



T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ

SERİ PORTTAN KABLOSUZ AĞ ÜZERİNE
HABERLEŞİLEBİLİR KAMERALI ARAÇ KONTROLÜ

ÖMER ÖNAL

YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK ANABİLİM DALI

KAHRAMANMARAŞ 2011

**T.C.
KAHRAMANMARA SÜTÇÜ MAMÜN VERS TES
FEN B L MLER ENST TÜSÜ**

**SER PORTTAN KABLOSUZ A LE
HABERLE EB LEN KAMERALI ARAÇ KONTROLÜ**

ÖMER ÖNAL

**Bu tez,
Elektrik Elektronik Anabilim Dalında
YÜKSEK L SANS
derecesi için hazırlanmıştır.**

KAHRAMANMARA 2011

Kahramanmara Sütçü mam Üniversitesi Fen Bilimleri Enstitüsü ö rencisi Ömer ÖNAL tarafından hazırlanan “Seri Porttan Kablosuz A ile Haberle ebilen Kameralı Araç Kontrolü” adlı bu tez, jürimiz tarafından 07 / 01 / 2011 tarihinde oy birli i ile Elektrik Elektronik Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmi tir.

Prof. Dr. M. Kemal KIYMIK (DANI MAN)

Elektrik Elektronik Müh., Kahramanmara Sütçü mam Üniversitesi

Doç. Dr. A. Serdar YILMAZ (ÜYE)

Elektrik Elektronik Müh., Kahramanmara Sütçü mam Üniversitesi

Yrd. Doç. Dr. Muharrem MAL (ÜYE)

Makine Müh., Kahramanmara Sütçü mam Üniversitesi

Yukarıdaki imzaların adı geçen ö retim üyelerine ait oldu unu onaylarım

Prof. Dr. M. Hakkı ALMA

Fen Bilimleri Enstitüsü Müdürü

TEZ B LD R M

Tez içindeki bütün bilgilerin etik davranı ve akademik kurallar çerçevesinde elde edilerek sunuldu unu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalı mada orijinal olmayan her türlü kayna a eksiksiz atıf yapıldı ını bildiririm.

Ömer ÖNAL

Bu çalı ma Kahramanmara Sütçü mam Üniversitesi Bilimsel Ara tırma Projeleri Koordinasyon Birimi Ba kanlı ı tarafından desteklenmi tir.

Proje No: 2010/5-5YLS

Not: Bu tezde kullanılan özgün ve ba ka kaynaktan yapılan bildiri lerin, çizelge, ekil ve foto rafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

**SERİ PORTTAN KABLOSUZ ARAÇLARLA HABERLEŞİLEN
KAMERALI ARAÇ KONTROLÜ
(YÜKSEK LİSANS TEZİ)**

ÖMER ÖNAL

ÖZ

Bu çalışmada, DC motorlu bir mobil aracın RF aracılığıyla bir bilgisayar tarafından kontrolü sağlanmıştır. Araç üzerine yerleştirilen kamera sayesinde bilgisayar başındaki kullanıcı kumanda ettiği aracın gittiği yerleri rahatlıkla görebilmektedir.

Sistem üç ana kısımdan oluşmaktadır. Bunlar, bilgisayar ve kullanıcı arayüzü programı, haberleşme kartı ve araç üzerinde bulunan alıcı kartıdır. Kullanıcıların aracı istedikleri şekilde yönetebilmeleri için kullanımı kolay bir arayüz programı hazırlanmıştır. Arayüz programından gelen verileri seri porttan alarak alıcı kartta bulunan antene iletmek için haberleşme kartı tasarlanmıştır. Ardından ise mikrodenetleyici ve ilgili entegreler birleştirilerek alıcı kartı oluşturulmuştur.

Çalışma esnasında hızı ve esnekliği göz önünde bulundurularak kullanıcı arayüz programında Delphi 10.0 tercih edilmiştir. Mikrodenetleyici de ise PIC-C programlama dili kullanılmıştır.

Anahtar Kelimeler: Uzaktan Kontrol; RF Sinyalleri; Seri Port

Kahramanmaraş Sütçü İmam Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Anabilim Dalı, 01 / 2011

Danışman: Prof. Dr. M. Kemal KIYMIK

Sayfa sayısı: 75

MOB L ROBOT CONTROL VIA WIRELESS NETWORK USING SERIAL PORT (M.Sc. THESIS)

ÖMER ÖNAL

ABSTRACT

The Purpose of this Project is DC motor control from a computer via RF connection. The user can monitor where the car goes from the computer with camera installed on the car.

The system consists of three main parts. These are computer and user interface program, transmit communication board which is connected to computer's serial port and receive communication board which is located on the car. A user friendly interface program has been developed for directing the car from the computer. A communication card was designed to send data generated from the serial port interface to the receive antenna. After that a receive card was designed by using a microcontroller and integrated circuits.

The programming of the GUI (Graphical User Interface) was chosen as Delphi 10 because of the high flexibility and performance of the program. Also PIC-C programming language was used for Microcontroler Programing.

Key Words: Remote Control, RF Signals, Serial Port

Kahramanmara Sütçü mam University
Institute for Graduate Studies in Science and Technology
Department of Electric & Electronics 01 / 2011

Supervisor: Prof. Dr. M. Kemal KIYMIK

Page Number: 75

SERİ PORTTAN KABLOSUZ ARAÇ İLE HABERLEŞEN KAMERALI ARAÇ KONTROLÜ

ÖZET

Kablosuz araç ile seri porttan araç kontrolü üzerine yapılan bu çalışmada amacımız insan hayatını tehlikelerden korumak ve kolaylaştırmaya yönelik sistemler için bir temel oluşturulmasıdır. Proje kapsamında, bilgisayarın standart seri portu RS232 ile haberleşerek RF vasıtasıyla kontrol edilebilen ve üzerinde bulunan kamera aracılığıyla görüntü aktarabilen bir araç tasarlanacaktır.

Tasarım donanımsal olarak iki temel kısımdan oluşmaktadır. İlk kısım olan haberleşme kartında, bilgisayara bağlı seri port ve bu seri porttan aldığı verileri karttaki alıcıya aktaracak olan verici bulunmaktadır. Verici kartında karakter göndermek için hem ucuz olması hem de kolay kullanımı sebebiyle ATX-34 verici entegresi seçilmiştir. İkinci kısımda yani alıcı kartında ise yollanan verileri alarak bunları mikrodenetleyiciye aktaracak olan ARX-34 alıcı entegre devresi seçilmiştir. Alıcı kartı üzerine, bilgisayardan gelen komutu yorumlayıp istenilen hareketi, mobil araca aktarmak üzere mikrodenetleyici yerleştirilmiştir. Mikrodenetleyici olarak Microchip firmasının PIC18F1320 entegresi seçilmiştir.

Mikrodenetleyici bir adet step motor sürücü entegresi ile beş adet röleyi kontrol etmektedir. Mobil araçta bulunan DC motorlar röle ile sürülmektedir. Araçın hareketi arka kısımda bulunan bir adet DC motor ile sağlanmaktadır. Bu motorlar yüksek torka fakat düşük dönme hızına sahiptirler. Mobil aracın sağ ve sola dönebilmesi ön tekerleklerde bulunan mekanik sayesinde gerçekleşmektedir. Bu mekanizmanın kontrolü DC motor sayesinde olmaktadır. Tepede bulunan kamera ise step motor ile kontrol edilmekte ve röle ile açılıp kapanmaktadır. Kamera aldığı görüntüyü RF ile bilgisayara aktarmaktadır. Kullanılan tüm bu entegrelerin uç ayrıntıları ve iç yapıları ilerleyen bölümlerde açıklanacaktır.

Projenin üçüncü temel kısmı ise yazılım kısmıdır. Mobil aracın sıradan bir kullanıcı tarafından kullanılabilmesi ve kameradan gelen görüntüyü izleyebilmek için bilgisayarda bir arayüz programı yazılmıştır. Bu arayüz programı yapılırken kolaylığı ve görsel özellikleri sebebiyle Delphi 10.0 programı seçilmiştir. Ayrıca mobil araç üzerindeki mikrodenetleyicinin programlanması PIC C dili ile gerekli yazılımlar hazırlanmıştır.

MOB L ROBOT CONTROL VIA WIRELESS NETWORK USING SERIAL PORT

SUMMARY

In this project a remote controlled car on top of that with a camera was designed in order to help human being in such places where might be dangerous or not possible for human being to enter in. In the scope of this project, a car which communicates with computer via RS232 serial port, using RF signals, was designed and a camera which sends the images around to computer was placed on it.

The design consists of two parts regarding to hardware. The first part which is called as communication board consists of a port which communicates with serial port and a transmitter which sends the commands to the receiver. As it was easier and cheaper to use, ATX-34 which is a product of UDEA company was used as transmitter. The second part which is called control board, consist of an ARX-34 receiver, PIC 18F1320 microcontroller and car mechanism.

The microcontroller controls five relays with a step motor driving chip and the two dc motors are driven by relays. The forward and backward movement of vehicle is provided by the dc motor in vehicle's rear side. These motors have high torque but low speed of rotation. The vehicle can turn right and left by using the mechanism in the vehicle's front side with the help of a dc motor. The camera on top the vehicle is controlled by using a step motor and turned on and off with a relay. The camera sends the images around to the computer via RF.

The third part of the project is software. In order to provide an easy usage of the system for non professionals, a computer user interface was designed to watch the images from the camera. Due to its easy of use and visuality, Delphi 10.0 was used for this user interface. And finally the microcontroller on the vehicle was programmed with PIC C.

TE EKKÜR

Tez çalı malarım boyunca Yüksek Lisans tezimi yöneten ve her konuda kendisinden destek gördü üm danı man hocam Sayın Prof. Dr. M. Kemal KIYMIK ve programlama konusunda bilgi ve tecrübesiyle yardımlarını esirgemeyen Elektrik-Elektronik Mühendisi Sayın Ejder UTU 'a sonsuz te ekkürlerimi sunarım.

“Seri Porttan Kablosuz A ile Haberle ebilen Kameralı Araç Kontrolü” adlı tezimde maddi destekte bulunan Kahramanmara Sütçü mam Üniversitesi Bilimsel Ara tırma Projeleri Koordinasyon Birimi Ba kanlı ına te ekkürlerimi sunarım.

Ç NDEK LER

	Sayfa No
ÖZ.....	I
ABSTRACT	II
ÖZET.....	III
SUMMARY	IV
TE EKKÜR	V
Ç NDEK LER.....	VI
Ç ZELGELER D Z N	IX
EK LLER D Z N	X
S MGELER ve KISALTMALAR D Z N.....	XII
1.G R	1
1.1. Mikrolemci Nedir?.....	2
1.2. Mikrodenetleyici Nedir?.....	2
1.2.1. Mikrolemci ve Mikrodenetleyicinin Kar ıla tırılması.....	3
1.2.2. Mikrodenetleyiciler Hakkında Genel Bilgiler	3
1.2.3. PIC Mikrodenetleyicilerin Tercih Sebepleri.....	4
1.2.4. PIC Türleri.....	5
2. ÖNCEK ÇALI MALAR.....	6
3.MATERYAL ve METOT	8
3.1.Materyal.....	8
3.1.1. Sistemin Genel Yapısı ve Çalışma Prensipleri	8
3.1.1.1. Sistemin Blok Diyagramı	8
3.1.1.2. Haberleşme Kartı.....	9
3.1.1.3. Alıcı Kartı	9
3.1.1.4. Kullanıcı Ara Yüz Programı.....	9
3.1.1.5. Sistemin Çalışması	9

3.1.2. Verici Kartında Kullanılan Entegreler ve Özellikleri	10
3.1.2.1. Seri Port ve RS232 Standartları.....	10
3.1.2.2. ATX-34 Verici Anten Entegresi	14
3.1.2.3. MAX232.....	17
3.1.2.4. Verici Kartının Bağlantı Ekli	18
3.1.2.5. Verici Kartının Genel Görünümü.....	19
3.1.3. Araçta Kullanılan Entegreler Ve Özellikleri	19
3.1.3.1. PIC18F1320 Mimarisi Ve Özellikleri	20
3.1.3.1.1. PIC18F1320' nin Pin Diyagramı.....	20
3.1.3.1.2. PIC18F1320' nin İç Yapısı.....	22
3.1.3.1.3. Osilatör Uçları ve Çeşitleri.....	23
3.1.3.1.4. Besleme Gerilimi.....	23
3.1.3.2. Röle ile DC Motor Sürmek.....	23
3.1.3.3. ARX-34 Alıcı Anten Entegresi.....	25
3.1.3.4. Rf Sinyalleri ile İletim Kurulan Kamera.....	26
3.1.3.5. Güç Kaynağı	27
3.1.3.6. 7805 Regülâtörü	27
3.1.3.7. DC Motorlar ve Aracın Hareket Ekli	28
3.1.3.8. Step Motor ve ULN2003 Sürücüsü	30
3.1.3.9. Alıcı Kartının Devre Şeması.....	32
3.1.3.10. Aracın Genel Görünümü	33
3.2. Metot.....	33
3.2.1. Mikrodenetleyici Yazılımı.....	33
3.2.1.1. Mikrodenetleyiciye Program Kodunun Yüklenmesi	33
3.2.2. Kullanıcı Ara Yüz Programı.....	35
4.BULGULAR ve TARTIŞMA	36
4.1. Uygulamanın CCS C Programı Kısmı.....	36

4.1.1. CCS C Derleyicisinde Yeni Bir Proje Olu turmak	36
4.1.2. CCS C' de Program Derlemek	39
4.1.3. CCS C' de Derleme Sonucu Olu an Dosya Formatları.....	40
4.1.4. CCS C' de Program Kodlarının Açıklaması	42
4.1.4.1. CCS C' de Öni lemci Fonksiyonları	46
4.1.4.2. #USE RS232() Fonksiyonu	47
4.1.4.3. RS232 Giri /Çıkı Fonksiyonları	48
4.1.4.4. RS232 Seri leti im Kesmeleri	48
4.2. Uygulamanın Arayüz Programı Kısmı	49
4.2.1. Delphi Kodlarının Açıklamaları	49
5. SONUÇ VE ÖNER LER	56
KAYNAKLAR.....	57
EKLER	58
EK-1: PIC C Kodları.....	58
EK-2: Delphi Kodları.....	63
EK-3: PIC C de Ön lemci Fonksiyonları	71
ÖZGEÇM	75

Ç ZELGELER D Z N

	Sayfa No
Çizelge 3.1. Verici Kartında Bulunan Elemanlar	10
Çizelge 3.2. 9 Pin’li RS–232 Port’unun Pin’lerinin Açıklamaları	14
Çizelge 3.3. ATX–34’un Uç Ayrıntıları.....	15
Çizelge 3.4. ATX–34’un Elektriksel Özellikleri.....	17
Çizelge 3.5. Alıcı Kartı Eleman Listesi.....	19
Çizelge 3.6. PIC18F1320’nin Osilatör Çe itleri	23
Çizelge 3.7. ARX-34’ün Uç Ayrıntıları	26
Çizelge 4.1. CCS C Derleyicisi, RS232 Giri /Çıkı Fonksiyonları	48

EK LLER D Z N

	Sayfa No
ekil 1.1. Mikrodenetleyici Sistem	4
ekil 3.1. Sistemin Blok Diyagramı	8
ekil 3.2. Paralel İletim	10
ekil 3.3. Seri İletim	11
ekil 3.4. Senkron ve Asenkron İletim	12
ekil 3.5. 9 ve 25 Pin’li RS232 Dişi ve Erkek Seri Portlar	13
ekil 3.6. RS-232 9 Pin’li Dişi Seri Port’un Fiziksel Görünümü ve Yapısı	13
ekil 3.7. ATX-34 ve ARX-34 Entegreleri	14
ekil 3.8. ATX-34 Genel Görünümü	15
ekil 3.9. Data Formatı	16
ekil 3.10. MAX232 Entegresinin Bacak Bağlantıları ve Ç Yapısı	18
ekil 3.11. Verici Kart Devre Şeması	18
ekil 3.12. Verici Kartının Genel Görünümü	19
ekil 3.13. PIC18F1320’nin Özellikleri	20
ekil 3.14. PIC18F1320’nin Dış Görünümü	21
ekil 3.15. PIC18F1320’nin Pin Diagramı	21
ekil 3.16. PIC18F1320’nin Ç Yapısı	22
ekil 3.17. Çift Kontaklı Röle	24
ekil 3.18. Röle ile DC Motor Sürücü Devresi	24
ekil 3.19. ARX-34’nin Genel Görünümü	25
ekil 3.20. Kablosuz Kamera	27
ekil 3.21. 7805’in Blok Yapısı	28
ekil 3.22. Fırçalı DC Motor	28
ekil 3.23. DC Motor Dış Görünümü	29
ekil 3.24. Step Motor Kontrol Devresi	30

ekil 3.25. Step Motor Çalışma ekli	30
ekil 3.26. ULN2003 Pin Diagramı	31
ekil 3.27. ULN2003 ile Step Motor Kontrolü	32
ekil 3.28. Alıcı Kartın Devre eması	32
ekil 3.29. Mobil Aracın Genel Görünü ü	33
ekil 3.30. PIC Kıt_2 Yükleme Kartı.....	34
ekil 3.31. HEX Dosyasının Mikrodenetleyiciye Yüklmesi	34
ekil 3.32. TV Kartı Ayarları	35
ekil 4.1. Kaynak Dosya Olu turma	36
ekil 4.2. Kaynak Dosyanın Kayıt Yeri ve sminin Belirlenmesi.....	37
ekil 4.3. Program Yazılan Ekran	38
ekil 4.4. “Project Options” Menüsü	38
ekil 4.5. Proje Dosyasının Derlenmesi	39
ekil 4.6. Proje Dosyasının Derlenme lemi Sonucu.....	40
ekil 4.7. Derleme Sonucu Olu an Dosyalar	42
ekil 4.8. Ara Yüz Ekran Paneli.....	49
ekil Ek.A.1. CCS C’de Ön lemciler Fonksiyonları	74

S İMGELER ve KISALTMALAR D Z N

DC	: Direct Current
EPROM	: Erasable Programmable Read Only Memory
EEPROM	: Electrical Eprom
RF	: Radio Frequency
PIC	: Peripheral Integrated Circuit
RAM	: Random Access Memory
ROM	: Read Only Memory
TTL	: Transistor Transistor Logic
USART	: Universal Synchronous Asynchronous Reciever
CPU	: Central Processing Unit
RPM	: Rotations Per Minute
BPS	: Bits Per Second
CCS	: Custom Costumer Services
I/O	: Input/Output
RISC	: Reduced Instruction Set Computer
PWM	: Pals Geni lik Modülasyonu
A/D	: Analog/Dijital
CE	: Collector Emitter
PC	: Personel Computer
Hz	: Hertz
Mhz	: Mega Hertz
F	: Farad
µf	: Mikrofarad
V	: Volt
A	: Amper
F	: Frekans

1. G R

Kablosuz a ile seri porttan araç kontrolü üzerine yapılan bu çalı mada amacımız insan hayatını tehlikelerden korumak ve kolayla tırmaya yönelik sistemler için bir temel olu turmaktır. Proje kapsamında, bilgisayarın standart seri portu RS232 ile haberle erek RF vasıtası ile kontrol edilebilen ve üzerinde bulunan kamera aracılı ı ile görüntü aktarabilen bir araç tasarlanacaktır.

Tasarım donanımsal olarak iki temel kısımdan olu maktadır. İlk kısım olan haberle me kartında, bilgisayara ba lı seri port ve bu seri porttan aldı ı verileri kar ıdaki alıcıya aktaracak olan verici bulunmaktadır. Verici kartında karakter göndermek için hem ucuz olması hem de kolay kullanımı sebebiyle ATX-34 verici entegresi seçilmi tir. İkinci kısımda yani alıcı kartında ise yollanan verileri alarak bunları mikrodenetleyiciye aktaracak olan ARX-34 alıcı entegre devresi seçilmi tir. Alıcı kartı üzerine, bilgisayardan gelen komutu yorumlayıp istenilen hareketi, mobil araca aktarmak üzere mikrodenetleyici yerle tirilmi tir. Mikrodenetleyici olarak Microchip firmasının PIC18F1320 entegresi seçilmi tir. Bunun sebebi bu entegrenin harici bir osilatör gerektirmemesi ve USART arabirimine sahip olması ve sistemin geli tirilmesine yönelik birçok geli mi arabirime sahip olmasıdır.

Mikrodenetleyici bir adet step motor sürücü entegresi ile be adet röleyi kontrol etmektedir. Mobil araçta bulunan DC motorlar röle ile sürülmektedir. Aracın hareketi arka kısımda bulunan bir adet DC motor ile sa lanmaktadır. Bu motorlar yüksek torka fakat dü ük dönme hızına sahiptirler. Böyle seçilmelerinin nedeni aracın herhangi bir zorlanma kar ısında (a ırı e im) sabit hızına devam edebilmesidir. Mobil aracın sa ve sola dönebilmesi ön tekerleklerde bulunan mekanik sayesinde gerçekleştirilmektedir. Bu mekanik in kontrolü DC motor sayesinde olmaktadır. Tepede bulunan kamera ise step motor ile kontrol edilmekte ve röle ile açılıp kapanmaktadır. Kamera aldı ı görüntüyü RF ile bilgisayara aktarmaktadır. Kullanılan tüm bu entegrelerin uç ayrıntıları ve içyapıları ilerleyen bölümlerde açıklanacaktır.

Projenin üçüncü temel kısmı ise yazılım kısmıdır. Mobil aracın sıradan bir kullanıcı tarafından da kullanılabilmesi ve kameradan gelen görüntüyü izleyebilmek için bilgisayarda bir ara yüz programı yazılmı tir. Bu ara yüz programı yapılırken kolaylı ı ve görsel özellikleri sebebiyle Delphi 10.0 programı seçilmi tir. Ayrıca mobil araç üzerindeki mikrodenetleyicinin programlanması PIC C dili ile gerekli yazılımlar hazırlanmı tir.

1.1. Mikrolemci Nedir?

Günümüzde kullanılan bilgisayarların özelliklerinden bahsedilirken 80386, 80486, Pentium-II, Pentium-III gibi kelimeler birer mikrolemcidir (Microprocessor). Mikrolemciler bilgisayar programlarının yapmak istediği tüm işlemleri yerine getirdiği için, çoğu zaman merkezi işlem ünitesi (CPU- Central Processing Unit) olarak da adlandırılır. PC adını verdiğimiz kişisel bilgisayarlarda kullanıldığı gibi, bilgisayarla kontrol edilen sanayi tezgâhlarında ve ev aletlerinde de kullanılabilir. Bir mikrolemci işlevini yerine getirebilmesi için ana kartındaki yardımcı elemanlara ihtiyaç duyar. Bunlar:

1. Input (Giriş) ünitesi.
2. Output (Çıkış) ünitesi.
3. Memory (Bellek) ünitesi.

Bu üniteler CPU chip'inin dışında, bilgisayarın ana kartı üzerinde bir yerde farklı chip'lerden veya elektronik elemanlardan oluşur. Aralarındaki iletişimi ise veri yolu (Data bus), adres yolu (Address bus) denilen iletim hatları yapar. Intel, Cyrix, AMD, Motorola mikrolemci üreticilerden birkaçıdır. Günümüzde mikrolemciler genellikle PC adını verdiğimiz kişisel bilgisayarlarda kullanılmaktadır (Altınbaşak, 2005).

1.2. Mikrodenetleyici Nedir?

Bir bilgisayar içerisinde bulunması gereken temel bileşenlerden RAM, I/O ünitesinin tek bir chip içerisinde üretilmiş biçimine mikrodenetleyici (Microcontroller) denir. Bilgisayar teknolojisi gerektiren uygulamalarda kullanılmak üzere tasarlanmış olan mikrodenetleyiciler, mikrolemcilere göre çok daha basit ve ucuzdur. Günümüz mikrodenetleyicileri otomobillerde, kameralarda, cep telefonlarında, fax-modem cihazlarında, fotokopi, radyo, TV, bazı oyuncaklar gibi sayılamayacak kadar pek çok alanda kullanılmaktadır. Günümüz mikrodenetleyicileri birçok chip üreticisi tarafından üretilmektedir. Her firma ürettiği chip'e farklı isimler vermektedir. Örneğin Microchip firması ürettiğine PIC adını verirken, Intel'in ürettiği ve 1980'lerin başında piyasaya sürdüğü 8051, bazen MCS-51 olarak da adlandırılır (Altınbaşak, 2005).

