

PARMAK-VURU PERFORMANSI ÖLÇÜM CİHAZI TASARIMI

Hüseyin YETER

Zonguldak Karaelmas Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalında

Yüksek Lisans Tezi

Olarak Hazırlanmıştır

ZONGULDAK

Haziran 2009

KABUL:

Hüseyin YETER tarafından hazırlanan “PARMAK-VURU PERFORMANS ÖLÇÜM CİHAZ TASARIMI” başlıklı bu çalışma jürimiz tarafından değerlendirilerek, Zonguldak Karaelmas Üniversitesi, Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak oybirliğiyle kabul edilmiştir 23.06.2009

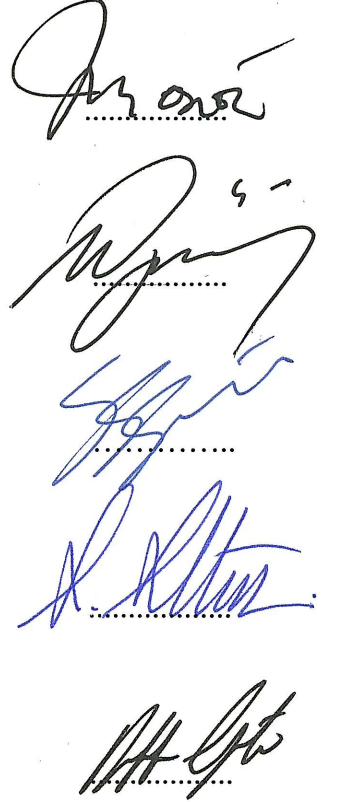
Başkan : Doç. Dr. Mahmut ÖZER (ZKÜ)
(Tez Danışmanı)

Üye : Yrd. Doç. Dr. Mustafa GÜMÜŞ (ZKÜ)
(2. Danışman)

Üye : Doç. Dr. Sırrı SUNAY GÜRLEYÜK (ZKÜ)

Üye : Doç. Dr. Ahmet ALTIN (ZKÜ)

Üye : Yrd. Doç. Dr. Rıfat HACIOĞLU(ZKÜ)



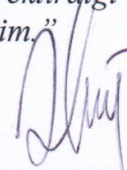
ONAY:

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım 20/7/2009

Prof. Dr. Kemal BÜYÜKGÜZEL
Fen Bilimleri Enstitüsü Müdürü



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”



Hüseyin YETER

ÖZET

Yüksek Lisans Tezi

PARMAK-VURU PERFORMANSI ÖLÇÜM CİHAZI TASARIMI

Hüseyin YETER

Zonguldak Karaelmas Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanları

Doç. Dr. Mahmut ÖZER

Yrd. Doç. Dr. Mustafa GÜMÜŞ

Haziran 2009, 97 sayfa

Parmağın vuru performansı ile dış uyarılara karşı tepki süreleri sinir ağlarının analizi ve beyin hasarları ile ilişkilendirilebilecek ritim bozukluklarının tespitinde önemli bir değişkendir. Bununla beraber değişik meslek gruplarına ait insanların parmak performanslarını ölçmek ve mesleki alanları ile performanslarını ilişkilendirebilmektedir. Bu verilerin toplanabilmesi, depolanabilmesi ve belirli standartların tespit edilebilmesi açısından parmak vuru ölçüm cihazlarının tasarımı ve geliştirilmesi önem arz etmektedir. Bu çalışmada hem karmaşık hem de standart test kriterleri seçilerek parmak vuru ölçümü yapabilen bir cihaz tasarımı gerçekleştirilmiştir. Koşullu ya da koşulsuz yapılan testlerde elde edilen sonuçlar cihazın hafızasında tutulmakta, mevcut lcd ekranda görüntülenebilmekte ve bilgisayar yardımıyla istenilen grafikler çizdirilebilmektedir. Tüm testlerin sonuçları ileri seviye analiz yapabilmek adına belirlenen bir adrese m.file olarak depolanabilmektedir.

Anahtar Sözcükler : Parmak vuru performansı, el performansı ölçümü, cihaz tasarımı

Bilim Kodu : 609.01.04

ABSTRACT

M.Sc. Thesis

FINGER-TAPPING PERFORMANCE MEASUREMENT DEVICES PLANNING

Hüseyin YETER

Zonguldak Karaelmas University

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

Thesis Advisors

Assoc. Prof. Mahmut ÖZER

Asist. Prof. Dr. Mustafa GÜMÜŞ

June 2009, 97 pages

With the warning against non-performance of the finger pulse neural network analysis of the reaction time and brain damage can be associated with a significant variable in determining the rhythm disorder. However, people belonging to different occupational groups and professional fields to measure the finger performance can be associated with the performance. This data can be collected, stored and will be determined and specific standards in terms of design and development of the finger pulse measurement device is important. In this study, the complex and the standard test criteria are selected to design a device that can do finger pulse measurement is true. Conditional or unconditional results obtained in the test device is kept in memory, with the help of available computer lcd screen display and the desired graphics can be drawn. All test results of advanced analysis to determine the name of an address can be stored as to m.file.

Key Words : Finger tapping performance, hand performance measurement, device design.

Science Code : 609.01.04

TEŞEKKÜR

Yazar hayatında dönüm noktası olan Mühendislik öğrenime başladığı çok zor zamanlarından bu günlere gelene kadar destek ve özverisini üzerinden hiç esirgemeyen, hazırlamış olduğu bu çalışmada göstermiş olduğu yol göstericiliği ve insani değerlerde mükemmel bir örnek olarak gördüğü Sayın Doç. Dr. Mahmut ÖZER (ZKÜ) 'e sonsuz teşekkürlerini bir borç bilir.

Hem spor alanında hemde akademik alanda göstermiş olduğu örnek davranışlarından ve desteğinden ötürü Sayın Yard. Doç. Dr. Mustafa GÜMÜŞ (ZKÜ) 'e teşekkürlerini bir borç bilir.

İlgi ve desteklerini her zaman yanında hissettiği babası Hüsnü YETER ve tüm ailesine teşekkürü borç bilir.

Bu çalışmaya göstermiş oldukları destek ve ilgiden ötürü Bakırköy Spor Klübü adına Sayın ZEYNEP ATAGÜL'e teşekkürü bir borç bilir.

İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET	iii
ABSTRACT.....	v
TEŞEKKÜR.....	vii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ.....	xi
ÇİZELGELER DİZİNİ.....	xiii
EK AÇIKLAMALAR DİZİNİ.....	xv
SİMGELER VE KISALTMALAR DİZİNİ	xvii
BÖLÜM 1 GİRİŞ.....	1
BÖLÜM 2 PARMAK VURU PERFORMANSI ÖLÇÜM CİHAZI	5
2.1 CİHAZIN TANITILMASI	5
2.2 TEST CİHAZININ DONANIMLARI	6
2.2.1 Butonlar.....	6
2.2.2 Sensörler.....	7
2.2.2.1 Ultrasonik Sensörler	7
2.2.2.2 Elektro-Optik Sensörler	11
2.2.2.3 Cisimden Yansımali Sensörler.....	13
2.2.3 Entegreler	19
2.2.3.1 Max232 Entegresi	19
2.2.3.2 Gerçek zaman saati DS1302	20
2.2.3.3 7805 Entegresi.....	21
2.2.3.4 24CXX Seri Eeprom	22
2.3 CXX SERİSİ SERİ EEPROMLAR.....	23
2.4 MİKRO DENETLEYİCİLER.....	29

İÇİNDEKİLER (devam ediyor)

	<u>Sayfa</u>
2.4.1 Mikro Denetleyici nedir?	29
2.4.2 Mikrodenetleyicilerin Kullanım Sebebi.....	29
2.4.3 PIC Nedir?.....	30
2.4.3.1 Neden PIC?	31
2.4.3.2 PIC Çeşitleri.....	32
2.4.3.3 PIC Bellek Çeşitleri.....	33
2.4.3.4 PIC Devresi.....	34
2.4.3.5 PIC18F452 Özellikleri ve Yapısı	34
2.4.3.6 PIC 18F452 Portlarının Fonksiyonları	37
2.4.3.7 PIC 18F452'nin Özel Fonksiyonları	38
2.4.3.8 PIC 18F452'nin RAM Belleği	40
2.4.4 PIC18F452'nin Besleme Uçları ve Beslenmesi	43
2.4.4.1 PIC18F452'nin Reset Uçları.....	43
2.5 SERİ HABERLEŞME	43
2.5.1 Seri Haberleşme Metodları	44
2.5.1.1 Asenkron Veri İletimi.....	44
2.5.1.2 Senkron Veri İletimi.....	45
2.5.2 İletim Yönlerine Göre Seri İletişim Modları.....	45
2.5.3 USART Birimi Ve Özellikleri.....	46
2.5.3.1 Sinyal Hızının Belirlenmesi.....	47
2.5.3.2 USART İle Asenkron Veri Göndermek	48
2.6 LCD	49
2.7 YAZILIMLAR	51
2.7.1 Mikrodenetleyici Yazılımı	51
2.7.2 Matlab Yazılımı.....	53
BÖLÜM 3 SONUÇ VE ÖNERİLER	61
KAYNAKLAR	63
EK AÇIKLAMALAR A YAZILIMLAR.....	67
EK AÇIKLAMALAR B CİHAZIN FOTOĞRAFLARI.....	93
ÖZGEÇMİŞ	97

ŞEKİLLER DİZİNİ

<u>No</u>	<u>Sayfa</u>
1.1 Dijital ve Analog parmak vuru testi cihazları	2
2.1 Test cihazının blok şeması.	5
2.2 Butonlarda oluşan arkların görünümü.	6
2.3 Ultrasonik ses dalgalarının yayılımı	8
2.4 Plaroid ve Hitechnic sensör tipi.....	9
2.5 Ultrosonic sensörün tipik ışıma paterni	9
2.6 Ultrosonic sensörün içyapısı	10
2.7 Elektromanyetik ışıma radiation, dağılımı Spektrum.....	11
2.8 Spektral dağılım(standartlaştırılmış)	12
2.9 Cisimden yansımali sensör kapatma eğrisi	14
2.10 Cisimden yansımali OU tip sensörün anahtarlama eğrisi.	14
2.11. Cisimden yansılmalı sensörün algılama çeşitleri.....	15
2.12 Cisimden yansılmalı sensörün algılama mesafesinin değışimi	15
2.13 Cisimden yansımali sensor yayılım alanları	16
2.14 Sharp gp2d120 mesafe algılama sensörü.....	18
2.15 PC seriportu ile PIC'in MAX232 ile birbirine bağlanması.....	20
2.16 DS1302 Entegresi.	20
2.17 7805 entegresi bağlanyı şeması	21
2.18 24 XX serisi eepromların pin şeması.	23
2.19 Data klok diyagramı.....	23
2.20 Byte yerleşim diagramı	24
2.21 Carry byte ve adres bloğu	24
2.22 Grup2 eeprom byte ve adresleme bloğu	25
2.23 Grup2 eeprom byte adreslemesi	25
2.24 Grup2 eeprom carry byte ve adres bloğu.	26
2.25 Seri eeprom okuma adreslemesi.	26
2.26 Seri eepromda ikinci okuma adreslemesi.....	27
2.27 Grup1 eeprom bilgi yazdırma adres bloğu.....	27

ŞEKİLLER DİZİNİ (devam ediyor)

<u>No</u>	<u>Sayfa</u>
2.28 Grup2 eepromda okuma adres bloğu.	28
2.29 Grup2 eepromda currnet read.	28
2.30 Grup2 seri eepromda yazma adresleme	28
2.31 PIC 18F452'nin Bacak Tanımlamaları	35
2.32 Asenkron veri iletişimin basit bağlantı şeması.....	44
2.33 Senkron seri iletişimin basit bağlantı şeması	45
2.34 LCD basit bağlantı şeması	50
2.35 Program Dallanması (algoritma).	52
2.36 Parmak vurular arası geçen süre ve yükseklik grafikleri.....	55
2.37 Tek parmak vuru grafiği	56
2.38 Çift parmak vuru grafiği	56
2.39 Hassas ölçüm grafiği.....	57
B.1 Cihazın fotoğrafları	95

ÇİZELGELER DİZİNİ

<u>No</u>	<u>Sayfa</u>
2.1 Ultrosonic sensörün teknik bilgileri	9
2.2 DIN 5031 'e göre optik spektrum sınıflandırması.....	12
2.3 PIC 18F8452'nin Bacaklarının Fonksiyonları	35
2.4 PIC16F452 Ram Bellek Haritası	41
2.5 Osilatör Çeşitleri ve Özellikleri.....	42
2.6 Frekansa Göre Kondansatör Seçimi	42
2.7 Senkron haberleşme frekans tablosu.	47
2.8 Asenkron haberleşme frekans tablosu	48
2.9 LCD Bacak bağlantıları	50
2.10 Elde edilen teset sonuçları.....	55
2.11 Yapılan ölçümler ve alınan sonuçlar.	59

EK AÇIKLAMALAR DİZİNİ

	<u>Sayfa</u>
A.1 : Yazılımlar	67
B.1 : Cihazın Fotoğrafları	93

SİMGELER VE KISALTMALAR DİZİNİ

KISALTMALAR

I	: işaret parmağı (index finger)
ITI	: iki parmak vurusu arasında geçen süre (intertapping interval)
L	: serçe parmak (little finger)
M	: orta parmak (middle finger)
PDF	: taşınabilir doküman formatı (portable document format)
PET	: Pozitron emisyon tomografisi (positron-emission tomography)
R	: yüzük parmağı (ring finger)

BÖLÜM 1

GİRİŞ

Kişilere özgü birçok ritimler bulunmaktadır. Bunlar kan basıncı, kalp atışı, göz kırpması gibi ritimlerdir. Bu ritimlerin en önemlilerinden olan parmak vuru performansı kişilere ait sportif faaliyetlerin, sanatsal yeteneklerin ve bazı sinirsel hastalıkların tespitinde kullanılan bir parametredir. Nöropsikolojik rahatsızlık ile yaş arasındaki bağlantı parmak vuru performansı ile yapılan testlerde birbiriyle ilişkili olduğu uzun zaman önce saptanmıştır (Bornstein et al. 1987). Erkeklerin motor hızının ve vuru şiddetinin kadınlardan daha iyi olduğu görülmüştür. (Leckliter and Matarazzo 1989) Dash ve Telles (1999) parmak vuru testini motor hızını belirlemek için kullanmışlardır. 10 gün boyunca Yoga yapan çocukların ve 30 gün boyunca Yoga yapan erişkinlerin performanslarında belirgin bir artış gözlemlemişlerdir. Volkow vd. (1998) parmak vuru testi ile dopamin D2 reseptörleri ve motor fonksiyonları arasında güçlü bir ilişki olduğunu bulmuştur. Parkinson hastalığının ilerleme seyrinin belirlenmesi zor olmasına rağmen (Rao et al. 2003) ve Muir vd. (1995) ve Jobbágy vd. (1997) bu hastalıkta motor semptomlarının önemini tahmin etmek için kullanmışlardır. Genellikle tercih edilen elin performansı diğerine göre çok yüksektir. Beyin hasarını takiben hasarın ters tarafındaki el performansında kötüleşme eğilimi olur (Spreen and Strauss 1998).

Parmak vuru testi 19. yüzyıldan itibaren motor yeteneklerinin belirlenmesinde uygulanmaya başlanmıştır (Jobbágy et al. 2005). Hollingworth (1914) kadınlarda menstrüasyon etkisini karakterize etmek için elektriksel bir sayıcı kullanarak denemeler gerçekleştirmiştir ve bundan itibaren vuru testi ataksinin belirlenmesinde (Notermans et al. 1994), akut inme hastalarının tedavi sürecinde (Heller et al. 1987), alkolik Korsakoff Sendromu (Welch et al. 1997) hastalarının testinde, Alzheimer hastalığının belirlenmesinde (Ott et al. 1995) ve kol motor fonksiyonlarının belirlenmesinde (Giovannoni et al. 1999) yaygın olarak kullanılmaya başlanmıştır. Horton (1999) zekâ seviyesi daha yüksek deneklerin baskın olmayan eli ile yaptığı parmak vuru testlerinde daha iyi nöropsikolojik test skoru elde ettiklerini belirlemiştir.

Örneğin profesyonel bir kemancı 3 oktav ölçeği 2 saniyeden daha kısa sürede, parmaklarını 80 milisaniye civarında aralıklarla basarak düzgün bir şekilde çalabilir (Arunachalam et al. 2005).

Nicholson ve Kimura (1996) erkeklerin kadınlara oranla tek parmakla daha hızlı vuru yaptıklarını gözlemiş ve sonuç olarak bunun erkeklerde ergenlik döneminde beyaz kasların kadınlardakine göre daha fazla artmasına bağlamışlardır. Ortalamada erkeklerin yaşlanmayla vuru test performanslarında ciddi bir düşme görülmezken kadınlarda yaşlanmayla performansta azalma gözlenmiştir (Ruff and Parker 1993).

Parmak vuru hızını kaydetmek için iki ana prensipte çalışan cihazlar bulunmaktadır. Bunlardan birincisi şekil 1.1 'in sol tarafında bulunan dijital parmak vuru ölçüm cihazıdır. Christianson et al. 2004). Diğeri ise şekil 1.1 'in sağ tarafında bulunan mekanik bir tuş ile sayım yapan bir cihazdır (Mitrushina et al.1989).



Şekil 1.1 Dijital ve Analog parmak vuru testi cihazları (Mitrushina et al. 1989).

Parmak vuru testinin çeşitli tipleri pratikte el vuru testi, bir veya çok parmaklı vuru testi, tek el vuru testi, çift el vuru testi olarak uygulanmaktadır. Bu testler serbest vuru veya bir zamanlama sinyaline bağlı önkoşullu test olarak sınıflandırılır. Şüphesiz işaret (I) ve orta (M) parmaklar, yüzük (R) ve serçe (L) parmaklara göre günlük işlevlerde daha baskın kullanılırlar. Raj ve Marquis (1999) hesap makinelerinde hangi parmağın daha yaygın kullanıldığını tespit etmek için yaptıkları araştırmalarda I parmağının %80'den daha fazla kullanıldığını, M, R ve L parmaklarının kullanımının çok daha az olduğunu gözlemlemişlerdir. I, M, L, R parmakları için tek parmak testi ve bu parmak çiftleri için çift parmak testini gerçekleştirmişlerdir. Çalışmalarında denek olarak parmak vuru performansı ile bağlantılı herhangi bir eğitim almamış genç erkekleri seçmişlerdir. Tek parmak vuru testinin sonucunda en iyi ritmin I

parmağıyla yapılan testlerde olduğu ve bunu sırasıyla M, L ve R parmakların takip ettiğini belirlemişlerdir. Çift parmak vuru testinde ise en iyi ritmin I-M kombinasyonu ile ölçüldüğünü ve en kötü kombinasyonun da R-L olduğunu gözlemişlerdir. Aoki vd. (2005) piyanistler ve sıradan deneklerle yaptıkları testlerde, I-M kombinasyonu ile yapılan vurularda piyanistlerin sıradan insanlara göre daha ritmik vuru yaptıklarını gözlemlemiş ve bunu eğitimle ilişkilendirmişlerdir. Ancak R-L kombinasyonu ile yapılan testlerde belirgin bir fark görememişlerdir ve R-L parmaklarının eğitimle performansında belirgin bir iyileşme olmadığını belirtmişlerdir.

Diğer tomografi tekniklerinden farklı olarak anatomik detaydan daha çok metabolik aktiviteyi göstermeye yönelik olan Pozitron Emisyon Tomografisi (PET) tekniği kullanılan bir diğer çalışmada, Aoki vd. (2004) R-L kombinasyonu ile yapılan parmak vuru testinde, I-M kombinasyonu ile yapılan parmak vuru testine ve tek parmak vuru testine göre korteks ve korteks altı bölgelerde aktivasyonun daha fazla olduğunu kanıtlamıştır. Ayrıca sıradan insanlarda parmaklarla uygulanan kuvvetin parmağa göre farklılık gösterdiğini belirleyen çalışmalar da vardır. Kinoshita vd. (1996), Radwin vd. (1992), ve Zatsiorsky vd. (1998, 2000) aynı yükseklikten vurulan parmakların kuvvetini ölçmüşlerdir. Bunun sonucunda I ve M parmakların, R ve L parmaklarına göre vuru şiddetinin daha fazla olduğunu görmüşlerdir.

Hareketlerin harici bir sinyalle tetiklenmesi (önkoşullu vuru) veya kişinin kendisi tarafından kontrol edilmesi (serbest vuru) motor kontrolün çift yönlülüğünün varlığı ile sağlanır (Bard et al. 1991, Bard et al. 1992, Bizzi et al. 1972, Goldberg 1985, Paillard 1948, Romo et al. 1992). Paillard ilk araştırmalarında (Paillard 1948), serbest vuru testinde el topuğunun kalkışının parmak hareketinden önce olduğunu fakat önkoşullu vuru testinde parmak hareketinin el topuğunun kalkışından önce olduğunu göstermiştir. Paillard'a göre (1948) önkoşullu vuruda merkezi motor komutları her iki sinire aynı anda iletilir, parmağın önce hareketinin sebebi iletim yollarının farklı olmasındandır. Serbest vuruda ise el topuğunun hareketi parmağın her iki koldan geri dönen sinyalin merkezi karşılaştırmaya ulaşmasını beklemesinden dolayı önce olur. Paillard'ın sonuçları yakın zamanda yapılan çalışmalarda tekrarlanmıştır (Bard et al. 1991, Bard et al. 1992).

Serbest vuru ve önkoşullu vuru arasındaki fark serbest vuru testinde hareket zamanının önceden bilinmesinden kaynaklanır. Önkoşullu testte işaret periyodik olarak verilirse serbest vuruya benzer bir durum ortaya çıkar (Billon et al. 1996). Fraisse vd. (1958) senkronizasyon

testi sırasında oluřan hatanın iletim zamanı farklılıklarından kaynaklandığını belirtmiştir. Senkronizasyon hatası vuru ile işaret arasında geçen süre ölçülerek hesaplanır. Daha sonra Fraisse (1980) deneklerin parmak ve ayaklarını senkronizasyon işaretine göre vurduklarında her iki motor cevabının da senkronizasyon işaretinden önce olduğunu görmüřtür ve senkronizasyon hatasının genliğinin sinirin merkezi karşılařtırıcıya olan uzaklığının bir fonksiyonu olduğunu göstermiştir. Topuk vurusu parmak vurusundan yaklaşık 20 ms daha öncedir.

Bundan önceki çalışmada (Taner ÖZDEMİR ZKÜ 2009) parmak vuru testini gerçekleřtirmek amacıyla yeni bir bilgisayar yazılımı, ‘ ‘ Mamut Hand Performance Test ‘ ‘ tasarlanmıştır. Bizim çalışmamızda ise parmak vuru performansını ölçmek amacıyla bir cihaz tasarımı yapılmıştır. Tasarlanan cihazda test kriterleri belirlenerek ön koşul seçimleri yapılabilmekte, deneylere ait kişisel bilgiler girilebilmektedir. Yapılan bir test sonucunda istenilen sonuçlar cihaz üzerinde bulunan lcd ekrana yazdırılabilmekte ve veriler saklanabilmektedir. Bu veriler test sonuçları ile beraber bilgisayara aktarılabilmekte, tasarlanan bilgisayar programı ile istenilen grafikler çizdirilebilmektedir.

BÖLÜM 2

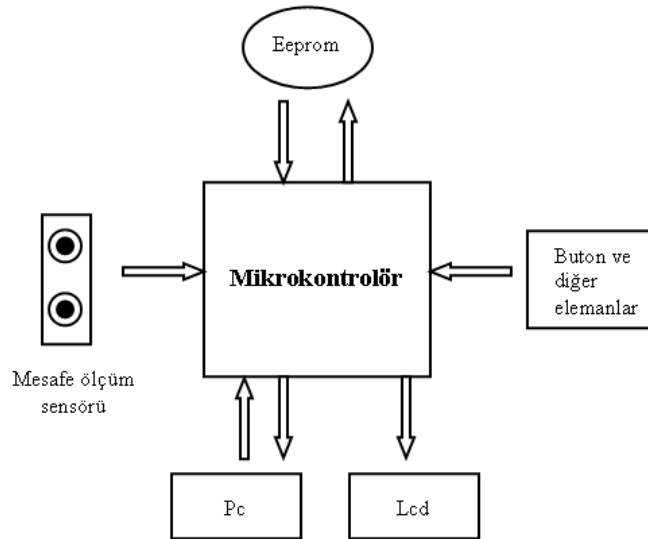
PARMAK VURU PERFORMANSI ÖLÇÜM CİHAZI

Bu bölümde parmak vuru testinin yapılabilmesi ile ölçülen değerlerin kaydedilip gerekli hesaplamaların yapılabilmesini sağlayan cihaz tanıtılmaktadır.

2.1 CİHAZIN TANITILMASI

Tasarlanan elektronik devre ve ölçüm sensörleri ile istenilen uyarı tiplerine göre parmak vuru tepkisi ve performans verileri alınmaktadır. Bu veriler devrede bulunan harici bir eeproma kaydedilmektedir. Yapılacak olunan değerlendirme ve matematiksel istatistik basamaklarına göre istenilen değerler hesaplanmakta ve fonksiyonları çizdirilmektedir.

Genel ölçüm ve test basamaklarına göre tarafımızdan belirlenmiş olan cinsiyet, yaş, uyarı tipi, uyarı süresi ve diğer alanlara göre performans istatistiği oluşturmaktır. Test cihazının blok şeması Şekil 2.1’de verilmiştir.



Şekil 2.1 Test cihazının blok şeması.

