



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK – ELEKTRONİK FAKÜLTESİ
ELEKTRİK MÜHENDİSLİĞİ

EMO PROJE YARIŞMASI

RF ÜZERİNDEN BİLGİSAYAR KONTROLLÜ
FORKLİFT ROBOT

PROJE DANIŞMANI : YRD.DOÇ.DR. ABDULLAH BAL

MURAT AYDIN SIR

METİN UMAR

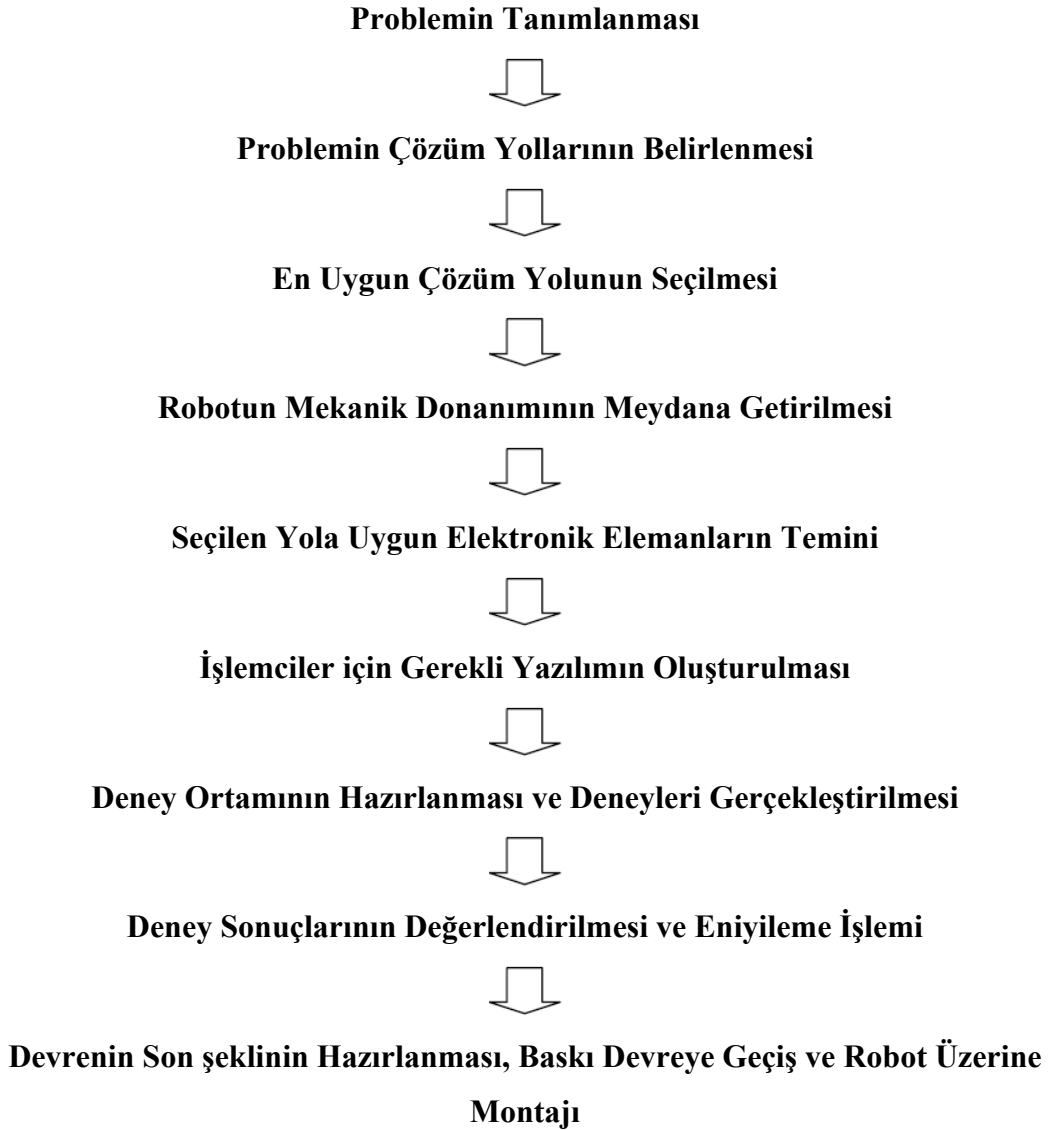
İstanbul , 2007

1. GİRİŞ

1.1 Proje Tanıtımı, Akış Şeması ve Malzeme Listesi

Forklift robotun amacı çeşitli endüstriyel sektörlerde insan sağlığı açısından olumsuz durumlar ve tehlikeler içeren ve mutlaka gerçekleştirilmesi gereken proseslerde iş ve can güvenliğini sağlamaktır. Bu projede robot çalışma sahası dışındaki bir operatör tarafından bilgisayar ve uzaktan kumanda vasıtasıyla kontrol edilmektedir. Böylece tehlikeli ortamdaki insan varlığı gereksinimi ortadan kaldırılmıştır.

AKIŞ DİYAGRAMI





Tez Raporunun Hazırlanması

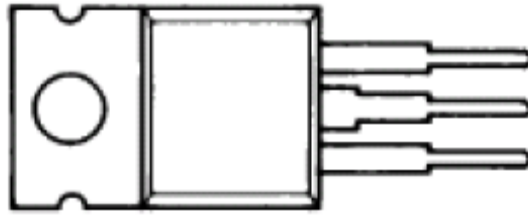
Tablo 1.1 Malzeme Listesi

MALZEME	ADEDİ
PIC16F628A	3
ARX-34 RF ALICI	1
ATX-34 RF VERİCİ	2
LM7805	1
LM7808	1
BD137 TRANSİSTÖR	12
ÇİFT KONTAKLI RÖLE	4
1N4001 DİYOT	12
PC817 OPTOKUPLÖR	4
DİRENÇ (Çeşitli Değerlerde)	22
KAPASİTE (Çeşitli Değerlerde)	20
BUTON	11
LED	3
MAX232 ENTEGRESİ	1
20 MHZ KRİSTAL	1
OYUNCAK EXCAVATOR	1
OYUNCAK FORKLİFT	1
DB9 RS232 KONNEKTÖR	1
BAKIR LEVHA (13x6 cm)	1

1.2 LM7805 ve LM7808 Gerilim Düzenleyicileri

LM78XX serisi pozitif gerilim düzenleyicileri, elektronik elemanların güç tüketimlerinin hızla azalmaya devam ettiği günümüzde, devre tasarımında sıklıkla kullanılmaktadır. Ayrıca LM79XX serisi negatif gerilim düzenleyicileri de bulunmaktadır. Bir LM7805 devresi, girişine uygulanan 15 Volt'luk gerilimi, 5Volt'a düşürüp sabitlerken; LM7905 devresi, aynı gerilim girişine uygulandığında bu gerilimi - 5 Volt'a düşürmektedir. Model olarak; 5, 6, 8, 9, 10, 12, 15, 18, 24 Volt çıkış gerilimi verebilen modeller bulunmaktadır. Ayrıca LM117XX serisi gerilim düzenleyiciler 1.2 Volt' tan 57 Volt'a kadar çıkış gerilimi sağlayabilmektedirler.

Bu projede kullanılan gerilim düzenleyici modeli LM7805 ve LM7808 modelidir. 3 bacağına sahip olan devrede giriş gerilimi 5 Volt ile 24 Volt arası seçilebilmektedir. Soğutucu blok ile devrenin ısınma problemine karşı önlem alınmakla birlikte; 24 Voltu aşan giriş gerilimi değerlerinde, devre aşırı ısınma sorunu ile karşı karşıya kalmaktadır ve soğutucu blok yeterli olamamaktadır. Aşağıda resmi görülen devrenin en üstteki bacağı, düzenlenmiş çıkış bacağıdır. En alttaki bacak pozitif doğru gerilim girişi, ortadaki bacak ise toprağa veya 0 Volt gerilime bağlanması gereken giriştir.



Şekil 1.1 LM7805 ve LM7808 Entegresi

1.3 PIC16F628A

Microchip firmasının üretmiş olduğu 8 bitlik 16F628A denetleyicisi projenin temel elemanıdır. 2. Bölümde denetleyici üzerinde ayrıntılı bir şekilde durulmuştur.

1.4 ARX-34 RF Alıcı

Udea Elektronik firmasının üretmiş olduğu ARX-34 radyo frekanslı alıcı devresi, 433 MHz frekansında çalışmaktadır. Küçük fiziksel boyutu ve düşük güç tüketimi sayesinde uzaktan kontrol sistemleri için kullanıma uygun olan devre, 17.3 cm'lik bir antenle

birlikte en başarılı biçimde çalışabilmektedir. Besleme gerilimi olarak 4.9 Volt ile 5.1 Volt arası bir gerilim seçilmelidir. 5.1 Volt'tan yüksek bir besleme gerilimi uygulanması durumunda devrenin bozulma olasılığı çok yüksektir. Ayrıca besleme geriliminde en fazla 100 mV dalgalanma olması, devrenin öngörüldüğü gibi çalışması açısından önemlidir. 300 baud ile 2400 baud hızları arasında veri transferi yapabilen devre, ev içi uygulamalarda en iyi sonucu 600 baud hızında vermektedir. -10 °C ile +55 °C arasında ortam sıcaklığı çalışabilen ARX-34; 5 mA besleme akımı çekmektedir. Hem sayısal hem de analog çıkışa sahip olmasına rağmen, üretici analog çıkışın test aşamasında olduğunu ifade etmektedir.

1.5 ATX-34 RF Verici

Udea Elektronik (<http://www.udea.com.tr>) firmasının üretmiş olduğu ATX-34 radyo frekanslı verici devresi, 433 MHz frekansında çalışmaktadır. 5 bacaklı yapıya sahip olan devre 17.3 cm'lik bir antenle birlikte en verimli biçimde çalışabilmektedir. Anten boyu hesabı nasıl yapılacağı ise ARX-34 alıcı tanıtılırken yapılmıştır. Besleme gerilimi olarak 5 Volt ile 12 Volt arası bir gerilim seçilmelidir. Besleme gerilimi değeri 12 Volt'a yaklaştıkça, vericinin performansı artmaktadır dolayısıyla daha uzaklara veri iletelebilmektedir. Ayrıca besleme geriliminde en fazla 100 mV dalgalanma olması devrenin öngörüldüğü gibi çalışması açısından önemlidir. 300 baud ile 2400 baud hızları arasında veri transferi yapabilen devre, ev içi uygulamalarda en iyi sonucu 600 baud hızında vermektedir. Hızın ayarlanması yazılım aracılığıyla yapılmıştır ve PIC aracılığıyla vericiye uygulanmıştır. -10 °C ile +55 °C arasında ortam sıcaklığı çalışabilen ATX-34; besleme gerilimi 5 Volt seçildiğinde 6.5 mA besleme akımı çekmektedir. Sadece sayısal girişi bulunan verici devreye, uzağa göndermek istediğimiz veri sayısallaştırılıp bağlanmalıdır.

1.6 Anten Özellikleri

Anten, elektrik işaretlerini (gerilim ve akım) elektromanyetik dalgalara ya da elektromanyetik dalgaları elektrik işaretlerine dönüştürmek için kullanılan araçtır. Udea firmasının üretmiş olduğu UHA-434 anten, ARX-34 alıcı ve ATX-34 verici devreleriyle uyumlu çalışmaktadır. 434 MHz frekansında çalışan anten, 4MHZ bant genişliğine, 50 Ω çıkış empedansına, ve 1.5 gerilim duran dalga oranına sahiptir. Kısaca gerilim duran dalga oranından bahsetmek gerekirse; VSWR genel olarak iletim hattının yük empedansı ile karakteristik empedansı arasındaki uygunsuz eşlemenin bir ölçüsüdür.

Eğer verici uygun bir antenle sonlandırılmamışsa (yani uygun yük) yansıma artacaktır. Dolayısıyla VSWR' de artacaktır ve giden gücün büyük bir kısmı yansıtacaktır. Bu durumda maksimum güç iletilmemiş olacaktır. Bunun için verici uygun empedans değeri olan bir antenle sonlandırılmalıdır, empedans değeri 50 Ohm olmalıdır.

2. PIC MİKRODENETLEYİCİLERİ

2.1 PIC Nedir?

PIC Serisi mikroişlemciler MICROCHIP firması tarafından geliştirilmiş ve üretim amacı çok fonksiyonlu mantıksal uygulamaların hızlı ve ucuz bir mikroişlemci ile yazılım yoluyla karşılanmasıdır. PIC'in kelime anlamı – Peripheral Interface Controller - giriş çıkış işlemcisidir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme ihtiyaç duyulduğu için geliştirilmiştir. Çok geniş bir ürün ailesinin ilk üyesi olan PIC16C54 bu ihtiyacın ilk meyvesidir. PIC işlemcileri RISC benzeri işlemciler olarak anılır. PIC16C54 12 bit komut hafıza genişliği olan 8 bitlik CMOS bir işlemcidir. 18 bacaklı dip kılıfta 13 G / Ç bacağına sahiptir, 20 MHz osilator hızına kadar kullanılabilir ve 33 adet komut içermektedir. 512 byte program EPROM'u ve 25 byte RAM'i bulunmaktadır. Bu hafıza kapasitesi ilk bakışta çok yetersiz gelebilir ama bir RISC işlemci olması birçok işlevin bu kapasitede uygulanmasına olanak vermektedir. PIC serisi tüm işlemciler herhangi bir ek bellek veya giriş / çıkış elemanı gerektirmeden sadece 2 adet kondansatör, 1 adet direnç ve bir kristal ile çalıştırılabilir. Tek bacadan 40 mA akım çekilebilmekte ve tümdevre toplamı olarak 150 mA akım akıtma kapasitesine sahiptir. Tümdevrenin 4 MHz osilator frekansında çektiği akım çalışırken 2 mA, bekleme durumunda ise 20uA kadardır. PIC 16C54'ün fiyatının yaklaşık 2.0 Amerikan Doları civarında olduğu düşünülürse bu işlemcinin avantajı kolayca anlaşılır. PIC 16C54 'un mensup olduğu işlemci ailesi 12Bit Core 16C5X olarak anılır. Bu gruba temel grup adı verilir. Bu ailenin üyesi diğer işlemciler PIC16C57, PIC16C58 ve dünyanın en küçük işlemcisi olarak anılan 8 bacaklı PIC12C508 ve PIC 12C509'dur. Interrupt kapasitesi ilk işlemci ailesi olan 12Bit Core 16C5X ailesinde bulunmamaktadır. Daha sonra üretilen ve orta sınıf olarak tanınan 14Bit Core- 16CXX ailesi birçok açıdan daha yetenekli bir grup işlemcidir. Bu ailenin temel özelliği interrupt kapasitesi ve 14 bitlik komut isleme hafızasıdır. Bu özellikler PIC'i gerçek bir işlemci olmaya ve karmaşık işlemlerde kullanılmaya yatkın hale getirmiştir. PIC16CXX ailesi en geniş ürün yelpazesine sahip ailedir. 16CXX ailesinin en önemli özellikleri seri

olarak devre üstünde dahi programlanması -ki bu özellik PIC16C5x de epey karmaşıktır, paralel programlanabiliyordu- interrupt kabul edebilmesi, 33 G / Ç, A/S Çevirici, USART, I2C, SPI gibi endüstri standardı giriş çıkışları kabul edecek işlemcilere ürün yelpazesinde yer vermesi. PIC 16CXX ailesinin amatör elektronikçiler arasında en çok tanınan ve dünyada üzerinde en çok proje üretilmiş, internetin gözdesi olan bireyi PIC16C84 veya yeni adıyla PIC16F84 dur. PIC 16F84 un bu kadar popüler olması onun çok iyi bir işlemci olmasından ziyade program belleğinin EEPROM - Elektrikle silinip yazılabilen bellek - olmasından kaynaklanmaktadır. Seri olarak dört adet kabloyla programlanması da diğer önemli avantajıdır. Bugüne kadar amatörcü bir işlemciyle uğraşmış herkesin en büyük sıkıntısı EPROM veya EPROM tabanlı işlemcileri programladıktan sonra morötesi ışık kaynağı ile silip tekrar programlamaktır. Bu çok zahmetli ve bir amatör için ekipman gerektiren yöntem olmuştur. Evde üretilmesi zor olan özel bir programlayıcı da madalyonun diğer yüzüdür. PIC16F84, amatörler tarafından internette en bol programlayıcısı bulunan işlemcilerden biridir. EPROM silmek diye bir kavram zaten söz konusu değildir ayrıca PIC16F84 EEPROM barındırdığından mümkün değildir. EEPROM belleği programlayan programlayıcı devre 1 saniye içinde aynı belleği silebilmektedir. Bu özellik tasarımcıya çok hızlı ve defalarca deneyerek program geliştirme avantajını getirmektedir. Bu tasarımcı için çok önemli ve gerekli bir özelliktir. Benzetim programları genel fikir vermek açısından çok yararlı olsalar da ciddi tasarımlarda devreyi fiziksel olarak gerçeklemek, deneyler yapmak kesinlikle gereklidir. Bu denemeleri yaparken işlemciyi devrenizden sökmek dahi gerekmez. Bu tip programlamaya ISP - In System Programming - denmektedir.

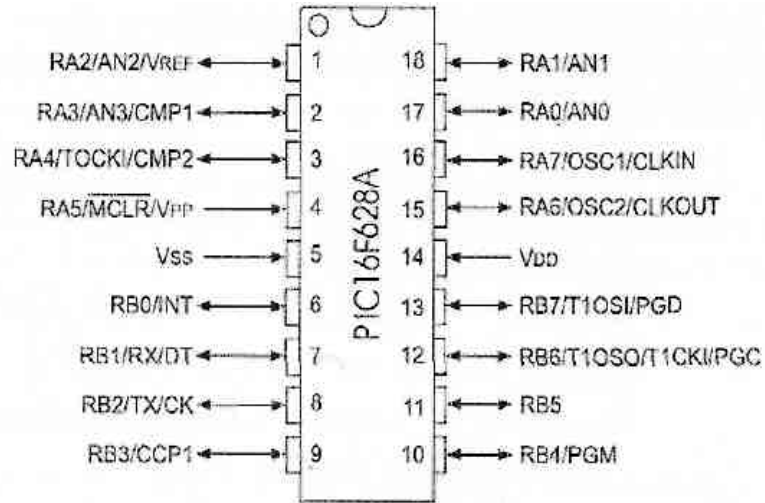
2.2 Neden PIC?

Kod Verimliliği	PIC, Harvard mimarisi temelli 8 bit'lik bir mikrodeneleyicidir. Bu, bellek ve veri için ayrı yerleşik bus'ların bulunduğu anlamına gelir. Böylelikle akış miktarı veriye ve program belleğine simültane erişim sayesinde arttırılmış olur. Geleneksel mikrodeneleyicilerde veri ve programı taşıyan bir tek yerleşik bus bulunur. Bu, PIC ile karşılaştırıldığında işlem hızını en az iki kat yavaşlatır.
Güvenilirlik	Tüm komutlar 12 veya 14 bit'lik bir program bellek sözcüğüne sığar. Yazılımın, programın veri kısmına atlamaya ve veriyi komut gibi çalıştırmasına gerek yoktur. Bu 8 bit'lik bus kullanan ve Harvard mimarisi temelli olmayan mikrodeneleyicilerde gerçekleşmektedir.
Komut Seti	16C5x ailesinde yazılımı yaratmanız için 33 komut öğrenmeniz yeterlidir. 16Cxx araçları içinse bu sayı 35'tir. PIC tarafından kullanılan komutların hepsi yazmaç temellidir ve 16C5x ailesinde 12 bit 16Cxx ailesindeyse 14 bit uzunluğundadır. CALL, GOTO ve bit test eden BTFSS, INCFSSZ gibi komutlar dışında, her bir komut, tek bir çevrimde çalışır. Mikrodeneleyicinin çalışmasını ve işletmesini sağlayan bilgidir. Başarılı bir uygulama veya ürün isteniyorsa yazılım hatasız olmalıdır. Yazılım C, BASIC veya Assembler gibi çeşitli dillerde veya ikilik (binary) olarak yazılabilir.

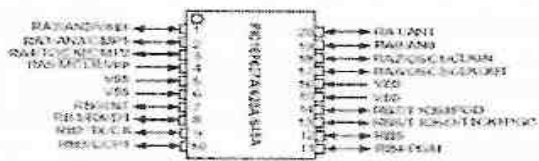
Hız	PIC, osilatör ve yerleşik saat yolu arasına bağlı yerleşik bir 4' lü bölünme' ye sahiptir. Bu, özellikle 4 MHz'lik kristal kullanıldığında komut sürelerinin hesaplanmasında kolaylık sağlar. Her bir komut döngüsü 1 ms' dir. PIC oldukça hızlı bir mikrodur. Örneğin 5 milyon komutluk bir programın, 20 MHz'lik bir kristalle adımlanması yalnız 1 saniye sürer. Bu süre 386 SX 33 işlemcisinin hızının neredeyse 2 katıdır.
Statik İşlem	PIC tamamıyla statik bir mikroişlemcidir. Başka bir deyişle saati durdurduğunuzda, tüm yazmaç içeriği korunur. Pratikte bunu tam olarak gerçekleştirmeniz mümkün değildir. PIC' i uyutma moduna getirdiğinizde, saat durur ve PIC' e uyutma işleminden önce hangi durumda olduğunu size hatırlatacak çeşitli bayraklar kurar. PIC uyuma modunda yalnızca 1 mA'dan küçük değere sahip bekleme akımı çeker.