1.2.1. Mikrolemci ve Mikrodenetleyicinin Karşılaştırılması

Mikrolemci ile kontrol edilecek bir sistemi kurmak için en azından bu üniteler bulunmalıdır; CPU, RAM, I/O ve bu ünitelerin arasındaki veri alı-veri ini kurmak için DATA BUS (data yolu) gerekmektedir. Elbette bu üniteleri yerle tirmek için baskılı devreyi de unutmamak gerekmektedir. Mikrodenetleyici ile kontrol edilecek sistemde ise yukarıda saydığımız ünitelerin yerine geçecek tek bir chip (Mikrodenetleyici) ve bir devre kartı kullanmak yetecektir. Tek chip kullanarak elektronik çözümler üretmenin maliyetinin daha düşük olacağı kesindir. Ayrıca da kullanım ve programlama kolaylığı da ikinci bir avantajdır. İşte yukarıda saydığımız nedenlerden dolayı son zamanlarda bilgisayar kontrolü gerektiren elektronik uygulamalarda mikrodenetleyici kullanmaya eğilimin artmasının haklılığını ortaya koyuyor (Altınbaşı, 2005).

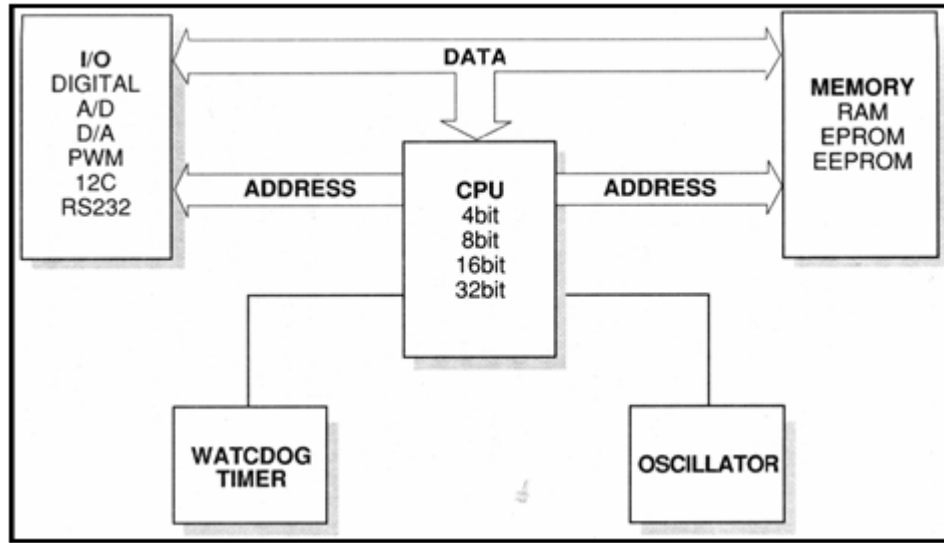
1.2.2. Mikrodenetleyiciler Hakkında Genel Bilgiler

Neredeyse her mikrolemci (CPU) üreticisinin ürettiği birkaç mikrodenetleyicisi bulunmaktadır. Bu denetleyicilerin mimarileri arasında çok küçük farklar olmasına rağmen aynı işlemleri yapabilmektedirler. Her firma ürettiği chip'e bir isim ve özelliklerini birbirinden ayırmak için de parça numarası vermektedir. Örneğin Microchip ürettiklerine PIC adını, parça numarası olarak da 12C508, 16C84, 16F84, 16C711 gibi kodlamalar verir. Intel ise ürettiği mikrodenetleyicilere MCS-51 ailesi adını vermektedir. Genel olarak bu adla anılan mikrodenetleyici ailesinde farklı özellikleri bulunan ürünleri birbirinden ayırt etmek için parça numarası olarak da 8031AH, 8051AH, 8751AHP, 8052AH, 80C51FA gibi kodlamalar kullanılmaktadır (Altınbaşı, 2005).

Bir uygulamaya başlamadan önce hangi firmanın ürünü kullanılacağına, daha sonra da hangi numaralı denetleyicinin kullanılacağına karar vermek gerekir. Bunun için mikrodenetleyici gerektiren uygulamada hangi özelliklerin olması gerektiği önceden bilinmesi gereklidir. Aşağıda bu özellikler sıralanmıştır:

- Programlanabilir dijital paralel giri/çıkışı .
- Programlanabilir analog giri/çıkışı .
- Seri giri/çıkışı (senkron, asenkron ve cihaz denetimi gibi).
- Motor veya servo kontrol için pilsinyali çıkışı .
- Harici giri vasıtasıyla kesme.

- Timer vasıtasıyla ile kesme.
- Harici bellek arabirimi.
- Harici bus arabirimi (PC ISA gibi).
- Dahili bellek tipi seçenekleri (ROM, EPROM, PROM ve EEPROM).
- Dahili RAM seçene i.
- Kayan nokta hesaplaması.



ekil 1.1. Mikrodenetleyici Sistem (Altınba ak, 2005)

Lojik uygulamalarının hızlı olması, fiyatının oldukça ucuz olması, 8 bitlik mikrokontroller olması, bellek ve veri için ayrı yerle ik bus'ların kullanılması, veri ve belle e hızlı olarak erişiminin sağlanması, PIC'e göre diğer mikrokontrolörlerde veri ve programı taşıyan bir tek bus bulunması, dolayısıyla PIC' in bu özelli i ile diğer mikrokontrolörlerden iki kat daha hızlı olması, herhangi bir ek bellek veya giriş/çıkış elemanı gerektirmeden sadece 2 kondansatör ve bir direnç ile çalışabilmeleri, yüksek frekanslarda çalışabilme özelli i, standby durumunda çok düşük akım çekmesi, interrupt kapasitesi ve 14 bit komut i leme hafızasına sahip olması gibi özelliklerinden dolayı PIC mikrodenetleyiciler tercih edilir (Altınba ak, 2005).

1.2.3. PIC Mikrodenetleyicilerin Tercih Sebepleri

Lojik uygulamalarının hızlı olması, fiyatının oldukça ucuz olması, 8 bitlik mikrokontroller olması, bellek ve veri için ayrı yerle ik bus'ların kullanılması, veri ve belle e hızlı olarak erişiminin sağlanması, PIC'e göre diğer mikrokontrolörlerde veri ve programı taşıyan bir tek bus bulunması, dolayısıyla PIC' in bu özelli i ile diğer

mikrokontrolörlerden iki kat daha hızlı olması, herhangi bir ek bellek veya giri /çıkı elemanı gerektirmeden sadece 2 kondansatör ve bir direnç ile çalışabilmeleri, yüksek frekanslarda çalışabilme özelliği, standby durumunda çok düşük akım çekmesi, interrupt kapasitesi ve 14 bit komut işlem hafızasına sahip olması gibi özelliklerinden dolayı PIC mikrodenetleyiciler tercih edilir (Altınbağ, 2005).

1.2.4. PIC Türleri

PIC genel olarak bugüne kadar üç deyimlik seride üretilmiştir. Bunlar 12FXX, 16FXXX ve 18FXXX'dır. 12FXX serisi 8 pin den oluşan, basit işlemleri yapabilen bir seridir. Daha sonra artan işlem karmaşıklığı ve kompleks durumlar için 16FXXX serisi geliştirilmiştir. Örneğin 16F87X serisi yüksek performanslı, full-statik, 8-bit mikrodenetleyicidir. Tüm PIC 16/17 mikrodenetleyiciler RISC mimarisini kullanmaktadır. Ve türlerine göre 18 veya 40 pin olabilirler. Son olarak 18 serisi geliştirilmiştir. Bu seri yüksek data RAM, EEPROM ve FLASH hafızaya olan, Harvard mimarisi kullanılan, toplam 75 komuttan oluşan PIC çeşitlidir. Ayrıca paralel port kontrolü, çarpma bölme sabit ve kayan nokta işlemi yapmak mümkündür. PIC'ler farklı kelime boylarında üretilmelerine rağmen harici veri yolu tüm PIC ailesinde 8-bit'tir (Altınbağ, 2005).

2. ÖNCEK ÇALI MALAR

B. Büchi (1996), sviçre Federal Teknoloji Enstitüsünde yapılan bir çalı mayla labirent içinde hareket eden mobil mini bir robot geli tirilmi tir. Robotun kontrol sistemi PIC16CS4 mikroi lemcisi ile gerçekte tirilmi tir. 2 adet SWATCH tipi motorla ve pille hareketi sa lanmı tır. Labirent içindeki kontrolü sa lamak için genel ve yerel geri besleme kontrolü kullanılmı tır. Genel geri beslemede infrared tabanlı olarak uzaktan kontrol sa lanmı tır. Operatör labirent içindeki bazı noktalarda kontrolü alabilmektedir. Yerel geri beslemede labirent içindeki hareketleri robot iki adet mesafe sensörü kullanarak otomatik olarak yapmaktadır.

S. Baluja, and D.A. Pomerleau (1997), yol üzerindeki eritleri kamera ile izleyerek yönünü bulan bir robot geli tirmi tir. Geli tirilen algoritma yol üzerinde sapmaya neden olan nesnelere olmasında fazladan erit olu turarak yoldan çıkmayı önlemektedir. Sistem sınırlı düzeyde çalı maktaki olup, yapay sinir a ları ile sonraki adımları tahmin etme üzerine kurulmu tur.

W. Li, G. Lu, Y. Wang (1997), Li W. ve arkadaşları çevre artlarında bir aracın otonom olarak kontrol edilebilmesi için beyaz yol eritlerini tanıyan bir metot geli tirmilerdir. Algoritmada bulanık mantık yöntemlerini kullanılmı lardır. Metodu THMR-III ismini verdikleri bir mobil robot üzerinde uygulamı lardır.

Victor J.S (1998), görüntü ile uzaktan kontrol edilen bir robotun kontrolünü gerçekte tirmi tir. Bir veya birkaç kameradan alınan ortamın görüntüleri i lenerek robotun kontrolü kamera kalibrasyonu yapılmadan gerçekte tirilmi tir. Sistem birkaç a amada ele alınmı tır. Bunlar; görsel bilgilerin i lenmesi, robotun ve kapalı devre kontrol sisteminin tasarımı, kamera yeri bilinmedi inde do ruluk analizinin yapılmasıdır.

A. Ollero (1999), Ollero ve arkadaşları yaptıkları çalı mada otonom ta ıtların yön bulma davranı ları üzerine çalı mı lardır. Özellikle net belirgin yollar, hareketli hedefler ve çevre özellikleri üzerine yo unla mı lardır. Bu amaçla birçok pozisyon belirleme teknikleri kullanılmı lardır. Uygulamalarını Romeo adını verdikleri bir robot ta ıt üzerinde gerçekte tirmilerdir.

L.B. Cremean (2006), Cremean L.B. ve 12 arkadaş ı çöl ortamında kendi yolunu bulan ve engellere çarpmadan ilerleyen gerçekte bir araç üzerinde deneme yapmı lardır. Bu çalı mada ise RF (radyo frekansı) sinyalleri ile uzaktan kumanda edilen bir robot araba

üzerine yine kontrolü ve görüntü alımı uzaktan olan bir kamera ba lanarak kontrol i lemi gerçeikle tirilmi tir. Uzaktan gönderilen sinyallerle kamera dü ey ve yatay ekseninde step motor kontrolü ile çevrilmi ve alınan görüntüler gerçek zamanlı olarak bilgisayar ekranından izlenmi tir. Böylece arabanın kontrolü bilgisayar ekranı kullanılarak, do rudan görsel temas kurulmadan sa lanmı tır.

Çayıro lu ve im ir (2008), Bu çalı mada RF (radyo frekansı) sinyalleri kullanarak, kamera monte edilmi tekerlekli bir mobil robot arabanın uzaktan kontrolü gerçeikle tirilmi tir. Uzaktan gönderilen sinyallerle, dü ey ve yatay ekseninde step motor kontrolü ile kamera yönlendirilmi ve alınan görüntüler gerçek zamanlı olarak bilgisayar ekranından izlenmi tir. Böylece arabanın kontrolü do rudan görsel temas kurulmadan, bilgisayar ekranı kullanılarak sa lanmı tır

Çavuo lu ve Kırmızı (2007), Projenin konusu, DC motorlu bir mobil araç tasarımının RF aracılı ıyla bir bilgisayar tarafından e zamanlı kontrolüdür. Araç üzerinde bulunan kamera yardımıyla bilgisayar ba ındaki kullanıcı kumanda etti i aracın hangi yöne gitti ini rahatlıkla görebilmektedir. Ayrıca aracın karanlık ortamlara da girebilece i dü ünülerek araca güçlü bir lazer led monte edilmi tir.

Parlaktuna ve ark. (2006), Bu çalı mada, gezgin robot uygulamalarının geli tirilmesinde kullanılmak amacıyla, Z80 mikroi lemci tabanlı bir platformun tasarımı gerçeklenmi tir. Dört tekerlekli olan gezgin platform adım motorları ile hareket ettirilmektedir. Bu platform üzerine konulan bir kamera ile çizgi takibi uygulaması yapılmı tır.

3.MATERİYAL ve METOT

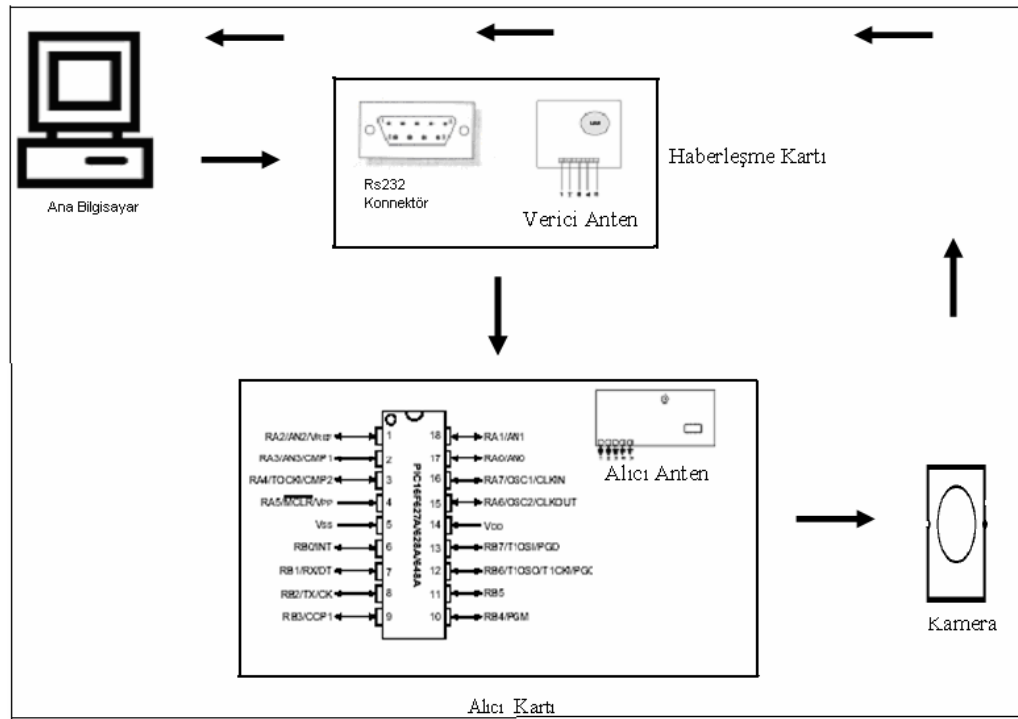
3.1.Materyal

3.1.1. Sistemin Genel Yapısı ve Çalışma Prensipleri

Bu bölümde sistemin temel yapısını gösteren blok diyagramı verilerek sistemin nasıl çalıştığı üzerinde durulacak ve ana bloklar hakkında bilgi verilerek bu blokların birbirleriyle nasıl iletişim kurdukları anlatılacaktır.

3.1.1.1. Sistemin Blok Diyagramı

Sistemin temel blok diyagramı ekil 3.1'deki gibidir.



ekil 3.1. Sistemin Blok Diyagramı (Çavuşoğlu ve Kırmızı, 2007)

ekil 3.1'de görüldüğü gibi, sistemde temel olarak üç kısımdan oluşmaktadır. Bilgisayarın seri portundan aldığı verileri karttaki alıcı antene aktaracak olan haberleşme kartı ile birlikte bilgisayardan aldığı veriler doğrudan mikrodenetleyici sayesinde üzerindeki cihazları kontrol eden bir alıcı kartı sistemde bulunmaktadır. Alıcı kartı üzerinde alıcı anten, mikrodenetleyici ve diğer cihazlar bulunmaktadır.

Bilgisayarda mobil aracı kontrol etmek için ara yüz programı bulunmaktadır. Ayrıca kameranın kendine ait bir RF alıcı-verici modül çifti olduğundan, sistemden alınan görüntü mikrodenetleyici üzerinden geçmeden doğrudan bilgisayara bağlı olan alıcıya

iletilmektedir. Kameradan alınan görüntü ara yüz programı sayesinde bilgisayarda izlenmektedir.

3.1.1.2. Haberle me Kartı

Haberle me kartının sistem içindeki görevi, kullanıcı ara yüz programından gelen verileri bilgisayarın seri portu yardımıyla alarak, üzerinden bulunan verici anten yardımıyla araç üzerinde bulunan alıcı antene iletmektir. Kullanıcı ile araç arasında bir köprü görevi görmektedir.

3.1.1.3. Alıcı Kartı

Bu kart üzerinde bulunan mikrodenetleyici sayesinde haberle me kartından alınan veriler kontrol edilmekte ve mobil aracın hareket etmesi sağlanmaktadır. Bilgisayardan gelen verilere göre mobil araç, ileri, geri, sağa veya sola hareket edebilmekte ve mobil araç üzerinde bulunan kamera step motor yardımıyla hareket ettirilerek sağa veya sola döndürülebilmektedir. Bu kullanıcıya daha geniş bir görüş alanı sağlamaktadır.

3.1.1.4. Kullanıcı Ara Yüz Programı

Kullanıcıların aracı istedikleri şekilde yönetebilmeleri için kullanımı kolay bir ara yüz programı hazırlanmıştır. Delphi 10.0 ile yapılan bu ara yüz programı ileri düzeyde bilgisayar bilgisi olmayan en temel kullanıcıları da hedeflediğinden son derece sade bir şekilde hazırlanmıştır. Ara yüz programı sayesinde kameradan alınan görüntüler bilgisayar ortamında izlenmektedir.

3.1.1.5. Sistemin Çalışması

Mobil araca veri göndermek için bilgisayarın seri portu kullanılmaktadır. Seri port RS-232 standartlarında ve asenkron olarak çalıştığından sinyal seviyeleri ve sinyal gönderim şekli TTL mantığına benzememektedir. Seri port çıkışından alınan bu verilerin mikrodenetleyicinin anlayabileceği TTL seviyelerine çevrilmesi için haberle me kartı üzerinde MAX232 sinyal dönüştürücüsü bulunmaktadır. MAX232 çıkışından alınan lojik seviyedeki veriler verici anten ile alıcı kartına yollandıktan sonra haberle me kartının görevi bitmektedir, hem alıcı hem de verici antenler RF iletişim şeklini kullanmaktadır.

Alıcı kartı üzerinde bulunan alıcı anten verileri alarak mikrodenetleyiciye iletir. Veri haberleşmesi seri port üzerinden RS-232 standartlarında yapıldığından asenkrondur ve bu nedenle yollanan komutlar 8-bitlik gruplar halinde bit bit gelmektedir. Mikrodenetleyici seçilirken seri iletişim birimi (USART) olan PIC18F1320 kullanılmıştır. Gelen bitler

mikrodenetleyicinin ilgili yazmacında tutulduktan sonra, PIC18F1320'e yüklenmiş olan yazılımla kullanıcı ara yüzünden gelen komutlar mikrodenetleyici sayesinde uygulamaya konulur.

3.1.2. Verici Kartında Kullanılan Entegreler ve Özellikleri

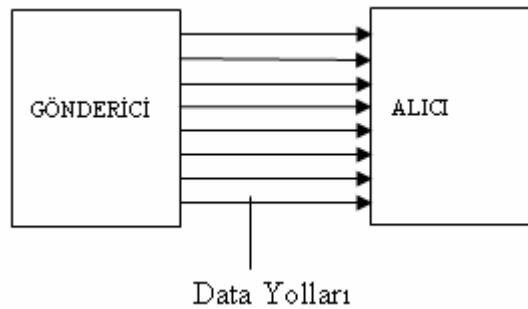
Verici kartı bilgisayarda yüklü bulunan kullanıcı ara yüz programı ile araç üzerinde bulunan alıcı kartı arasında RF sayesinde haberleşmeyi sağlamaktadır. Verici kartında bulunan elemanlar ve bu elemanların özellikleri bu bölümde anlatılacaktır.

Çizelge 3.1 Verici Kartında Bulunan Elemanlar

Verici Kartı Eleman Listesi	Adet
Seri port kablosu	1
RS232 konnektor	1
16 bacaklı konnektor	1
MAX232	1
ATX-34	1
1 uF Kondansatör	5
7805 Regülatörü	1

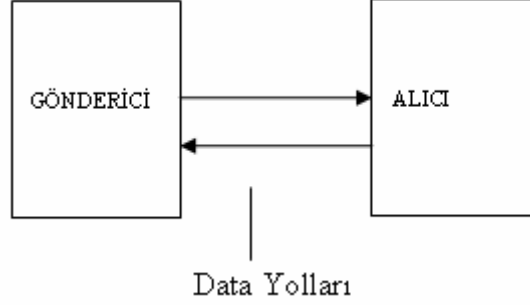
3.1.2.1. Seri Port ve RS232 Standartları

Bilgisayarlar ile iletişim kurabilmek için bir takım yöntemler vardır. Seri haberleşme, paralel haberleşme gibi. Paralel iletişimde her bir bilgi için bir data hattı vardır. Yani 8 bit'lik bir veri gönderimi için 8 tane data hattına ihtiyaç vardır (ekil 3.2). Paralel haberleşmede tüm bit'ler aynı anda data yollarından gönderilir. Bu nedenle paralel iletişim seri iletişime göre daha hızlıdır. Fakat paralel iletişim ile uzun mesafelere veri iletmek, hem uygun değildir, hem de pahalıdır (Çiçek, 2007).



ekil 3.2. Paralel iletişim

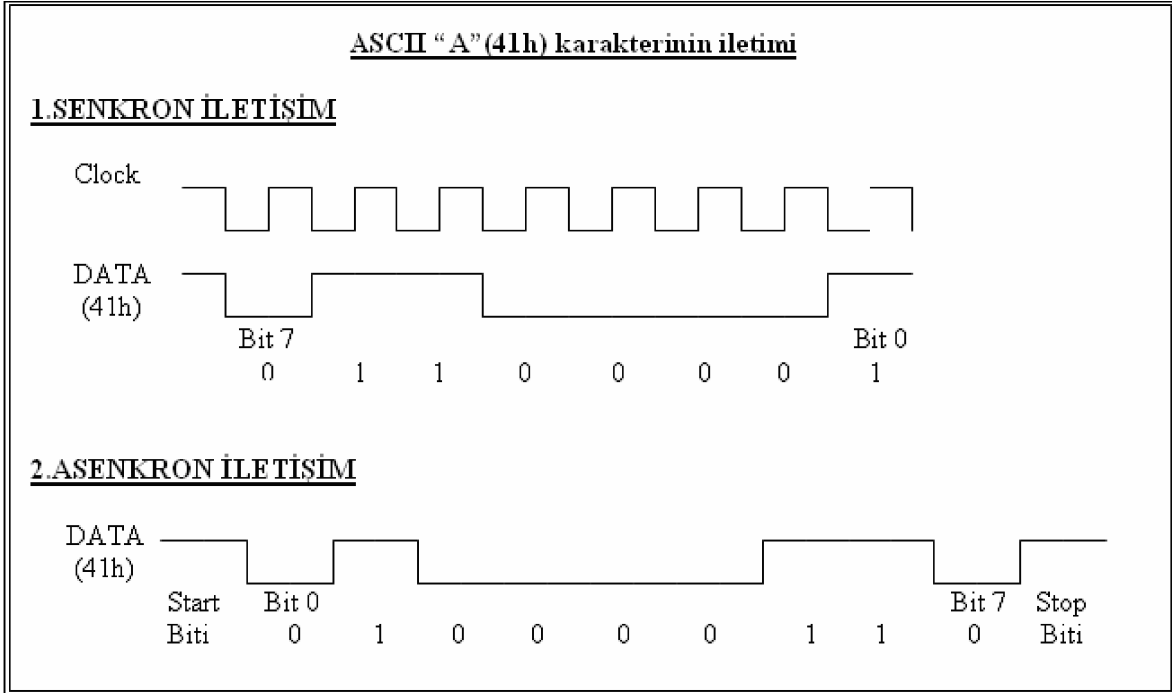
Bu sebeple seri ileti im daha yava olmasına ra men, hem daha ucuz hem de uzun mesafelere iletim için uygundur. Seri haberle mede bilgiler tek bir kanal üzerinden gönderilip, tek bir kanal üzerinden alınır.



ekil 3.3. Seri leti im

Seri port üzerinde bitler yani “lojik 1” ya da “lojik 0” de erleri 8-bit olarak tek bir kablo aracılı ı ile pe i sıra iletilirler. Seri port, lojik de erleri -3 V ile + 25 V arasında iletebilir. Yani seri port 50 V maksimum voltaj de i im aralı ına sahiptir. Bunun sonucu olarak da seri portta olu an kayıp önemli de ildir. Seri ileti imde az tel kullanılır ve cihaz ile bilgisayar arasındaki 3 telli kablo seri ileti im için yeterlidir. Seri haberle mede, gönderici kısımda 8-bit veri, paralelden seriye çevrilir ve daha sonra tek bir hattan kar ıya gönderilir. Alıcı, seri veriyi paralele çevirerek 8-bit veriyi olu turur. Bir linkteki veri akı ının kontrolü için gerekli sinyallerden biri saat (clock) sinyalidir. Hem gönderici hem de alıcı cihazda, bir bitin ne zaman gönderilece ine veya alınaca ına karar verilirken bir saat sinyali kullanılır. Veri gönderen ve alan uçların belli kurallar çerçevesinde haberle mesi gerekir. Verinin nasıl paketlenece i, bir karakterdeki bit sayısını, verinin ne zaman ba layıp bitece i gibi bilgileri bu kurallar belirler. Bu kurallar çerçevesine, protokol adı verilir (Çavuo lu ve Kırmızı, 2007).

ekil 3.4’de gösterildi i gibi seri portlarda 2 çe itiletim vardır. Senkron ve asenkron iletimdir. Her biri saatleri farklı ekilde kullanır.



