

168861

KOCAELİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNTERNET ÜZERİNDEN ARAÇ KONTROLÜ

YÜKSEK LİSANS TEZİ

Elif Pınar ŞAHİN

Anabilim Dalı: Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Nevcihan DURU

MAYIS 2005

KOCAELİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNTERNET ÜZERİNDEN ARAÇ KONTROLÜ

YÜKSEK LİSANS TEZİ

Elif Pınar ŞAHİN

Tezin Enstitüye Verildiği Tarih: 26 Mayıs 2005

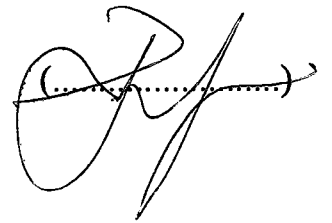
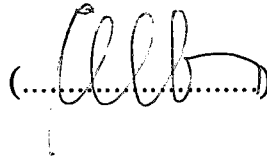
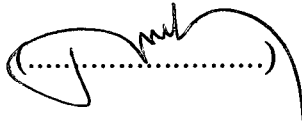
Tezin Savunulduğu Tarih: 30 Haziran 2005

Tez Danışmanı

Üye

Üye

Yrd.Doç.Dr. Nevcihan DURU Prof.Dr.Elbrus CAFEROV Yrd.Doç.Dr.Erhan BÜTÜN



MAYIS 2005

İNTERNET ÜZERİNDEN ARAÇ KONTROLÜ

Elif Pınar ŞAHİN

Anahtar Kelimeler: Araç Kontrolü, Uzaktan Kontrol, Paralel Port Programlama.

Özet: Bu çalışmada, paralel porta gönderilen veriyi kullanarak bir uzaktan kumanda devresinin kontrolü ve TCP/IP teknolojisi kullanılarak telsiz kameradan elde edilen verilerin internet üzerinden kullanıcıya ulaştırılması sağlanmıştır. Projenin gelişim sürecinde, teorisi anlatılan konu radyo frekansı kumandalı bir araba ve bu arabaya monte edilmiş bir telsiz kamera kullanılarak modellenmiştir. Bu model robot, kumanda devresine entegre edilen bir elektronik devre ile kontrol edilmektedir. Kamera kullanarak elde edilen görüntü internet üzerinden bazı yardımcı programlar kullanılarak kullanıcıya iletilmektedir.

VEHICLE CONTROL OVER INTERNET

Elif Pınar ŞAHİN

Keywords : Vehicle Control, Remote Control, Parallel Port Programming.

Abstract: In this study, the control of a remote controller circuit by sending data through the parallel port has been made available. Captured view by a wireless camera is transmitted to the user over the internet by TCP/IP protocol. The system design is simulated with an RF controlled car and a wireless camera during the development process. The specified model robot was controlled by a circuit attached its remote controller. The captured sight of the robot broadcasted over the internet by using some third party software.

ÖNSÖZ ve TEŞEKKÜR

Bilişim teknolojilerinin insan hayatındaki rolü inkar edilemeyecek kadar büyümüş ve bu rol her geçen gün biraz daha önem kazanmıştır. Uzaktan kontrol sistemleri ise bu bütünün en vazgeçilmez bileşenlerinden olmuştur. Buna paralel olarak da insanların kaygıları da her geçen gün artmaktadır. Bir kaç yıl öncesine kadar en teknolojik cihaz televizyon veya bilgisayarken artık çok farklı özelliklere sahip cep telefonu ile uzaktan kumanda edilebilen ev ve araba alarmları günlük yaşamın doğal bir parçası haline gelmiştir.

Bu tez akademik yönü kadar bir uygulama projesi olma özelliğini de bünyesinde barındırmaktadır. Bilgisayardaki paralel port kullanılarak kontrolü sağlanan model araba, bu arabaya monte edilmiş telsiz bir kamera ve internet üzerinden uzaktaki bir bilgisayardan bu kontrolün sağlanarak görüntünün elde edilmesi amaçlanmış ve gerekli teorik yaklaşımlarla beraber uygulaması başarıyla tamamlanmıştır.

Yapılan bu çalışmanın ülkemizde uzaktan kontrol edilebilen robot uygulamalarının yaygınlaştırılmasında ve gelecekte de farklı projelerle insanoğlunun hizmetine sunulmasında katkıda bulunmasını dilerim.

Hayatım boyunca edindiğim değer yargılarımı ve eğitimimi borçlu olduğum aileme, başta Yrd.Doç.Dr Nevcihan Duru olmak üzere tüm hocalarıma, yüksek lisans eğitimim ve tez çalışmam boyunca desteğini esirgemeyen eşime teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖZET	ii
ABSTACT	iii
ÖNSÖZ ve TEŞEKKÜR	iv
İÇİNDEKİLER	v
SEMBOLLER LİSTESİ ve KISALTMALAR	vii
ŞEKİLLER LİSTESİ	viii
TABLolar LİSTESİ	x
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. TCP/IP	8
2.1. Temel TCP/IP	8
2.2. İnternet Adresleri	8
2.3. Protokoller	9
2.4. Yönlendirme	10
2.5. Uzaktan Bağlantı Kurulması	11
BÖLÜM 3. VERİ İLETİMİ ve PARALEL PORT	12
3.1. Veri İletimine Genel Bakış	12
3.2. Paralel Veri İletimi Nedir?	12
3.2.1. Paralel veri iletiminin avantajları	13
3.2.2. Paralel veri iletiminin dezavantajları	13
3.3. Paralel Port	13
3.3.1. Veri portu	14
3.3.2. Durum portu	15
3.3.3. Kontrol portu	15
3.4. Paralel Port Sinyalleri	16
3.4.3. Meşgul (busy) pini (Pin 11):	17

3.4.4. Onay (acknowledge) pini (Pin 10):.....	18
3.4.5. Seçme (select) pini (Pin 13):.....	18
3.4.6. Sayfa sonu (paper end) pini (Pin 12):	18
3.4.7. Hata (error) pini (Pin 15):.....	18
3.4.8. Başlangıç (initialize) pini (Pin 16):.....	19
3.4.9. Giriş seçici (select input) pini (Pin 17):	19
BÖLÜM 4. İNTERNET ÜZERİNDEN ARAÇ KONTROLÜ.....	20
4.1. Giriş.....	20
4.2. Tekerlekli Gezer Robot Kinematığı.....	22
4.3. Uygulamanın Adımları ve Gelişim Süreci	34
4.3.1. Uygulamanın bileşenleri.....	35
4.3.2. Uygulamanın gelişim adımları	36
SONUÇ ve ÖNERİLER	56
KAYNAKLAR	58
EKLER	60
ÖZGEÇMİŞ	68

SEMBOLLER LİSTESİ ve KISALTMALAR

D : Aks uzunluğu

L : İki aks arası mesafe

Φ : Tekerleklerin dönme açısı

ω : Açısal hız

θ : Hareket açısı

AGP : Accelerated Graphics Port (Hızlandırılmış Grafik Portu)

BIOS : Basic Input Output System (Temel Giriş Çıkış Sistemi)

CPU : Central Processing Unit (Merkezi İşlem Birimi)

IP : Internet Protocol (İnternet Protokolü)

I/O : Input Output (Giriş Çıkış)

LAN : Local Area Network (Yerel Ağ Bağlantısı)

LPT : LinePrinter (Satır Yazıcı)

PCI : Peripheral Component Interconnect (Yan Bileşen Bağlantısı)

RAM : Random Access Memory (Rasgele Erişimli Bellek)

RF : Radio Frequency (Radyo Frekansı)

ROM : Read Only Memory (Salt Okunur Bellek)

TCP : Transmission Control Protocol (İletim Denetimi Protokolü)

UDP : User Datagram Protocol (Kullanıcı Veri Bloğu iletişim Protokolü)

USB : Universal Serial Bus (Evrensel Seri Yol)

VNC : Virtual Network Computing (Sanal Ağ İşlemesi)

WMR: Wheeled Mobile Robot (Tekerlekli Gezer Robot)

ŞEKİLLER LİSTESİ

Şekil 1.1. Kullanıcının internet üzerinden araca ulaşımı	3
Şekil 1.2. Web arayüzü	3
Şekil 1.3. NOMAD – Gezegen gezgini	4
Şekil 1.4. CRASAR arama ve kurtarma robotu	4
Şekil 1.5. İnsan eli hareketine sahip robot kolu modeli	5
Şekil 1.6. Bir cerrah robot	5
Şekil 3.1. Lojik 0 ve lojik 1	12
Şekil 3.2. Paralel port sinyalleri	16
Şekil 3.3. Paralel portun yapısı	16
Şekil 4.1. Ackerman tekerlekli gezer robot modeli	24
Şekil 4.2. Ackerman dönme geometrisi	26
Şekil 4.3. Aracın gerçek ölçüleri	28
Şekil 4.4. Hesaplamalar sonucu elde edilen değerler	29
Şekil 4.5(a). Aracın kinematik modelinin Matlab simülasyonu	33
Şekil 4.5(b). Aracın yörüngesi	34
Şekil 4.6. Uygulamanın genel yapısı	36
Şekil 4.7. RF ile gönderilen sinyal	37
Şekil 4.8. Arabanın genel yapısı	38
Şekil 4.9. Transistörlü devre blok şeması	38
Şekil 4.10. Yeni kumanda	39
Şekil 4.11. Aracın geri – sağ hareketi ve porttan okunan değer	42
Şekil 4.12. Aracın ileri – sol hareketi ve porttan okunan değer	42
Şekil 4.13. Aracın ileri hareketi ve porttan okunan değer	43
Şekil 4.14. Aracın ileri – sağ hareketi ve porttan okunan değer	43
Şekil 4.15. Aracın geri – sol hareketi ve porttan okunan değer	44
Şekil 4.16. Aracın geri hareketi ve porttan okunan değer	44
Şekil 4.17. Kameranın sağ hareketi ve porttan okunan değer	46
Şekil 4.18. Kameranın sol hareketi ve porttan okunan değer	47

Şekil 4.19. Geliştirilen programın arayüzü	47
Şekil 4.20. Kameranın alıcısı ve Dazzle bağlantısı	48
Şekil 4.21. Kamerayla ilgili ayarlar	49
Şekil 4.22. Görüntü biçiminin seçilmesi	49
Şekil 4.23. Görüntünün çözünürlüğünün ayarlanması	50
Şekil 4.24. Kameranın kontrolü	51
Şekil 4.25. İki yönlü röle kullanılarak yönsüz röle tasarımı	51
Şekil 4.26. Geliştirilen model araba	52
Şekil 4.27. VNC'nin platform bağımsızlığı	53
Şekil 4.28. VNC arayüzü	54
Şekil 4.29. VNC viewer ayarları	55



TABLULAR LİSTESİ

Tablo 3.1. Paralel port pinlerindeki sinyaller ve giriş – çıkış durumları	15
Tablo 4.1. Arabanın hareketine ilişkin bit gösterimi	39



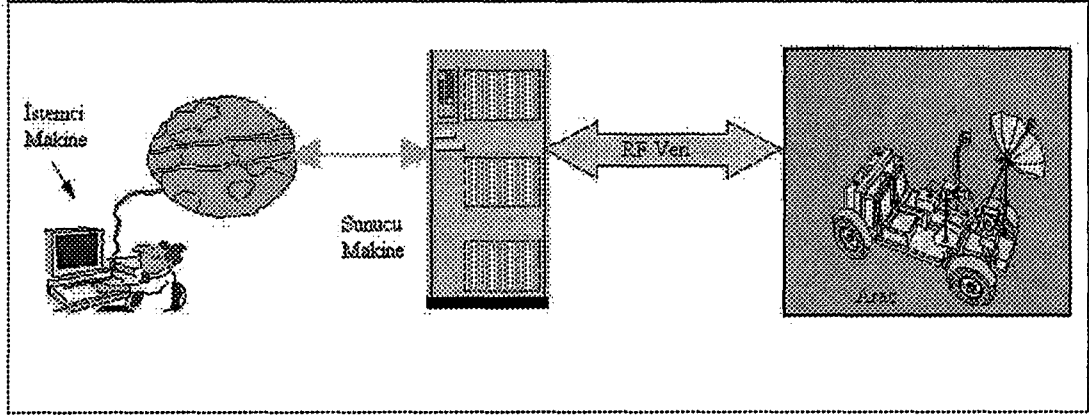
1. GİRİŞ

Uzaktan kontrol edilebilen ve farklı amaçlarda kullanılan robot uygulamaları, bilgisayar bilimlerinin son yıllarda geliştirdiği ve insanlığın hizmetine sunduğu, geniş kullanım alanları bulunan popüler bir konu olmuştur.

Robot teknolojisinin gelişimini ve literatürde yapılmış çalışmaları incelediğimizde her geçen yıl nasıl hızlı bir büyüme gösterdiğini görebiliriz. 1946'da, J. Presper Eckert ve John Mauchly, Pennsylvania Üniversitesi'nde ilk elektronik bilgisayar olarak bilinen ENIAC isimli bilgisayarı geliştirmiştir. Whirlwind adlı bir başka bilgisayar, MIT'de ilk olarak bir bilimsel problemi çözmüş ve 1948'de MIT'den Norbert Wiener Elektronik, Mekanik, ve Biyolojik sistemlerin denetim ve iletişimini inceleyen, "Sibernetik" başlıklı kitabı yayınlanmıştır. 1951'de Raymond Goertz, ABD Atom Enerjisi Komisyonu için uzaktan işletilen bir kol tasarlamıştır. 1954'te George Devol, programlanabilir genel amaçlı robotu tasarlamış ve patent başvurusunu yapmıştır. 1956'da G. Devol ve Joseph F. Engelberger, Unimation Inc. adlı dünyanın ilk robot firmasını kurmuş ve 1958'de satış amaçlı ilk ticari robot üretilmiştir. 1959'da MIT'de servomekanizma laboratuvarında robot kullanılarak bilgisayar destekli üretim amaçlı bir gösteri yapılmış, 1959'da Planet firması, ilk genel amaçlı ticari robotu pazarlamaya başlamıştır. 1960'da ise Harry Johnson ve Veljko Milenkovic'in tasarladığı Versatran isimli robot satışa sunulmuştur. Unimation robotlarının adı Unimate Robot sistemleri olarak değiştirilmiş ve 1962'de General Motors ilk kez bir endüstriyel robotu (Unimate) üretim hattında kullanmaya başlamıştır. Robot, sıcak parçaları kalıp döküm makinesinden alarak işlemek amacıyla kullanılıyordu. 1963'te bilgisayar denetimli, altı eklemlilik ilk yapay kol (Rancho arm) geliştirilmiştir. 1964'te dünyanın önde gelen bazı üniversite ve araştırma merkezlerinde (MIT, Stanford Araştırma Enstitüsü, Stanford Üniversitesi, Edinburgh Üniversitesi) ilk kez Yapay Zeka araştırmaları başlamış ve laboratuvarları

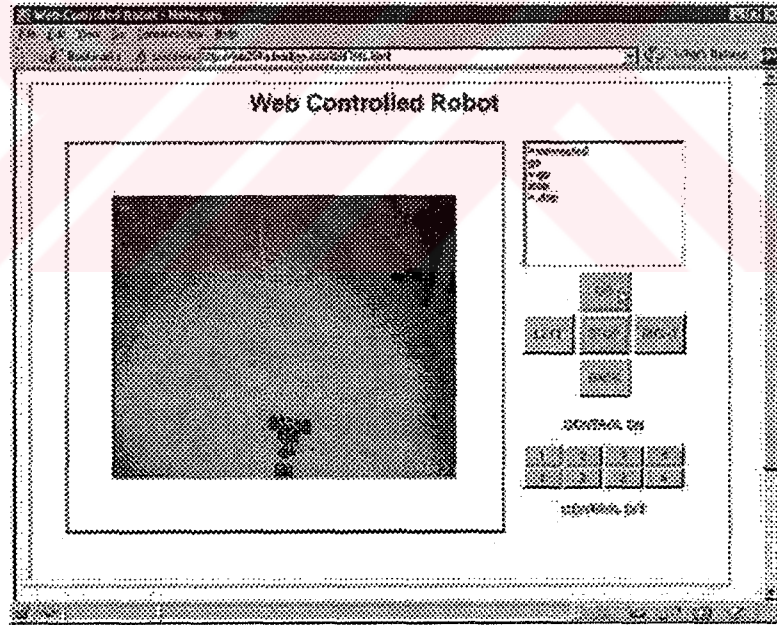
açılmıştır. 1965'te Dendral isimli ilk uzman sistem yazılımı geliştirilmiş, 1966'da nokta kaynağı yapan ilk robotlar üretilmiştir. 1967'de Japonya, ilk kez robot ithal ederek robot teknolojisini kullanmaya başlamıştır. 1968'de Marvin Minsky, on ayaklı ahtapot benzeri bir robot geliştirmiş ve aynı yıl Stanford Araştırma Enstitüsü'nce Shakey isimli ve görme yeteneği olan ilk gezer robotla bu teknolojiye ilklerden olmuşlardır. 1970'de Stanford Üniversitesi'nce bir robot kol geliştirmiş ve bu robot kol Stanford Kolu adıyla araştırma projelerinde bir standart olarak yerleşmiştir. 1973'te Richard Hohn, Cincinnati Milacron Corporation adına ilk mini-bilgisayar denetimli robotu geliştirmiş ve geliştirilen bu robot T3 (The Tomorrow Tool) olarak adlandırılmıştır. 1974'te Stanford kolunu geliştiren Prof. Scheinman Vicarm Inc. isimli bir firma kurarak mini bilgisayar kullanan robot kollarının pazarlamasına başlamış ve 1974'te dokunma ve basınç algılayıcıları kullanarak küçük parçaların montajını yapabilen ilk robot, üretim hattında kullanılmaya başlanmıştır. 1976'da Viking 1 ve Viking 2 uzay araçlarında robot kollar kullanılmış, 1977'de ASEA isimli Avrupalı bir robot firması, iki ayrı boyutta robot üretimine başlamıştır. 1977'de Star Wars sinema filmindeki C3PO ve R2D2 robot animasyonlarıyla robot kelimesi geniş insan kitlelerine yayılmasını sağlamış ve 1978'de Puma isimli robot üretilerek ve pazarlanmaya başlamıştır. 1979'da Stanford Cart isimli gezer robot, üzerine monte edilmiş bir kameradan alınan görüntüleri kullanarak engellerle dolu bir odayı engelleri aşarak boydan boya geçmeyi başarmıştır. 1984'te SRI tarafından Shakey'den daha fazla gelişmiş bir gezer robot olan Lakey üretilmiştir. 1990'da ABD'de 12 dolaylarında robot firması görülürken, Japonya'da kırktan fazla robot firması kurulmuştur. 1993 – 1994'te önceki robotlara göre ucuz maliyetli Erratic ve Pioneer1 isimli gezer robotlar üretilmiş ve 1998'de de robot oyuncak Furby piyasaya çıkmıştır (Enden, 2003).

İnternet üzerinden robot kontrolü üzerine 1999 ve 2000 yıllarında Bradley Üniversitesi'nde yapılan iki proje bu alandaki ilk çalışmalardandır. Bu robot RS232 iletişim protokolünü destekleyen bir mikro denetleyiciye sahiptir. Denetleyici bilgisayardan gelen bilgiyi 8 bitlik veri yolu ile harici motora ve servo mekanizmasına iletilmiş, böylece robotun hareketi gerçekleştirilmiştir. Şekil 1.1'de robotun iletişimine ait gösterim yapılmıştır.



