yolun izlenmesi ve diğer görevlerin gerçekleştirilmesi için gerekli işlemler taktik katmanında, donanım kontrolü ise uygulama katmanında ele alınmıştır. Sakınma davranışı ise donanım seviyesinde ele alınarak diğer katmanların kararlarını baskılayabilecek bir yapı kazandırılmıştır. Sakınma davranışının bir bölümü ise uygulama katmanının bilgisayar üzerinde çalışan kısmında gerçekleştirilmiştir. Böylece üst katmanların kararları ne olursa olsun robotun kendini engellerden koruması sağlanmıştır.

Bu yazılım alışmasının ardından genel kontrol altyapısı zerinde alışılmıştır. İlk olarak tm yazılımsal hesaplama ve karar verme iřlemlerinin robot zerinde alışması dřnlmřse de hedeflenen senaryolar iin gerekli iřlem gcn saėlayabilecek gml sistemlerin maliyeti nedeniyle robotun kontrol mekanizmasının iřlem yknn bir bilgisayar ve gml bir sistem zerine paylařtırılması yoluna gidilmiřtir. Karar verme ve biliřsel fonksiyonlar iin bilgisayar zerinde alışan bir programın, donanım fonksiyonları iin ise gml bir sistemin tasarlanması ve uygulaması gerekleřtirilmiřtir.

Genel sistem yapısının belirlenmesinin ardından robotun donanım gereksinimleri zerine alışmalar yapılmıřtır. Hareket ve kontrol mekanizmaları, gerekli olabilecek algılayıcılar belirlenmiřtir. Donanım kontrol iin gerekli olan motor kontrol devreleri, ana bilgisayar ile haberleřme iin gerekli iletiřim sistemleri incelenmiřtir.

İlk olarak hareket sistemlerinin tasarlanması ve gereklenmesi zerine alışılmıştır. Yrme, srnme, diferansiyel srř, ackerman srř, senkro srř, paletli srř vb teknikler incelenmiř; esnekliėi ve ızgara tabanlı haritalardaki performansı gz nne alınarak paletli srř tekniėinin kullanılmasına karar verilmiřtir. İlk prototip'te paletli bir hareket mekanizması kullanılmıştır. Fakat paletlerin yarattıėı srtnmeden kaynaklanan konum hataları ve g gereksinimi nedeniyle tekerlekli diferansiyel srř metodu tercih edilerek ikinci bir prototip tasarlanmıřtır.

Hareket mekanizmasının ardından robotun evreyi algılaması, engellerden sakınması ve hareket sırasında oluřabilecek konum ve yn hatalarını dzeltmesi iin gerekli olan algılayıcılar belirlenerek bu algılayıcıları ve hareket mekanizmalarını kontrol edebilecek gml bir sistemin tasarımına geilmiřtir. Bu ařamada eřitli mikro iřlemci trleri incelenmiř ve Microchip firmasının pic mikro iřlemci ailesi tercih edilmiřtir.

Hareket sistemi, haberleřme sistemi, algılayıcılar ve diėer alt sistemlerin kontrol iin bir kontrol kartının yanı sıra bu kontrol kartına baėlı olacak diėer kontrol elemanları belirlenmiřtir. Elektronik donanım seilirken mmkn olan en leklenebilir ve geniřlemeye msait yapı saėlanmaya alışılmıştır. oėu yurtdıřından temin edilen elektronik donanımın ulařmasının ardından her bir kontrol kartının ve algılayıcının ana kontrol kartı tarafından kontroln saėlayacak mikro iřlemci programlarının yazılmasına bařlanmıřtır. Bu iřlem sonucunda ayrı ayrı her

bir donanım için kontrol programları elde edilmiş ve bu programların tek bir çatı altından yönetilmesi sağlanmıştır.