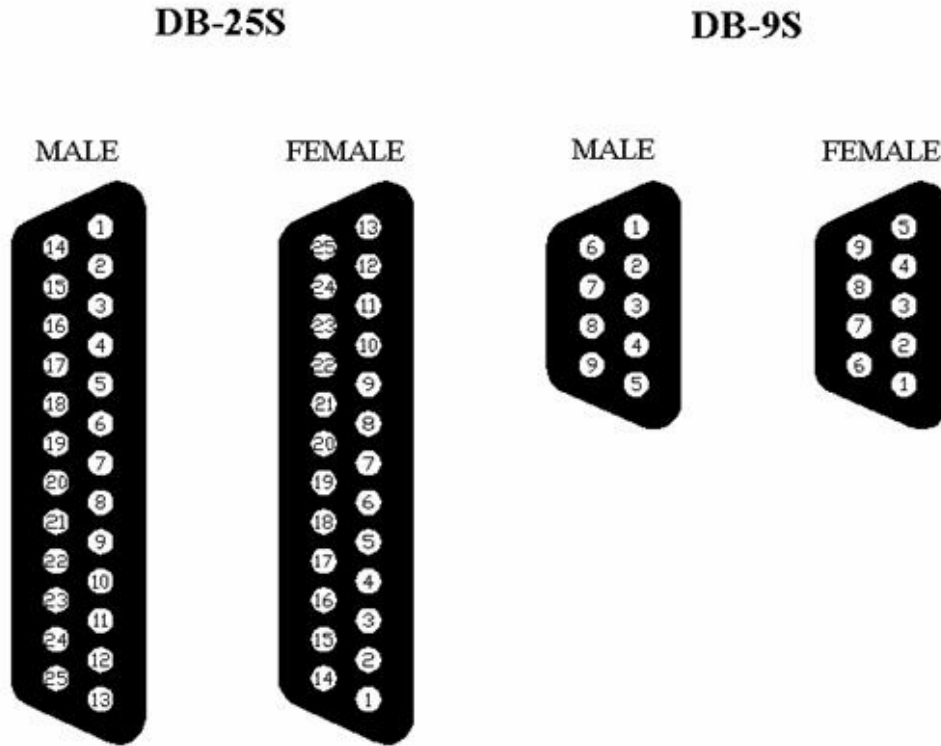
ekil 3.4. Senkron ve Asenkron İletim (Çavuşoğlu ve Kırmızı, 2007)

Senkron seri iletişimde gönderilen veri bit'leri ile alınan veri bit'leri arasında bir uyum olmalıdır. Yani senkron ifadesi, alıcı devre ile verici devrenin eş zamanlı çalışması anlamına gelmektedir. Bu alıcı-verici arasındaki uyumu sağlamak için data kablosu yanında birde clock sinyali kablosu bulunur. Böylece bilgi iletişimi için start (başlangıç) ve stop (bit) bit'lerine gerek kalmaz. Senkron seri iletişim asenkron seri iletişime göre daha hızlıdır. Ancak senkron iletişim karmaşık ve pahalı devreler içerir.

Asenkron seri iletişimde ise sadece veri hattı bulunur. Alıcı ve verici devreler veri iletişiminde eş zamanlı çalışmazlar. Senkron iletişimdeki clock palsi yerine asenkron iletişimde start bit'i, stop bit'i ve parity bit'i vardır. Start (başlangıç) bit'i, bilginin gönderilmeye başlandığını alıcı tarafa bildirmek için kullanılır. Asenkron iletişimin ilk bit'i her zaman start bit'idir. Stop(bit) bit'i, gönderilen bilginin bittiğini alıcı tarafa belirtir. Stop bit'inden sonra gönderilen bilgi yeniden start bit'i ile başlar. Parity (eşlik) bit'i, alıcı tarafa gönderilen karakterin doğru olarak iletilip iletilmediğini kontrol etmek için kullanılan bit'tir. Asenkron veri iletişiminde start bit'i her zaman lojik-0, stop bit'i ise her zaman lojik-1 olmalıdır (Çiçek, 2007).

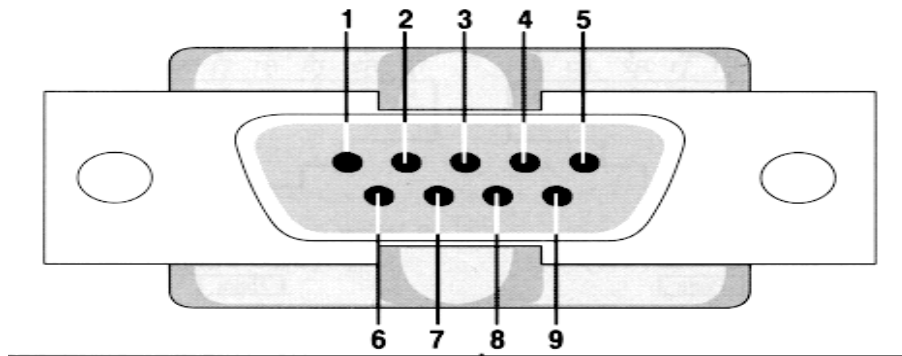
Seri iletişim hızı, saniyede gönderilen bit sayısı olarak (bps-bits per second) tanımlanır. Bu orana seri iletişimin hızını belirtmek için kullanılan baud oranları, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 olarak çeşitli değerlerde dir.

Bilgisayarlarda asenkron seri ileti im için 9 ve 25 uçlu RS-232 port'ları vardır. ekil 3.5'de 9 ve 25 pin'li RS-232 di i ve erkek seri portlar görülmektedir.



ekil 3.5. 9 ve 25 Pin'li RS232 Di i ve Erkek Seri Portlar

Günümüzde daha çok 9 pin'li seri port kullanılmaktadır. RS-232 port'unun pin'lerinin görüntüsü ekil 3.6'da, pin'lerin açıklamaları da Çizelge 3.2'de verilmi tir.



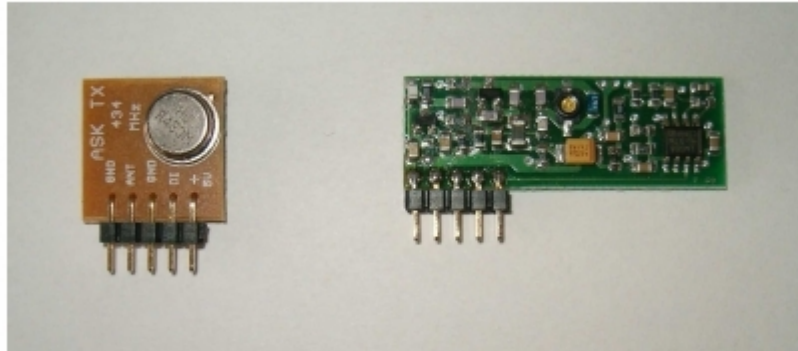
ekil 3.6. RS-232 9 Pin'li Di i Seri Port'un Fiziksel Görünü ü ve Yapısı

Çizelge 3.2. 9 Pin’li RS–232 Port’unun Pin’lerinin Açıklamaları

Pin No	Sembol	Açıklama
1	CD	Veri Ta ıyıcı Algılayıcı (Data Carrier Detect)
2	RD-RXD	Veri Alma (Receive Data)
3	TD-TXD	Veri Gönderme (Transmit Data)
4	DTR	Veri Terminal Hazır (Data Terminal Ready)
5	GND	Toprak (Signal Ground)
6	DSR	Veri Hazır (Data Set Ready)
7	RTS	Gönderme iste i (Request To Send)
8	CTS	Göndermek için Sil (Clear To Send)
9	RI	Halka Gösterici (Ring ndicator)

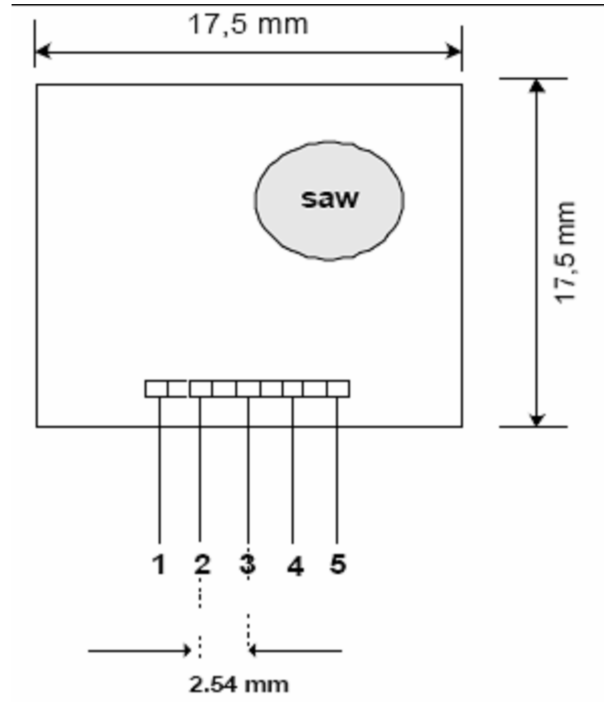
3.1.2.2. ATX–34 Verici Anten Entegresi

Seri port ile araç kablosuz haberleşme için, haberleşmenin kesintisiz olabilmesi için RF iletim im ekli kullanılmaktadır. Kullanılan alıcı-verici modül ise Udea Elektronik A. .'nin üretti i ARX–34 ve ATX–34 entegreleridir. Bu entegreler ekil 3.7’de gösterilmiştir.



ekil 3.7. ATX–34 ve ARX–34 Entegreleri

ATX–34 entegresinin görevi, MAX232 entegresinden gelen sayısal veriyi, araç üzerinde bulunan ARX–34 alıcısına bir anten vasıtası ile yollamaktır. RF yaklaşık 433Mhz’de ve UHF bandında çalışmaktadır. Kısa mesafeli uzaktan kontrol uygulamaları için düşük fiyatı nedeniyle ideal bir çözümdür. ATX-34’ün genel görünümü ile uç ayrıntıları ve elektriksel özellikleri öyledir. ekil 3.8’de ATX–34 entegresinin genel görünümü verilmiştir (Anonim, 2004).



ekil 3.8. ATX-34 Genel Görünü

Besleme voltajı: Modüller içerisinde bir voltaj regülatörü bulunmamaktadır. Tasarım daha çok pil ile kullanıma uygundur. ARX-34 modülü 4,9 – 5,1 V regüle edilmiş bir voltaj kaynağına, ATX-34 ise 5–12 V arasında regüle edilmiş bir voltaj kaynağına ihtiyaç duymaktadır. Modüller belirtilen devrelerin altında bir besleme yapıldığında kararsız çalışacaktır. Besleme voltajı ve topraklama GND bağlantısı belirtilen devrelerin üzerinde veya ters olursa, modül devre kalıcı hasarlara uğrayabilir. Pin sırası ve özellikleri Çizelge 3.3’de verilmiştir. (Anonim, 2004).

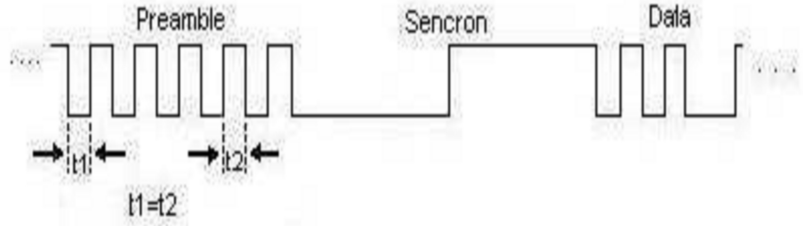
Çizelge 3.3. ATX-34’un Uç Ayrıntıları (Anonim, 2004)

Pin No	Pin Adı	I/O	Açıklama
1	GND	-	Kontrol kartınızı toprak hattına bağlayınız
2	ANT	O	50 Ohm empedans anten bağlantı noktası
3	GND	-	Kontrol kartınızı toprak hattına bağlayınız
4	DIN	I	DIGITAL INPUT
5	Vcc	-	+5V DC besleme terminali

Data formatı: ATX-34 modülün de, dijital data giri i için DIN pini bulunur. DIN pini RF ile gönderilecek sinyallerin kullanıcı tarafından verildi i giri tir. Standart data protokolü su ekildedir. ekil 3.9'da gösterilmi tir (Anonim, 2004).

TX : preamble + sencron + data1+.....+dataX

En basit haberle me sistemlerinde bile mesajın ba langıcı için bir preamble kullanılması neredeyse zorunludur. Preamble veri olarak ardı ık 1 ve 0' lar dan olu an (01010101...) bir bit dizinidir. 5 byte 0x55 veya 0xAA olabilir. Gönderilen 1 ve 0'ların süreleri e it olmalıdır. Kısaca preamble donanım senkronizasyonunu sa lamaktadır. Sencron ise yazılımın senkronizasyonuna yardımcı olur. Bit senkronizasyonunun sa lanması ve mesaj ba langıcının do ru tayini için kullanılması gereklidir. Bu bit dizininin boyu uygulama gereksinimleri veya kısıtlamalarına göre de i ebilmekle birlikte 5 byte 0x00 + 5 byte 0xFF olabilir veya bunun ne olaca na ki i kendisi karar verebilir. Data gönderirken araya bo luk girmemeli, girer ise tekrar preamble ve sencron gönderilmelidir (Anonim, 2004).



ekil 3.9. Data Formatı (Anonim, 2004)

Anten: Verimli data transferi ve alımı için gerekli en önemli iki nokta iyi bir anten ve do ru RF topraklama seçilmesidir. Anten olmadan data'nın uzun mesafelere gönderilmesi mümkün de ildir. Modüller basit bir anten ba lantı pinine sahiptir. Uygun bir UHF anten do rudan bu pine ba lanabilir. En basit anten 17.3cm uzunlu undaki bir kablonun anten giri ine lehimlenmesi ile yapılabilir. En iyi ileti im mesafesi her iki taraftaki antenlerin birbirini görmesi ile elde edilebilir. Herhangi bir obje veya metal bir engel ileti im mesafesini dü ürecektir. Ayrıca sinyal göndermeleri, gönderilen sinyallerin metal yüzeylerden, binalardan vb. gelen yansımalarından etkilenirler. Bu yanlı data alımlarına yol açabilir (Anonim, 2004).

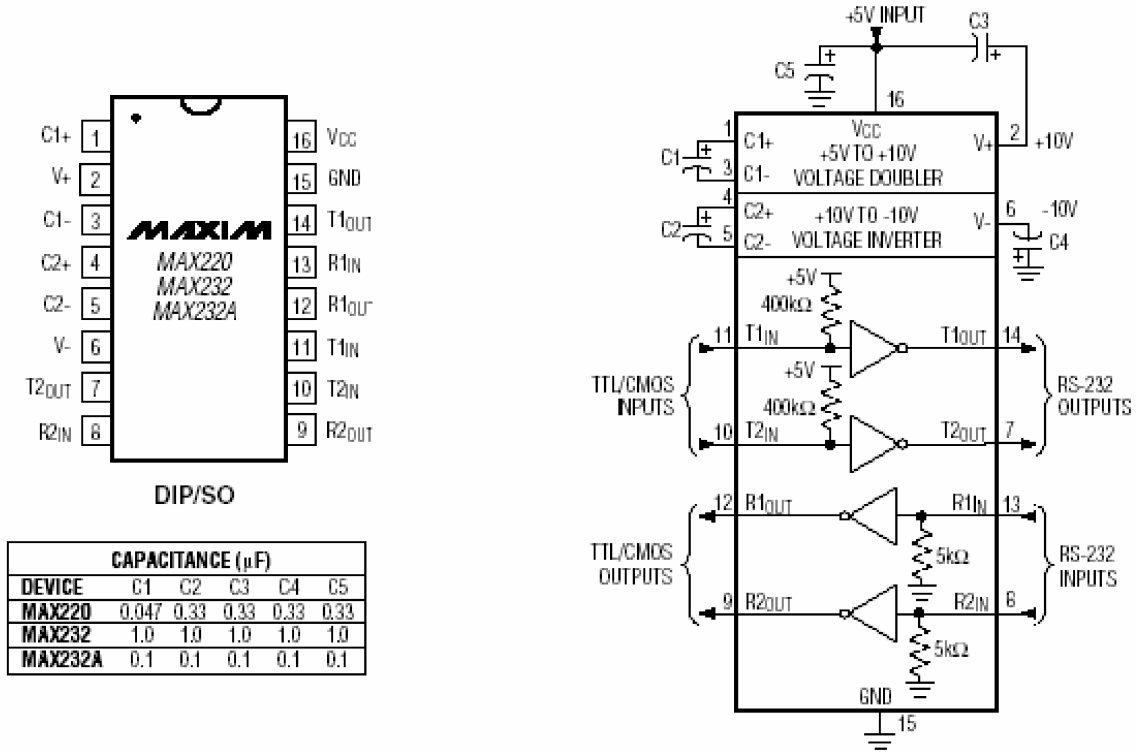
ATX-34 entegresinin elektriksel özellikleri Çizelge 3.4'de verilmi tir.

Çizelge 3.4. ATX-34'un Elektriksel Özellikleri (Anonim, 2004)

	Min	Typ.	Max	Unit	Not
Frequency		433.920		Mhz	
Bandwidth		±2		Mhz	
Data Rate	0.3		2.4	Kbit/s	
RF Sensitivity		-108		dBm	
Voltage supply	4.9		5.1	Vcc	Use regulated voltage source Max Riple 100mV
Supply current		5		mA	
Logic "0" DOUT voltaj	0		0.1*Vcc	Vdc	
Logic "1" DOUT voltaj	0.8*Vcc		Vcc	Vdc	Max. 5 mA current source
RX on time		10		Ms	
Working Temperature	-10		+55	°C	ETSI 300 220

3.1.2.3. MAX232

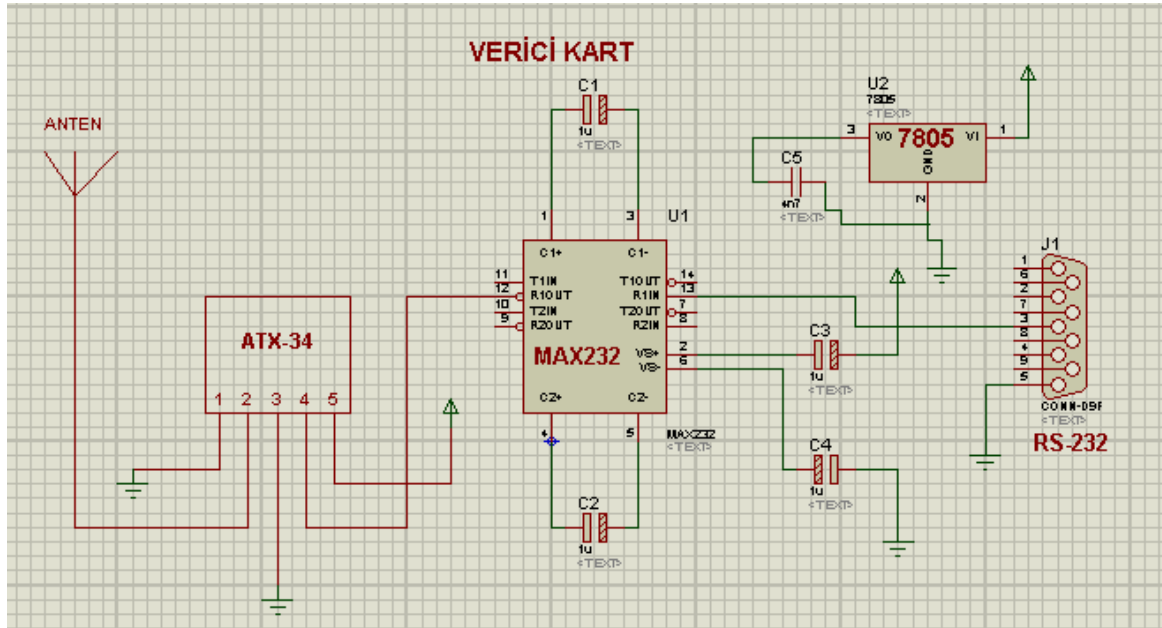
Bilgisayar seri portlarının kullandığı RS232 sinyallerinin elektriksel özelliği, 5 Volt "lojik 1" ve 0 Volt "lojik 0" de erine karşılık gelen standart TTL mantığına benzememektedir. RS232'de -12 Volt "lojik 1", +12 Volt "lojik 0" de erine karşılık gelir. Bu özellik daha uzun mesafeli kablolarda, güçlendirme gerektirmeden iletilim sağlar. RS232 ile birlikte çalışan birçok devrede bu voltaj farkını dengeleyici ek yongalar kullanılır. Bunlardan en yaygın olanı MAXIM firmasının ürettiği MAX232 sürücüsüdür. MAX232 gerilim seviye dönüştürücüsüdür. RS232 +- 12 V kullandıktan MAX232 bunu mikroileminin çalıştığı 5 V ya da 3,3 V'a dönüştürerek uyumu sağlar. Bu entegrenin bacak bağlantıları ve iç yapısı ekil 3.10'da gösterilmiştir (Çavuşoğlu ve Kırmızı, 2007).



ekil 3.10. MAX232 Entegresinin Bacak Ba lantıları ve ç Yapısı (Anonim, 2004)

3.1.2.4. Verici Kartının Ba lantı ekli

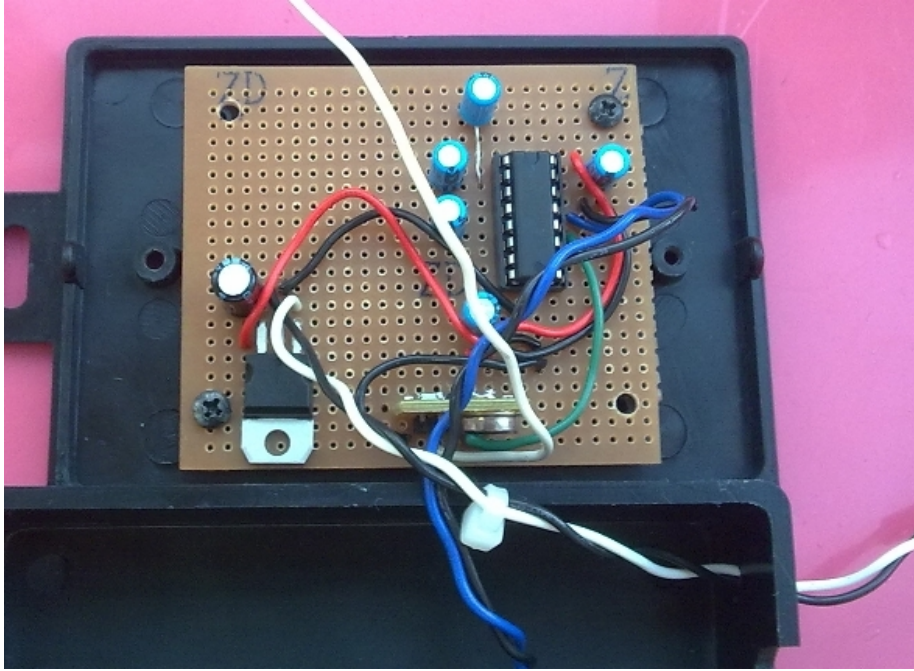
Verici kartın Proteus'da çizilmi ba lantıları ekil 3.11'de gösterilmi tir.



ekil 3.11. Verici Kart Devre eması

3.1.2.5. Verici Kartının Genel Görünü ü

Verici kartı devresinin genel görünü ü ekil 3.12’de gösterilmi tir.



ekil 3.12. Verici Kartının Genel Görünü ü

3.1.3. Araçta Kullanılan Entegreler Ve Özellikleri

Alıcı kartı, verici kartından alınan verilerin mikrodenetleyiciye iletildi i ve mobil aracın hareketini sa layan devredir. Alıcı kartı üzerindeki elemanların listesi Çizelge 3.5’de verilmi tir. Bu elemanlar tek tek açıklanacaktır.