Şekil 1.1. Kullanıcının internet üzerinden araca ulaşımı

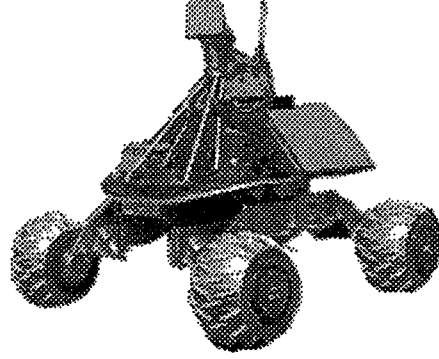
Robotun bir bileşeni de istemci ile araç arasındaki iletişimi kuran sunucudur. Sunucu, Java Appletler sayesinde gelen HTML dokümanını yorumlama yetisine sahiptir. Bu çeşit sunucular Netscape, Microsoft ve SUN Microsistem sunucularıdır. Ayrıca harici portlarla iletişim kurabilirler (Huggins, Malinowski, 2000).



Şekil 1.2. Web arayüzü

Uygulama, NASA ve Carnegie Mellon Üniversitesi Robotik Enstitüsü (Pittsburgh, PA) tarafından geliştirilmiştir. Amacı, insanoğlunun çalışamayacağı coğrafi bölgelerde meteorolojik araştırma yapmaktır.

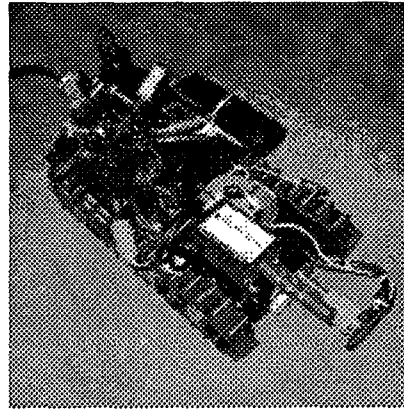
Dört tekerleđi olan ve ufak bir arabayı andıran bu robot saatte yaklaşık 20 km. hızla hareket edebilmektedir. İlk araştırmasını Atacama ölu'nde yapmıştır. Bu robot 360° görmeyi sađlayan kamera ünitesine, iki boyutlu görmeyi sađlayan kamera yerleşimine ve ölçüm amaçlı pek çok algılayıcıya sahiptir.



Şekil 1.3. NOMAD – Gezegen gezgini

Araştırmacılar bu robotu geliştirerek Antarktika'da ve Ay'da da araştırma yapmayı planlamışlardır (Siuru, 1997).

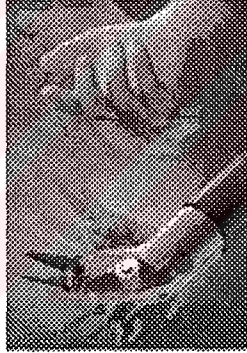
Güney Florida Üniversitesi'ndeki CRASAR (Center for Robot-Assisted Search and Rescue) biriminde geliştirilen bir robot, 11 Eylül'deki terörist saldırısından sonra New York'da göçük altında insan arama ve kurtarma amaçlı kullanılmıştır.



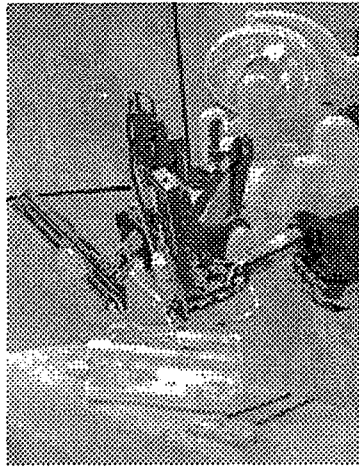
Şekil 1.4. CRASAR arama ve kurtarma robotu

Uzaktan kontrol edilebilen bu robot, üzerindeki algılayıcılar sayesinde kurbanın yaşayıp yaşamadığı bilgisini alabilmekte, tene dokunan özel amaçlı algılayıcılar, doku altında kan dolaşımı olup olmadığını anlayabilmekte, tene dokunmadığı durumlarda ise ortamdaki karbondioksit (CO₂) miktarını ölçerek içeride bir canlı olup olmadığı bilgisini dışarıya gönderebilmektedir. Bu robotun yardımıyla göçük altındaki birçok hayat kurtarılmıştır (Port, 2003).

Günümüzde bazı cerrahlar, özellikle kalp ameliyatlarında, uzaktan kontrol edilebilen robotlardan faydalanmakta ve gelecekte de bu sayının gittikçe artması beklenmektedir. Bu robotlar, sahip oldukları “parmaklarla” hastanın göğsünde sadece bir kurşun kalem genişliğinde delikler açarak ameliyatın çok daha az komplikasyonla gerçekleşmesini sağlamaktadırlar. Şekil 1.5 ve Şekil 1.6’da cerrahi amaçlı kullanılan robotlar gösterilmiştir.



Şekil 1.5. İnsan eli hareketlerine sahip robot kolu modeli



Şekil 1.6. Bir cerrah robot

Arařtırmacılar teknolojinin de gelişmesiyle birlikte 10 yıl içinde birçok ameliyatta robotların yaygın olarak kullanılacağını belirtmektedirler.

Bu tezin bir diğeri önemi ise mekatronik olma özelliğidir. Mekatronik; bilgisayar, elektronik, mekanik, hidrolik, pnömatik, algılayıcılar tekniğı, servo kontrol tekniğı, lojik kontrol ve robotik tekniklerinin hepsinin veya birkaçının bir arada kullanıldığı bir alandır ve bu alanda, ülkemizde de, bazı özel ve kamu üniversitelerinde, ön lisans ve mühendislik düzeyinde eğitim vermeye başlanmıştır. Mekatronik ilk kez 1960'ların sonunda Japonya'nın Yaskawa Elektrik Şirketi'nde görevli bir mühendis tarafından elektrik motorlarının bilgisayarla kontrolünün sağlanması için kullanılmıştır. Temelde ise "mekanik" ve "elektronik" kelimelerinin uygun bir şekilde parçalanması ve bu parçaların birleştirilmesi ile bu kelime ortaya çıkmıştır.

Bu kavram Japonya'dan yola çıkarak tüm dünyaya yıllar ilerledikçe yayılmaya ve yerleşmeye başlamıştır. Dünyada artan uygulamalarıyla gelişen mekatronik, ön lisans, lisans ve lisans üstü programlarıyla da üniversitelerde desteklenmektedir. Türkiye'de mekatronik eğitimi, son 5 yılda akademik kabul görerek ilerlemektedir (Gürbüz, 2004).

Projenin gelişim aşamalarında kullanılan TCP/IP, Paralel Port' un yapısı ve bu port kullanılarak verinin bir vericiye iletilmesi gibi temel teknikler bu bölüm içinde açıklanmaktadır.

İlk bölümde, uzaktan kontrol edilebilen robot çalışmalarının önemi, bilgisayar bilimlerindeki gelişmeler, robot uygulamalarının bu gelişim içindeki olağan süreci, üniversitelerde ve endüstride yapılmış robot uygulamaları anlatılmıştır.

İkinci ve üçüncü bölümlerde bu tezin hazırlanışı sırasında kullanılan iki önemli teknoloji, TCP/IP ve paralel port programlamaya yer verilmiş ve temel kavramlar anlatılmıştır.

Dördüncü ve en önemli bölümde, öncelikle tekerlekli bir gezer robotun kinematik modeli Ackerman denklemlerinden faydalanılarak oluşturulmuştur. Elde edilen denklemler, arabanın ölçüleri ve test değerleri kullanılarak robotun matematiksel modeli çıkartılmış, çıkartılan bu model Matlab programı kullanılarak simüle edilmiştir. Daha sonra, paralel port üzerinden gönderilen verilere göre kontrol edilebilen, üzerine eklenmiş telsiz kamera ile çevresini görebilen ve bu görüntüyü internet üzerinden uzaktaki başka bir bilgisayara gönderebilen bir model aracın gelişimi adım adım anlatılmıştır.

Son bölümde ise uzaktan kontrolün önemi vurgulanmış, hazırlanan bu projenin özel amaçlara göre nasıl farklılaştırılabileceği üzerinde durulmuştur.

Bu çalışmada, Türkiye’de yapılan robotik çalışmalarına katkıda bulunmak ve yakın gelecekte günlük hayatımızın olağan bir parçası haline gelecek robot uygulamalarına farklı bir açıdan ışık tutmak amaçlanmıştır.

2. TCP/IP

2.1. Temel TCP/IP

TCP/IP, ilk defa ABD'de ARPA NET (Advanced Research Projects Agency Network) adı altında, askeri bir proje olarak geliştirilmiştir. Önceleri askeri amaçlı düşünülen proje, daha sonra üniversiteler tarafından kullanılmaya başlanmış, ardından ABD'nin dört bir yanında birbirinden bağımsız geliştirilen ağlar, tek bir omurga altında NSF NET olarak adlandırıldı ve ulusal boyutu aşarak dünyaya yayılmıştır. İnternet'in doğuşu da bu tarihe denk gelir(Çağiltay, 1994).

İnternete bağlı olan her makinenin bir adresi vardır. Bu adres sayesinde bir bilgisayardan diğerine ulaşmak mümkün olur. İnternet adresi 4 bayttan (32 bit) ibaret olup, yazılırken her bayt arasına bir nokta konulur

2.2. İnternet Adresleri

İnternet üzerinde 3 sınıf adres vardır. Avrupa'da RIPE (Réseaux IP Européens) tarafından dağıtılan bu adresler daha sonra o yerin ağ yöneticisi tarafından uygun şekilde bölünebilir. Bu bölümlendirmeye "subnetting" işlemi adı verilir. Bu sayede ağlar gruplaşarak her birisinin yönetimi bağımsız hale getirilmiş, aynı zamanda da kısıtlı olan IP adresleri daha verimli bir şekilde kullanılmış olur.

Üç çeşit internet adresi aşağıda belirtilmiştir:

- A sınıfı internet adresi: Adresin ilk baytı 1 ile 126 arasında bir sayıdır. Bu adrese verilen yetkiyle toplam 2^{24} makine adreslenebilir.

- B sınıfı internet adresi: Adresin ilk baytı 128 ile 192 arasında bir sayıdan oluşur. Bu adresin subnetlere bölünmesiyle 65534 farklı makine adreslenebilir.

- C sınıfı internet adresi: Adresin ilk baytı 192 ile 223 arasındadır. C sınıfı bir adresi bloğuyla bağlı 254 bilgisayar adreslenebilir.

Bir internet adresi iki kısımdan meydana gelir: ağ adresi ve düğüm adresi. Her makine, bir ağ üzerinde bulunur ve bu adres "ağ adresi" olarak adlandırılır.

Yukarıda söz edilen alt ağlara ayırma, çok sıkça kullanılan bir yöntem olup getirdiği birtakım kolaylıklar vardır. Büyük bir iletişim ağı alt ağlara ayrılırsa, kontrol edilmesi daha kolaylaşır.

2.3. Protokoller

İnternet protokollerinin yaygınlaşmasındaki en önemli etkenlerden biri herhangi bir ağ donanımına ya da firmaya özel olmamalarıdır. İşletim sistemi ya da ağ bağlanma yöntemi ne olursa olsun internete bağlamak mümkündür.

İnternet protokollerinin temeli İnternet Protocol (IP)'dir. İnternet üzerinde yönlendirme (routing) gibi temel ağ işlemlerinin gerçekleştirildiği protokol katmanıdır. IP paketlerinin her biri kendi başlarına aradaki ağ cihazları tarafından yönlendirilerek paket içinde belirtilen adrese ulaştırılır. Bu sırada fiziksel ağ farklılıklarından kaynaklanan paket parçalanmaları (fragmentation) ve bunların yeniden birleştirilmeleri aradaki ağ cihazlarının aşırı yüklenmelerini önlemek gibi görevler de IP katmanı tarafından gerçekleştirilir. IP bağlantı temelli (connection oriented) bir ağ protokolü değildir. Bunun yanı sıra IP, paketlerin içeriklerinin doğruluğunu da garanti etmez. IP katmanı sadece başlık kısmında oluşan hataları bulur ve düzeltir (Feit, 1999).

Kullanıcı uygulamalarının IP katmanına doğrudan ulaşmaları yoktur. IP ve uygulama programları arasındaki bağlantıyı sağlayan iki protokol vardır: Transmission Control

Protocol (TCP) ve User Datagram Protocol (UDP). TCP, İnternet protokolleri arasındaki en önemli protokollerden biridir. TCP, IP katmanının sağlamadığı bağlantı temelli, güvenilir servisi sağlar. TCP kullanarak ağ üzerinden veri aktaran programlar, bir dosyadan okuyormuş ya da yazıyormuş gibi güvenle ağ bağlantısını kullanabilirler. Arada oluşan hatalar TCP tarafından onarılır. IP protokol katmanına uygulama programları doğrudan erişemediklerinden, hata kontrolü ve bağlantı gerektirmeyen ya da bu işlemleri kendileri gerçekleştirmek isteyen uygulamalar UDP kullanarak ağ üzerinden iletişim sağlarlar (Forouzan, 2000).

Yukarıdaki protokollerin yanı sıra İnternet standardı olmuş birçok uygulama protokolü de vardır. Bunlar arasında, TELNET, FTP, SMTP ve HTTP gösterilebilir. İnternet üzerinde paketler son makineye ulaştıktan sonra, ilgili uygulama programına ulaşabilmesi için port adı verilen sanal numaralar kullanılır. Servis veren uygulamalar, önceden belirlenmiş standart port numaraları kullanırlar. Örnek olarak SMTP 25 numaralı TCP portunu, talk ise 518 numaralı UDP portunu kullanır (Forouzan, 2000).

2.4. Yönlendirme

Birden fazla ağa fiziksel bağlantısı bulunan ve ağlar arası bilgi paketlerinin geçişine olanak tanıyan araçlara yönlendirici (router) adı verilir.

İki ayrı ağda yer alan makinelerin haberleşmesine imkan tanıyan yönlendiriciler, kendilerine gönderilen paketleri bir ağdan diğerine iletirler. Herhangi bir ağ içerisinde yer alan makineler aynı IP ağ içerisinde bulunan makinelerle haberleşebilirler. Başka adreslere ulaşabilmek için bulunulan ağ dışındaki ağlarla bağlantısı olan ve o ağ üzerinde bulunan bir makine ile haberleşmeleri gereklidir. Bu makine bilgisayar tanımlarında yönlendirici (gateway) olarak verilir.

2.5. Uzaktan Baęlantı Kurulması

İnternet üzerinde kesinlikle yalnız olunmadığına göre, kimi durumlarda kullanıcı kendi sistemine dışındaki bir başka sistemden girmek isteyebilir. Bunun için uyarlanmış çeşitli teknikler ve yardımcı programlar mevcuttur. TCP/IP ise daha önce de belirtildiği gibi yerel aę veya İnternet üzerinde rahatlıkla kullanılabilen bir protokoldür.



3. VERİ İLETİMİ ve PARALEL PORT

3.1. Veri İletimine Genel Bakış

Bir noktadan diğer bir noktaya sayısal, başka bir deyişle ikili bilgilerin iletilmesine "veri iletimi" denir. Veri iletim sistemleri; bilgisayar - bilgisayar, bilgisayar - terminal veya bilgisayar - aygıt arasındaki veri iletimlerini gerçekleştirir. Sayısal hale dönüştürülebilen ses, görüntü gibi analog bilgilerin iletilmesi de veri iletimi ile gerçekleştirilir. Sayısal tekniklerin, yüksek verimliliğin yanı sıra maliyetlerinin de düşük olması veri iletiminde kullanılmasının en önemli sebeplerindendir. Sayısal sinyaller, her biri bir voltaj seviyesiyle tanımlanan ve birbirinden farklı iki durumdan oluşan "ikili darbe" lerdir. Bu darbeler iki seviye arasında değişir. Bu seviyeler sırasıyla; "0" veya "düşük" ve "1" veya "yüksek" seviyeleridir. İkili 0 seviyesi genellikle 0 Volt veya ground, binary 1 seviyesi de genellikle +5 Volt olarak tanımlanır. Ayrıca kullanılan sisteme göre bunların dışında başka voltaj değerleri de kullanılabilir.



Şekil 3.1. Lojik 0 ve lojik 1

3.2. Paralel Veri İletimi Nedir?

Seri veri iletimi yöntemi gibi paralel veri iletimi de veri iletiminde kullanılan temel yöntemlerden birisidir. Sayısal olarak kodlanmış bilginin tüm bitleri aynı anda transfer ediliyorsa buna "paralel veri iletimi" denir.

3.2.1. Paralel veri iletiminin avantajları

Seri veri iletiminde, bir defada bir karakterin sadece bir biti iletilir. Alıcı makine doğru haberleşme için karakter uzunluğunu, başla – bitir (start – stop) bitlerini ve iletim hızını bilmek zorundadır. Paralel veri iletiminde, bir karakterin tüm bitleri aynı anda iletildiği için başla – bitir bitlerine ihtiyaç yoktur. Dolayısıyla doğruluğu daha yüksektir.

Paralel veri iletimi, bilginin tüm bitlerinin aynı anda iletimi sebebiyle çok hızlıdır. Paralel veri iletiminde, bir defada bir karakter iletildiği için bilgi iletim hızı "cps" (Character Per Second – Saniyedeki Karakter Sayısı) olarak bilinir. Sıradan bir paralel portun veri iletim hızı yaklaşık 100.000 cps'dir. Benzer şekilde, sıradan bir seri portun veri iletim hızı yaklaşık 9600 bps'dir. Pek çok paralel port uygulamasında bir karakter yaklaşık olarak birkaç mikro saniyede alıcı devreye iletilebilir.

3.2.2. Paralel veri iletiminin dezavantajları

Veri iletiminde kullanılan portlardaki kablolar pahalıdır. Telefon ve seri iletişim bağlantılarında genellikle ikili hatlar kullanılır. Tipik paralel port bağlantıları için genellikle 25'li hatlar kullanılır. Ancak bunların sadece 8 tanesi veri hattıdır. Geriye kalan diğer hatlardan kontrol sinyalleri iletilir. Çok kablolu hatları pahalı olması, paralel iletişiminin kısa mesafelerde tercih edilmesinin sebeplerinden biridir.

3.3. Paralel Port

Bir bilgisayardaki anakartın en önemli bileşenleri veri yolları (Portlar), CPU, RAM, BIOS, yonga seti, ROM, I/O devreleridir. Anakart, sistemin çalışmasını organize eder. Bu organizasyon anakart üzerinde bulunan yonga setleri sayesinde gerçekleşir. Anakart üzerinde bulunan ve bilgisayara veri giriş/çıkış için kullanılan pinlere veya bağlantı noktalarına port denir. Çevre birimlerini programlamak ve onlardan bilgi

almak için kullanılan elektriksel yollara da port denir. Örneğin: Paralel Port (LPT), Seri Port, AGP, PCI Portu, USB portu gibi.

Bir paralel port DB-25 olarak adlandırılan 25-pinli adaptörden oluşur. Adaptör, erkek (male) pinler veya dişi (female) pinlerdir. Çoğu zaman paralel portlara LPT portu da denilmektedir. LPT, LinePrinTer sözcüğünden alınmıştır. Bunun sebebi en çok yazıcıları bağlamak için kullanılması gerçeğine dayanmaktadır. Ancak, son yıllarda paralel portlar bilgisayara başka tip aygıtları bağlamak için de kullanılmaktadır.

Paralel portlar isimlerini, verilerin porttan paralel bir biçimde, yani bir seferde bir bayt olarak iletilmesi gerçeğinden alırlar. Port 8 adet veri hattı içerir ve baytın her biti, bayttaki diğer bitlerle hemen hemen aynı anda farklı bir hattan iletilir. Paralel portlar LPT1, LPT2 gibi isimlendirilir.