2.2 TEST CİHAZININ DONANIMLARI

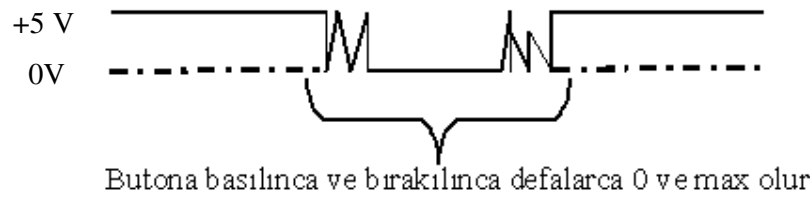
Devremizde kullanılan donanımları incelersek;

1. Butonlar
2. Sensörler
3. Entegreler
4. Mikroişlemciler
5. Lcd

2.2.1 Butonlar

Devremizde butonları iki ana amaç için kullanmaktayız. Bunlardan birincisi parmağın uyarı aldıktan sonra yapmış olduğu hareketin durumunu ve sayısını ölçmemiz için ikincisi ise tasarlamış olduğumuz programın yapılandırılmasında, deneklere ait bilgilerin girilmesinde ve test seçeneklerinin belirlenmesinde kullanmaktayız.

Butonların her birine birer uçları +5 V'a bağlanmış ve diğer uçları sırasıyla 330Ω değerindeki R1, R2, R3 dirençleri ve bu dirençlere paralel 1μF değerindeki C1, C2, C3 kapasitörleri ile toprağa bağlanmıştır. Bunun nedeni, butonlara 1 defa basıldığında pic'in pinlerine ulaşan gerilim seviyesinin defalarca 0-5V arasında dalgalanmasıdır. Butonlara basıldığında pic'in pinlerindeki olası gerilim eğrisi aşağıdaki gibi olmaktadır.



Şekil 2.2 Butonlarda oluşan arkların görünümü.

Butonlarda basma ve bırakma anında esnasında ark oluşur, bu da gerilim dalgalanmalarına neden olur. Bu dalgalanmalar esnasında birden fazla “0” seviyesine iniş olmaktadır.

Butonlarda meydana gelen bu elektriksel olayı en aza indirmek için buton uçlarına bir kondansatör ve paralel direnç bağlamak donanımsal çözümlerden bir tanesidir.

2.2.2 Sensörler

Devremizde parmak mesafesinin ölçülmesinde kullanılmak üzere birden fazla sensör denenmiştir. Bunlar aşağıda açıklanmaktadır.

Mekanikteki en temel ölçü uzunluk ölçüsüdür. Konum, hareket, yer değişimi terimleri birbirine çok yakın durmaktadır. Konum algılayıcı (Position Sensor) ya da hareket transdüseri (Motion Transducer) terimlerine sık sık rastlanmaktadır. Yer değişimi transdüseri (Displacement Transducer), teknik olarak en doğru ifade sayılabilir. Temel olarak lineer ve açısal yer değişimi algılayıcı olarak ikiye ayrılırlar.

Yer değişim algılayıcıları ölçme teknikleri açısından aşağıdaki gibi sınıflandırılabilir.

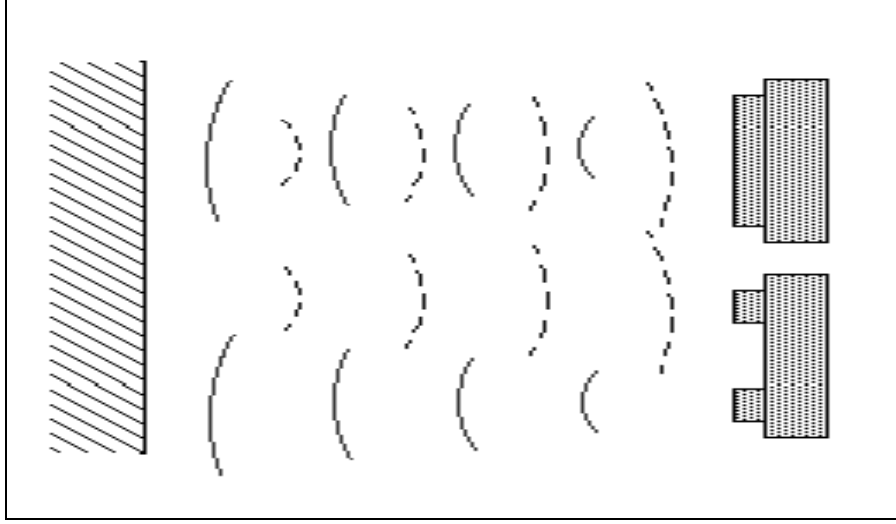
1. Ultrasonik
2. Elektro-Optik
3. Relüktans
4. Potansiyometrik
5. Strain-Gage
6. Endüktif
7. Açısal ve Doğrusal kodlar
8. Kapasitif
9. Konum Şalterleri

2.2.2.1 Ultrasonik Sensörler

Ultrasonik sensörler genellikle robotlarda engellerden kaçmak, navigasyon ve bulunan yerin haritasını çıkarmak amacıyla kullanılmaktadır. Bu türden çalışmaları ilk olarak, Polaroid firması ultrasonik sensörü kullanarak ve bunu bir aletin içine koyup kamera uzaklığını anlayan sistem geliştirmiştir (URL-1 2009).

Çalışma Prensipleri:

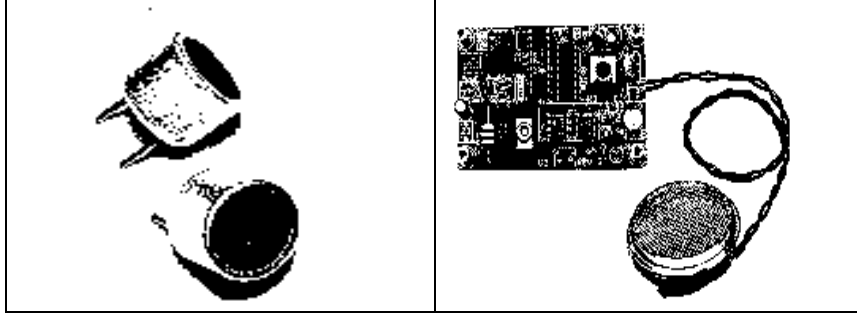
Ultrasonik uzaklık sensörü, piezoelektrik transducerden gelen 40khz ultrasonik sesin kısa darbelerini yayarak çalışmaktadır. Ses enerjisinin küçük bir kısmı sensörün önündeki cisimlerden yansıyarak dedektöre yani farklı bir piezoelektrik transducere gelir. Şekil 2,3’de Alıcı yükselteci yansıyan işareti (ekoları) sinyal dedeksiyon sistemine veya mikrokontrolöre gönderir. Sinyalin havadaki hızına bağlı olarak mikrokontrolör, cisimlerin ne kadar uzakta olduklarını zamanlama prosesi koşarak belirler.



Şekil 2.3 Ultrasonik ses dalgalarının yayılımı (URL-1 2009).

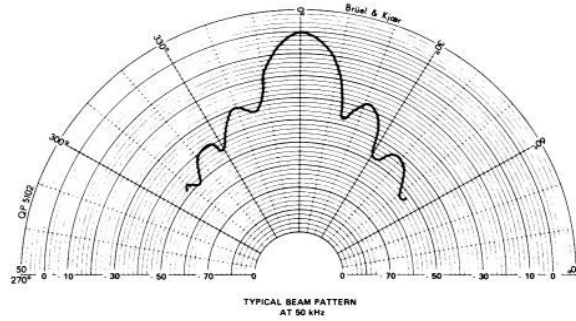
Ultrasonik uzaklık sensörleri fiziksel olarak iki çeşitte piyasada bulunmaktadır. Buna rağmen temel işlevleri aynıdır.

- Polaroid sensör tipi: Ultrasonik ses dalgalarının yayılması ve algılanması tek bir piezoelektrik transducer tarafından yapılır.
- Hitechnic sensör tipi: Ultrasonik ses dalgalarının yayılması verici transducer, dalgaların algılanması ise alıcı transducer tarafından yapılır. Bu tipteki uzaklık dedeksiyon işleminde 2 tane transducer kullanılır.



Şekil 2.4 Plaroid ve Hitechnic sensör tipi (URL-1 2009).

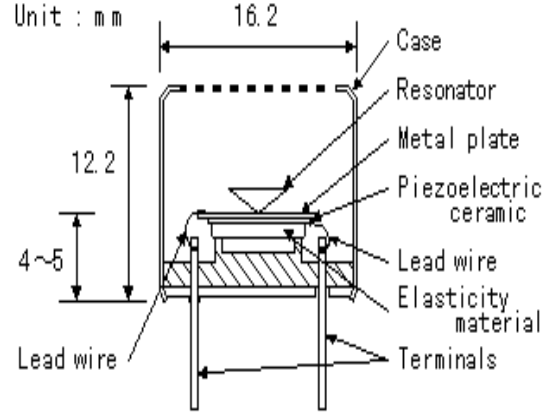
Ultrasonik sensörlerde yansıyan işaretin dönme süre bilgisine göre işlem yapıldığında bazı anlaşılmazlıktan kaynaklanan yorum hatası yapılabilmektedir. Örneğin sensörün yüzü kendine daha yakın düz bir cisim ile paralel olsun. Bu cismin arkasında ise yansıtıcı yüzeyi olan çok geniş bir duvar olduğunda, sensör tarafından algılanan bilgi sensörün önünde bulunan yakın cisme göre yorumlanır. Buna rağmen bazen yansıyan işaretin dönme süre bilgisi anlamlı cisimi algılamamızda bizi yanıltabilir. Eğer cismin bulunduğu yüzey, sensörün gerçek yüzeyi ile açısal olarak ölçeklendirilirse, informasyon bilgisi 30 derece konisi şekil 2,5'in içinde bulunan en yakın noktaya göre kaydedilir.



Şekil 2.5 Ultrasonic sensörün tipik ışınma paterni (URL-1 2009).

Çizelge 2.1 Ultrasonic sensörün teknik bilgileri (URL-1 2009).

TEKNİK BİLGİLER		
Rezonans Frekansı (KHz)		40
Ses Basınç Düzeyi (dB)		115<
Hassasiyet (dB)		-64<
Ölçüler (mm)	Yarıçap	16,2
	Yükseklik	12,2
	Terminal Aralığı	10,0



Şekil 2.6 Ultrasonic sensörün içyapısı (URL-1 2009).

* Ultrasonik Uzaklık Sensörün Kullanım Avantajları:

a.Kontaksız Ölçüm

Hedef cismi dokunmadan havayı kullanarak nispeten geniş mesafelerden ölçer.

b.Cisim Menzilemle

Cisim mesafesini çoğunlukla görünüş veya yakınlık analizine göre ölçebilir.

c.Uzaklıkla. Orantılı. Çıkış

Sensörün elektriksel çıkışları ölçülen hedef uzaklığıyla orantılı veya bu uzaklığa bağlıdır.

d.YüksekÇözünürlük

Ultrasonik sensörler hedef cisimle ilgili informasyonu doğru ve ince farkları gösterebilme yeteneğine sahiptir.

e.Hedefin Optik Karakteristiklerinden Etkilenmeme

Ultrasonik sensörler algılaması ortamın ışık seviyesinden, hedefin renginden veya hedefin optik geçirgenlik/yansıtıcılık özelliklerinden etkilenmez.

f.Hassasiyet

Büyük veya küçük cisimleri algılayabilir

Bizim uygulamamızda yakın mesafelerdeki çözünürlüğünün ve stabilizasyonun buna bağlı olarak hassasiyetinin verimli olmadığı tespit edilmiştir.Kapalı bir hacimde yapılan testlerde yansıtıcı özelliği iyi olan duvarlardan gelen sinyallerin parmak dan gelen sinyalleri etkilediği tespit edilmiştir.Bu sensör daha çok duran cisimlerin tespitinde daha verimli sonuçlar vermektedir.Örnek olarak sıvı seviye ölçüm devrelerinde , park sensörü devrelerinde ve duran cisim mesafe ölçümü devrelerinde kullanılabilir.

2.2.2.2 Elektro-Optik Sensörler

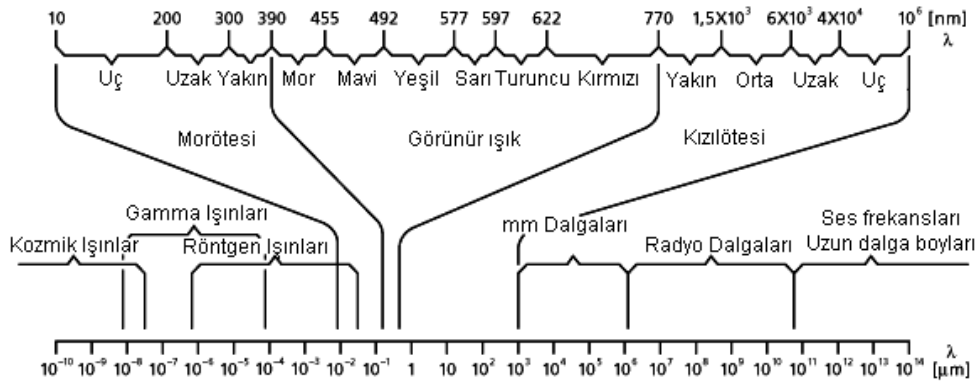
İndüktif ve kapasitif sensörlere ek olarak, günümüz otomasyon teknolojisinde opto-elektronik sensörler gittikçe daha önemli olmaktadır. Bunlar, dokunmasız makine hareketlerini algılama ve daha önemlisi makinelerde ve fabrikalarda farklı ürünleri emniyetli olarak algılama olanağı sağlar.

Optik sensörler yüksek performansları ve gittikçe küçülen tasarımları ile ivme kazanmaktadır. Çünkü, büyük olmalarından dolayı indüktif ve kapasitif sensörlerle çözümlenemeyen uygulamalarda kullanılabilirler.

Büyük indüktif ve kapasitif sensörlerde, sensörle hedef cisim arasındaki en uzun mesafe 60-100 mm dolaylarındadır. Fakat optik sensörler küçük boyutlarda bile birkaç metrelik alanı kontrol edebilir. Bu sensörler üç farklı algılama ilkesine göre sınıflandırılabilir: karşılıklı sensörler, yansıtıcı sensörler ve cisimden yansımali sensörler. Her algılama ilkesi aşağıda ayrıntılı anlatılan farklı özelliklere sahiptir. Bu notlarda farklı algılama ilkeleri, bu sistemlerin avantajları ve dezavantajları, uygulamaya yönelik uygun sensör seçim kriterleri anlatılmıştır.

Optik Algılamanın Temel İlkesi

Tam olarak opto-elektronik ne demektir? Bu optik ve elektronik kelimelerinin birleştirilmesi ile oluşturulmuştur. Anlamı, dokunmaksızın bir cismi ışık (optik) yardımıyla algılama, sonra elektronik olarak değerlendirme ve sinyale dönüştürme demektir.



Şekil 2.7 Elektromanyetik ışınım radiation, dağılımı Spektrum (URL-1 2009).

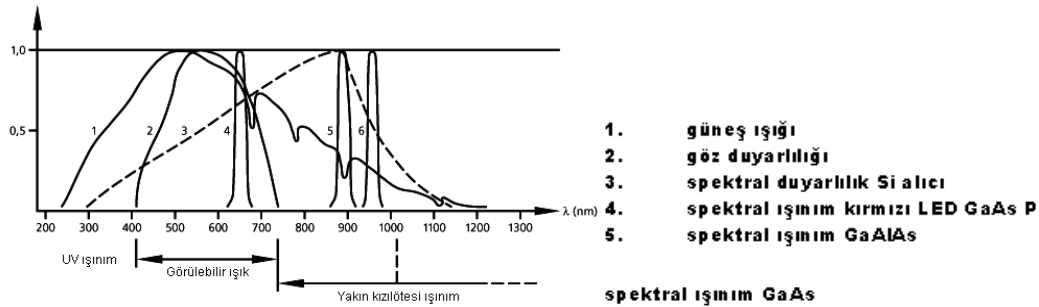
Bir cismi algılamak için fotosele gerekli olan ışık 1 nm ile 10 nm arasındaki elektromanyetik ışınım aralığındaki dalgalardan oluşur Bu aralık UV ışık, görülebilir ışık (insan gözü ile) ve IR ışık bölgelerine ayrılmıştır.

Çizelge 2.2 DIN 5031 'e göre optik spektrum sınıflandırması (URL-1 2009).

Dalga boyu aralığı	Işınım tanımlama
100 nm - 280 nm	UV - C
280 nm - 315 nm	UV - B
315 nm - 380 nm	UV - A
380 nm - 440 nm	Açık mor
440 nm - 495 nm	Açık mavi
495 nm - 558 nm	Açık yeşil
558 nm - 640 nm	Açık sarı
640 nm - 750 nm	Açık kırmızı
750 nm - 1400 nm	IR - A
1.4 mm - 3.0 mm	IR - B
3.0 mm - 1000 mm	IR - C

Bölgeler arasındaki geçiş ve görülebilir ışığın renkleri arasındaki geçiş süreklidir (gökkuşağı). Genellikle ışık kaynağı olarak dalga boyu 880 nm olan kızıl ötesi (infrared) ışık kullanılır.

Fakat bazı özel durumlarda dalga boyu 660 nm olan kırmızı ışıkta kullanılır.



Şekil 2.8 Spektral dağılım(standartlaştırılmış) (URL-1 2009).

Kızıl ötesi ışık olası dış kaynakların etkilerine karşı olabildiğince çok bağışıklık kazandırmak için çeşitli nedenlerle kullanılır.

Birincisi, alıcı olarak kullanılan transistör en yüksek duyarlılığa kızıl ötesi ışıkta sahiptir.

İkincisi, çok küçük toz parçalarının çapından daha uzun dalga boyu olan ışığın hiçbir sorunla karşılaşmadan bu parçacıkları geçmesi olayından yararlanır. Kirlenme ve toza karşı koruma sağlamak uzun dalga boylu ışınım (UV değil IR) kullanımının nedeni budur.

Üçüncüsü, kızıl ötesi ışık kullanımıyla sensörler görülebilir aralıktaki dış ışık kaynaklarından daha az etkilenir.

2.2.2.3 Cisimden Yansımali Sensörler

Cisimleri opto-elektronik olarak algılamanın diğer bir yöntemi ise cisimden yansımali tip sensörlerdir. Burada da verici ve alıcı aynı kılıf içine yerleştirilmiştir.

Fakat cisimden yansımali sensörler, bir prizmatik yansıtıcı veya yansıtıcı kağıttan yansıyan ışıkla değil de hedef cisimden yansıyan ışıkla çalışır. Bu, indüktif ve kapasitif yaklaşım anahtarları çalışma ilkelerine benzeyen tek opto-elektronik algılama ilkesidir. Onlarda cismi doğrudan algılar:

Cisim var (yansıma var) yaklaşım anahtarı algılar

Cisim yok (yansıma yok) yaklaşım anahtarı algılamaz.

Cisimden yansımali sensörlerin önemli avantajları şunlardır:

Monte edilecek sadece bir sensör

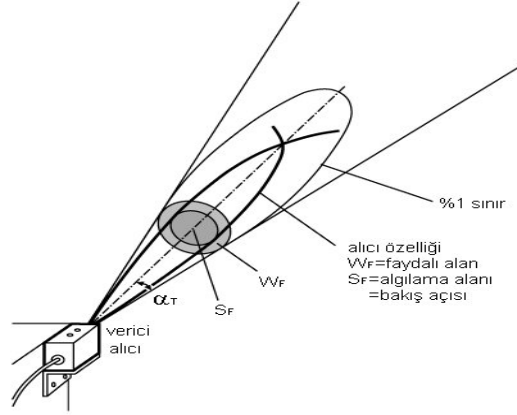
Yanlış ayarlama ve yansıtıcı kirlenmesi yok

Şeffaf cisimler karşılıklı ve yansıtıcı sensörlerden daha iyi algılanabilir.

Alıcının doğru duyarlılık ayarı, şeffaf cisimlerden az miktar'daki yansımaların değerlendirilmesini sağlar

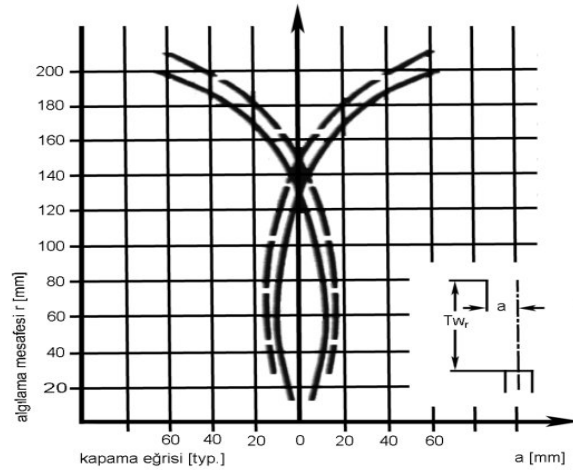
Cisimden yansımali sensörlerin de bazı dezavantajları vardır: cisimden yansıyan ışığın değerlendirilmesi ve algılaması nedeni ile cisimin algılaması büyük oranda cisim yüzeyinin özelliklerine bağlıdır(pürüzsüz, yansıtıcı beyaz gri siyah genel olarak cisimlerin yansıtma oranının daha düşük olmasından dolayı, verici ile alıcı arasındaki ışının açıkça kesildiği

(alıcıda kızıl ötesi ışık var /yok) karşılıklı ve yansıtıcı sensörlere göre olası maksimum algılama mesafesi daha kısadır. Cisimden yansımali sensörde, alıcıya ulaşan kızıl ötesi ışık anahtarlama noktasına doğru artar, yani: önce çok az, az, biraz, biraz, daha ve anahtarlama noktası.



Şekil 2.9 Cisimden yansımali sensör kapatma eğrisi (URL-1 2009).

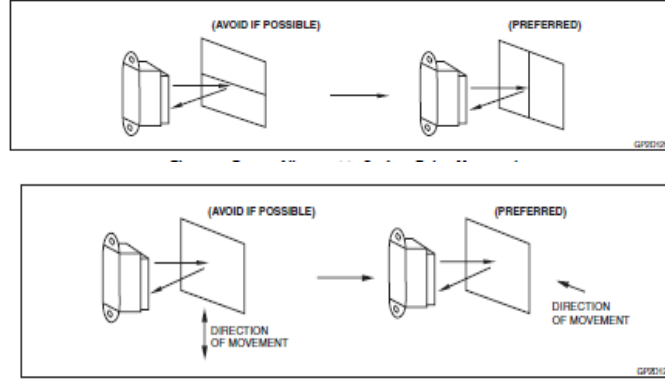
Şekil 2.9 cisimden yansımali sensörün tipik alıcı eğrisini gösterir. Şekilde görülen eğriler hedef cismin (gri kodak kartının beyaz arka tarafı), indüktif ve kapasitif yaklaşım anahtarlarındakine benzer olarak yandan veya önden yaklaşması ile elde edilmiştir.



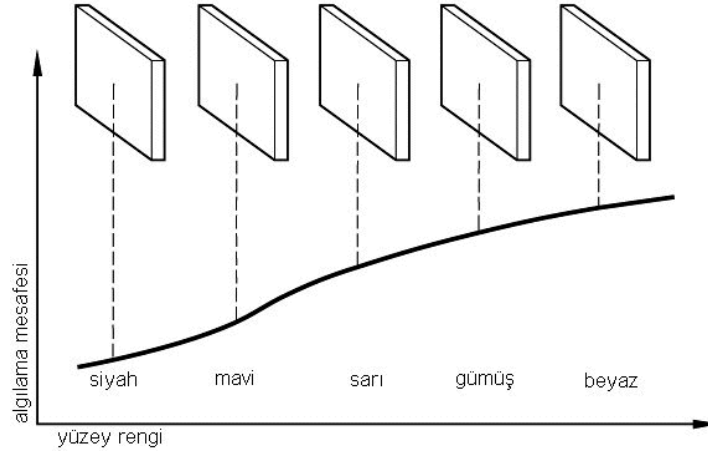
Şekil 2.10 Cisimden yansımali OU tip sensörün anahtarlama eğrisi (URL-1 2009).

Açma / kapama eğrisi de diğer yaklaşım anahtarlarının eğrisine benzer. Emniyetli anahtarlama sağlamak için herhangi bir boyuttaki cismin yandan ve önden yaklaşımında histerisis açıkça belirlenebilir.

Kolaylıkla anlaşılabilir olduğu gibi, daha büyük bir cisim küçük olandan ve yansıtıcı beyaz olan Cisim mat siyah olandan daha fazla ışık yansıtır. Dolayısıyla algılama mesafesi cismin boyutlarına ve rengine bağlıdır (Şekil 2.12.)



Şekil 2.11. Cisimden yansımali sensörün algılama çeşitleri (URL-1 2009).



Şekil 2.12 Cisimden yansımali sensörün algılama mesafesinin değişimi (URL-1 2009).