Donanım kontrol yapısının gerçekleşmesini takiben ana bilgisayar ile robot arasındaki iletişimi sağlamak üzere kablosuz iletişim sisteminin tasarlanmasına geçilmiştir. İlk olarak Irda protokolü düşünülmüşse de iletişim mesafesi ve bağlantının görüşe dayalı olması nedeniyle Bluetooth teknolojisi tercih edilmiştir. Bu aşamada bilgisayar ile ana kontrol kartına bağlı tüm alt sistemlerin kontrolü sağlanmıştır. Bluetooth seri port profili kullanılarak ana bilgisayar üzerinde çalışan kontrol programı ile ana kontrolör üzerinde mikroişlemci yazılımının seri haberleşme yapması sağlanmıştır. Temin edilen gömülü Bluetooth modülünün uygulamasında yaşanan sıkıntılar ve testler sırasında yaşanan arıza sonucunda IEEE 802. 11B standardını destekleyen kablosuz bir modül kullanılmasına karar verilmiştir. Lantronics firmasının wibox modülü sağladığı teknik özellikler ve ölçeklenebilir yapısı nedeniyle tercih edilmiştir.

Tüm sistemlerin testleri gerçekleştirildikten sonra elektronik donanım mobil robot platformu üzerine monte edilerek ilk olarak fonksiyonel testler gerçekleştirilmiş ardından senaryo testlerine geçilmiştir. Yapılan fonksiyonel testlerde robotun motorlarından kaynaklanan sistematik açısal hataların olduğu görülmektedir. Her ne kadar aynı marka ve motel motorlar kullanılmış olsa da motorlar arasında işlevsel farklılıklar oluşabilmektedir. Bu sistematik hatanın önüne geçilebilmesi için sağ ve sol motorlar ilerleme sırasında farklı hızlarda çalıştırılmaktadır. Karşılaşılan diğer bir sorun ise sonar algılayıcılarının yerden yansıyan ses dalgalarını algılayarak objeleri daha yakında rapor etmeleridir. Bu sorun algılayıcıların yatay düzlemle 10°'lik bir açı yapacak şekilde monte edilmeleri ile giderilmiştir. Bu düzeltmelerin ardından yapılan testlerde robotun barındırdığı tüm donanım fonksiyonlarının, beklenen işlevleri kabul edilebilir bir hata payı ile yerine getirdiği gözlenmektedir.

Yazılım kontrol yapısının da yapılan testlerde senaryolarda belirlenen kriterlerin üzerinde bir başarımla gösterdiği görülmüştür. Yapılan yol bulma simülasyonları robotun çok karmaşık haritalarda bile çok kısa sürede en kısa yolu bulabildiğini kanıtlamıştır. Ancak robotun ön tanımlı bir ortamda hareket etmesi gereği gerçek ortamlarda yapılacak simülasyonlarda harita boyutunu sınırlamaktadır. 4x4 boyutunda hazırlanan haritalarda robot ilgili senaryolarda çalıştırılmış ve başarılı

olduğu görülmüştür. Robotun tam performansının ölçülebilmesi için algılayıcı ve motorlarının yazılımsal olarak simüle edilmesi gerekmektedir.

Robotun tasarlanması sırasında esnek bir yapı izlenmesi ve ölçeklenebilirliğin birinci önemli kriter olarak alınması robotun farklı donanımlar eklenerek saha başarımının ve fonksiyonelliğinin artırılabilmesini sağlamaktadır. Örneğin eklenecek bir gps donanımı robotun açık alanda hareket ve yön bulma özelliğini artıracak ve ön tanımlı bir sahada çalışma zorunluluğunu ortadan kaldıracaktır. Ayrıca günümüzde 2000\$ ve altında bir maliyet ile temin edilebilen lazer uzaklık ölçme cihazları ve bina içi konumlandırma işlemi için kullanılan evolution northstar gibi sistemler robota eklenerek oluşan konum hataları ve haritaya bağımlılık azaltılabilmektedir.

Yazılım olarak yapılacak değişiklikler ile robotun bir engele çarpmadan serbest olarak dolaşabilmesi, duvarları takip etmesi, ışıktan kaçması veya ışığa yönelmesi gibi senaryolar robota uygulanabilecektir.

Robota bir kol eklenerek çevresi ile etkileşiminin artırılması planlanan geliştirmeler arasındadır. Ayrıca CMUcam adı verilen ve görüntü işleme yeteneğine sahip kameranın eklenmesi robotun renkli nesneleri takip edebilmesini veya yerdeki çizgileri izleyebilmesini sağlamaktadır. Bu donanım özellikleri, maliyetleri nedeniyle robota eklenememiştir.