Paralel portlar başlangıçta tek yönlü bir yapıda geliştirilmişler, veriler çevre birimlerine iletebilmiş, fakat ters yönde iletim gerçekleştirilememiştir. Çift yönlü paralel port 1987'de ortaya çıkmış ve çevre birimlerinin bilgisayar ile ters yönde de iletişim kurmaları sağlanmıştır, örneğin, bir yazıcı bilgisayara durumuyla ilgili bilgi gönderebilmiştir (Axelson, 1999).

Paralel porttaki her bir pinin dış aygıtlarla anlaşmasını sağlayan özel görevi vardır. Görevler 3 temel kısma ayrılabilir; veri (data) portu, durum (status) portu ve kontrol (control) portu.

3.3.1. Veri portu

Paralel port üzerinde Veri portuna ait 8 adet (D0-D7) pin vardır. Bu port paralel portun taban adresini kullanır. 8 tane Veri pini olduğundan 8 bitlik veri çıkışı almak mümkündür. Yani bu 8 tane pinin "1" ya da "0" değerlerini alması ile veri akışı sağlanmış olur. Veri portu normalde veri çıkışı için kullanılmaktadır. Fakat bazı özel ayarlar yaparak, eğer bilgisayar da destekliyorsa veri girişi yapmak da mümkün olabilir.

3.3.2. Durum portu

Durum portu sayesinde, 15 - 13 - 12 - 11- 10 numaralı pinlerden, 5 bit sayısal giriş yapılabilir. Bu 5 pine herhangi bir müdahale bulunmadığında "1" değeri okunur.

Bu pinlerin tamamı "giriş" amaçlı kullanılır.

3.3.3. Kontrol portu

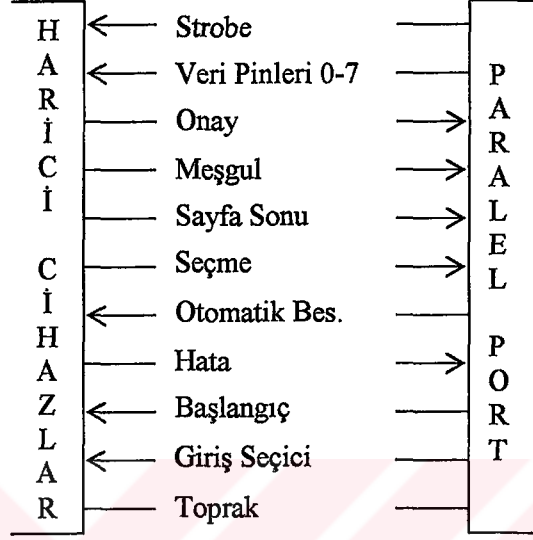
Kontrol portunu hem giriş hem de çıkış için kullanmak mümkündür. Paralel port üzerinde kontrol amaçlı 4 tane pin vardır. Bu pinlere veri göndermediğimiz zaman durumu "1" dir. Bu pinlerden bir tanesine veri gönderildiğinde o pinin değeri "0" olacaktır. Veri ve Durum pinlerinin yetmediği zamanda Kontrol pinleriyle çıkış ya da giriş almak mümkündür. Paralel port pinlerindeki sinyaller Tablo 3.1'de gösterilmiştir.

Tablo 3.1. Paralel port pinlerindeki sinyaller ve giriş – çıkış durumları

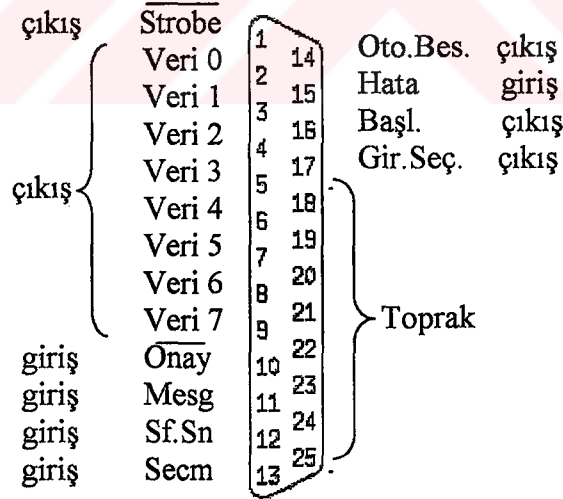
Sinyal Adı	Bit	Pin	Yön
- Strobe	K0	1	Çıkış
+ Veri Biti 0	V0	2	Çıkış
+ Veri Biti 1	V1	3	Çıkış
+ Veri Biti 2	V2	4	Çıkış
+ Veri Biti 3	V3	5	Çıkış
+ Veri Biti 4	V4	6	Çıkış
+ Veri Biti 5	V5	7	Çıkış
+ Veri Biti 6	V6	8	Çıkış
+ Veri Biti 7	V7	9	Çıkış
- Onay	D6	10	Giriş
+ Meşgul	D7	11	Giriş
+ Sayfa Sonu	D5	12	Giriş
+ Giriş Seçici	D4	13	Giriş
- Otomatik Besleme	K1	14	Çıkış
- Hata	D3	15	Giriş
- Başlangıç	K2	16	Çıkış
- Seçme	K3	17	Çıkış
Toprak	-	18-25	Toprak

3.4. Paralel Port Sinyalleri

Örnek olarak verilecek olan IBM' in yazıcı (printer) portunda, 25 pinli bağlantı kullanılır. Bu bağlantılar ve fonksiyonları Şekil 3.2 ve Şekil 3.3'de gösterilmiştir.



Şekil 3.2. Paralel port sinyalleri



Şekil 3.3. Paralel portun yapısı

Yazıcı ile bilgisayar arasındaki paralel iletişimde sadece bilgi iletimi yapılmaz. Bunun dışında yazıcı devresinden de birçok bilgi, bağlı bulunduğu bilgisayara paralel hat üzerinden gönderilir. Yazıcıdan bilgisayara gönderilen bu sinyallere Yazıcı Geribesleme denir. Bu sinyallerle yazıcının görevini yapmaya hazır olup olmadığı, yazıcıda hata olup olmadığı, yazıcının doğru işlem yapıp yapmadığı bilgisayara iletilmiş olur.

3.4.1. Veri pinleri (Pin 2-9):

Yazıcıya gönderilecek karakterler öncelikle 8'li veri hattına yüklenir. Bilindiği gibi; ASCII kodları 1 bayttan yani 8 bitten oluşur. ASCII kodlarının her bir biti için bir data hattı kullanılır. Gönderilen bilgi sinyalleri standart TTL voltajıdır. Lojik 1 bilgisi +5V, lojik 0 bilgisi 0 V ile ifade edilir.

3.4.2. Strobe pini (Pin 1):

Veri hatlarına sadece bilgi bitlerini yüklemek yazıcının o karakteri yazması için yeterli değildir. Veri bitleri sürekli olarak değişir. Sekizinin de aynı anda porta verildiğini garanti edilmesi gerekir. Garanti edilmezse yazdırma işleminde hatalar oluşur.

Veri bitlerinin sekizi birden aynı anda porta verildiği durumda, bilgisayarın bu hattan lojik 0 bilgisi gönderir. Eğer veri bitlerinin sekizi birden aynı anda porta verilemiyorsa, bilgisayar lojik 1 sinyalini bu hattan yazıcıya gönderir. Bilgisayar bu sinyali kullanarak eşzamanlılığı garanti eder. Bu sinyal, karakterin yazılabileceğini yazıcıya bildirir.

3.4.3. Meşgul (busy) pini (Pin 11):

IBM PS/2 serisinde paralel iletişimle bir karakterin gönderilmesi için minimum 2 ms. gereklidir. Yani paralel hatla 1 sn.'de minimum 500.000cps gönderilebilir. Ancak bir yazıcı aynı hızda bu bilgileri kağıt üzerine basamaz. Yazıcı bilgisayardan strobe sinyalini gönderilebilir. Ancak bir yazıcı aynı hızda bu bilgileri kağıt üzerine

basamaz. Yazıcı bilgisayardan strobe sinyalini alır almaz karakter basma sürecini başlatır. Yazıcının yeni gelen bir bilgiyi hafızaya alması ve karakteri basması anında bilgisayara meşgul pini ile kısa bir meşgul sinyali gönderilir.

3.4.4. Onay (acknowledge) pini (Pin 10):

Acknowledge pini ile, yazıcıdan bilgisayara bir sinyal iletilir. Bu sinyal ile "Bir önceki karakter alındı, kağıda basıldı ve şimdi yeni karakter için yazıcı hazır durumdadır " mesajı verilir. Yeni karakter için yazıcı hazır duruma geldiğinde lojik 0, hazır olmadığı durumda ise lojik 1 sinyali bu hattan iletilir. Bilgisayar bu sinyali almadan yeni karakter göndermez, bir önce gönderdiği bilgiyi hatta tutar.

3.4.5. Seçme (select) pini (Pin 13):

Bu hat, yazıcının seçilip seçilmediğini gösterir. Yazıcının portla bağlantısı var ve yazıcı kullanıma hazır ise yazıcı, select hattından lojik 1 sinyali gönderir. Eğer bu hatta lojik 0 sinyali varsa bilgisayar paralel porta bilgi iletmez.

3.4.6. Sayfa sonu (paper end) pini (Pin 12):

Yazdırma işlemi sırasında en çok karşılaşılan problem kağıdın bitmesidir. Böyle bir durumda yazıcı bu hattan lojik 1 bilgisi gönderir. Bilgisayar da, yazıcıya bilgi akışını keser ve bilgisayar ekranında kağıdın bittiği mesajı görülür.

3.4.7. Hata (error) pini (Pin 15):

Bu sinyal yazıcıdan bilgisayara gönderilir. Yazıcıda bir hata oluştuğu, "Hata" adı verilen bu sinyal ile bilgisayara bildirilir. Ancak yeteri kadar hat olmadığından, yazıcıdaki problemin tam olarak ne olduğu ve nerede olduğu bilgisayar tarafından tespit edilemez. Hata olmadığı durumlarda lojik 1, hata olduğu durumlarda lojik 0 seviyesinde bir hata sinyali gönderilir. Ayrıca hata oluştuğu durumlarda busy (1), select (0) ve fault (0) sinyallerinin hepsi ikaz durumlarına geçerler.

3.4.8. Başlangıç (initialize) pini (Pin 16):

Initialize yazıcı hattı, yazıcının reset butonu tarzındadır. Bilgisayar yazıcıyı başlangıç konumuna geçirmek için bu sinyali lojik 0 seviyesine çeker. Böylelikle sanki yazıcı kapanıp tekrar açılmış gibi olur.

3.4.9. Giriş seçici (select input) pini (Pin 17):

Bazı yazıcılar bağlı bulunduğu bilgisayar tarafından açılabilir veya kapanabilir. IBM Giriş Seçici sinyali, yazıcıyı açıp kapatan düğmesini kumanda eder. Giriş Seçici sinyali lojik 1 seviyesindeyken bu düğme "Açık" konumundadır ve aktarılan bilgileri yazıcı alır, lojik 0 seviyesindeyken bu düğme "Kapalı" konumundadır ve aktarılan bilgileri yazıcı alamaz. Bu sistemde kullanılan ve bilgisayar kontrol edilen bu tip düğmelere "DIP (Dual Inline Pine – İkili Giriş Pini) anahtar" denir.

3.4.10. Otomatik besleme (auto feed) pini (Pin 14):

Bu sinyal bilgisayar tarafından üretilir. Yazdırılan satır bittikten sonra bir önceki satırın yazılması için kağıdı döndüren mekanizmayı kontrol eder. Bir satır yazılırken lojik 0 bilgisi gönderilir. Satır bittikten sonra yeni bir satıra geçilmesi için lojik 1 bilgisi gönderilerek yazıcıdaki bu mekanizma çalıştırılmış olur.

3.4.11. Toprak (ground) pini (Pin 18-25):

Bilgi taşıyan her bir paralel hat bir toprak ucu ile eşleştirilmiştir. Bunun amacı; her bir bilgi sinyali için referans seviye oluşturmaktır. Bu toprak uçlarına "dönüş" denir ve bunlar 0V seviyesindedir (Axelson, 1999).

4. İNTERNET ÜZERİNDEN ARAÇ KONTROLÜ

4.1. Giriş

Robot kelimesi ilk olarak 1920 yılında kullanılmış olsa da, robotlara ait ilk kavramlar ve robot benzeri ilk makinelere ait bilgiler M.Ö. 3000 yıllarına kadar uzanmakta. Eski Mısır, eski Yunan ve Anadolu uygarlıklarında otomatik su saatleri benzeri makinelerin geliştirildiği bilinmektedir Homeros' un İlyada eserinde insan yapımı kadın hizmetçilerden söz edilmektedir. M. Ö. 100 yıllarında yaşamış olan İskenderiye'li bir mühendisin, otomatik açılan kapılar, fiskiyeler vb. düzenekleri su ve buhar gücüyle çalıştırdığı eski kitaplarda yazılmıştır. Daha yeni çağlarda Leonardo da Vinci'nin yürüyen mekanik aslanı olduğu söylenmektedir. Bu süreç içinde özellikle Batı dünyasında iyi bilinmeyen El Cezeri'nin (M. S. 12. yüzyıl) robot teknolojisi konusunda çok sayıda ve zamanına göre çok ileri öneri ve uygulamaları bulunmaktadır(Erden, 2003).

Yakın zamanda yapılan bazı robot tanımları ise şu şekildedir;

“Normal koşullarda insanlara atfedilen işlevleri yapan ya da şekilsel olarak insana benzeyen otomatik bir düzenek” (Webster Sözlüğü 1993).

“Çeşitli işleri yapabilmek için programlanmış hareketlerle malzeme, parça, alet ya da özel cihazları taşımak için tasarlanmış çok işlevli, tekrar programlanabilir düzenek” (Amerika Robot Enstitüsü, 1979).

Günümüz koşullarında ve robot sıfatıyla anılan çok sayıda örneği incelediğimizde, bu tanımlardan birincisinin yetersiz olduğu, robot kavramlarının insan işlevleri ve

şekliyle tanımlandırılması ve sınırlandırılmasının yanlış olduğu açıktır. Bu nedenle, robot tanımı olarak, "canlılara benzer işlevleri olan ve davranış biçimleri sergileyen makineler" diyebiliriz. Robotların temel özellikleri olarak da işlevsel olarak kendi kendilerine yeter ve programlanabilir olmaları sayılabilir.

Robotlar, düşlediğimiz gibi bir insan görünümündeki robotlardan çok farklılar. İnsanın yalnızca koluyla kısmen benzerlik gösteren bu robotlar, günümüzde de üretimin yoğun ve hassas olduğu birçok fabrikada sıklıkla kullanılmakta ve monoton işleri insana bırakmadan sabırla yapmaktadırlar. Gelişen teknolojiyle birlikte zaman içerisinde robotlar yalnızca robot kol olmaktan çıkıp etrafını algılayabilen, etrafına tepki verebilen ve bir noktadan başka bir noktaya gidebilen makineler haline gelmişler, bu tip robotlar, gezer robotlar diye adlandırılmışlardır. Gezer robotların en güzel örneklerinden biri NASA'nın tasarladığı ve Mars'a araştırma yapmak için gönderdiği Sojourner'dır. Gezer robotlar, yalnızca uzay araştırmalarında değil, günlük hayatımız da birçok farklı uygulamada kullanılmaya başlanmıştır. Electrolux'ün Trilobite adlı gezer robotu buna bir örnek olarak verilebilir. Trilobite, insansız elektrik süpürgesi olarak çalışmak için tasarlanmış bir gezer robottur ve amacı, oda içerisinde dolaşarak, hiçbir yere çarpmadan, yerleri süpürmek ve enerjisi bittiği zaman da pilini şarj etmek şeklindedir. Trilobite, bu işleri yaparken insanlardan yardım almamaktadır. Gezer robotlar, bilinen taşıtlara benzeyen tekerlekli araçlar olabildiği gibi, daha doğal hareket sağlayan iki ya da daha çok bacaklı olabilir, ya da yüzeyde yürümek yerine değişik ortamlarda hareket etme becerisi gösterebilmektedir. Bu kapsamda diğer örneklerse; sualtı robotları, uçan robotlar, sürünen robotlar ve toprak altında hareket edebilen robotlardır. Robotlar, genel amaçlı bir tanımlama yapılırsa, canlıların işlev ve yaşam biçimlerini taklit eden ve programlanabilir yetenek ve zekaya sahip, gelişmiş ve çok disiplinli öğeler içeren makinelerdir. Bu tanımda kullanılan, "canlıların işlev ve davranış biçimleri" deyimini şöyle açıklanmaktadır; insanların işlevleri arasında; hareket etmek (yürümek, koşmak, sıçramak, vb.), iletişim (konuşmak, yazmak, resim yapmak, gülmek, ağlamak, vb.), yararlı bir iş yapmak (çamaşır yıkamak, yemek yapmak, bitki dikmek, vb.), profesyonel düzeyde çalışmak (fabrika ya da bir iş yerinde çalışmak) düşünülebilir. İnsan dışındaki canlılar düşünüldüğünde hareket etme yöntemleri olarak; yürümek, sürünmek, uçmak, yüzmek (su üstünde ya da su altında), toprak

altında tünel açarak ilerlemek ve benzeri hareket biçimleri anlaşılmaktadır. Canlıların davranış biçimleri ise, bireysel ve toplumsal etmenlerin etkileşimiyle gelişir. Bu kavramların robotlar üzerinde uygulanmasıyla, hareket eden, iş yapan, iletişim yetenekleri olan, çeşitli davranış biçimi sergileyebilen tüm makineleri robot olarak tanımlamak mümkündür (Erden, 2003).

İnsan robot etkileşiminde, robot ve insan birbirlerine göre üç değişik konumda olabilirler. Bu konumlar sırası ile yakın, orta ve uzak mesafe olarak sınıflandırılabilir. Kullanıcı ve robot aynı ortamı paylaşıyorlarsa yakın mesafede bulunmaktadırlar. Kullanıcı, robotu direkt olarak görebilmektedir. Kullanıcı, robotu sınırlı olarak görme imkanına sahip ise orta mesafededirler. Buna örnek olarak bir ofis robotu düşünülebilir. Ofis robotu koridorda dolaşırken kullanıcı robotu sınırlı da olsa görebilir. Kullanıcının robotu görme imkanı olmadığı konum ise uzak mesafe olarak adlandırılmaktadır. Orta ve uzak mesafeler için, kullanıcı robotun bulunduğu ortam hakkında bilgi sahibi olmayabilir (Bingül, ve diğ. 2001). Mekanın önceden kullanıcı veya robot tarafından bilinmesi her zaman mümkün olmayacağı açıktır. Bu çalışmada, kullanıcı gezer araç üzerine eklenen kamera sayesinde robotun bulunduğu ortamı tanımıyor dahi olsa onu kolaylıkla kontrol edebilmektedir.

4.2. Tekerlekli Gezer Robot Kinematığı

Muir ve Neuman (1986) tekerlekli gezer robotların kinematik modellemesini geliştirirken bir takım tanımlamalar yapmışlardır. Bunlar;

- Her tekerlekli gezer robot, yerle temas eden birden fazla tekerleğe sahiptir. Bu nedenle kapalı – bağ zincirleri ile modellenebilir.
- Gezer robotun tekerlekleri hem döner hem de bağlantı noktasına göre kendisi ve yer arasında iletimi sağlar.
- Gezer robotların bazı tekerlekleri bazı durumlarda hareket ettirilmeyebilir.
- Bir gezer robotun yörünge kontrolü yapılırken her eklemin konum, hız ve ivme değerlerinin ölçülmesi gerekmez.