2.3 PIC16F628A'nın DIŞ GÖRÜNÜŞÜ

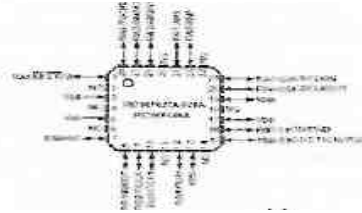
Microchip PIC16F628A'yı üç farklı kılıf tipiyle üretmektedir. Kullanımı en kolay olan PDIP tipi piyasada en kolay bulunandır. Bunun dışında 20 bacaklı SSOP kılıflı ve 28 bacaklı QFN kılıflı tipleri de mevcuttur.



Şekil 2.1 PDIP kılıflı PIC16F628A'nın dış görünüşü



A



B

Şekil 2.2 A -SSOP kılıflı PIC16F628A B-QFN kılıflı PIC16F628A

PIC16F628A ile bellek kapasiteleri hariç diğer tüm özellikleri aynı olan PIC16F627A, PIC16F648A da mevcuttur. Tabloda aralarındaki farklılıklar görülmektedir.

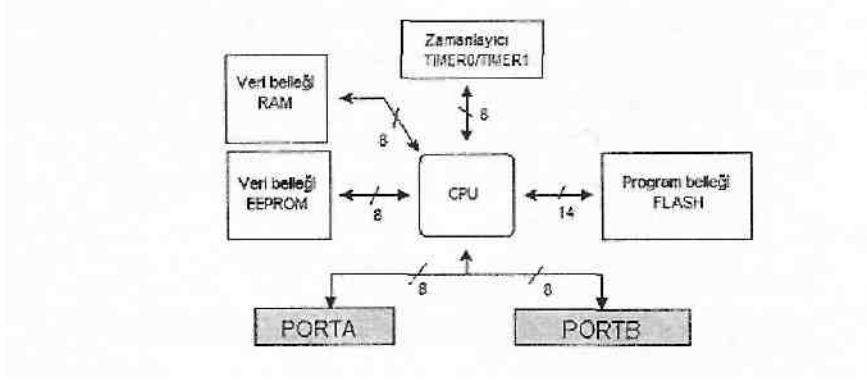
Tablo 2.1 PIC16F627A, PIC16F628A, PIC16F648A arasındaki farklar

Device	Program Memory	Data Memory		I/O	CCP (PWM)	USART	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F627A	1024	224	128	16	1	Y	2	2/1
PIC16F628A	2048	224	128	16	1	Y	2	2/1
PIC16F648A	4096	256	256	16	1	Y	2	2/1

nW (Nano Watt) teknolojisi ile üretildiğinden çok az enerji harcar. Flash belleğe sahip olması nedeniyle clock girişine uygulanan sinyal kesildiğinde registerleri içerisindeki veri aynen kalır. Saat sinyali tekrar verildiğinde PIC içerisindeki program tekrar çalışmaya başlar. RA0-RA7 pin'leri ve RB0-RB7 pin'leri I/O port'larıdır.

2.4 PIC16F628A'nın ÇALIŞMASI

Harvard mimarisi ile üretilen PIC16F628A 8-bit'lik bir RISC işlemcidir. Program belleği ve veri belleğine farklı adres/veri yolu ile ulaşılır. Mikroişlemcinin çalışmasını çizilebilecek en basit bir blok diyagram üzerinde görelim. Bu diyagram tüm PIC'lerin çalışması için geçerlidir.



Şekil 2.3 PIC mikrodeneleyicilerin en basitleştirilmiş blok diyagramı.

CPU: İçerisinde PC-program counter (Program sayıcısı), ALU (Aritmetic-Logic-Unit) bulunan ve diğer bloklar arasındaki çalışmayı organize eden ve flash bellekteki kullanıcı programını çalıştıran kısımdır.

Program belleği (FLASH): Programların saklandığı ve defalarca yazılıp, silinebilen flash bellek.

Veri belleği (EEPROM): PIC üzerindeki gerilim kesilse bile kaybolması istenmeyen verilerin saklandığı bellek. Bu bellek üzerine de defalarca yazma/silme yapılabilir.

Veri belleği (RAM) : Programın çalışması esnasında geçici olarak saklanması gereken veriler için kullanılır. GPR (General Purpose Register- Genel amaçlı saklayıcı) olarak ta adlandırılan bu bellekte programın yazılması esnasında değişkenler içerisinde atanacak veriler bu alanda saklanır.

Bu bellekte ayrıca SPR (Special Function Register- Özel amaçlı saklayıcı) adı verilen saklayıcılarda vardır. Bunlara PIC'in çalışma biçimini yönlendiren veriler yazılır.

PORTA ve PORTB : Mikrodenetleyicinin dışarıya açılan elektronik kapısıdır. PIC'e girilecek ve PIC'ten dışarıya aktarılacak tüm veriler bu bloklar içerisindeki elektronik devreler aracılığı ile yapılır.

Zamanlayıcı (TIMER0/TIMER1/TIMER2) : Programın çalışmasından bağımsız olarak içerisindeki sayı binary olarak artan 8-bit'lik (Not: TIMER1->16-bit) bir saklayıcıdır. Her 4 harici saat sinyalinde bir defa içerisindeki sayı bir defa artarak 255'e ulaştığında tekrar 0'dan başlayarak saymaya devam eder.

Bir mikroişlemcinin çalışması kısaca şöyledir: Program belleğinden CPU tarafından alınan komutun kodu çözülerek işlenir. Komutun yapacağı işleme göre diğer ünitelerden veri okunabilir ya da bu ünitelere veri gönderilebilir. Program komutları gerektiriyorsa zamanlayıcıdan da veri alabilir. Sonuç olarak dış ortama elektronik olarak bağlantısı bulunan PORTA/PORTB'den gönderilen lojik veri mikroişlemcinin kontrol ettiği üniteye veri girişi olur. Bu veri lojik "1" veya "0" olduğu gibi bazı mikroişlemcilerde analog bir gerilim değeri de olabilir. PIC'in dış ortama lojik "1" olarak verdiği 5V luk gerilim altında 25 mA lik akım birçok elektronik devreyi sürmek için yeterli olmayabilir, ihtiyaca göre gerekli yükselteç devreleri kullanılarak güç devreleri de sürülebilir.

Bir Komutun İşlenme Evreleri

PIC içerisinde bir komutun ne şekilde çalıştığını adım adım açıklayabilmek için CPU içerisinde bulunan ve işlevlerinin bilinmesi gereken birkaç ünite daha vardır. Şimdi bunlardan bahsedelim:

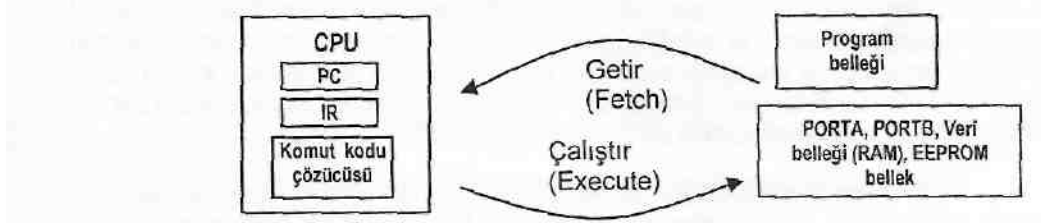
Program Counter (Program sayıcısı): CPU içerisinde bulunan 13-bit'lik bir saklayıcıdır. Program belleğinden okunacak bir sonraki komutun adresi saklanır. Genellikle her bir komut işlendikten sonra içerisindeki adres sayısı bir artar. Ancak bir atlama komutu çalıştığında bu defa komutun belirttiği adres PC içerisine yazılır.

Instruction Register (Komut saklayıcısı) : İşlenecek olan komutun işlenmeden önce belleğe çağrıldığı saklayıcıdır.

Mikrodenetleyicilerde bir komutun işlenmesinde iki evre vardır:

Fetch (Getir) : Program belleğinden alınan bir komut İR (Komut kaydedicisine) getirilir.

Execute (Çalıştır) : Komutun yapacağı işlemin ne olduğuna bağlı olarak işlemin gerçekleştirilmesi.

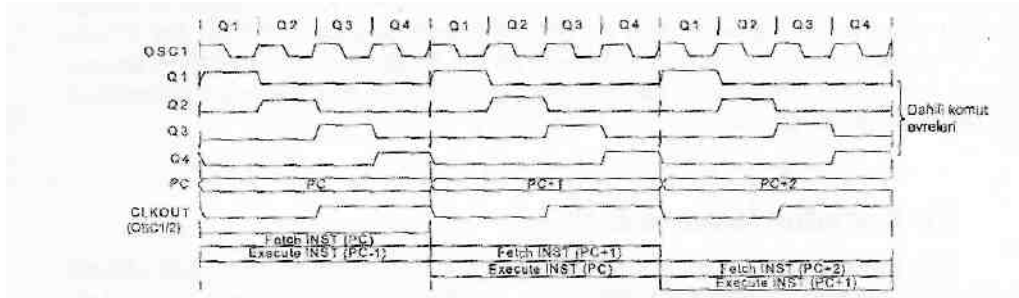


Şekil 2.4 Bir komutun işleme evreleri.

Saat Sinyali ve Komut Evrelerinin Zamanlama Diyagramı

OSC1 ucundan uygulanan saat girişi dahili olarak Q1, Q2, Q3, Q4 olarak adlandıracağımız 4 eşit zaman aralıklı kare dalgaya bölünür. Bu dört saat çevrimi bir komut çevrimini (bir makine çevrimi de denilir) oluşturur. Bu süre içerisinde bir komut çalıştırılır.

Program sayıcı her Q1 evresinde bir artırılır. Komutun program belleğinden getirilmesi (fetch) ve komut sayıcının (IR) içerisine yerleştirilmesi Q4 evresinden olur. Komutun kodunun çözülmesi ve çalıştırılması (execute) Q1 ile Q4 arasındaki evrede olur. Harici OSC1 saat sinyali ve ona bağlı olarak komutun akış evreleri tabloda gösterilmiştir.

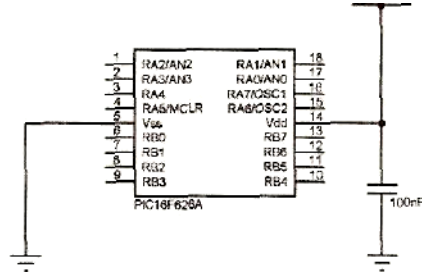


Şekil 2.5 Saat sinyali ve komut evrelerinin zamanlama diyagramı.

2.4.1 BESLEME GERİLİMİ

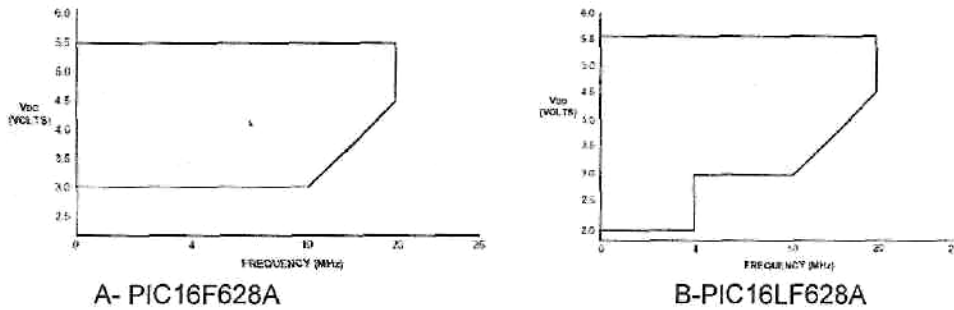
PIC'in çalışması için gerekli olan gerilim 5 ve 14 numaralı pin'lerden uygulanır. 14 numaralı Vdd ucu +5 V'a, 5 numaralı Vss ucu da toprağa bağlanır. PIC'e gerilim

uygulandığında devredeki fazla akım çeken elemanların devreye girmesiyle ya da şebekedeki gerilim dalgalanmaları nedeniyle PIC'in resetlenmesini önlemek amacıyla Vdd ile Vss arasına 0.1 uf lık bir dekuplaj kondansatörü bağlamak gerekir. Bu durumda kararlı bir gerilim değeri elde edilmiş olur.



Şekil 2.6 PIC16F628A'ya besleme geriliminin bağlanması

PIC16F628A besleme geriliminin değeri normal şartlarda (-40° ile $+125^{\circ}$ arasında) 3 - 5.5V arasında seçilebilir. Ancak uygulama devresinin gereksinimine göre PIC16LF628A kullanılarak çalışma gerilimi 2 - 5.5V arasında seçilebilir. Şekil 3-7'deki her iki tip PIC'in -40° ile $+125^{\circ}$ ortam sıcaklığındaki gerilim-frekans eğrisi verilmiştir. Eğrinin taralı olan kısmındaki değerler PIC için izin verilen gerilim ve frekans değerleridir.



Şekil 2.7 PIC16F628A ve PIC16LF628A'nın besleme gerilim-frekans eğrileri.

2.4.2 OSİLATÖR UÇLARI ve ÇEŞİTLERİ

PIC belleğinde bulunan program komutlarının çalıştırılması için bir kare dalga sinyaline gereksinim vardır. Bu sinyale saat (clock-klok olarak okunur) sinyali denir. Saat sinyalinin hızı komutların işleniş hızını da attırır. PIC16F628A'nın saat sinyalinin hızı en fazla 20 MHz olabilir ve giriş 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 şunlardır:

LP - Düşük frekanslı kristal (Low Power).

XT - Kristal veya seramik rezonatör (Xtal).

HS - Yüksek hızlı kristal veya seramik rezonatör (High Speed).

ER - Harici direnç (External Resistor).

EC - Harici saat girişi (External Clock).

INTRC - 4 Mhz lik dahili osilatör (Intemal Resistor-Capacitor). PİC'e bağlanan saat osilatörünün tipi programlama esnasında PIC içerisinde bulunan konfigürasyon bit'lerine yazılmalıdır. Osilatör tipini belirten kodları (ER, XT, HS gibi) kullanarak konfigürasyon bit'lerinin nasıl yazılacağı programlama örnekleri verilirken detaylı olarak incelenecektir.

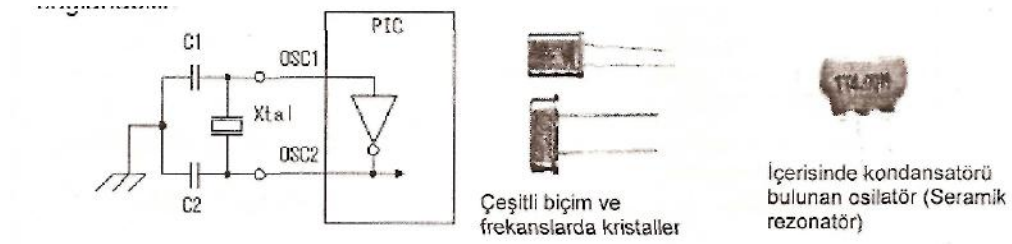
Seçilecek olan osilatör tipi PİC'in kontrol ettiği devrenin hız gereksinimine bağlıdır. Aşağıdaki tablo hangi osilatör tipinin hangi frekans sınırları içerisinde kullanılabileceğini gösterir.

Osilatör tipi	Frekans sınırı
RC	0~4 MHz
LP	5~200 KHz
XT	100 KHz ~ 4 MHz
HS	4~20 MHz

2.4.2.1 LP, XT ve HS Osilatörler

Kristal osilatörler içerisinde osilasyonu sağlayan element kristaldir. PİC'e bağlantılarında harici olarak kondansatör gerektirirler.

Seramik rezonatör osilatörlerde ise osilasyon elemanı seramiktir. Seramik rezonatörler için gereken kondansatörler üretimleri esnasında dahili olarak yerleştirildiklerinden harici kondansatör bağlamak gerekmez. Fiyatları ucuz ve hassastırlar (+/- %1.3). Küçük bir seramik kondansatöre benzeyen rezonatörlerin üç ucu vardır. Bu uçlardan ortadaki toprağa, diğer iki ucu da OSC1 ve OSC2 uçlarına bağlanırlar. Hangi ucun OSCVe bağlanacağı önemli değildir, her ikisi de bağlanabilir.



Şekil 2.8 Kristal osilatör ve devreye bağlantısı.

Kristal kullanılarak yapılan osilatörler zamanlamanın önemli olduğu yerlerde kullanılır. Kristal osilatörlerin kullanıldığı devrelerde kristale bağlanacak kondansatörün seçimine özen göstermek gerekir. Aşağıda hangi frekansta kaç pF lık kondansatör kullanılacağını gösteren tablo görülmektedir.

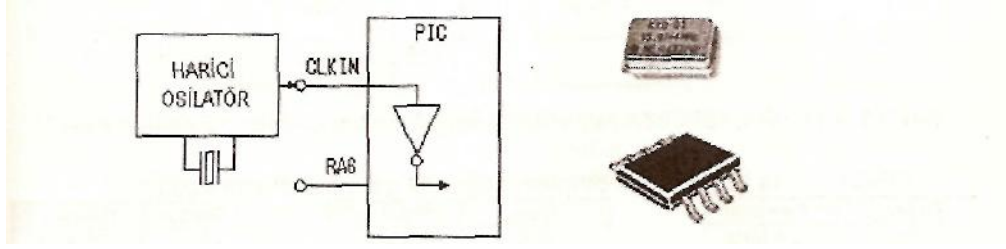
Tablo 2.2 LP, XT ve HS kristallere bağlanması gereken kondansatör değerleri.

Mode	Freq	OSC1(C1)	OSC2(C2)
LP	32 kHz	15 - 30 pF	15 - 30 pF
	200 kHz	0 - 15 pF	0 - 15 pF
XT	100 kHz	68 - 150 pF	150 - 200 pF
	2 MHz	15 - 30 pF	15 - 30 pF
	4 MHz	15 - 30 pF	15 - 30 pF
HS	8 MHz	15 - 30 pF	15 - 30 pF
	10 MHz	15 - 30 pF	15 - 30 pF
	20 MHz	15 - 30 pF	15 - 30 pF

2.4.2.2 EC Osilatör

Kristal kontrollü saat osilatörleri zamanlamanın çok hassas olması gerektiğinde kullanılır. Bu tip saat osilatörleri metal bir kutu veya normal bir entegre devre görünümündedir. Entegre görünümünde olan osilatörlere (Real-time-clock) gerçek zamanlı saat adı da verilmektedir. Genellikle programlanabilir olduklarından aynı chip'ten farklı frekansta çok hassas saat sinyali alınabilir. Bu tip osilatörlere kondansatör bağlantısı gerekmez.

Bu tip osilatör kullanmanın en önemli özelliği, RA6 ucunun normal I/O ucu olarak kullanılabilmesidir.



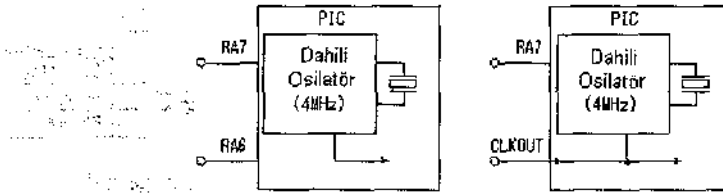
Şekil 2.9 Harici osilatörler ve devreye bağlantısı.

2.4.2.3 IntRC Osilatör

PIC16F628A, 4 Mhz lik dahili bir osilatöre sahiptir. RA6 ve RA7 uçları normal I/O ucu olarak kullanılmak istenildiğinde bu osilatör tipi seçilir. Bu durumda PIC'e hiçbir harici osilatör elemanı bağlamak gerekmez. PIC'in programlanması esnasında yazılan konfigürasyon bilgileri ile bu osilatör iki farklı şekilde çalıştırılabilir.

1. RA6 ve RA7 uçlarının her ikisinin birlikte I/O ucu olarak kullanılabilir.
2. RA7, I/O ucu olarak RA6 da CLOCKOUT ucu olarak kullanılabilir.

CLOCKOUT ucundan dahili osilatör frekansının $\frac{1}{4}$ 'i alınır. Bu saat çıkışı PIC ile eşzamanlı çalışması istenen harici donanımlarda saat sinyali girişi olarak kullanılabilir.

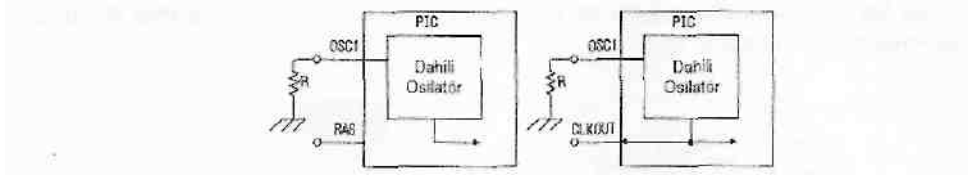


Şekil 2.10 PIC16F628A'nın dahili osilatörü.

2.4.2.4 ER Osilatörü

Bu osilatör tipinde dahili osilatöre dışardan sadece bir direnç bağlanır. PIC'e yazılacak konfigürasyon bilgilerine bağlı olarak iki farklı biçimde çalışması sağlanabilir.

1. OSC1'e direnç bağlanarak RA6 ucu I/O olarak kullanılabilir.
2. OSC1'e direnç bağlanarak RA6 ucundan OSC1/4 değerinde saat sinyali alınır.



Şekil 2.11 PIC16F628A'nın osilatör sinyalinin harici dirençle sağlanması.