Günümüzde daha büyük bütçelerle geliştirilen robotların, barındırdıkları gelişmiş donanım sistemleri nedeniyle daha üst bir saha performansı sergiledikleri aşikardır. Ancak nispeten düşük bir bütçe ile gerçekleştirilen bu araştırma çalışması, proje başında belirlenen hedeflere ulaşılması nedeniyle başarılı olduğu düşünülmektedir. Proje sonunda hedeflenen otonom hareket ve karar verme özellikleri sağlanarak belirlenen amaca ulaşılmıştır. İleride yapılacak çalışmalarla robotun saha başarımının artırılması planlanmaktadır.

## KAYNAKLAR

- [1] **Borenstein, J. Everett, B. and Feng, L.,** 1996. Navigating Mobile Robots: Systems and Techniques. A. K. Peters, Ltd. , Wellesley, MA
- [2] **Albus, J. and Quintero, R. ,** 1990. Towards a Reference Model. Architecture for Real Time Intelligent Control Systems (ARTICS. ) Robotics and Manufacturing, ASME Press Series, N. Y. pp 243-250.
- [3] **Gray, J. O.,** 1991. Advanced Robotics and Integrated Approach. Mechatronics and Robotics I ed. P. A. McConall, P. Drews and K. H. Robrock. IOS Press. Amsterdam. 43-50.
- [4] **Miller, D. P.,** 1993. Intelligent Mobile Robots. Perception and Performance, *Proceedings 6th International Conference on Advanced Robotics*, Tokyo, November 1993, 419-422
- [5] **Brooks, R. A,** 1986. A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Automation and Robotics*, Vol. R. A2 No. 1, March 1986, 14-23,
- [6] **Brooks, R. A. ,** 1989. A Robot that Walks. Emergent behaviour from a carefully evolved network. *MIT. Artificial Intelligence Laboratory Report 1091*,.
- [7] **Corfield, S. J. et al,** 1991. Architectures for Real Time Intelligent Control. Control of Autonomous Vehicles, *IEE Computing and Control Engineering Journal*. Vol. 2 No. 6, 1991, 254-262.
- [8] **Lyons, D. M. and Hendricks, A. J,** 1992. Planning for Reactive Robot Behaviour, *IEEE Int. Conference on Robotics and Automation*. Nice, May 1992.
- [9] **Arkin, R. C. ,** 1990. Integrating Behavioural, perceptual and world knowledge in Reactive Navigation Robotics and Autonomous Systems 6, 105-122
- [10] **Thorpe, C. E. , and Herbert, M. ,** 1995. Mobile Robotics. Perspectives and Reality, *Proceedings ICAR*, Barcelona, 1995, 497-506.
- [11] **W. Burgard, D. Fox, D. Hennig, and T. Schmidt,** 1996. Estimating the absolute position of a mobile robot using position probability grids. *In Proc. of the Fourteenth National Conference on Artificial Intelligence*, 896-901,

- [12] **D. Fox, W. Burgard, and S. Thrun**, 1997. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, March 4 (1), 23–33,
- [13] **J. Kleinberg**, 1994. The localization problem for mobile robots. *In Proc. of the 35th IEEE Symposium on Foundations of Computer Science*,
- [14] **B. Kuipers and Y. T. Byun. A**, 1981. Robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, **8** 1981
- [15] **Nourbakhsh, R. Powers, and S. Birchfield**, 1995. DERVISH an office-navigating robot, *AI Magazine*, **16** (2)
- [16] **S. Thrun and A. Bucken**, 1996. Integrating grid-based and topological maps for mobile robot navigation, *In Proc. of the Fourteenth National Conference on Artificial Intelligence*
- [17] **Martin C. Martin and Hans P. Moravec**, 1996. Robot Evidence Grids Carnegie Mellon University.
- [18] **Elfes, A.**, 1990. Occupancy grids: Stochastic spatial representation for active robot perception, *Proceedings of 6th conference on Uncertainty in AI*, july 1990, 66-70
- [19] **Hans P. Moravec**, 1988. Sensor Fusion in Certainty Grids for Mobile Robots, *AI Magazine*, VOL. 9, NO. 2, 1988, 61-74
- [20] **Hart, P. E. , Nilsson, N. J. ; Raphael, B. ,** 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Transactions on Systems Science and Cybernetics* SSC4 (2): 100–107.
- [21] **Hart, P. E. , Nilsson, N. J. ; Raphael, B. ,** 1972. Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *SIGART Newsletter* **37**, 28–29.
- [22] **Dijkstra E. W. ,** 1959. A note on two problems in connexion with graphs. In: *Numerische Mathematik*. 1, 269–271
- [23] **Stentz, A.**, 1994. Optimal and efficient path planning for partially-known environments, *Proceedings IEEE International Conference on Robotics and Automation*, 1994, vol. 4, 3310-3317
- [24] **Goto, Y. , Stentz, A. ,** 1987. Mobile Robot Navigation: The CMU System, *IEEE Expert*, Vol. 2, No. 4, Winter, 1987.