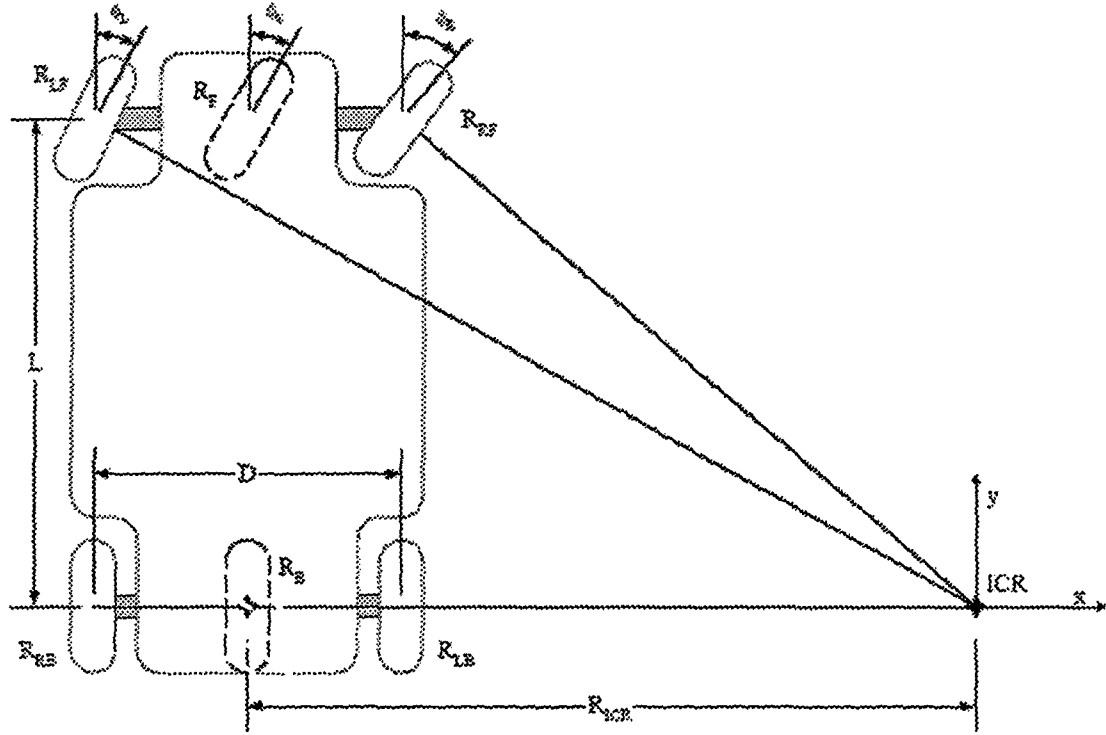
Muir ve Neuman kinematik modelleme için birkaç kabulde de bulunmuşlardır.

- Robot sert, katı, mekanizmalardan inşa edilir.
- Robot her bir tekerleğine bağlı maksimum bir dönen bağlantıya sahiptir.
- Bütün dönen aksların hareketi yüzeye diktir.
- Yüzey pürüzsüzdür.
- Tekerlek ve yüzey arasında kayma olmaz.
- Bağlantı noktalarındaki dönme gerilimi çok küçüktür.

Bu kabul ve tanımlarla Muir ve Neuman üç tekerlekli bir gezer robotun kinematik modellemesini açıklamışlardır.

Bu tezde tasarlanan araç, ön motoru aracın sağ – sol yönünü, arka motoru aracın ileri – geri hareketini kontrol eden, dört tekerlekli bir modeldir. Bu robota uygun kinematik tasarım Ackerman modeli ile açıklanmaktadır.

Ackerman'ın tekerlekli gezer robot (Wheeled Mobile Robots – WMR)" şekli, 4 tekerlekli bir otomobilinkine denk bir modeldir. Ackerman'ın geliştirdiği WMR kinematik hesaplamalardaki özellikler şu şekildedir; robot dört tekerleğe sahiptir, bunlardan öndeki iki tane robota yön vermek için kullanılır; itişe bir etkisi yoktur. Arkadaki iki tekerlek ise dönmez sadece robotun ileri ve geri hareketini sağlar. Formüllerde tekerleklerin yarıçapları R_{LF} (LF – left front: sol ileri) gibi yönü belirten alt indisler kullanılarak, her bir tekerleğin açısal hızı ω_{LF} gibi benzer yöntem ile gösterilmiştir. Öndeki dönmeyi sağlayan iki tekerleğin dönme açıları ise ϕ_L ve ϕ_R ile, aynı aksa bağlı tekerlekler arasındaki mesafe D , ön ve arka akslar arası uzaklık L ile sembolize edilip gezer robot kinematiği modellenmektedir (McNamee, 2003).



Şekil 4.1. Ackerman dönen tekerlekli gezer robot modeli

Ackerman'ın modeli ön tekerleklerin dönmesiyle ilgili bir kısıtlama getirmektedir. Bu kısıtlama, sağ ve sol tekerleklerin belirlenmiş bir referans noktasına göre eşit açılarda dönmesidir. Şekilde bu nokta ICR (instantaneous center of rotation) ile gösterilmiştir. Belirlenen bu anlık referans noktasına göre dönme açıları arasında şu ilişki yazılmıştır;

$$\pm \cot \phi_R \pm \cot \phi_L = \frac{D}{L} \quad (4-1)$$

Denklem 4 – 1, Ackerman kısıtlaması olarak adlandırılmaktadır. Denklemdeki eksi işareti sağa veya sola dönüşte iç kısımda kalan tekerlek için uygulanır. Ackerman modelinde hesaplamaları kolaylaştırmak için dönüşlerde tek ve sanal bir referans noktasına göre yarıçap ve dönme açısı tanımlanmaktadır. Bunlar sırasıyla R_F ve ϕ_S olarak gösterilir. Bu sanal tekerleğin dönme açısı, sağ ve sol tekerleklerin dönme açıları cinsinden;

$$\begin{aligned}\cot\phi_S &= \cot\phi_R \pm \frac{D}{2L} \\ &= \cot\phi_L \pm \frac{D}{2L}\end{aligned}\tag{4-2}$$

olarak hesaplanır.

Sonraki aşamada kinematik modeli oluşturabilmek için Ackerman WMR modelinde tanımlanan ICR'ye ait değişkenler tanımlanır. Şekil 4.2'de gösterildiği gibi sağa doğru çok küçük bir dönüş için, uygulanan ϕ_S , ω ve değişkenlerine göre robotun durum vektörü

$$\mathbf{x} = [x \quad y \quad \theta]^T$$

şeklinde tanımlanmaktadır.

Zamandaki dt gibi küçük bir değişiklik, durum vektörü üzerinde diferansiyel farka neden olur.

$$d\mathbf{x} = [dx \quad dy \quad d\theta]^T$$

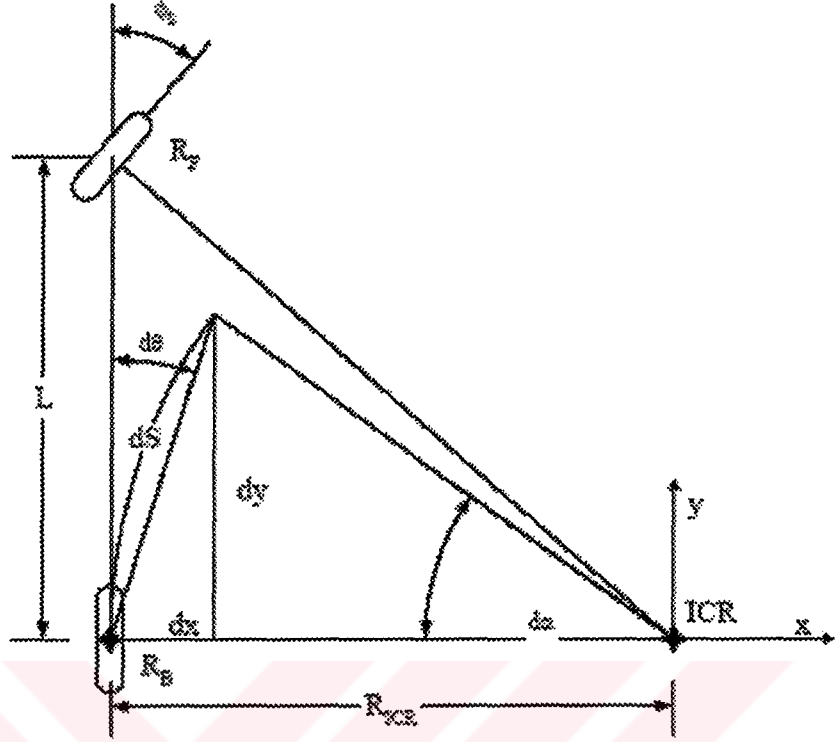
Arka tekerleklerle seçilen anlık referans noktası arasındaki uzaklık R_{ICR} ;

$$R_{ICR} = L \cot\phi_S\tag{4-3}$$

ve dönen tekerleklerle ICR arasındaki uzaklık R_S ;

$$R_S = \frac{L}{\sin\phi_S}\tag{4-4}$$

formülleri ile hesaplanmaktadır.



Şekil 4.2. Ackerman dönme geometrisi

Her bir tekerlek dönüşü sırasında bağlantı noktasından başlayan dS yayını takip eder. Bu dönüşünde $d\alpha$ açısını oluşturur.

$$dS = R_{ICR} d\theta = R_B \omega dt \quad (4 - 5)$$

Bu denklemdaki R_B sanal arka tekerleğin açısıdır. Böylece aracın yönelmesinde oluşan diferansiyel fark;

$$d\theta = \frac{dS}{dt} \tan \phi_S \quad (4 - 6)$$

ile elde edilmektedir

Hareket sonucunda oluşan yer deęişme;

$$\begin{aligned} dx &= dS \cos\phi_S \cos\theta \\ dy &= dS \cos\phi_S \sin\theta \end{aligned} \quad (4-7)$$

denklemlerini çözerek bulunmaktadır.

Anlık tekerlek hızı ise;

$$\frac{dS}{dt} = \omega R \quad (4-8)$$

dir.

Durumdaki diferansiyel fark, tekerleklerin açısal hızı cinsinden şöyle bulunur;

$$\begin{aligned} dx &= dt \omega R_B \cos\phi_S \cos\theta \\ dy &= dt \omega R_B \cos\phi_S \sin\theta \\ d\theta &= dt \frac{\omega R_B}{L} \tan\phi_S \end{aligned} \quad (4-9)$$

Son olarak, denklem 4 – 9 kullanarak Ackerman WMR kinematik modelinin durum uzay şekli denklem 4 – 10’ daki gibi elde edilir.

$$\dot{\mathbf{x}} = \frac{d}{dt} [x \quad y \quad \theta]^T$$

$$= \begin{bmatrix} R_B \cos\phi_S \cos\theta & 0 \\ R_B \cos\phi_S \sin\theta & 0 \\ R_B \tan\phi_S / R_S & 0 \end{bmatrix} \begin{bmatrix} \omega \\ \phi \end{bmatrix} \quad (4-10)$$

Elde edilen bu denklemler ve modellenen aracın bilgileri ile kinematik hesaplama yapılmıştır. Buna göre;

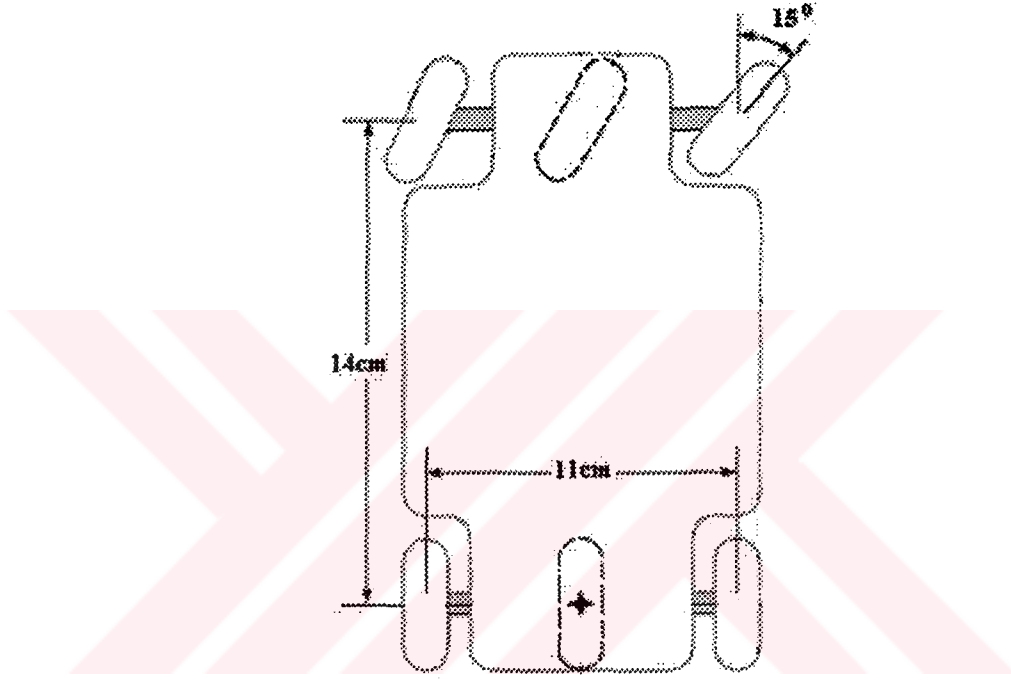
Aracın doğrusal hızı, $v = 3 \text{ km/sa} = 83.33 \text{ cm/sn}$

İki tekerlek arasındaki uzaklık, $D = 11 \text{ cm}$

Aracın ön ve arka akslar arasındaki mesafe, $L = 14 \text{ cm}$

$t = 1 \text{ sn}$

$\phi = 15^\circ$



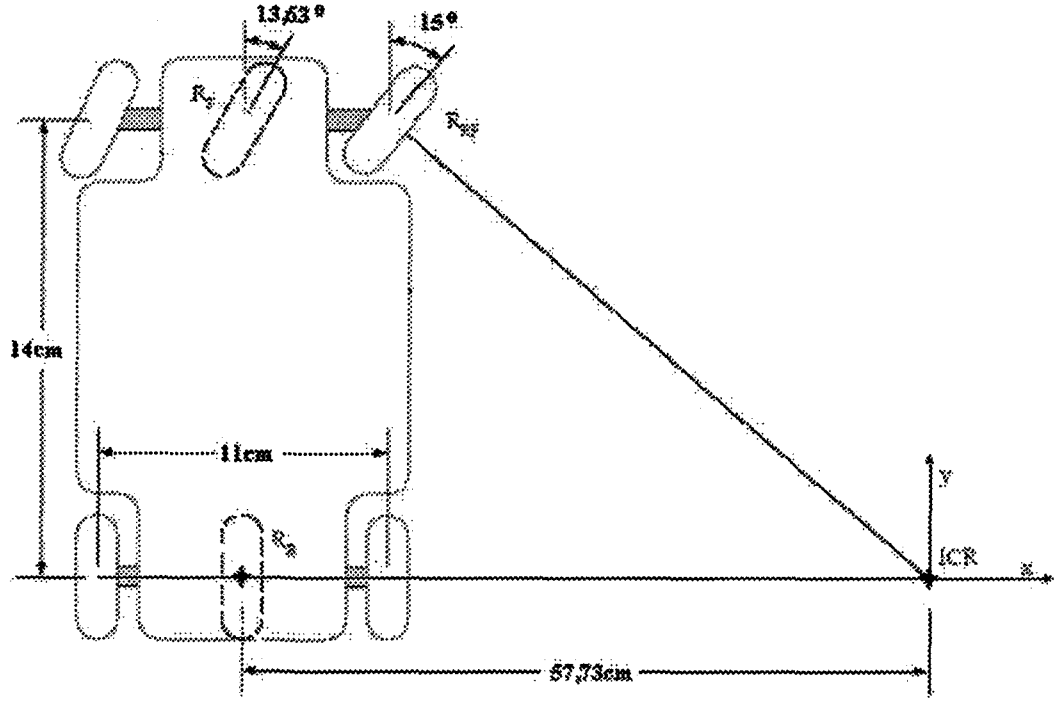
Şekil 4.3. Aracın gerçek ölçüleri

$$\cot \theta_s = \cot 15^\circ + (D / 2L)$$

$$\theta_s = 13.63^\circ$$

$$R_{ICR} = L \cdot \cot \theta_s = 57.73 \text{ cm}$$

olarak hesaplanır. Denklem 4 – 4 kullanılarak $R_s = 58.33 \text{ cm}$ bulunur.



Şekil 4.4. Hesaplamalar sonucu elde edilen değerler

Açısal hız $\omega = v / R_S = 1.40$ rad

$t = 1$ sn seçildiğinden $\theta = 79.89^\circ$ olarak elde edilmiştir.

Denklem 4 – 5 kullanılarak hesaplanan $R_B = 58.73$ cm.'dir.

Son adım olarak Denklem 4 – 10' deki x ve y değerleri sırasıyla 9.95cm. ve 55.63cm. olarak hesaplanmıştır.

Yapılan bu modelleme, Matlab programı kullanılarak simule edilmiş ve elde edilen sonuçlar aşağıda verilmiştir:

```
>> v = 83.33;
>> D = 11;      // İki tekerlek arasındaki mesafe
>> L = 14;     // Aracın uzunluğu
>> t = 0:0.1:1; // Örnekleme aralığı
```



```
>> fi = 0.26;           // Sapma açısının radyan karşılığı
```

```
>> fiS = Acot (cot(fi) + (D / (2*L)))
```

```
fiS =
```

```
0.2363
```

```
>> RICR = L * cot(fiS)
```

```
RICR =           // Anlık dönme merkezi
```

```
58.1273
```

```
>> RS = L / sin(fiS)
```

```
RS =           // Sanal merkez
```

```
59.7895
```

```
>> w = v / RS     // Açısal hızın bulunması
```

```
w =
```

```
1.3937
```

```
>> delta = w * t
```

```
delta =
```

```
Columns 1 through 6
```

```
0  0.1394  0.2787  0.4181  0.5575  0.6969
```

Columns 7 through 11

0.8362 0.9756 1.1150 1.2544 1.3937 // Sapma açısının radyan değerleri

>> Ddelta = (180 * delta) / 3.14

Ddelta =

Columns 1 through 6

0 7.9895 15.9790 23.9685 31.9580 39.9475

Columns 7 through 11

47.9370 55.9265 63.9160 71.9054 79.8949 //Sapma açısındaki
değişikliğinin derece karşılığı

>> RB = (RICL * delta) / (w * t)

RB =

58.1273

>> yolx = RB*cos(delta)*cos(fiS)

yolx =

Columns 1 through 8

56.5113 55.9634 54.3301 51.6432 47.9547 43.3363 37.8774 31.6840

Columns 9 through 11

```
24.8761 17.5858 9.9544 // Her örnekleme sonunda konumun x eksenini
değerleri
```

```
>> yoly = RB*sin(delta)*cos(fiS)
```

```
yoly =
```

```
Columns 1 through 8
```

```
0 7.8506 15.5490 22.9459 29.8977 36.2698 41.9384 46.7938
```

```
Columns 9 through 11
```

```
50.7416 53.7054 55.6277 // Her örnekleme sonunda konumun y eksenini
değerleri
```