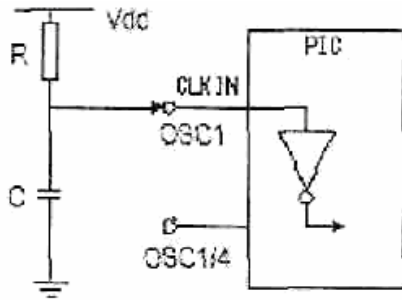
Tablo 2.3 PIC16F628A'ya harici olarak bağlanan dirençlerle elde edilecek frekans değerleri.

OSC1 ucuna bağlanacak direncin değerine göre frekans ta değişir.

Direnç	Frekans	Direnç	Frekans	Direnç	Frekans
0	10.4 MHz	20K	5.3 Mhz	220K	800 KHz
1K	10.0 MHz	47K	3 Mhz	470K	300 KHz
10K	7.4 MHz	100K	1.6 Mhz	1M	200 KHz

2.4.2.5 RC Saat Osilatörü

PIC16F628A'da bu tip osilatörün kullanılmasına gerek duyulmaz. Ancak dahili osilatörü bulunmayan diğer PIC'lerde ve zamanlamanın çok hassas olması gerekmeyen durumlarda kullanılır. Belirlenen değerden yaklaşık %20 sapma gösterebilirler. Bir direnç ve kondansatörden oluşan bu osilatörün maliyeti oldukça düşüktür. OSC1 ucundan uygulanan saat frekansı R ve C değerlerine bağlıdır. Şekil 3-12'de RC osilatörün saat girişine bağlantısı ve çeşitli R, C değerlerinde elde edilen osilatör frekansları örnek olarak verilmiştir.



R	C	f _{osc} (yaklaşık)
10 K	20 pf	625 KHz
10 K	220 pf	80 KHz
10 K	0.1 µf	85 Hz

Şekil 2.12 RC Osilatörün PIC'e bağlantısı ve örnek R ve C değerleri

OSC1 ucundan uygulanan harici saat frekansının 1/4'ü OSC2 ucunda görülür. Bu saat frekansı istenirse devrede kullanılan diğer bir elemanı sürmek için kullanılabilir.

2.4.3 Reset Uçları ve Reset Devresi

Bir PIC'in reset (yeniden başla) yapılması, çalışan programın kesilerek başlangıç noktasından itibaren tekrar çalışmaya zorlanması demektir. Mikrodenetleyici uygulamalarında çok gerekli olan bu işlemlerin bazıları donanımsal, bazıları da yazılımsal olarak yapılır. PIC'e dışardan gerilim uygulamak suretiyle yapılan reset donanımsal resettir.

PIC16F628A'e birbirinden farklı kaynaktan 6 reset (yeniden başlatma) uygulanabilir. Bu uygulamaların hangisinin aktif olacağı PIC'in içerisindeki bir register (saklayıcı) içerisine yazılan veri bit'leri ile seçilir. Configuration (yapılanma) saklayıcısından bahsedilirken bu ayarların nasıl yapılacağı daha geniş olarak açıklanacaktır.

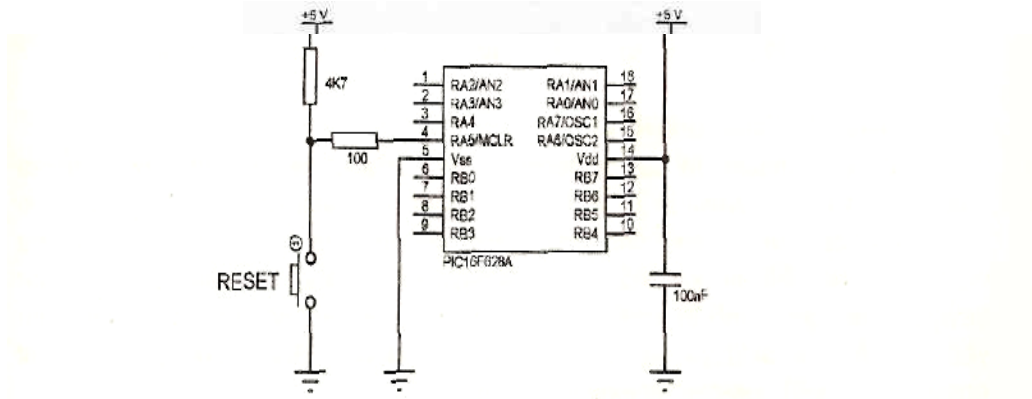
- a) Power-On Reset (POR) _____
- b) Normal çalışma esnasında MCLR resetlemesi.
- c) Sleep (uyuma) modunda çalışırken MCLR resetlemesi.
- d) Normal çalışma esnasında VVDT resetlemesi.
- e) Sleep (uyuma) modunda VVDT resetlemesi.
- f) Brown-Out Reset (BOR)

Power On Reset (POR)

PIC16F628A'nın besleme uçlarına gerilim uygulandıktan sonra PIC uçlarındaki gerilimin uygun seviyeye çıkmasına kadar reset durumunda bekleten dahili (on-chip) bir reset devresi vardır. Bu devreye POR (Power-On-Reset) devresi denir. PIC uçlarındaki gerilim uygun seviyeye ulaştığında program başlangıç adresinden itibaren normal çalışmasına başlar.

Normal Çalışma Esnasında MCLR Resetlemesi

MCLR ucu, kullanıcının programı kesip, kasti olarak başlangıca döndürebilmesi için kullanılır. PIC'in 4 numaralı MCLR ucuna uygulanan gerilim 0 V olunca programın çalışması başlangıç adresine döner. Programın ilk adresten itibaren tekrar çalışabilmesi için reset ucuna uygulanan gerilimin yeniden +5 V olması gerekir. Şekil 3-13'de bir butona basıldığında PIC'i resetleyen devre görülmektedir.



Şekil 2.13 PIC16F628A'nın donanımsal (MCLR) reset devresi.

MCLR ucuna uygulanacak olan maksimum gerilim $-0.3V$ ile $14V$ arasındadır. Herhangi bir nedenle MCLR ucuna uygulanan gerilim V_{ss} geriliminin aşırı düşüşlerinde akım 80 mA 'in üzerine çıkabilir. Bu durumda programda kilitlenmeler olabilir. Bu nedenle MCLR ucunu direk olarak V_{ss} 'ye bağlamak yerine $50\text{--}100\Omega$ luk bir dirençle bağlamak, emniyet açısından gereklidir.

Sleep Modunda Çalışırken MCLR Resetlemesi

PIC uyku modunda çalışırken MCLR ucundan uygulanacak OV, PIC'in resetlenmesini sağlar. Uyku modu, PIC'in minimum güç çekme durumudur ve bu anda program çalışmaz. İlerdeki bölümlerde daha detaylı bilgi bulacaksınız.

Normal Çalışma Esnasında WDT Resetlemesi

PIC'in normal olarak çalışırken WDT (Watch Dog Timer) sayıcısından gelen bir kesme ile resetlenmesidir. PIC'e besleme gerilimi verildiği sürece çalışan bir RC osilatörün tetiklediği sayıcıdır. PIC'in yapması gereken bir işlemin istenilen sürede yapılamaması neticesinde sistemde bir aksaklık olmasın diye programı devam ettirmeden reset yaptırır.

Sleep Modunda WDT Resetlemesi

PIC uyuma modunda çalışırken VVDT sayıcısından gelen bir kesme sinyali ile resetlenmesidir.

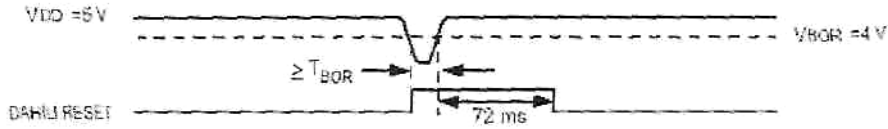
Brown Out Reset (BOR)

PIC16F628A içerisinde (on-chip) bulunan reset devresidir. PIC'in yapılanma (configuration) saklayıcısına yazılan veri bit'leri ile aktif hale getirilebilir. Bu devrenin amacı, PIC'in besleme geriliminin belirli bir süre boyunca ($T_{BOR}=100 \mu s$) belirli bir değerin ($V_{BOR}=4V$) altına düştüğünde PIC'i resetlemektir. Ancak PIC'in ne kadar süreyle reset durumunda kalacağını yine PIC içerisinde bulunan PWRT (Power Up Timer) belirler. Şimdi PWRT sayıcısından kısaca bahsedelim.

Power Up Timer (PWRT)

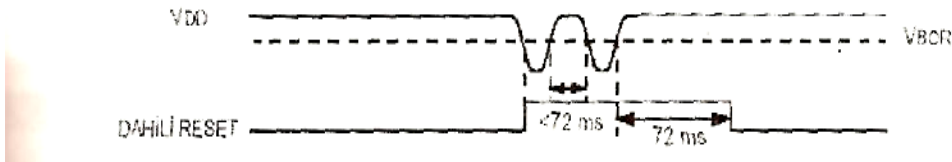
PIC içerisindeki bu sayıcı dahili bir RC osilatör ile tetiklenir. BOR reseti oluştuğunda reset süresinin 72 ms daha devam etmesini sağlar. PVVRT sayıcısı aynı zamanda POR (Power On Reset) reset durumunda da PIC'in resette kalma süresini belirler. PIC'in uçlarındaki gerilim seviyesinin kabul edilebilir ($V_{BOR}=4V$) bir seviyeye gelince 72 ms lik süre başlar. Yapılanma registerine yazılacak 1-bit'lik bir veri ile PWRT'yi aktif ya da pasif duruma geçirilebilir. BOR reset durumu aktif edildiğinde PWRT sayıcısı da aktif edilmelidir.

Şimdi BOR reset durumunu bir zaman-gerilim eğrisiyle inceleyelim:



Şekil 2.14 Besleme geriliminin 100 μs 'den fazla 4V'un altına düşmesi ve BOR reset durumunu gösteren zaman- gerilim eğrisi.

Şekil 2.14'de Vdd gerilimi $T_{BOR}=100 \mu s$ den daha uzun süre $V_{BOR}=4V$ gerilimin altına düşmektedir. Reset durumu oluştuğu anda PVVRT sayıcısı çalışarak PIC'i 72mS süreyle reset durumunda tutmaktadır.



Şekil 2.15 : BOR reset durumunda 73 mS'lik süre bitmeden gerilimin tekrar 4V'un altına düşmesi durumu.

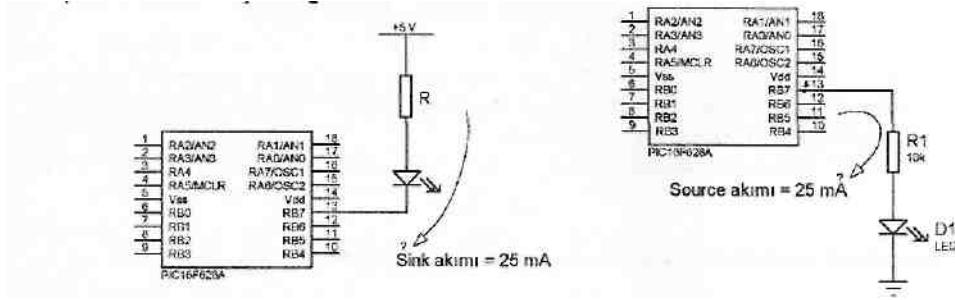
Şekil 2.12'de gerilim ilk V_{BOR} gerilimin altına düşmesiyle PWRT çalışmaya başlar. Ancak 72 mS lik süre bitmeden tekrar V_{BOR} altına düşmektedir. Bu durumda PWRT tekrar çalışmaya başlar ve PIC'in resette kalma süresi daha da uzar. Sürenin bu şekilde uzatılması genellikle ani gerilim düşümleri yaşanan uygulama devreleri için çok uygundur. Örneğin bir bilgisayar yazıcısının motorunu çalıştıran PIC uygulamasını düşünelim. Kısa süreli gerilim dalgalanmalarında motorun çalışmaya başlayıp durması ve tekrar çalışıp durması daha büyük donanımsal arızalara neden olabilir.

2.4.4 I/O (GİRİŞ/ÇIKIŞ) PORT'LARI

Bu port'lardan girilen dijital sinyaller vasıtasıyla PIC içerisinde çalışan programa veri girilmiş olur. Program verileri değerlendirerek port'ları kullanmak suretiyle dış ortama dijital sinyaller gönderir. Dijital sinyallerin gerilim değeri ya 5V ya da 0V'tur. PIC 16F628A'nın dış ortama verilebileceği maksimum akım 25 mA dir. Bazı PIC'ler de (örneğin PIC16F84) bu akım 20 mA olabilir. Dış ortamdan PIC16F628A'ya doğru akabilecek maksimum akım da 25 mA dir. 25 mA lik akım motor, güçlü lamba gibi fazla akım çekmesi gereken alıcıları çalıştırmak için yeterli değildir. Ancak aşağıda sıralanan bazı alıcıları direk olarak çalıştırmak mümkündür.

1. LED : Tam parlaklık verebilmesi için 25 mA lik akım yeterlidir.
2. BUZZER : Çok küçük akım çektikleri için direk olarak sürülebilir.
3. RÖLE : Düşük akımla çalışan bazı röleler. 5V'ta 25 mA bobin akımı yeterli olanlar.
4. LCD display : Hiçbir yükseltme devresi gerektirmeden sürülebilirler.
5. 7 SEGMENT DISPLAY : Her segment'e bir port ucu bağlanarak sürülür.

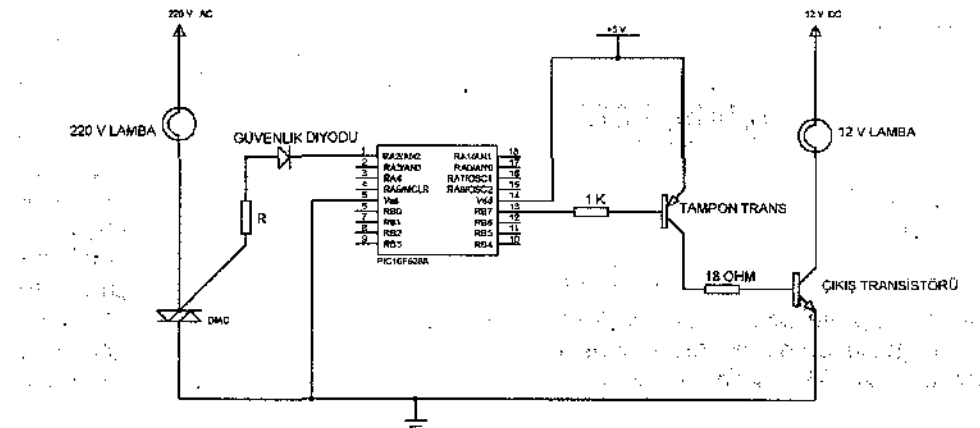
Dış ortama gönderilen bu sinyallerin akımı yeterli olmadığı durumda yükselteç devreleri, transistörler vasıtasıyla yükselterek kumanda edilecek cihaza uygulanır. Hatırlatma amacıyla sink ve source akımlarının ne olduğundan bahsedelim. Sink akımı, gerilim kaynağından çıkış port'una doğru akan akıma, source akımı ise I/O pin'inden GND'ye doğru akan akıma denir.



Şekil 2.16 Sink ve source akımları.

2.4.5 Yüksek Akımlı Yüklen Sürmek

I/O port'larından geçebilecek 25 mA lik akımı bir sınırlama direnci bağlamak suretiyle LED'leri doğrudan sürebilir. Bu akımlar aynı zamanda LCD, lojik entegre ve hatta 220 V luk şehir şebekesine bağlı bir lambayı kontrol eden triyakı bile tetiklemeye yeterlidir. Çıkış akımı yetmediği durumda yükselteç devreleri kullanarak daha yüksek akımlara kumanda etmek mümkün olabilir.



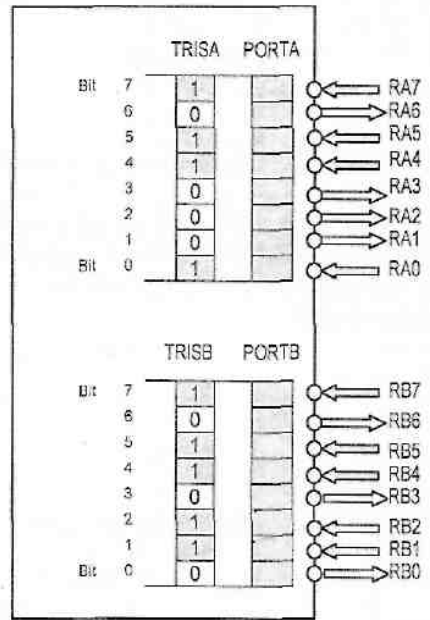
Şekil 2.17 Yüksek akım çeken alıcılar için çeşitli yükseltme devreleri.

PIC16F628A'nın 16 adet giriş/çıkış ucu vardır. Bunlardan 8 tanesi A port'unda (RA0-RA7), 8 tanesine de B port'undadır (RB0-RB7). RA5/MCLR/Vpp ucunun dışındaki tüm uçlar veri girişi ya da çıkışı olarak yapılandırılabilir. RA5 ucu sadece giriş olarak yapılandırılabilir. I/O ucu olarak kullanılamayan sadece Vdd(5V) ve Vss (GND) vardır. Tüm PIC mikrodenetleyicilerde olduğu gibi PIC16F628A'da da bir ucu yapılandırmak suretiyle aynı uca farklı işlemler yaptırılabilir. Bu da PIC'Ierin daha az bacak sayısı ile daha fazla özelliğe sahip olması demektir.

2.4.6 A Port'u ve Özellikleri

PORTA'nın detaylı elektronik yapısı microchip firmasının Web sitesinde yayınladığı datasheet'lerde (veri katalogu) ayrıntılı olarak bulunabilir. PORTA'nın özelliklerini verebilmek için yapısındaki elektronik detaylara girmeden blok olarak düşünersek, özelliklerini aşağıdaki gibi verebiliriz:

1. A port'u içerisinde 8-bit uzunluğunda adına PORTA registeri denilen bir veri tutucu (latch) vardır.
2. PORTA saklayıcısına bir veri gönderildiğinde içerisindeki silinmedikçe veya yeni bir veri yazılmadıkça aynen saklanır. Bu kaydediciye dışarı gönderilen veya dışarıdan alınan veriler tutulur.



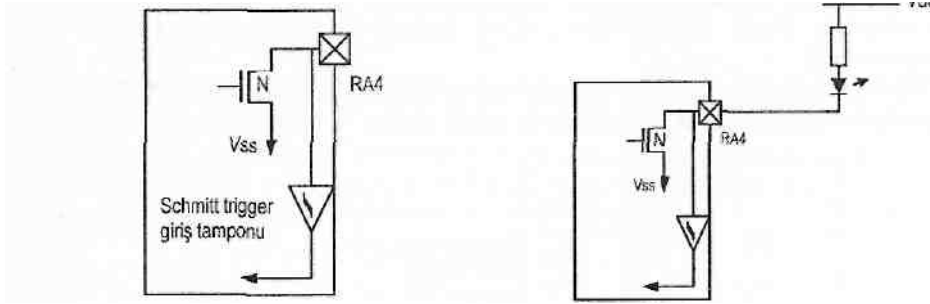
Sekil 2. 18 A ve B portlarının TRISA ve TRISB saklayıcıları ile yönlendirilmesi.

3. A port'unun yapılandırılmasını sağlayan ikinci bir saklayıcıya da TRISA saklayıcısı denir. Bu saklayıcı RAM belleği içerisinde özel amaçlı bir saklayıcıdır.
 4. TRISA saklayıcısı içerisine yazılan veri bit'leriyle ("1" ya da "0") PORTA'nın uçlarının yönlendirilmesi (giriş ya da çıkış) yapılır.
- Şekil 3-18'de de görüldüğü gibi 8-bit'lik TRISA saklayıcının ilgili bit'inin "0" yazılmasıyla o ucun çıkış, "1" yazılmasıyla o ucun giriş olarak yönlendirilmesi sağlanır.

PORTA'nın tüm uçları **Schmitt Trigger (ST)** yapıdadır. Schmitt trigger yapı ile TTL (Transistor-Transistor-Logic) yapı arasındaki farklar şunlardır:

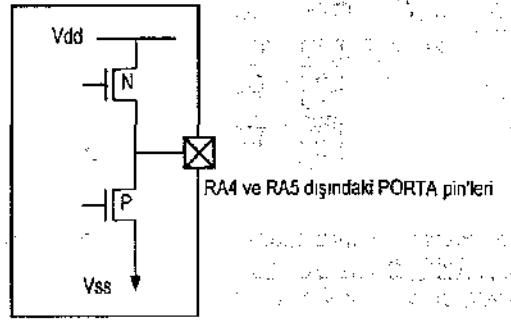
1. ST yapılı uçlarda "0" dan "1"e veya "1" den "0" a geçiş süresi daha kısadır. Bu nedenle daha yüksek frekanslı veri girişi/çıkışı yapılmasına uygundur.
2. ST yapılı uçlarda "0" ve "1" i belirleyen gerilim aralıkları birbirinde uzaktır. Bir uçtan girilen 0-1V arasındaki gerilimleri "0" V, 4-5.5 V arasındaki gerilimler-» "1" olarak algılanır.
3. TTL yapılı uçlarda "0" dan "1"e veya "1" den "0" a geçiş süresi daha uzundur.
4. TTL yapılı uçlarda "0" ve "1" i belirleyen gerilim aralıkları birbirine daha yakındır. Bir uçtan girilen 0-2V arasındaki gerilimler-» "0"V, 2-5.5 V arasındaki gerilimler-» "1" olarak algılanır.