- [25] **Korf, R. E.** , 1987. Real-Time Heuristic Search: First Results, *Proceedings 6th National Conference on Artificial Intelligence*, July, 1987
- [26] **Kocabaş, Ş. , Öztemel, E.** , 1998. AISim: An Intelligent Agent for Distributed Interactive Simulation, *Computer Generated Forces and Simulation of Behavior (CGF-98)*. 12-14 Mayıs 1998, Orlando, Florida. 257-261
- [27] **Kocabaş, Ş. , Öztemel, E.** , 1998. Harp Oyunlarında Yapay Zeka Uygulamaları. TSK-MODSİM, *Modelleme ve Simülasyon Konferansı*, 1-3 Nisan 1998. Kara Harp Okulu, Ankara.
- [28] **Sony Corporation**, 2006. Aibo Global Official Website  
<http://www.sony.net/Products/aibo/>
- [29] **WowWee Corporation**. 2006. WowWee Web Portal  
<http://www.wowwee.com>
- [30] **Russel S. , Norvig P.** , 2003. Artificial Intelligence, A Modern Approach, Prentice Hall, 2nd Edition, New Jersey
- [31] **Doğan A.** , 2002. Yapay Zeka, Kariyer Yayıncılık, İstanbul
- [32] **Turing A. M.**, 1950. Computing Machinery and Intelligence, *Mind*, Vol. 59, No. 236, 433-460 <http://cogprints.org/499/00/turing.html>
- [33] **Searle, J. R.** , 1980. Minds, Brains, and Programs, *Behavioral and Brain Sciences*, Vol. 3. , 417-424.
- [34] **Searle, J. R.** , 1983. "Can Computers Think?" from Chalmers (ed. ), *Philosophy of Mind*, Oxford, 2002, 669-675.
- [35] **Özaktaş H.** , 1998. Bilgi Çağında Akıl, Beden Sorunu. Bilkent Üniversitesi. , Ankara
- [36] **McCarthy J. , Minsky M. L. , Rochester N. , Shannon C. E.** , 1955A Proposal For The Dartmouth Summer Research Project On Artificial Intelligence
- [37] **McCarthy J.** , 2004. What Is Artificial Intelligence?, Computer Science Department Stanford University, Stanford
- [38] **Whitehead A. N, Russell B.** , 1910–1913 Principia Mathematica, Cambridge University Press
- [39] **IEEE802. 11**, 2006. IEEE802. 11 standards, IEEE,USA  
<http://standards.ieee.org/getieee802/802.11.html>