Çizelge 3.5. Alıcı Kartı Eleman Listesi

Alıcı Kartı Elemanları	Adet
PIC 18F1320	1
Röle	5
ARX-34	1
Step Motor ve ULN2003	1
DC Motor	2
7805	1
Kamera	1
Güç Kayna 1	2
Transistör	5

3.1.3.1. PIC18F1320 Mimarisi Ve Özellikleri

Mikrodenetleyiciler her yıl hızla geliştirilmekte ve onlarca yeni PIC piyasaya sürülmektedir. Her üretilen PIC daha hızlı, daha ucuz ve daha fazla i levi yerine getirecek biçimde düzenlenmektedir. Bu PIC' ler arasında PIC18F1320' de bulunmaktadır. 16 bit'lik komut seti, yüksek çalışma hızı, A/D çevrim, PWM, zamanlayıcılar, 32 kesmeye imkan vermesi, CAN arabirimi, seri ve paralel haberleşme protokolleri ile bu aile Microchip PIC serisi içinde en güçlü mikrodenetleyici ailesidir (ekil 3.13) (ahin, Dayanık ve Altınbağ, 2006).

Features	PIC18F1220	PIC18F1320
Operating Frequency	DC – 40 MHz	DC – 40 MHz
Program Memory (Bytes)	4096	8192
Program Memory (Instructions)	2048	4096
Data Memory (Bytes)	256	256
Data EEPROM Memory (Bytes)	256	256
Interrupt Sources	15	15
I/O Ports	Ports A, B	Ports A, B
Timers	4	4
Enhanced Capture/Compare/PWM Modules	1	1
Serial Communications	Enhanced USART	Enhanced USART
10-bit Analog-to-Digital Module	7 input channels	7 input channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes
Programmable Brown-out Reset	Yes	Yes
Instruction Set	75 Instructions	75 Instructions
Packages	18-pin SDIP 18-pin SOIC 20-pin SSOP 28-pin QFN	18-pin SDIP 18-pin SOIC 20-pin SSOP 28-pin QFN

ekil 3.13. PIC18F1320'nın Özellikleri (Anonim, 2005)

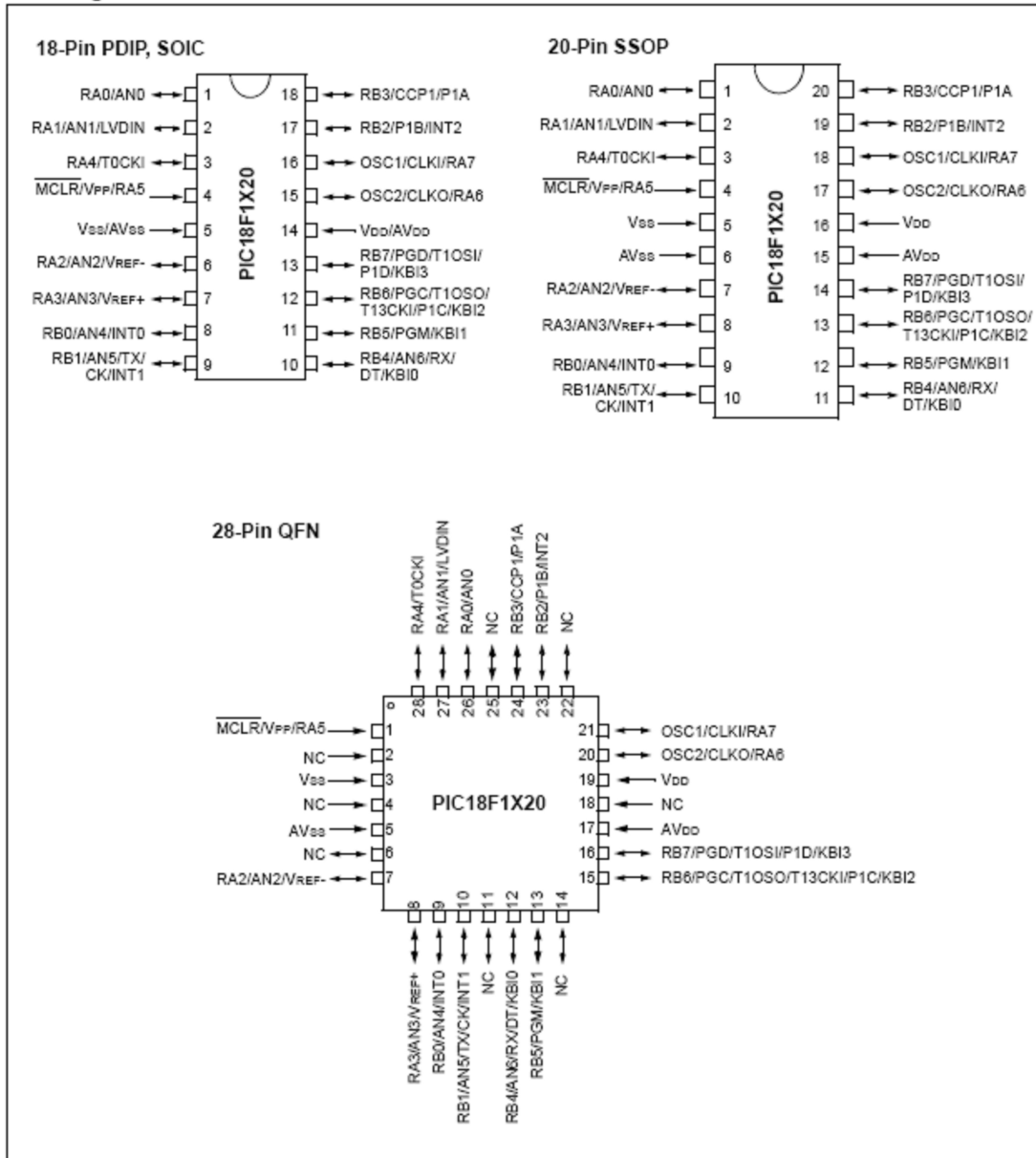
3.1.3.1.1. PIC18F1320' nin Pin Diyagramı

Microchip 18F1320'i üç farklı kılıf tipinde üretmektedir. Kullanımı en kolay olan PDIP tipi, piyasada en kolay bulunanıdır. Bunun dışında 20 bacaklı SSOP kılıflı ve 28 bacaklı QFN kılıflı tipleri de mevcuttur. ekil 3.14'de PIC18F1320'nin dış görünüşü ve ekil 3.15'de kılıf tipleri gösterilmektedir.



ekil 3.14. PIC18F1320'nin Dış Görünüşü

Pin Diagrams



ekil 3.15. PIC18F1320'nin Pin Diagramı (Anonim, 2005)

3.1.3.1.2. PIC18F1320'nin  Yapısı

ekil 3.16. PIC18F1320'nin  Yapısı (Anonim, 2005)

3.1.3.1.3. Osilatör Uçları ve Çe itleri

PIC belle inde bulunan program konutlarının çalı tırılması için bir kare dalga sinyaline gereksinim vardır. Bu sinyale saat (clock) sinyali denir. Saat sinyalinin hızı komutların i leni hızını da arttırır.PIC18F1320'nin saat sinyalinin giri için kullanılan iki ucu vardır. Bunlar OSC1 (16.pin) ve OSC2 (15.pin) uçlarıdır. Bu uçlara farklı tipte osilatörlerden elde edilen saat sinyalleri uygulanabilir. Saat osilatörü tipleri Çizelge 3.6'da gösterilmi tir (Anonim, 2005).

Çizelge 3.6. PIC18F1320'nin Osilatör Çe itleri (Anonim, 2005)

LP	Low-Power Crystal
XT	Crystal/Resonator
HS	High-Speed Crystal/Resonator
HSPLL	High-Speed Crystal/Resonator with PLL enabled
RC	External Resistor/Capacitor with FOSC/4 output on RA6
RCIO	External Resistor/Capacitor with I/O on RA6
INTIO1	Internal Oscillator with FOSC/4 output on RA6 and I/O onRA7
INTIO2	Internal Oscillator with I/O on RA6 and RA7
EC	External Clock with FOSC/4 output
ECIO	External Clock with I/O on RA6

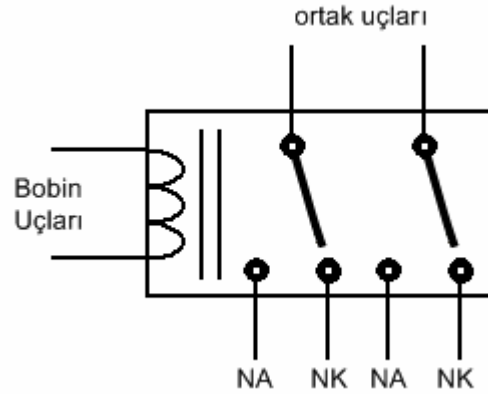
3.1.3.1.4. Besleme Gerilimi

PIC'in çalı ması için gerekli olan gerilim 5 ve 14 numaralı pin'lerden uygulanır.14 numaralı Vdd ucu +5V'a, 5 numaralı Vss ucu da topra a ba lanır. PIC'e gerilim uygulandı nda devredeki fazla akım çeken elemanların devreye girmesiyle yâda ebekedeki gerilim dalgalanmaları nedeniyle PIC'in resetlenmesini önlemek amacıyla Vdd ile Vss arasına 0.1uF'lik bir dekuplaj kondansatörü ba lamak gerekir. Bu durumda karalı gerilim elde edilmi olur (Anonim, 2005).

3.1.3.2. Röle ile DC Motor Sürmek

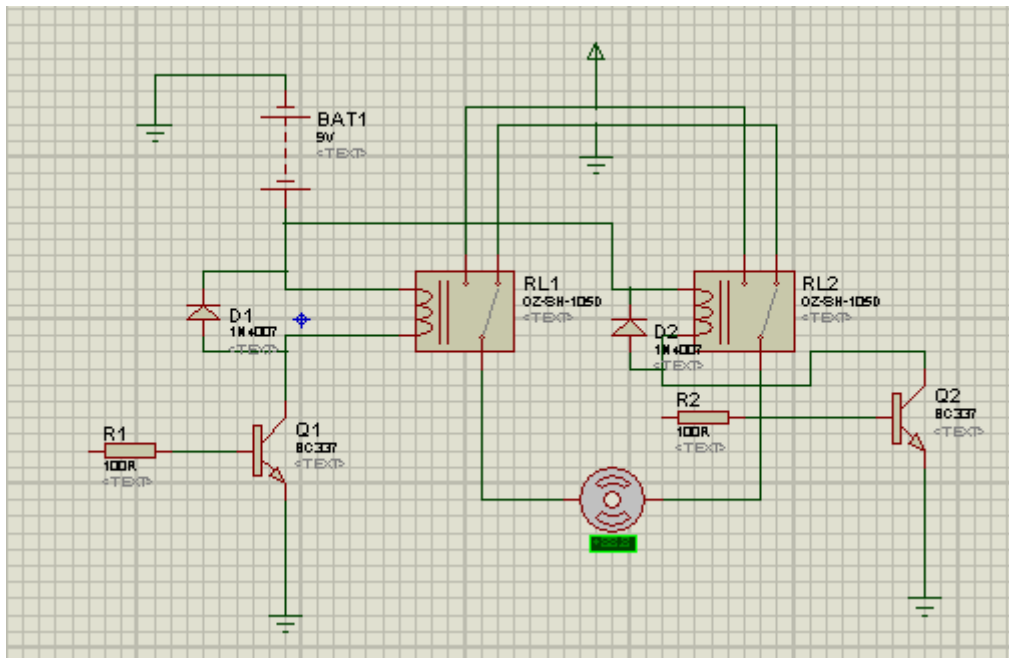
Röleler küçük de erli bir akımın olu turdu u elektromanyetik alan ile yüksek güçte veya akımda çalı an bir alıcıyı (yükü) çalı tırabilmek (anahtarlayabilmek) için bir ya da daha fazla anahtar grubunu kontrol eden (anahtarları açan ya da kapatan) elemanlardır (ekil 3.17). Röleler, elektromıknatis, palet ve kontaklar olmak üzere üç kısımdan olu ur.

Rölenin bobinine bir gerilim uygulandı ında, röle enerjilenir ve paletini çeker. Böylece kontaklar kapanır ve yük kontrol edilmi olur (Anonim, 2007).



ekil 3.17. Çift Kontaklı Röle

Bu projede mobil aracın DC motorlarının sa , sol, ileri ve geri hareketleri için her biri için birer röle kullanılmı tır. Ayrıca kamerayı açıp kapamak içinde bir tane röle kullanılmı tır. ekil 3.28'de rölenin DC motora ba lantı ekli gösterilmi tir.



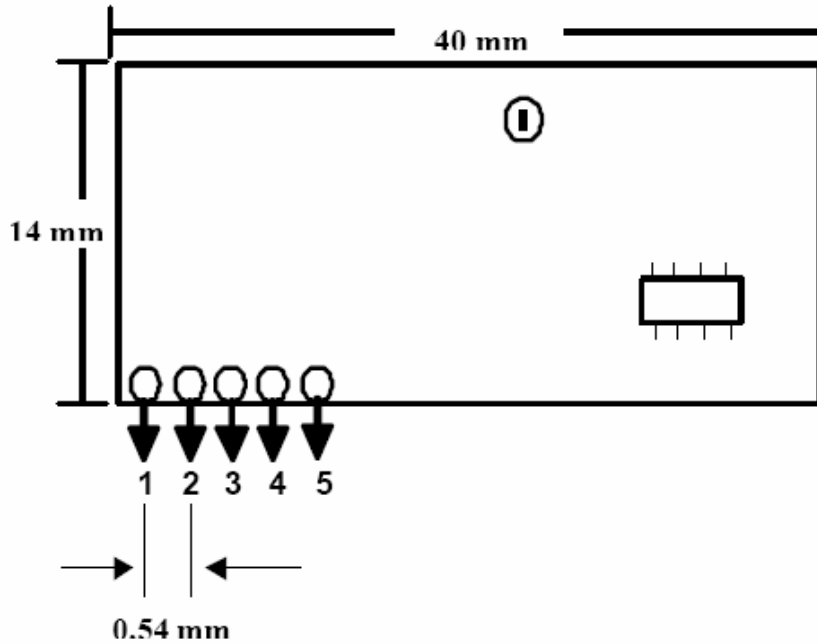
ekil 3.18. Röle ile DC Motor Sürücü Devresi

ekil 3.18'de verilen devrede, 5 volt ile çalı an röleler kullanılmı tır. 5 Voltluk akım rölenin ayaklarından birine gidecektir. Akım, rölenin di er baca ından topra a ula amadı ı için röle aktif hale geçmeyecektir. Transistör data pininden gelen +5Volt

sayesinde tetiklenecek ve kollektör-emitör (CE) ilettime geçecektir. Burada, transistör, tıpkı bir anahtar gibi kullanılmı tır. Dolayısı ile rölenin ucu topra a ula abilecek ve aktif hale geçecektir. Rölenin bobinine bir gerilim uygulandı ından, röle enerjilenir ve paletini çeker. Böylece röle konumunu de i tirerek devrenin tamamlanmasını sa lamaktadır. Diyotun kullanılmasının sebebi ise transistör kesime girdi i anda bobinde bir zıt emk olu ur ve önlem alınmazsa transistöre zarar verir. Röle bobinine ters paralel ba lanan bir diyot ile bu sorun kolayca a ılır.

3.1.3.3. ARX-34 Alıcı Anten Entegresi

ARX-34 entegresinin görevi, haberle me kartında bulunan ATX-34 entegresinden gelen sayısal veriyi alarak mikrodenetleyicinin seri giri ucuna vermektir. RF yakla ık 433Mhz'de ve UHF bandında çalı maktadır. Kısa mesafeli uzaktan kontrol uygulamaları için dü ük fiyatı nedeniyle ideal bir çözümdür. ARX-34'ün genel görünü ü ekil 3.19'da gösterilmektedir (Anonim, 2004).



ekil 3.19. ARX-34'nin Genel Görünü ü

ARX-34'ün uç ayrıntıları Çizelge 3.7'de gösterilmiştir.

Çizelge 3.7. ARX-34'ün Uç Ayrıntıları

Pin No	Pin Adı	I/O	Tanımlama
1	ANT	I	Anten
2	GND	-	Toprak
3	VCC	-	+5 Besleme
4	AOUT	O	Analog Çıkı
5	DOUT	O	Dijital Çıkı

ARX-34'nün besleme voltajı: ARX-34 içerisinde bir voltaj regülatörü bulunmamaktadır. Tasarım pil kullanımı düşünülerek yapılmıştır. Bu nedenle besleme voltajında belirtilen değerlere dikkat edilmelidir. Modül belirtilen değerlerin altında bir besleme yapıldığında kararsız çalışacaktır. Besleme voltajı +5 VDC ve topraklama GND bağlantısı belirtilen değerlerin üzerinde veya ters olursa, modülde kalıcı tahribatlara yol açılabilir. Düşük akım sarfiyatlarının sağlanabilmesi için modül içerisine ters polarizasyondan koruyacak bir devre konulmamıştır. Besleme voltajında çalışma sürecinde ± 100 mV değişimlerin üzerindeki değişimler modülün kararsız çalışmasına neden olur. Besleme devresinde regülatör IC kullanılması önerilir (Anonim, 2004).

Anten: Verimli data transferi ve alımı için gerekli en önemli iki nokta iyi bir anten ve doğru RF topraklama seçilmesidir. Anten olmadan datanın uzun mesafelere gönderilmesi mümkün değildir. Modül basit bir anten bağlantı pinine sahiptir. Uygun bir UHF anten olduğundan bu pine bağlanabilir. ARX-34 modülüne bağlanabilecek en basit anten 17.3cm uzunluğundaki bir kablonun anten girişine lehimlenmesidir. Anteni, modülden uzak bir yere bağlanması gerekiyorsa 50 Ohm Coax anten kablosu kullanılması gerekmektedir. Anten kablosunun topraklanması, modülün anten girişine yakın bir yerden yapılmalıdır (Anonim, 2004).

3.1.3.4. RF Sinyalleri ile İletim Kurulan Kamera

Yapılan bu çalışmada, RF sinyalleri ile haberleşerek görüntü alıp gönderen, güvenlik kamerası sistemi kullanılmaktadır (Şekil 3.20.). Kablosuz kamera alıcısı radyo dalgalarıyla haberleştiğinden, alıcısı aynen TV alıcıları gibi kameradan görüntü almaktadır. Alıcı üzerinde bulunan tuner ile uygun frekansın bulunması gerekir. Ayrıca bu alıcı bilgisayara TV kartı aracılığıyla bilgisayara bağlanır. Bu güvenlik kamerasından alınan görüntüler geliştirilen yazılım sayesinde bilgisayar ekranına aktarılmaktadır. Kablosuz mikro kamera 9

volt pil ya da 220 VAC- 9 VDC adaptörü ile çalı tırılmaktadır. Alıcı ünite ise 12 VDC adaptörü ile çalı tırılmaktadır. Güvenlik kamerası 25-50 metre görüntü aktarımına sahiptir.



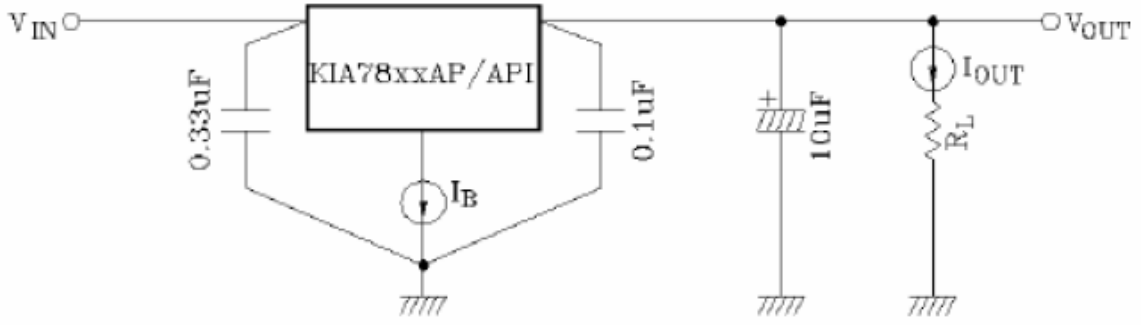
ekil 3.20. Kablosuz Kamera

3.1.3.5. Güç Kayna ı

Mobil aracda iki farklı güç kayna ı kullanılmaktadır. Mikrodenetleyiciyi ve mikro kamerayı kontrol etmek için 12 V'luk akü kullanılmaktadır. DC motorları sürmek için 4.8 V'luk pil kullanılmaktadır.

3.1.3.6. 7805 Regülâtörü

Güç kayna ının kullanılması ve düzenlenmesi sistemin önemli sorunlarından biridir. +12 V'luk bir akü ile bu sorun önemli ölçüde çözülmü tür. Ancak mikrodenetleyici ve di er entegrelerin besleme gerilimi +5 V oldu undan + 12 V'luk gerilim 7805 regülâtör entgesi ile + 5 V'a çevrilmektedir.7805'in blok yapısı ekil 3.21'de verilmi tir.

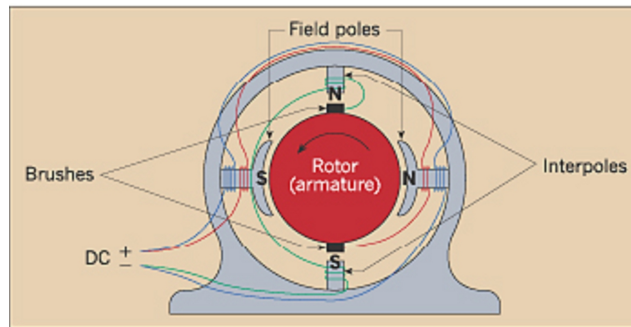


ekil 3.21. 7805'in Blok Yapısı

3.1.3.7. DC Motorlar ve Aracın Hareket ekli

Mobil aracın hareketi, üzerindeki dört adet tekerlek ile sa lanmaktadır. Arka tekerleklere ba lı bir DC motor ile ileri ve geri hareket sa lanmaktadır. Ön tekerleklere di li sistem ba lanmı tır. Mobil robotun sa a ve sola hareketleri ise ön tekerleklere ba lı di li sistemi kontrol eden DC motor tarafından sa lanmaktadır. Mobil aracın DC motoru 4.8 V'luk pille çalı maktadır.

DC motor: Elektrik enerjisini, mekanik enerjiye dönü türen elemanlara “motor” denir. Elektrik motorları AC ve DC olmak üzere sınıflandırılırlar. DC motorlar, çalı ması ve kullanımı bakımından en kolay ö renilebilecek motorlardır. Bir mıknatısın N ve S kutupları arasında bir bobin yerle tirilip, bobinden akım geçirilirse iletken manyetik alanın dı na itilir ve bunun sonucunda bir hareket meydana gelir (ekil 3.22). Basit bir dc motor, bu prensibe göre çalı ır.



ekil 3.22. Fırçalı DC Motor

Bu tip projelerde en sık tercih edilen motor tipi DC motorlardır (ekil 3.23). DC motorlar ucuz, küçük ve etkilidir. Ayrıca boyut, ekil ve güç bakımından çok çe itli

olmaları da DC motorların sık kullanılmalarının bir di er sebebidir. DC motorların kullanımına dair temel özellikler a a ıda açıklanmı tır.



ekil 3.23. DC Motor Dı Görünü ü

Yön: DC motorlara bir güç kayna ı ba landı ında DC motorun dönü yönü akımın yönüne ba lıdır. Akımın yönü terslendi inde DC motorun dönü yönü de terslenmi olur.