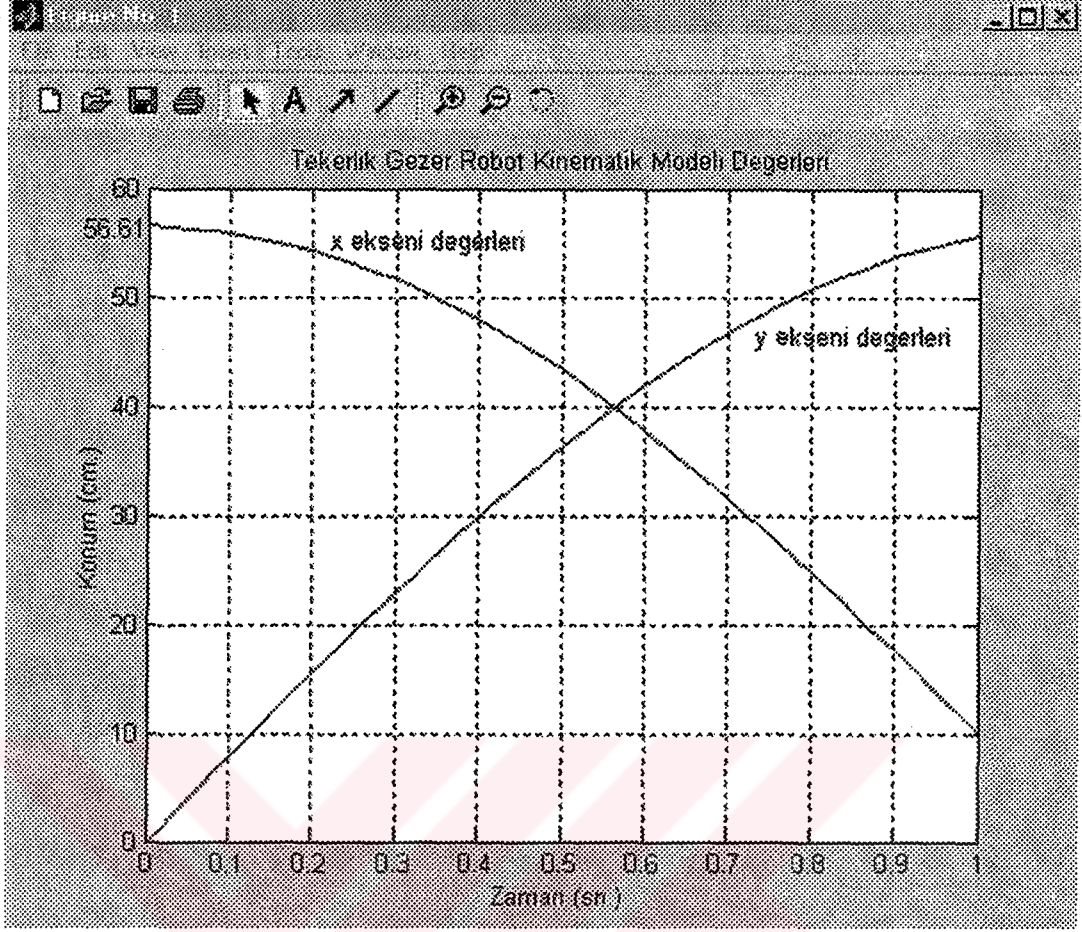
```
>> plot(t,yolx,'r-')
```

```
>> hold
```

```
Current plot held
```

```
>> plot(t,yoly,'b-')
```

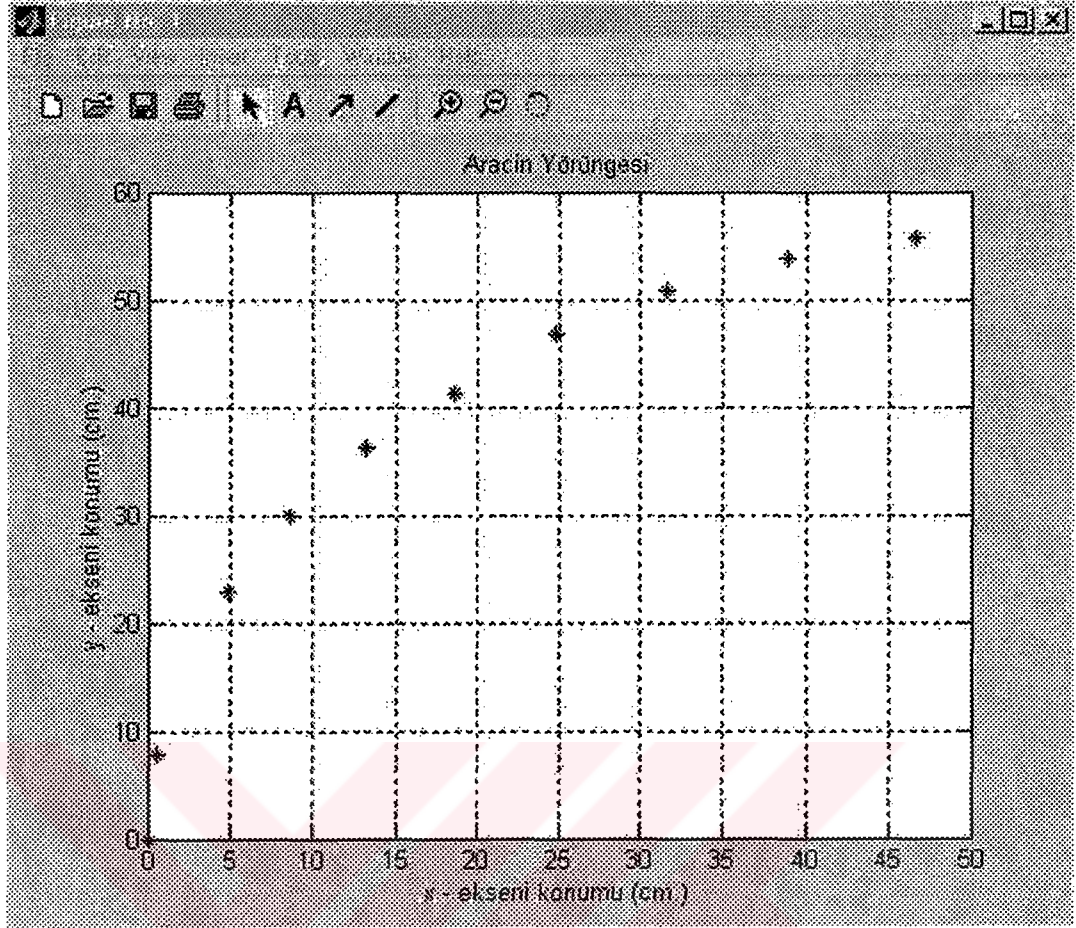
```
>> grid
```



Şekil 4.5(a). Aracın kinematik modelinin Matlab simülasyonu

Hesaplanan “yolx” ve “yoly” değişkenleri aracın örneklenen zaman aralıklarında aldığı konum bilgilerini ifade etmektedir. Seçilen referans noktasına göre y eksenini başlangıçta 0 iken hareket tamamlandığında 55.62cm. değerine ulaşmaktadır. x eksenini ise harekete başlanıldığı anda 56.61cm. iken 1 sn. sonunda 9.95cm. değerini almaktadır. Hesaplanan bu değişkenler Şekil 4.5(a)’da grafiksel olarak gösterilmiştir.

Şekil 4.5(b)’de seçilen zaman aralığında hesaplanan x ve y konum değerlerinin bileşkelerinden elde edilen aracın yörüngesi gösterilmiştir.



Şekil 4.5(b). Aracın yörüngesi

4.3. Uygulamanın Adımları ve Gelişim Süreci

Daha önceki bölümde anlatılan teknolojiler sayesinde artık bilgi rahatça ulaşılabilirlik özelliği kazanmıştır. İstenilen veya merak edilen herhangi bir konuda araştırma yapmak, ulusal veya uluslararası gelişmeleri takip oldukça kolaylaşmıştır. Bu tezin amacı, var olan bilgileri kullanarak somut bir cihaz ortaya çıkartmaktır; uzaktan kontrol edilebilen ve kendi üzerine monte edilmiş farklı algılayıcılardan elde ettiği bilgiyi gönderen ufak bir model araç oluşturmaktır. Bu tezin diğer bir özelliği ise kullanışlı ve geliştirilmeye açık olmasıdır. Kullanışlı bir tezdir, çünkü insan hayatına doğrudan girebilecek ve bir çok kullanıcı tarafında kolayca kullanılacak bir modeldir. Geliştirilmeye açıktır, çünkü üzerine yapay zeka ve bulanık mantık içeren özellikler eklenerek bir çok alanda, eğitim amaçlı olabileceği gibi savunma teknolojilerinde, araştırma geliştirme projelerinde de kullanılabilir (Luo, 2003; Mann, 2004).

Model robotun geliştirilmesi sürecinde kullanılan cihazlar ve bunların birbirleriyle etkileşimi şu şekildedir;

4.3.1. Uygulamanın bileşenleri

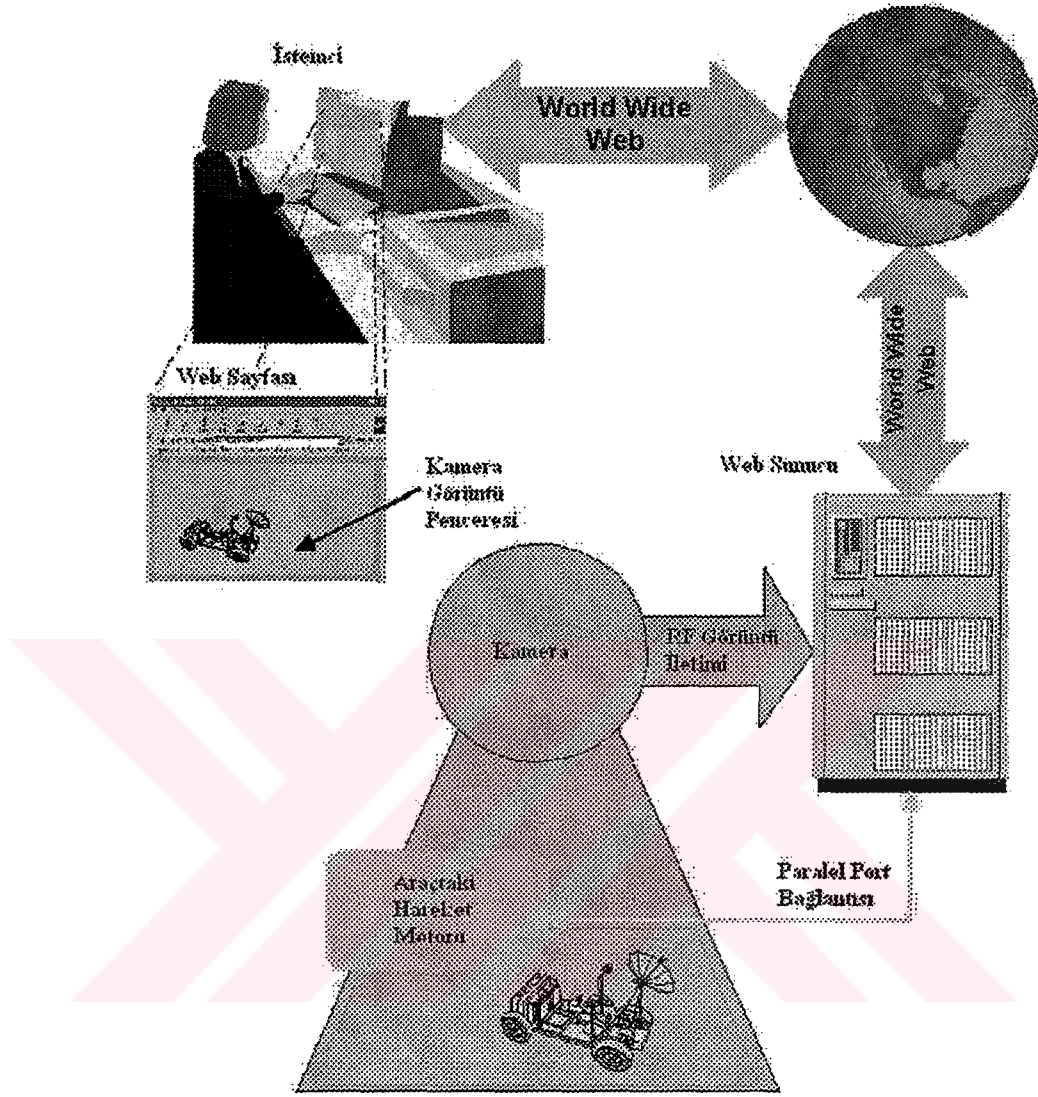
Projenin mekatronik olma özelliğinden önceki bölümlerde bahsedilmiştir. Bir mekatronik sistemi oluşturan disiplinler şunlardır;

1. Mekanik Sistemler (bilgisayarlı tasarım, bilgisayarlı üretim, boşluksuz ve hassas çalışan mekanik iletim sistemleri)
2. Kontrol Sistemleri (kontrol sistem tasarımı, gerçek zamanlı kontrol)
3. Bilgisayar Sistemleri (algoritma uygulaması - kodlama, iletişim protokolü oluşturma ve gerçek zamanlı programlama)
4. Elektrik-Elektronik Sistemler (eyleciler (motorlar vb) ve algılayıcılar(sensörler), analog/dijital, dijital/analog kartlar, bilgi toplama ve derleme kartları) (Gürbüz, 2004).

Bu prensiplerden yola çıkarak tasarlanan projenin bileşenleri;

- Uzaktan kumandalı araba
- Bilgisayarın paralel portu üzerinden kontrolü sağlayan ve arabanın kumanda devresine eklenmiş elektronik bir kontrol devresi
- Telsiz kamera ve alıcısı
- Dazzle (Görüntü Yakalama(Capture) Cihazı)
- İnternet üzerinden görüntü almayı sağlayan yardımcı program (UltraVNC)

Bu prensiplerden yola çıkılarak geliştirilen sistemin genel yapısı aşağıdaki şekilde gösterilmiştir.



Şekil 4.6. Uygulamanın genel yapısı

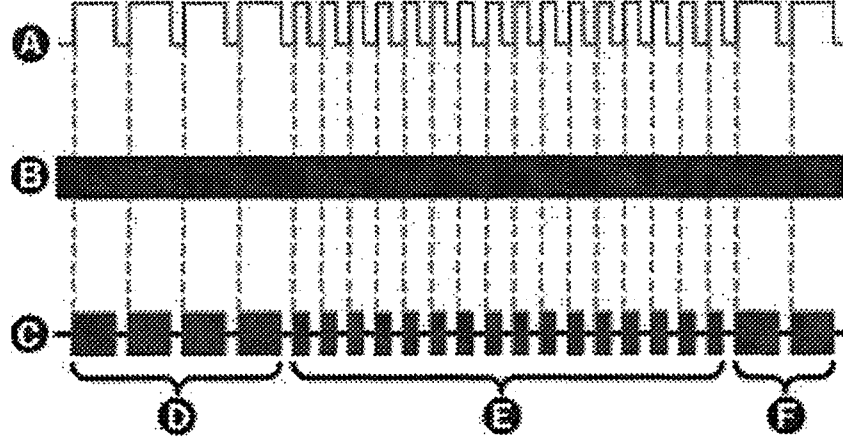
4.3.2. Uygulamanın gelişim adımları

Uygulamanın gelişimi adım adım şöyledir;

- 1. Adım: Kumanda içine kontrol devresinin yerleştirilmesi

Öncelikle 27 MHz. frekansla kontrol edilebilen bir arabanın çalışma prensibini açıklamak gerekirse; kumanda devresinden gönderilen bilgi, taşıyıcı sinyalle birlikte

giden darbe veya pals serisidir. Her seri kendi içinde eşzamanlılık (senkronizasyon) palslerini de taşır. Bu eşzamanlılığı sağlayan sinyal 700µs. aralıklarla gönderilen 2.1 ms.'lik dört palsdir (Tyson, 2004).



Şekil 4.7 RF ile gönderilen sinyal

A : Pals serisi

B : 27 MHz. taşıyıcı sinyal

C : Gönderilen bilgi sinyali

D : 4 tane 700µs. aralıklı 2.1 ms. uzunluğundaki eşzamanlılık sinyali

E : Her biri 700µs. uzunluğunda ve 700µs. aralıkla gönderilen bilgi palsleri

Hareketi gerçekleştirecek pals parçaları:

İleri – 16 pals

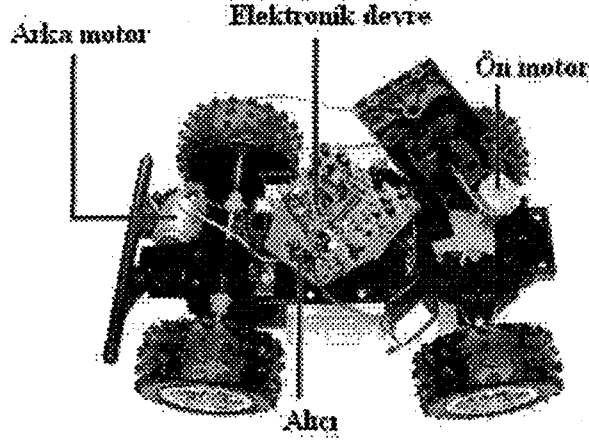
Geri – 40 pals

İleri/Sol – 28 pals

İleri/Sağ – 34 pals

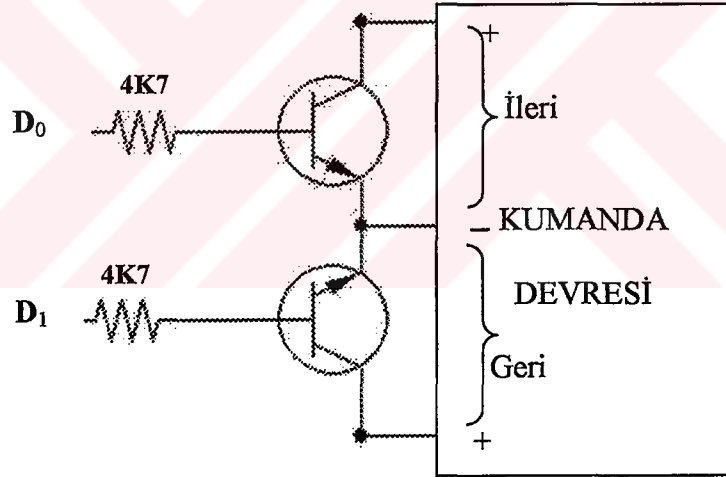
Geri/Sol – 52 pals

Geri/Sağ – 46 pals



Şekil 4.8. Arabanın genel yapısı

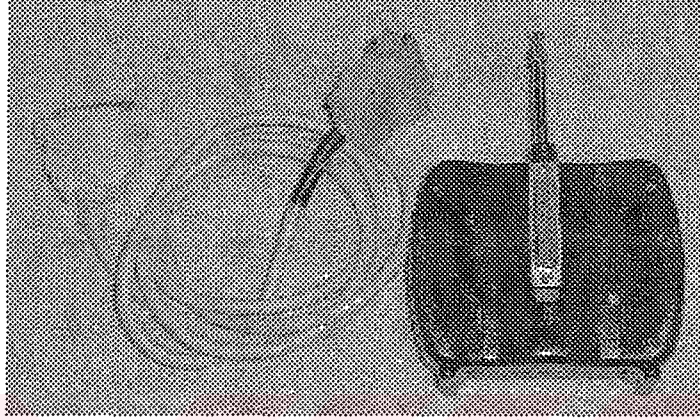
Kumanda ünitesinin içine entegre edilen ve paralel porttan gelen veri ile kontrolü sağlayan bir elektronik devre tasarlanmıştır. Devre blok şeması Şekil 4.9'da gösterilmiştir.



Şekil 4.9 Transistörlü devre blok şeması

Bu devreyi açıklamak gerekirse, transistör, kontrollü olarak akımı tek yönde ileten devre elemanıdır. Emiter, Base ve Kollektör uçlarına uygulanan gerilim değerlerine bağlı olarak doğru ve ters poları kazanırlar. Base akımı sıfır iken transistör kesimdedir. Base akımı artırıldığında, transistör iletme geçer. Base girişindeki akım nedeniyle oluşan gerilim, Emiter ve Kollektör arasındaki eşik gerilimin değerinden büyükse bu iki uç arasında akımı geçirerek bir anahtar görevi yapmaktadır. Bu özelliği kullanılarak kumanda üzerinde mekanik olarak yapılan ileri – geri , sağ – sol

anahtarlama elektriksel hale getirilmiştir. Şekil 4.9’da görülen devre ileri, geri, sağ ve sol olmak üzere her dört yön içinde tekrarlıdır. İleri ve geri hareketler tek başına işleyebilirken, sağ ve sol yönler ancak ileri veya geriyle beraber çalışabilir. Benzer bir yapı kameranın hareketini kontrol edecek devre için de kullanılmıştır. Elde edilen yeni kumanda Şekil 4.10’da gösterilmiştir.