RA4 ucu öpen drain (açık kolektör) özelliği nedeniyle çıkış olarak kullanılmak istenildiğinde sink akımı yönünde akım geçirilmelidir. Örneğin bir LED'i sürmek için Vdd besleme gerilimine seri bir dirençle birlikte bağlanmalıdır.



Şekil 2.19 Açık kolektör özellikli RA4 ucunun yapısı ve çıkış olarak kullanılması.

RA4 dışında kalan tüm PORTA uçları dijital çıkış olarak yönlendirildiklerinde CMOS çıkışa sahiptir. CMOS yapılı PORTA uçlarının yapısı şekil 3-20'de görülmektedir. N-kanal CMOS iletimdeyken port çıkış ucu 5V, P-kanal CMOS iletimdeyken de çıkış ucu 0V seviyesindedir.



Şekil 2. 20 CMOS yapıdaki PortA uçlarının yapısı.

RA6 ve RA7 Uçları

RA6 ve RA7 uçları PIC'in osilatör girişi olarak kullanılabildiği gibi normal I/O ucu olarak kullanılabilir. Bunun için gerekli kurulumlar konfigürasyon bitleri ile yapılır.

MCLR Ucu

16F628A'nın 4. pin'indeki RA5/MCLR ucu istenirse normal giriş port'u olarak kullanılabilir. Bunun için gerekli kurulumlar konfigürasyon bit'leri ile yapılır.

Tablo 2.4 PortA'nın uçları ve farklı kullanım özellikleri

Pin no	Pin adı	İşlevi	Açıklama
17	RA0/AN0	RA0	İki yönlü dijital I/O port'u
		AN0	Analog karşılaştırmacı girişi
18	RA1/AN1	RA1	İki yönlü dijital I/O port'u
		AN1	Analog karşılaştırmacı girişi
1	RA2/AN2/ V _{REF}	RA2	İki yönlü dijital I/O port'u
		AN2	Analog karşılaştırmacı girişi
		V _{REF}	V _{REF} çıkışı
2	RA3/AN3/CMP1	RA3	İki yönlü dijital I/O port'u.
		AN3	Analog karşılaştırmacı girişi
		CMP1	Karşılaştırmacı 1 çıkışı
3	RA4/T0CKI/CMP2	RA4	İki yönlü dijital I/O port'u. Çıkış olarak kullanıldığında open drain (açık kolektör) tiptedir.
		T0CKI	TMR0 harici clock girişi veya karşılaştırmacı çıkışı
		CMP2	Karşılaştırmacı 2 çıkışı
4	RA5/MCLR/Vpp	RA5	Dijital giriş port'u
		MCLR	Reset giriş ucu. MCLR olarak yapılandırıldığında bu pin'den girilen 0V PIC'i reset eder. MCLR/Vpp girişi normal çalışma esnasında Vdd gerilimini geçmemelidir.
		Vpp	Programlama modunda 13 V luk gerilimi girişi
15	RA6/OSC2/CLKOUT	RA6	İki yönlü dijital I/O port'u.
		OSC2	LP, XT ve HP osilatör modlarında kristalin bir ucu bağlanır.
		CLKOUT	ER osilatörü ve dahili RC osilatörü kullanıldığında OSC1 frekansının ¼'ünün alındığı pin.
16	RA7/OSC1/CLKIN	RA7	İki yönlü dijital I/O port'u.
		OSC1	LP, XT ve HP osilatör modlarında kristalin bir ucu bağlanır. Harici osilatör girişi
		CLKIN	Harici saat kaynağı girişi.

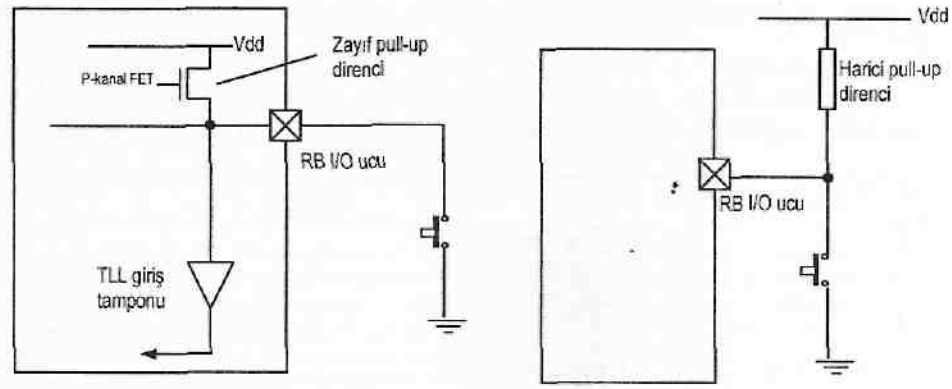
2.4.7 B Port'u ve Özellikleri

B port'unun çıkışında FET kullanılmayıp normal dijital kapı devreleriyle sürülmüştür. Bu nedenle PORTB dijital giriş/çıkış port'u olarak kullanıldığında TTL yapıya sahip olmasıyla PORTA'dan farklıdır. PORTB'nin tüm çıkışları bir FET ile pull-up yapılmıştır. Pull-up, giriş ucunu Vdd potansiyeline çekmek demektir. Normal olarak bir giriş ucunu Vdd seviyesine çekmek için bir direnç kullanılır. Aynı görevi PIC16F628A içerisinde port uçlarına bağlı bir FET yapmaktadır. FET iletimde değilken yüksek drain-source direnci vasıtasıyla port uçlarını zayıf pull-up (weak pull-up) yapmaktadır.

PORTB'nin bir ucu giriş olarak yönlendirildiğinde zayıf pull-up direncinin etkisi chip tarafından otomatik olarak aktif yapılır. Çıkış olarak yönlendirildiğinde ise pull-up direnci pasif yapılır. PORTB'nin bu özelliğinden dolayı çoğu uygulamalarda harici olarak bağlanan pull-up direncine gerek duyulmaz. PIC'e enerji verildiğinde (Power-on-reset) ise tüm dahili pull-up dirençleri iptal edilir.

PORTB'nin bir ucu giriş olarak yönlendirildiğinde zayıf pull-up direncinin etkisi chip tarafından otomatik olarak aktif yapılır. Çıkış olarak yönlendirildiğinde ise pull-up direnci pasif yapılır. PORTB'nin bu özelliğinden dolayı çoğu uygulamalarda harici olarak bağlanan pull-up direncine gerek duyulmaz. PIC'e enerji verildiğinde (Power-on-reset) ise tüm dahili pull-up dirençleri iptal edilir.

harici olarak bağlanan pull-up direncine gerek duyulmaz. PIC'e enerji verildiğinde (Power-on-reset) ise tüm dahili pull-up dirençleri iptal edilir.



Şekil 2.21 TTL özellikteki PortB uçları ve zayıf pull-up yapısı.

PORTB uçları değişik biçimde yapılandırılarak USART, CCP, harici kesme girişi, TMR1 saat girişi gibi özelliklerde de kullanılabilir. Tablo 3-5'de uçların hangi işlevleri olduğu ve açıklamalar bulunmaktadır. Programlama detayları anlatılırken bu uçların yapılandırma saklayıcılarıyla ne şekilde işlevlerinin değiştirilebileceği daha detaylı olarak verilecektir.

Tablo 2.5 PortB'nin uçları ve farklı kullanım özellikleri.

Pin no	Pin adı	İşlevi	Açıklama
6	RB0 / INT	RB0	İki yönlü dijital I/O port'u. Yazılımla zayıf pull-up yapılabilir.
		INT	Harici kesme girişi.
7	RB1/RX/DT	RB1	İki yönlü dijital I/O port'u. Yazılımla zayıf pull-up yapılabilir.
		RX	USART veri alma pin'i
		DT	Senkron (eşzamanlı) data I/O pin'i
8	RB2/TX/CK	RB2	İki yönlü dijital I/O port'u. Yazılımla zayıf pull-up yapılabilir.
		TX	USART veri gönderme pin'i
		CK	Senkron clock I/O pin'i.
9	RB3/CCP1	RB3	Dijital I/O port'u. Yazılımla zayıf pull-up yapılabilir.
		CCP1	Capture / Compare / PWM / I/O
10	RB4/PGM	RB4	İki yönlü dijital I/O port'u. Pin'deki seviye değişikliğinde kesme oluşturur. Yazılımla zayıf pull-up yapılabilir.
		PGM	Düşük gerilimle programlama modunda giriş pin'i.
11	RB5	RB5	İki yönlü dijital I/O port'u. Pin'deki seviye değişikliğinde kesme oluşturur. Yazılımla zayıf pull-up yapılabilir.
		RB6	Dijital I/O port'u // Timer1 clock girişi
12	RB6/T1OSO/T1CKI/PGC	T1OSO	Timer1 osilatör çıkışı.
		T1CKI	Timer1 clock girişi
		PGC	Programlama modunda clock girişi.
13	RB7/T1OSI/PGD	RB7	İki yönlü dijital I/O port'u. Pin'deki seviye değişikliğinde kesme oluşturur. Yazılımla zayıf pull-up yapılabilir.
		T1OSI	Timer1 osilatör girişi.
		PGD	Programlama modunda data girişi/çıkışı ucu.

2.4.8 PIC16F628A'nın BELLEĞİ

Harvard mimarisine sahip olan PIC'lerin program ve RAM belleği olmak üzere iki ayrı bellek bloğundan oluştuğunu mikrodenetleyici mimarilerini anlatırken bahsetmiştik. PIC16F628A, 2K'lık FLASH program belleğine sahiptir. Bu belleğin hücrelerine erişmek için adreslerini saklayan ve PIC'e uygulanan her 4 saat sinyalinde bir defa içerisindeki sayı bir artan saklayıcı bulunduğunu ve adına program sayıcısı (PC-Program Counter) denildiğini bu bölümün başlarında anlatmıştık. PIC16F628A'nın 13-bit uzunluğunda program sayıcısı vardır. 13 bit'lik bu sayıcı 8Kx14 program hücrelerini adresleyebilecek kapasiteye sahiptir ($2^{13}=8192$ bayt=8K).

2.4.8.1 Program Belleği

PIC16F628A'nın 2 Kbyte'lık program belleğinin içerisinde 14 bit uzunluğundaki program komutları saklanır. Program belleği flash olmasına rağmen, programın çalışması esnasında veri

PIC16F628A'nın 00h~1FFh aralığındaki veri belleği 4 parçaya ayrılmıştır. Bu parçaların her biri BANK olarak adlandırılır. Veri belleği içerisinde 8-bit'lik özel amaçlı saklayıcılar (SFR-Special Function Registers) ve genel amaçlı saklayıcılar (GPR-General Purpose Registers) bulunmaktadır.

Özel Amaçlı Saklayıcılar

PIC'in CPU'sunun ve çevresel arabirimlerin (PORTA, PORTB, USART gibi) istenilen biçimde çalışmasını sağlamak amacıyla bazı yönlendirme veri bit'lerinin yazıldığı saklayıcılardır. Her BANK'ın ilk 32 adresi bu saklayıcılara ayrılmıştır. Statik RAM (SRAM) yapısındadırlar. Programlama örnekleri verildikçe sırası geldiğinde bu saklayıcılardan ayrıntılı olarak bahsedilecektir.

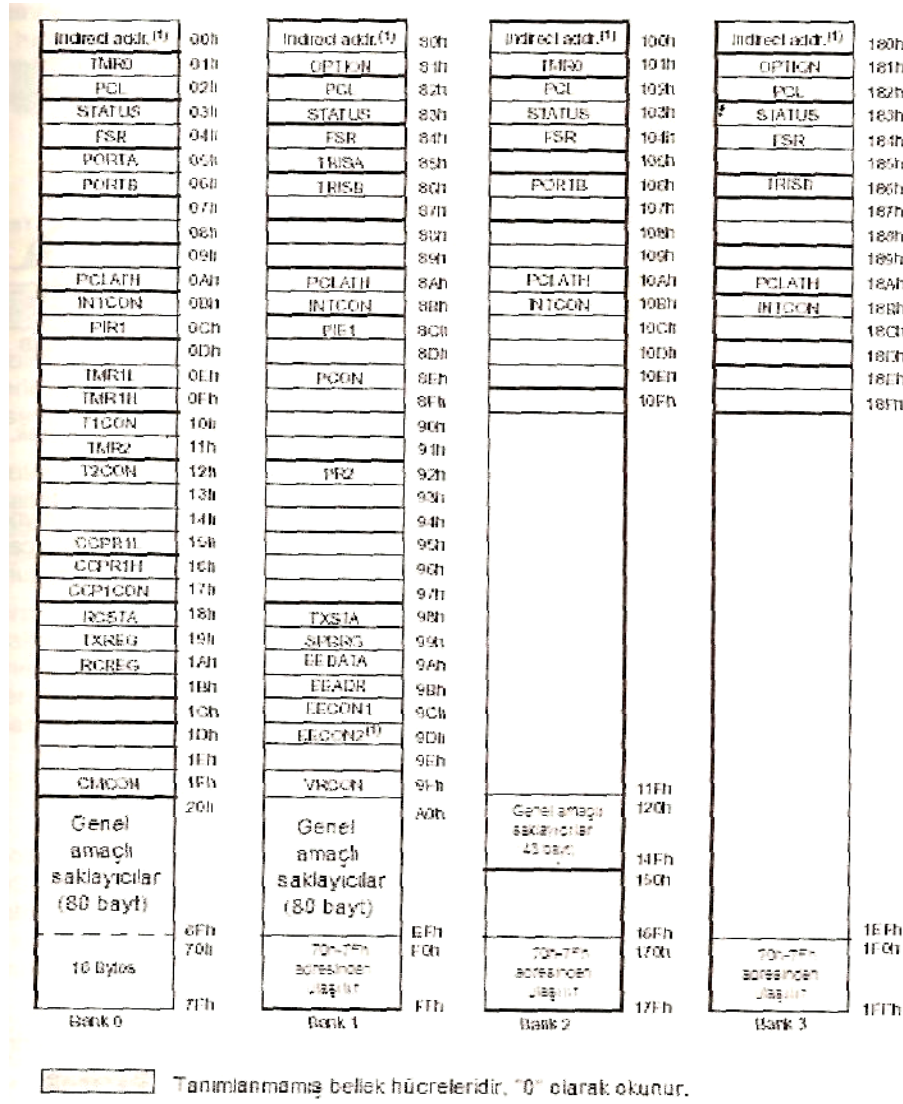
Genel Amaçlı Saklayıcılar

PIC16F628A'nın veri belleğinde toplam 8-bit'lik 224 adet GPR bulunmaktadır. PIC programlarında kullanılan değişkenler içerisinde saklanması gereken veriler için kullanılır. Çoğu zaman RAM da denilir. Bellek hücresinin adresi kullanılarak direkt olarak ulaşılabilirdiği gibi bellek seçme saklayıcısı (FSR-File Select Register) kullanılarak endirekt olarak ta ulaşılabilirler. SRAM yapısındadırlar.

Şekil 3-23'e dikkat edilirse 4 BANK'tan oluşan PIC16F628A'nın veri belleğindeki bazı özel amaçlı saklayıcılar diğer bellekte de bulunmaktadır. Örneğin PCL, STATUS, FSR, PCLATH, INTCON gibi. Bir bank'taki saklayıcıyı kullanabilmek için o bank'a geçmek gerekir. Bazı sık kullanılması gereken özel saklayıcıların birden fazla bank'ta da bulunmasının nedeni, bank değiştirme işlemine gerek duyulmaksızın kullanılabilmesi içindir.

Program içerisinde bir değişkenin adresi belirlenirken şu adresler kullanılmalıdır: Bank0'ın h'20'-h'6F', bank1'in h'AO'-h'EF', bank2'nin h'120'- h'14F'.

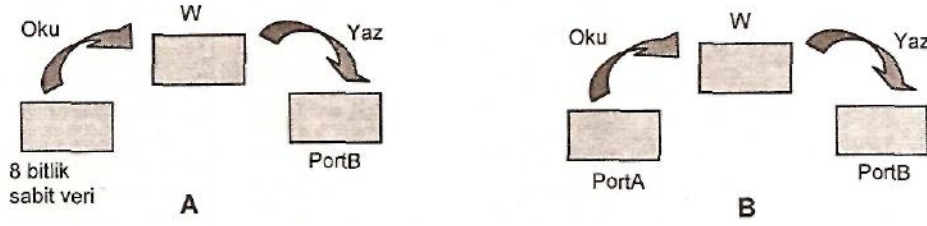
Veri belleğinin en alt kısmında kalan 16 adres alanı da program içerisinde kullanılan değişken yeri olarak tanımlanabilir. Ancak bu alanların diğerlerinden farkı hep h'70' ile h'7F' arasında bir adres bilgisi ile ulaşılabilmesidir. Yani buradaki bir genel amaçlı saklayıcıya bank değiştirme işlemi yapılmadan ulaşılabilir.



Şekil 2. 23 PIC16F628A'nın veri (RAM) belleği haritası.

2.4.8.3 W saklayıcısı

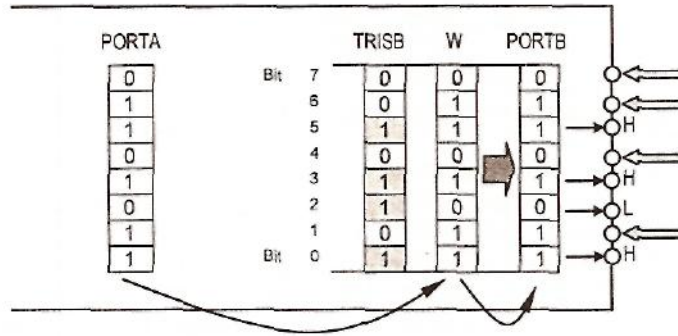
PIC16F628A'nın veri belleği alanında görülmeyen bir de 8-bit'lik W saklayıcısı vardır. W saklayıcısı bir akümülatör veya geçici depolama alanı olarak düşünülebilir. W saklayıcısına diğer saklayıcılarda olduğu gibi hücre adresini kullanılarak direkt olarak ulaşmak mümkün değildir. Aslında W saklayıcısının bir bellek adresi de yoktur. Komutun işlevi W saklayıcısı üzerindeyse adresi belirtilmeden ulaşılabilir. PIC'te gerçekleşen tüm aritmetik işlemler ve atama işlemleri için W saklayıcısı kullanılmak zorunluluğu vardır. >>>



Şekil 2. 24 A) Sabit bir veriyi W saklayıcısına yazıp, portB'ye aktarmak.

B) PortA'nın içeriğini W'ye yazıp, portB'ye aktarmak.

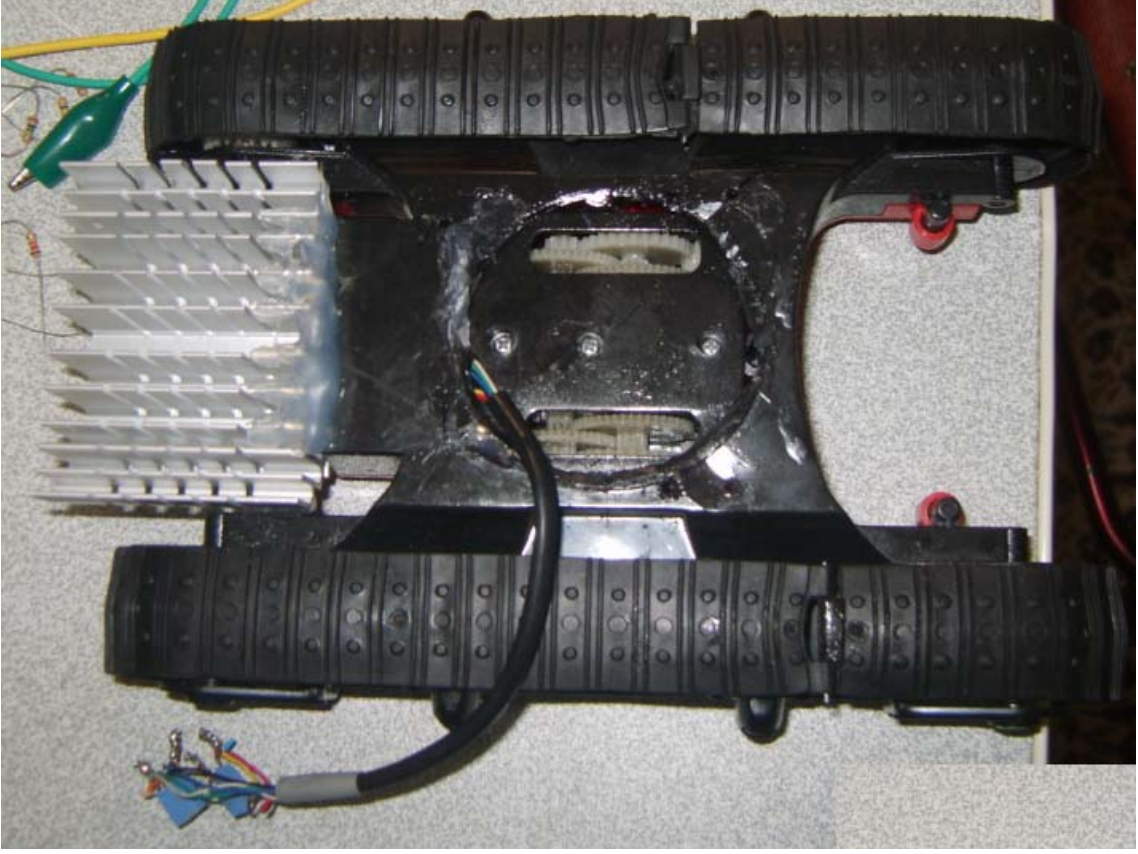
Örneğin PORTA'dan okunan veriler PORTB'ye aktarılmak isteniyorsa PORTA'nın içeriği önce W saklayıcısına aktarılır. Daha sonra da W'nin içeriği PORTB'ye aktarılır. Bu registerin kullanım özellikleri yine programlama konusunda detaylı olarak ele alınacaktır.