- [40] **Electronics Industries Association**, 1969. "EIA Standard RS-232-C Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Data Interchange", Greenlawn NY
- [41] **Philips Semiconductors**, 2001. I2C-bus specification,  
[http://www.semiconductors.philips.com/acrobat_download/literature/9398/39340011.pdf](http://www.semiconductors.philips.com/acrobat_download/literature/9398/39340011.pdf)
- [42] **Microchip Technology Inc.**, 2001. PIC16F87XA Datasheet, Microchip Technology Inc, USA
- [43] **Devantech Ltd**, 2006. MD22 - Dual 50Volt 5Amp H Bridge Motor Drive datasheet, Devantech Ltd, UK
- [44] **Devantech Ltd**, 2006. SD21 - 21 Servo Control Module datasheet, Devantech Ltd, UK
- [45] **Feng L. , Borenstein J. , Everett H. R. ,** 1994. Sensors and Methods for Autonomous Mobile Robot Positioning, *University of Michigan Technical Report UM-MEAM-94-21*, University of Michigan, MI
- [46] **Sharp Electronics**, 2004. SharpGP2D12 Datasheet, Sharp Electronics Co. Japan
- [47] **Devantech Ltd**, 2006. SRF10 - High Performance Ultrasonic Range Finder datasheet, Devantech Ltd, UK
- [48] **Devantech Ltd**, 2006. SRF08 - High Performance Ultrasonic Range Finder datasheet, Devantech Ltd, UK
- [49] **Devantech Ltd**, 2006. CMPS03 Magnetic Compass datasheet, Devantech Ltd, UK
- [50] **Analog Devices Inc. ,** 2005, ADXL311  $\pm 2$  g, Dual Axis Accelerometer datasheet, Analog Devices Inc. , USA
- [51] **Dallas Semiconductor**, 2001. DS18B20 High-Precision 1-Wire Digital Thermometer Datasheet, Dallas Semiconductor Corp. , USA
- [52] **Devantech Ltd**, 2006. SP03 - Text to Speech Synthesizer Datasheet, Devantech Ltd, UK
- [53] **Martin F. G.,** 1996. Robot Builder's Guide, Massachusetts Institute of Technology Press, USA
- [54] **Walter J. ,** 1996. Rapid Learning in Robotics, Technische Fakultät, Universität Bielefeld, AG Neuroinformatik Göttingen: Cuvillier, 1996

- [55] **Lin Ming C.** , 1993. Efficient Collision Detection for Animation and Robotics, Phd Department of Electrical Engineering and Computer Science University of California, Berkeley Berkeley, CA
- [56] **Iovine J.** , 2004. PIC Robotics: A Beginner's Guide to Robotics Projects Using the PICmicro, McGraw-Hill Companies Inc. , USA
- [57] **Iovine J.** , 2000. PIC Microcontroller Project Book, McGraw-Hill Companies Inc. , USA
- [58] **Hellebuyck C.** , 2003. Programming PIC Microcontrollers with PicBasic, Newnes publications, USA
- [59] **Shircliff D. R.** , 2002. Build A Remote-controlled Robot, McGraw-Hill Companies, Inc. , USA
- [60] **Williams D. H.** , 2003, PDA Robotics, McGraw-Hill Companies Inc. , USA
- [61] **Miles P. , Carroll T.** , 2002. Build Your Own Combat Robot, McGraw-Hill Companies Inc. , USA
- [62] **Sandin Paul E.** , 2003. Robot Mechanisms and Mechanical Devices Illustrated, McGraw-Hill Companies Inc. , USA
- [63] **Mukhar K. , Johnson D.** , 2003. The Ultimate Palm Robot, McGraw-Hill Companies Inc. , USA
- [64] **Bergren C. M.** , 2003. Anatomy Of A Robot, McGraw-Hill Companies Inc. , USA
- [65] **Selig J. M.** , 1992. Introductory Robotics, Prentice Hall International (UK) Ltd. UK
- [66] **Ristic Z.** , 2006, Comprehensive manual for mikrobasic. , MikroElektronika Yugoslavia, Belgrad
- [67] **Mikro Elektronika**, 2005, EasyPIC3 Manual, MikroElektronika Yugoslavia, Belgrad