Özet olarak; cisimden yansımali sensörlerin önemli özellikleri şunlardır:

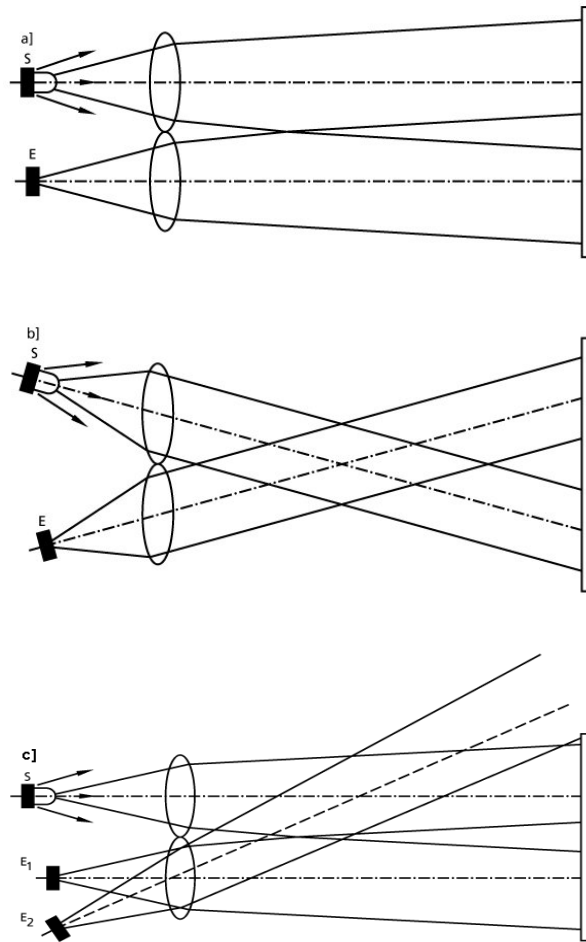
- a. Algılama mesafesi büyük oranda algılanacak cismin yüzey özelliklerine ve rengine bağlıdır. Burada algılanacak olunan cisim parmak olduğundan yüzey özellikleri ve renk bakımından oldukça elverişlidir.
- b. Cisimden yansımali sensörler öyle bir şekilde yerleştirilmelidir ki: arka taraf değil de sadece cisim kolaylıkla algılanmalıdır.

Bu aşağıdaki durumlarda gerçekleşir:

- 1-cisimi normal algılama mesafesi içine yerleştirin
- 2-duyarlılığı potansiyometre ile cisimi algılayıncaya kadar artırın ve bu noktayı unutmayın.
- 3-cisimi alın ve duyarlılığı arka tarafı algılayıncaya kadar artırın.
- 4-potansiyometreyi algılama gecene kadar geri çevirin
- 5-şimdi, olası ise, bu uygulama için optimum algılama mesafesi için potansiyometreyi 2 ve 4'deki konumlarının ortasına ayarlayın.

c. algılama mesafesi dışındaki yansıtıcı veya çok parlak cisimler kolaylıkla cisimden yansımali sensörün çalışmasını etkileyebilir.

Peki, bu tür sorunlar cisimden yansımali sensörlerde nasıl çözülebilir.



(a), geri bastırmalı (b) cisimden yansımali sensör (odaklanmış ışın ilkesi) ve geri palan bastırmalı (c).

Şekil 2.13 Cisimden yansımali sensor yayılım alanları (URL-1 2009).

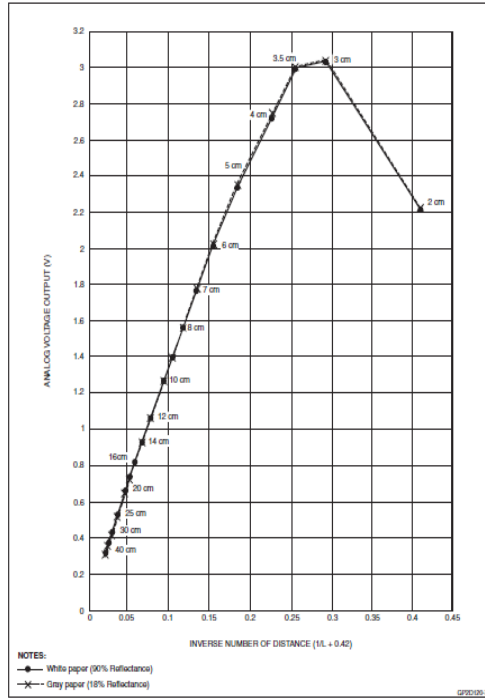
Şekil'in üst kısmı cisimden yansımali sensörün çalışma ilkesini gösterir. Alt taraftaki şekilde görüldüğü gibi, verici ve alıcının konumundaki deęişiklik bir çeşit arka tarafın etkisini bastırmaya neden olur. Arka tarafın nasıl olduğundan bağımsız olarak, verici ile alıcı kesişen alanının dışındaki cisim çok zor algılanacaktır. Bu yöntem tasarım biçiminden dolayı "odaklanmış ışın ilkesi "olarak adlandırılır. Bu. Çoğu kısa mesafeli cisimden yansımali sensörler için kullanılır.

(Not: alıcı duyarlılığı fabrikada özel algılama mesafesi değerine göre ayarlanmıştır ve deęiştirilmemelidir.) verici ile alıcı arasındaki ışının açısı özel lens yapım yöntemiyle ile sağlanır. Aynı tip normal cisimden yansımali sensörle karşılaştırıldığında, bu ilkeden dolayı bu tiplerde % 70 algılama mesafesi kaybı olur. Fakat algılama mesafesi neredeyse tümüyle cismin rengi ve yüzey özelliklerinden(beyaz kâğıt, kara keçe) bağımsızdır. Aynı zamanda, bu tip sensör yukarıda anlatılan arka tarafın etkisinden daha az etkilenir.

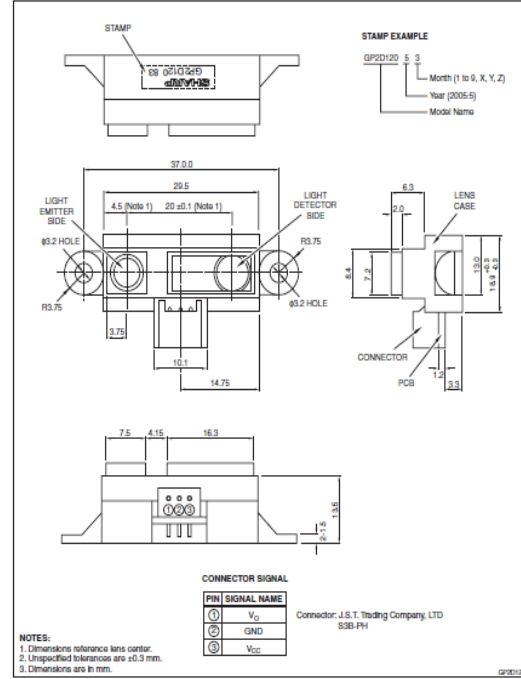
Cisimden yansımali sensörün kısaca tipik özellikleri:

- Cismin doğrudan algılanması, dolayısı ile yansıtıcı veya ikinci bir birime gerek yok
- Şeffaf cisimlerin iyi algılanır
- Karşılıklı / yansıtıcı sensörlerle karşılaştırıldığında kısa algılama mesafesi
- Algılama mesafesi algılanacak cismin yansıtıcılık kalitesine baęlı (rengi, yüzeyi) (daha da kısa algılama mesafeleri olan kısa mesafeli cisimden yansıtıcı sensörler hariç)
- Arka taraftan etkilenme olası(ayna, metal, beyaz) (kısa mesafeli cisimden yansımali sensörler yine hariç)

Opto–elektronik sensörler algılama mesafeleri ile karşılaştırıldıklarında çok küçük boyutlardadır. Dış etkilere karşı büyük oranda baęışıklıkları vardır. Toz, nem girişine (IP67) ve tümüyle reçine ile doldurulduklarından harici vibrasyona karşı da korunmuşlardır. Ancak, kolayca ulaşılamayan, çok küçük, çok sıcak veya çok sulu yerlerde kullanılabilirler. Bizim tercih ettiğimiz sharp gp2dl120 model kompact tasarıma sahip alıcı ve verici parça halinde tasarlanmış olan sensörün hem hassasiyet hem de çözünürlüğü ve tepki süresi oldukça iyidir. Aşağıda sensöre ait teknik bilgileri bulunmaktadır (URL-2 2009).



PACKAGE SPECIFICATIONS

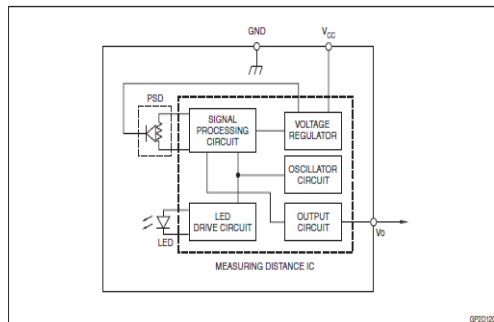
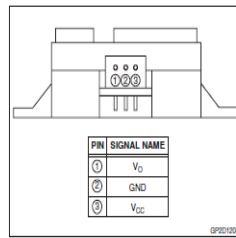
GP2D120
Optoelectronic Device

FEATURES

- Analog output
- Effective range: 4 to 30 cm
- Typical response time: 39 ms
- Typical start up delay: 44 ms
- Average Current Consumption: 33 mA

DESCRIPTION

The GP2D120 is a distance measuring sensor with integrated signal processing and analog voltage output.



ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings

T_a = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	RATING	UNIT
Supply Voltage	V _{CC}	-0.3 to +7	V
Output Terminal Voltage	V _O	-0.3 to (V _{CC} + 0.3)	V
Operating Temperature	T _{opr}	-10 to +60	°C
Storage Temperature	T _{stg}	-40 to +70	°C

Operating Supply Voltage

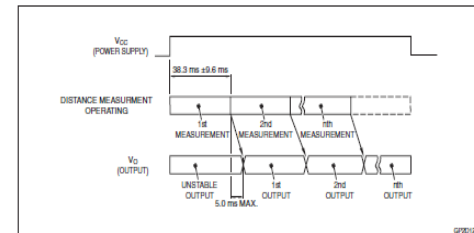
PARAMETER	SYMBOL	RATING	UNIT
Operating Supply Voltage	V _{CC}	4.5 to 5.5	V

Electro-optical Characteristics

T_a = 25°C, V_{CC} = 5 VDC

PARAMETER	SYMBOL	CONDITIONS	MIN.	TYP.	MAX.	UNIT	NOTES
Measuring Distance Range	ΔL		4	—	30	cm	1, 2
Output Terminal Voltage	V _O	L = 30 cm	0.25	0.4	0.55	V	1, 2
Output Voltage Difference	ΔV_O	Output change at ΔL (30 cm - 4 cm)	1.95	2.25	2.55	V	1, 2
Average Supply Current	I _{CC}	L = 30 cm	—	33	50	mA	1, 2

- NOTES:
- Measurements made with Kodak R-27 Gray Card, using the white side, (90% reflectivity).
 - L = Distance to reflective object.



Şekil 2.14 Sharp gp2d120 mesafe algılama sensörü (URL-3 2009).

SHARP

GP2D120

RELIABILITY

The reliability of requirements of this device are listed in Table 1.

Table 1. Reliability

TEST ITEMS	TEST CONDITIONS	FAILURE JUDGEMENT CRITERIA	SAMPLES (n), DEFECTIVE (C)	NOTES
Temperature Cycling	One cycle -40°C (30 min.) to +70°C in 30 minutes, repeated 55 times	Initial $\times 0.8 > V_D$ $V_D > \text{Initial} \times 1.2$	n = 11, C = 0	1
High Temperature and High Humidity Storage	+40°C, 90% RH, 500h		n = 11, C = 0	1
High Temperature Storage	+70°C, 500h		n = 11, C = 0	1
Low Temperature Storage	-40°C, 500h		n = 11, C = 0	1
Operational Life (High Temperature)	+80°C, $V_{CC} = 5V$, 500h		n = 11, C = 0	1
Mechanical Shock	100 m/s ² , 6.0 ms 3 times each +X, +Y, +Z direction		n = 6, C = 0	1
Variable Frequency Vibration	10-to-55-to-10 Hz in 1 minute Amplitude: 1.5 mm 2h in each X, Y, Z direction		n = 6, C = 0	1

NOTES:
1. Test conditions are according to Electro-optical Characteristics, shown on page 2.
2. At completion of the test, allow device to remain at normal room temperature and humidity (non-condensing) for two hours.
3. Confidence level: 90%, Lot Tolerance Percent Defect (LTPD): 30%/40%.

MANUFACTURER'S INSPECTION

Inspection Lot

Inspection shall be carried out per each delivery lot.

Inspection Method

A single sampling plan, normal inspection level II based on ISO 2859 shall be adopted.

Table 2. Quality Level

DEFECT	INSPECTION ITEM/TEST METHOD	AQL (%)
Major Defect	Electro-optical characteristics defect	0.4
Minor Defect	Defect on appearance and dimension (split, chip, scratch, stain)*	1.0

NOTE: *Any one of those that affects the Electro-optical Characteristics shall be considered a defect.

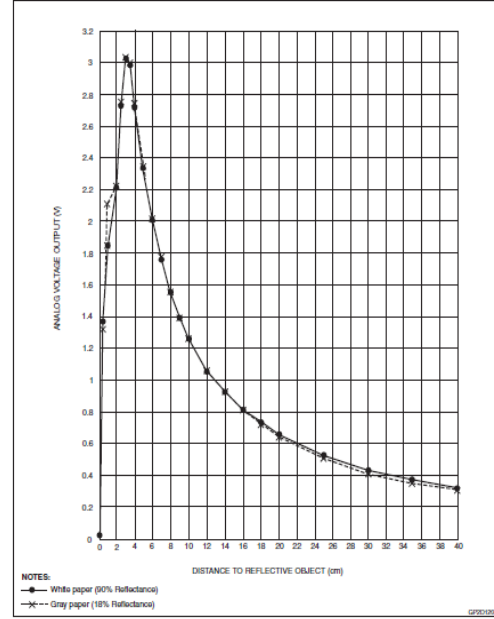


Figure 4. GP2D120 Example of Output Distance Characteristics

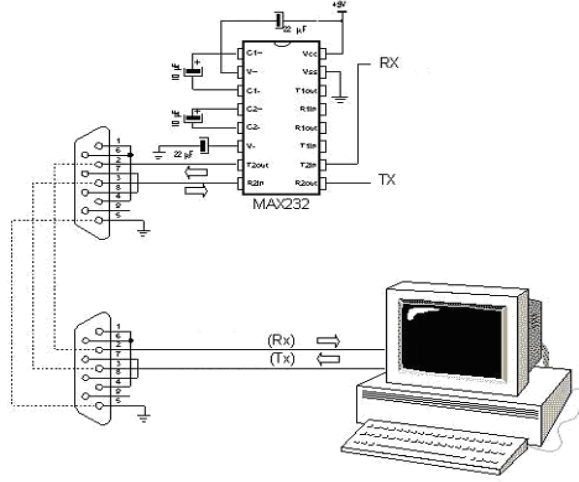
Şekil 2.14 (devam ediyor).

2.2.3 Entegreler

Devre düzeneğimizde kullandığımız entegreler şunlardır;

2.2.3.1 Max232 Entegresi

Max232 entegresi 5V ve 0V olarak bildiğimiz 1 ve 0 değerlerini seri veri daha uzağa gidebilsin diye 12V ve 0V düzeyine getirip aynı zamanda tersleyen bir entegredir. PIC mikro denetleyicinin bacaklarını seri portun Rx ve Tx girişlerine doğrudan girmek yerine arada Max232 entegresini kullanmak verilerin hatasız iletimi açısından gereklidir. Max232 entegresi -12, +12 V seviyesindeki seri port sinyallerini TTL (+5, 0 V) seviyesine veya TTL seviyesini -12 V, +12 V sinyallerine çevirir.

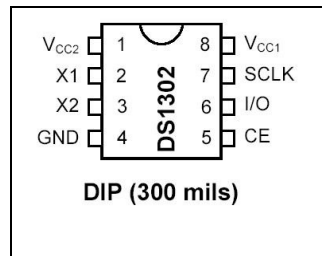


Şekil 2.15 PC seriportu ile PIC'in MAX232 ile birbirine bağlanması (Kaya 2007).

2.2.3.2 Gerçek zaman saati DS1302

DS1302; RTC, takvim ve 31 byte'lık RAM' a (pil korumalı) sahiptir. Mikroişlemci ile basit seri bir ara yüzle haberleşir. Gerçek zaman saati (RTC) saniyeyi, dakikayı, saati, günü, ayı, yılı, haftanın gününü sayar ve 2100 yılına kadar tarih bilgileri yüklüdür. 2.0 V ile 5.0 V arasında gerilimlerde çalışabilir. 2.0 V'ta 300 nA'den daha az akım tüketir. Saat ve RAM hafızasına yazma ve okuma için seri olarak okuma veya yazma işlemi yapabilme (burst) moduna sahiptir. TTL ile uyumludur ($V_{cc} = 5\text{ V}$). — 40 °C ile +85 °C arasında çalışabilir.

DS1302'ye bağlanan 32.768KHz'lik bir osilatör ve 5V DC voltaj ile teste hangi tarih ve saat de başladığımızı kaydedilmektedir. Basit bir seri ara yüze sahip olup kullanımı oldukça basittir.

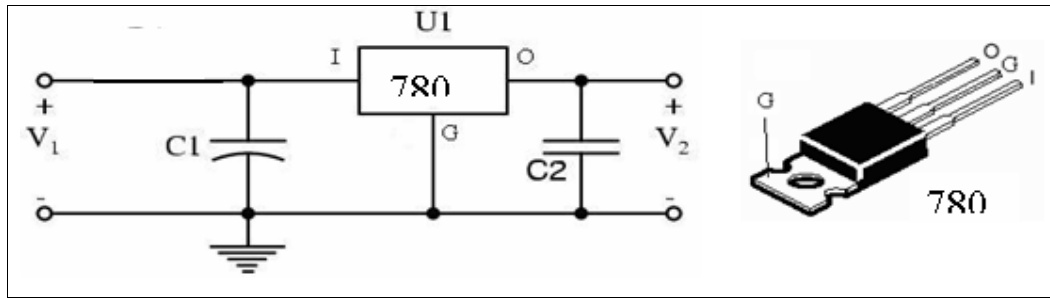


Şekil 2.16 DS1302 Entegresi.

DS1302 Saniye, Dakika, Saat, Haftanın Günü, Ayın günü, Ay ve Yıl bilgilerini saymakta ve saklamaktadır. Ay'ların kaç gün olduğu otomatik olarak ayarlanmıştır. Yani 29 veya 31 olan toplam ay uzunlukları otomatik olarak sayılmakta ve herhangi bir değişikliğe ihtiyaç duyulmamaktadır. Saat 24 veya 12 biçimlerinde çalıştırılabilir. Bu projede 24 biçiminde çalıştırılmıştır. 2-5.5V DC Voltaj aralığında çok düşük güç tüketimiyle rahatlıkla çalışmaktadır. 8 Pin'lik bu entegre 3 bacağından seri olarak haberleşmektedir. İki ayrı güç girişine sahip olan bu entegreye elektrik kesintilerinde oluşabilecek kayıpları önlemek için ayrıca yedek pil bağlantısı yapılabilmektedir. İçerisinde programlanabilen şarj devresi olduğu için şarj edilebilir ya da edilemez herhangi bir pil burada kullanılabilir (Kaya 2007).

2.2.3.3 7805 Entegresi

Devrede +5V DC gerilimi sağlamak amacıyla regüle devresi kullanılmıştır.



Şekil 2.17 7805 entegresi bağlantı şeması (URL-4 2009).

Şekil 2.17 'deki devre, bir kaç elemanla birlikte sabit bir gerilim regülatörü kullanarak istenilen doğru gerilimi elde etmek için kullanılır. İstenilen doğru gerilim değeri için 7805 entegresi kullanılmalıdır. C1 değeri 330 μ F 25V elektrolitik kapasitör, C2 değeri 0.1 μ F 25V seramik kapasitör olmalıdır.

7805 serisi regülatörlerin çalışması için giriş geriliminin çıkış gerilimi değerinden yaklaşık 3 V fazla olması gereklidir. Aksi takdirde çıkış gerilimi istenilen değerde olmayacaktır. Dolayısıyla 5 V çıkış için minimum 8 V DC gerilimin entegrenin girişine uygulanması gereklidir. Şekildeki devreden yaklaşık olarak 200 mA değerine kadar akım çekilebilir. Bu akım seviyesinde entegreye soğutucu bağlamanıza gerek olmayabilir. Daha fazla akım çekmek isteniyorsa girişe bağladığınız kaynağın güçlü olması gereklidir (URL-4 2009).

2.2.3.4 24CXX Seri Eeprom

Mikrokontrolör uygulamalarında dahili hafızaların yeterli olmadığı durumlarda harici seri hafızalara başvurulur. Amaç veri toplamak ve saklamak veya saklanmış verileri kullanmak olabilir. Veri erişim hızının yüksek olmadığı ve giriş/çıkış bacak sayısının önem kazandığı mikrokontrolör uygulamalarında çevre donanım elemanlarının (eeprom, rtc, sıcaklık ve nem sensörleri gibi) mikrokontrolöre bağlantı şekli önem kazanmış ve mümkün olan en az sayıda bağlantı olması arzulanmıştır. Bu şekilde data/clock (veri/saat pulsu) veya 1 wire (tek telli) gibi bağlantı standartları geliştirilmiştir. Tabiidir ki bu bağlantıları destekleyen protokoller de geliştirilmiştir.

Bu makalede 24CXX seri eeprom donanımını ve protokollerden patentli Philips firmasına ait olan i2c'nin seri eepromlarda kullanımını anlattıktan sonra bu devremizde kullanımını anlatacağız.

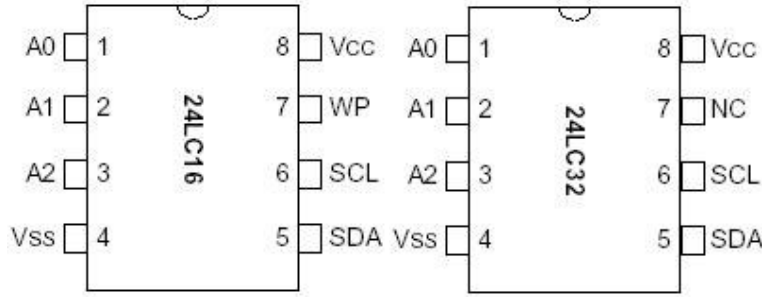
i2c protokolünü anlatmadan önce data/clock yapısının nasıl işlediğine bir gözatalım. Bilgi akışının mikrokontrolörden seri eeprom yönüne doğru olduğunu varsayalım. Seri eeproma gelen her bir saat sinyali, data hattında kendisi tarafından okunması gereken bir bilginin hazır olduğunu bildirir. Bu bilgi seri eeprom tarafından okunur. Mikrokontrolör tarafından üretilen clock (saat) sinyali seri eepromun adım adım kontrol edilmesini sağlar.

Data/clock seri haberleşmesinde clock (saat) sinyali aktiflenmeden önce seri hale dönüştürülmekte olan bilginin sıradaki biti veri hattında hazır tutulur. Aktiflenen saat sinyali ile bu veri biti seri eeproma iletilmiş olur. Bilgi akışı ters yönde ise her bir saat sinyali aktiflenmesinde seri eeprom data hattına bilginin sıradaki bitini koyar. Mikrokontrolörde bu bilginin okunup paralel hale getirilmesi sağlanır.

Seri eeproma bilginin gönderilmesi aşamasında bazı ilave bilgiler gönderilirken, bazı bilgiler de alınır. Bunlar, başla, okuma-yazma komutu, adres bilgisi, alındı bilgisi ve tamam-devam bilgisi, bitir komutu vs. dirler.

2.3 CXX SERİSİ SERİ EEPROMLAR

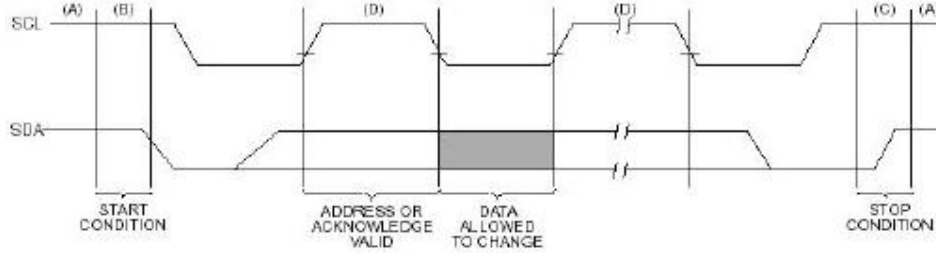
24 serisi eepromların bacak isimlendirmesi ve görevleri bakımından iki gruba ayrılmaktadır. Bazı üreticilerin ürünlerinde farklılık olmakla beraber genelde bu iki grup yapı standartlaşmıştır. 24c01 den 24c16 ya kadar olanlara grup1 dersek, 24c32 den 24c512 ye kadar olanlara grup2 olarak tarif edebiliriz.



Şekil 2.18 24 XX serisi eepromların pin şeması (URL-5 2009).

[Vss: toprak, Vcc: besleme, WP: yazma koruması, SCL: clock (saat), SDA: data (veri), NC: boş (no connection), AO, A1, grup2 seri eepromlarda çipi seçmek için kullanılır.]

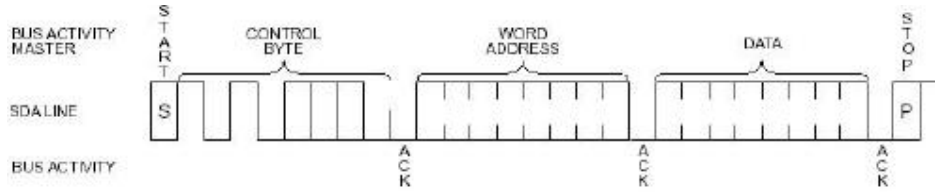
Haberleşmenin başlatılması ve bitirilmesi (START/STOP) için kullanılan komut her iki grup için de aynıdır.



Şekil 2.19 Data klok diyagramı (URL-5 2009).

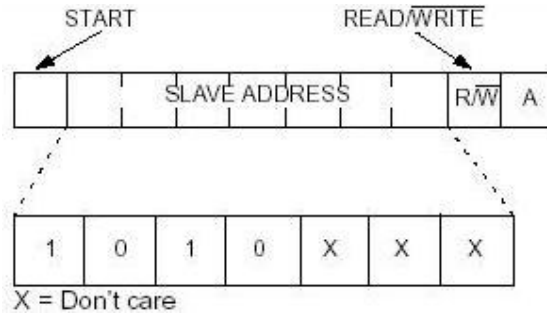
Başla komutu, saat sinyali Vcc de iken, veri hattının 0 Volta çekilmesi ile aktiflenir. Bitir komutu ise yine saat sinyali Vcc de iken, veri hattının 0 Volttan Vcc ye çekilmesi ile aktiflenir. Verilerin hat üzerinde değiştirilmesi saat sinyali 0 volt düzeyinde iken gerçekleştirilir.