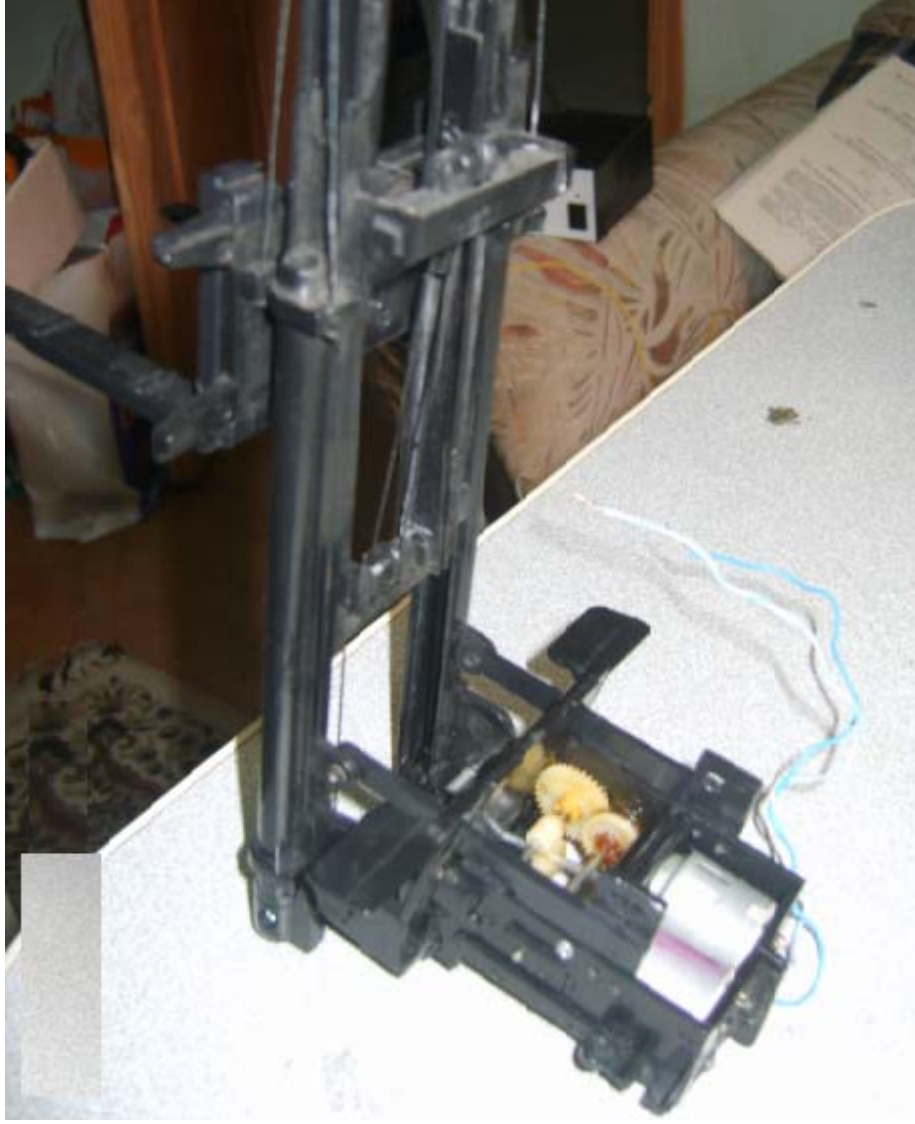
Şekil 2. 25 TRISB ile yönlendirilmiş PORTB'ye, PORTA'dan alınan verilerin aktarılması.

3. ROBOTUN MEKANİK TASARIMI

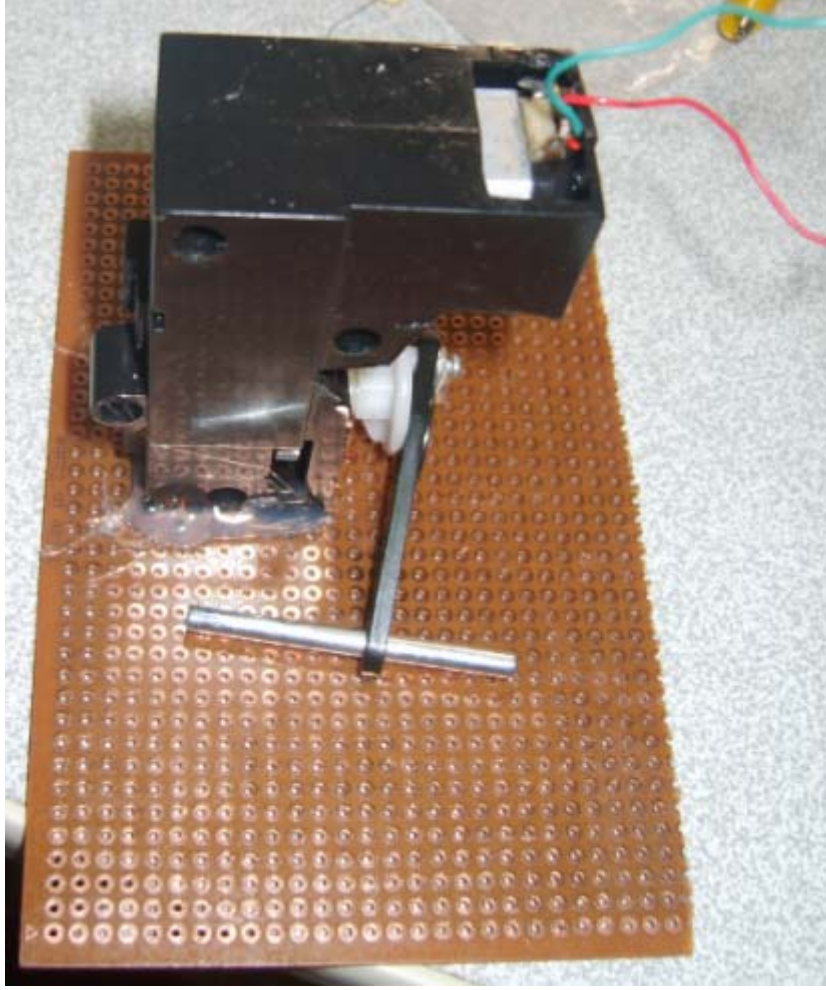
Robotun gerekli işlevleri yerine getirebilmesi için sahip olması gereken mekanik özellikler belirlenmiştir. Bu özellikleri sağlayabilecek şekilde iki oyuncak alınmıştır. Bunlardan robotun çatalını oluşturan kısım oyuncak forkliftten alınmıştır. Diğer kısım ise robotun hareket sistemini oluşturan paletlerdir ve bu palet kısmı da oyuncak excavatordan alınmıştır. Daha sonra bu oyuncaklar demontaj edilerek robot için gerekli olan mekanik işlevleri sağlayan kısımları alınıp uygun bir şekilde birbirlerine monte edilmiştir. Ayrıca her iki oyuncakta da özellik olarak bulunmayan ancak forklift robotun malzemeleri rahat bir şekilde alıp bırakabilmesi için gerekli olan bom al-bom ver özelliği yine bu oyuncaklardan sökülen diğer parçaların birleştirilmesiyle sağlanmıştır.



Resim 3.1 Forklift robotun palet kısmı

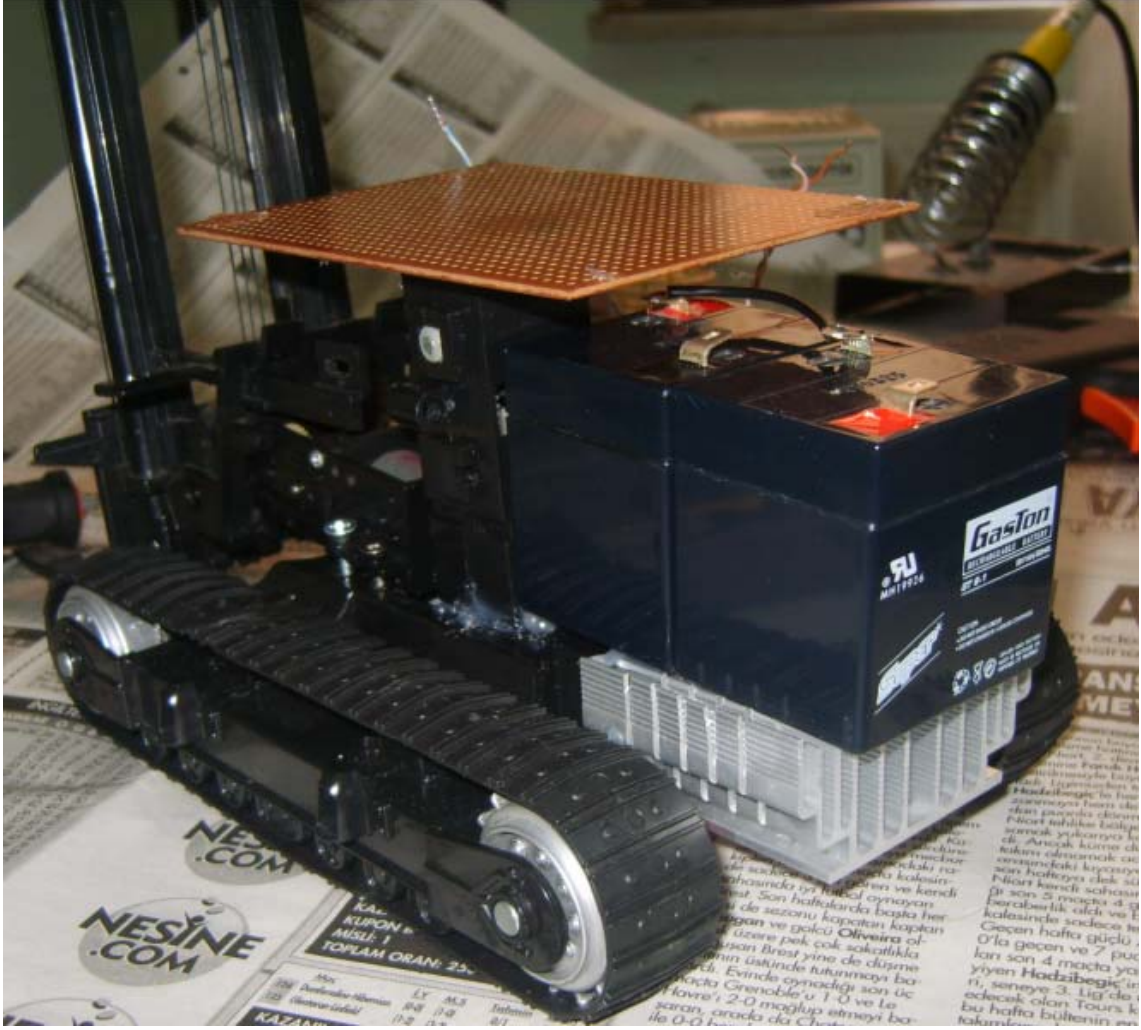


Resim 3.2 Forklift robotun çatal kısmı



Resim 3.3 Bom al-bom ver hareketini sağlayan redüktörlü motor

Yukarıdaki resimlerde görülen parçalar uygun bir şekilde biraraya getirilerek forklift robotun mekanik yapısı oluşturulmuştur. Robotun çatal kısmı dört adet vida yardımıyla palet kısma monte edilmiştir. Daha sonra bu iki kısım arasındaki bağlantıyı kuvvetlendirmek için bu iki ayrı parçanın birbirine temas eden yüzeyleri sıcak silikon kullanılarak sağlamlaştırılmıştır. Bom al-bom ver hareketini yapacak redüktör ise palet kısmı üzerinde uygun bir uzaklığa yalnızca sıcak silikon kullanılarak sabitlenmiştir. Redüktör üzerine uygulanan mekanik kuvvetin büyük olmasından dolayı forklift robotun enerjisini sağlayacak olan iki adet 6V – 1.3 Ah’lık aküler ardı ardına redüktörün arkasına yerleştirilmiştir. Akülerin birbirleriyle ve redüktörle olan montajı sıcak silikon kullanılarak gerçekleştirilmiştir. Bu yapı Resim.4’te gösterilmiştir.



Resim 3.4 Forklift robotun mekanik kısmı

4.DEVRE TASARIMI

Bu kısımda forklift robotun işlevlerini yerine getirecek olan motorları süren ve RF haberleşmeyi sağlayan elektriksel devreler detaylı bir şekilde anlatılmıştır.

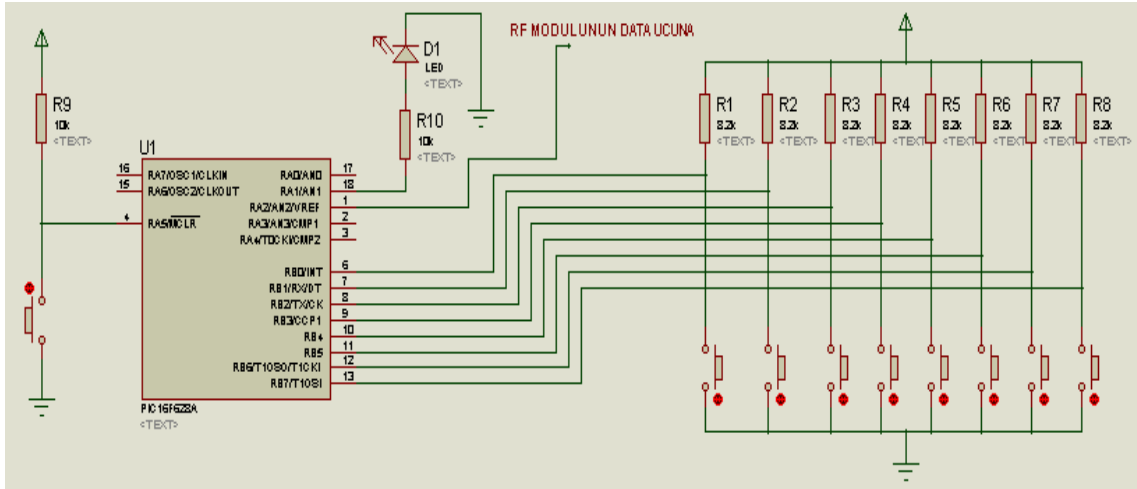
4.1 Verici Taraf Devre Tasarımı

Forklift robotun hareket etmesi için gerekli olan kontrol sinyallerini oluşturan iki ayrı kısım mevcuttur. Bunlardan birincisi tamamen insan kontrolünde olan uzaktan kumanda kısmıdır. İkincisi kısmı ise RS232 seri port vasıtasıyla bilgisayar bağlantısını sağlayan bilgisayar iletişim devresidir. Bu kısımlar aşağıda açıklanmıştır.

4.1.1 Uzaktan Kumanda

Uzaktan kumanda devresinde istenilen komutları yollamak için sekiz adet buton kullanılmıştır. Bunlardan bir tanesi çatal ve palet hareketleri arasında hangisinin aktif olacağını belirlemek için kullanılmıştır. Forklift robotun üç farklı hız kademesinde hareket etmesi amaçlandığından üç adet vites seçim butonu kullanılmıştır. Kalan diğer dört buton ise çatal ve paletin kontrolünde çatal için yukarı-aşağı, bom al-bom ver, palet için ise ileri-geri, sağ-sol hareketlerini gerçekleştirmek için kullanılmıştır.

Butonlardan gelen bu kontrol sinyallerini algılayarak RF verici modüle uygun bir şekilde iletmek için PIC16F628A mikrodenetleyicisi kullanılmıştır. Bu mikrodenetleyicinin JAL dilinde olan yazılımı ve Şekil 4.1’de ise uzaktan kumanda verici devresi aşağıda verilmiştir.



Şekil 4.1 Uzaktan kumanda verici devresi

Uzaktan Kumanda Vericisinin JAL Programı

```
include 16f628_4
include rf
include jlib
port_b_direction = all_input
```

```

forever loop
if pin_b7 == low then
  if pin_b2 == low then
    rf_trans = "U"
    rf_trans = "1"
  end if
  if pin_b0 == low then
    rf_trans = "U"
    rf_trans = "2"
  end if
  if pin_b3 == low then
    rf_trans = "U"
    rf_trans = "3"
  end if
  if pin_b1 == low then
    rf_trans = "U"
    rf_trans = "4"
  end if
else
  if pin_b2 == low then
    rf_trans = "U"
    rf_trans = "5"
  end if
  if pin_b0 == low then
    rf_trans = "U"
    rf_trans = "6"
  end if
  if pin_b3 == low then
    rf_trans = "U"
    rf_trans = "7"
  end if
  if pin_b1 == low then
    rf_trans = "U"

```

```

    rf_trans = "8"
  end if
end if
if pin_b4 == low then
  rf_trans = "U"
  rf_trans = "A"
end if
if pin_b5 == low then
  rf_trans = "U"
  rf_trans = "B"
end if
if pin_b6 == low then
  rf_trans = "U"
  rf_trans = "C"
end if
if port_b == 0b_1111_1111 | port_b == 0b_0111_1111 then
  rf_trans = "U"
  rf_trans = "D"
end if
end loop

```

JAL'daki RF Fonksiyonunun İçeriği

```
include rfp
```

```
include hd447804
```

```
var byte i, k = 0, 1
```

```
function rf_recv ( byte out vx ) return bit is
```

```
var bit vx_0 at vx : 0
```

```
var bit rf_veri_geldi, hata = false
```

```

var byte tm
rf_veri_geldi = false
rf_rcv_led = low

if rf_rcv == high then
  tmr0 = 0
  while rf_rcv == high & tmr0 < 255 loop end loop
  if ( tmr0 > 62 ) & ( tmr0 < 156 ) then
    tmr0 = 0
    while rf_rcv == low & tmr0 < 255 loop end loop
    if ( tmr0 > 32 ) & ( tmr0 < 80 ) then
      vx = 30

      for 8 loop
        rf_rcv_led = high
        vx = vx << 1
        while rf_rcv == high & tmr0 < 255 loop end loop
        tm = tmr0
        if (tm < 5) then hata = true end if
        rf_rcv_led = low
        tmr0 = 0
        while rf_rcv == low & tmr0 < 255 loop end loop
        if tmr0 > 40 then vx_0 = high else vx_0 = low end if

      end loop
      rf_rcv_led = low
      if hata == false then rf_veri_geldi = true end if

    end if

  end if

  return rf_veri_geldi
end function

```

```

procedure rf_rcv_lcd is
var byte data
  if i == 0 then hd44780_line1 end if
  if i == 16 then hd44780_line2 end if
    if rf_rcv ( data ) then
      hd44780 = data
      i = i + 1
    end if
  if i > 31 then
    hd44780_clear
    i = 0
  end if
end procedure

```

```

procedure high_25us ( byte in x ) is
assembler
  local T1
  bsf rf_trns_asm
t1:
  nop nop nop nop nop
  nop nop nop nop nop nop
  nop nop nop nop nop
  nop nop nop nop nop nop

  decfsz x,f
  goto t1

  bcf rf_trns_asm
end assembler

end procedure

```

```

procedure bekle_25us ( byte in x ) is
assembler

```

```

local T1
t1:
  nop nop nop nop nop
  nop nop nop nop nop
  nop nop nop nop nop
  nop nop nop nop nop
  nop nop
    decfsz x,f
    goto t1
    nop
    nop
end assembler
end procedure

procedure rf_trans'put ( byte in vx ) is
  var bit vx_7 at vx : 7

  high_25us ( 160 )

  rf_trns = low
  bekle_25us ( 80 )

  for 9 loop
    rf_trns_led = high

    high_25us ( 20 )
    rf_trns = low
    if vx_7 == low then bekle_25us ( 40 )
      else bekle_25us ( 60 )
    rf_trns_led = low
  end if
  vx = vx << 1
end loop
rf_trns_led = low

```

```
end procedure
```

```
intcon_gie = low
```

```
option = 0b_0100_0100
```

```
rf_trns = low
```

4.1.2 Bilgisayar İletişim Devresi

Forklift robotun bilgisayar ile kontrol edilebilmesi için bilgisayardan gelecek olan kontrol sinyallerini algılayıp robot üzerinde bulunan alıcı devreye RF sinyal olarak gönderilmesi için 20 MHz lik kristal ile çalışan PIC16F628A mikrodnetleyicisi kullanılmıştır. Bu mikrodnetleyici bilgisayardan gelen kontrol sinyallerini RS232 seri comport üzerinden almaktadır. Bu seri iletişim için PIC16F628A'nın USART özelliği kullanılmıştır. Bu mikrodnetleyicinin JAL dilinde olan yazılımı ve Şekil 4.2'de ise bilgisayar iletişim verici devresi aşağıda verilmiştir.