Şekil 4.10. Yeni kumanda

- 2. Adım: Kumanda devresinin paralel porttan gelen bilgi ile kontrolü

Yukarıda anlatılan devreyi sürmek için paralel porttan gelen sinyalden yararlanılmıştır. Bu sinyaller porta Borland Delphi’de geliştirilen bir program vasıtasıyla gönderilmiştir. Paralel portun veri pinlerinden 4 tanesi, D₀, D₁, D₂, D₃, arabanın kontrolünü, aynı zamanda D₂ ve D₃, kameranın kontrolünü sağlamak amaçlı tasarlanmıştır. Veri bitlerinin durumları ve gerçekleşen işlemler Tablo 4.1’de gösterilmiştir.

Tablo 4.1. Aracın hareketine ilişkin bit gösterimi

Hareket	Bitlerin durumu	Onlu karşılığı
Araba ileri	0001	1
Araba geri	0010	2
Kamera sol	0100	4
Kamera sağ	1000	8
Araba İleri-sol	0101	5
Araba İleri-sağ	1001	9
Araba geri-sol	0110	6
Araba geri-sağ	1010	10

Tasarlanan bu değerler Parmon (Parallel Port Monitor – Paralel Port Görüntüleyicisi) adlı paralel port değerlerinin izleyebileceği bir programla test edilmiştir. Klavye veya fare kullanılarak gönderilen bilgiler paralel portun veri pinlerinin istenilen değerlere ulaşmasını sağlamıştır. Prosedürler ve test sonuçları aşağıda gösterilmiştir.

Hazırlanan bu devreyi paralel port dan veri göndererek çalıştırılabilmek için Delphi'de geliştirilmiş kodun bir modülü şu şekildedir;

```
procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
case key of
VK_INSERT: begin
PortOut(888,0);
PortOut(888,5);
end;
VK_DELETE: begin
PortOut(888,0);
PortOut(888,6);
end;
VK_HOME: begin
PortOut(888,0);
PortOut(888,1);
end;
VK_END : begin
PortOut(888,0);
PortOut(888,2);
end;
VK_PRIOR: begin
PortOut(888,0);
PortOut(888,9);
```

```
end;  
VK_NEXT: begin  
PortOut(888,0);  
PortOut(888,10);  
end;  
else  
PortOut(888,0);  
end;  
  
end;
```

Veri göndermek için : PORTOUT (port adresi, bayt)

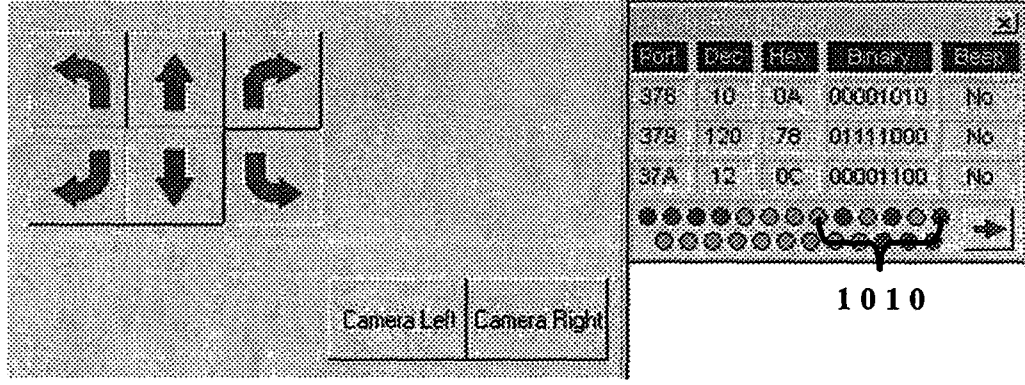
Paralel Port adresi $(0378)_{16}$ iken, kod içinde bu değer $(888)_{10}$ olarak belirtilir.

$(0378)_{16} = (888)_{10}$

PORTOUT komutunun ikinci parametresi ise porta gelen bilginin desimal karşılığıdır. Her bir yön için elde edilen sayılar Tablo 4.1' de gösterilmiştir. Örneğin, ileri için $(0000\ 0001)_2 = (1)_{10}$ veya ileri – sol için $(0000\ 0101)_2 = (5)_{10}$

Aşağıda paralel porta veri göndermek için Delphi' de geliştirilen kodun prosedürleri gösterilmiştir.

```
procedure TMainForm.sağ_geri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
PortOut(888,10);  
end;
```

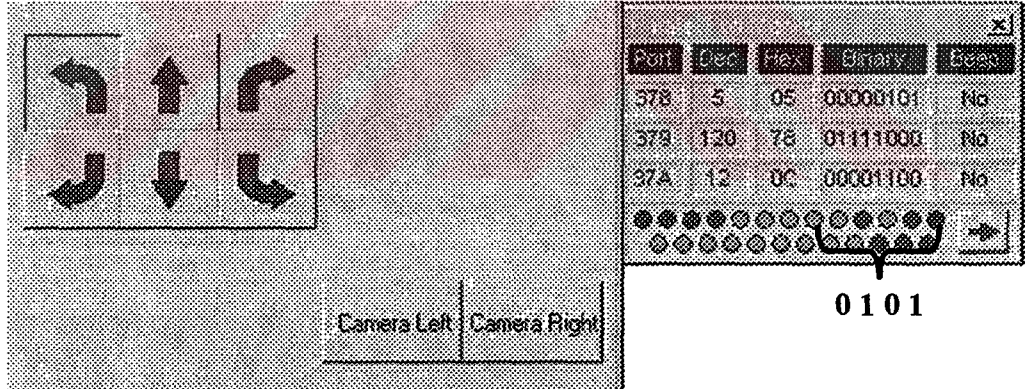


Şekil 4.11. Aracın geri – sağ hareketi ve porttan okunan değer

```

procedure TMainForm.sol_ileri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);
PortOut(888,5);
end;

```



Şekil 4.12. Aracın ileri – sol hareketi ve porttan okunan değer

Porta gönderilen $(10)_{10} = (1010)_2$ bilgisi arabanın geri – sağ, $(5)_{10} = (0101)_2$ bilgisi ise aracın ileri – sol hareketini gerçekleştirmektedir.

```

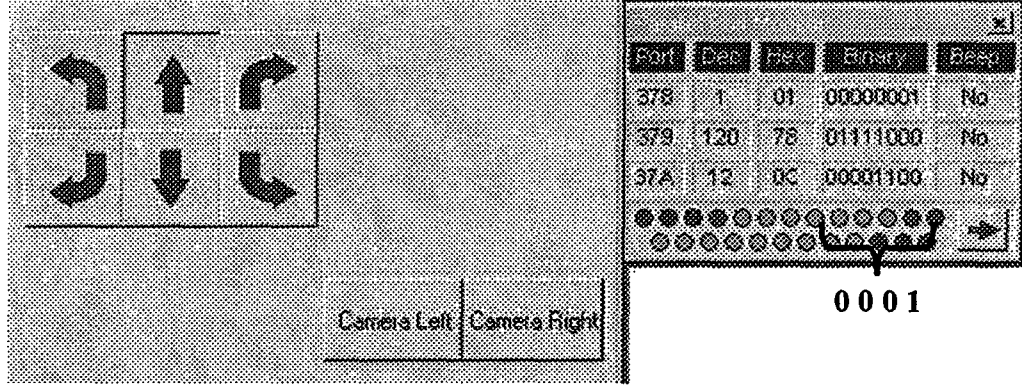
procedure TMainForm.ileri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

```

```

begin
PortOut(888,0);
PortOut(888,1);
end;

```

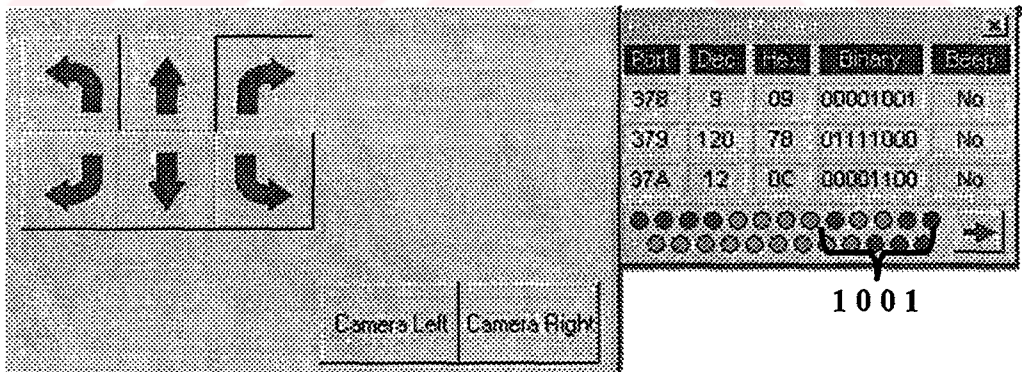


Şekil 4.13. Aracın ileri hareketi ve porttan okunan değer

```

procedure TMainForm.sag_ileri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);
PortOut(888,9);
end;

```



Şekil 4.14. Aracın ileri - sağ hareketi ve porttan okunan değer

```

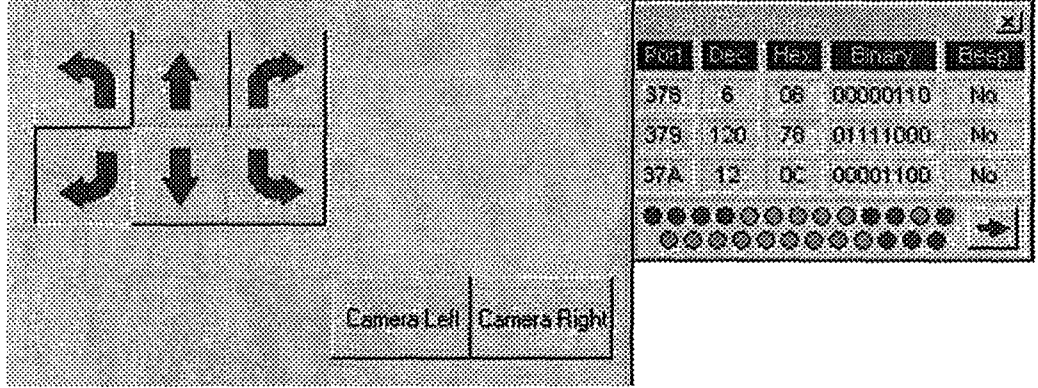
procedure TMainForm.sol_geri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);

```

```

PortOut(888,6);
end;

```

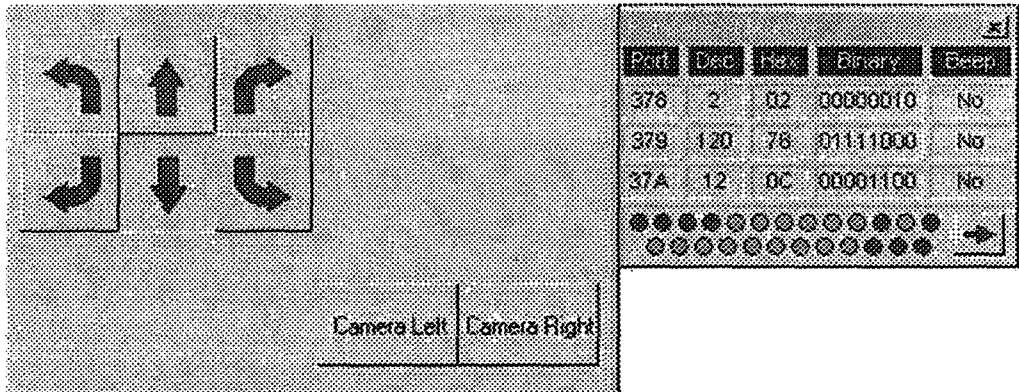


Şekil 4.15. Aracın geri - sol hareketi ve porttan okunan değer

```

procedure TMainForm.geri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);
PortOut(888,2);
end;

```



Şekil 4.16. Aracın geri hareketi ve porttan okunan değer

Arabanın her hareketinden önce bir önceki durumu ifade eden bitler "0" lanarak yeni değer gönderilir.

```
procedure TMainForm.FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
case key of
VK_UP: begin
PortOut(888,0);
PortOut(888,1);
end;
VK_RIGHT: PortOut(888,4);
VK_LEFT: PortOut(888,8);
VK_DOWN: begin
PortOut(888,0);
PortOut(888,2);
end;
VK_INSERT: begin
PortOut(888,0);
PortOut(888,9);
end;
VK_DELETE: begin
PortOut(888,0);
PortOut(888,10);
end;
VK_HOME: begin
PortOut(888,0);
PortOut(888,1);
end;
VK_END : begin
PortOut(888,0);
PortOut(888,2);
end;
VK_PRIOR: begin
PortOut(888,0);
PortOut(888,5);
```



```

end;
VK_NEXT: begin
PortOut(888,0);
PortOut(888,6);
end;
else
PortOut(888,0);
end;

```

Yukarıda belirtilen kodda model arabanın ileri, geri, sağ ve sol kontrolleri gerçekleştirilmiştir. Devamında ise kameranın kontrolüne ilişkin bit değerleri verilmiştir.

```

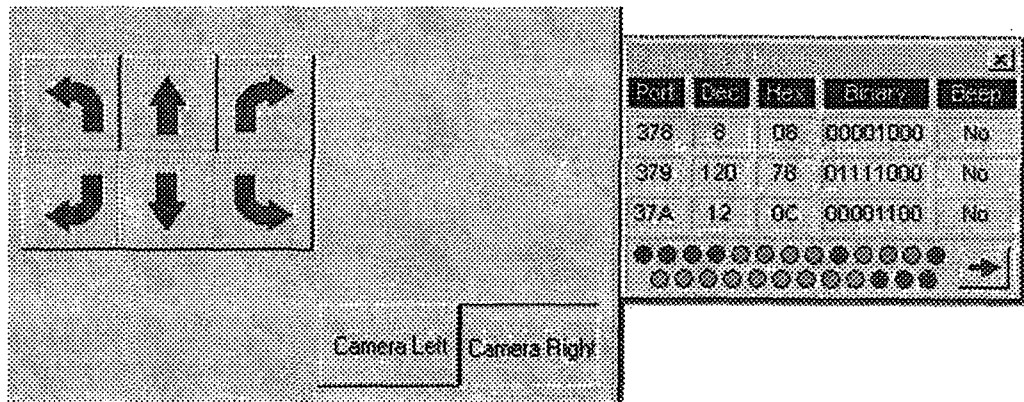
procedure TMainForm.Camera_Right1MouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);
end;

```

```

procedure TMainForm.Camera_Right1MouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,8);
end;

```

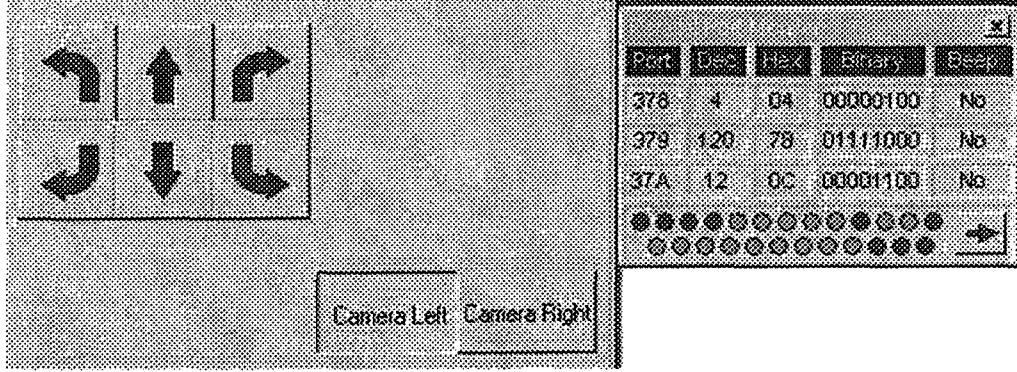


Şekil 4.17. Kameranın sağ hareketi ve porttan okunan değer

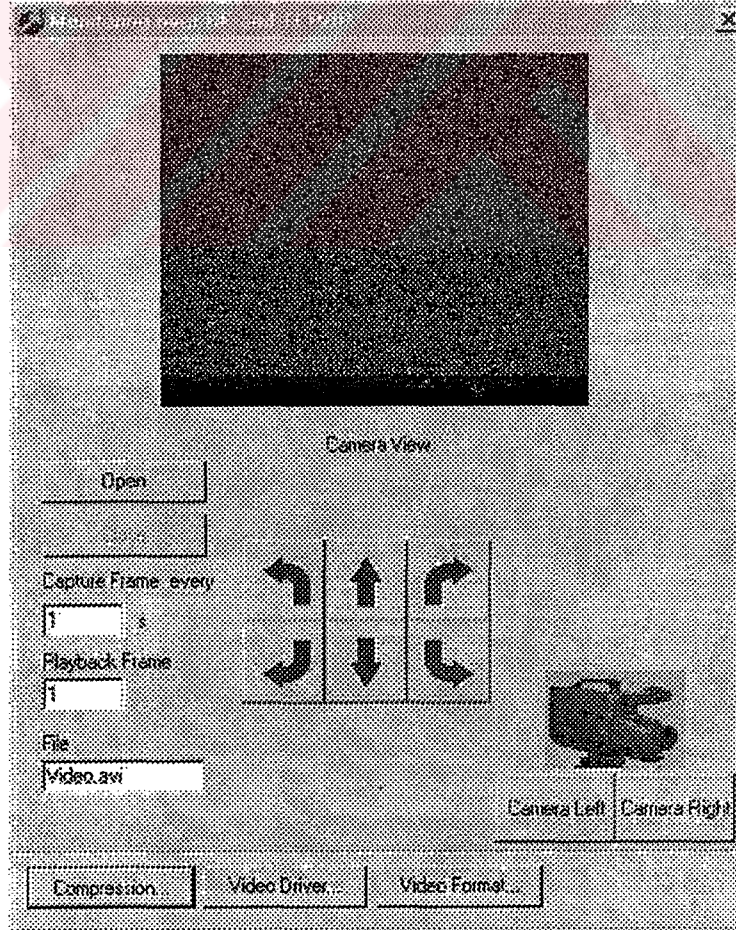
```

procedure TMainForm.Camera_Left1MouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,4);
end;

```



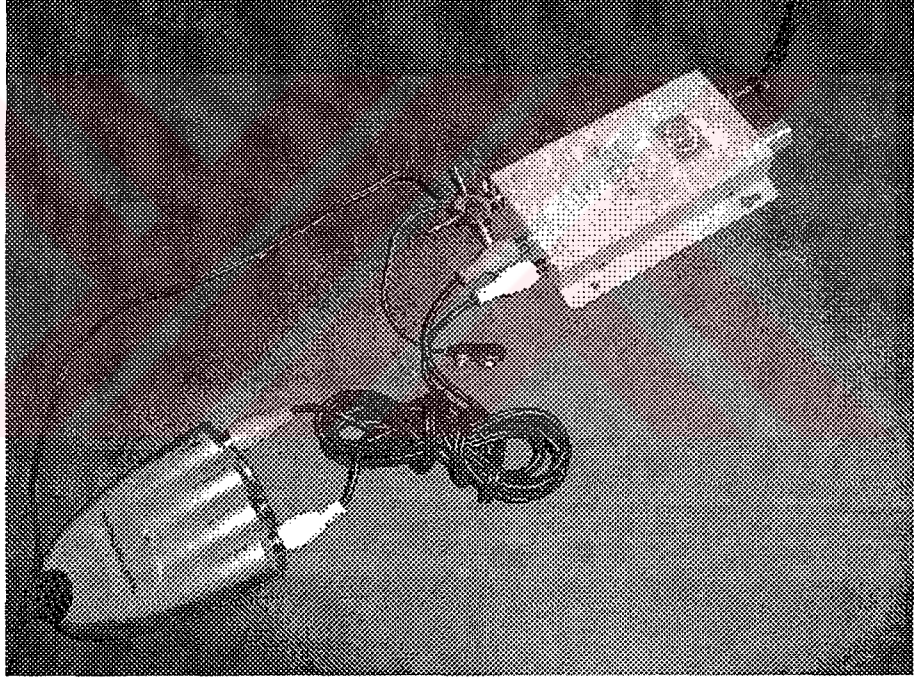
Şekil 4.18. Kameranın sol hareketi ve porttan okunan değer



Şekil 4.19. Geliştirilen programın arayüzü

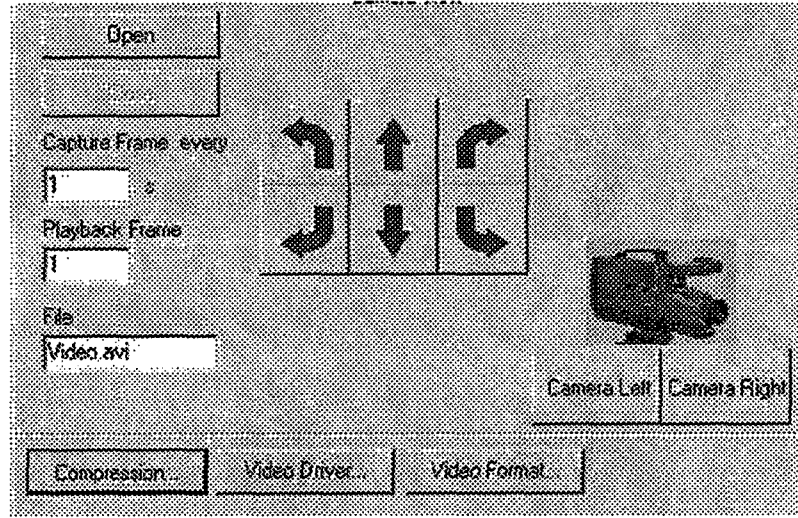
Kodun tamamı Ek – A’da verilmiştir.