Grup1 seri eepromlarda kontrol baytından sonra bir tane adres baytı gönderilir.



Şekil 2.20 Byte yerleşim diagramı (URL-5 2009).

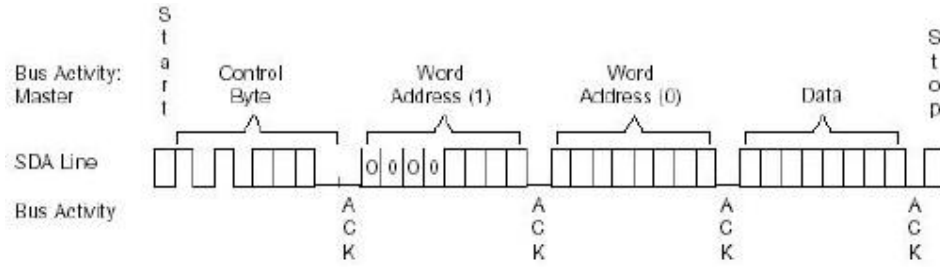
24c02 256 baytlık hafıza olduğu için bir baytlık adres bilgisi ile kolayca adreslenebilmektedir. 24c04-24c16 arası seri eepromlarda bir bayt çipin bütün adreslerine erişmek için yeterli olmadığından kontrol baytıdaki boşta kalan 3 bit bu çip içerisindeki alanları adreslemekte kullanılır (Aşağıdaki resimde XXX ile gösterilen alan).



Şekil 2.21 Carry byte ve adres bloğu (URL-5 2009).

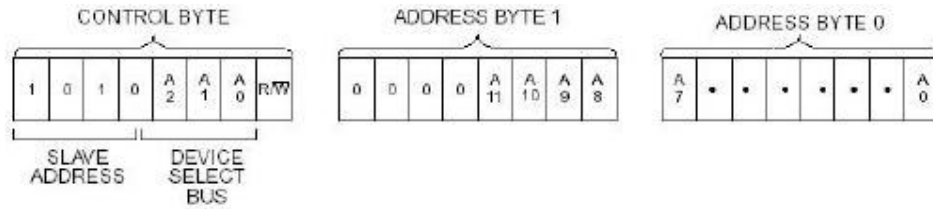
Grup 1 çiplerini yazmaya karşı korumak için bir bacak ayrılmıştır. Çip üzerinde A0,A1,A2 olarak gösterilen bacakların bir görevi yoktur. Boşta bırakılabilir, ya da toprak veya Vcc ye bağlanabilir.

Grup 2 seri eepromlarda kontrol baytıdan sonra iki tane adres baytı gönderilir. 2 bayt ile en fazla 64 Kbayt adreslenebildiği için 24c32 (4 Kbayt) den 24c512 (64Kbayt) ye kadar ki çipler bu iki adres baytı ile rahatlıkla adreslenebilmektedir.



Şekil 2.22 Grup2 eeprom byte ve adresleme bloğu (URL-5 2009).

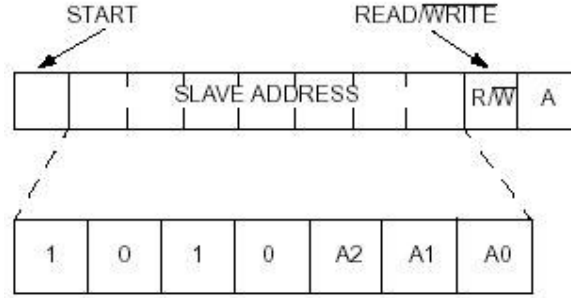
Kontrol baytı içerisinde kalan 3 bitlik boş alan ise çiplerin adreslenmesinde (chip select ya da device select)



Şekil 2.23 Grup2 eeprom byte adreslemesi (URL-5 2009).

Bu sayede aynı data/clock hattı üzerinde bu çiplerden en fazla 8 tanesi bağlanabilir hale getirilmiştir. Çip üzerinde A0,A1,A2 olarak gösterilen bacakların görevi çiplerin adreslenmesini sağlamaktır. Örnek olarak bir çipde bu bacaklar A0 toprağa, A1 Vcc ye, A2 toprağa bağlanmış ise, bu çipe bilgi yazmak ya da bilgi okumak için kontrol baytı A2 biti 0, A1 biti 1, A0 biti 0 olarak hazırlanmalıdır. Yazmaya karşı koruma grup2 çiplerde bulunmamaktadır.

Kontrol baytı hazırlanması da i2c protokolünde standartlaştırılmıştır. İlk 4 biti seri eepromlarda ve ramlerde hex (10) ya da 0xA ile başlar. I2C protokolü kullanan bazı rtc lerde (real time clock) de kontrol baytı 0xA ile başlamaktadır. Seri eeprom haberleşmesinde kontrol baytı 0xA ile başlamazsa, geri kalan bilgiler ne kadar da standarta uysa haberleşme gerçekleşmez. Bu şekilde aynı veri / saat yolu üzerinde farklı kontrol baytı ile başlayan çipler de bağlanabilmektedir (bilgi kaybının ve harici gürültülerin azaltılması için haberleşme hızı düşürülmeli ve hat mümkün olduğunca kısa tutulmalıdır).

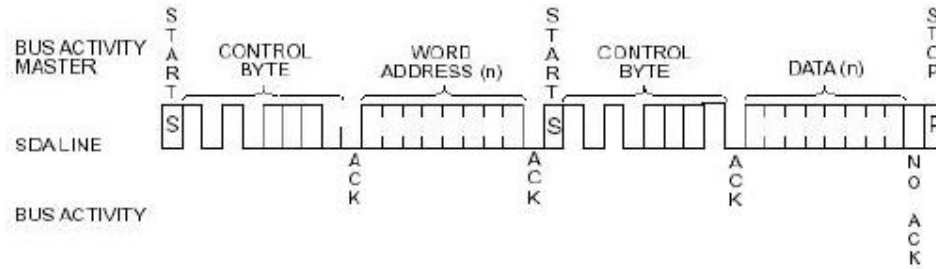


Şekil 2.24 Grup2 eeprom carry byte ve adres bloğu (URL-5 2009).

Kontrol baytında ikinci 4 bitlik alanda A2, A1, A0 ve okuma/yazma biti bulunmaktadır. 4. bit 0 (sıfır) ise yazma (mikrokontrolörden Out), 1 (bir) ise okuma (mikrokontrolöre Input).

Haberleşme sırasında mikrokontrolörden gönderilen her bir bayt bilgi arkasından seri eeprom alındı bilgisi olarak bir bit cevap verir (acknowledge, 0 volt). Bunun tersi durumda da yani seri eepromdan bilgi okunurken de mikrokontrolör, gelen her bir bayt bilgi arkasından bir bitlik alındı bilgisi göndermesi gerekir. Haberleşmenin durdurulması alındı bilgisinin tersi (nack, Vcc) ve ardından stop komutunun gönderilmesiyle sağlanır. Haberleşmenin herhangi bir zamanda kesilmesi için başlat veya bitir komutu göndermek yeterlidir.

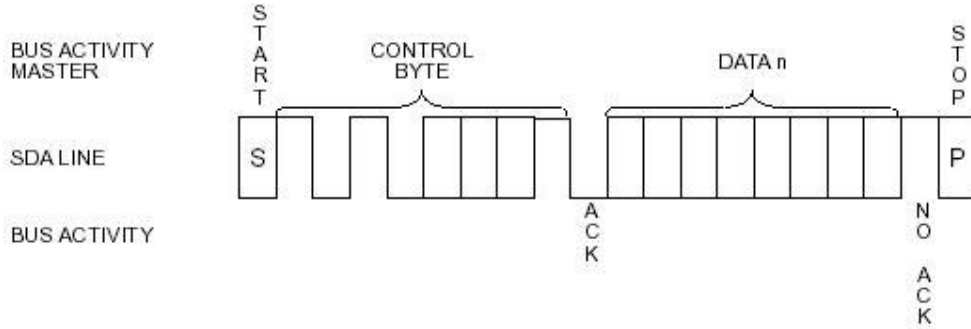
Herhangi bir adresten okuma işlemi şekilde görüldüğü gibidir.



Şekil 2.25 Seri eeprom okuma adreslemesi (URL-5 2009).

Seri eeprom okuma komutunda doğrudan okunacak adres bilgisi gönderilemediği için seri eeprom önce o adrese yazdırma işlemi yapılacağı gibi kandırılır fakat yazdırma işlemi yapılmadan tekrar start komutu gönderilir. Bu şekilde bilgi okunacak adresin seri eeprom içerisinde saklanması sağlanır. Daha sonra okuma işlemi seri eeprom içerisinde saklanan bu adresten yapılır. Seri eeprom kendisinden istenen bilgiyi seri olarak gönderdikten sonra en son okunan adres değerini bir arttırır ve mikrokontrolörden gelecek tamamı (nack) devamı (ack) bilgisini bekler. Okuma işlemi istenirse kesintisiz olarak bu şekilde devam ettirilebilir.

Yukarıda da belirttiğimiz gibi her okuma işlemi bittikten sonra seri eeprom içerisinde en son okunan adres değeri bir arttırılarak saklanır (stop komutu gönderilmiş olsa bile). Eğer en son okunan adresten sonraki adres okunmak istenirse bu adres tekrar gönderilmez ve Current Read adı verilen işlemle o adresteki bilgi okunmuş olur.

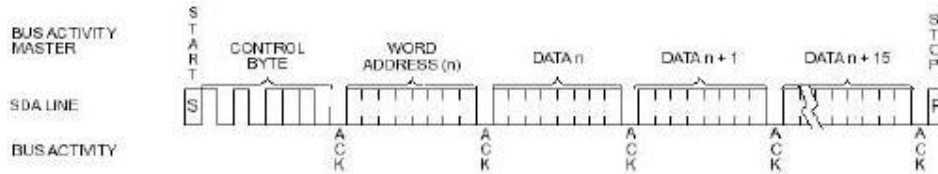


Şekil 2.26 Seri eepromda ikinci okuma adreslemesi (URL-5 2009).

Bu işlem birim zamandaki haberleşme sayısını arttırmakta kulllanılır.

Bu okuma işleminden sonra da seri eeprom içerisindeki en son okunan adres bilgisi tekrar bir arttırılarak saklanır.

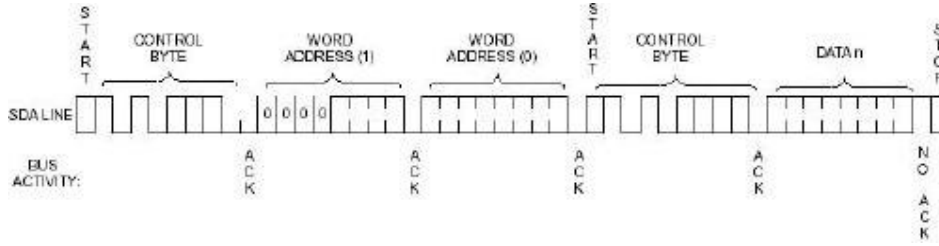
Herhangi bir adrese yazdırma işlemi şekilde görüldüğü gibidir.



Şekil 2.27 Grup1 eeprom bilgi yazdırma adres bloğu (URL-5 2009).

Bu şekilde görünen işlem page write işlemidir ve isteğe bağlı sayıda (Grup 1 için en fazla 16 bayt, grup 2 için en fazla $8 \times 8 = 64$ bayt) bilgi aynı anda yazdırılabilir. Page write işlemi dikkatli yapılması gereken bir işlemdir ve mutlaka her bir tipin datasheetinden dikkatlice okunup uygulanması gerekir. Page sınırlarından sonra devam eden okuma ya da yazma işlemleri tampon belleğin başına dönülüp, bilgilerin kaybolmasına neden olabilir!

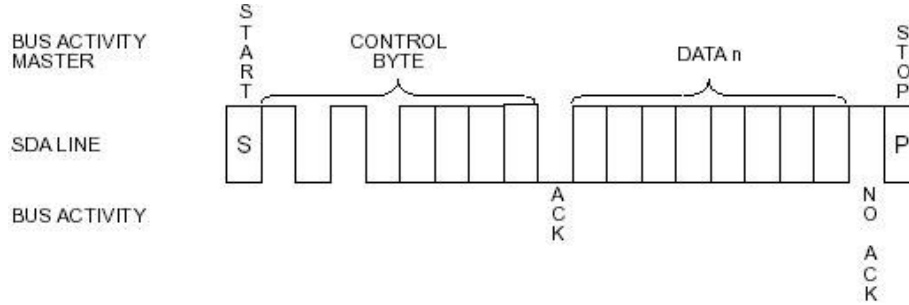
Herhangi bir adresten okuma işlemi şekilde görüldüğü gibidir.



Şekil 2.28 Grup2 eepromda okuma adres bloğu (URL-5 2009).

Okuma işlemi için adres girmek Grup1 de anlatıldığı gibidir. NO ACK komutu yerine ACK komutu gönderilerek sonraki 1 bayt bilgide okunabilirdi. Bu şekilde tekrar tekrar start-kontrol-adres1-adres0-start-kontrol baytları gönderilmeden hafızanın tamamı okunabilir.

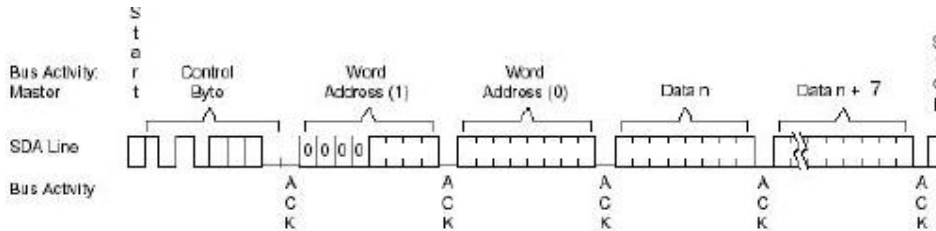
Current Read işlemi



Şekil 2.29 Grup2 eepromda curnet read (URL-5 2009).

Herhangi bir adrese yazdırma işlemi aşağıdaki şekilde görüldüğü gibidir.

Bu işlem page write işlemidir. Data n ile gösterilen bilgi seri eeproma gönderildikten sonra STOP komutu gönderilirse bir baytlık bilgi seri eeproma yazdırılmış olur (URL-5 2009).



Şekil 2.30 Grup2 seri eepromda yazma adresleme (URL-5 2009).

2.4 MİKRO DENETLEYİCİLER

2.4.1 Mikro Denetleyici nedir?

Mikro denetleyici terimini açıklamak için öncelikle mikroişlemci ve mikrobilgisayar konularını bilmemiz gerekmektedir. Mikroişlemci tüm bilgisayarlarda ve bilgisayar donanımına sahip elektronik cihazlarda bulunan, her türlü aritmetik ve mantıksal işlemin yapıldığı merkezi bir işlem ünitesidir. Bir mikroişlemci genelde şu kısımlardan oluşur:

1. Aritmetik-Mantık Ünitesi (ALU)
2. Komut Çözücü (Instruction Decoder)
3. Yazmaçlar (Registers)
4. Veri Yolu Kontrol Devresi (Data Bus Controller Circuitry)

Mikroişlemcinin işlevini yerine getirebilmesi için bazı yardımcı elemanlara ihtiyacı vardır. Bunlar:

- a) Giriş Üniteleri
- b) Çıkış Üniteleri
- c) Bellek Üniteleri şeklinde sıralanabilir.

Bir mikrobilgisayarı oluşturan temel bileşenlerin tek bir yonga içerisinde üretilmiş haline mikrodenetleyici denir. Bilgisayar, mikrobilgisayar ve mikrodenetleyicilerin sahip olduğu kabiliyetler, sağladığı imkânlar, fiyatları, donanım özellikleri ve kullanım alanları farklıdır. Bunlardan en ucuz, kullanımı kolay ve basit olanı mikrodenetleyicilerdir.

2.4.2 Mikrodenetleyicilerin Kullanım Sebebi

Elektronik sistemlerde sayısal devrelere sıklıkla rastlanır. Pek çok aritmetik ve mantıksal işlemi sayısal devreler sayesinde yapmak mümkündür. Sayısal bir sistemde gereken fonksiyonları yerine getiren bir devre kurmak veya tasarlamak için birkaç seçenek vardır:

İstenilen fonksiyonu yapmak üzere üretilmiş basit bir entegre devre piyasada mevcut olabilir. Sayısal elektrik sinyalleri ile ifade edilen sayıları karşılaştırıp ‘büyük’, ‘küçük’ veya ‘eşit’

gibi sonuçlar veren entegre devreler piyasada mevcuttur ve oldukça ucuzdur. Bu durumda doğru olan bu devreleri kullanmaktır.

PLD olarak adlandırılan programlanabilir yongalar her türlü mantıksal işlemi yapma kabiliyetine sahiptir. İstenilen fonksiyona göre tekrar programlanabilir olması en önemli özelliğidir. Kullanımı yukarıda bahsedilen entegre devreler kadar kolay değildir. Ayrıca maliyeti de daha yüksektir.

Bahsedeceğimiz son seçenek amacımıza uygun bir mikrodenetleyici kullanmaktır. Basit entegre devrelerin ve PLD'lerin yetersiz kaldığı durumlarda mikrodenetleyici kullanmak özellikle karmaşık sayısal sistemler için avantajlıdır. Günümüzde mikrodenetleyicileri programlamak PLD'leri programlamaktan zor olmadığı gibi mikrodenetleyiciler PLD'lerin sağlayamadığı pek çok kabiliyete de sahiptir. Bu üstün özellikleri sebebiyle pek çok elektronik kontrol sisteminde mikrodenetleyiciler tercih edilir.

2.4.3 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. PIC işlemcileri RISC benzeri işlemciler olarak anılır. PIC serisi tüm işlemciler herhangi bir ek bellek veya giriş / çıkış elemanı gerektirmeden sadece 2 adet kondansatör, 1 adet direnç ve bir kristal ile çalıştırılabilmektedir. Tek bacadan 40mA 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 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'ün bu kadar popüler olması onun çok iyi bir işlemci olmasından ziyade program belleğinin EEPROM olmasından kaynaklanmaktadır.

2.4.3.1 Neden PIC?

Bilgisayar denetimi gerektiren bir uygulamayı geliştirirken seçilecek mikrodnetleyicinin ilk olarak tüm isteklerinizi yerine getirip getirmeyeceğine, daha sonra da maliyetinin düşüklüğüne bakmalıyız. Ayrıca, yapacağımız uygulamanın devresini kurmadan önce seçtiğimiz mikrodnetleyicinin desteklediği bir yazılım üzerinde benzetim yapıp yapamayacağınızı da dikkate almalıyız.

PIC'leri ele alınmamızın nedenlerini kısaca şöyle sıralayabiliriz:

- Yazılımın Microchip'ten veya internetten parasız olarak elde edilebilmesi.
- Çok geniş bir kullanıcı kitlesinin bulunması.
- PIC'lerin çok kolaylıkla ve ucuz olarak elde edilebilmesi.
- Elektronik hobi olarak uğraşanların bile kullanabildikleri basit elemanları kullanarak yapılan donanımla programlanabilmesi.
- Çok basit reset, clock sinyali ve güç devreleri gerektirmeleri.

PIC çevresel üniteler adı verilen lamba, motor, role, ısı ve ışık sensörü gibi 1/0 elemanların denetimini çok hızlı olarak yapabilecek şekilde tasarlanmış bir chip'tir. PIC mikrodnetleyicileri günümüzde birçok uygulamada kullanılmaktadırlar. PIC'in tasarımcılar tarafından tercih edilmesinin sebepleri:

- Kod Verimliliği

PIC, Harvard mimarisi temelli 8 bit'lik bir mikrodnetleyicidir. Bu, bellek ve veri için ayrı yerleşik bus'ların bulunduğu anlamına gelir. Böylelikle akış miktarı veriye ve program belleğine anında erişim sayesinde arttırılmış olur. Geleneksel mikrodnetleyicilerde 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 mikrodnetleyicilerde 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, INCFSZ gibi komutlar dışında, her bir komut, tek bir çevrimde çalışır. Mikrodenetleyicinin 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.

- 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.

2.4.3.2 PIC Çeşitleri

Microchip ürettiği mikrodenetleyicileri 4 farklı gruba (Genellikle aile diye adlandırılır.) ayırarak isimlendirmiştir. PIC ailelerine isim verilirken kelime boyu (VWord lenght) göz önüne alınmıştır. Bir CPU veya MCU'nun dâhili veri yolu uzunluğuna kelime boyu denir. Microchip PIC'leri 12.14.16 bit'lik kelime boylarında üretmektedir ve buna göre sıradaki aile isimlerini vermektedir. PIC16C5XXailesi 12-bit kelime boyu, PIC16CXXX ailesi 14-bit kelime boyu, PIC17CXXX ailesi 16-bit kelime boyu, PIC12CXXX ailesi 12-bit/14-bit kelime boyuna sahiptir. Bir CPU veya MCU'nun chip dışındaki harici ünitelerle veri alışverişini kaç bit ile yapıyorsa buna veri yolu bit sayısı denir. PIC'ler farklı kelime boylarında üretilmelerine rağmen harici veri yolu tüm PIC ailesinde 8-bit'tir.

2.4.3.3 PIC Bellek Çeşitleri

Farklı özellikte program belleği bulunan PIC'ler microchip firması tarafından piyasaya sürülmektedir. Bunlar:

- Silinebilir ve programlanabilir bellek (Erasable Programmable Memory-Eprom).
- Elektriksel olarak silinebilir ve programlanabilir bellek Flash bellek olarak da adlandırılır.
- Sadece okunabilir bellek, Rom.

Her bir bellek tipinin kullanılacağı uygulamaya göre avantajları ve dezavantajları vardır. Bu avantajlar; fiyat, hız, defalarca kullanmaya yatkınlık gibi faktörlerdir. Eprom bellek hücrelerine elektrik sinyali uygulayarak kayıt yapılır. Eprom üzerindeki enerji kesilse bile bu program bellekte kalır. Ancak silip yeniden başka bir program yazmak için ultra-viole ışını altında belirli bir süre tutmak gerekir. Bu işlemler Eprom silici denilen özel aygıtlarla yapılır. Eprom bellekli PIC'ler iki farklı ambalajlı olarak bulunmaktadır:

- Seramik ambalajlı ve cam pencereci olan tip, silinebilir olan tiptir.
- Plastik ambalajlı ve penceresiz olan tipler ise silinemez (OTP) tiptir.

Seramik ambalajlı ve pencereci olan bellek içerisindeki programın silinmemesi için pencere üzerine ışık geçirmeyen bir bant yapıştırılır. Ultra-viole ışığı ile silinmesi istendiğinde bu pencere açılır ve silici aygıt içerisinde belirli bir süre bekletilir. Plastik ambalajlı Eprom'lar ise programlandıktan sonra silinmesi mümkün değildir ve fiyatı silinebilen tipe göre oldukça ucuzdur. Silinemeyen tipe OTP (Bir defa programlanabilir) olarak adlandırılır. Eprom belleği bulunan bir PIC içerisinde program yazmak için PIC programlayıcı vasıtasıyla elektriksel sinyal gönderilir. Eprom üzerindeki enerji kesilse bile bu program bellekte kalır. Programı silmek veya farklı yeni bir program yazmak istendiğinde PIC programlayıcıdan elektriksel sinyal gönderilir. Bu tip belleğe sahip olan PIC'ler genellikle uygulama geliştirme amacıyla kullanılırlar. Microchip bu tip belleğe çoğu zaman FLASH bellek olarak da adlandırmaktadır. Fiyatları silinemeyen tiplere göre biraz pahalıdır. Bellek erişim hızları ise Eprom ve ROM'lara göre daha yavaştır. ROM program belleğine sahip PIC'lerin programları fabrikasyon olarak yazılırlar. Eprom ve Eprom eşdeğerlerine nazaran fiyatları oldukça düşüktür. Ancak fiyatının düşüklüğünden dolayı gelen avantaj bazen çok pahalıya da mal olabilir. ROM bellekli PIC programlarının fabrikasyon olarak yazılması nedeniyle PIC'in elde

edilme süresi uzundur. Programda oluşabilecek bir hatanın PIC'e program yazıldıktan sonra tespit edilmesi, eldeki tüm PIC'lerin atılmasına da neden olabilir. Bu tip PIC'ler çok miktarda üretilecek bir ürünün maliyetini düşürmek amacıyla seçilir. Program hataları giderilemediği için uygulama geliştirmek için uygun değildir. Microchip, ROM program bellekli PIC'lere parça numarası verirken "CR" (PIC16CR62, PIC16CR84 gibi) harfleri kullanılır.

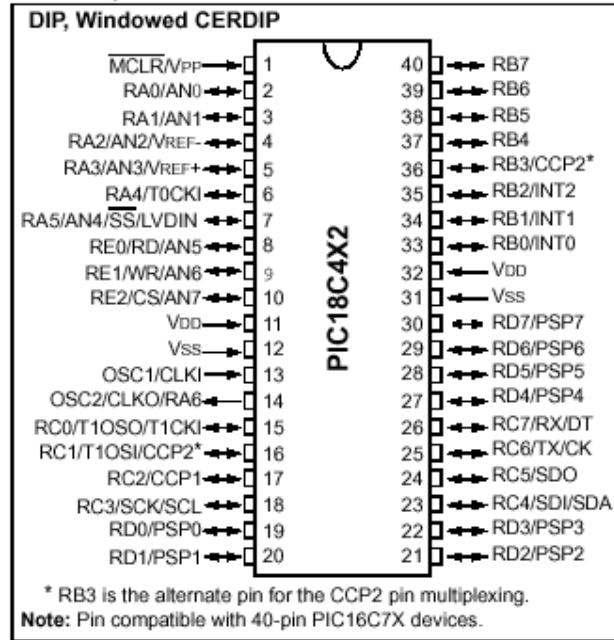
2.4.3.4 PIC Devresi

PIC' i çalıştırmak için gerekli olan donanım bir osilatördür. Osilatör, bir direnç ve kapasitör veya bir kristal/seramik rezonatör ve iki kapasitör formunda olabilir. Reset hattı kaynak gerilimine bağlanabilir. Dâhili devreler, normal şartlarda PIC' in yeniden başlatılmasını sağlayacaklardır. Bir sonraki aşama harici reset bileşenlerinin ve gerekiyorsa gerilim düşüklüğünden kaynaklanabilecek reset korumasının sağlanmasıdır. 100 nF'lik dekuplaj kapasitörü PIC' e olabildiğince yakın olarak güç kaynağına bağlanmalıdır.