Bilgisayar İletişim Vericisinin JAL Programı

```
include 16f628_20
```

```
include jlib
```

```
include h_rs232
```

```
include rf_tx_20
```

```
var byte v
```

```
serial_setup ( 96 )
```

```
forever loop
```

```
    if h_asynch_poll ( v ) then
```

```
        for 3 loop
```

```
            rf_trans = "U"
```

```
            rf_trans = v
```

```

end loop
v = 0
end if

```

```

end loop

```

JAL'daki H_RS232 Dosyasının İçeriği

```

if (target_cpu != pic_16 ) & (target_cpu != pic_14 ) then
    pragma error -- not support
end if

```

Procedure serial_setup (byte in BAUD_CONSTANT) is begin

```

var volatile byte RCtemp

```

```

    if target_clock == 4_000_000 then
        if BAUD_CONSTANT == 192 then
            if target_cpu == pic_14 then Bank_1 end if
            SPBRG = 12
            TXSTA_BRGH = high
            if target_cpu == pic_14 then Bank_0 end if
        elsif BAUD_CONSTANT == 96 then
            if target_cpu == pic_14 then Bank_1 end if
            SPBRG = 25
            TXSTA_BRGH = high
            if target_cpu == pic_14 then Bank_0 end if
        elsif BAUD_CONSTANT == 24 then
            if target_cpu == pic_14 then Bank_1 end if
            SPBRG = 103
            TXSTA_BRGH = high
            if target_cpu == pic_14 then Bank_0 end if
        elsif BAUD_CONSTANT == 12 then
            if target_cpu == pic_14 then Bank_1 end if
            SPBRG = 207
            TXSTA_BRGH = high
            if target_cpu == pic_14 then Bank_0 end if
        end if
    end if
end serial_setup

```

```

end if
end if
if target_clock == 10_000_000 then
  if BAUD_CONSTANT == 192 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 31
    TXSTA_BRGH = high
    if target_cpu == pic_14 then Bank_0 end if
  elsif BAUD_CONSTANT == 96 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 64
    TXSTA_BRGH = high
    if target_cpu == pic_14 then Bank_0 end if
  elsif BAUD_CONSTANT == 24 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 64
    TXSTA_BRGH = low
    if target_cpu == pic_14 then Bank_0 end if
  elsif BAUD_CONSTANT == 12 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 129
    TXSTA_BRGH = low
    if target_cpu == pic_14 then Bank_0 end if
  end if
end if
if target_clock == 20_000_000 then
  if BAUD_CONSTANT == 192 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 64
    TXSTA_BRGH = high
    if target_cpu == pic_14 then Bank_0 end if
  elsif BAUD_CONSTANT == 96 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 129

```

```

    TXSTA_BRGH = high
    if target_cpu == pic_14 then Bank_0 end if
elseif BAUD_CONSTANT == 24 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 129
    TXSTA_BRGH = low
    if target_cpu == pic_14 then Bank_0 end if
elseif BAUD_CONSTANT == 12 then
    if target_cpu == pic_14 then Bank_1 end if
    SPBRG = 255
    TXSTA_BRGH = low
    if target_cpu == pic_14 then Bank_0 end if
end if
end if

```

```

if target_clock == 40_000_000 then
    if BAUD_CONSTANT == 192 then
        if target_cpu == pic_14 then Bank_1 end if
        SPBRG = 0x81
        TXSTA_BRGH = high
        if target_cpu == pic_14 then Bank_0 end if
    elseif BAUD_CONSTANT == 96 then
        if target_cpu == pic_14 then Bank_1 end if
        SPBRG = 0x40
        TXSTA_BRGH = low
        if target_cpu == pic_14 then Bank_0 end if
    end if
end if

```

```

    if target_cpu == pic_14 then Bank_1 end if
    TXSTA_TXEN = High
    if target_cpu == pic_14 then Bank_0 end if
    RCSTA_CREN = high
    RCSTA_SPEN = high

```

end procedure

Procedure serial_recv (byte out data) is begin

```

if target_cpu == pic_14 then
    Bank_0
    while ! PIR1_RCIF loop end loop
    file_get ( 0x1A , data )
end if
if target_cpu == pic_16 then
    while ! PIR1_RCIF loop end loop
    data = RCREG
end if
end procedure

```

Procedure serial_transmit (byte in data) is begin

```

if target_cpu == pic_14 then
    Bank_0
    while ! PIR1_TXIF loop
    end loop
    file_put ( 0x19 , data )
end if

```

```

if target_cpu == pic_16 then
    while ! PIR1_TXIF loop end loop
    TXREG = data
end if
end procedure

```

procedure H_asynch'put(byte in x) is

```

    serial_transmit ( x )
end procedure

```

function H_asynch_poll(byte out x) return bit is

```

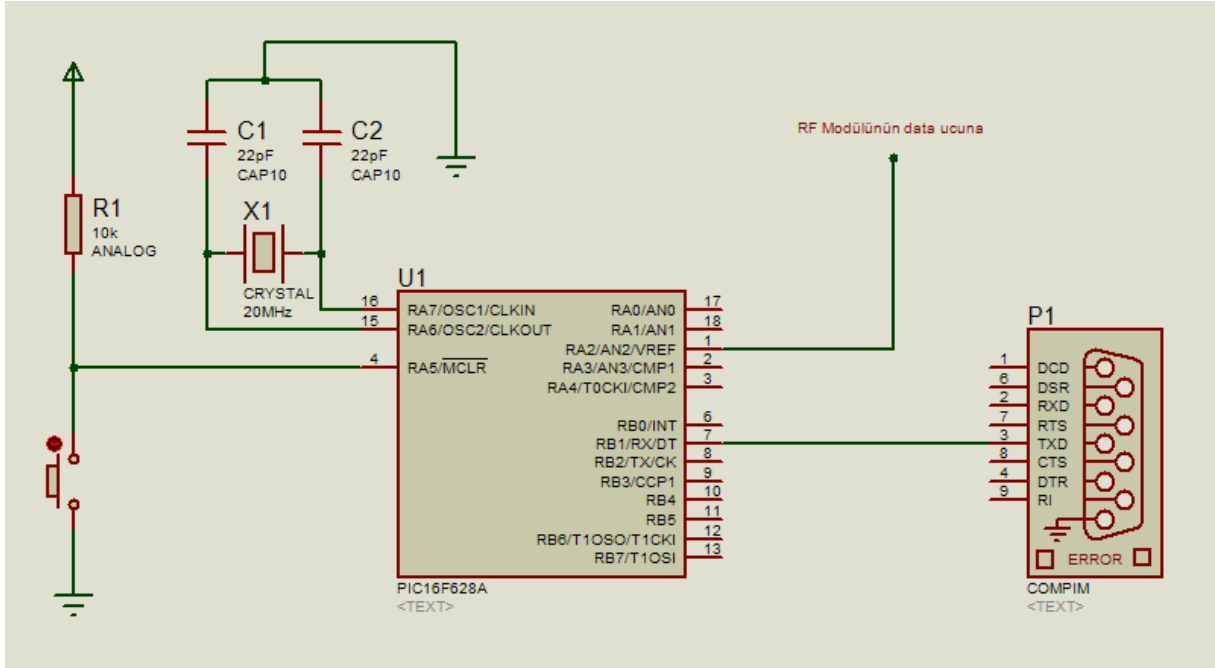
if PIR1_rcif then
    serial_recv ( x )

```

```

return true
else
    return false
end if
end function

```



Şekil 4.2 Bilgisayar iletişim verici devresi

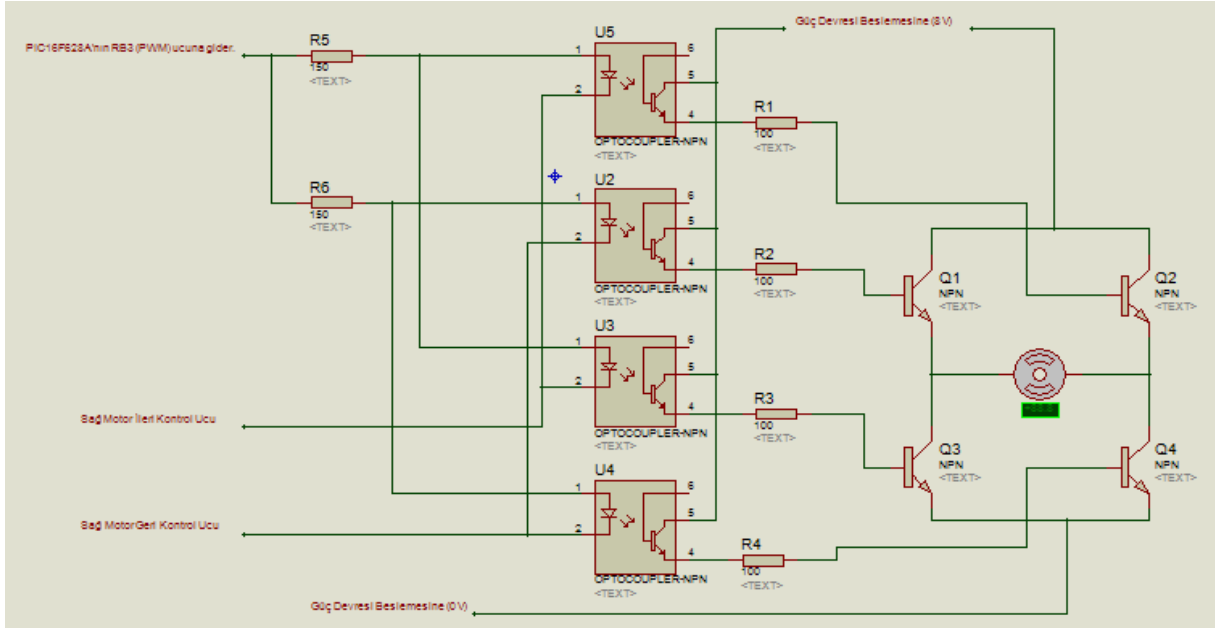
4.2 Alıcı Taraf Devre Tasarımı

Forklift robotun alıcı devresi robot üzerinde bulunan dört adet motorun, verici devrelerinden gönderilen kontrol sinyalleri doğrultusunda sürülmesi işlemlerini yerine getirir. Verici devrelerinden gönderilen RF kontrol sinyalleri, alıcı kart üzerinde bulunan RF alıcı modül tarafından çözülerek PIC16F628A mikrodenetleyicisine gönderilir. PIC16F628A mikrodenetleyicisi aldığı bu çözülmüş sinyalleri içerisindeki programa göre işleyerek gerekli motor sürme devrelerine çıkış sinyallerini uygular.

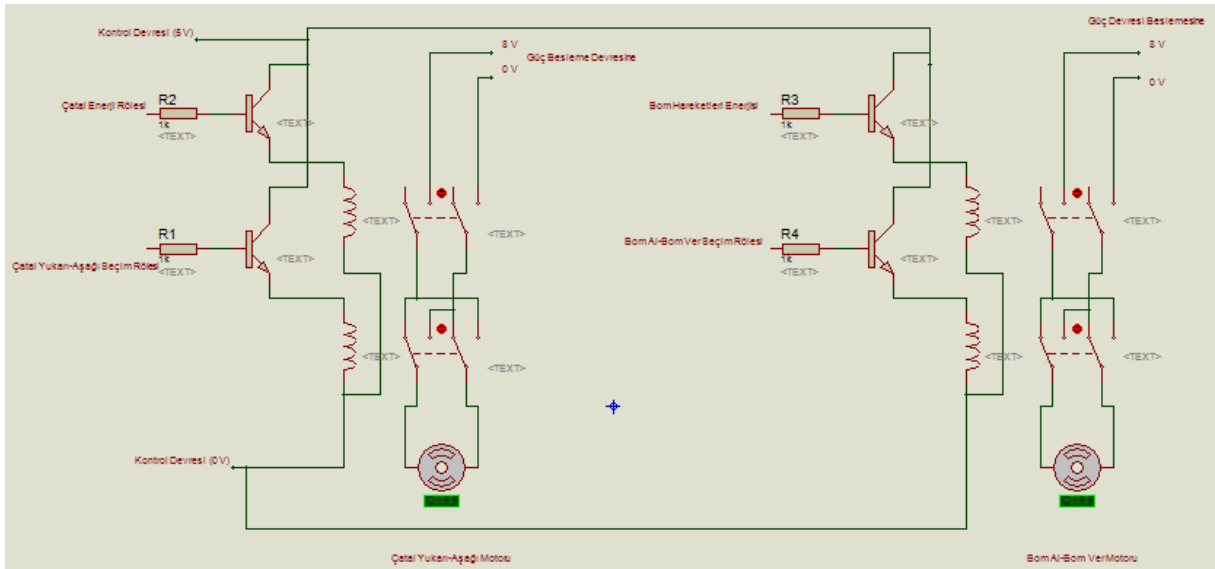
Forklift robot üzerinde dört adet motor bulunmaktadır. Bu motorlardan iki tanesi robotun hareketini sağlayan sağ ve sol palet motorlarıdır. Sağ ve sol palet motorlar robotun üç farklı hızda hareket etmesi istendiğinden PWM ile kontrol edilmiştir. Diğer iki motor ise çatal hareketleri için kullanılmıştır. Çatal hareketlerinde PWM kontrolüne ihtiyaç duyulmamıştır.

Sağ ve sol palet motorların PWM kontrolü herbir motor için ayrı bir H köprüsü kullanılarak gerçekleştirilmiştir.H köprüsünde kullanılan transistörler motorları besleyebilecek şekilde akım değerine sahip olan BD137 transistörleridir.H köprüsünde bir zaman anı için karşılıklı çapraz paralel kollardaki transistörler ilettime geçirilerek motorun bir yönde dönmesi sağlanmıştır.Motorun diğer yana dönmesinde ise H köprüsündeki diğer transistörler ilettime geçirilmiştir.H köprüsünde üç adet farklı hız elde etmek için transistörlerin beyzlerine 100 Ω 'luk dirençler üzerinden PWM sinyal uygulanmaktadır. Kontrol devresi ile güç devresini birbirinden elektriksel olarak yalıtım için PC817 optokuplör entegreleri kullanılmıştır. Ayrıca motorlarda oluşan zıt emk dan dolayı transistörlerin olumsuz bir şekilde etkilenmemesi için transistörlerin kollektör ve emiter uçlarına ters paralel bağlı 1N4001 diyotları kullanılmıştır.Böylece bu diyotlar motor zıt emklarını üstlerine alarak transistörleri ters yönde zorlamazlar.

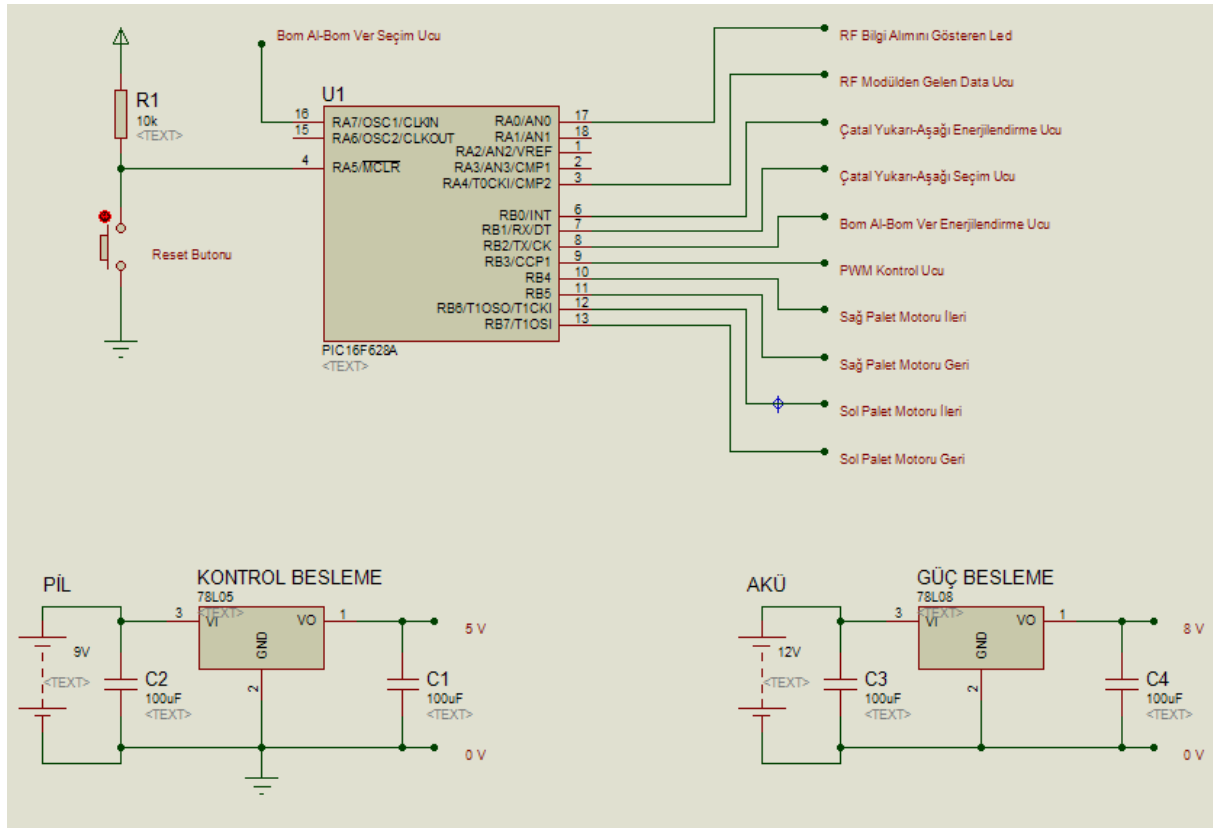
Çatal hareketlerini gerçekleştiren motorlar ise sadece her iki yönde dönmesi gerektiğinden bu işlemi gerçekleştirmek için her bir motor iki röle ile sürülmektedir.Bu rölelerden ilki motoru enerjilendirir, ikincisi ise motorun dönüş yönünü belirler.Rölelerin bobinlerinde de endüktans özelliği görüldüğünden dolayı burada da bir zıt emk söz konusudur.Bu yüzden rölelerin bobin uçlarındaki zıt emk yı engellemek için 1N4001 diyotları kullanılmıştır.PIC16F628A'nın çıkış akımı 25 mA ile sınırlı olduğundan ve bu akım röleleri sürmek için yeterli olmadığından dolayı her bir röleyi sürmek için BD137 güç transistörü kullanılmıştır.Burdaki güç transistörlerinin beyzleri 1 k Ω 'luk dirençler üzerinden PIC16F628A'nın ilgili bacaklarına direk olarak bağlanmıştır. Bu mikrodenetleyicinin JAL dilinde olan yazılımı ve Şekil 4.3 ,Şekil4.4 ve Şekil 4.5'de ise alıcı devresi kısım kısım aşağıda verilmiştir.Devrenin bitmiş halinin ve robota monte edilmiş halinin resimleri aşağıda verilmiştir.



Şekil 4.3 Palet motoru sürme devresi (Sağ ve sol motor için aynı düzendeki devreler kullanılmaktadır.)