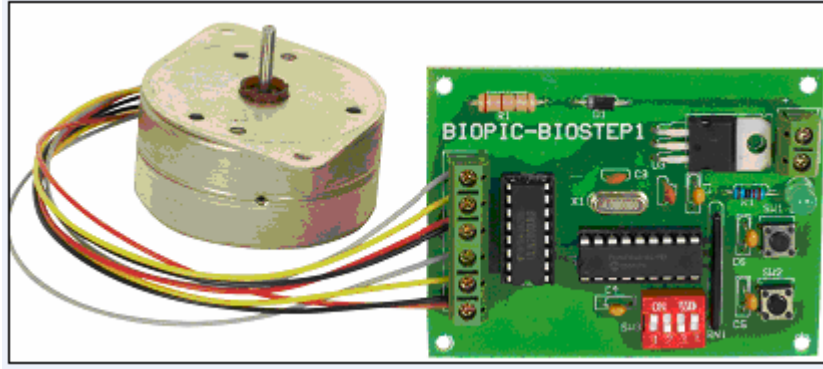
Hız: Bir motorun hızı rpm (rotations per minute - bir dakikada tamamlanan devir sayısı) ile ölçülür. Motorun hızı voltaja ve yüke ba lıdır. Bir DC motorun hızının voltaja ve yüke göre de i imini de erlendirmek için iki durum dü ünülebilir. Bunlardan ilki; DC motora yük binmeyen ya da sabit bir yükün oldu u bir sistemdir. Böyle bir sistemde DC motorun hızı uygulanan voltaja ba lıdır ve voltaj arttıkça hız da artar. kinci durum ise; DC motora binen yükün zamana ya da gerçekte tirilen göreve göre de i ti i bir sistemdir. Bu durumda DC motorun hızı yüke ba lı olacaktır. Yük arttıkça uygulanan güç de artar ve güç arttıkça hız azalır.

Voltaj: Küçük DC motorlar 1,5 V ile 48 V arasında de i en voltaj de erlerine sahip olarak bulunabilirler. Her bir DC motor için belirtilen voltaj de eri, o DC motorun kendi verilen hız, güç ve akım de erlerinde stabil çalı tı ı voltaj de eridir. Robotlarda ve di er sistemlerde DC motorları kullanırken de bu voltaj de eri, DC motora verilecek maksimum çalı ma voltajını belirledi i için önemlidir.

Akım: Bir DC motor belirtilen voltaj de erinde çalı tırıldı ında DC motorun çekece i akım yüke ba lıdır. Yük arttıkça DC motorun çekti i akım da artar. DC motor, maksimum akım sınırının a ılaca ı fazla bir yükte çalı tılmamalıdır. Böyle bir durumda DC motor kısa devreye neden olur ve uygulanan güç ısıya dönü ür. Bu durum uzun sürerse DC motor yanabilir. Genellikle DC motorların uygulama akımı aralı ı 50 mA den ba layıp 2A üzerine kadar çıkabilir.

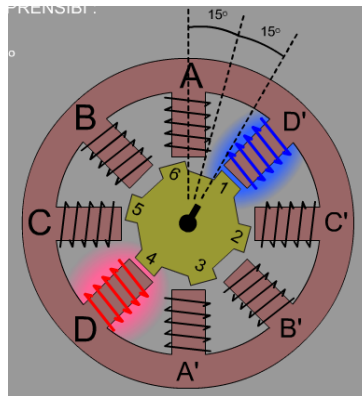
3.1.3.8. Step Motor ve ULN2003 Sürücüsü

Bu çalı mada güvenlik kamerasını sa a sola hareket ettirmek için Bipolar (çift kutuplu) step (adım) motor ve sürücü entegre devresi kullanılmı tır (ekil 3.24). BIOSTEP devresi PIC 16f628 ile yapılmı bir step motor kontrol devresidir. PIC'lerin çıkı akımı step motoru sürmeye yetmedi i için bir adet ULN2003 kullanılmı tır.



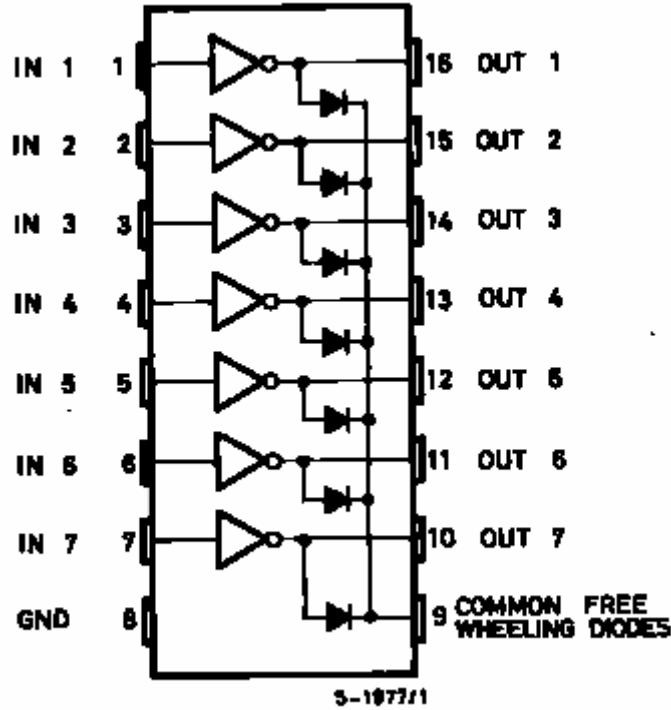
ekil 3.24. Step Motor Kontrol Devresi

Bipolar step motor; bobinlerdeki akımı, ters yöne çeviren torca neden olan bir uyartım meydana getirir. Step motorlar ald ı sayısal darbelere ba lı olarak belirli bir açıda dönme hareketi yapan motorlardır. sminden de anla ılaca ı üzere, sadece tekil dönme adımları atabilir (ekil 3.25). Bu motorlar lojik "0", lojik "1" prensibine göre çalı ırlar. Bu özelli inden dolayı kullanım alanı oldukça geni tir. Motora uygulanacak sinyallerin frekansı de i tirilerek motorun hızı kontrol edilebilir. Step motorlarda stator sargılarına uygulanan sinyallerin sırası de i tirilerek dönü yönü de i tirilir. Step motorlarının hangi yöne dönece i, devir sayısı, dönü hızı gibi de erler mikrolemci veya bilgisayar yardımıyla kontrol edilmektedir.

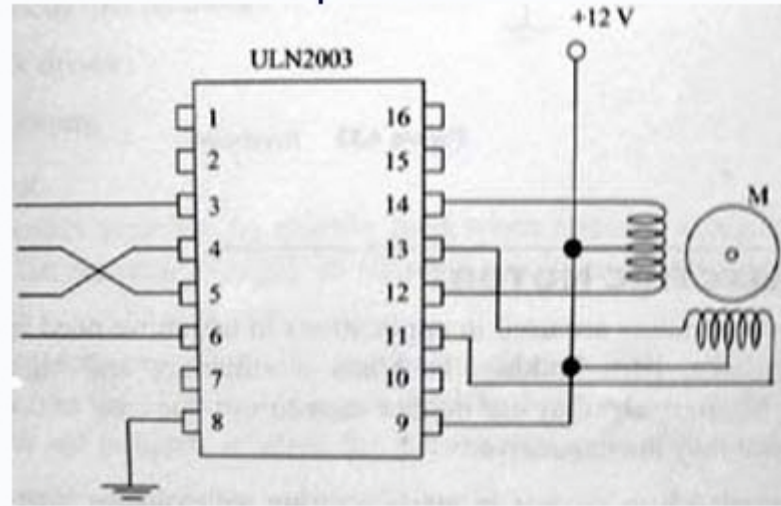


ekil 3.25. Step Motor Çalı ma ekli

Adım motor sürücüsü olarak ULN2003 entegresini kullanılmı tır. Sürücü devresi olarak kullanılan ULN2003, içerisinde 7 adet NPN transistor ve dahili diyot barındırmaktadır. ekil 3.26'da görüldü ü gibi 8 numaralı baca a toprak, 9 numaralı baca a ise +9V uygulanır. Ayrıca giri bacaklarına +5V uygulandı ında kar ısındaki bacakları lojik 0 yapar. Bir adım motor sürülmek istendi inde ULN2003'ün giri pinlerine belirli aralıklarla sırasıyla lojik 1 de eri gönderilir. ULN2003'ün çıkı pinleri ise adım motora ba lanır.ULN2003 entegresinin giri ine sırasıyla lojik 1 uygulandı ından, çıkı larını sırasıyla lojik 0 yapar ve böylece adım motor sargıları sırasıyla topra a ba lanır. Adım motorun ortak ucu +9V'a ba landı ından adım motor adım adım dönmeye ba lar. ULN2003'ün step motora ba lantı ekli ekil 3.27'de gösterilmi tir (Çelik, 2005).

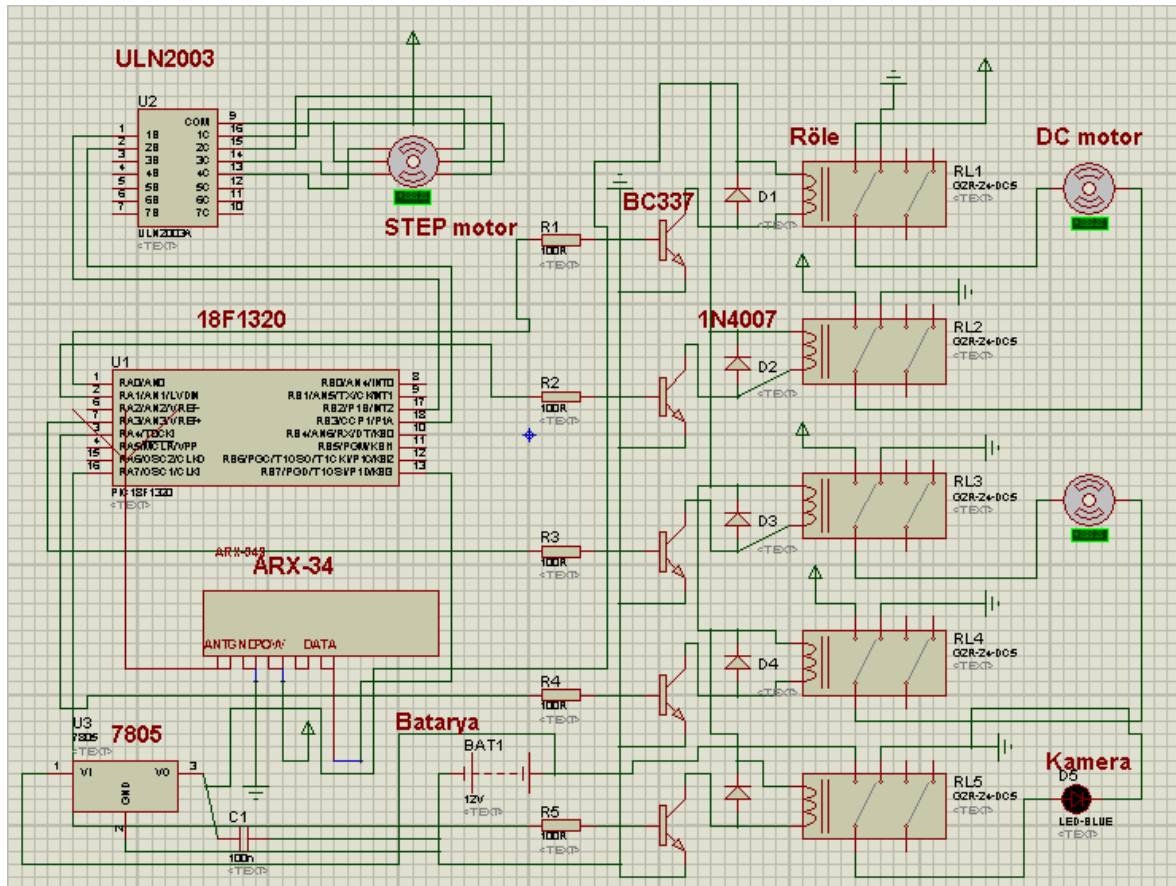


ekil 3.26. ULN2003 Pin Diyagramı

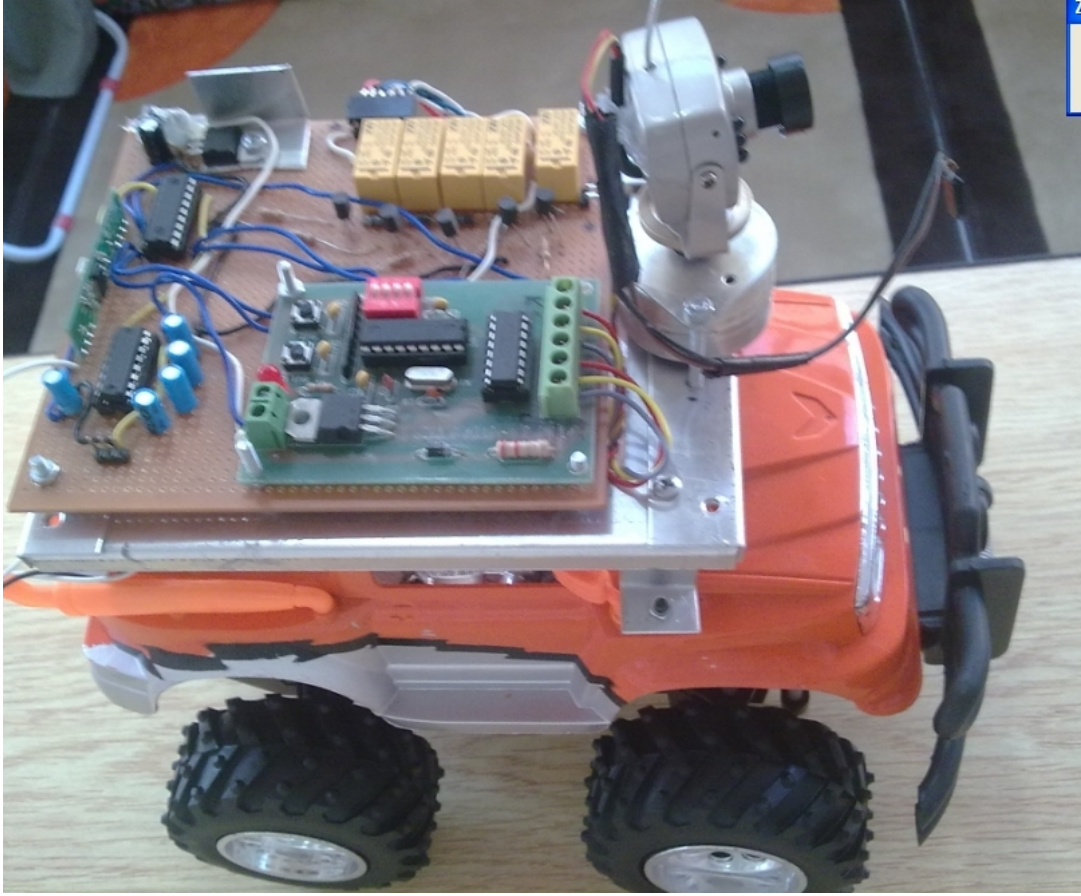


ekil 3.27. ULN2003 ile Step Motor Kontrolü

3.1.3.9. Alıcı Kartının Devre eması



ekil 3.28. Alıcı Kartın Devre eması

3.1.3.10. Aracın Genel Görünü ü

ekil 3.29. Mobil Aracın Genel Görünü ü

3.2. Metot**3.2.1. Mikrodenetleyici Yazılımı**

Tezde kullanılan mikrodenetleyici Microchip firmasının üretti i PIC18F1320'dir. PIC18F1320, PIC-C programlama dili ile programlanmıştır. PIC-C, üst düzey programlama dili olan C'nin, PIC için düzenlenmiş sürümüdür. PIC bu program sayesinde çok kolay ve hızlı biçimde istenilen komutları yerine getirebilmektedir. Bu tezde kullanılmasının sebebi kolay yazılımıdır. Mikrodenetleyiciye yüklenen PIC-C program kodlarının tamamı Ek 1 'de verilmiştir.

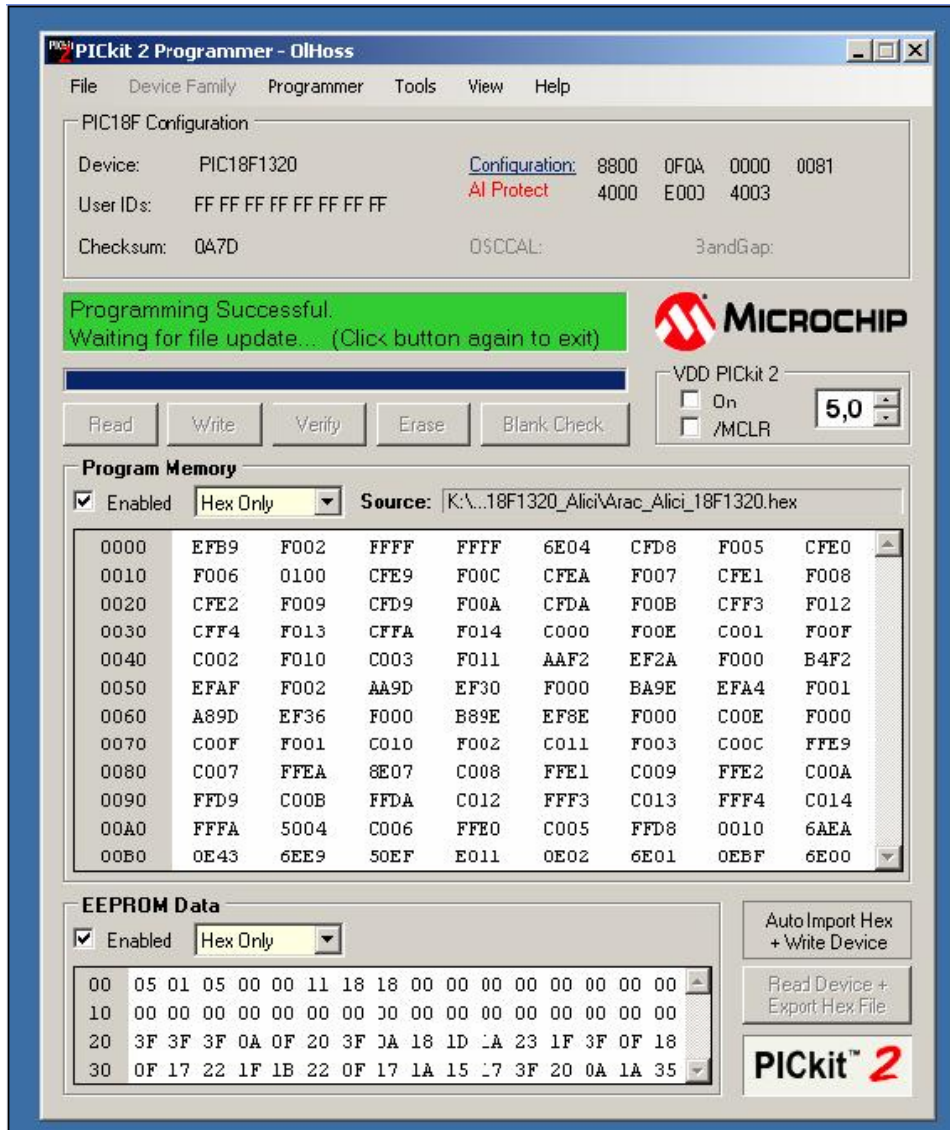
3.2.1.1. Mikrodenetleyiciye Program Kodunun Yüklenmesi

PIC C 'de program derlendikten sonra meydana gelen HEX dosyasını mikrodenetleyiciye yüklemek için microchip firmasının kendi üretti i PIC Kıt_2 usb programlayıcısı kullanılmıştır. ekil 3.30'da PIC Kıt_2 yükleme kartı gösterilmektedir.

ekil 3.31'da PIC Kit_2 programı ile HEX dosyasının mikrodenetleyiciye yüklenmesi gösterilmektedir.



ekil 3.30. PIC Kit_2 Yükleme Kartı

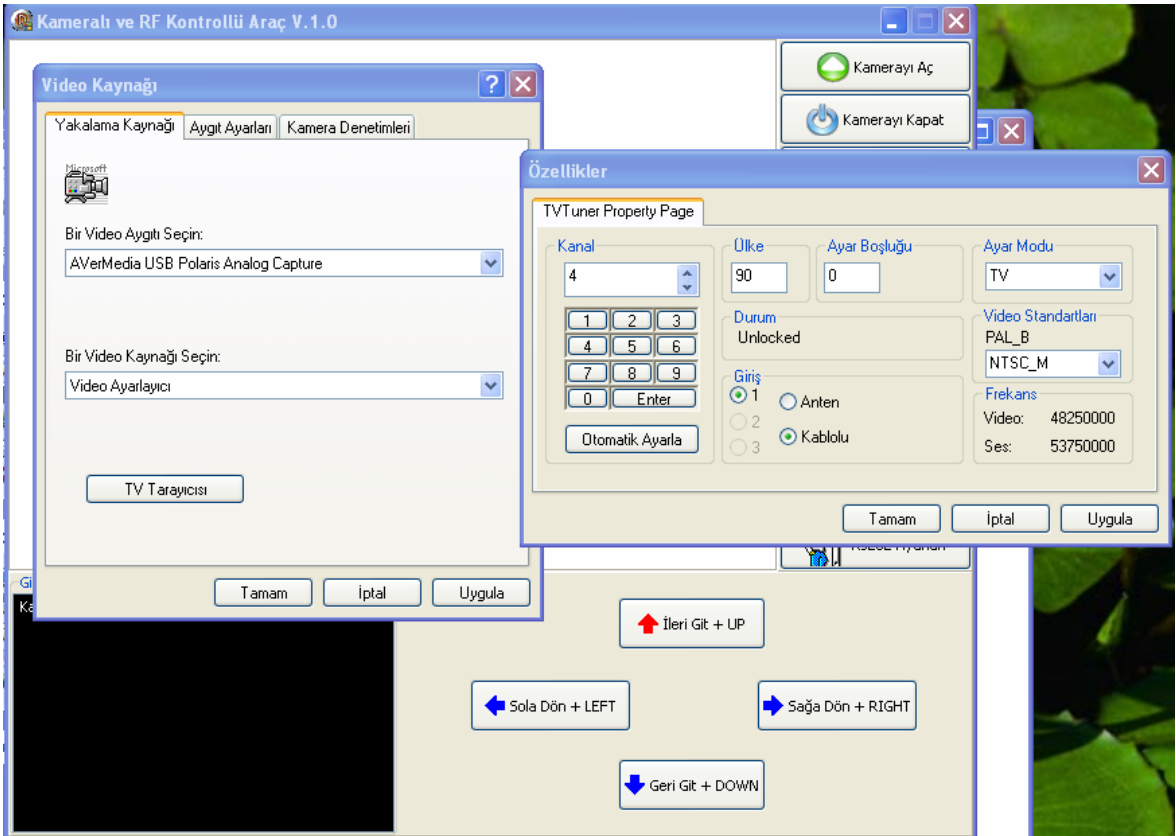


ekil 3.31. HEX Dosyasının Mikrodenetleyiciye Yüklenmesi.

3.2.2. Kullanıcı Ara Yüz Programı

Bilgisayardan kontrolleri yapabilmek için Delphi 10.0 yazılımı kullanılarak bir arayüz programı hazırlanmıştır. Delphi kodları Ek 2'de verilmiştir. Program görüntü izleme ve mobil aracı kontrol amaçlı olarak kullanılmaktadır. Mobil aracı kontrol amaçlı olarak aracı ileri, geri, sağ ve sola hareket ettirmek. Ayrıca kamerayı da sağ ve sola çevirmektir. Görüntü ile ilgili olarak da fotoğraf çekmek ve görüntü izlemektir.