2.4.3.5 PIC18F452 Özellikleri ve Yapısı

PIC18F452, belki en popüler PIC işlemcisi olan PIC16F84'ten sonra kullanıcılarına yeni ve gelişmiş olanaklar sunmasıyla hemen göze çarpmaktadır. Program belleği FLASH ROM olan PIC18F452'de, yüklenen program PIC16F84'te olduğu gibi elektriksel olarak silinip yeniden yüklenebilmektedir. Özellikle PIC16C6X ve PIC16C7X ailesinin tüm özelliklerini barındırması, PIC18F452'yi kod geliştirmede de ideal bir çözüm olarak gündeme getirmektedir. Konfigürasyon bitlerine dikkat etmek şartıyla C6X veya C7X ailesinden herhangi bir işlemci için geliştirilen kod hemen hiçbir değişikliğe tabi tutmadan F452'e yüklenebilir ve çalışmalarda denenebilir. Şekil 2.31'de PIC 18F452'nin bacakları ve Çizelge 2.3'de bacaklarının fonksiyonları gösterilmiştir.

Pin Diagrams



Şekil 2.31 PIC 18F452'nin Bacak Tanımlamaları (URL-6 2009).

Çizelge 2.3 PIC 18F8452'nin Bacaklarının Fonksiyonları (URL-6 2009).

Pin Adı	Pin No	I/O	Buffer Tipi	Açıklamalar
OSC1/CLKIN	13	I	ST/CMOS (3)	Osilatör clock girişi (kristal veya harici kaynak)
OSC2/CLKOUT	14	O	0	Osilatör kristal çıkış ucu
MCLR/Vpp	1	I/P	ST	Resetleme girişi / Programlama anında programlama gerilimi girişi
Port A iki yönlü giriş/çıkış portudur.				
RA0/AN0	2	I/O	TTL	Analog olarak kullanılabilir.
RA1/AN1	3	I/O	TTL	Analog olarak kullanılabilir.
RA2/AN2/VREF	4	I/O	TTL	Analog olarak kullanılabilir.
RA3	5	I/O	TTL	Analog olarak kullanılabilir.
RA5/SS/AN4	6	I/O	ST	Bu pin TMR0 için bir clock girişi olabilir.
RA5/SS/AN4	7	I/O	TTL	SSP Slave seçme pini veya analog giriş/çıkış olabilir.
PORTB iki yönlü giriş/çıkış portudur. Giriş konumunda iken dâhili pullup devresi aktifleşebilir. Dış kesme Girişi olarak seçilebilir.				

Çizelge 2.3 (devam ediyor)

RB0/INT	33	I/O	TTL/ST(1)	
RB1	34	I/O	TTL	
RB2	35	I/O	TTL	
RB3/PGM	36	I/O	TTL	Düşük akımla programlamada da kullanılabilir.
RB4	37	I/O	TTL	Kesme girişi olarak seçilebilir.
RB5	38	I/O	TTL	Kesme girişi olarak seçilebilir.
RB6/PGC	39	I/O	TTL/ST(2)	Kesme girişi olarak seçilir. Seri rogramlamada clock pinidir.
RB7/PGD	40	I/O	TTL/ST(2)	Kesme girişi olarak seçilir. Seri programlamada data (veri) pinidir.
Port C iki yönlü giriş/çıkış portudur.				
RC0T1OS0/T1CK1	15	I/O	ST	Timer1 osc. Çıkışı veya saat girişi olarak kullanılabilir.
RC1/T1OS1/CCP2	16	I/O	ST	Timer1 osc giriş veya Capture2 giriş/Compare2 Çıkışı/PWM2 çıkışı olarak kullanılabilir.
RC2/CCP1	17	I/O	ST	Timer1 osc giriş veya Capture1 giriş/Compare1 Çıkışı/PWM1 çıkışı olarak kullanılabilir.
RC3/SCK/SCL	18	I/O	ST	SPI ve I 2 modunda, seri saat giriş/çıkışında kullanılır.
RC4/SD1/SDA	23	I/O	ST	SPA modda SPI giriş verisi veya I 2 C modda I/O için seçilebilir.
RC5/SDO	24	I/O	ST	SPA modda SPI çıkış verisi için seçilebilir.
RC6/TX/CK	25	I/O	ST	USART asenkron gönderme ya da senkron saat için seçilebilir.
RC7/RX/DT	26	I/O	ST	USART asenkron alma ya da senkron veri için seçilebilir.
Port D iki yönlü giriş/çıkış portudur.				

Çizelge 2.3 (devam ediyor)

RD0/PSP0	19	I/O	TTL/ST(1)	PSP 0. biti olarak kullanılabilir.
RD1/PSP1	20	I/O	TTL/ST(1)	PSP 1. biti olarak kullanılabilir.
RD2/PSP2	21	I/O	TTL/ST(1)	PSP 2. biti olarak kullanılabilir.
RD3/PSP3	22	I/O	TTL/ST(1)	PSP 3. biti olarak kullanılabilir.
RD4/PSP4	27	I/O	TTL/ST(1)	PSP 4. biti olarak kullanılabilir.
RD5/PSP5	28	I/O	TTL/ST(1)	PSP 5. biti olarak kullanılabilir.
RD6/PSP6	29	I/O	TTL/ST(1)	PSP 6. biti olarak kullanılabilir.
RD7/PSP7	30	I/O	TTL/ST(1)	PSP 7. biti olarak kullanılabilir.
Port E iki yönlü giriş/çıkış portudur.				
RE0/RD/AN5	8	I/O	ST/TTL(3)	Analog olarak ya da PSP okuma kontrollü kullanılabilir.
RE1/WR/AN6	9	I/O	ST/TTL(3)	Analog olarak ya da PSP okuma kontrollü kullanılabilir.
RE2/CS/AN7	10	I/O	ST/TTL(3)	Analog olarak ya da PSP okuma kontrollü kullanılabilir.
NC	-		-	Bu pinler içerde kontrol edilmiyor, bağlı değil.
VSS	8	P	-	Ground (toprak) uç
VDD	19	P	-	Pozitif kaynak ucu

2.4.3.6 PIC 18F452 Portlarının Fonksiyonları

Port A

Her bir biti bağımsız olarak giriş veya çıkış olarak tanımlanabilmektedir. 6 bit genişliğindedir. RA0, RA1, RA2, RA3, RA4 ve RA5 bitleri analog / sayısal çevirici olarak konfigüre edilebilmektedir. Buna ek olarak RA2 ve RA3 gerilim referansı olarak da konfigüre edilebilmektedir.

Port B

Her bir biti bağımsız olarak sayısal giriş veya çıkış olarak tanımlanabilmektedir. 8 bit genişliğindedir. B portunun her bacağı dahili bir dirençle Vdd'ye bağlıdır. RB4 - RB7 bacakları aynı zamanda bacakların sayısal durumlarında bir değişiklik olduğunda INTCON

yazmacının 0. biti olan RBIF bayrağını 1 yaparak kesme oluşturmaktadır. Bütün bunların yanı sıra RB6 ve RB7 yüksek gerilim programlama, RB3 ise düşük gerilim programlama modlarında da kullanılmaktadır.

Port C

Her bir biti bağımsız olarak sayısal giriş veya çıkış olarak tanımlanabilmektedir. 8 bit genişliğindedir. Tüm port bacakları Schmitt Trigger girişlidir. TRISE yazmacının 4. biti olan PSPMODE bitini 1 yaparak “parallel slave mode” da kullanılabilir. Bu fonksiyon aracılığıyla 8 bit genişliğindeki her hangi bir mikroişlemci bus’ına bağlanabilir.

Port D

Her bir biti bağımsız olarak sayısal giriş veya çıkış olarak tanımlanabilmektedir. 8 bit genişliğindedir. Tüm port bacakları Schmitt Trigger girişlidir. TRISE yazmacının 4. biti olan PSPMODE bitini 1 yaparak paralel slave modda kullanılabilir. Bu fonksiyon aracılığıyla 8 bit genişliğindeki herhangi bir mikroişlemci bus’ına bağlanabilir.

Port E

Her bir biti bağımsız olarak giriş veya çıkış olarak tanımlanabilmektedir. 3 bit genişliğindedir. RE0, RE1 ve RE2 bacaklarında Schmitt Trigger giriş tamponları vardır. Her bir bacak analog / sayısal çevirici olarak konfigüre edilebilmektedir. RE portlarının A / D veya sayısal giriş / çıkış olarak seçiminde kullanılmaktadır. İşlemciye ilk defa gerilim uygulandığında üç port e biti de A / D çeviricidir. Eğer RE portunun bazı bitlerini sayısal giriş / çıkış olarak kullanmak istenirse ADCON1 yazmacında değişiklik yapılması gerekecektir.

2.4.3.7 PIC 18F452’nin Özel Fonksiyonları

- Paralel Slave Port

TRISE yazmacının PSPMODE biti 1 yapıldığında PORTD 8 bit genişliğinde mikroişlemci portu olarak kullanabilir. Bu arada RE0, RE1 ve RE2’yi, TRISE ve ADCON1 yazmaçlarında ilgili ayarları yaparak sayısal giriş olarak da tanımlamak gerekmektedir. Böylece harici bir mikroişlemci, RE0, RE1 ve RE2’yi kontrol olarak kullanarak 8 bitlik veri

bus'ına bağı PIC16F452'nin port d kısmına hem veri yazabilmekte, hem de okuyabilmektedir.

- USART

USART, yani senkron / asenkron alıcı verici PICF877'deki iki seri giriş / çıkış biriminden biridir. Seri iletişim ara yüzü (SCI) olarak da bilinen USART, monitör veya PC gibi aygıtlara tam çift yönlü asenkron bağlantıda kullanılmak üzere konfigüre edilmiştir. A / D veya D / A ara yüzlerine, seri kullanılmak üzere konfigüre edilebilmektedir. USART aşağıdaki gibi konfigüre edilebilmektedir.

Asenkron: Tam çiftyönlü

Senkron: Master, yarım çift yönlü

Senkron: Slave, yarım çift yönlü RC6 verici, RC7 ise alıcı port olarak kullanılmaktadır. RCSTA (0x18) ve TXSTA (0x98) yazmaçları da konfigürasyonda kullanılmaktadır.

- Master Snchronous Serial Port (MSSP)

MSSP birimi, diğer çevre birimleri veya mikroişlemcilerle seri iletişimde kullanılmaktadır. Bu çevre birimleri seri EEPROM, kaydırmalı yazmaçlar (shift register), gösterge sürücüler, A / D çeviriciler vb. Olabilir. MSSP birimi aynı anda aşağıdaki iki moddan birine konfigüre edilebilir. RC5: Seri veri çıkışı (SDO: Serial data out) RC4: Seri veri girişi (SDI: Serial data in) RC3: Seri saat (SCK: Serial clock)Bu modlardan birine göre konfigüre etmek içinse SSPSTAT (senkron seri port durum yazmacı, 0x94), SSPCON (senkron seri port kontrol yazmacı, 0x14) ve SSPCON2 (senkron seri port kontrol yazmacı 2,0x91) yazmaçları ayarlanmalıdır.

- Analog / Sayısal Çevirici Birimi

A / D çevirici birimi 16C7X ailesinden farklı olarak 10 bittir. Toplam 8 A / S kanal bulunmaktadır. F877'nin güzel bir özelliği de işlemci SLEEP modunda iken bile A / D çeviricinin geri planda çalışmasıdır. A / D kanalları için RA4 hariç diğer RA portları ve RE portları kullanılabilir.

Aşağıda ilgili yazmaçlar ve adresleri gösterilmiştir.

ADRESH 0x1E; A / D sonuç yazmacı (high register)
ADRESL 0x9E; A / D sonuç yazmacı (low register)
ADCON 0 0x1F; A / D kontrol yazmacı 0
ADON1 0x9F; A / D kontrol yazmacı 1

- Yakalama / Karşılaştırma ve PWM Birimi

Her capture/compare ve pwm birimi 16 bitlik yakalama yazmacı, 16 bitlik karşılaştırma yazmacı veya 16 bitlik PWM yazmacı olarak kullanılmaktadır. Yakalama modunda, TMR1 yazmacının değeri, RC2/CCP1 bacağının durumunda bir gelişme olduğunda CCPR1H: CCPR1L yazmaçlarına yazılmakta ve PIR1 yazmacının 2. biti olan CCP1IF kesme bayrağı 1 olmaktadır. RC2 bacağının durumu, her düşen kenarda, her yükselen kenarda, her yükselen 4. veya 16. kenarda kontrol edilecek şekilde CCP1CON yazmacı aracılığıyla ayarlanarak konfigüre edilebilir.

Karşılaştırma modu ise CCPR1 yazmacındaki 16 bitlik değer düzenli olarak TMR1 yazmaç değeriyle karşılaştır ve bir eşitlik olduğunda RC2/CCP1 bacağı CCP1CON yazmacında yaptığımız ayara göre 1, 0 olur veya durumunu korur. PWM modunda ise RC2/CCP1 bacağı 10 bit çözünürlükte darbe genişlik modülasyonlu bir işaret üretecek şekilde konfigüre edilebilir. PR2 yazmacı darbe genişlik periyodunun tayininde kullanılmaktadır.

2.4.3.8 PIC 18F452'nin RAM Belleği

PIC18F452'nin 0x00~7Fh adres aralığına ayrılmış olan RAM belleği vardır. Bu bellek içerisindeki dosya yazmaçları içerisine yerleştirilen veriler PIC işlemcisinin çalışmasını kontrol etmektedir. File register adı verilen özel veri alanlarının dışında kalan diğer bellek alanları, normal RAM bellek olarak kullanılmaktadırlar. Çizelge 2.4'de PIC18F452'nin kullanıcı RAM bellek haritası görülmektedir.

Çizelge 2.4 PIC16F452 Ram Bellek Haritası (URL-6 2009).

File Address		File Address		File Address		File Address	
Indirect addr. ⁽¹⁾	00h	Indirect addr. ⁽¹⁾	80h	Indirect addr. ⁽¹⁾	100h	Indirect addr. ⁽¹⁾	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADDD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h	General Purpose Register 16 Bytes	117h	General Purpose Register 16 Bytes	197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
General Purpose Register 96 Bytes		General Purpose Register 80 Bytes	EFh	General Purpose Register 80 Bytes	16Fh	General Purpose Register 80 Bytes	1EFh
		accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h-7Fh	1F0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

PIC18F452'nin Saat ve Osilatör Uçları

PIC18CXX mikrodenetleyicilerinde 4 çeşit osilatör bulunmaktadır. Kullanıcı bu 4 çeşitten birini seçerek iki konfigürasyon bitini (FOSC1 ve FOSC2) programlayabilir. Bu osilatör çeşitleri Çizelge 2,6'da verilmiştir. PIC18F452'de saat uçları 13. ve 14. bacaklardır. Hazırlanacak olan PIC programlarında kullanılan osilatör tipi PIC programının çalışma hızını ve hassasiyetini etkileyeceğinden dolayı amaca uygun bir osilatör devresi kullanılmalıdır. Çizelge 2,5'de farklı osilatör çeşitleri ve özellikleri görülmektedir. Osilatör tipinin seçiminde dikkat edilecek bir başka nokta ise, seçilecek olan osilatörün kullanılan PIC'in özelliğine uygun olmasıdır.

Çizelge 2.5 Osilatör Çeşitleri ve Özellikleri (Kaya 2007).

Osilatör Tipi	Tanımı	Özelliği	Frekansı
LP	Kristal osilatör veya seramik rezonatör	Asgari akım	40KHz
XT	Kristal osilatör veya seramik rezonatör	Genel amaçlı	4MHz
HS	Kristal osilatör veya seramik rezonatör	Yüksek hız	20MHz
RC	Direnç/Kapasitör zaman sabitli	Düşük maliyet	4MHz

Kristal Osilatör / Seramik Rezonatör

XT, LP ve HS modları, RC osilatörlere nazaran çok daha hassastırlar. Bu modlar, kristal osilatör veya rezonatörlerin, OSC1 / CLKIN ve OSC2 / CLKOUT uçlarına bağlanmalarıyla kurulmaktadır. Çizelge 2.6 'da hangi frekansta kaç pF'lık kondansatör kullanılması gerektiği belirtilmiştir.

Çizelge 2.6 Frekansa Göre Kondansatör Seçimi (Kaya 2007).

Osilatör Tipi	Frekans	Kondansatör
LP	32 KHz	33 – 68 pF
	200 KHz	15 – 47 pF
	100 KHz	47 – 100 pF
XT	500 KHz	20 – 68 pF
	1	15 – 68 pF
	2	15 – 47 pF
	4	15 – 33 pF
	8	15 – 47 pF
HS	20	15 – 47 pF

RC Osilatör

Zamanlamanın çok hassas olmadığı durumlarda RC ikilisi osilatör kaynağı olarak kullanılmaktadır. RC osilatör, maliyetin azaltılmasını sağlamaktadır. Kullanıcı dış R ve C elemanlarının toleransı nedeniyle meydana gelen değişiklikleri de dikkate almalıdır. Direncin değeri 3 ila 100 KOhm arasında seçilmelidir. 1MOhm gibi yüksek direnç değerleri osilatörü gürültü ve nem gibi çevresel etkilere karşı duyarlı hale getirir. Direnç 2 KOhm değerinin altında ise, osilatör kararsız hale gelebilmekte, hatta tamamıyla durabilmektedir.

2.4.4 PIC18F452'nin Besleme Uçları ve Beslenmesi

PIC18F452'nin besleme gerilimi 11, 12 ve 31, 32 numaralı bacaklardan uygulanmaktadır. 11 ve 32 numaralı Vdd ucu +5 V'a ve 12, 31 numaralı Vss ucu toprağa bağlanır. PIC'e ilk defa enerji verildiği anda meydana gelebilecek gerilim dalgalanmaları nedeniyle, oluşabilecek istenmeyen arızaları önlemek amacıyla 100nF'lık dekuplaj kondansatörünün devreye bağlanması gerekmektedir. PIC'ler CMOS teknolojisi ile üretildiklerinden 2 ila 6 V arasında çalışabilmektedirler. +5 V'luk bir gerilim ise ideal bir değer olmaktadır.

2.4.4.1 PIC18F452'nin Reset Uçları

Kullanıcının programı kasti olarak kesip başlangıca döndürebilmesi için PIC'in 1 numaralı ucu MCLR olarak kullanılmaktadır. MCLR ucuna 0 Volt uygulandığında programın çalışması başlangıç adresine döner. Programın ilk başlangıç adresinden itibaren tekrar çalışabilmesi için, aynı uca +5 Volt gerilim uygulanmalıdır (Kaya 2007).

2.5 SERİ HABERLEŞME

Bilgisayarların dış dünya ile haberleşmeleri kullanıcıya veya kullanım ihtiyacına göre seri veya paralel olarak gerçekleştirilebilir. Paralel giriş / çıkışta 8 bit data, 1 bit data hazır, 1 bit data istek, 1 bit GND olmak üzere 11 tel ile iletişim gerçekleştirilir. Bu haberleşme yöntemi hızlıdır. Ancak uzak mesafelerle yapılan haberleşmede maliyet çok fazladır. Seri haberleşmede biri veri, diğeri GND olmak üzere iki tel yeterli olduğundan dolayı seri haberleşmeye geçilmiştir. Ayrıca haberleşmede gürültüden kurtulma çok önemlidir. Paralel haberleşmede lojik-0 için 0V, lojik-1 için 5V kullanılmaktadır. Burada 1-2V arası belli bir değere karşı düşmemektedir. Ancak gürültü bu değeri aşabilir. Seri haberleşmede ise lojik-1 (-3)-(-25)V, lojik-0 ise (3)-(25)V arasında kabul edildiğinden gürültülere daha dayanıklıdır. Yani $\pm 25V = 50V$ dalgalanma aralığı vardır. Seri giriş/çıkış arayüzleri bilgisayarla paralel, dış dünya ile seri haberleşen elemanlardır. Bilgisayardan aldıkları paralel veriyi kendi içerisinde seriye çevirip dışarıya verirken, dışarıdan aldığı seri veriyi paralele çevirip bilgisayara verir (Kaya 2007).

2.5.1 Seri Haberleşme Metodları

Seri iletişimdeki amaç harfleri ifade eden 8 veya 16 bit’lik bir dizi veriyi diğer bir çevre birimine ulaştırmaktır. Seri iletişimde senkron ve asenkron veri iletişimi adı verilen iki metod kullanılmaktadır. Senkron “eş zamanlı” , asenkron ise “eş zamansız” anlamına gelmektedir.

2.5.1.1 Asenkron Veri İletimi

Klavye, fare ve modem gibi çoğu PC cihazları asenkron veri iletimi metoduyla veri iletimi yaparlar. Gönderilen 8 bit’lik veri başında start ve stop bit’leri bulunduğundan senkron veri iletimine göre yavaş olmasına rağmen daha az kablo kullanılması en büyük avantajıdır.

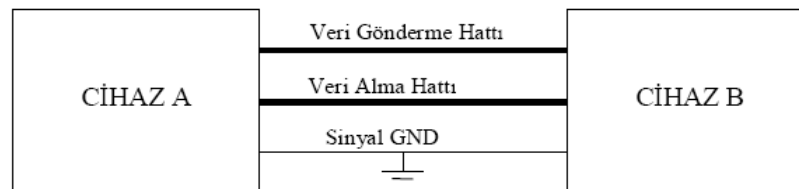
Asenkron iletişimin temel özellikleri şunlardır.

- Transfer edilecek veriler karakter bazında yapılır.
- Gönderilecek her bir karakter verisinin başlangıcını ve bitişini belirten start ve stop

Bit’leri de gönderilir.

- Karşılıklı olarak haberleşecek cihazların iletişim parametreleri aynı olmalıdır (Hız, Kodlama seti vb.)

Asenkron seri iletişimin bağlantı şeması şekilde gösterilmiştir



Şekil 2.32 Asenkron veri iletişimin basit bağlantı şeması (URL-1 2009).

Asenkron veri iletişiminde alıcı ve verici için dikkat edilmesi gereken en önemli protokollerden birisi hızdır. Göndericinin saat frekansı ile alıcının saat frekansı arasındaki farklılık verilerin doğru alınamamasına neden olur.

2.5.1.2 Senkron Veri İletimi

Senkron veri iletişiminde veri ile birlikte saat sinyali de gönderilir. Bu durum start ve stop bit'lerinin gerekliliğini ortadan kaldırır. Aynı zamanda senkron iletişim karakter blokları bazında olduğu için asenkron iletişime göre daha hızlıdır. Ancak daha karmaşık devreler içerdiğinden, pahalıdır. Senkron deyimi alıcı ile vericinin eşzamanlı çalışması anlamına gelir. Saat sinyali ihtiyacı da bu durumdan ileri gelir. Şekil 2.33 senkron iletişimi yapan cihazların basit bir bağlantı şemasını gösterir.



Şekil 2.33 Senkron seri iletişimin basit bağlantı şeması (URL-1 2009).

Senkron iletişimin temel özellikleri şunlardır;

- Hata saptama ve koruması yapılır.
- Hız genellikle gönderici tarafından saptanır
- Senkron cihazlar asenkron cihazlara göre daha hızlı ve daha pahalıdır
- Veriler bloklar halinde gönderilir
- Blok formatları kullanılan protokole göre değişir.

2.5.2 İletim Yönlerine Göre Seri İletişim Modları

Seri veri iletişimi verinin karşılıklı veya tek yönlü oluşuna göre full duplex, half duplex ve simplex olmak üzere 3 farklı biçimde olabilir.

- Full duplex (çift yönlü): Bu tür iletişimde her iki taraf da telefon konuşmalarında olduğu gibi aynı anda veri gönderebilir.

- Half duplex (yarım çift yönlü): İki nokta tek bit yol üzerinden telsiz konuşmalarında olduğu gibi farklı zamanlarda sadece bir yönde veri gönderebilir.
- Simplex (tek yönlü): İki nokta arasında tv ve radyo yayınlarında olduğu gibi sadece bir yönde veri gönderilebilir.

2.5.3 USART Birimi Ve Özellikleri

PIC18F452'in içerisinde donanımsal olarak senkron ve asenkron haberleşme yapabilmeyi sağlayan USART birimi bulunmaktadır. USART biriminin çalışma modu düzenlenerek PC ile iletişim, D/A çevirici veya A/D çevirici entegrelerle iletişim EEPROM ile iletişim sağlanabilir. USART birimi aşağıdaki iletişim modlarında çalıştırılabilir.

Asenkron (full duplex – tam çift yönlü veri iletimi)

Senkron – master (half duplex – yarı çift yönlü)

Senkron – slave (half duplex – yarı çift yönlü)

USART biriminin adı (Universal Synchronous Asynchronous Receiver Transmitter) kelimelerinin baş harflerinden alınmıştır. Kelimelerin Türkçe karşılığı “Evrensel Senkron Asenkron Alıcı Verici “ dir. Görüldüğü gibi USART biriminin senkron haberleşme özelliği de bulunmasına rağmen PIC ile hedef tespiti projesinde tasarımcılar tarafından en çok kullanılan ve piyasadaki birçok cihazın üzerindeki seri haberleşme birimi olan asenkron veri iletişimi kullanılmıştır.