Bu kod aracılığıyla paralel porta gönderilen veriler kumanda sayesinde, radyo frekansı(RF) ile, arabaya iletmektedir. Verici, gelen sinyale frekans modülasyonu uygulamakta böylece, modülasyonlu mesajı elektromanyetik dalga yayınımla göndermektedir. Mesaj, yine elektromanyetik dalga yayınımlı yoluyla alınarak sinyalin demodülasyonu yapılmaktadır. Alıcıya gelen mesaj bir uyarı sinyali ile sistemi harekete geçirmektedir.



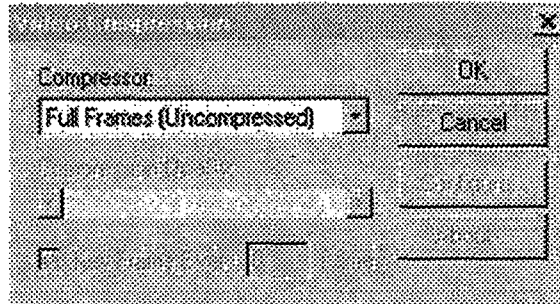
Şekil 4.20. Kameranın alıcısı ve Dazzle bağlantısı

Dazzle, kamerayla kaydedilen video filmlerin, istenilen ortamda paylaşımına olanak veren yeni nesil USB video bağlantı ünitesidir. Üzerinde audio/video girişleri bulunan Dazzle ile, analog kamera veya TV’den yakalanan görüntüler, Dazzle MGI VideoWave 4 yazılımı sayesinde yeniden yaratılabilmektedir



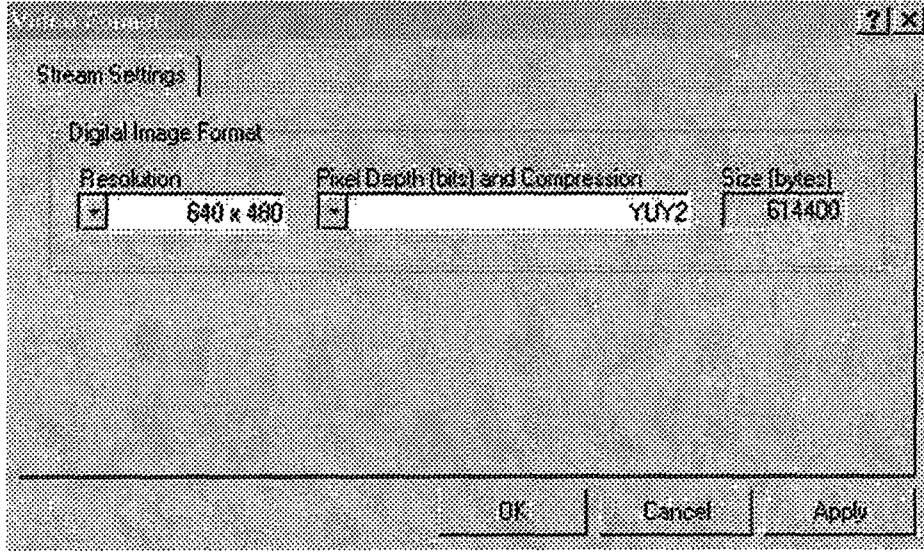
Şekil 4.21. Kameralarla ilgili ayarlar

Kullanıcı eğer isterse görüntüyü kaydetme hakkına sahiptir. Şekil 4.21’de gösterilen “File” alanına dosyanın ismini yazarak “Open” seçeneğini aktif ettiğinde kameradan elde edilen görüntü kaydedilebilmektedir. “Capture Frame” kayıt esnasında kaç saniyede bir kare görüntünün kayıt edileceğini belirler. Girilen değer tamsayı olmalıdır ve gerçek görüntüye en yakın kayıt bu değer’in “1” olduğu anda elde edilmektedir. “Playback Frame” ise görüntüleme sırasında kaç saniyede bir kare gösterileceğini belirler. Bu alana yazılacak değer de tamsayı olmalıdır. Her iki seçenekte görüntünün daha yavaş kaydedilmesini, böylece söz konusu dosyanın daha küçük olmasını sağlamaktadır.



Şekil 4.22. Görüntü biçiminin seçilmesi

“Compression” seçeneği ile kaydedilecek görüntüde bir sıkıştırma yapıp yapılmayacağı belirlenir. Uygun sıkıştırma ile de dosyanın boyutu ve biçimi ayarlanabilir.



Şekil 4.23. Görüntünün çözünürlüğünün ayarlanması

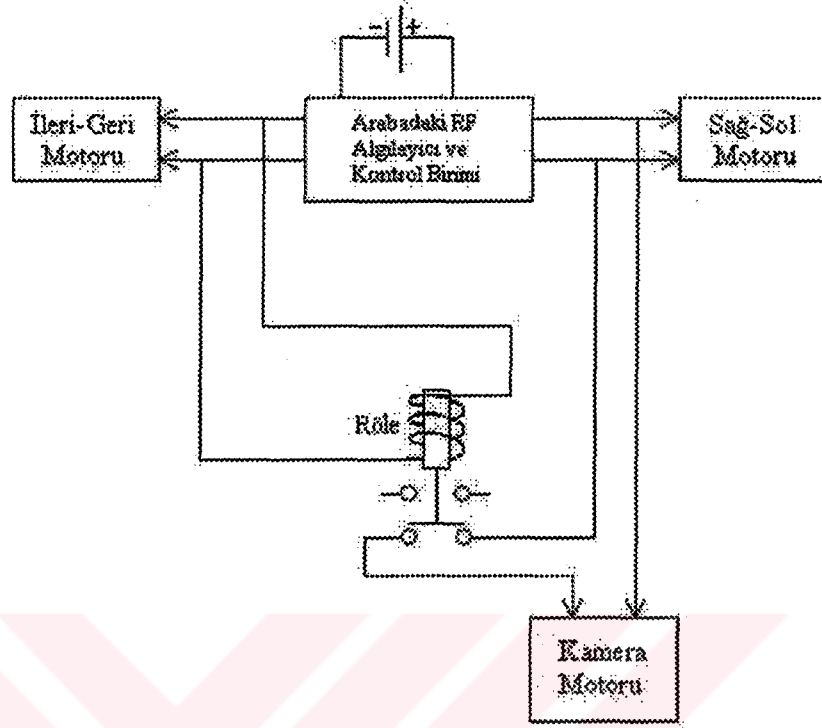
“Video Format” özelliği ise görüntünün çözünürlüğüyle ilgili ayarların yapılmasını sağlar. Minimum çözünürlük 160 x 120 iken maksimum 720 x 576 ya destek vermektedir. Yapılan testlerden en uygun kalite 640 x 480 ile elde edilmiştir.

- 3. Adım: Kameranın hareketi

Model arabaya eklenmiş telsiz kamera, bir motora bağlanan mil kullanılarak hareketli hale getirilmiştir. Bu motorun kontrolü de paralel porttan gelen bilgiyle sağlanmıştır. Aşağıdaki şekilde bu devreye ait blok gösterim yer almaktadır.

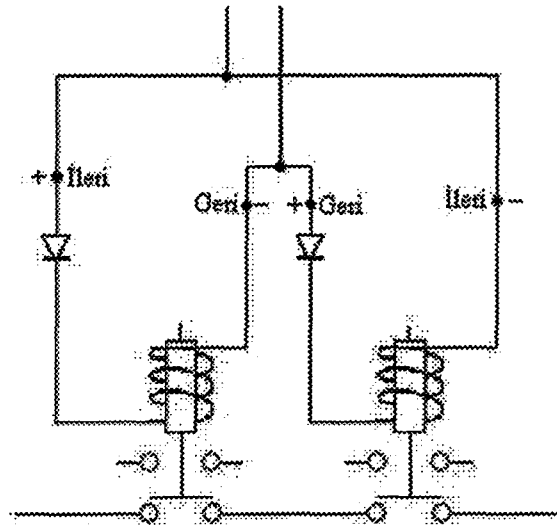
Şekil 4.24’de gösterilen devre, araba hareket halindeyken kameranın hareket etmemesi durumu düşünülerek tasarlanmıştır. Çünkü kullanıcı yön tayin etmekte zorlanır. Arabanın ileri – sağ, ileri – sol, geri sağ ve geri – sol yönlerinde hareket edebilmesi için paralel porttan gelen bilgide her iki yönü kontrol eden bitlerin lojik 1 olması gerekmektedir. İleri – geri motorunu tetikleyen bir sinyal geldiğinde Şekil 4.24’de gösterilen röle enerjilenerek anahtarı “kapalı” durumdan “açık” duruma getirmektedir. Böylece kameranın hareketini sağlayan motora giden devreyi keser. Bu tasarım sayesinde porttan gelen bilgi bitlerinden ileri veya geri sinyali geliyorsa

sadece araba hareket etmiş olur. Kameranın hareketi ise sadece D_2 ve D_3 veri pinlerinden lojik 1 geldiği zaman gerçekleşmektedir.



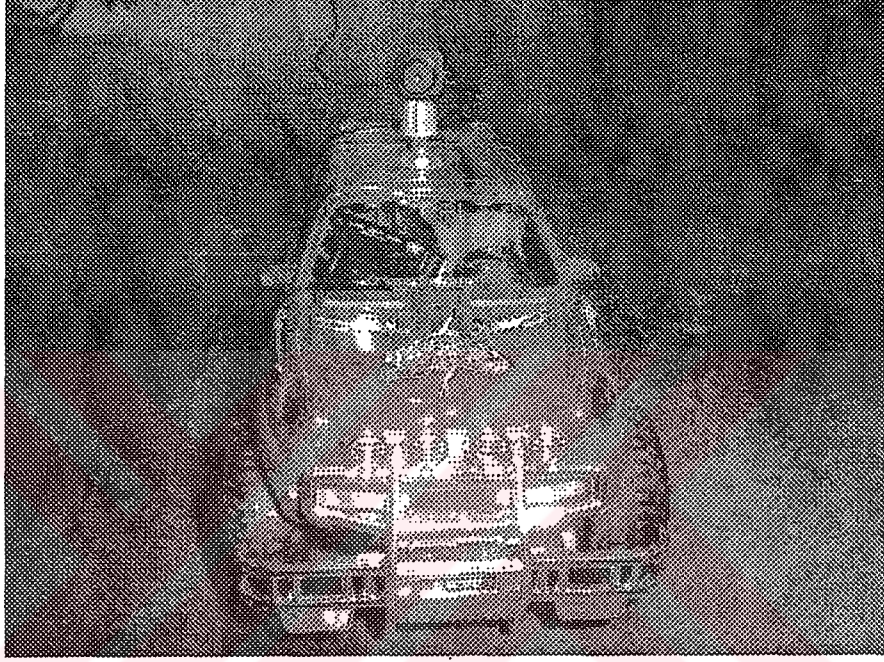
Şekil 4.24. Kameranın kontrolü

Projede kullanılan yönsüz röle, 2 ayrı yönlü röle kullanarak tasarlanmıştır. Bu tasarım ise Şekil 4.25' de gösterilmiştir.



Şekil 4.25. İki yönlü röle kullanılarak yönsüz röle tasarımı

Yönlü rölede yapısındaki diyot nedeniyle akım sadece bir yönde geçer. Paralel port ileri – geri motoru aktif edildiğinde bu iki röleden biri anahtarı açar ve kamera kontrol motorunun çalışmamasını sağlar. Kameranın sağa veya sola hareketinde tekrar “0” noktasına dönmesi kullanıcı tarafından yapılmaktadır. Çünkü kamera motorunun gücü kendi pilinin durumuna bağlıdır ve motorun birim zamandaki hareketinin, dolayısıyla kameranın yer değiştirmesinin, hızı sabit değildir.



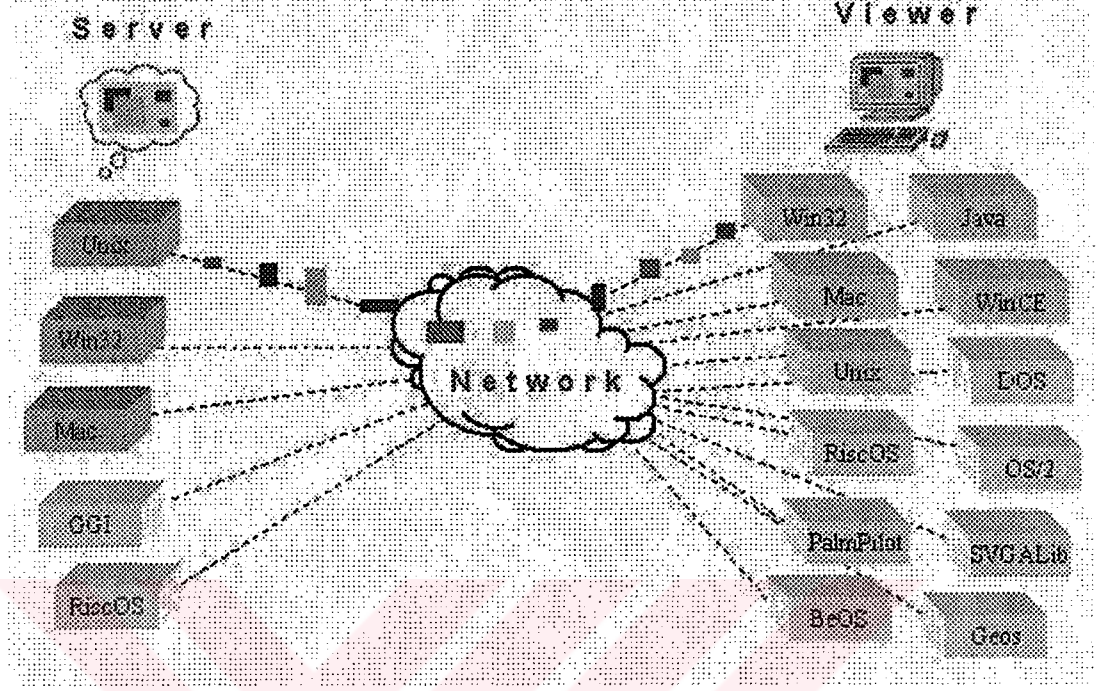
Şekil 4.26. Geliştirilen model araba

- 4. Adım: Kameradan elde edilen görüntünün internet üzerinde iletilmesi

Görüntünün internet üzerinden iletilmesinde Sanal Ağ İşlemesi (Virtual Network Computing – VNC) programından yararlanıldı. 1999 yılı başlarında AT&T Olivetti Araştırma Laboratuvarında VNC “Virtual Network Computing” (Sanal Ağ İşlemesi) üzerine çalışmaya başladı. VNC'yi kullanabilmek için TCP/IP ağ bağlantısının, bir vncserver ve vncviewer yazılımının çalıştığı makineye bağlantı kurabilmek için vncviewer'a ihtiyacı vardır (Tarbouriech, 2000).

VNC internet üzerindeki herhangi bir bilgisayarla etkileşimli bir bağlantı kurmaya izin vermektedir. Bu iki bilgisayarın, yani server ve viewer'ın, aynı yapıda olması

gerekmez. Endüstride, akademik dünyada ve iş hayatında çok sayıda kişiye hızlı ve kolay hizmet sağlamaktadır.



Şekil 4.27. VNC'nin platform bağımsızlığı

VNC'yi diğer uzaktan kontrol sistemlerinden ayıran 3 temel özelliği vardır;

1. Platform bağımsız olması : VNC birçok işletim sistemi için mevcuttur. VNC ile birbirine bağlanan iki bilgisayar üzerinde farklı işletim sistemleri olabilir (Windows, Linux, Solaris vb.)
2. Çok küçük ve kolay kullanılabilir olması: Windows için kullanılan viewer 150K büyüklüğünde ve bir disket sürücü üzerinden dahi çalışabilmektedir.
3. Kullanıma açık olması: Programın kendisi ve kodu serbestçe kullanılmaktadır.

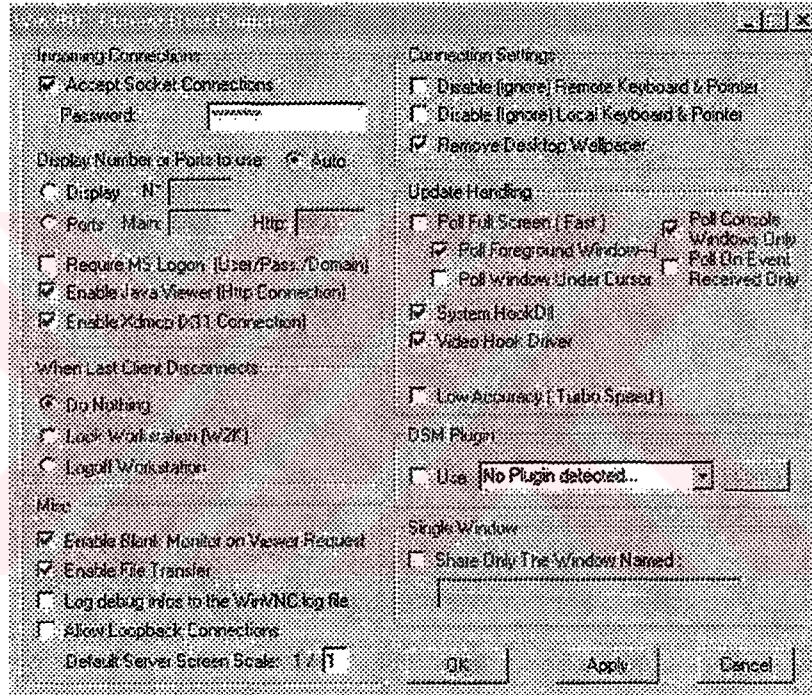
VNC iki ayrı bölümden oluşur;

- Sunucu Tarafı: Mevcut vncserver perl dilinde yazılmış bir programdır ve isteğe göre değişiklik yapmak mümkündür. vncserver ilk kullanımda bir şifre ister. Şifre bu

sunucuya bağlanmak için daha sonra gerekli olacaktır. Herhangi bir X sunucu için vncserver birçok seçenek sunmaktadır.