Kablosuz kamera alıcısı radyo dalgalarıyla haberleştiğinden, alıcısı aynen TV alıcısı gibi kameradan görüntü almaktadır. Bu alıcı bilgisayara TV kartı aracılığıyla bağlanır. TV kartının video girişine bağlanan alıcı, aldığı bilgileri bu sayede bilgisayara aktarabilir (ekil 3.32).



ekil 3.32. TV Kartı Ayarları

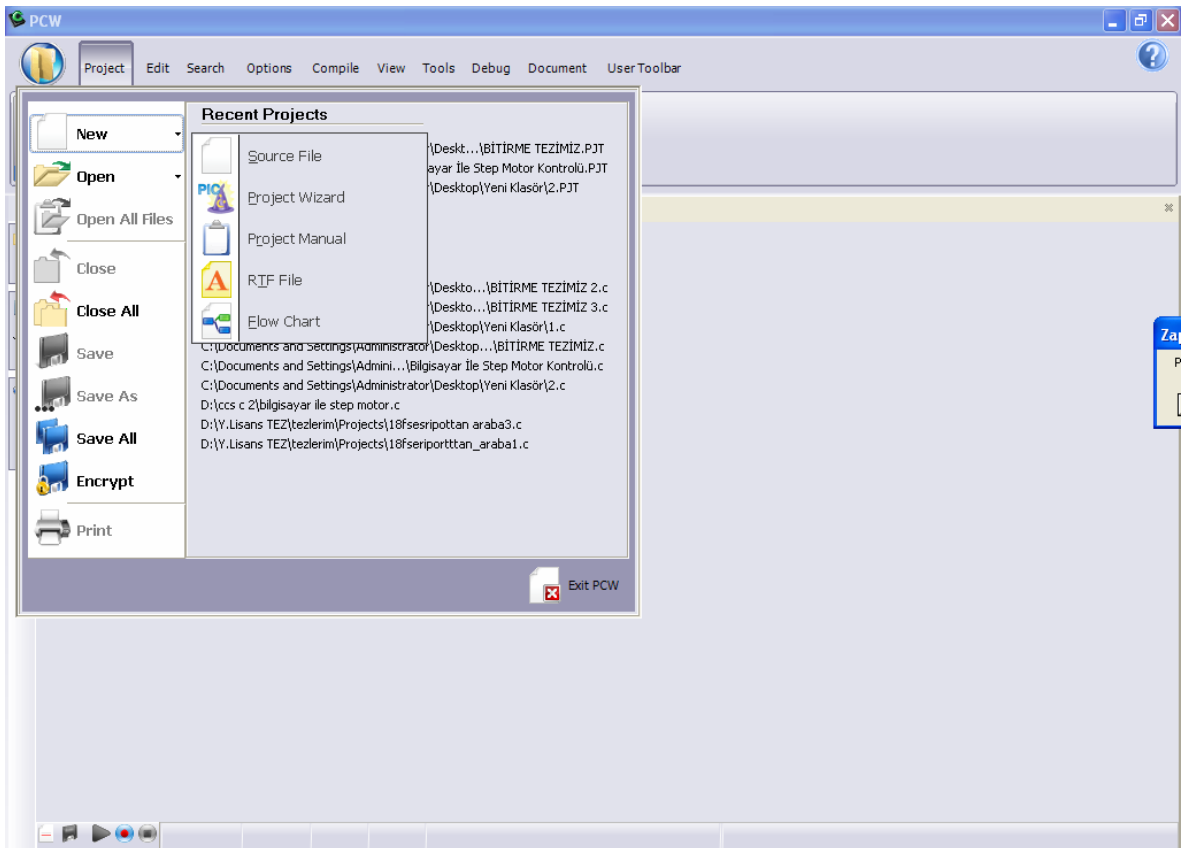
4.BULGULAR ve TARTI MA

4.1. Uygulamanın CCS C Programı Kısmı

CCS C derleyici programı Custom Costumer Services Inc firmasının bir ürünüdür. CCS firması PIC mikrodenetleyicilerinin C dilinde programlamasını sa layan programlar geli tiren bir firmadır. CCS C derleyicisi üç de i ik derleyiciden olu maktadır bunlar; PCB, PCM ve PCH' dır. PCB 12 bit'lik mikrodenetleyiciler için, PCM 14 bit'lik mikrodenetleyiciler ve PCH ise 16 bit'lik mikrodenetleyiciler için kullanılmaktadır. Bu projede PCH 16 bit'lik derleyicisi kullanılmaktadır.

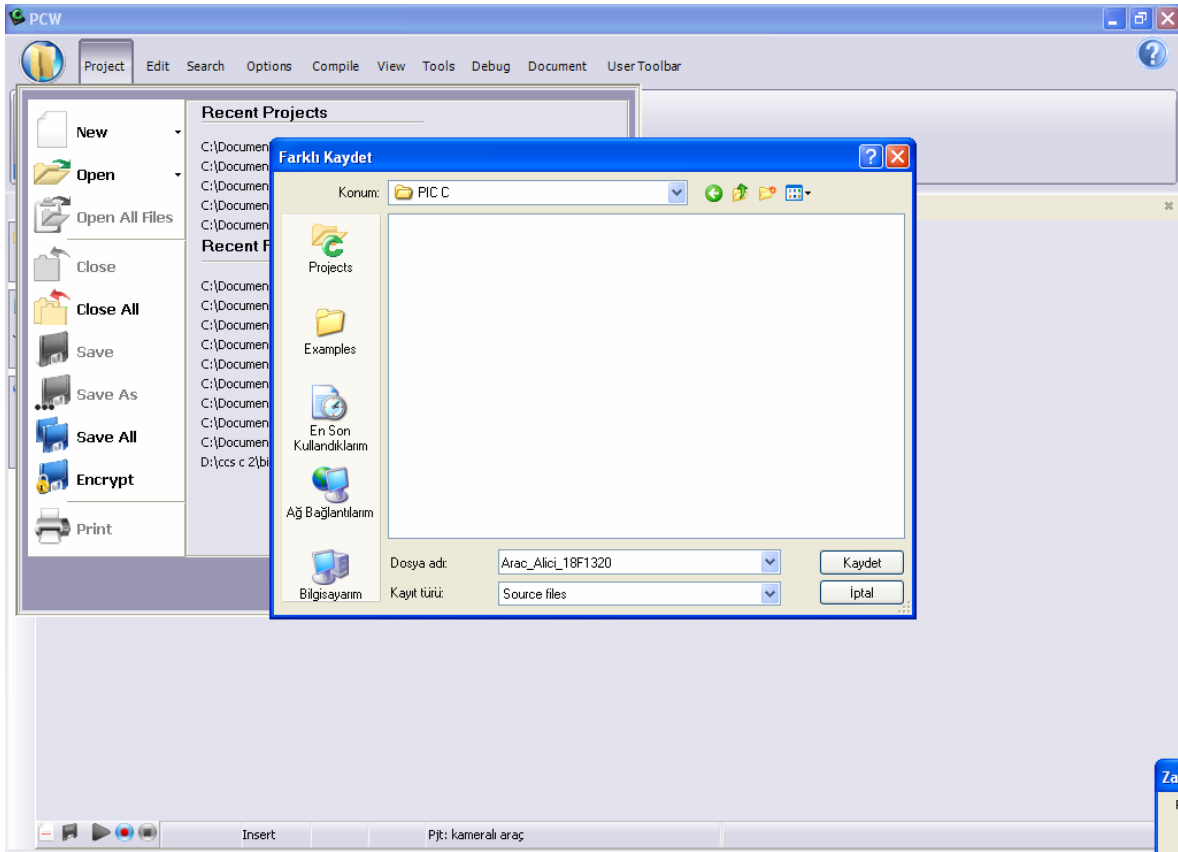
4.1.1. CCS C Derleyicisinde Yeni Bir Proje Olu turmak

CCS C de yeni proje olu turmak için sırasıyla u i lemler gerçekte rilir: İlk ba ta bir kaynak dosya (Source File) olu turulmaktadır. Bunun için “File” menüsünden “New” komutu seçilir oradan da “Source File” komutu tıklanır (ekil 4.1).



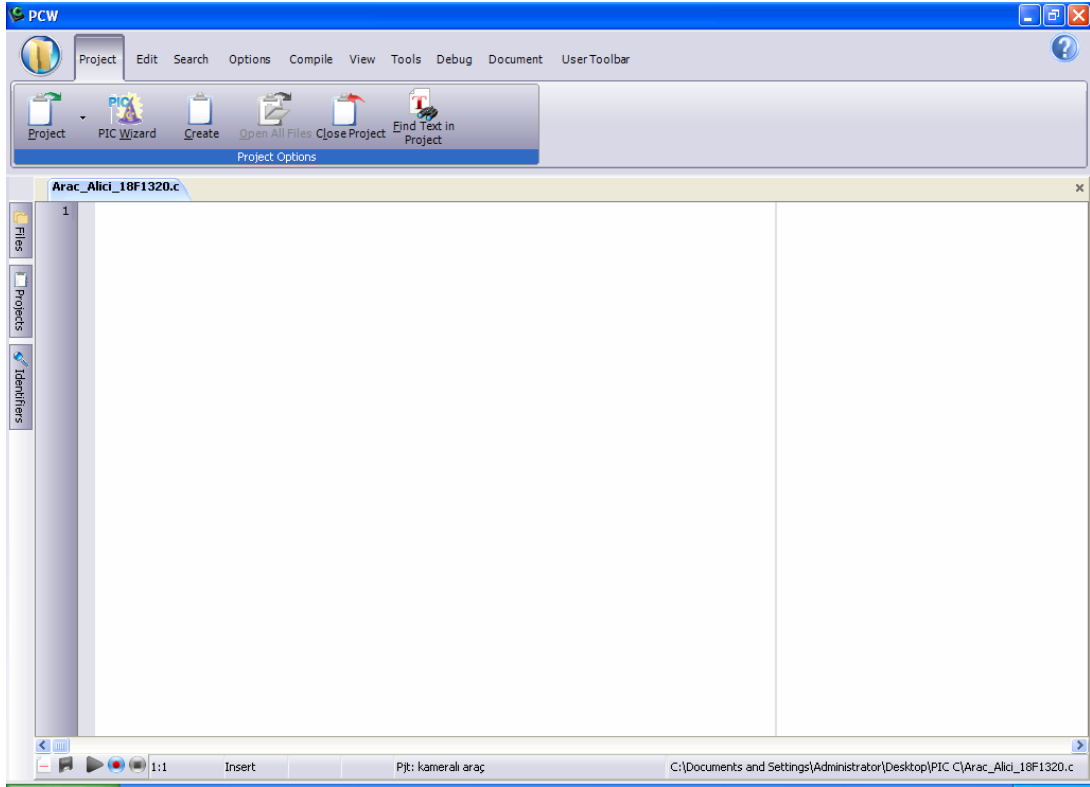
ekil 4.1. Kaynak Dosya Olu turma

Gelen menüde dosyanın kaydedilece i yer ve ismi belirtilir (ekil 4.2). Proje masaüstünde PIC C klasörüne Araç_Alıcı_18F1320 ismi adı altında kaydedilmektedir.

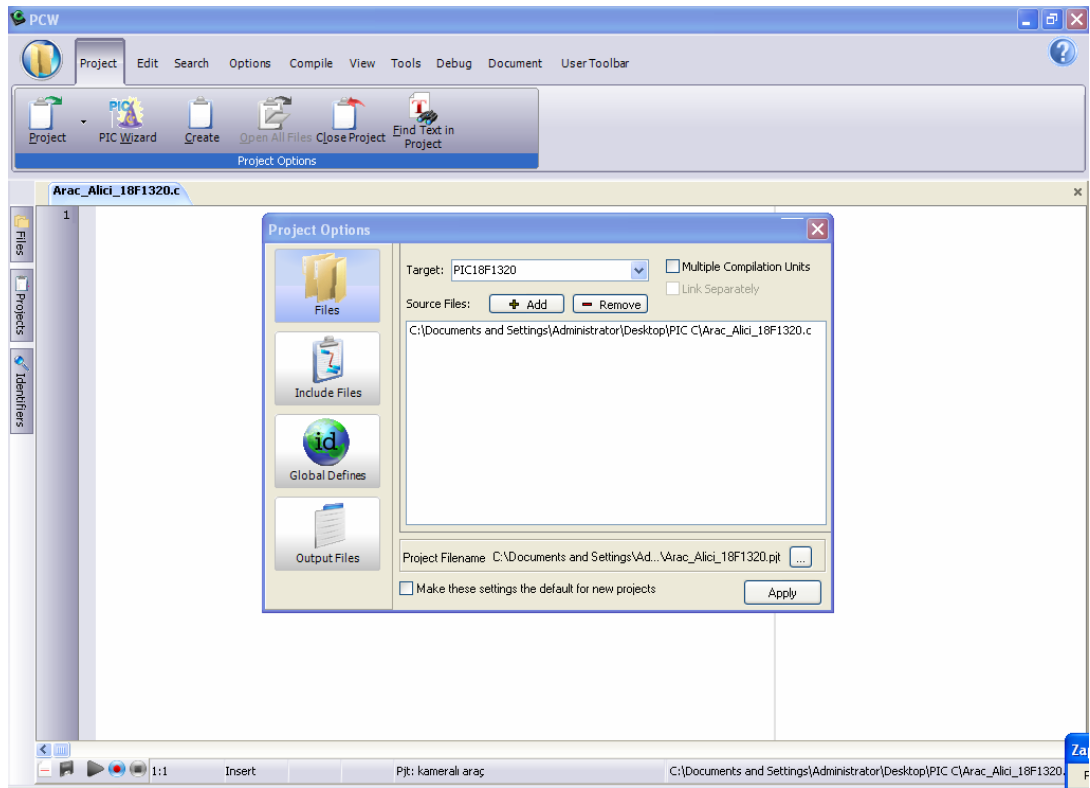


ekil 4.2. Kaynak Dosyanın Kayıt Yeri ve sminin Belirlenmesi

Daha sonra programı yazacağımız ekran karşımıza gelir (ekil 4.3). Daha sonra ayarlamaları yapmak için “Project” menüsüne tıklanır ve gelen sekmelerden “Create” komutu seçilir. Gelen menüde Araç_Alici_18F1320 kaynak dosyası seçilir ve aç butonu tıklanır. Gelen menüde kullanılacak mikrodenetleyici ve kullanılacak kaynak dosya seçilir (ekil 4.4). Bu menüde sol taraftaki sekmelerden “Include Files” butonu tıklanarak programın derleme esnasında kullanılacak dosyaların hangi dizinde olduğunu belirtir.”Output File” butonuna tıklanmış ise önümüze derleme sonucunda oluşturulacak dosyaların seçildiği menü ekrana gelir. Tüm bu işlemler yapıldıktan sonra “Apply” butonu tıklanır.



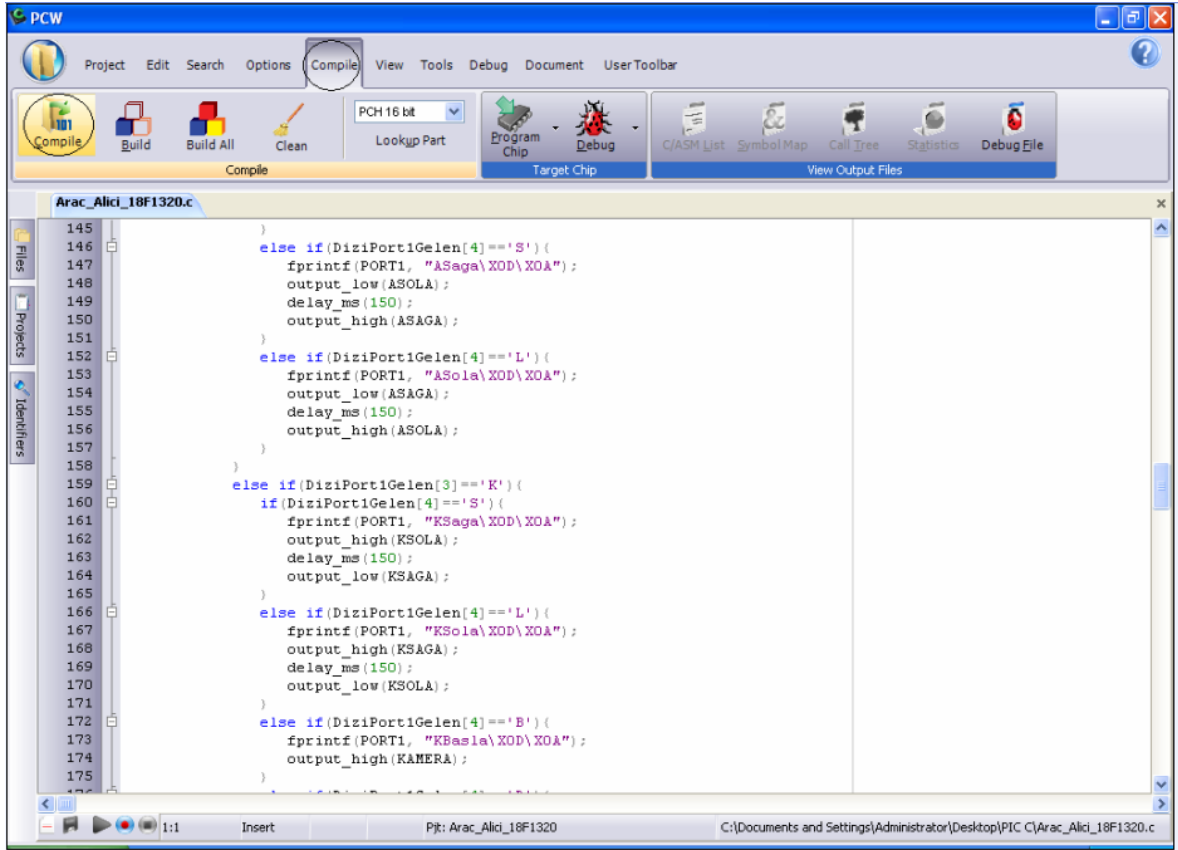
ekil 4.3. Program Yazılan Ekran



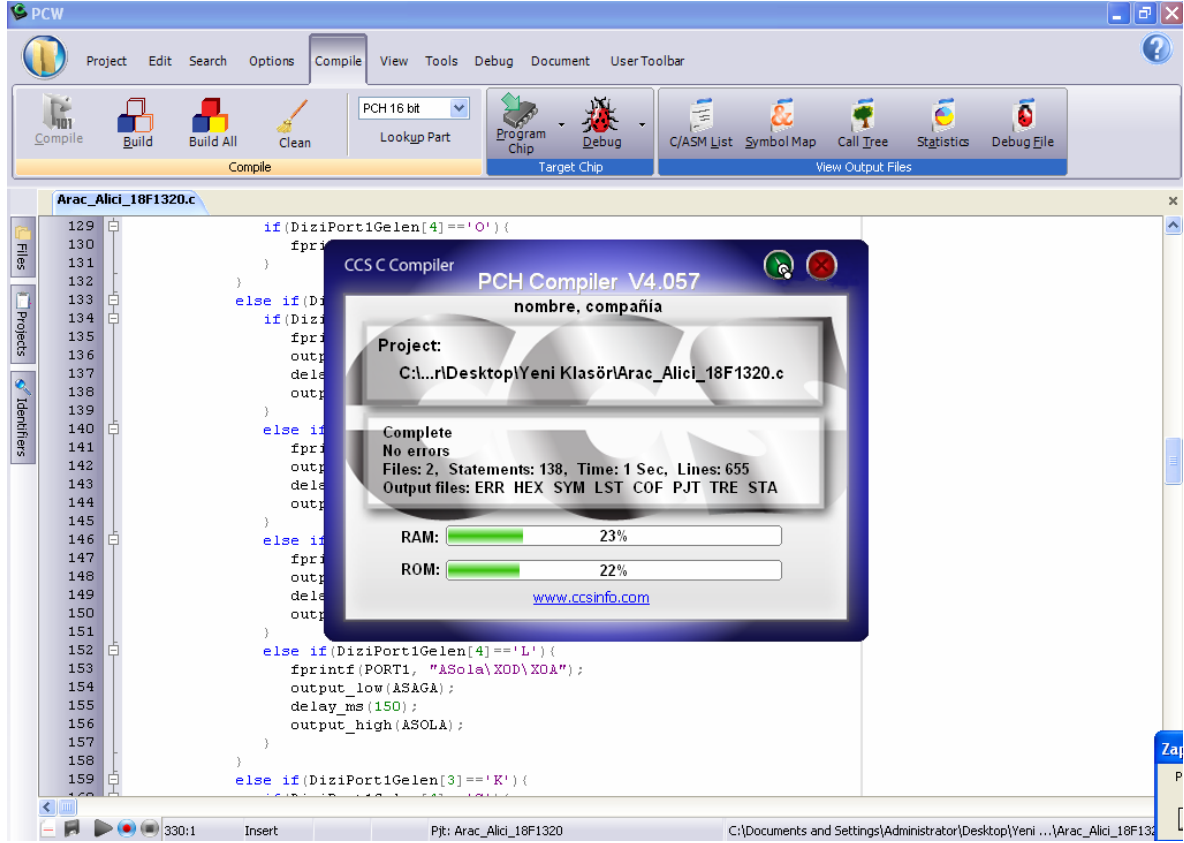
ekil 4.4. "Project Options" Menüsü

4.1.2. CCS C' de Program Derlemek

Program kodları yazıldıktan sonra dosyanın derlenmesi için “Compile” menüsüne tıklanır ve gelen sekmelerden “Compile” butonuna tıklanır (ekil 4.5) veya F9 tu una basılır. Derleme i lemi sonunda ekrana derleme sonuçlarıyla ilgili bilgiler gelir (ekil 4.6).



ekil 4.5. Proje Dosyasının Derlenmesi



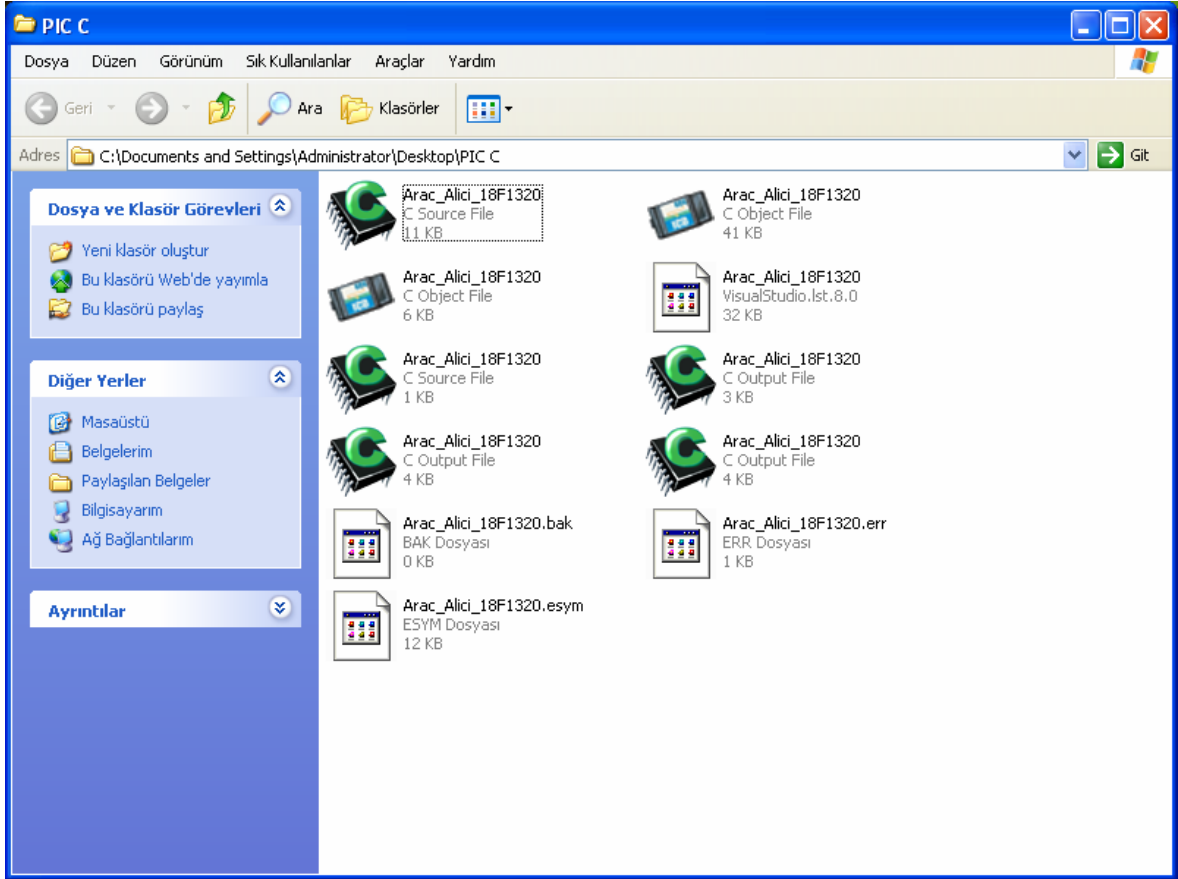
ekil 4.6. Proje Dosyasının Derlenme İlemi Sonucu

4.1.3. CCS C' de Derleme Sonucu Olu an Dosya Formatları

Derleme İlemi sonucunda CCS C programı çe İtli uzantıda dosyalar olu turur (ekil 4.7). A a ıda derleme İlemi sonucunda olu an dosyalar ile bu dosyaların ne İçerdi İ hakkında bilgi verilmi tir.

- .C = C kodları barındıran kaynak dosya.
- .H = Ba lık dosyası veya özel fonksiyon dosyaları.
- .PJT = CCS C proje dosyası.
- .LST = CCS C komutlarının assembly kar ılı mın listelendi İ dosya.
- .SYM = Kaydedicilerin ve program de İ kenlerinin kaydedildi İ adresleri gösteren dosya.
- .STA = RAM, ROM ve yı n(stack) kullanım istatisti İni veren dosyadır.
- .TRE = Programda kullanılan fonksiyonları ve RAM ve ROM'daki adreslerini verir.