İletişim yaparken amaç, verileri ifade ettiğimiz “1” ve “0” durumlarını gönderen taraftan alıcı tarafa bildirmektir. Asenkron seri iletişimde transfer edilecek olan verinin (8 veya 9 bit) başına start bit'i, verinin sonuna da stop bit'i konulur. Böylece gönderilen veri bloğu alıcı tarafa bildirilmiş olur. Start bit'i “0” seviyesinde, stop bit'i de daima “1” seviyesindedir. “1” seviyesinde bekleyen bir devreden veri gönderilmiyor demektir. İlk gönderilen “0” seviyeli bir sinyal, alıcı tarafından 8 bit'lik bir veri gönderilmeye başlanıyor olarak algılanır. Bundan sonra sinyal hızına bağlı olarak verinin transferi tamamlanır. Transfer edilen veri bloğunun sonunda “1” seviyesindeki sinyal veri sonunu bildirir. Sinyal hızı gönderici ve alıcı taraflarındaki bağımsız bir timer (zamanlayıcı) tarafından sağlanır. Bu zamanlayıcılar arasındaki çok küçük farklar veri iletiminin doğru yapılmasını engelleyebilir.

USART ile asenkron mod veri iletişimde PIC18F452'nin RX ucu alıcı, TX ucu da verici ucu olarak kullanılmaktadır. Bu nedenle aynı zamanda veri gönderme ve alma işlemi yapmak mümkün olabilmektedir (full duplex).

2.5.3.1 Sinyal Hızının Belirlenmesi

USART'ın sinyal hızı BRG üretici tarafından kontrol edilir. USART'ın hem senkron hem de asenkron modunda çalışmasındaki sinyal hızı BRG ile belirlenir. BRG kontrolü ise SPBRG adlı saklayıcı ile yapılır. BRG 8 bit'lik bağımsız olarak çalışan bir zamanlayıcının kontrolünü yapmaktadır. Asenkron çalışma modunda TXSTA saklayıcısının BRGH adlı 2. bit'i sinyal hızının modunu (yüksek veya alçak hız modu) belirler. SPBRG kayıtçısına herhangi bir değer yazıldığında BRG zamanlayıcısı resetlenir. PIC'in saat frekansına ve SPBRG saklayıcısına bağlı olarak BRG'nin ne değer alacağı aşağıdaki formül ile hesaplanabilmektedir.

Mod Düşük hız (BRGH=0) Yüksek hız (BRG=1)

Asenkron (SYNC=0) $FOSC/(64(X+1))$ $FOSC/(16(X+1))$

Senkron (SYNC=1) $FOSC/(4(X+1))$

FOSC: PIC saat sinyali frekansı

X: SPBRG kayıtçısı içerisindeki değer (0 – 255 arasında)

Baud rate bit/sn olarak ifade edilen seri iletişim hızı birimidir. Örneğin bir bit'in gönderilme süresi 3.33msn ise baud rate $1\text{sn}/3.33\text{msn} = 300$ baud olur. Yani 1 sn içerisinde 300 bit gönderilmiştir anlamına gelir.

Çizelge 2.7 Senkron haberleşme frekans tablosu (URL-1 2009).

Fosc = 4 MHz		Fosc = 10 MHz		Fosc = 20 MHz	
Baud Rate	SPBRG (desimal)	Baud Rate	SPBRG (desimal)	Baud Rate	SPBRG (desimal)
300	207	-	-	-	-
1200	51	1202	129	1221	255
2400	25	2404	64	2404	129
9600	6	9766	15	9766	31
19200	2	19531	7	16531	15
28800	1	31250	4	31250	9
33600	-	31250	4	34722	8
57600	0	52083	2	62500	4
HIGH	255	610	255	1221	255
LOW	0	158250	0	312500	0

Çizelge 2.8 Asenkron haberleşme frekans tablosu (URL-1 2009).

F _{osc} = 4 MHz		F _{osc} = 10 MHz		F _{osc} = 20 MHz	
Baud Rate	SPBRG (desimal)	Baud Rate	SPBRG (desimal)	Baud Rate	SPBRG (desimal)
300	-	-	-	-	-
1200	207	-	-	-	-
2400	103	2441	255	-	-
9600	25	9415	64	9615	129
19200	12	19531	31	19231	64
28800	8	28409	21	29070	42
33600	6	32895	18	33784	36
57600	3	56818	10	59524	20
HIGH	255	2441	255	4883	255
LOW	0	625000	0	1520000	0

2.5.3.2 USART İle Asenkron Veri Göndermek

USART ile asenkron veri gönderiminde RCSTA kayıtçısının SPEN adlı 7. biti “1” yapılarak RC6 ucu TX ucu olarak kullanılmaya hazırlanmış olur. Gönderilecek olan veri TXREG kayıtçısına yüklendiğinde bu veri donanım tarafından TSR’ye (Transmit Shift Register-Gönderici Kaydırmalı Kaydedici) transfer edilir ve TX ucundan gönderilir. Verinin gönderilme işlemi TXSTA kayıtçısının TXEN adlı 5.bit’i “1” yapılarak aktif edilir. En düşük değerlikli (LSB) bit’i ilk olarak gönderilir. TXREG kayıtçısının içeriği TSR kaydırmalı kaydedicisine gönderildiğinde PIR1 kayıtçısının TXIF adlı 4. bit’i “1” olur ve bu anda bir kesme oluşur. Bu kesmeyi algılamak için PIE1 kayıtçısının TXIE biti “1” yapılmış olmalıdır. Burada oluşan kesme, TXREG kayıtçısının içeriğinin TSR’ye gönderildiğini ve yeni bir verinin TXREG kayıtçısına yüklenebilecek duruma geldiğini belirtir. TXREG kayıtçısına her yeni veri yüklenişinde oluşan bu kesmenin algılanmasıyla takip eden veriler arka arkaya devamlı olarak gönderilir. TXIF bit’ini yazılımsal olarak sıfırlamak mümkün değildir. TXIF bit’i sadece TXREG kayıtçısına yeni bir veri yüklendiğinde otomatik olarak sıfırlanır. TSR kayıtçısındaki verinin seri olarak gönderilmesi bittiğinde TXSTA kayıtçısının TMRT bit’i “1” olur. Bu durum TSR kayıtçısının boşaldığının gösterir. Verinin gönderilme işleminin bittiğini belirten herhangi bir kesme olmadığından TSR kayıtçısının boş olup olmadığını anlamak için TMRT bit’i periyodik olarak kontrol edilmelidir. TSR kayıtçısı veri belleği üzerinde bulunmamaktadır. Bu nedenle kayıtçıya direkt olarak erişerek yazma/okuma işlemi yapmak mümkün değildir. Eşlik (parity) bit’i donanımsal olarak desteklenmemektedir, ancak yazılımsal olarak 9. bit belirlenebilir. Bunun için TXSTA kayıtçısının TX9D adlı 0. bit’ine

gönderilecek olan verinin 9. bit'i yazılır. 9bit'lik veri iletimi yapılacağı da yine TXSTA kayıtçısının TX9 adlı 6. bit'i "1" yapılarak belirlenir. TX9D bit'ine yazılacak olan 9. bit TXREG kayıtçısına 8 bit'lik veri yazılmadan önce yazılmalıdır. Çünkü TXREG kayıtçısına veri yazılır yazılmaz veri gönderilmeye başlanır. Bu nedenle 9. bit'i sonradan yazmak mümkün olmaz. Veri gönderimi yapılırken TXEN bit'ini sıfırlamak, göndericiyi resetleyeceğinden RC6/TX/CK ucunun yüksek empedans durumuna gelmesine neden olur. Asenkron veri gönderimi yapılırken yapılması gereken işlemler aşağıda verilmiştir;

1. İstenilen baud rate hızı değeri SPBRG kayıtçısına yüklenir. Eğer yüksek hızlı baud rate isteniyorsa TXSTA kayıtçısının BRGH alı 2. bit'i "1" yapılır.

2. Asenkron seri iletişim portunu aktif etmek için TXSTA kayıtçısının SYNC adlı 4. bit'i "0", RCSTA saklayıcısının da SPEN adlı 7. bit'i "1" yapılır.

3. Kesme kullanımı isteniyorsa PIE1 kayıtçısının TXIE adlı 4. bit'i "1" yapılır.

4. 9 bit'lik veri gönderimi isteniyorsa, TXSTA kayıtçısının TX9 adlı 6. bit'i "1" yapılır.

5. Göndermeyi aktif yapmak için TXSTA kayıtçısının TXEN adlı 5. bit'i "1" yapılır. Bu anda TXREG kayıtçısına veri yazmak mümkün duruma gelir ve PIR1 kayıtçısının TXIF adlı 4. bit'i "1" olur.

6. 9 bit'lik veri gönderme seçilmişse, 9. bit TXSTA kayıtçısının TX9D adlı 0. Bit'ine yazılmalıdır.

7. 8 bit'lik veri TXREG kayıtçısına yüklenerek göndere işlemi başlatılır (URL-1 2009, Kaya 2007).

2.6 LCD

Lcd panelleri en basit olarak PC'nin seri veya paralel portundan yada bir PicMicro'dan (örneğin 16F84, 16F877... gibi) veri göndererek kontrol edilebilir. Çeşitli satır ve karakter boyutlarında olabilirler, örneğin; 1x8, 2x8, 1x16, 1x20, 2x16, 2x20, 2x20, 2x10, 1x40, 2x40 gibi... İlk sayı satır sayısını, ikinci sayı bir satırdaki karakter sayısını belirtir.

LCD'nin PC veya PIC ile haberleşmesi için 14 pin'e ihtiyacı vardır. Bu pinlerin 14'ü ya aynı sıra halinde LCD'nin sol üst kenarında ya da 7'şerli iki sıra halinde arka tarafta bulunurlar. LCD'ler paralel veri iletişimi için 4 veya 8 bit kullanırlar.



Şekil 2.34 LCD basit bağlantı şeması (URL-7 2009).

Çizelge 2.9 LCD Bacak bağlantıları

LCD Pinleri Pin	İşlevi
1	Vss, Toprak (Ground), Logic Vss, Logic Ground
2	Vcc, +5 Volt, Logic Vcc, Logic Power
3	VLc, VBias, Bias, Contrast (Kontrast)
4	RS, Register Select
5	R/W, Read/Write(Okuma ya da Yazma Modu)
6	E, Enable, Strobe
7-14	D0-D7 (DATA girişleri)
15*	Led+, A, Backlight+, Backlight Anode (LCD Panel ısıığı (+5 Volt))
16*	Led-, K, Backlight-, Backlight Cathode (Toprak (Ground))

*15 ve 16. bacaklar bazı displayler de yoktur. Bu uçlar paneli aydınlatmak için kullanılırlar.

VO (kontrast) : 3 numaralı kontrast girişini doğrudan 2.5K 'lık (10K, 20K da olur) bir direnç ile toprağa bağlanır. Buradaki direncin değerini arttırarak kontrast düşürülebilir, azaltarak kontrast yükseltilebilir. Eğer bu kontrast ile sürekli oynanmak istenirse buraya 2K5'lık bir ayarlı direnç (potansiyometre) bağlanabilir.

RS (register select): RS girişi "0" (ground) olduğu zaman LCD panelin komut saklayıcısı (PIC' ten veriyi komutlar halinde alır), +5V olduğu zaman ise veri saklayıcısı (veriyi 8 veya 4 bitlik karakter bilgisi şeklinde alıp verir) seçilmiş olur.

R/W (read/write): Bu giriş ground yani "0" olduğunda, LCD panel yazma modunda dır. Karakterleri yazdırırken, yazma modunu kullanılır, bu yüzden bu bacak panele yazı yazdırmak için topraklanır.

E (enable) : LCD ve pinler arasındaki gerçek veri alışverişini sağlayan bacaktır. Bu girişi microchip'e program aracılığıyla tanıttuktan sonra PIC kendisi veri gönderileceği zaman 230 nanosaniyeden fazla olmak üzere bu bacağı enable pulsu sağlar.

DATA: 7-14 numaralı pinleri doğrudan bir PORT' un pinlerine bağlanır. Böylece istenilen komut ya da veri LCD panele gönderilir. Veriyi 4 bitlik veri yoluyla mı yoksa 8 bitlik veri yoluyla mı LCD'ye gönderilmeli sorusuna yanıt olarak PIC' in bacak sayısının az olduğunu düşünülerek 4 bitlik veri yolunun seçilmesi uygun olur.

Ama paralel porttan veri göndermek isterseniz 8 bitlik veri yolu kullanılmalıdır ve tamamı bir PORT' a bağlanmalıdır. 4 bitlik veri yolu kullanılırken de bir portun alt 4 bitine veya üst 4 bitine bağlanmalıdır.

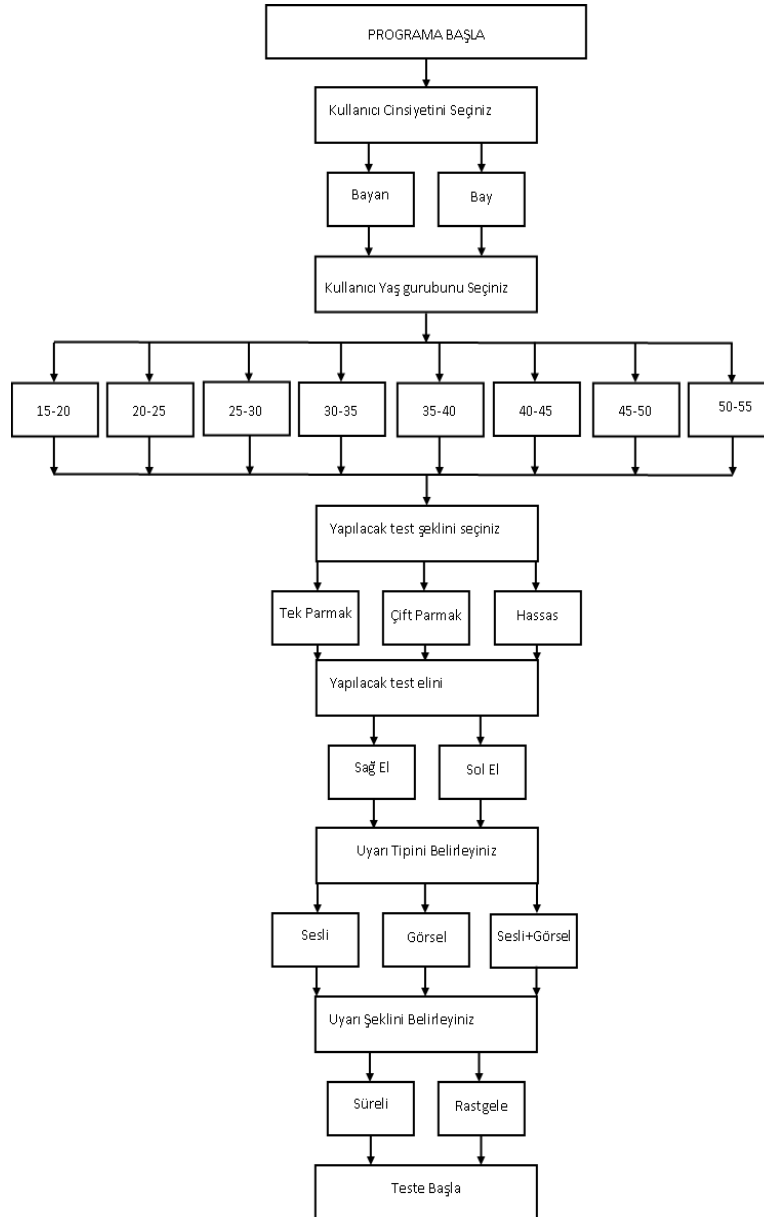
Enable, Register Select ve R/W uçları herhangi bir portun pinine bağlanabilir ve bu program içerisinde belirtilmelidir (Kaya 2007).

2.7 YAZILIMLAR

2.7.1 Mikrodenetleyici Yazılımı

Kullandığımız mikrodenetleyici için c programlama dilini kullanmaktayız. Daha sonra yazmış olduğumuz kodlar hex dosyasına derlenip pic'e yüklenmektedir. Program ek'de bulunmaktadır ve gerekli açıklamaları belirtilmiştir.

Test algoritması ise:



Şekil 2.35 Program Dallanması (algoritma).

Matematiksel formüller ise şu şekildedir;

Toplam süre (ITI_{tot}) ,

$$ITI_{tot} = \sum_{n=1}^{n=N} |ITI(n)| \quad (2.1)$$

Tapping rate (vuru frekansı f_t),

$$f_t = \frac{N}{ITI_{tot}} * 1000 \quad (2.2)$$

Ortalama ITI ($\langle ITI \rangle$),

$$\langle ITI \rangle = \frac{ITI_{tot}}{N-1} \quad (2.3)$$

ITI'nin standart sapması (σ_{ITI}) ve CV,

$$\sigma_{ITI} = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N-1} (ITI(n) - \langle ITI \rangle)^2} \quad (2.4)$$

$$CV = \frac{\sigma_{ITI}}{\langle ITI \rangle} \quad (2.5)$$

denklemlerdir.

2.7.2 Matlab Yazılımı

Elde edilen değerler ve parmağın her vuruda havada kalış dataları alınarak seri eeproma kaydedilmekte ve seri veri transferi ile matlab'a aktarılmaktadır. Matlab da oluşan dataların grafikleri çizdirilmekte ve inceleme yapılabilmektedir. Tek parmak vuru, Çift parmak vuru ve hassas vuru ölçümlerine göre hazırlanmış olan matlab kodları ekte bulunmakta ve gerekli açıklamaları yapılmaktadır.

Test cihazımızda elde edinilmek istenilen veriler ve bu verilerin işlenmesi aşağıda belirtilmiştir. Test'e başlamadan kişi ile test bilgilerinin neler olduğu sisteme kaydedilmekte ve daha sonra test seçenekleri belirlenerek ölçüme başlanmaktadır.

Test kriterleri ise;

A-Cinsiyet – (Kadın – Erkek)

B-Yaş – (15-20 / 20-25 / 25-30 / ...)

C-Test şekli –(Tek parmak-Çift parmak-Hassas Ölçüm)

D-Test eli – (Sağ – Sol)

E-Uyarı tipi – (Sesli – Görsel)

F-Test tipi – (Zamanlı – Zamansız)

G- Test süresi(sn) – (30 – 60 -90)

Kıstaslarına göre ayrılmak üzere;

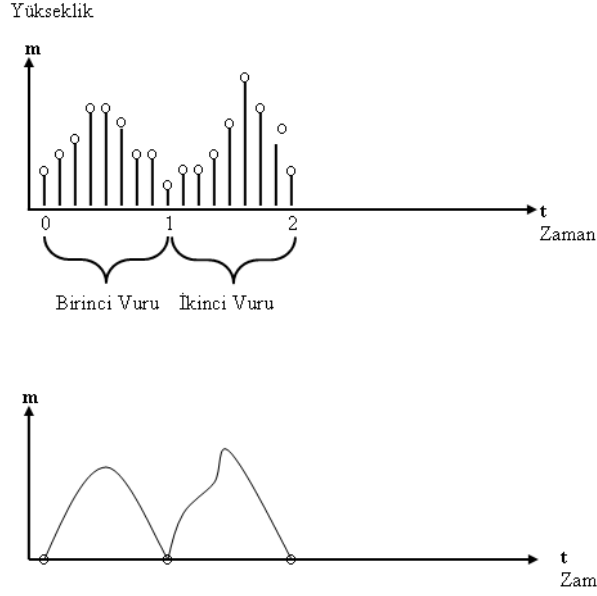
*Belirlenen sürede yapılan toplam vuru sayısı

*Vurular arası geçen süre

*Vurular arası parmağın havada kalış dataları (Yükseliş ve Alçalış)

Değerleri ölçülüp database'e kaydedilmektedir. Daha sonra hesap yöntemleri ile alınan değerlerin ortalaması, standart sapması ve varyansı hesaplanarak lcd ekrana yazdırılmaktadır. Bununla beraber istenilmesi durumunda datalar seri haberleşme ile bilgisayara gönderilerek burada istenilen eğri ve grafikler oluşturulmakta yapılan ölçümlerin istatistikleri saklanabilmektedir.

Bir vurudan diğer vuruya kadar parmağın yapmış olduğu yükseliş ve alçalışın zamana göre değerlerini alıyoruz. Kullandığımız sensör analog çıkışlı olduğundan tasarladığımız bir analog dijital çevirici yardımıyla bu değerleri dijitale çevirip belleğimize kaydediyoruz. Bu aşamada üretici firma tarafından verilen eğriye göre alınan gerilim değerleri mesafe birimine (cm) çevrilmektedir.



Şekil 2.36 Parmak vurular arası geçen süre ve yükseklik grafikleri.

Yukarıdaki şekilden de anlaşılacağı üzere her vuruda parmağın yükseliş ve alçalış değerleri farklılık göstermektedir. Örneğin butona aynı sürede aynı sayıda yapılan iki ayrı vurudan, yakın mesafeden yapılan vuru ile uzak mesafeden yapılan vurunun performans değeri bir değildir.

Yapılan bir deneyde;

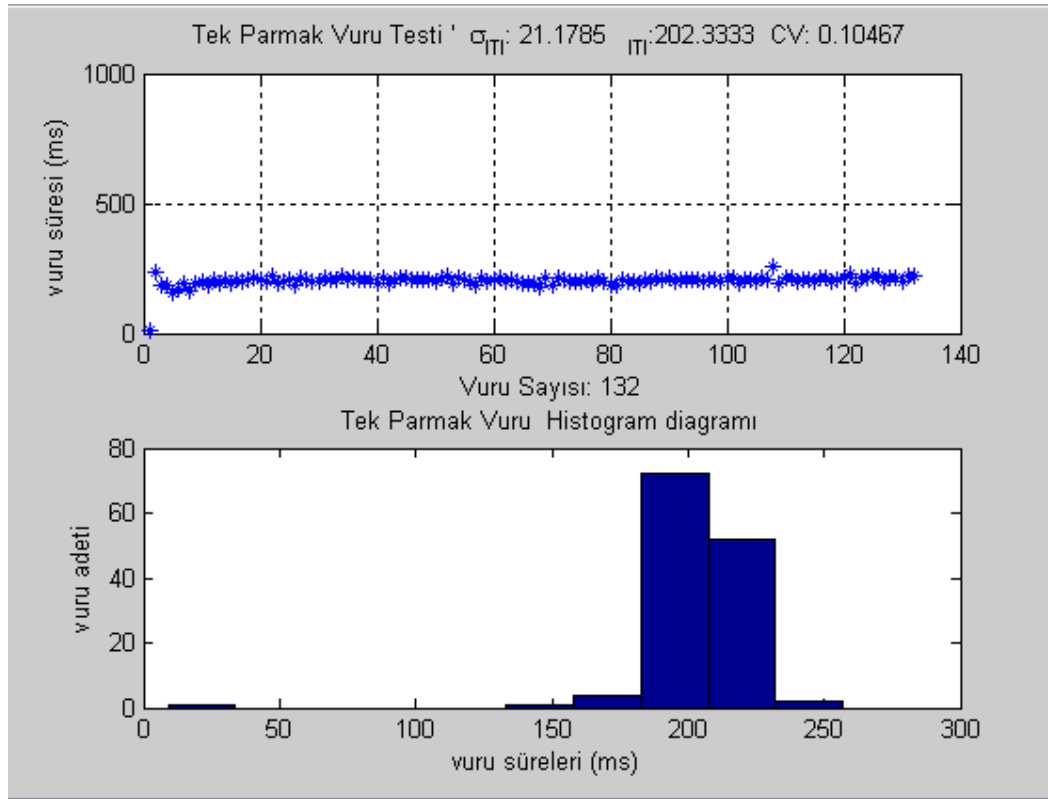
- 1.Tek parmak ölçümü
- 2.Çift parmak ölçümü
- 3.Hassas ölçüm

Tesetleri yapılmış ve değerler ile alınan grafikler şu şekildedir.

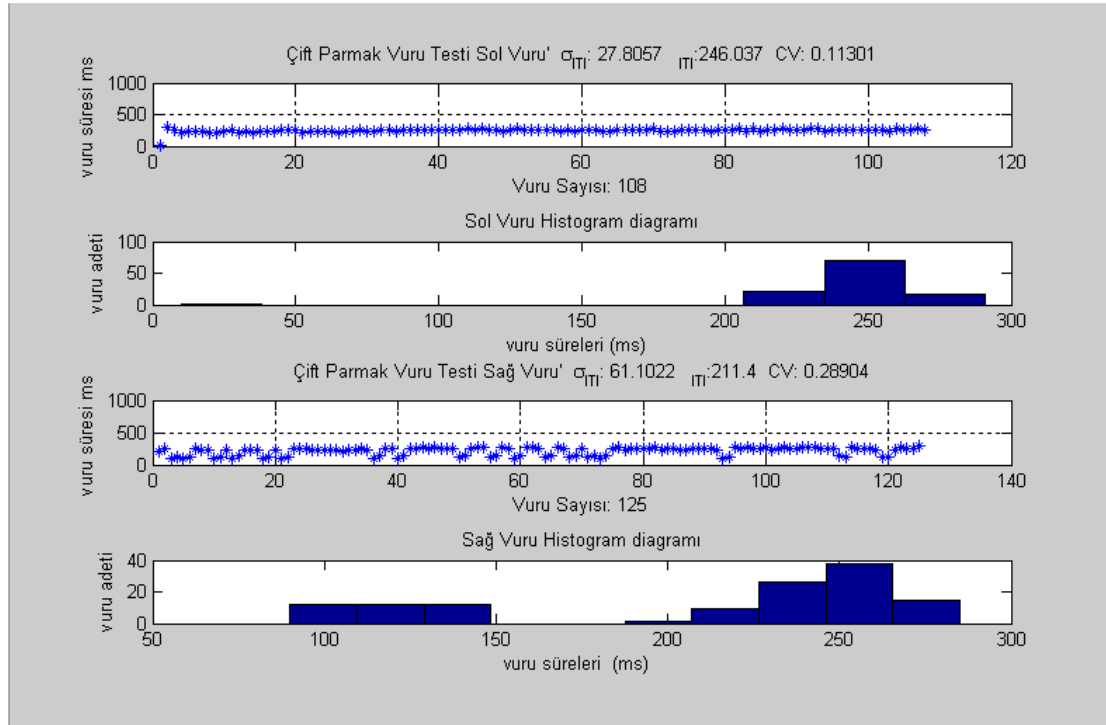
Çizelge 2.10 Elde edilen teset sonuçları.