- İstemci Tarafı (Görüntüleyici): İstemci vncviewer denen tek bir komuttur. VNC sunucusuna bağlanmak için vncviewer'ın sunucunun IP adresini ve şifreyi yazması yeterlidir. Bu tüm görüntüleyiciler için her platformda mevcuttur.

Şekil 4.28'de kullanılan VNC programının sunucu bölümüne ait arayüz görülmektedir.



Şekil 4.28. VNC arayüzü

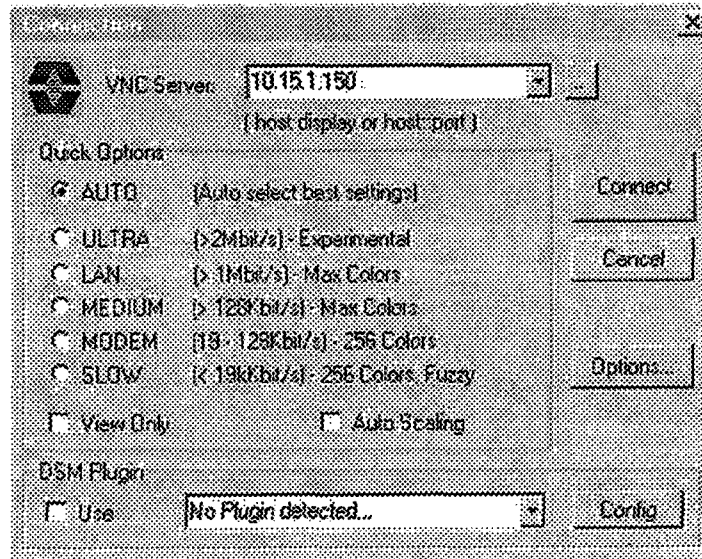
Yerel ağda, VNC oldukça hızlı çalışmaktadır. Herhangi bir işletim sisteminde seçilen bir program çalıştırılabilmektedir. Örneğin, eğer bir makinede internet bağlantısı yoksa, internet bağlantısı olan bir vnc sunucusuna bağlanmakta ve bağlanılan makinenin tarayıcısıyla internette sörf yapılabilmektedir.

Günümüzde birçok windows uygulaması geliştirilmekte ve kullanılmaktadır. Her yeni sürümün sunucuda ve istemcilerde güncellenmesi gerekmektedir. VNC

sayesinde her istemcide uygulama durdurabilir ve istenilen bir başka program yüklenebilir. Kısaca tüm ağ uzaktan, herhangi bir yerden, belli komutlarla kontrol edilmektedir.

Tüm iletişim programları aynı zamanda bir güvenlik açığı olarak görülmektedir. Yapılabilecek en doğru iş bu riski en aza indirmektir. Güvenli bir VNC için öncelikle güvenli bir ağın olması gerekir. Firewall, SSL, SSH güvenlik için kullanılabilir. Başka bir güvenlik yolu da bağlantı yapacak olan IP numaralarını sınırlandırmaktır (Green, 2002).

Proje kapsamında VNC ile farklı ortamlarda testler yapılmıştır. LAN ortamında yapılan testlerde veri aktarım hızı çok yüksek olduğu için aracın kontrolünde ve görüntünün aktarılmasında bir sorunla karşılaşılmamıştır. Dial-up internet bağlantısı olan iki bilgisayarla yapılan testlerde ise gecikmeler saptanmıştır. Şekilde gösterilen bağlantı hızına uygun seçeneklerle bazı düzenlemeler yapılabilmektedir. Dial-up bağlantısında veri iletim hızı saniyede 56Kbit in altında olduğundan "SLOW" seçeneğinde en hızlı kontrol ve görüntü iletimi gerçekleşmektedir. Yapılan testlerde görülmüştür ki; uzaktan paralel portta veri göndermede bir gecikme olmamıştır, görüntünün iletilmesiyle ilgili de Dial-up – Dial-up bağlantıda 3sn.'lik, ADSL – ADSL bağlantıda 1sn.'lik farklar saptanmıştır.



Şekil 4.29. VNC viewer ayarları

SONUÇ ve ÖNERİLER

Robot teknolojisi bilimsel ve teknolojik olarak kolektif bir çalışmanın ürünüdür. Bu teknolojinin üretilmesinde ve uygulamasında sayılamayacak kadar çok kişi ve kurum katkıda bulunmuş, yine çok geniş bir yelpaze içinde tanımlanan tüm bilim ve mühendislik kolları bu gelişim süreci içinde ve değişik düzeylerde rol almıştır. İnsan yaşamına benzeterek; robot teknolojisini emekleme çağını henüz yeni tamamlayarak, iki ayağı üzerinde yalnızca bir kaç adım atabilmiş küçük bir çocuk olarak düşünülebilir. Böyle bir insan yavrusunun gelişerek yetişmiş bir insan olarak ulaşacağı düzeyi, günümüzde içinde yaşadığımız robot teknolojisi düzeyiyle karşılaştırıldığında, bundan sonraki nesilleri nasıl bir teknolojinin beklediğini açıkça görülebilmektedir. Bu teknolojik gelişim sürecinin içinde yaşamak, bu süreci gözlemlemek, bu ilk adımlardan yararlanmak gerçekten heyecan verici bir olgu. Bu heyecanı duyan ve yaşayan insanların robot teknolojisine gelecek yıllarda yapacakları daha yoğun ve verimli katkılarıyla, önümüzdeki yıllar, tüm insanların ve toplumların konfor ve güvenliğinde hissedilir önemli değişimlere neden olacaktır.

Bu tez de robotik uygulamaların gelişiminde ufak bir adım olarak düşünülmüş, uzaktan kontrol edilebilen, internet üzerinden görüntünün iletimini sağlayan bir araç geliştirilerek Türkiye’de bu konuda yapılabilecek çalışmalara öncülük edebilmeyi amaçlanmıştır.

Projenin gelişim süreci kısaca şöyle özetlenebilir; ilk aşama aracın bilgisayarın paralel portunu kullanarak kontrolünü sağlanmıştır. Bunu gerçekleştirebilmek için tasarlanan elektronik devre ve program ilgili bölümlerde ayrıntılı olarak anlatılmıştır. Aracın üzerine entegre edilen telsiz kamera ve Dazzle sayesinde görüntüyü bağlı bulunduğu bilgisayar ekranına göndermektedir. Daha sonra ki aşamada ise, bu kontrol ve görüntüleme işlemini internet üzerinden bağlanılabilecek uzaktaki bir

bilgisayar üzerinden yapabilmektedir. Bu testler özellikle farklı internet bağlantılarında yapılmıştır. Testlerin sonucunda başarılı sonuçlar elde edilmiştir. Gelişim süreci içinde rastlanan tek sorun internet üzerinden veri iletimi sırasında olagelen yavaşlıktır. Bu kullanıcılara veya tasarıma bağlı bir sorun değildir. Yerel ağlarda yapılan testlerde böyle bir sorunla karşılaşılmamıştır. Bunlara ek olarak, gezer aracın kinematik modeli çıkartılmış, belirli örnekleme aralıklarında aracın durumu ve yörüngesi hesaplanarak Matlab'da çizilmiştir.

Bu tezin en büyük özelliklerin biri ise, geliştirilmeye açık bir tasarım olmasıdır. Üzerine eklenebilecek farklı amaçlı algılayıcılarla bu araç çok çeşitli alanlarda, eğitim, savunma teknolojileri veya tıp gibi, hizmet verebilir.



KAYNAKLAR

1. AXELSON, J., 1999. Parallel Port Complete; programming, interfacing and using the PC' s parallel port, Lakeview Research, Madison.
2. BİNGÜL, Z., SEKMEN, A. Ş., KOKU, A. B. 2001. Hareketli Robot Kontrolünde Mesafe ve Mekan Önbilgisinin Önemi, Siu, 2001, KKTC.
3. ÇAĞILTAY, K., 1994. Herkes İçin İnternet, Tübitak, Ankara.
4. ERDEN, A., 2003. Robotik, Bilim ve Teknik Dergisi, <http://www.biltek.tubitak.gov.tr/postervekitapciklar/kitapciklar/robotik1.pdf>, Ankara.
5. FEIT, S., 1999. TCP/IP: architecture, protocols, and implementation with Ipv6 and IP security, McGraw – Hill, New York.
6. FOROUZAN, B.A., 2000. TCP/IP Protocol Suite, McGraw – Hill, Boston.
7. GREEN, J.H., 2002. Voice and Video over IP, McGraw – Hill, New York.
8. GÜRBÜZ, R., 2004. Türkiye’de Mekatronik Eğitimi, ITHET 2004 Konferansı, İstanbul.
9. HUGGINS, B., 2000. Telerobotics Senior Projects, <http://cegt201.bradley.edu/projects/proj2000/prjtelrb/>, Bradley Üniversitesi, Illinois.
10. MALINOWSKI, A., 2000. Bradley Web – Controlled Robots, <http://gdansk.bradley.edu/olekmali/projects/telebot/>, Bradley Üniversitesi, Illinois.
11. MANN, C., 2004. Technology Review, A Remote Control For Your Life, Massachusetts.
12. McNAMEE, L., 2003. Photogrammetric Calibration of Mobile Robot Kinematics, Ottawa.
13. MUIR, P.F., NEUMAN, C. P., 1986. Kinematic Modeling of Wheeled Mobile Robots, Robotics Institute Technical Report No. CMU – RI – TR – 86 – 12, Pittsburgh.
14. LUO, R., 2003. Proceedings of the IEEE, Special Issue on Networked Intelligent Robots Through The Internet, p. 367 – 369, New York.
15. LUO, R., 2003. Proceedings of the IEEE, Networked Intelligent Robots Through The Internet: Issues and Opportunities, p. 371 – 380, New York.

16. PESCOVITZ, D., 2005. Sony Research' s Parisian Play Station, http://www.technologyreview.com/articles/05/01/wo/wo_pescovitz011005.asp, Paris
17. PORT, O., 2003. Business Week, Issue.3851, Washington.
18. SIEGWART, R., NOURBAKHSI, I. R. 2004. Introduction to Autonomous Mobile Robots, The MIT Press, p. 47 – 61, Massachusetts.
19. SIURU, B., 1997. Electronics Now, Remote Control Robot – Planetary Explorer, vol.68, Pittsburgh.
20. TAN, J., CLAPWORTHY G.J., 2003. Proceedings of the IEEE, Virtual Environments for Internet-Based Robots, p.389 – 390, New York.
21. TARBOURIECH, G., 2000. Virtual Network Computing, Pittsburgh.
22. Türk Dil Kurumu, Bilgisayar Terimleri Sözlüğü, <http://www.tdk.gov.tr>.
23. TYSON, J., 2004. Radio Control, stuffo.howstuffworks.com, North Carolina.
24. WAGNER, M., 1998. Robotic Antarctic Meteorite Search, <http://www.frc.ri.cmu.edu/projects/meteorobot/Nomad/Nomad.html#Atacama>, Pittsburgh.
25. WEBSTER, P., RICHARDSON T., WEATHERALL, J., WOOD, K., McLACHLAN, C., HARTER A., STAFFORD-FRASER, Q., 2004. Virtual Network Computing, <http://www.uk.research.att.com/archive/vnc/>, Cambridge.

EKLER

Ek – A : Geliştirilen kod

```
unit mainwnd;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
Videocap, videodisp, vfw, StdCtrls, ExtCtrls, Buttons, syncobjs;
```

```
type
```

```
TMainForm = class(TForm)
```

```
Panel1: TPanel;
```

```
Btformat: TButton;
```

```
BtDriver: TButton;
```

```
Panel3: TPanel;
```

```
Label13: TLabel;
```

```
BTCompress: TBitBtn;
```

```
VideoCap1: TVideoCap;
```

```
BtStop: TButton;
```

```
BtStart: TButton;
```

```
EdFile: TEdit;
```

```
Label2: TLabel;
```

```
Label1: TLabel;
```

```
EdCap: TEdit;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
EdPaly: TEdit;
```

```
Timer1: TTimer;
```

```
sol_ileri_button: TSpeedButton;
```

```
sol_geri_button: TSpeedButton;
```

```
ileri_button: TSpeedButton;
```

```
geri_button: TSpeedButton;
```

```
sag_ileri_button: TSpeedButton;
```

```
sag_geri_button: TSpeedButton;
```

```
Camera_Left1: TSpeedButton;
```

```
Camera_Right1: TSpeedButton;
```

```
procedure FormCreate(Sender: TObject);
```

```
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
procedure BtFormatClick(Sender: TObject);
```

```
procedure BtStartClick(Sender: TObject);
```

```
procedure BtStopClick(Sender: TObject);
```

```

procedure BtDriverClick(Sender: TObject);
procedure Video1Db1Click(Sender: TObject);
procedure BTCompressClick(Sender: TObject);
procedure BtCapClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure sol_ileri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure sol_ileri_buttonMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure ileri_buttonMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure ileri_buttonMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure sag_ileri_buttonMouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure sag_ileri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure sol_geri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure sol_geri_buttonMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure geri_buttonMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure geri_buttonMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure sag_geri_buttonMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure sag_geri_buttonMouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure FormKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure Camera_Left1MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure Camera_Right1MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure Camera_Right1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure Camera_Left1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
private
count:integer; // Frame Counter
{ Private-Deklarationen}
public
{ Public-Deklarationen}
end;

var

```



```

MainForm: TMainForm;
implementation

uses DlgTreiber;

{$R *.DFM}
procedure PortOut(Port : Word; Data : Byte); stdcall; external 'io.dll';
function PortIn(Port:Word):Byte; stdcall; external 'io.dll';

procedure TMainForm.FormCreate(Sender: TObject);
begin
VideoCap1.Videopreview:= true;
end;

procedure TMainForm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
VideoCap1.StopCapture;
end;

procedure TMainForm.BtFormatClick(Sender: TObject);
begin
VideoCap1.DlgVFormat;
end;

procedure TMainForm.BtStartClick(Sender: TObject);

begin
Videocap1.VideoFileName:=EdFile.text;

VideoCap1.MicrosecPerFrame:= strtoint(edPaly.text)*1000*1000;

VideoCap1.CapSingleFramesOpen;
Timer1.Interval:= strtoint(edCap.text) * 1000;
BtStart.Enabled:= false;
btStop.Enabled:= true;
btFormat.Enabled:= false;
btDriver.Enabled:= false;
btCompress.Enabled:= false;
count:=0;
Timer1.Enabled:= true;
end;

procedure TMainForm.BtStopClick(Sender: TObject);
begin
Timer1.Enabled:= false;
VideoCap1.CapSingleframesClose;
BtStart.Enabled:= true;
btStop.Enabled:= false;
btFormat.Enabled:= true;

```

```

btDriver.Enabled:= true;
btCompress.Enabled:= true;

end;

procedure TMainForm.BtDriverClick(Sender: TObject);

Var DrvList:TStrings;
begin
DlgEinstell:=TDlgEinstell.Create(Self);
drvList:= GetDriverList;
dlgEinstell.Combobox1.Items:= drvList;
VideoCap1.DriverOpen:= false;
dlgEinstell.ComboBox1.Itemindex:= VideoCap1.DriverIndex;
if DlgEinstell.ShowModal = mrOK then
begin
videoCap1.DriverIndex:= dlgEinstell.combobox1.ItemIndex;
end;
VideoCap1.DriverOpen:= true;
VideoCap1.Videopreview:= true;
DlgEinstell.Free;
drvList.Clear;
drvList.Free;
end;

procedure TMainForm.Video1DbClick(Sender: TObject);
begin
if BtStart.Enabled Then BtStartClick(Sender) else BtStopClick(Sender);
end;

procedure TMainForm.BTCompressClick(Sender: TObject);
begin
VideoCap1.DlgVCompression;
end;

procedure TMainForm.BtCapClick(Sender: TObject);
begin
VideoCap1.CapSingleFrame;
end;

procedure TMainForm.Timer1Timer(Sender: TObject);
begin
VideoCap1.CapSingleFrame;
end;

procedure TMainForm.sol_ileri_buttonMouseUp(Sender: TObject; Button:
TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin

```

```
PortOut(888,0);  
end;
```

```
procedure TMainForm.ileri_buttonMouseUp(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
end;
```

```
procedure TMainForm.sag_ileri_buttonMouseUp(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
end;
```

```
procedure TMainForm.sol_geri_buttonMouseUp(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
end;
```

```
procedure TMainForm.geri_buttonMouseUp(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
end;
```

```
procedure TMainForm.sag_geri_buttonMouseUp(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
end;
```

```
procedure TMainForm.sag_geri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
PortOut(888,10);  
end;
```

```
procedure TMainForm.sol_ileri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);
```

```
PortOut(888,5);  
end;
```

```
procedure TMainForm.ileri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
PortOut(888,1);  
end;
```

```
procedure TMainForm.sag_ileri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
PortOut(888,9);  
end;
```

```
procedure TMainForm.sol_geri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
PortOut(888,6);  
end;
```

```
procedure TMainForm.geri_buttonMouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,0);  
PortOut(888,2);  
end;
```

```
procedure TMainForm.FormKeyDown(Sender: TObject; var Key: Word;  
Shift: TShiftState);  
begin  
case key of  
VK_UP: begin  
PortOut(888,0);  
PortOut(888,1);  
end;  
VK_RIGHT: PortOut(888,4);  
VK_LEFT: PortOut(888,8);  
VK_DOWN: begin  
PortOut(888,0);  
PortOut(888,2);  
end;  
VK_INSERT: begin  
PortOut(888,0);  
PortOut(888,9);  
end;
```

```

VK_DELETE: begin
PortOut(888,0);
PortOut(888,10);
end;
VK_HOME: begin
PortOut(888,0);
PortOut(888,1);
end;
VK_END : begin
PortOut(888,0);
PortOut(888,2);
end;
VK_PRIOR: begin
PortOut(888,0);
PortOut(888,5);
end;
VK_NEXT: begin
PortOut(888,0);
PortOut(888,6);
end;
else
PortOut(888,0);
end;

end;

procedure TMainForm.FormKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
PortOut(888,0);
end;

procedure TMainForm.Camera_Left1MouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);
end;
procedure TMainForm.Camera_Right1MouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,0);
end;

procedure TMainForm.Camera_Right1MouseDown(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
PortOut(888,8);
end;

```

```
procedure TMainForm.Camera_Left1MouseDown(Sender: TObject;  
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  
begin  
PortOut(888,4);  
end;  
  
end.
```



ÖZGEÇMİŞ

1977 yılında İel'de doğdu. İlk ve Orta öğrenimini Ankara'da, lise öğrenimi Kocaeli'de tamamladı. 1994 yılında girdiđi Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü'nden 1999 yılında Teknik Öğretmen olarak mezun oldu.

2000 yılından bu yana İstanbul Bilgi Üniversitesi Fen Edebiyat Fakültesi Bilgisayar Bilimleri Bölümü'nde Araştırma Görevlisi olarak görev yapmaktadır.