Şekil 4.4 Çatal motorları sürme devresi



Şekil 4.5 PIC16F628A bağlantı uçları ve besleme regülatörleri

Alıcı Devresinin JAL Programı

```
include 16f628_4
```

```
include rf
```

```
include jlib
```

```
include pwm
```

```
pin_b0_direction = output
```

```
pin_b1_direction = output
```

```
pin_b2_direction = output
```

```
pin_b4_direction = output
```

```
pin_b5_direction = output
```

```
pin_b6_direction = output
```

```
pin_b7_direction = output
```

```
pin_a5_direction = input
```

```
pin_a7_direction = output
```

```
assembler  
bank 0  
movlw 0x07  
movwf 0x1F  
end assembler
```

```
var byte m
```

```
forever loop
```

```
if rf_rcv(k) then  
while k == "U" loop
```

```
if rf_rcv(m) then  
  if m == "1" then  
    pin_b2 = high  
    pin_a7 = low  
  end if
```

```
  if m == "2" then  
    pin_b2 = high  
    pin_a7 = high  
  end if
```

```
  if m == "3" then  
    pin_b0 = high  
    pin_b1 = low  
  end if
```

```
  if m == "4" then  
    pin_b0 = high  
    pin_b1 = high  
  end if
```

```
if m == "5" then  
  pin_b4 = low  
  pin_b6 = low  
end if
```

```
if m == "6" then  
  pin_b5 = low  
  pin_b7 = low  
end if
```

```
if m == "7" then  
  pin_b5 = low  
  pin_b6 = low  
end if  
if m == "8" then  
  pin_b4 = low  
  pin_b7 = low  
end if
```

```
if m == "A" then H_pwm ( 200 ) end if  
if m == "B" then H_pwm ( 227 ) end if  
if m == "C" then H_pwm ( 255 ) end if
```

```
if m == "D" then  
  pin_b0 = low  
  pin_b1 = low  
  pin_b2 = low  
  pin_b4 = high  
  pin_b5 = high  
  pin_b6 = high  
  pin_b7 = high  
  pin_a7 = low  
end if
```

```

    k = 0

end if
end loop
end if

end loop

```

JAL'daki PWM Dosyasının İçeriği

```

include pwmp

if target_clock != 4_000_000 then
    pragma error
end if

procedure delay_10usn ( byte in x ) is
assembler
    local T1
t1:
    nop
    nop
    decfsz x,f
    goto t1
end assembler

end procedure

procedure pwm (byte in x ,byte in tekrar ) is
    for tekrar loop
        Pwm_pin = high

```

```

    delay_10usn ( x )
    Pwm_pin = low
    delay_10usn ( 255 - x )
end loop
end procedure

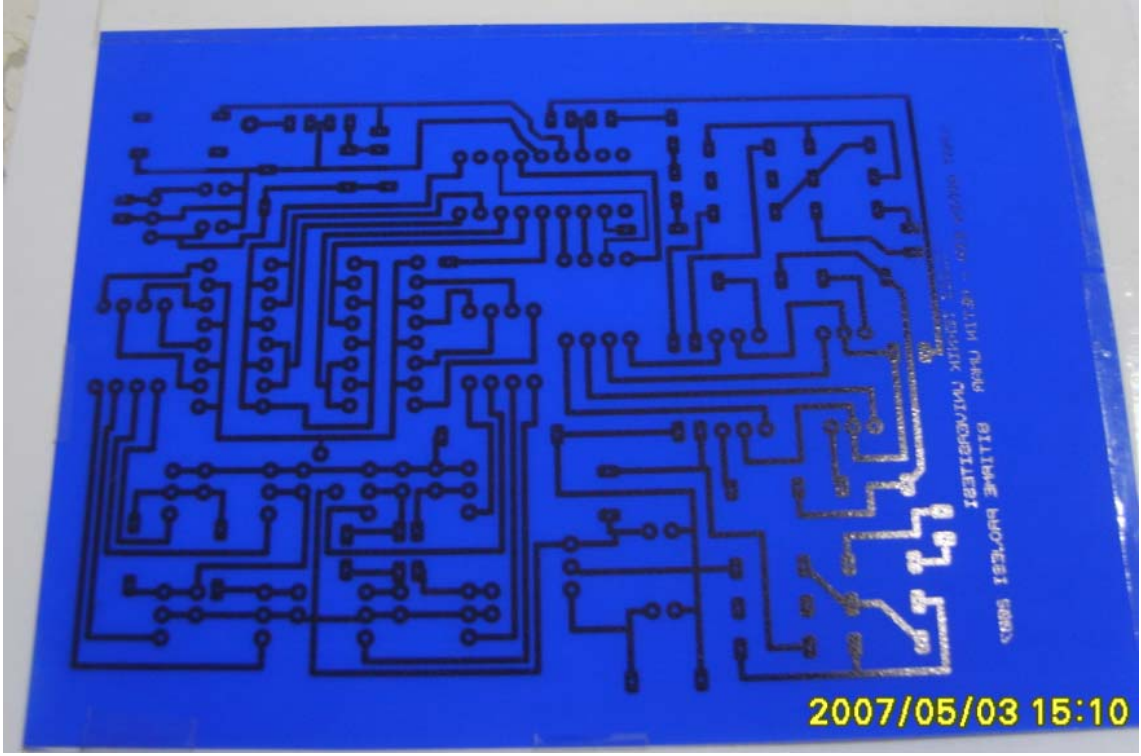
```

Procedure H_pwm (byte in value) is

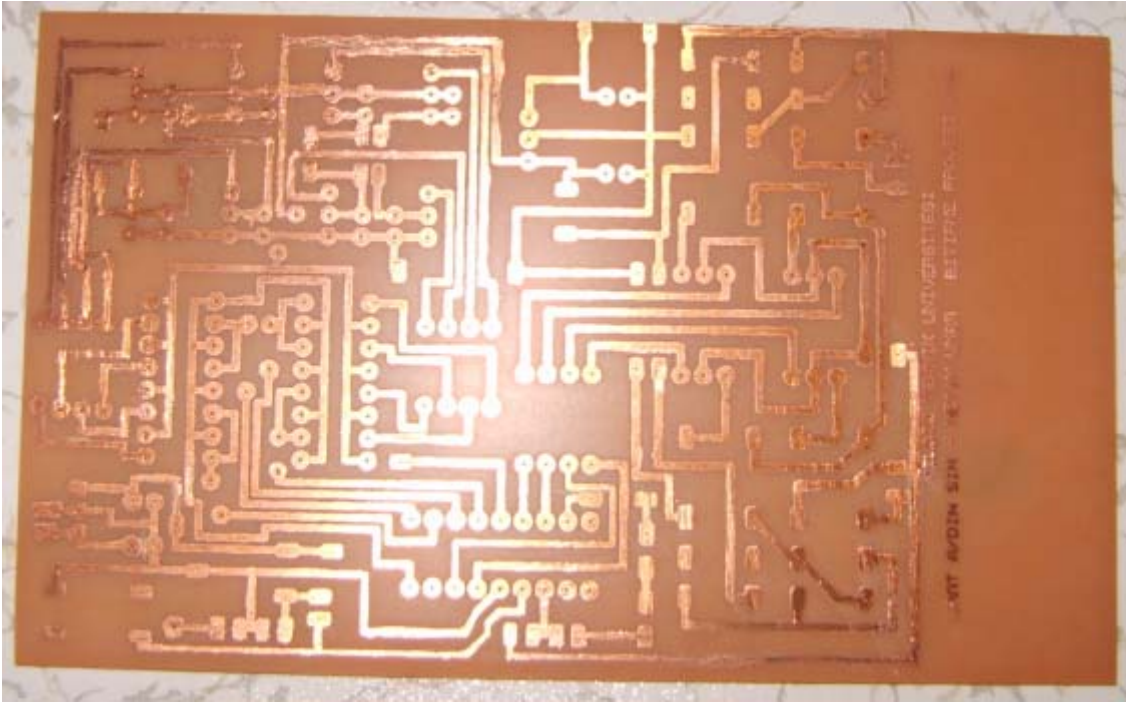
```

    if value == 0 then
        f877_CCP1CON = 0
        pin_c2_direction = input
    else
        if f877_CCP1CON == 0 then
            Bank_1
            f877_PR2 = 0xFF
            Bank_0
            f877_CCPR1L = value
            pin_c2_direction = output
            pin_c2 = low
            f877_T2CON = 0x04
            f877_CCP1CON = 0x0C
        else
            f877_CCPR1L = value
        end if
    end if
end procedure

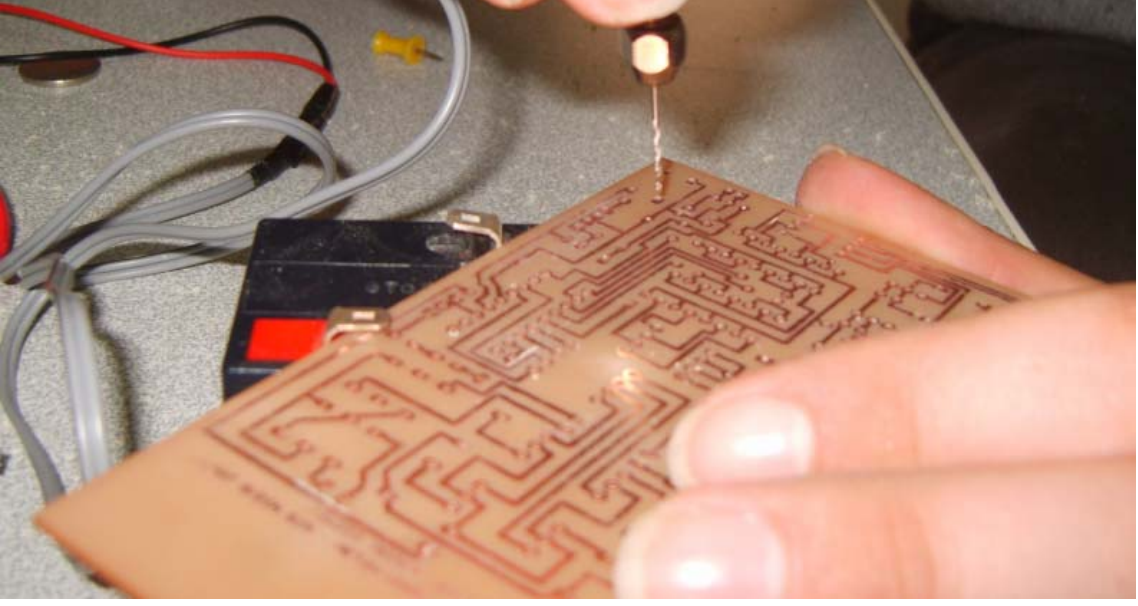
```



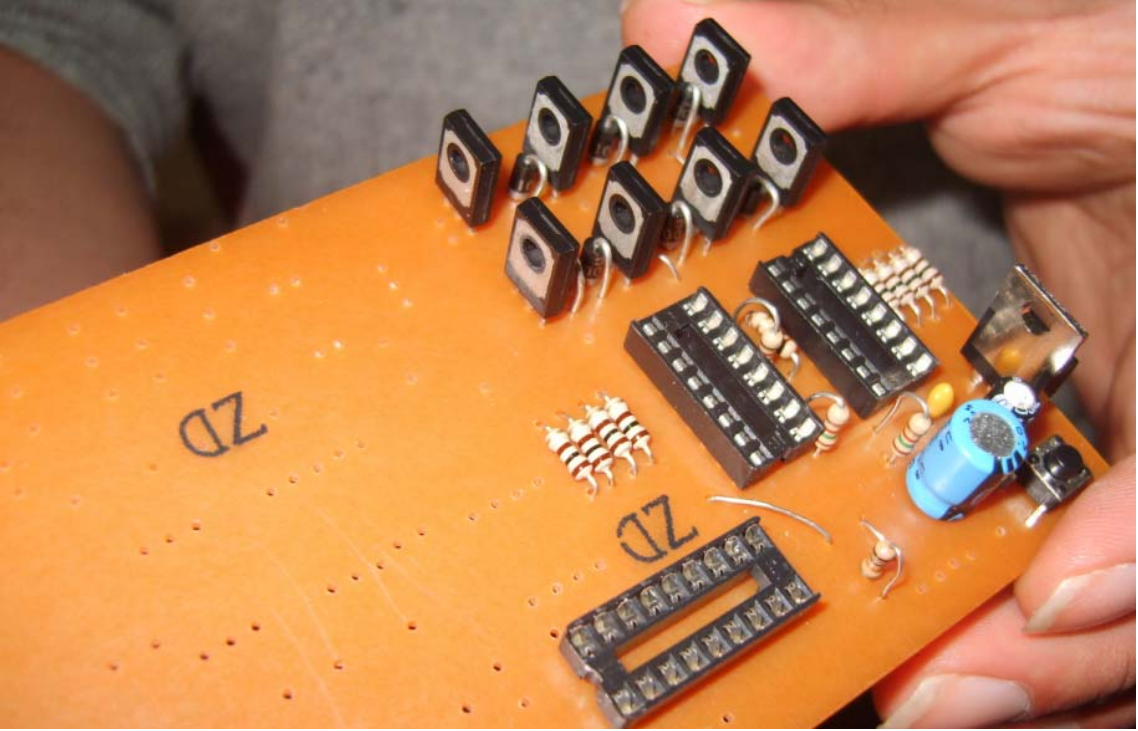
Resim 4.1 Baskı devrenin PNP kağıdı çıktısı



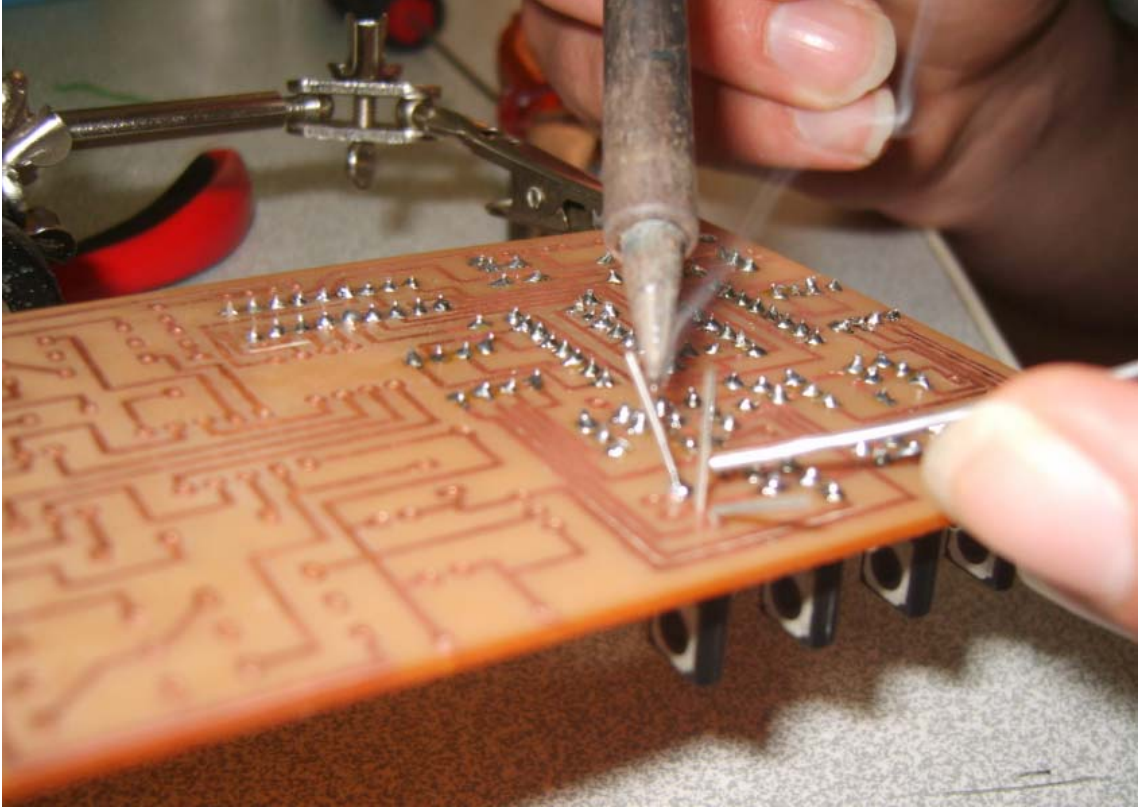
Resim 4.2 Baskı devrenin son hali



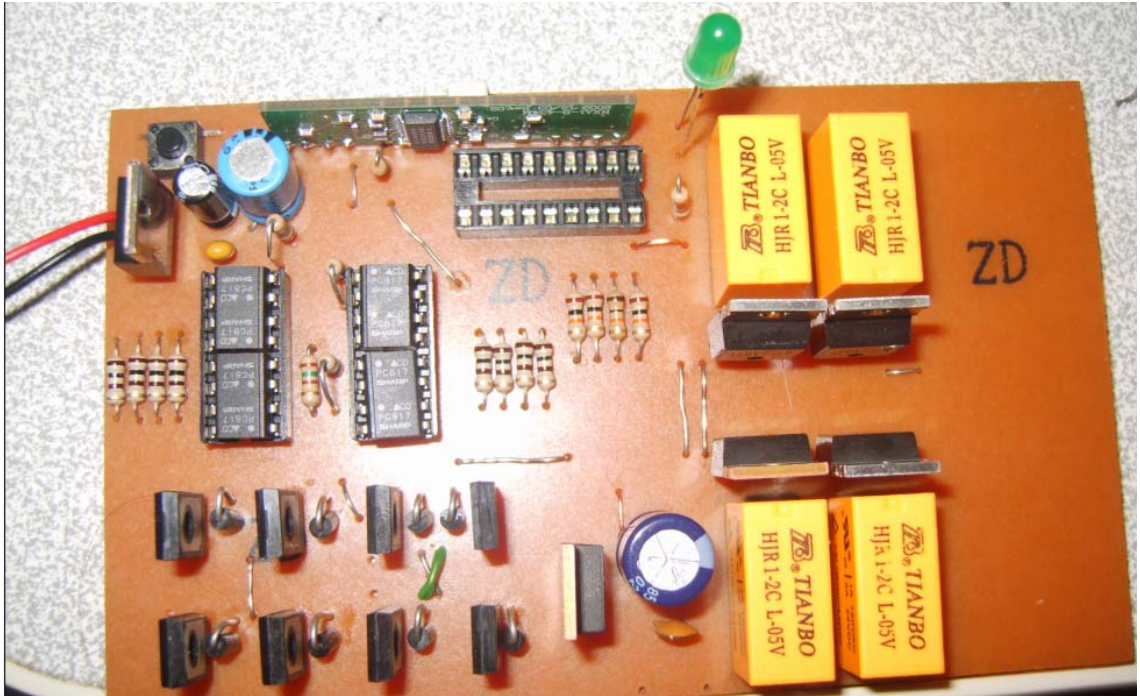
Resim 4.3 Baskı devrenin delinmesi



Resim 4.4 Baskı devreye elemanların yerleştirilmesi



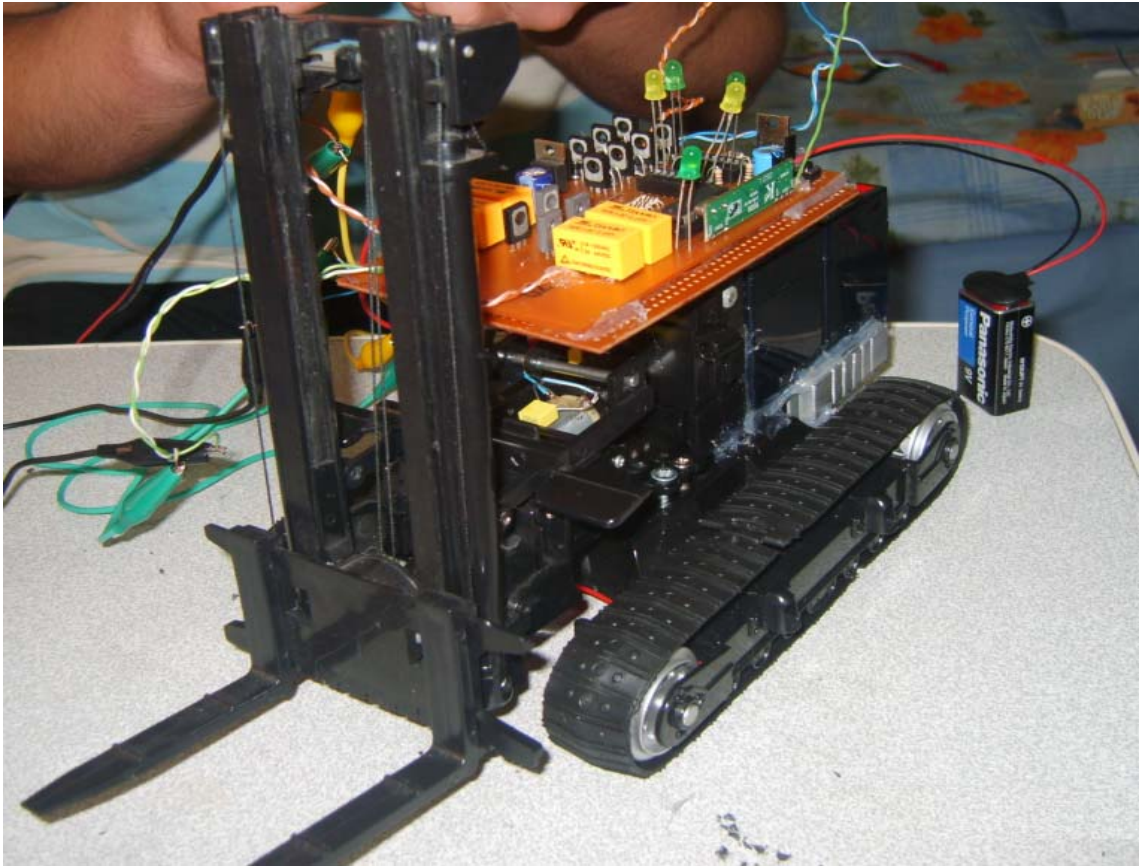
Resim 4.5 Baskı devreye elemanların lehimlenmesi



Resim 4.6 Baskı devrenin tamamlanmış hali



Resim 4.7 Alıcı devre , uzaktan kumanda devresi ,aküler ve pil



Resim 4.8 Baskı devrenin robot üzerine monte edilmiş hali

5 JAL PROGRAMLAMA DİLİ

Bu bölümde JAL programlama dili hakkındaki detaylı bilgi aşağıda verilmiştir.

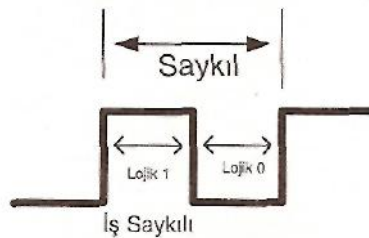
5.1 JAL'IN TANITIMI

JAL, Microchip ve Unicom firmasının mikro denetleyicileri için geliştirilmiş ücretsiz (Freeware), yüksek seviyeli ve açık kaynak koduna sahip bir derleyicidir. Jal ücretsiz olması sebebiyle Internet üzerinden rahatlıkla indirilebilir ve lisans probemi yaşamadan gönül rahatlığı ile kullanılabilir. JAL'ın açık kaynak koduna sahip olması sebebiyle kullanıcılar JAL'ın kaynak kodlarından yararlanıp programlar yazabilir veya JAL'ın kaynak kodlarını geliştirebilirler. JAL yazım kuralları olarak PASCAL ve C'yi temel almış olup oldukça sade ve anlaşılır bir program yazım tekniği kullanır. Dolayısıyla önceden Pascal veya C kullanan kişiler hızla jala uyum sağlayabileceklerdir. Yeni başlayanlar ise kolay yazım kuralları sebebiyle uyum sağlamakta güçlük çekmeyeceklerdir.