- .HEX = Programın makine koduna çevrilmi dosyası.
- .COF = Makine kodları ile hata ayıklama(debugger) bilgilerinin oldu u dosya.
- .COD = Hata ayıklama bilgilerinin oldu u ikilik kodlardan olu mu dosya.
- .ERR =Derleme sonucu olu an hata(error) ve uyarıları(warning) veren dosya.
- .ESYM = IDE (Integrated Development Environment –tümle ik geli tirme ortamı) kullanıcıları için olu turulan bir dosyadır. Program de i ken ve açıklama bilgileri yer alır.
- .RFT = CCS C kurulum dizini içinde “rtfeditör.exe” adında bir program vardır. Bu dosya word programı gibi bir metin editörü programıdır. Bu program ile hazırlanan metin dosyaları. RFT(Rich Text File) uzantılı olarak kaydedilir.
- .RVF = RVF(Rich View Format) dosya uzantısı, CCS C kurulum dizini içinde “rtfeditör.exe” programı olu turulan metin dosyalarının asıl kayıt uzantısıdır.
- .DGR = CCS C kurlum dizini içinde “flowcharter.exe” adında bir akı diyagramı çizmeye yarayan bir program vardır. Bu program. DGR uzantılı dosyalar olu turmaktadır (Çiçek, 2007).



ekil 4.7. Derleme Sonucu Olu an Dosyalar

4.1.4. CCS C' de Program Kodlarının Açıklaması

PIC C program kodlarının bir kısmı aşağıda açıklanmaktadır. Programın tamamı ise EK 1'de verilmektedir.

```
#include <18F1320.h> // Kullanılacak denetleyicinin başlık dosyası tanıtılıyor.
```

```
#fuses WDT, WDT128, INTRC_IO, NOFCMEN, BROWNOUT, BORV27, PUT, CPD,
STVREN, NODEBUG, NOLVP, WRT, NOWRTD, NOWRTC, IESO, NOEBTR,
NOEBTRB, NOMCLR, PROTECT, NOCPB, NOWRTB // Denetleyici konfigürasyon
ayarları.
```

```
#use delay(clock=8000000, RESTART_WDT) // Gecikme fonksiyonu için kullanılacak
osilatör frekansı belirtiliyor.
```

```
#use rs232(baud =600, PARITY=N, xmit=PIN_B1, rcv=PIN_B4, BITS =8, ERRORS,
STOP=2, STREAM = PORT1) // RS232 protokolü ayarları belirtiliyor.
```

```
#define AILERI PIN_A1 // Sabit tanımlaması yapılıyor, program her A LER gördü ü  
yerde P N_A1 varmı gibi i lem yapar.
```

```
#define AGERI PIN_A0
```

```
#define ASAGA PIN_A3
```

```
#define ASOLA PIN_A4
```

```
#define KAMERA PIN_A7
```

```
#define KSAGA PIN_B2
```

```
#define KSOLA PIN_B3
```

```
//-----
```

```
char DiziPort1Gelen[13]; // Karakter türünde dizi adı bir de i ken tanımlanıyor.
```

```
int KarakterRs232; // Rs232 adında 8 bit'lik de i ken tanımlanıyor.
```

```
int SayacPort1Gelen;
```

```
//-----
```

```
#int_RDA // RX ucuna veri gelince meydana gelen kesme.
```

```
void SeriPort1DataGeldiInterrupt(void)
```

```
{
```

```
    KarakterRs232 = fgetc(PORT1); // PORT1'den gelen veriyi KarakterRs232'ye aktar.
```

```
    if(KarakterRs232==3 && DiziPort1Gelen[0]==2) // DiziPort1Gelen'den gelen 0. veri  
    2(start of text) ise ve KarakterRs232 3(end of text) ise a a ıdaki fonksiyonları i le.
```

```
    {
```

```
        if(DiziPort1Gelen[3]=='R'){ // DiziPort1Gelen'den 3. veri 'R' ve 4. veri 'S' ise  
mikrodenetleyiciyi resetle.
```

```
            if(DiziPort1Gelen[4]=='S'){
```

```
                reset_cpu(); // Kullanılan mikrodenetleyicinin resetlenmesini sa lar.
```

```
            }
```

```
        }
```

else if(DiziPort1Gelen[3]==‘S’){ // DiziPort1Gelen’den 3. veri ‘S’ ve 4. veri ‘T’ ise aracı stopla.

```
if(DiziPort1Gelen[4]==‘T’){  
    output_low(AILERI); // Bu komut ile PIN_A1 ucu lojik 0 yapılımtır.  
    output_low(AGERI);  
    output_low(ASAGA);  
    output_low(ASOLA);  
    output_high(KSAGA);  
    output_high(KSOLA);  
}  
}
```

else if(DiziPort1Gelen[3]==‘A’){ // DiziPort1Gelen’den 3. veri ‘A’ ve 4. veri ‘T’ ise araç ileri, de ilse bir alt komuta bak.

```
if(DiziPort1Gelen[4]==‘T’){  
    output_low(AGERI); // Bu komut ile PIN_A0 ucu lojik 0 yapılımtır.  
    delay_ms(150); // 150 ms gecikme veriliyor.  
    output_high(AILERI); // Bu komut ile PIN_A1 ucu lojik 1 yapılımtır.  
}
```

else if(DiziPort1Gelen[4]==‘G’){ // DiziPort1Gelen’den 3. veri ‘A’ ve 4. veri ‘G’ ise arç geri, de ilse bir altkomuta bak.

```
output_low(AILERI);  
delay_ms(150);  
output_high(AGERI);  
}
```

//-----

#int_RTCC // Timer 0 kesmesi.

void Timer9KesmeGeldi(void)

```
{
    restart_wdt(); // getc() fonksiyonu veri okumayı beklerken WDT'ı sıfırlayarak, WDT'ın
denetleyici resetlemesini önler.
}

void main(void) // Ana program fonksiyonu.
{
    setup_adc(ADC_OFF); // ADC biriminin kullanılmayacağını belirtir

    setup_adc_ports(NO_ANALOGS); // Denetleyicide A/D çevrimi için ayrılan hiçbir
pin'in analog girişi için kullanılmayacağını belirtir.

    SET_TRIS_A(0b00000000); // A portu çıkışı olarak kullanılacağını belirtir.

    SET_TRIS_B(0b11111100); // B0, B1 pin'i çıkış diğer pin'lerin girişi olarak
kullanılacağını belirtir.

    setup_timer_0(RTCC_INTERNAL | RTCC_8_BIT | RTCC_DIV_128); // Timer 0
zamanlayıcı ayarlarını yapmaya yarayan fonksiyondur. Sinyal kaynağının dahili olduğu ve
bölme oranının 128 olduğu belirtilmiştir.

    port_b_pullups(TRUE); // B portunun pull-up özelliği aktif hale getiriliyor.

    output_low(AILERI); // Bu komut ile PIN_A1 ucu lojik 0 yapılmıştır.

    output_low(AGERI);

    output_low(ASAGA);

    output_low(ASOLA);

    output_high(KSAGA);

    output_high(KSOLA);

    output_low(KAMERA);

    enable_interrupts(INT_RTCC); // nt_rtcc kesmesi aktif edilir.

    enable_interrupts(INT_RDA); // nt_rda kesmesi aktif edilir.

    enable_interrupts(GLOBAL); // Aktif edilen tüm kesmeler izin verir.

    while(TRUE) // Sonsuz döngü oluşturulur.
```

```
{  
    ;  
}  
reset_cpu(); // Mikrodenetleyiciyi resetler.  
}
```

4.1.4.1. CCS C' de Öni lemci Fonksiyonları

PIC C programında ön i lemci fonksiyonları tablo halinde Ekler bölümünde ekil Ek.3.1'de verilmi tir. Bu programda kullanılan bazı ön i lemci fonksiyonları a a ıda açıklanacaktır.

#fuses: Bu komut ile kullanılan mikrodenetleyici ile ilgili konfigürasyon bit'lerinin durumu belirtilir. A a ıda konfigürasyon kodları ve açılımları verilmi tir.

- LP = Low Power Osc. –Dü ük frekanslı kristal.
- XT = Kristal veya seramik resonatör.
- HS = High Speed – Yüksek hızlı kristal veya seramik resonatör.
- RC = Resistör Capacitor- Direnç- kondansatör osilatör.
- WDT = Watch Dog Timer kullanılmak istenirse seçilir. WDT birimi denetleyici içinde gömülü bulunan ve ek olarak dı arıdan hiçbir elemanın ba lantısına gerek duymayan bir R/C osilatördür. Denetleyici osilatör giri lerine ba lanan (OSC1/CLKIN ve OSC2/CLKOUT) osilatörden ba ımsızdır. WDT birimi, denetleyici içindeki programın veya mikrodenetleyicinin kontrol etti i bir sistemin kararlı çalı masının bekçili ini yapar. Kullanım amacı PIC denetleyicilerde programın istenmeyen bir ekilde kilitlenmesi durumunda PIC'in resetlenmesini sa lamaktır.
- NOWDT = No Watch Dog Timer. WDT kullanılmayacaksa seçilir.
- PUT = Power Up Timer isteniyorsa seçilir.
- NOPUT = No Power Up Timer. PUT istenmiyorsa seçilir.
- CPD = Code Protected Data. Data EEPROM belle inin korumalı olması istendi inde seçilir. Data EEPROM koruması aktif ise dı arıdan cihazların data EEPROM içeri ine ula ması engellenir.

- NOCPD = No Code Protect Data. EEPROM belle i koruması istenmedi inde kullanılır.
- PROTECT = Program belle ine(ROM) yazılan verilerin okunmasını engeller.
- NOPROTECT = ROM belle e okumaya kar ı koruma koymaz.
- WRT = Program hafızasına yazma i lemini korumaya almak için kullanılır.
- NOLVP = No Low Voltage Programing. Dü ük gerilimle programlama modu istenmedi inde seçilir (Çiçek, 2007).

PIC C programında bu fonksiyon #fuses WDT, WDT128, INTRC_IO, NOFCMEN, BROWNOUT, BORV27, PUT, CPD, STVREN, NODEBUG, NOLVP, WRT, NOWRTD, NOWRTC, IESO, NOEBTR, NOEBTRB, NOMCLR, PROTECT, NOCPB, NOWRTB ekinde kullanılmı tır.

nclude: Bu komut ile kullanılacak mikrodenetleyicinin tanımlama dosyası derleyiciye tanıtılır. #include <18F1320.h> gibi.

4.1.4.2. #USE RS232() Fonksiyonu

Bu komut derleyiciye hangi pin'lerin seri ileti im için kullanılaca nı, seri ileti im hızının ne olaca nı, ileti im için parity bit'inin ne olaca ı gibi seri ileti im ile ilgili ayarları derleyiciye bildiren ön i lemci komutudur. Fonksiyondaki "sabitler" kısmına a a ıda verilen sabitlerden kullanılması gerekenler yazılır. Her sabit virgöl ile birbirinden ayrılır. PIC 18F1320'nin içinde RS232 ileti imi için UART donanım modülü vardır. Bu modülü kullanarak RS232 ileti imi belli ve sabit pin'lerle yapılır. Fakat CCS C derleyicisi bize istedi imiz pin'leri kullanarak RS232 ileti imi sa lar.

#use rs232 sabitleri

- Baud=x RS232 ileti im hızını belirten sabit (baud=9600 gibi).
- Xmit=pin RS232 veri gönderme ucunu belirten sabittir (xmit=pin_A2).
- Rcv=pin RS232 veri alma ucunu belirten sabittir (rcv=pin_A3).
- Parity=x RS232 ileti im e lik bit'i durumunu bildirir.x yerine N(e lik bit'i yok), E(e lik bit'i çift),O(e lik bit'i tek),(parity=N gibi).
- Stop=x RS232 stop bit'inin kaç bit oldu unu belirtir. X yerine 1 veya 2 yazılır. Varsayılan stop bit'i 1'dir.

- nvert RS232 pin'lerinin polaritesini de i tirir. Yani xmit ile rcv pinleri yer de i tirir.
- Uart1 xmit ve rcv pin'leri olarak denetleyici içinde bulunan birinci uart modülü pin'lerinin kullanılaca ını belirtir.
- Uart2 xmit ve rcv pin'leri olarak denetleyici içinde bulunan ikinci uart modülü pin'lerinin kullanılaca ını belirtir.
- Restart_wdt= getc() fonksiyonu veri okumayı beklerken WDT'ı sıfırlayarak, WDT'in denetleyici resetlemesini önler (Çiçek, 2007).

PIC C programında bu fonksiyon #use rs232(baud=600, xmit=PIN_B1, rcv=PIN_B4, stop=2, parity=N) ekinde kullanılmaktadır. Bu komut ile ileti im hızının 600 oldu unu, veri gönderme ucunun B port'unun 1.bit'i, veri alma ucunun B port'unun 4.bit'i oldu unu, stop bit'inin 2 bit olaca ını, e lik bit'inin olmayaca ını belirtir.

4.1.4.3. RS232 Giri /Çıkı Fonksiyonları

RS232 ileti imi sırasında karakter alımı –gönderimi, string (ifade) alımı –gönderimi gibi i lemlerin yapılmasını sa layan fonksiyonlardır. Bu fonksiyonların kullanılabilmesi için programın ba ında # use rs232 ön i lemcisi fonksiyonun tanıtılması gerekmektedir. Temel C dilinde olan ve CCS C'de kullanılan giri /çıkı fonksiyonları Çizelge 4.1.'de verilmi tir (Çiçek, 2007).

Çizelge 4.1. CCS C Derleyicisi,RS232 Giri /Çıkı Fonksiyonları

Komut	lem
Getchar()	Klavyeden bir karakter okur ve enter tu una basılmasını bekler.
Getch()	Kavyeden bir karakter okur ve enter tu una basılmaksızın okunan karakteri bir karaktere aktarır. Klavyeden girilen karakter ekranda görüntülenmez.
Putc(), Putchar()	Ekrana bir karakter yazar.
Gets()	Kavyeden girilen bir stringi(kelime dizini) bir de i kene aktarır.
Puts()	Bir stringi ekrana yazar.
Printf()	String ifadelerin belli bir formatta gönderilmesini sa lar.

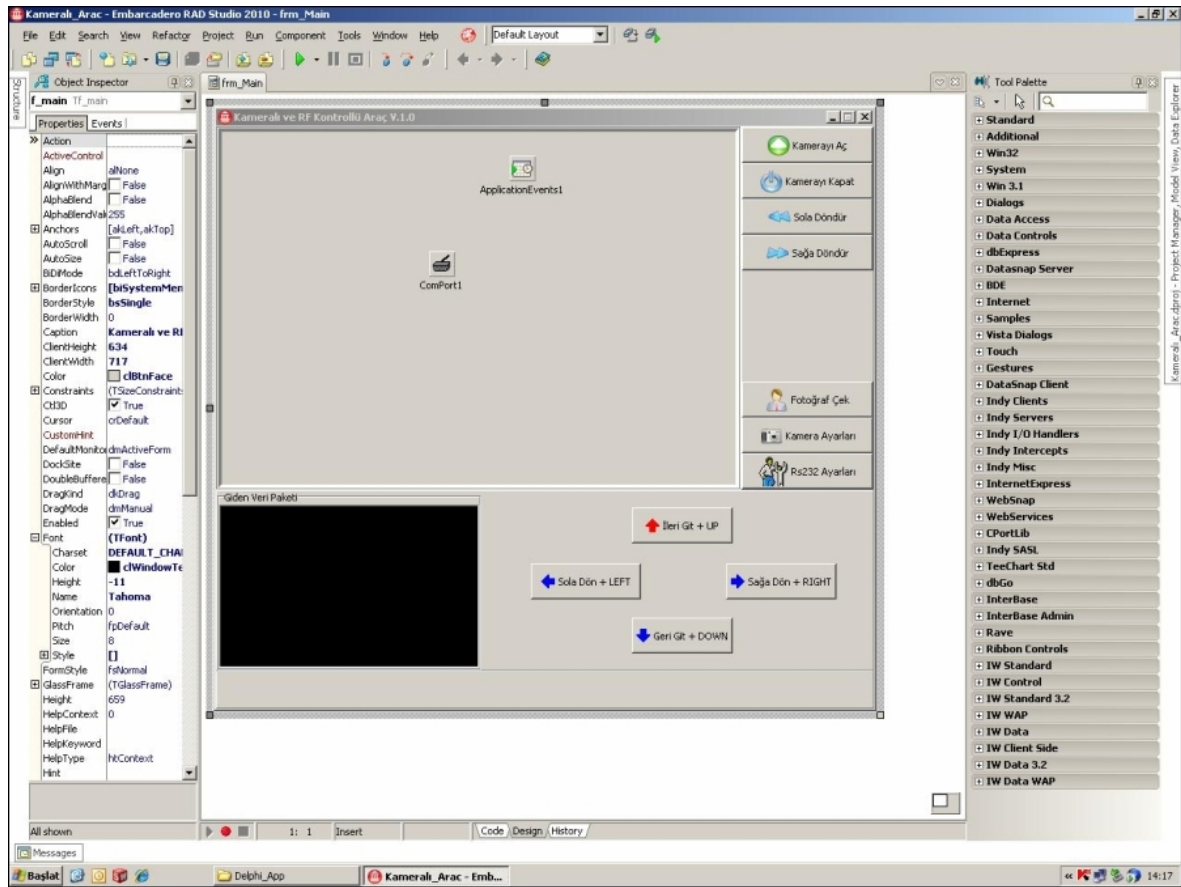
4.1.4.4. RS232 Seri leti im Kesmeleri

INT_ RDA kesmesi seri denetleyicinin seri haberle me veri alma ucuna (RX) bilgi geldi i zaman aktif olur. Kesmenin kullanımı önceden anlatılan kesme i lemleri ile

aynıdır. Fakat bu kesmede kesme fonksiyonunda, kesme pasif haline getirilmelidir. Aksi takdirde program hep kesmeye gidecektir. Bazı denetleyicilerde INT_RDA0, INT_RDA1, INT_RDA2 kesmeleri de bulunmaktadır. Aynı zamanda gönderme tamponu boş oldu unda meydana gelen #INT_TBE (Transmit Buffer Empty) kesmeside mevcuttur (Çiçek, 2007).

4.2. Uygulamanın Arayüz Programı Kısmı

Tüm kontrolleri bilgisayardan yapabilmek için Delphi 10.0 yazılımı kullanılarak bir arayüz kontrol programı yazılmıştır. Ara yüz ekran paneli ekil 4.8'de gösterilmiştir.



ekil 4.8. Ara Yüz Ekran Paneli

4.2.1. Delphi Kodlarının Açıklamaları

Delphi kodlarının bir kısmını aşağıda açıklanmaktadır.

```
unit frm_Main;
```

```
interface
```

```
uses
```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls, Buttons, CPort; AppEvents;

type

```
Tf_main = class(Tform)
```

```
  p_alt: TPanel;
```

```
  p_ekran: TPanel;
```

```
  p_veriler: TPanel;
```

```
  p_video: TPanel;
```

```
  b_ba lat: TSpeedButton;
```

```
  b_durdur: TSpeedButton;
```

```
  b_fotograf: TSpeedButton;
```

```
  b_kamsol: TSpeedButton;
```

```
  b_kamsag: TSpeedButton;
```

```
  b_ileri: TSpeedButton;
```

```
  b_geri: TSpeedButton;
```

```
  b_saga: TSpeedButton;
```

```
  b_sola: TSpeedButton;
```

```
  ComPort1: TComport;
```

```
  gb_giden: TGroupBox;
```

```
  m_giden: TNemo;
```

```
  b_ayarlar: TSpeedButton;
```

```
  b_kamera_set: TSpeedButton;
```

```
procedure Tf_main.b_ileriClick(Sender: TObject); // ----- Araç leri Butonu
```

```
begin
```

```
  ComPort1.WriteStr('UUU' + paket_chk_sum('00AI'));
```

```
m_giden.Lines.Insert(0, 'Araca leri Komutu Verildi...' + FormatDateTime('hh:mm:ss',
Now));

end;

//-----Araç Geri Butonu

procedure Tf_main.b_geriClick(Sender: TObject);

begin

    ComPort1.WriteStr('UUU' + paket_chk_sum('00AG'));

    m_giden.Lines.Insert(0, 'Araca Geri Komutu Verildi...' + FormatDateTime('hh:mm:ss',
Now));

end;

//-----Araç Sağa Butonu

procedure Tf_main.b_sagaClick(Sender: TObject);

begin

    ComPort1.WriteStr('UUU' + paket_chk_sum('00AS'));

    m_giden.Lines.Insert(0, 'Araca Sağa Komutu Verildi...' + FormatDateTime('hh:mm:ss',
Now));

end;

//-----Araç Sola Butonu

procedure Tf_main.b_solaClick(Sender: TObject);

begin

    ComPort1.WriteStr('UUU' + paket_chk_sum('00AL'));

    m_giden.Lines.Insert(0, 'Araca Sola Komutu Verildi...' + FormatDateTime('hh:mm:ss',
Now));

end;

//-----Kamera Sağa Butonu

procedure Tf_main.b_kamsagClick(Sender: TObject);

begin
```

```
ComPort1.WriteStr('UUU' + paket_chk_sum('00KS'));

m_giden.Lines.Insert(0, 'Kameraya Saga Komutu Verildi...' +
FormatDateTime('hh:mm:ss', Now));

end;

//-----Kamera Sola Butonu

procedure Tf_main.b_kamsolClick(Sender: TObject);

begin

ComPort1.WriteStr('UUU' + paket_chk_sum('00KL'));

m_giden.Lines.Insert(0, 'Kameraya Sola Komutu Verildi...' +
FormatDateTime('hh:mm:ss', Now));

end;

//-----RS232 Com Port Yazma lemi.

function TCustomComPort.WriteStr(const Str: AnsiString): Integer;

begin

if FOverlapped then

result := _WriteStrWrapper(Str)

else begin

result := 0;

if _SyncWrite(PAnsiChar(Str),Length(Str)) then

result := Length(Str);

end;

end;

//-----Ba lat Butonu levi

procedure Tf_main.b_baslatClick(Sender: TObject);

begin

ComPort1.WriteStr('UUU' + paket_chk_sum('00KB'));

m_giden.Lines.Insert(0, 'Kamera Ba latıldı...' + FormatDateTime('hh:mm:ss', Now));
```

```
// ----  
  
try  
  
    Video := TVideo.Create(p_ekran);  
//    TVideo.Create(p_ekran);  
  
except  
  
    on E:Exception do ShowMessage('Foto raf Çekme leminden Sonra Kamera Ba lat  
Yapınız...');  
  
    end;  
  
end;  
  
//-----Durdur Butonu levi  
  
procedure Tf_main.b_durdurClick(Sender: TObject);  
  
begin  
  
    ComPort1.WriteStr('UUU' + paket_chk_sum('00KD'));  
  
    m_giden.Lines.Insert(0, 'Kamera Durduruldu...' + FormatDateTime('hh:mm:ss', Now));  
  
    // ----  
  
    try  
  
        Video.Destroy;  
  
    except  
  
        end;  
  
end;  
  
//-----Foto raf Çek Butonu levi.  
  
procedure Tf_main.b_fotografClick(Sender: TObject);  
  
begin  
  
    try  
  
        Video.TakePicture('Fotograf');  
  
        m_giden.Lines.Insert(0, 'Foto raf Çekildi...' + FormatDateTime('hh:mm:ss', Now));
```

```
except
    on E:Exception do ShowMessage('Foto raf Çekme leminden Önce Kamerayı
Ba latınız...');
end;
end;
constructor TVideo.Create(Owner: TPanel); // ----- Video Ekranı Olu turuluyor.
begin
    try
        VideoHwnd := capCreateCaptureWindowA('Video', WS_CHILD or WS_VISIBLE, 0,
0, Owner.Width, Owner.Height, Owner.Handle, 0);
        If (SendMessage(VideoHwnd, WM_CAP_DRIVER_CONNECT, 0, 0) <> 0) then begin
            SendMessage(VideoHwnd, WM_CAP_SET_PREVIEW, -1, 0);
            SendMessage(VideoHwnd, WM_CAP_SET_PREVIEWRATE, 100, 0);
            SendMessage(VideoHwnd, WM_CAP_SET_SCALE, -1, 0);
            Parent := Owner;
            Owner.OnResize := Resize;
        end;
    except
        ShowMessage('Video Penceresi Olu turulamıyor!');
    end;
    BMPFile := ExtractFilePath(Application.ExeName) + 'pic.bmp';
end;
//-----Video Ekranı Yok Ediliyor.
destructor TVideo.Destroy;
begin
    if (VideoHwnd <> 0) then begin
        SendMessage(VideoHwnd, WM_CAP_DRIVER_DISCONNECT, 0, 0);
```

```
SetParent(VideoHwnd, 0);  
SendMessage(VideoHwnd, WM_CLOSE, 0, 0);  
end;  
inherited;  
end;  
//-----Video Kayna 1 Ayarlanıyor.  
procedure TVideo.SetSource;  
begin  
    SendMessage(VideoHwnd, WM_CAP_DLG_VIDEOSOURCE, 0, 0);  
end;
```

5. SONUÇ VE ÖNERİLER

Uygulamalı olarak yapılan bu tezde PC ile uzaktan bir mobil arabanın RF sinyalleri vasıtası ile kontrolü gerçekleştirilmiştir. Araç bilgisayardan kontrol etmek için Delphi görsel programlama dili kullanılmıştır. Delphi programıyla oluşturulan kontrol paneli sayesinde butonlara basıldığında seri porttan bir karakter verici entegreye gitmekte ve oradan da anten vasıtasıyla alıcı entegre devresine gönderilmektedir. Araca gelen veriler de mikrodenetleyiciye iletilerek ilgili entegreler ve bunlara bağlı cihazlar kontrol edilmektedir. Mikrodenetleyici, cihazları kontrol edebilmesi için PIC C programlama dili ile derlenmiştir.