## **EKLER**

### **EK-A Bilgisayar Programı**

CD ile verilmiştir

### **EK-B Mikro İşlemci programı**

CD ile verilmiştir

## EK-C PIC16F877A Pin Tanımları

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI OSC1  CLKI	13	14	30	32	I  I	ST/CMOS ⁽⁴⁾	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO OSC2  CLKO	14	15	31	33	O  O	—	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP MCLR  VPP	1	2	18	18	I  P	ST	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low Reset to the device. Programming voltage input.
RA0/AN0 RA0 AN0	2	3	19	19	I/O I	TTL	PORTA is a bidirectional I/O port.  Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	4	20	20	I/O I	TTL	Digital I/O. Analog input 1.
RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF	4	5	21	21	I/O I I O	TTL	Digital I/O. Analog input 2. A/D reference voltage (Low) input. Comparator VREF output.
RA3/AN3/VREF+ RA3 AN3 VREF+	5	6	22	22	I/O I I	TTL	Digital I/O. Analog input 3. A/D reference voltage (High) input.
RA4/T0CKI/C1OUT RA4  T0CKI C1OUT	6	7	23	23	I/O  I O	ST	Digital I/O – Open-drain when configured as output. Timer0 external clock input. Comparator 1 output.
RA5/AN4/SS-/C2OUT RA5 AN4 SS- C2OUT	7	8	24	24	I/O I I O	TTL	Digital I/O. Analog input 4. SPI slave select input. Comparator 2 output.

**Legend:** I = input      O = output      I/O = input/output      P = power  
— = Not used      TTL = TTL input      ST = Schmitt Trigger input

**Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	9	I/O I	TTL/ST ⁽¹⁾	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.  Digital I/O. External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	36	39	11	12	I/O I	TTL	Digital I/O. Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	39	43	16	16	I/O I	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	40	44	17	17	I/O I/O	TTL/ST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming data.

**Legend:** I = input      O = output      I/O = input/output      P = power  
— = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	34	I/O O I	ST	PORTC is a bidirectional I/O port.  Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	36	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK  SCL	18	20	37	37	I/O I/O  I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	24	26	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	44	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	1	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.

**Legend:** I = input      O = output      I/O = input/output      P = power  
— = Not used      TTL = TTL input      ST = Schmitt Trigger input

- Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	38	I/O I/O	ST/TTL ⁽³⁾	PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus.  Digital I/O. Parallel Slave Port data.
RD1/PSP1 RD1 PSP1	20	22	39	39	I/O I/O	ST/TTL ⁽³⁾	
RD2/PSP2 RD2 PSP2	21	23	40	40	I/O I/O	ST/TTL ⁽³⁾	
RD3/PSP3 RD3 PSP3	22	24	41	41	I/O I/O	ST/TTL ⁽³⁾	
RD4/PSP4 RD4 PSP4	27	30	2	2	I/O I/O	ST/TTL ⁽³⁾	
RD5/PSP5 RD5 PSP5	28	31	3	3	I/O I/O	ST/TTL ⁽³⁾	
RD6/PSP6 RD6 PSP6	29	32	4	4	I/O I/O	ST/TTL ⁽³⁾	
RD7/PSP7 RD7 PSP7	30	33	5	5	I/O I/O	ST/TTL ⁽³⁾	
RE0/ $\overline{RD}$ /AN5 RE0 RD AN5	8	9	25	25	I/O I I	ST/TTL ⁽³⁾	PORTE is a bidirectional I/O port.  Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE1/ $\overline{WR}$ /AN6 RE1 WR AN6	9	10	26	26	I/O I I	ST/TTL ⁽³⁾	
RE2/ $\overline{CS}$ /AN7 RE2 CS AN7	10	11	27	27	I/O I I	ST/TTL ⁽³⁾	
Vss	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12, 13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

**Legend:** I = input      O = output      I/O = input/output      P = power  
— = Not used      TTL = TTL input      ST = Schmitt Trigger input

**Note** 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.  
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.  
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

## **EK-D Robotun Resmi**

## **ÖZGEÇMİŞ**

Okan Okutkan 9 Mart 1979'da Isparta'da doğdu. İlkokulu İzmir Çakırbeyli ilkokulu, orta okulu ise Torbalı Lisesinde okudu. Ardından İzmir Selma Yiğitalp Lisesinin Ağırlıklı Yabancı Dil Bölümünü bitirdi. 1997 yılında İstanbul Teknik Üniversitesi Makina Fakültesi Makina Mühendisliği Bölümündeki eğitimine başladı. Öğrenci olduğu süre zarfında Makina Fakültesi bilgi işlem laboratuvarında sistem yöneticisi olarak çalıştı. Lisans eğitimini tamamlamasının ardından özel sektörde bilişim danışmanı olarak çalışmaya başladı. 2003 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Savunma Teknolojileri Anabilim Dalı Bilişim Teknolojileri bölümünde yüksek lisans eğitimine başladı.