Microchip firması kendi ürettiği mikro denetleyicilerine PIC adını pektedir. JAL kullanarak PIC16C84, PIC16F84, PIC16F877, PIC12C509A, PIC12C508, PIC12CE674, PIC16F628, PIC18F252, PIC18F242, PIC18F442, PIC18F452, PIC12F629, PIC12F675, PIC16F88 PIC türleri ve Unicom firmasının ' B SX28 isimli mikro denetleyicileri için program yazılabilir. Ayrıca ileriki bölümlerde açıklanacağı gibi temel mimarisi aynı olan mikro denetleyiciler için de program yazılmasına imkan sağlar. Örneğin PIC16F628 için yazılacak bir JALprogramı, PIC16F628 ile aynı temel özelliklere sahip PIC16F627 ve PIC16F648 için de kullanılabilir.

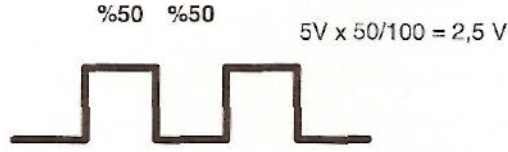
5.2 PWM'NİN TANIMI

PWM (Pulse Width Modulation) pals genişlik modülasyonu anlamına gelmektedir. DC motorların hız ayarında, RC servolarda, İR veya RF iletişimde sinyali modüle etmekte kullanılır. PWM sinyali yazılım ile oluşturulabildiği gibi PIC16F877 gibi üstün özellikli mikro denetleyicilerde donanımsal olarak da oluşturulabilir. Bu bölümde daha çok DC motorların hız kontrolünde kullanılmak üzere yazılımsal ve donanımsal PVVM sinyali üretmek üzerinden durulacaktır.



Şekil 5.1 Kare dalga sinyal.

PWM sinyali aslında basit kare dalgaya benzer fakat kare dalgadaki gibi palsin lojik 1 ve lojik 0 bölümleri her zaman birbirine eşit değildir. PVVM sinyalinde palsin lojik 1 ve lojik 0 kısımlarının oranlarının değiştirilmesi ile farklı gerilimde çıkış sinyali oluşturulur. PWM sinyalinde tam bir sinyale saykıl denir, saykılın lojik 1 kısmına iş saykılı denilmektedir. İş saykılının tüm saykıla oranı çıkıştaki sinyalin değerini verir. PIC'in pin çıkışları 5V verebilir. Eğer PIC'in çıkışından saykılın %50'si oranında iş saykılı olan bir



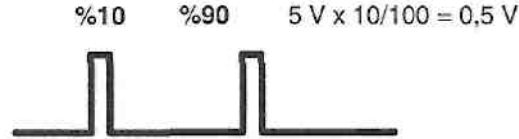
Şekil 5.2 %50 Oranında iş saykılı olan kare dalga.

sinyal üretilirse çıkıştaki sinyalin değeri $5 \text{ v} \times 50/100 = 2,5 \text{ v}$ olur. Sinyal'in yarısı lojik 1 yarısı lojik 0'dan oluşursa çıkışta 5 volt'un yarısı alınır.

İş saykılı, tüm saykılın %80'ni olan bir sinyalin gerilim değeri;

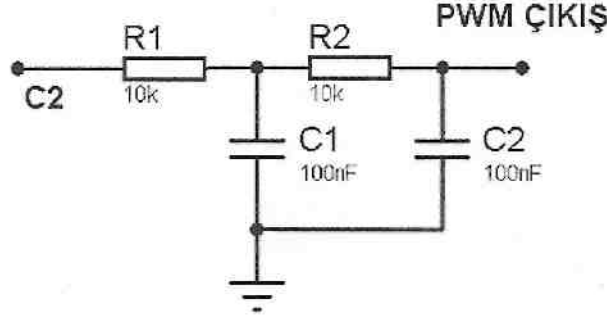
$5 \times 80/100 = 4 \text{ v}$ ölçülür. Çıkıştaki gerilimi düşürmek için iş saykılının tüm saykıla oranını düşürmek gerekir. Örneğin iş saykılı tüm saykıl'ın %10'u olan bir sinyalin değeri;

$5 \text{ V} \times 10/100 = 0,5 \text{ v}$ ölçülür.



Şekil 5.3 %10 Oranında iş saykılı olan kare dalga.

PIC'den PWM sinyali üretirken dikkat edilecek konulardan biri de çıkışın direkt değil de bir filtre devresi üzerinden alınması gerektiğidir. Şekil 5.4'de PIC'in B3 pin'ine bağlı filtre devresi görülmektedir.



Şekil 5.4'de PIC'in B3 pin'ine bağlı filtre devresi

5.2.1 YAZILIMSAL PWM

Yazılımsal olarak PWM sinyali üretmek için, süresi değiştirilebilir bir ka-e dalga programı yazmak yeterlidir.

5.2.2 JAL'da Yazılımsal PWM Özelliğinin Kullanımı

JAL'da PWM sinyali ile ilgili alt programlar pwm.jal kütüphane dosyasında bulunur. Pwm.jal dosyası ise yazılımsal olarak PWM sinyali üretilecek PIC pin'ini tanımlamakta kullanılır. pwm.jal kütüphane dosyasındaki yazılımsal pwm sinyali üretmek ile ilgili alt programlar aşağıdaki gibidir;

procedure pwm (byte in x , byte in tekrar) : Yazılımsal PWM sinyali üretmek için kullanılır. x değişkeni üretilecek sinyalin iş saykılıının toplam saykıla oranını belirler. x değişkeni 0..255 arasında değer alabilir. Tablo 5.1'de iş saykılı oranına göre x değişkenine yazılacak değerlerin tablosu görülmektedir. Tekrar değişkeni ise her seferinde tekrar edilecek saykıl sayısını belirler.

Tablo 5.1 PWM sinyali üretme oran tablosu

İş Saykılıının Oranı	Formül	Yazılacak Değer
%12,5	$12,5 \cdot 255 / 100 = 31,875$	32
%25	$25 \cdot 255 / 100 = 63,75$	64
%50	$50 \cdot 255 / 100 = 127,5$	128
%100	$100 \cdot 255 / 100 = 255$	255

5.3 DONANIMSAL PWM

PIC16F877 ve PIC16F628 gibi bazı PIC türlerinde PWM sinyali donanımsal olarak da üretilebilir. PIC16F877'de PWM sinyali C2 pin'inden üretilirken, PIC16F628'de B3 pin'inden üretilir. PIC16F877'de PWM sinyali üreten C2 pin'ine CCP1 adı verilir. PIC16F877'de PWM sinyalinin üretiminde TMR2 sayıcısı kullanılır. PWM sinyalinin iş saykılıının süresi 10 bit

hassasiyetindedir. PWM sinyalinin iş saykılı CCPR1L ve CCPR1H kaydedicilerinde tutulurlar Dilenirse CCPR1H kaydedicisine değer yüklenmeyerek 8 bit hassasiyetinde de değer verilebilir.

PIC'de donanımsal PWM sinyali üretebilmek için aşağıdaki adımlar izlenmelidir;

- PWM sinyalinin periyodu PR2 kaydedicisine yazılır.
- PWM sinyalinin iş saykılı CCPR1L kaydedicisine kaydedilir.
- CCP1 pin'i çıkış yapılarak low yapılır (TRISC<2>).
- TMR2'nin prescaler değeri yüklenip TMR2 kaydedicisi yetkilendirilir.
- CCP1 modülü PWM işlemi için kurulur.

Tablo 5.2 20 Mhz'lik PIC'de üretilebilecek PWM sinyalinin frekansı ve hassasiyeti.

PWM frekansı	1,22 Khz	4,88 Khz	19,53 Khz	78,12 Khz	156,3 Khz	208,3 Khz
TMR prescaler (1,4,16)	16	4	1	1	1	1
PR2 Değeri	0XFF	0XFF	0XFF	0X3F	0X1F	0X17
Maksimum Hassasiyet (bit)	10	10	10	8	7	5,5

5.3.1 JAL'da Donanımsal PWM Özelliğinin Kullanımı

PIC'de donanımsal PWM sinyali üretmek için pwm. jal kütüphane dosyası içerisindeki H_pwm (x) alt programı kullanılır. H_pwm (x) alt programının komutunda parantez içindeki parametresine iş saykılıının değeri yazılır. İş saykılıının değeri 0..255 arasında bir sayıdır. İş saykılına değer olarak 255 girilirse bu iş saykılıının tüm saykılın %100'ünü, 128 girilirse iş saykılıının tüm saykılın %50'sini kapsadığı anlamına gelir. Tablo 17.1'de iş saykılı oranı için yazılması gereken değerlerin tablosu bulunmaktadır.

5.3.2 Donanımsal PWM Uygulamaları

PIC'in BO pin'indeki butonuna basılınca iş saykılı %12,5, B1 pin'indeki butonuna basılınca iş saykılı %25, B2 pin'indeki butonuna basılınca iş saykılı %50, B3 pin'indeki butonuna basılınca iş Saykılı %100 olan donanımsal PWM sinyalinin üretecek JAL programı. Bu örnek için şekil 17.6 kullanılabilir.

```
include 16f877_4
```

```
include jlib
```

```

include pwm    -- Unite7\Ornl7_3.jal

port_b_direction = all_input -- Buton giriři h_pwm { 125 }

forever loop
if pin_b0 == high then h_pwm      ( 32 ) end if
if pin_b1 == high then h_pwm      { 64 } end if
if pin_b2 == high then h_pwm      ( 128 } end if
if pin_b3 == high then h_pwm      ( 255 ) end if
end loop

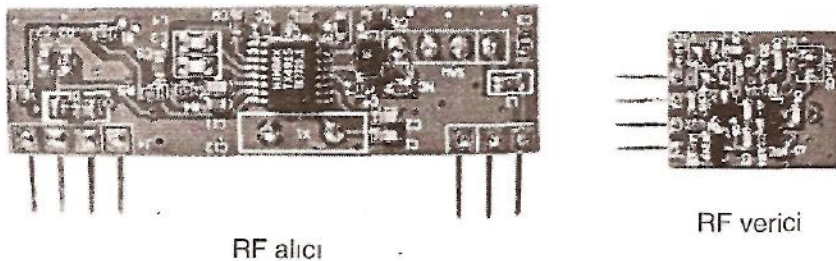
```

Donanımsal PWM sinyali üretirken h_pwm komutuna parametre olarak girilecek x değişkeninin değeri için $x = \text{oran} * 255 / 100$ formülü kullanılabilir. Örneğin iş saykıl'ının toplam saykıl'a oranı %12,5 olacağı zaman h_pwm alt programına parametre olarak $12,5 * 255 / 100 = 31,875$ yaklaşık 32 değeri yazılmalıdır. Tablo 17.1'de %12,5, %25, %50, %100 iş saykıl'ı oranına sahip PWM sinyali üretmek için gerekli değerler görülmektedir. Donanımsal PWM sinyali üretmek için döngü içinde devamlı h_pwm alt programını çalıştırmaya gerek yoktur. PIC H_pwm alt programı ile ayarlanan sürelerde donanımsal olarak devamlı PWM sinyalini üretecektir.

5.4 RF İLETİŞİM

RF (Radyo Frekans) kablosuz iletişim, radyo sinyallerinin kullanıldığı iletişim yöntemidir. Kablosuz iletişimin mesafesi küçük kit'ler ile 10-100 metre arasında değişirken büyük güçteki vericiler ile kilometrelerce alanı kapsayabilirler. Günümüzdeki bilgisayarlarda kullanılan kablosuz iletişim cihazlarının 54 Mbps hızında çalışabilmektedirler.

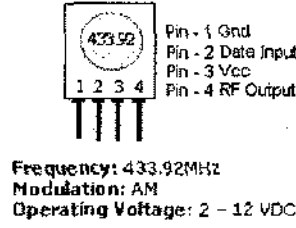
Bu bölümde amatör uygulamalarda kullanılabilecek ve açık alanda yaklaşık 50-100 metre mesafeli ve 1200 Bps hızında çalışabilen RF kit'ler üzerinde durulacaktır.



Şekil 5.5 RF alıcı ve verici kit'ler.

RF kit'lerin etkili mesafeleri yaklaşık 50-100 metre civarındadır. Bu mesafe temelde vericinin çıkış gücüne ve kullandığımız antene göre değişebilir. RF kit'lerin modülasyon frekansları genellikle 433.92 Mhz ve çıkış güçleri de 10 mW civarındadır.

TWS-434A RF Transmitter

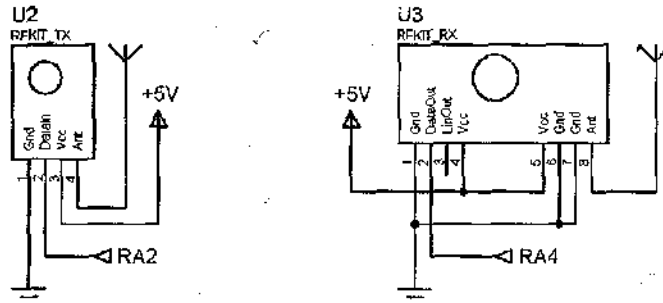


Şekil 5.6 RF verici kit'in pin bağlantısı.

RF kit'ler yurt dışından www.rentron.com adresinden temin edilebileceği gibi büyük şehirlerdeki elektronikçilerde de bulunmaktadır. Temin edilen RF kit ile bu bölümde kullanılacak kitler arasında pin bağlantıları farklı olabilir.

RF verici kit'lerin bazıları 12V'a kadar çalışmaktadır. Genellikle otomobillerin kapı kilidi kumandalarında bulunurlar. RF vericilerde kullanılmak için 12V'luk küçük piller bulunmaktadır. RF verici kit'in dijital veri girişi Data input pin'idir. RF verici 2 numaralı Data input pin'den aldığı dijital veriyi RF vericinin modülasyon frekansı ile modüle ederek 4 numaralı RF output pin'den radyo sinyalleri olarak havaya yayınlar.

Şekil 21.15'de görüldüğü gibi RF alıcı biraz daha büyüktür. Sekiz numaralı pin anten bağlantısı içindir ve antenin 30 cm olması yeterlidir. RF verici tarafından yayılan radyo frekans dalgaları RF alıcının 8 numaralı anten pin'inden bir anten vasıtası ile alınır. Alıcı içerisindeki elektronik devre alınan RF sinyalini demodüle ederek dijital veriye dönüştürür ve 2 numaralı dijital data output pin'inden dışarıya verir.



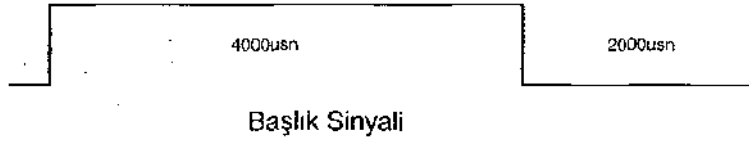
Şekil 5.7 RF alıcı ve verici kit'lerin PIC ile olan bağlantıları.

Şekil 5.7'da görüldüğü gibi RF alıcı PIC'in A4 pin'ine ve RF verici ise PIC'in A2 pin'ini kullanmaktadır. RF alıcı ve verici kit'ler düşük hızlarda direkt olarak bilgisayarın seri port'una

bağlanarak kullanılabilirler. Normal RF kitler yaklaşık olarak 2.4 Kbps ..10 Kbps arasında iletişim hızlarına sahiptir. RF kit'ler üzerinde bu bölümde yapılacak uygulamada seri iletişim protokolü her zaman uygun olmayabilir. Çünkü bit'ler arasında kayma olabilmektedir. Bu sebepten RF iletişimde İR iletişimde kullanılan protokolün benzeri kullanılacaktır.

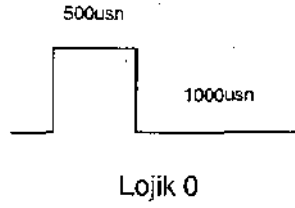
5.4.1 RF İletişim Protokolü

Başlık sinyali : Gönderilen sinyalin başına eklenen sinyaldir. Alıcı başlık sinyalinin özellikleri uygunsa sinyalin geri kalanını almaya başlar. Başlık sinyali 4 msn yüksek ve 2 msn düşük sinyalden oluşur.



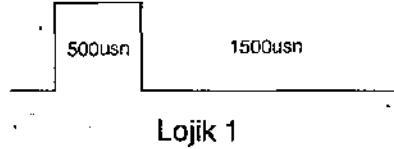
Şekil 5.8 RF iletişimde başlık sinyali.

Lojik 0'in ifade edilmesi : RF iletişimde lojik 0, 500 //sn yüksek sinyal ve 1000 //sn düşük sinyali ile ifade edilir.



Şekil 5.9 RF iletişimde lojik 0'in ifade edilmesi sinyali.

Lojik 1'in ifade edilmesi : RF iletişimde lojik 1, 500 //sn yüksek sinyal ve 1500//sn düşük sinyali ile ifade edilir.



Şekil 5.10 RF iletişimde lojik 1'in ifade edilmesi sinyali.

5.4.2 JAL'da RF İletişimin Kullanımı

rf. jal kütüphane dosyası radyo frekans (RF) kit'ler kullanarak bilgi alıp göndermek ile ilgili alt programları bulundurur. RF iletişimde kullanılan pin'ler ile ilgili tanımlamalar rfp. jal dosyasında yapılmaktadır. Bu kütüphane dosyasında bilgi alımı esnasında TMR0

kullanılmaktadır, rf. jal dosyasının içerisindeki alt programlar ve değişkenler aşağıdaki gibidir;

rf_trans : RF iletişim protokolüne göre 1 byte'lık bilgi göndermek için kullanılır. rf_trans alt programı pseudo tipinde olduğundan bilgi göndermek için direkt değer atanabilir.

rf_rcv (data) : RF iletişim protokolüne göre 1 byte'lık bilgi almak için kullanılır. rf_rcv alt programı bir fonksiyondur ve geçerli bir RF bilgi geldiğinde true değerini alır. Gelen bilgi parametre olarak kullanılan data değişkeninde tutulur.

RF iletişimde kullanılacak PIC pin'lerinin tanımları rfp.jal kütüphane dosyasından değiştirilebilir, rfp. jal dosyasını içeriği aşağıdaki gibidir;

var volatile bit	rf_buton	is pin_a3
var volatile bit	rf_buton_dir	is pin_a3_direction
var volatile bit	rf_rcv	is pin_a4
var volatile bit	rf_rcv_dir	is pin_a4_direction
var volatile bit	rf_rcv_led .	is pin_aO
var volatile bit	rf_rcv_led_dir	is pin_aO_direction
var volatile bit	rf_trns	is pin_a2
var volatile bit	rf_trns_asm	at port_a : 2
var volatile bit	rf_trns_dir	is pin_a2_direction
var volatile bit	rf_trns_led	, is pin_al
var volatile bit	rf_trns_led_dir	is pin_al_direction
rf_trns_led_dir	= output	
rf_trns_dir	= output	
rf_rcv_led_dir	= output	
rf_rcv_dir	= input	
rf_buton_dir	= input	

6 SONUÇLAR

Bu projede endüstride insan güvenliğini daha üst seviyelere çıkartmak için forklift robot prototipi üzerinde çalışılmıştır. Forklift robotun hareket kabiliyetini artırmak için tekerlek yerine paletler kullanılmıştır. Böylece forklift robotun olduğu yerde 360 derece dönebilmesi sağlanmıştır. Forklift robotun kontrolü ise hem uzaktan kumanda hem de bilgisayar vasıtasıyla sağlandığından bu iki noktanın da robotun üzerinde olmamasından dolayı robotun üzerinde kontrol amaçlı bir insan varlığına ihtiyaç kalmamıştır. Sanayide uygulanma alanları açısından ise nükleer tesisleri, ortam sıcaklığının insan için uygun olmadığı aşırı değerlerde çalışması gereken ortamları örnek gösterebiliriz.

Devrenin maliyeti açısından ise forklift robotun yerine getirdiği tüm işlevleri (motor hareketlerinin çift yönlü kontrolü, RF sinyalinin algılanıp değerlendirilmesi, PIC16F628A mikrodenetleyicisinde bulunan bir adet PWM modülünden iki adet motorun çift yönlü olarak sürülmesi, çatal hareketlerinin çift yönlü kontrolü) sadece bir tane PIC16F628A ile gerçekleştirilmiştir. Böylece maliyet minimum seviyede tutulmuştur.

Bu projenin genişletilmeye yönelik bir sonraki adımı çalışma ortamına yerleştirilen kameralardan elde edilen görüntünün bilgisayar vasıtasıyla işlenerek gerekli kontrol sinyallerinin üretilmesi ve üretilen bu sinyallerin yine bilgisayar vasıtasıyla projemizde hazırlamış olduğumuz bilgisayar iletişim verici devresi üzerinden forklift robota gönderilerek insan gereksinimi tamamen ortadan kaldırmaktır. Böylece bu tehlikeli endüstriyel ortamlardan insan tamamen izole edilmiş olacaktır.

Bu projenin yapım aşamasında edindiğimiz tecrübelerden bazıları şunlardır:

- ✓ Maliyeti minimum düzeyde tutabilmek için projenin gerçekleştirilmesi düşünülen mikrodenetleyicinin sahip olduğu özelliklere iyi bir şekilde hakim olmak gerektirir.
- ✓ Mikrodenetleyiciyi programlama dili olarak hem açık kaynak kodlu olması hasebiyle hem de PASCAL tabanlı bir yazılım diline sahip olduğundan JAL (**J**ust **A**nother **L**anguage) PIC programlama açısından kolay öğrenilebilir ve uygulanabilir.
- ✓ Motorun çift yönlü hareket etmesine ihtiyaç duyulan sistemlerde eğer PWM kontrol kullanılmayacaksa H köprüsündeki kaynak kısa devresi dezavantajına karşı röleler kullanılarak bir avantaj sağlanabilir.