Tek Parmak (30 sn)	Çift Parmak (30 sn)		Hassas (30 sn)
Vuru Sayısı=132	Sol Vuru	Sağ Vuru	Alınan
Ortalama =202	Vuru Sayısı=108	Vuru Sayısı=125	Örnek Sayısı=502
Ssapma =21	Ortalama =246	Ortalama =211	Vuru Sayısı =116
Varyans =0,104	Ssapma =27	Ssapma =61	
	Varyans =0,113	Varyans =0,289	

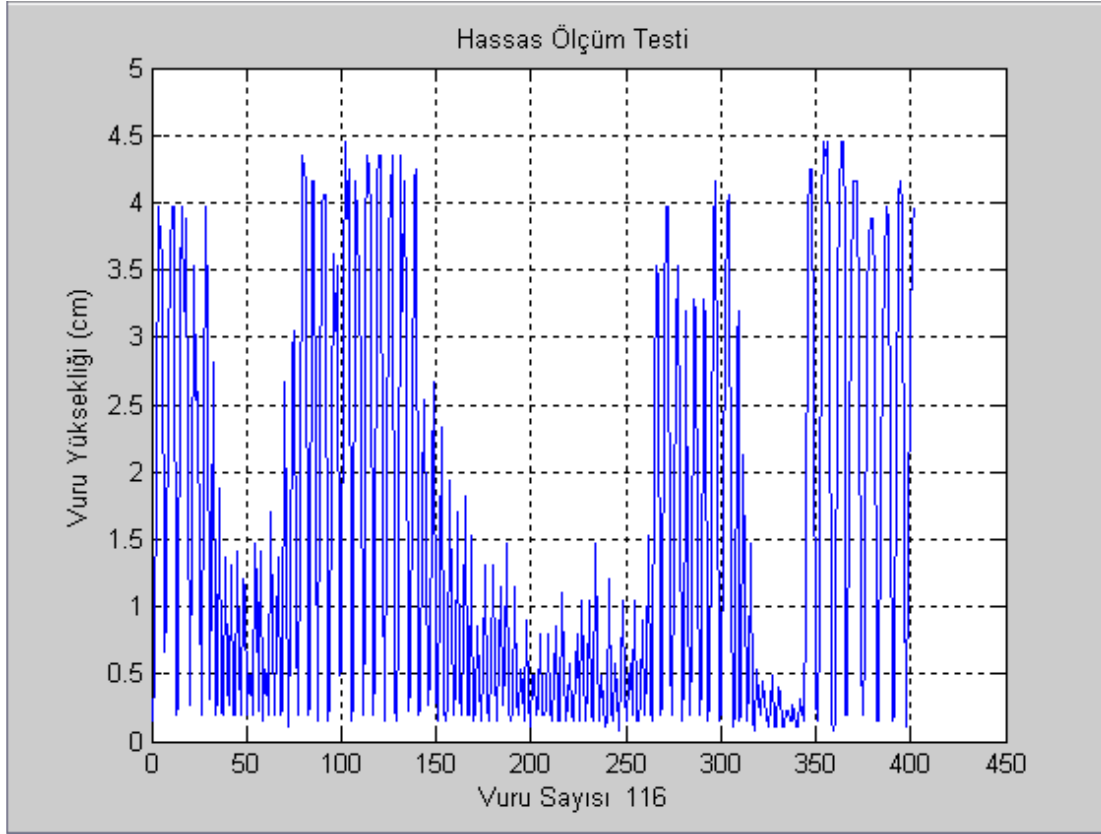
Elde edilen deęerlere ait grfikler ise;



Şekil 2.37 Tek parmak vuru grafięi.



Şekil 2.38 Çift parmak vuru grafięi.



Şekil 2.39 Hassas ölçüm grafiği.

Yapılan testler 30 sn test süresi baz alınarak tek parmak, çift parmak, hassas ölçüm basamakları uygulanarak yapılmıştır. Elde edilen sonuçlar hem lcd ekrana yazdırılmıştır hemde bilgisayara aktarılarak doğrulanmıştır. Sonuçlar incelendiğinde;

Tek parmak vuru testinde kişinin vuru performansının yüksek olduğu yapmış olduğu vuru sayısından anlaşılmakta olup vuru yapılan elini kullanma kabiliyetinin yüksek olduğu cv değerinin ve sapmasının küçük olmasından anlaşılmaktadır.

Çift parmak vuru testinde ise kişinin sol vuru yaptığı elini daha aktif kullandığı ve bu parmağa ait vuru süresinin daha stabil olduğu gözlemlenmiştir.

Yapılan hassas ölçüm vuru testinde ise parmağın havada kalış süreleri ile yükseliş değerleri incelendiğinde kişinin, teste başladığı kondisyonu arttırmaya çalıştığı bu zaman aralığında parmağın yükseklik değerlerinin düştüğü görülmüştür. Belirli süre dinlenip tekrar vuru

sayısını arttırmak için yüksekliği düşürdüğü ve sonra vuru ahenginden uzaklaştığı görülmüştür. Buradan hareketle vuru performansının düşük, dikkatinin dağınık ve kondisyonun yeterli düzeyde olmadığı görülmektedir.

Yapılan deneylerde ortalama vuru sürelerinin 200 ila 300 ms arasında olduğu görülmüştür. Bu değer, kişinin aktif kullandığı parmağı, yaşı, yapmış olduğu meslek veya spor dalı ile değişim gösterdiği görülmüştür. Bununla beraber kişilerin aynı vuruyu yapma tekrarı (vuru ahengi) ile hesaplanan bir değer olan cv (varyans) değeri arasında ters bir orantılı olduğu tespit edilmiştir. Uzun test sürelerinde bu değer büyüdüyü yani kişilerin vuru tekrarlarının ahenginin bozulduğu gözlemlenmiştir. Kişinin kondisyonu ne kadar yüksek ise bu değer o oranda küçük kalır. Hesaplanan bir diğer değerimiz ise standart sapma değeridir. Bu değer kişinin fazla vuru yapması ve vuruları yaparken aynı geçikme süresi ile yapmasıyla doğru orantılıdır. Buradanda diğer bir kriter olan performans değerini inceleyebilmekteyiz.

Bakırköy Spor Klübü Basketbol Yıldız takımı ile yapılan testlerin sonuçları Çizelge 2.11’de verilmiştir. Bu testlerin ortalamasına bakıldığında ortalama vuru süresi 274 ms sol vuru, 265 ms sağ vuru olduğu görülmüştür. Takım içerisinde yapılan incelemede Berk TUBA isimli denek’in vuru adeti ile standart sapması incelendiğinde diğer arkadaşlarına göre oldukça düşük olduğu görülmüştür. Antrenörü Zeynep ATAGÜL ile yapılan konuşma ve yapılan antreman gözleminde Berk’in oldukça aktif ve kondisyonlu olduğu izlenilmiştir. Yine yapılan bir diğer gözlemde şut yüzdesi diğer arkadaşlarına göre yüksek olan Berkin Al’in ise cv ve ssapma değerlerinin diğer arkadaşlarından daha küçük olduğu görülmüş ve isabetli şut sayısı ile bu değerlerin bir ilişkisi olduğu tespit edilmiştir.

Spor yapmayan aynı yaş gruplarından almış olduğumuz değerlerin yapanlara kıyasla daha düşük olduğu tespit edilmiştir. Örnekle, ortalama vuru sürelerinin 320 ila 380 ms arasında olduğu gözlemlenmiştir.

Farklı bir tespitimiz ise futbolcuların çoğu değerlerinin basketbolculara kıyasla düşük çıkmasıdır. Buda yapmış oldukları spor dallarında ellerini kullananların kullanmayanlara oranla daha iyi sonuç aldıklarını göstermektedir. Genel olarak bakıldığında herhangi bir sporla uğraşanların kondisyon değerlerinin spor yapmayanlara kıyasla daha iyi olduğu gözlemlenmiştir.

Çizelge 2.11 Yapılan ölçümler ve alınan sonuçlar.

No	Adı Soyadı	Cinsiyet	Yaş	Sağ El	Sol El
1	Çağrı Aytemiz	E	12	Ss :68 Ort :261 Vuru:103 Cv :0,26	Ss :122 Ort :272 Vuru:98 Cv :0,44
2	Berkin Al	E	13	Ss :20 Ort :223 Vuru:121 Cv :0,896	Ss :54 Ort :221 Vuru:122 Cv :0,244
3	Aytunç Yıldırım	E	12	Ss :80 Ort :346 Vuru:78 Cv :0,231	Ss :94 Ort :322 Vuru:83 Cv :0,291
4	Ekin Candan	E	11	Ss :65 Ort :333 Vuru:82 Cv :0,195	Ss :82 Ort :325 Vuru:85 Cv :0,252
5	Berk Tuba	E	14	Ss :35 Ort :191 Vuru:136 Cv :0,183	Ss :67 Ort :178 Vuru:146 Cv :0,376
6	Eren Sönmez	E	13	Ss :43 Ort :296 Vuru:90 Cv :0,145	Ss :56 Ort :282 Vuru:94 Cv :0,195
7	Göker Korel	E	12	Ss :47 Ort :340 Vuru:80 Cv :0,138	Ss :72 Ort :318 Vuru:85 Cv :0,226
8	Mete Kılıç	E	13	Ss :72 Ort :372 Vuru:72 Cv :0,195	Ss :87 Ort :320 Vuru:82 Cv :0,268
9	Emir Çeltilioğlu	E	13	Ss :69 Ort :239 Vuru:111 Cv :0,288	Ss :37 Ort :178 Vuru:149 Cv :0,207
10	Tuğrul Yüksel	E	13	Ss :52 Ort :245 Vuru:109 Cv :0,212	Ss :63 Ort :236 Vuru:114 Cv :0,266
11	Zeynep Atagül	K	25	Ss :16 Ort :199 Vuru:131 Cv :0,08	Ss :33 Ort :192 Vuru:135 Cv :0,171

BÖLÜM 3

SONUÇ VE ÖNERİLER

Tasarlanan cihaz istenilen her ortamda bağımsız şekilde (bilgisayar desteğine gerek olmadan) test ve ölçüm yapabilmekte, elde ettiği verileri saklayabilmektedir. Yapılan deneylerde kişilere ait fiziksel bilgilerin ön koşul olarak belirlenebilmesine, ölçüm şeklinin ve diğer test seçeneklerinin belirlenmesine olanak sağlamaktadır. Vurular arası geçen sürelerin ölçülerek saklanması, parmağın yapacağı bir tam vuruda butondan uzaklaşma ve yaklaşma değerlerinin ölçülüp saklanmasına olanak sağlamaktadır. Bu değerleri kullanarak matematiksel formülüzasyonlar ile belirli değerlere ulaşmaya olanak sağlamaktadır. Bu değerleri kendi üzerinde bulunan lcd ekrana yazdırmakta ve istenilmesi durumunda verileri bilgisayara transfer edip grafiksel analiz yapabilmeye ve ölçüm datalarının depolanabilmesine olanak tanımaktadır. Kendi içerisinde bulunan şarj edilebilir güç kaynağı ile 200 den fazla ölçüm yapabilmektedir.

Tasarlanan cihazın seri eepromunun kapasitesi arttırılarak test kişi sayısı artırılabilir. Boyutları daha minimize edilerek kolay taşınabilirliği artırılabilir. Mevcutda kullanılan 2x16 lcd ekran 4x16 ya da grafik lcd kullanılarak görsel kolaylık sağlanabilir. Matlab programı yerine kullanıcı arayüzü olan bir exe. program tasarlanabilir.

KAYNAKLAR

- Aoki T, Furuya S and Kinoshita H** (2005) Finger-Tapping Ability in Male and Female Pianists and Nonmusician Controls. *Motor Control*, 9: 23-39
- Bard C, Paillard J, Teasdale N, Fleury M and Lajoie Y** (1991) 'Self-induced versus reactive triggering of synchronous hand and heel movements in young and old subjects'. In: *J. Requin and G.E. Stelmach (eds.), Tutorials in motor neuroscience. Dordrecht: Kluwer Academic Publishers*, 189-196.
- Bard C, Paillard J, Lajoie Y, Fleury M, Teasdale N, Forget R and Lamarre Y** (1992) Role of afferent information in the timing of motor commands: A comparative study with a deafferented patient. *Neuropsychologia* 30: 201-206.
- Billon M, Bard C, Fleury M, Blouin J and Teasdale N** (1996) Simultaneity of two effectors in synchronization with a periodic external signal. *Human Movement Science*, 15: 25-38
- Bizzi E, Kalil R E and Morasso P** (1972) Two modes of active eye-head coordination in monkeys. *Brain Research* 40: 45-48.
- Bornstein R A, Paniak C and O'Brien W** (1987) Preliminary data on classification of normal and braindamaged elderly subjects. *The Clinical Neuropsychologist*, 1: 315-323.
- Christianson M K and Leathem J M** (2004) Development and Standardisation of the Computerised Finger Tapping Test: Comparison with other finger tapping instruments. *New Zealand Journal of Psychology*, Vol.33.
- Dash M and Telles S** (1999) Yoga training and motor speed based on a finger tapping task. *Indian J Physiol Pharmacol*, 43: 458-462.
- Fraisse P** (1980) Les synchronisations sensori-motrices aux rythmes. In: *J. Requin (ed.), Anticipation et comportement. Paris: CNRS*.
- Fraisse P, Oleron G and Paillard J** (1958) Sur les repères sensoriels qui permettent de contrôler les mouvements d'accompagnement de stimuli périodiques. *Année Psychologique*, 58: 321-338.
- Giovannoni G, van Schalkwyk J, Fritz VU and Lees AJ** (1999) Bradykinesia akinesia in co-ordination test (BRAIN TEST): an objective computerised assessment of upper limb motor function. *J Neurol Neurosurg Psychiatry*, 67: 624-629.

KAYNAKLAR (devam ediyor)

- Goldberg G** (1985) Response and projection: a reinterpretation of the premotor concept . In: E.A. Roy (ed.), *Neuropsychological studies of apraxia and related disorders*. North-Holland: *Elsevier*, 251-266.
- Heller A, Wade D T, Wood V A, Sunderland A, Hewer R L and Ward E** (1987) Arm function after stroke: measurement and recovery over the first three months. *Journal of Neurology, Neurosurgery, and Psychiatry*, 50: 714-719
- Hollingworth LS** (1914) Functional Periodicity. In: *Classics in the History of Psychology*, <http://psychclassics.yorku.ca/Hollingworth/Periodicity/chap3.htm>. (25.01.2009).
- Horton AM** (1999) Above-average intelligence and neuropsychological test score performance. *Int J Neurosci*, 99: 221–31.
- Jobbágy Á, Harcos P, Karoly R and Fazekas G** (2005) Analysis of finger-tapping movement, *Journal of Neuroscience Methods*. 141: 29-39
- Jobbágy Á, Furnée EH, Harcos P, Tárczy M, Krekule I and Komjáthi L** (1997) Analysis of movement patterns aids the early detection of Parkinson's disease. In: *19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 30 October–2 November. Chicago, IL, USA , 1760–1763.
- Kamakura N, Matsuo M, Ishii H, Mitsuboshi F and Miura Y** (1980) Patterns of static prehension in normal hands. *American Journal of Occupational Therapy*, 34, 437-445.
- Kaya H** (2007) Parmak vuru performansı ölçüm sistemi ZKÜ.
- Kızıltan E, Barut C and Gelir E** (2006) A high-precision, low cost system for evaluating fingertapping tasks, *Int J Neurosci*. 116: 1471-1480.
- Kinoshita H, Murase T and Bandou T** (1996) Grip posture and forces during holding cylindrical objects with circular grips. *Ergonomics*, 39: 1163-1176.
- Leckliter I N and Matarazzo J D** (1989) The influence of age, education, IQ, gender, and alcohol abuse on Halstead-Reitan neuropsychological test battery performance. *Journal of Clinical Psychology*, 45: 484-512.
- Mitrushina M N, Boone K B and D'Elia L F** (1999) *Handbook of Normative Data for Neuropsychological Assessment*. New York: Oxford University Press.
- Muir SR, Jones RD, Andreae JH and Donaldson IM** (1995) Measurement and analysis of single and multiple finger tapping in normal and Parkinsonian subjects. *Parkinsonism Relat Disord*, 1: 89–96.
- Nicholson K G and Kimura D** (1996) Sex differences for speech and manual skill. *Perceptual and Motor Skills*, 82, (1): 313.

KAYNAKLAR (devam ediyor)

- Notermans N C, van Dijk G W, van der Graaf Y, van Gijn J and Wokke J H** (1994) Measuring ataxia: quantification based on the standard neurological examination. *Journal of Neurology, Neurosurgery, and Psychiatry*, 57: 22-26
- Ott BR, Ellias SA and Lannon MC** (1995) Quantitative assessment of movement in Alzheimer's disease, *J Geriatr Psychiatry Neurol*, 8: 71-75.
- Özer M, Özdemir T and Gümüş M** (2008) A Computer Software for An Efficient Finger-Tapping Test, *BIOSIGNAL* 2008, 19: 32.
- Özdemir T** (2009) Parmak Vuru Performansı Ölçüm Sistemi, Zonguldak Karaelmas Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Anabilim Dalı, Yüksek Lisans Tezi (yayımlanmamış) s.1-80
- Paillard J** (1948) Quelques donnees psychophysiologiques relatives au declenchement de la commande motrice. *Annee Psychologique* 47-48, 28-47.
- Radwin R G, Oh S, Jensen T R and Webster J G** (1992) External finger forces in submaximal five-finger static pinch prehension *Ergonomics* 35: 275 -288
- Raj R and Marquis C** (1999) Finger dominance. *Journal of Hand Surgery*, 24: 429-430.
- Rao G, Fisch L, Srinivasan S, D'Amico F, Okada T, Eaton C, Robbins C, Simel D L and Rennie D** (2003) Does This Patient Have Parkinson Disease? *JAMA*, 289: 347-353.
- Romo R E, Scamati and Schultz W** (1992) Role of primate basal ganglia and frontal cortex in the internal generation of movements. II. Movement-related activity in the anterior striatum. *Experimental Brain Research* 91: 385-395.
- Ruff and Parker S B** (1993) Gender and age-specific changes in motor speed and eye-hand co-ordination in adults: Normative values for the Finger Tapping and Grooved Pegboard tests. *Perceptual and Motor Skills*, 76, (3): 1219-1230.
- Russell E W, Neuringer C and Goldstein G** (1970) *Assessment of brain damage: A neuropsychological key approach*. New York: Wiley-Interscience.
- Spreen O and Strauss E** (1998) *A compendium of neuropsychological tests*. (2nd ed.) New York: Oxford University Press.
- URL-1** (2009) <http://www.elektrotekno.com/about11860.html>, (01.06.2009).
- URL-2** (2009) <http://www.kdrkkd.googlepages.com/SENSRLER.doc>, (01.06.2009)
- URL-3** (2009) <http://www.superrobotica.com/download/sharp/gp2d120.pdf>, (01.06.2009)
- URL-4** (2009) <http://www.elk.itu.edu.tr/~deniz/mylibrary/circuits/ecir.html>, (01.06.2009)
- URL-5** (2009) <http://antrak.org.tr/index.php?id=481&option=comcontent&task=view>, (01.06. 2009).

KAYNAKLAR (devam ediyor)

URL-6 (2009) http://web.media.mit.edu/~earroyo/voice_chip/pic18F4x2.pdf, (01.06.2009)

URL-7 (2009) <http://www.modtronix.com/images/lcd162b-yhy.jpg>, (01.06.2009).

Volkow ND, Gur RC, Wang GJ, Fowler JS, Moberg PJ, Ding YS and Hitzemann R (1998) Association between decline in brain dopamine activity with age and cognitive and motor impairment in healthy individuals. *Am J Psychiatry*, 155: 344–349.

Welch LW, Cunningham AT, Eckardt MJ and Martin PR (1997) Fine motor speed deficits in alcoholic Korsakoff's syndrome. *Alcohol Clin Exp Res*, 21: 134–139.

Zatsiorsky V M, Li Z M and Latash M L (1998) Coordinated force production in multi finger tasks: Finger interaction and neural network modeling. *Biological Cybernetics*, 79: 139-150.

Zatsiorsky V M, Li Z M and Latash M L (2000) Enslaving effects in multi-finger force Production. *Experimental Brain Research*, 131: 187-195.

**EK AÇIKLAMALAR A
YAZILIMLAR**

Pic yazılımı

```
#if defined(__PCH__)
#include <18F452.h>

#fuses XT,NOWDT,NOPROTECT,NOLVP
#use delay(clock=4000000)

#use i2c(master,sda=pin_c4,scl=pin_c3)
#use rs232(baud=1200, xmit=PIN_C6, rcv=PIN_C7)
#ROM int8 0xF00000 = {0,0,0,0,0,0}
#endif
/*
float ort,ss;

for(i=0;i<8;i++)
{ort+=x[i];}
ort/=8;

for(i=0;i<8;i++)
{
ss=sqrt(pow(x[i]-ort)/8);

}
*/
#include <STDLIB.H>
#include <math.h>
#include <lcd.c>

#include <2432.c>
#include <DS1302.C>
unsigned long int mSeconds=0,mSeconds2,sagVuruSayisi=0,solVuruSayisi=0,mSecondsToplam=0,ll=0;
unsigned long int maksimumYukseklk=255,hassasSayac=0,yukseklk=0,hassasVuru=0;
unsigned long int mSecondSol=0,mSecondSag=0;
int32 ss,ort;
long int sayici=0,k,kk;
long int rnd,sure1=0, sure2=0,say=0;
float gegen_sure=0.0,ssapma=0.0,cv=0.0;
int value,vurusSayici=0;
int hafiza=1,kisiSayisi=0;
int menuFlag=0,secenekFlag=0,vuru=0,islemFlag=0,sonucFlag=0;
```

```

int testSekli=0,testSuresi=0,zamanFlag=0,tempSec=0;
byte day,mth,year,dow,hr,min,sec,hour,testBaslangic,testBitis;
byte veri,adres;
char onay,kont;

//-----timer0

#include <avr/interrupt.h>
void clock_isr() {

if(testSekli==2)
{ ++mSeconds;
if(mSeconds==50)
{
write_ext_eeprom((hafiza*2000+hassasSayac+10),(BYTE)((yukseklk&0xFF00)>>8));
write_ext_eeprom((hafiza*2000+hassasSayac+11),(BYTE)(yukseklk&0X00FF));

hassasSayac+=2;;
mSeconds=0;
}if(mSeceond==50)
}if(tetSekli==2)
if(testSekli==0){++mSeconds;}
if(testSekli==1){++mSeconds;}
}
//-----timer0 son
//-----timer2
#include <avr/interrupt.h>
void clock_isr2() {
++mSeconds2;
//-----timer2 son
void zamanlayici() {
setup_timer_2(T2_DIV_BY_4,255, 1);
set_timer2(0);
setup_counters( RTCC_INTERNAL, RTCC_DIV_4 | RTCC_8_BIT);
set_timer0(0);
enable_interrupts(INT_RTCC);
enable_interrupts(GLOBAL);
}
void isik() {
vurusSayici=0;

```

```

rnd=rand(); kk=rnd/100;
  for (k=0;k<=kk;k++)
  {

    delay_ms(50);
  }

  output_high(PIN_e2);
  delay_ms(300);
  output_low(PIN_e2);
}

void ses() {
sayici=0;
rnd=rand(); kk=rnd/100;
  for (k=0;k<=kk;k++)
  {

    delay_ms(50);
  }

  output_high(PIN_b3);
  delay_ms(300);
  output_low(PIN_b3);
}

void sesIsik() {
  output_high(PIN_b3);
  output_high(PIN_e2);
  delay_ms(100);
  output_low(PIN_b3);
  output_low(PIN_e2);
}

//-----rbo
#INT_EXT
void ext_isr0() {
  menuFlag=1;
  printf(LCD_PUTC,"\f->1)Test'e Basla\n2)Sonuc Al");
  secenekFlag=2;
}

//-----rb0 son
//-----rb1
#INT_EXT1
void ext_isr1() {
  write_ext_eeprom((hafiza*2000+sagVuruSayisi+20),(BYTE)((mSeconds&0xFF00)>>8));