Mobil arabanın kontrolü ve inceleme için ortamın görüntüleri RF sinyalleri ile çalışan bir kamera sayesinde alınarak kontrol merkezindeki bilgisayara aktarılmıştır. Böylece, güvenlik nedeniyle insanların giremediği bir ortama mobil arabanın gönderilmesi ve uzaktan gerekli fotoğraflar ve görüntülerin alınması mümkün hale gelmiştir. Burada temel amaç insan hayatını kolaylaştırmak ve insan yaşamı için riskli olabilecek bölgelerde daha sağlıklı bir çalışma ortamı sağlamaktır. Bu tip araçlar maden aramalarında, su altı aramalarında, arama kurtarma çalışmalarında, uzay araştırmalarında ve askeri keşiflerde kullanılmaktadırlar.

Mobil aracın geliştirilmesine yönelik arabaya mekanik bir kol eklenip, kullanım sahasında iş yapılabilir. Ayrıca bilgisayara gelen görüntü işlenerek mobil aracın otomatik yönlendirilmesi yapılabilir.

KAYNAKLAR

- Altınba ak, O. 2005. Mikrodenetleyiciler ve PIC Programlama. Melisa Matbaası, stanbul, 323s.
- Anonim, 2004. ARX-34_TD_E Data Sheet, UHV Ask Data Receiver, Udea, www.udea.com.tr.
- Anonim, 2004. ATX-34_TD_E Data Sheet, UHV Ask RF Transmitter, Udea, www.udea.com.tr.
- Anonim, 2005. Max232 Data Sheet, Maxim, www.maxim-ic.com.
- Anonim, 2005. PIC18F1320 Data Sheet, Microchip, www.microchip.com.tr.
- Anonim, 2007. Bili im Teknolojileri: Röle, MEGEP, Ankara, 31s.
- Anonim, 2008. Texas Instrument, 3-Terminal Voltage Regulator, 7805 Data Sheet, www.alldatasheets.com.
- Baluja, S., Pomerlau, D.A. 1997. Robotics and Autonomous Systems, 329s.
- Buchi, B, 1996. at all, Seventh International Symposium on Micro Machine and Human Science, 203.
- Creman, L.B. 2006. at all, Journal of Field Robotics,
- Çavun o lu, ., Kırmızı, F., 2007. Seri Port ile Haberle ebilen Uzaktan Kumandalı Araç – Yıldız Teknik Üniversitesi Projesi, A ustos 2007 STANBUL, s.20-25
- Çiçek, S. 2007. CCS C le PIC Programlama. Melisa Matbaası, stanbul, 425s.
- Lı, W., Lu, G., Wang, Y. 1997. Pattern Recognition Letters, 771s.
- Ollero, A. 1999. at all, Control Engineering Practice, 1291s.
- ahin, H., Dayanık, A., Altınba ak, C., 2006. PIC Programlama Teknikleri ve PIC16F877A. Melisa Matbaası, stanbul, 527s
- Victor, J.S. 1998. Robotics and Autonomous Systems, 221s.

EKLER**EK-1: PIC C Kodları**

```
#include <18F1320.h>

#fuses      WDT, WDT128, INTRC_IO, NOFCMEN, BROWNOUT, BORV27, PUT,
CPD, STVREN, NODEBUG, NOLVP, WRT, NOWRTD, NOWRTC, IESO, NOEBTR,
NOEBTRB, NOMCLR, PROTECT, NOCPB, NOWRTB

#use      delay(clock=8000000, RESTART_WDT)

#use      rs232(baud =600, PARITY=N, xmit=PIN_B1, rcv=PIN_B4, BITS =8, ERRORS,
STOP=2, STREAM = PORT1)

#define    AILERI PIN_A1

#define    AGERI PIN_A0

#define    ASAGA PIN_A3

#define    ASOLA PIN_A4

#define    KAMERA PIN_A7

#define    KSAGA PIN_B2

#define    KSOLA PIN_B3

char DiziPort1Gelen[13];

int KarakterRs232;

int SayacPort1Gelen;

#int_RDA

void SeriPort1DataGeldiInterrupt(void)

{

    KarakterRs232 = fgetc(PORT1);

    if(KarakterRs232 == 2 || SayacPort1Gelen > 12) SayacPort1Gelen = 0;

    else if(KarakterRs232==3 && DiziPort1Gelen[0]==2)

    {
```

EK-1 Devam: PIC C Kodları

```
if(DiziPort1Gelen[3]=='R'){
    if(DiziPort1Gelen[4]=='S'){
        reset_cpu();
    }
}
else if(DiziPort1Gelen[3]=='S'){
    if(DiziPort1Gelen[4]=='T'){
        fprintf(PORT1, "Dur\X0D\X0A");
        output_low(AILERI);
        output_low(AGERI);
        output_low(ASAGA);
        output_low(ASOLA);
        output_high(KSAGA);
        output_high(KSOLA);
    }
}
else if(DiziPort1Gelen[3]=='A'){
    if(DiziPort1Gelen[4]=='T'){
        output_low(AGERI);
        delay_ms(150);
        output_high(AILERI);
    }
    else if(DiziPort1Gelen[4]=='G'){
        output_low(AILERI);
        delay_ms(150);
    }
}
```

EK-1 Devam: PIC C Kodları

```
        output_high(AGERI);
    }
else if(DiziPort1Gelen[4]=='S'){
    output_low(ASOLA);
    delay_ms(150);
    output_high(ASAGA);
}
else if(DiziPort1Gelen[4]=='L'){
    output_low(ASAGA);
    delay_ms(150);
    output_high(ASOLA);
}
}
else if(DiziPort1Gelen[3]=='K'){
    if(DiziPort1Gelen[4]=='S'){
        output_high(KSOLA);
        delay_ms(150);
        output_low(KSAGA);
    }
else if(DiziPort1Gelen[4]=='L'){
    output_high(KSAGA);
    delay_ms(150);
    output_low(KSOLA);
}
}
else if(DiziPort1Gelen[4]=='B'){
```

EK-1 Devam: PIC C Kodları

```
        output_high(KAMERA);
    }
    else if(DiziPort1Gelen[4]=='D'){
        output_low(KAMERA);
    }
}
}
}
DiziPort1Gelen[SayacPort1Gelen++] = KarakterRs232;
}

#int_RTCC
void Timer9KesmeGeldi(void)
{
    restart_wdt();
}

void main(void)
{
    setup_adc(ADC_OFF);
    setup_adc_ports(NO_ANALOGS);
    SET_TRIS_A(0b00000000);
    SET_TRIS_B(0b11111100);
    setup_timer_0(RTCC_INTERNAL | RTCC_8_BIT | RTCC_DIV_128);
    port_b_pullups(TRUE);
    output_low(AILERI);
    output_low(AGERI);
    output_low(ASAGA);
```

EK-1 Devam: PIC C Kodları

```
output_low(ASOLA);  
output_high(KSAGA);  
output_high(KSOLA);  
output_low(KAMERA);  
enable_interrupts(INT_RTCC);  
enable_interrupts(INT_RDA);  
enable_interrupts(GLOBAL);  
while(TRUE)  
{  
    ;  
}  
reset_cpu();  
}
```

EK-2: Delphi Kodları

```
unit frm_Main;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, Buttons, CPort; AppEvents;

type

    Tf_main = class(Tform)

        p_alt: TPanel;

        p_ekran: TPanel;

        p_veriler: TPanel;

        p_video: TPanel;

        b_ba_lat: TSpeedButton;

        b_durdur: TSpeedButton;

        b_fotograf: TSpeedButton;

        b_kamsol: TSpeedButton;

        b_kamsag: TSpeedButton;

        b_ileri: TSpeedButton;

        b_geri: TSpeedButton;

        b_saga: TSpeedButton;

        b_sola: TSpeedButton;

        ComPort1: TComport;

        gb_giden: TGroupBox;

        m_giden: TNemo;

        b_ayarlar: TSpeedButton;
```

```
b_kamera_set: TSpeedButton;
```

EK-2 Devam: Delphi Kodları

```
ApplicationEvents1: TApplicationEvents;  
procedure Tf_main.b_ileriClick(Sender: TObject); // ----- Araç İleri Butonu  
begin  
    ComPort1.WriteStr('UUU' + paket_chk_sum('00AI'));  
    m_giden.Lines.Insert(0, 'Araca İleri Komutu Verildi...' + FormatDateTime('hh:mm:ss',  
Now));  
end;  
//-----Araç Geri Butonu  
procedure Tf_main.b_geriClick(Sender: TObject);  
begin  
    ComPort1.WriteStr('UUU' + paket_chk_sum('00AG'));  
    m_giden.Lines.Insert(0, 'Araca Geri Komutu Verildi...' + FormatDateTime('hh:mm:ss',  
Now));  
end;  
//-----Araç Sağa Butonu  
procedure Tf_main.b_sagaClick(Sender: TObject);  
begin  
    ComPort1.WriteStr('UUU' + paket_chk_sum('00AS'));  
    m_giden.Lines.Insert(0, 'Araca Sağa Komutu Verildi...' + FormatDateTime('hh:mm:ss',  
Now));  
end;  
//-----Araç Sola Butonu  
procedure Tf_main.b_solaClick(Sender: TObject);  
begin
```

```
ComPort1.WriteStr('UUU' + paket_chk_sum('00AL'));
```

EK-2 Devam: Delphi Kodları

```
m_giden.Lines.Insert(0, 'Araca Sola Komutu Verildi...' + FormatDateTime('hh:mm:ss',  
Now));
```

```
end;
```

```
//-----Kamera Sağa Butonu
```

```
procedure Tf_main.b_kamsagClick(Sender: TObject);
```

```
begin
```

```
ComPort1.WriteStr('UUU' + paket_chk_sum('00KS'));
```

```
m_giden.Lines.Insert(0, 'Kameraya Sağa Komutu Verildi...' +  
FormatDateTime('hh:mm:ss', Now));
```

```
end;
```

```
//-----Kamera Sola Butonu
```

```
procedure Tf_main.b_kamsolClick(Sender: TObject);
```

```
begin
```

```
ComPort1.WriteStr('UUU' + paket_chk_sum('00KL'));
```

```
m_giden.Lines.Insert(0, 'Kameraya Sola Komutu Verildi...' +  
FormatDateTime('hh:mm:ss', Now));
```

```
end;
```

```
//-----RS232 Com Port Yazma İlemi.
```

```
function TCustomComPort.WriteStr(const Str: AnsiString): Integer;
```

```
begin
```

```
if FOverlapped then
```

```
result := _WriteStrWrapper(Str)
```

```
else begin
```

```
result := 0;
```

```
if _SyncWrite(PAnsiChar(Str),Length(Str)) then
    result := Length(Str);
end;
end;
function Tf_main.paket_chk_sum(veri:string):string;
var H1, i : integer; Hex, HexDigit : string; DEC : LongInt;
begin
    //-----
    DEC := 0;
    for i := 1 to Length(veri) do DEC := Ord(veri[i]) xor DEC;
    //-----
    HexDigit := '0123456789ABCDEF';
    HEX := '';
    repeat
        H1 := DEC mod 16;
        DEC := DEC div 16;
        HEX := concat(Hexdigit[H1+1], HEX);
    until DEC < 1;
    //----- Tek Basamaklı olursa önüne sıfır koy 2 basamaklı yap.
    for i := 1 to 2 - Length(HEX) do HEX := '0' + HEX;
    paket_chk_sum := #02 + veri + #05 + HEX + #03;
end;
//-----Ba lat Butonu levi
procedure Tf_main.b_baslatClick(Sender: TObject);
begin
    ComPort1.WriteStr('UUU' + paket_chk_sum('00KB'));
```

EK-2 Devam: Delphi Kodları

```
m_giden.Lines.Insert(0, 'Kamera Ba latıldı...' + FormatDateTime('hh:mm:ss', Now));  
  
// -----  
  
try  
  
    Video := TVideo.Create(p_ekran);  
  
//    TVideo.Create(p_ekran);  
  
except  
  
    on E:Exception do ShowMessage('Foto raf Çekme leminden Sonra Kamera Ba lat  
Yapınız...');  
  
    end;  
  
end;  
  
//-----Durdur Butonu levi  
  
procedure Tf_main.b_durdurClick(Sender: TObject);  
  
begin  
  
    ComPort1.WriteStr('UUU' + paket_chk_sum('00KD'));  
  
    m_giden.Lines.Insert(0, 'Kamera Durduruldu...' + FormatDateTime('hh:mm:ss', Now));  
  
    // -----  
  
    try  
  
        Video.Destroy;  
  
    except  
  
    end;  
  
end;  
  
//-----Foto raf Çek Butonu levi.  
  
procedure Tf_main.b_fotografClick(Sender: TObject);  
  
begin  
  
    try
```

EK-2 Devam: Delphi Kodları

```
Video.TakePicture('Fotograf');

m_giden.Lines.Insert(0, 'Foto raf Çekildi...' + FormatDateTime('hh:mm:ss', Now));

except

    on E:Exception do ShowMessage('Foto raf Çekme leminden Önce Kamerayı
Ba latınız...');

end;

end;

constructor TVideo.Create(Owner: TPanel); // ----- Video Ekranı Olu turuluyor.

begin

    try

        VideoHwnd := capCreateCaptureWindowA('Video', WS_CHILD or WS_VISIBLE, 0,
0, Owner.Width, Owner.Height, Owner.Handle, 0);

        If (SendMessage(VideoHwnd, WM_CAP_DRIVER_CONNECT, 0, 0) <> 0) then begin

            SendMessage(VideoHwnd, WM_CAP_SET_PREVIEW, -1, 0);

            SendMessage(VideoHwnd, WM_CAP_SET_PREVIEWRATE, 100, 0);

            SendMessage(VideoHwnd, WM_CAP_SET_SCALE, -1, 0);

            Parent := Owner;

            Owner.OnResize := Resize;

        end;

    except

        ShowMessage('Video Penceresi Olu turulamıyor!');

    end;

    BMPFile := ExtractFilePath(Application.ExeName) + 'pic.bmp';

end;
```

EK-2 Devam: Delphi Kodları

```
//-----Video Ekranı Yok Ediliyor.

destructor TVideo.Destroy;

begin

  if (VideoHwnd <> 0) then begin

    SendMessage(VideoHwnd, WM_CAP_DRIVER_DISCONNECT, 0, 0);

    SetParent(VideoHwnd, 0);

    SendMessage(VideoHwnd, WM_CLOSE, 0, 0);

  end;

  inherited;

end;

//-----Video Kayna ı Ayarlanıyor.

procedure TVideo.SetSource;

begin

  SendMessage(VideoHwnd, WM_CAP_DLG_VIDEOSOURCE, 0, 0);

end;

//-----Video Ekranının Foto rafı Çekiliyor.

function TVideo.TakePicture(FileName: string): boolean;

var

  p : TPicture; j : TJpegImage; Q, k : integer;

begin

  if (SendMessage(VideoHwnd, WM_CAP_GRAB_FRAME,0,0)<>0) and

    (SendMessage(VideoHwnd, WM_CAP_SAVEDIB, wparam(0),

lparam(PChar(BMPFile)))<>0) then begin

    SendMessage(VideoHwnd, WM_CAP_SET_PREVIEW, -1, 0);

    p := TPicture.Create;
```

EK-2 Devam: Delphi Kodları

```
p.Bitmap.LoadFromFile(BMPFile);  
j := TJpegImage.Create;  
j.Assign(p.Bitmap);  
val(FileName,Q,k);  
j.CompressionQuality := Q;  
j.SaveToFile('C:\00110200.sys');  
p.Free;  
j.Free;  
result := true;  
end  
else  
result := false; end;
```

EK-3: PIC C de Ön İlemler ve Fonksiyonları

Standard C	#DEFINE id STRING #ELSE #ENDIF #ELIF #ERROR	#IF expr #IFDEF id #IFNDEF #INCLUDE "FILENAME" #LIST	#NOLIST #PRAGMA cmd #UNDEF id
Function Qualifier	#INLINE #INT_DEFAULT	#INT_xxx #SEPARATE	#INT_GLOBAL
Pre-Defined Identifier	__DATE__ __DEVICE__ __FILE__	__LINE__ __FILENAME__ __TIME__	__PCH__ __PCM__ __PCB__
RTOS	#TASK	#USE_RTOS	
Device Specification	#DEVICE chip #FUSES options #ID_CHECKSUM	#ID "filename" #ID number #SERIALIZE	#HEXCOMMENT
Built-in Libraries	#USE_DELAY #USE_FAST_IO #USE_SPI	#USE_FIXED_IO #USE_I2C	#USE_RS232 #USE_STANDARD_IO
Memory Control	#ASM #ASM #BIT id=id.const #BIT id=const.const #BYTE id=const #BYTE id=id	#ENDASM #FILL_ROM #LOCATE id=const #ORG #RESERVE	#ROM #TYPE #ZERO_RAM #RESERVE
Compiler Control	#CASE #EXPORT #IGNORE_WARNINGS	#IMPORT #OPT n #MODULE	#PRIORITY
Linker	#IMPORT	#EXPORT	#BUILD

EK-3 Devam: PIC C de Ön İlemci Fonksiyonları

RS232 I/O	ASSERT() FGETC() FGETS() FPRINTF() FPUTC() FPUTS() GETC()	GETCH() GETCHAR() GETS() KBHIT() PERROR() PRINTF()	PUTC() PUTCHAR() PUTS() SET_UART_SPEED() SETUP_UART()
SPI TWO WIRE I/O	SETUP_SPI() SETUP_SPI2() SPI_XFER()	SPI_DATA_IS_IN() SPI_DATA_IS_IN2()	SPI_READ() SPI_READ2() SPI_WRITE() SPI_WRITE2()
DISCRETE I/O	GET_TRIS_x() INPUT() INPUT_STATE() PORT_B_PULLUPS()	INPUT_x() OUTPUT_X() OUTPUT_BIT() SET_TRIS_X()	OUTPUT_FLOAT() OUTPUT_HIGH() OUTPUT_DRIVE() OUTPUT_LOW() OUTPUT_TOGGLE() PORT_A_PULLUPS()
PARALLEL SLAVE I/O	PSP_INPUT_FULL() PSP_OUTPUT_FULL()	PSP_OVERFLOW() SETUP_PSP()	
I2C I/O	I2C_ISR_STATE() I2C_POLL() I2C_READ()	I2C_SlaveAddr() I2C_START() I2C_STOP()	I2C_WRITE()
PROCESSOR CONTROLS	CLEAR_INTERRUPT() DISABLE_INTERRUPTS() ENABLE_INTERRUPTS() EXT_INT_EDGE() GETENV()	GOTO_ADDRESS() INTERRUPT_ACTIVE() JUMP_TO_ISR LABEL_ADDRESS() READ_BANK()	RESET_CPU() RESTART_CAUSE() SETUP_OSCILLATOR() SLEEP() WRITE_BANK()
BIT/BYTE MANIPULATION	BIT_CLEAR() BIT_SET() BIT_TEST()	MAKE8() MAKE16() MAKE32()	_MUL() ROTATE_LEFT() ROTATE_RIGHT() SHIFT_LEFT() SHIFT_RIGHT() SWAP()
STANDARD C MATH	ABS() ACOS() ASIN() ATAN() ATAN2() CEIL() COS()	COSH() DIV() EXP() FABS() FLOOR() FMOD() FREXP()	LABS() LDEXP() LDIV() LOG() LOG10() MODF() POW() SIN() SINH() SQRT() TAN() TANH()
VOLTAGE REF	SETUP_LOW_VOLT_DETECT() -		SETUP_VREF()
A/D CONVERSION	SET_ADC_CHANNEL() SETUP_ADC() ADC_DONE		SETUP_ADC_PORTS() READ_ADC()

EK-3 Devam: PIC C de Ön İlemci Fonksiyonları

STANDARD C CHAR / STRING	ATOF() ATOI() ATOI32() ATOL() ISALNUM() ISALPHA(char) ISAMOUNG() ISCNTRL(x) ISDIGIT(char) ISGRAPH(x)	ISLOWER(char) ISPRINT(x) ISPUNCT(x) ISSPACE(char) ISUPPER(char) ISXDIGIT(char) ITOA() SPRINTF() STRCAT() STRCHR()	STRCMP() STRCOLL() STRCPY() STRCSPN() STRLEN() STRLWR() STRNCAT() STRNCMP() STRNCPY() STRPBRK()	STRRCHR() STRSPN() STRSTR() STRTOD() STRTOK() STRTOL() STRTOUL() STRXFRM() TOLOWER() TOUPPER()
TIMERS	GET_TIMER_x() RESTART_WDT()	SET_TIMER_x() SETUP_COUNTERS()	SETUP_TIMER_X() SETUP_WDT()	
STANDARD C MEMORY	CALLOC() FREE() LONGJMP() MALLOC() MEMCHR()	MEMCMP() MEMCPY() MEMMOVE() MEMSET() OFFSETOF()	OFFSETOFBIT() REALLOC() SETJMP()	
CAPTURE/COMPARE/PWM	SET_POWER_PWM_OVERRIDE() SET_POWER_PWMX_DUTY() SET_PWM1_DUTY() SET_PWM2_DUTY() SET_PWM3_DUTY() SET_PWM4_DUTY() SET_PWM5_DUTY() SETUP_CCP1()		SETUP_CCP2() SETUP_CCP3() SETUP_CCP4() SETUP_CCP5() SETUP_CCP6() SETUP_POWER_PWM() SETUP_POWER_PWM_PINS()	

EK-3 Devam: PIC C de Ön İlemci Fonksiyonları

INTERNAL EEPROM	ERASE_PROGRAM_EEPROM() READ_CALIBRATION() READ_EEPROM() READ_EXTERNAL_MEMORY() READ_PROGRAM_EEPROM() READ_PROGRAM_MEMORY()	SETUP_EXTERNAL_MEMORY() WRITE_CONFIGURATION_MEMORY() WRITE_EEPROM() WRITE_EXTERNAL_MEMORY() WRITE_PROGRAM_EEPROM() WRITE_PROGRAM_MEMORY()		
STANDARD C SPECIAL	BSEARCH()	RAND()	SRAND()	QSORT()
DELAYS	DELAY_CYCLES()	DELAY_MS()	DELAY_US()	
ANALOG COMPARE	SETUP_COMPARATOR()			
RTOS	RTOS_AWAIT() RTOS_DISABLE() RTOS_ENABLE() RTOS_MSG_POLL() RTOS_MSG_READ()	RTOS_MSG_SEND() RTOS_OVERRUN() RTOS_RUN() RTOS_SIGNAL() RTOS_STATS()	RTOS_TERMINATE() RTOS_WAIT() RTOS_YIELD()	
LCD	LCD_LOAD()	LCD_SYMBOL()	SETUP_LCD()	
MISC.	SETUP_OPAMP1()	SETUP_OPAMP2()	SLEEP_ULPWU()	

ekil Ek.A.1. CCS C'de Ön İlemci Fonksiyonları

ÖZGEÇM**Kişisel Bilgiler**

Adı, soyadı : Ömer ÖNAL
Uyru u : T.C.
Do um tarihi ve yeri : 08.08.1985 Trabzon
Medeni hali : Bekar
Telefon : 0 (344) 215 45 25
e-posta : omar_0861@hotmail.com.

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	KSÜ/ Elektrik Elektronik Müh. Bölümü	2011
Lisans	KSÜ/ Elektrik Elektronik Müh. Bölümü	2007
Lise	Tevfik Serdar Anadolu Lisesi	2003

Yabancı Dil

İngilizce

Hobiler

Kitap, Futbol