```

```

write_ext_eeprom((hafiza*2000+sagVuruSayisi+21),(BYTE)(mSeconds&0X00FF));
sagVuruSayisi+=2;
mSeconds=0;
}
//-----rb1 son
//-----rb2
#INT_EXT2
void ext_isr2() {
write_ext_eeprom((hafiza*2000+1000+solVuruSayisi+20),(BYTE)((mSeconds2&0xFF00)>>8));
write_ext_eeprom((hafiza*2000+1000+solVuruSayisi+21),(BYTE)(mSeconds2&0X00FF));
solVuruSayisi+=2;
mSeconds2=0;
}
//-----rb2 son
#define LOWTOHIGH TRUE
#define HIGHTOLOW FALSE
short int dbutton4, dbutton5, dbutton6, dbutton7;
#int_rb
void detect_rb_change() {
int current;
static int last=0;
set_tris_b(0xF1);
current=input_b();
#if LOWTOHIGH
if ((!bit_test(last,4))&&(bit_test(current,4))) { write_eeprom(0x00, 0 );
hafiza=read_eeprom(0x00);printf(lcd_putc,"\frb1--- %d",hafiza); delay_ms(500);}
if ((!bit_test(last,5))&&(bit_test(current,5)))
{ hafiza=read_eeprom(0x00);printf(lcd_putc,"\frb2---- %d",hafiza);delay_ms(500); }
if ((!bit_test(last,6))&&(bit_test(current,6)))
{printf(LCD_PUTC,"%f%d-%d-%d-%d-%d",read_ext_eeprom(0),read_ext_eeprom(1),r
ead_ext_eeprom(2),read_ext_eeprom(3),read_ext_eeprom(4)); delay_ms(5000);}
if ((!bit_test(last,7))&&(bit_test(current,7)))
{printf(LCD_PUTC,"%f%d-%d-%d-%d-%d",read_ext_eeprom(100),read_ext_eeprom(101),
read_ext_eeprom(102),read_ext_eeprom(103),read_ext_eeprom(104)); delay_ms(5000);}
#elif HIGHTOLOW
if ((!bit_test(current,4))&&(bit_test(last,4))) { }
if ((!bit_test(current,5))&&(bit_test(last,5))) { dbutton5=1;}
if ((!bit_test(current,6))&&(bit_test(last,6))) { dbutton6=1;}
if ((!bit_test(current,7))&&(bit_test(last,7))) { dbutton7=1;}
#endif
}

```

```

    last=current;
}

void clear_delta() {
    dbutton4=0;
    dbutton5=0;
    dbutton6=0;
    dbutton7=0;
}

##### MAIN FONKSIYONU #####

void main() {
    set_tris_a (0xff);
    set_tris_b (0xf7);
    set_tris_e (0xf3);
    clear_delta();
    setup_adc_ports( RA0_RA1_RA3_ANALOG );
    // setup_adc( ADC_CLOCK_INTERNAL );
    setup_adc( ADC_CLOCK_DIV_32);
    set_adc_channel( 0 );
    lcd_init();
    zamanlayici();
output_low(PIN_e2);
    output_low(PIN_b3);
    enable_interrupts(INT_EXT);
    enable_interrupts(INT_EXT1);
    enable_interrupts(INT_EXT2);
    enable_interrupts(INT_RB);
    enable_interrupts(INT_RTCC);
    enable_interrupts(INT_TIMER2);
    enable_interrupts(GLOBAL);
    menuFlag=11;
    islemFlag=3;
    init_ext_eeprom();
    rtc_init();
    rtc_set_datetime(25,11,06,05,19,28);
    while (TRUE) {
        rtc_get_date( day, mth, year, dow);
        rtc_get_time( hour, min, sec );
        if(menuFlag==0){

```

```

if( read_ext_eeprom( (hafiza*2000+4) ) == 1 ) { isik();}
    if( read_ext_eeprom( (hafiza*2000+4) ) == 0 ) { ses();}
////*****HASSAS-----*****
if(testSekli==2){ hassasVuru=0;hassasSayac=0; //disable_interrupts(INT_EXT1);
    disable_interrupts(INT_EXT2);
    while ( testSuresi !=0 ) {
yukseklk = read_adc();

        rtc_get_time( hour, min, sec );
        if(sec!=tempSec){ testSuresi--; }
        tempSec=sec;
    }// while ( testSuresi !=0 )
disable_interrupts(INT_RTCC);
    disable_interrupts(INT_TIMER2);
write_ext_eeprom((hafiza*2000+8),(BYTE)((hassasSayac&0xFF00)>>8));
write_ext_eeprom((hafiza*2000+9),(BYTE)(hassasSayac&0X00FF));
sesIsik();delay_ms(300);sesIsik();delay_ms(300);sesIsik();
printf(LCD_PUTC,"\fornek sayisi %lu",hassasSayac/2);delay_ms(5000);
    menuFlag=10;
}
////*****HASSAS----SON-----*****
////*****TEK PARMAK-----*****
if(testSekli==0){
    disable_interrupts(INT_EXT2);
    while ( testSuresi !=0 ) {
        rtc_get_time( hour, min, sec );
        if(sec!=tempSec){ testSuresi--; }
        tempSec=sec;
    }// while ( testSuresi !=0 )

disable_interrupts(INT_RTCC);
    disable_interrupts(INT_TIMER2);
    disable_interrupts(INT_EXT1);
    disable_interrupts(INT_EXT2);
write_ext_eeprom((hafiza*2000+10),(BYTE)((sagVuruSayisi&0xFF00)>>8));
write_ext_eeprom((hafiza*2000+11),(BYTE)(sagVuruSayisi&0X00FF));
sesIsik();delay_ms(300);sesIsik();delay_ms(300);sesIsik();
    menuFlag=10;

```

```

// STANDART SAPMA *****
sayici=0;
ort=0;
for(ll=0;ll<sagVuruSayisi;)
{
mSeconds=0;
mSeconds=read_ext_eeprom(hafiza*2000+ll+20);
mSeconds<=<=8;
mSeconds|=read_ext_eeprom(hafiza*2000+ll+21);
ll+=2;
sayici++;
ort+=mSeconds;
    }//for(ll=0;ll<sagVuruSayisi;)--ORTALAMA--
    ort=ort/(sayici);write_ext_eeprom(hafiza*2000+14,ort);
//printf(LCD_PUTC,"fORTALAMA:%lu",ort );delay_ms(1000);
ss=0;ll=0;
for(ll=0;ll<sagVuruSayisi;)
{
mSeconds=0;
mSeconds=read_ext_eeprom(hafiza*2000+ll+20);
mSeconds<=<=8;
mSeconds|=read_ext_eeprom(hafiza*2000+ll+21);
//printf("sag\nrgs:%lu ++ y:%u",mSeconds,read_ext_eeprom(hafiza*1000+ll+12) );
ll+=2;
if(mSeconds>ort)
{ssapma=mSeconds-ort;}
if(mSeconds<ort)
{ssapma=ort-mSeconds;}
ss=ss+(ssapma*ssapma)/(sayici-1);
    }//for(ll=0;ll<sagVuruSayisi;)--STANDART SAPMA--
ss=sqrt((float)ss);write_ext_eeprom(hafiza*2000+15,ss);
printf(LCD_PUTC,"fSSAPMA: %ld\nORTALAMA: %ld",ss,ort );delay_ms(5000);
printf(LCD_PUTC,"fSAG VURU: %ld",sagVuruSayisi/2);delay_ms(5000);
// STANDART SAPMA ***** BİTİŞ
cv=(float)ss/(float)ort;
printf(LCD_PUTC,"fCV: %f",cv);delay_ms(5000);
write_ext_eeprom(hafiza*2000+16,cv*100);
sagVuruSayisi=0;solVuruSayisi=0;
}
////*****TEK PARMAK----SON-----*****

```

```

////*****ÇİFT PARMAK*****
if(testSekli==1){mSeconds2=0;mSeconds=0;
    while ( testSuresi !=0 ) {
        rtc_get_time( hour, min, sec );
        if(sec!=tempSec){testSuresi--;}
        tempSec=sec;
    }// while ( testSuresi !=0 )
disable_interrupts(INT_RTCC);
    disable_interrupts(INT_TIMER2);
    disable_interrupts(INT_EXT1);
    disable_interrupts(INT_EXT2);
write_ext_eeprom((hafiza*2000+10),(BYTE)((sagVuruSayisi&0xFF00)>>8));
write_ext_eeprom((hafiza*2000+11),(BYTE)(sagVuruSayisi&0X00FF));
write_ext_eeprom((hafiza*2000+12),(BYTE)((solVuruSayisi&0xFF00)>>8));
write_ext_eeprom((hafiza*2000+13),(BYTE)(solVuruSayisi&0X00FF));
sesIsik();delay_ms(300);sesIsik();delay_ms(300);sesIsik();
    menuFlag=10;
// STANDART SAPMA SAĞ*****
sayici=0;
ort=0;
for(ll=0;ll<sagVuruSayisi;)
{
mSeconds=0;
mSeconds=read_ext_eeprom(hafiza*2000+ll+20);
mSeconds<=<=8;
mSeconds|=read_ext_eeprom(hafiza*2000+ll+21);
ll+=2;
sayici++;
ort+=mSeconds;
    }//for(ll=0;ll<sagVuruSayisi;)--ORTALAMA--

    ort=ort/(sayici);write_ext_eeprom(hafiza*2000+14,ort);

//printf(LCD_PUTC,"fORTALAMA:%lu",ort );delay_ms(1000);
ss=0;ll=0;

for(ll=0;ll<sagVuruSayisi;)
{
mSeconds=0;
mSeconds=read_ext_eeprom(hafiza*2000+ll+20);

```

```

mSeconds<=8;
mSeconds|=read_ext_eeprom(hafiza*2000+11+21);
//printf("sag\nrgs:%lu ++ y:%u",mSeconds,read_ext_eeprom(hafiza*1000+11+12) );
11+=2;

if(mSeconds>ort)
{ssapma=mSeconds-ort;}
if(mSeconds<ort)
{ssapma=ort-mSeconds;}
ss=ss+(ssapma*ssapma)/(sayici-1);
    }//for(11=0;11<sagVuruSayisi;)--STANDART SAPMA--
ss=sqrt((float)ss);write_ext_eeprom(hafiza*2000+15,ss);
//printf(LCD_PUTC,"f sspma:%ld ",ss );delay_ms(2000);
printf(LCD_PUTC,"fSSAPMA: %ld\nORTALAMA: %ld",ss,ort );delay_ms(5000);
printf(LCD_PUTC,"fSAG VURU: %ld",sagVuruSayisi /2);delay_ms(5000);
cv=(float)ss/(float)ort;
printf(LCD_PUTC,"fCV: %f",cv);delay_ms(2000);
write_ext_eeprom(hafiza*2000+16,cv*100);
// STANDART SAPMA SAĞ***** BİTİŞ
// STANDART SAPMA SOL*****
sayici=0;
ort=0;
cv=0;
for(11=0;11<solVuruSayisi;)
{
mSeconds2=0;
mSeconds2|=read_ext_eeprom(hafiza*2000+11+1000+20);
mSeconds2<=8;
mSeconds2|=read_ext_eeprom(hafiza*2000+11+1000+21);
11+=2;
sayici++;
ort+=mSeconds2;
    }//for(11=0;11<solVuruSayisi;)--ORTALAMA--
    ort=ort/(sayici);write_ext_eeprom(hafiza*2000+17,ort);
//printf(LCD_PUTC,"fORTALAMA:%lu",ort );delay_ms(1000);
ss=0;11=0;
for(11=0;11<solVuruSayisi;)
{
mSeconds2=0;
mSeconds2|=read_ext_eeprom(hafiza*2000+11+1000+20);

```

```

mSeconds2<=<=8;
mSeconds2|=read_ext_eeprom(hafiza*2000+11+1000+21);
11+=2;
if(mSeconds2>ort)
{ssapma=mSeconds2-ort;}
if(mSeconds2<ort)
{ssapma=ort-mSeconds2;}
ss=ss+(ssapma*ssapma)/(sayici-1);
    }//for(11=0;11<solVuruSayisi;)--STANDART SAPMA--
ss=sqrt((float)ss);write_ext_eeprom(hafiza*2000+18,ss);
printf(LCD_PUTC,"\fSSAPMA: %ld\nORTALAMA: %ld",ss,ort );delay_ms(5000);
printf(LCD_PUTC,"\fSOL VURU: %ld",solVuruSayisi/2 );delay_ms(5000);
// STANDART SAPMA SOL***** BİTİŞ
cv=(float)ss/(float)ort;
printf(LCD_PUTC,"\fCV: %f",cv);delay_ms(2000);
write_ext_eeprom(hafiza*2000+19,cv*100);
sagVuruSayisi=0;solVuruSayisi=0;
}
////*****ÇİFT PARMAK---SON-----*****
    }//if(menuFlag==0)
    if(menuFlag==1){
        printf(LCD_PUTC,"\f->1)Test'e Basla\n2)Sonuc Al");islemFlag=1;;
        while( menuFlag==1 )
        {
            if(input(PIN_e0))
            {
                if(secenekFlag==2) {printf(LCD_PUTC,"\f 1)Test'e Basla\n->2)Sonuc
Al");delay_ms(300);islemFlag=0;}
                if(secenekFlag==1) {printf(LCD_PUTC,"\f->1)Test'e Basla\n2)Sonuc Al");
delay_ms(300);islemFlag=1;}
                if(secenekFlag==2){secenekFlag=1;}else{secenekFlag=2;}
            }
            if(input(PIN_e1))
            { menuFlag=2; }
        }//while
    }//if(menuFlag==1)
if(islemFlag==1)
{
    if(menuFlag==2){
        secenekFlag=2;    printf(LCD_PUTC,"\fCinsiyet\n-

```

```

>K");hafiza=read_eeprom(0x00);write_ext_eeprom
((hafiza*2000),1);delay_ms(200);
    while( menuFlag==2 )
    {
        if(input(PIN_e0))
        {
            if(secenekFlag==2) {printf(LCD_PUTC,"\fCinsiyet\n->K");delay_ms(300);write_ext_eeprom
((hafiza*2000),1);}
            if(secenekFlag==1) {printf(LCD_PUTC,"\fCinsiyet\n->E"); delay_ms(300);write_ext_eeprom
( (hafiza*2000),2);}
            if(secenekFlag==2){secenekFlag=1;}else{secenekFlag=2;}
        }
        if(input(PIN_e1))
        { menuFlag=3; }
    }//while
    }//if(menuFlag==2)
    if(menuFlag==3){
        secenekFlag=1;   printf(LCD_PUTC,"\fYas Grubu\n->0-15");write_ext_eeprom
((hafiza*2000+1),0);delay_ms(200);
        while( menuFlag==3 )
        {
            if(input(PIN_e0))
            {
                if(secenekFlag==1) {printf(LCD_PUTC,"\fYas Grubu\n->0-15");write_ext_eeprom
((hafiza*2000+1),1);delay_ms(200);}
                if(secenekFlag==2) {printf(LCD_PUTC,"\fYas Grubu\n->15-20");write_ext_eeprom
((hafiza*2000+1),2);delay_ms(200);}
                if(secenekFlag==3) {printf(LCD_PUTC,"\fYas Grubu\n->20-25");write_ext_eeprom
((hafiza*2000+1),3);delay_ms(200);}
                if(secenekFlag==4) {printf(LCD_PUTC,"\fYas Grubu\n->25-30");write_ext_eeprom
((hafiza*2000+1),4);delay_ms(200);}
                if(secenekFlag==5) {printf(LCD_PUTC,"\fYas Grubu\n->30-35");write_ext_eeprom
((hafiza*2000+1),5);delay_ms(200);}
                if(secenekFlag==6) {printf(LCD_PUTC,"\fYas Grubu\n->35-40"); write_ext_eeprom
((hafiza*2000+1),6);delay_ms(200);}
                if(secenekFlag==7) {printf(LCD_PUTC,"\fYas Grubu\n->40-45"); write_ext_eeprom
((hafiza*2000+1),7);delay_ms(200);}
                if(secenekFlag==8) {printf(LCD_PUTC,"\fYas Grubu\n->45-50"); write_ext_eeprom
((hafiza*2000+1),8);delay_ms(200);}
                if(secenekFlag==9) {printf(LCD_PUTC,"\fYas Grubu\n->50-55"); write_ext_eeprom

```

```

((hafiza*2000+1),9);delay_ms(200);}

    if(secenekFlag==10) {printf(LCD_PUTC,"\fYas Grubu\n->55-60"); write_ext_eeprom
(hafiza*2000+1),10);delay_ms(200);}

    if(secenekFlag==11) {printf(LCD_PUTC,"\fYas Grubu\n->60-65"); write_ext_eeprom
((hafiza*2000+1),11);delay_ms(200);}

    if(secenekFlag==12) {printf(LCD_PUTC,"\fYas Grubu\n->65-70"); write_ext_eeprom
((hafiza*2000+1),12);delay_ms(200);}

    if(secenekFlag==10){ secenekFlag=1;}else{ secenekFlag++;}
    delay_ms(500);
}
if(input(PIN_e1))
{ menuFlag=4; }
} //while
} //if(menuFlag==3)
if(menuFlag==4){
    secenekFlag=1;    printf(LCD_PUTC,"\fYapilacak Test \nSeklini Seciniz");write_ext_eeprom
((hafiza*2000+2),0);delay_ms(200);
    while( menuFlag==4 )
    {
        if(input(PIN_e0))
        {
            if(secenekFlag==1) {printf(LCD_PUTC,"\fTek Parmak");delay_ms(200);write_ext_eeprom
((hafiza*2000+2),0);testSekli=0;}

            if(secenekFlag==2) {printf(LCD_PUTC,"\fCift Parmak"); delay_ms(200);write_ext_eeprom
((hafiza*2000+2),1);testSekli=1;}

            if(secenekFlag==3) {printf(LCD_PUTC,"\fHassas Olcum"); delay_ms(200);write_ext_eeprom
((hafiza*2000+2),2);testSekli=2;}

            if(secenekFlag==3){ secenekFlag=1;}else{ secenekFlag++;}
            delay_ms(200);
        }
        if(input(PIN_e1))
        { menuFlag=5; }
    } //while
} //if(menuFlag==4)
if(menuFlag==5){
    secenekFlag=1;    printf(LCD_PUTC,"\fYapilacak Test \nParmagini Seciniz");write_ext_eeprom
((hafiza*2000+3),0);delay_ms(200);
    while( menuFlag==5 )
    {
        if(input(PIN_e0))

```

```

{
    if(secenekFlag==1){printf(LCD_PUTC,"\fSag
El");delay_ms(200);write_ext_eeprom((hafiza*2000+3),0);}
    if(secenekFlag==2){printf(LCD_PUTC,"\fSol El");
delay_ms(200);write_ext_eeprom((hafiza*2000+3),1);}
    if(secenekFlag==2){secenekFlag=1;}else{secenekFlag++;}
    delay_ms(200);
}
if(input(PIN_e1))
{ menuFlag=6; }
} //while
} //if(menuFlag==5)
if(menuFlag==6){
    secenekFlag=1;    printf(LCD_PUTC,"\fUyari Tipini\nBelirleyiniz");write_ext_eeprom
((hafiza*2000+4),0);delay_ms(200);
    while( menuFlag==6 )
    {
        if(input(PIN_e0))
        {
if(secenekFlag==1){printf(LCD_PUTC,"\fSesli");delay_ms(200);write_ext_eeprom((hafiza*2000+4),0);}
            if(secenekFlag==2){printf(LCD_PUTC,"\fGoruntulu");
delay_ms(200);write_ext_eeprom((hafiza*2000+4),1);}
            if(secenekFlag==2){secenekFlag=1;}else{secenekFlag++;}
            delay_ms(200);
        }
        if(input(PIN_e1))

        { menuFlag=7; }
    } //while
} //if(menuFlag==6)
if(menuFlag==7){
    secenekFlag=1;    printf(LCD_PUTC,"\fUyari Suresini\nBelirleyiniz");
write_ext_eeprom((hafiza*2000+5),0);delay_ms(200);
    while( menuFlag==7 )
    {
        if(input(PIN_e0))
        {
            if(secenekFlag==1) {printf(LCD_PUTC,"\fRasgele");delay_ms(500);
write_ext_eeprom((hafiza*2000+5),0);zamanFlag=0;testSuresi=rand()+10;

```

```

    if(testSuresi > 100){testSuresi=100;}
    printf(LCD_PUTC,"%f%d",testSuresi);delay_ms(100);} // if(secenekFlag==1)
    if(secenekFlag==2) {printf(LCD_PUTC,"%fZamanli");
delay_ms(300);write_ext_eeprom((hafiza*2000+5),1);zamanFlag=1;}
    if(secenekFlag==2){ secenekFlag=1;}else{ secenekFlag++;}
    delay_ms(300);
    }
    if(input(PIN_e1))
    { menuFlag=8; }
    }//while
    }//if(menuFlag==7)
    if(menuFlag==8){
        if(zamanFlag==1)
        {
            secenekFlag=1;    printf(LCD_PUTC,"%fUyari Suresini\nBelirleyiniz");
write_ext_eeprom((hafiza*2000+6),0);delay_ms(200);

            while( menuFlag==8 )
            {
                if(input(PIN_e0))
                {
                    if(secenekFlag==1) {printf(LCD_PUTC,"%f30 saniye");delay_ms(200);
write_ext_eeprom((hafiza*2000+6),0);testSuresi=30;}
                    if(secenekFlag==2) {printf(LCD_PUTC,"%f60 saniye"); delay_ms(200);
write_ext_eeprom((hafiza*2000+6),1);testSuresi=60;}
                    if(secenekFlag==3) {printf(LCD_PUTC,"%f90 saniye"); delay_ms(200);
write_ext_eeprom((hafiza*2000+6),2);testSuresi=90;}
                    if(secenekFlag==3){ secenekFlag=1;}else{ secenekFlag++;}
                    delay_ms(200);
                }
                if(input(PIN_e1))

                { menuFlag=9; }
                }//while
            } // if(zamanFlag==1)
menuFlag=9;
        }//if(menuFlag==8)
        if(menuFlag==9){
            secenekFlag=1;    printf(LCD_PUTC,"%fBASLAT");delay_ms(50);
            while( menuFlag==9 )
            { // printf(LCD_PUTC,"%fBASLAT %d",hafiza);

```

```

printf(LCD_PUTC, "\f%d-%d-%d-%d-%d-%d-
%d", read_ext_eeprom(hafiza*2000), read_ext_eeprom(hafiza*2000+1),
read_ext_eeprom(hafiza*2000+2), read_ext_eeprom(hafiza*2000+3),
read_ext_eeprom(hafiza*2000+4), read_ext_eeprom(hafiza*2000+5), read_ext_eeprom(hafiza*2000+6));

delay_ms(300);
if(input(PIN_e1))
{
menuFlag=0; //printf(LCD_PUTC, "\fBASLADI"); delay_ms(10);
hafiza++; write_eeprom(0x00, hafiza ); hafiza--;
sagVuruSayisi=0; solVuruSayisi=0; tempSec=100;
vurusSayici=0;
rtc_get_time( hour, min, sec );
testBaslangic=sec;
mSeconds=0; hassasSayac=0; mSeconds2=0;
enable_interrupts(INT_EXT);
enable_interrupts(INT_EXT1);
enable_interrupts(INT_EXT2);
enable_interrupts(INT_RB);
enable_interrupts(INT_RTCC);
enable_interrupts(INT_TIMER2);
}
}while
} //if(menuFlag==9)
} //if(islemFlag==1)

////*****SONUÇ AL*****
//kk=read_eeprom(0x00);
//kk=kk-1;
if(islemFlag==0)
{
menuFlag=1;
secenekFlag=1;
if(menuFlag==1){
testsekli=read_ext_eeprom(kk*2000+2);
disable_interrupts(INT_RTCC);
disable_interrupts(INT_TIMER2);
disable_interrupts(INT_EXT1);
disable_interrupts(INT_EXT2);
printf(LCD_PUTC, "\f-> 1. Kisi\n 2. Kisi"); delay_ms(500); sonucFlag=0;
while( menuFlag==1 )
{

```

```

if(input(PIN_e0))
{
if(secenekFlag==1) {printf(LCD_PUTC,"f-> 1. Kisi\n 2. Kisi");delay_ms(200);sonucFlag=0;}
if(secenekFlag==2) {printf(LCD_PUTC,"f-> 2. Kisi\n 3. Kisi"); delay_ms(200);sonucFlag=1;}
if(secenekFlag==3) {printf(LCD_PUTC,"f-> 3. Kisi\n 4. Kisi"); delay_ms(200);sonucFlag=2;}
if(secenekFlag==4) {printf(LCD_PUTC,"f-> 4. Kisi\n 5. Kisi"); delay_ms(200);sonucFlag=3;}
if(secenekFlag==5){secenekFlag=1;}else{secenekFlag++;}
delay_ms(300);
}
if(input(PIN_e1))
{ menuFlag=2; }
} //while
} //if(menuFlag==1)
if(menuFlag==2)
{
kk=sonucFlag;
if( read_ext_eeprom((kk*2000+2)) ==0){testSekli=0;}
if( read_ext_eeprom((kk*2000+2)) ==1){testSekli=1;}
if( read_ext_eeprom((kk*2000+2)) ==2){testSekli=2;}
//*****HASSAS*****
if(testSekli==2){
hassasSayac=0;
hassasSayac=read_ext_eeprom(kk*2000+8);
hassasSayac<=&8;
hassasSayac|=read_ext_eeprom(kk*2000+9);
printf(LCD_PUTC,"fornek sayisi %lu",hassasSayac/2);
onay=getc();
if(onay=='y'){
printf("%ld",hassasSayac/2);
}
for(ll=0;ll<hassasSayac;)
{
yukseklk=0;
yukseklk=read_ext_eeprom(kk*2000+ll+10);
yukseklk<=&8;
yukseklk|=read_ext_eeprom(kk*2000+ll+11);
do{
onay=getc();
if((ll/2)%4==0)
{kont='a';}

```

```

if((ll/2)%4==1)
{kont='b';}
if((ll/2)%4==2)
{kont='c';}
if((ll/2)%4==3)
{kont='d';}
    if(onay==kont)
    {printf("%lu",yukseklik);
      }
    }while (onay!=kont) ;

ll+=2;
}for(ll=0;ll<sagVuruSayisi;)
}if(testSekli)
//*****HASSAS*****
//*****TEK PARMAK*****

if(testSekli==0){
sagVuruSayisi=read_ext_eeprom(kk*2000+10);
sagVuruSayisi<=8;
sagVuruSayisi|=read_ext_eeprom(kk*2000+11);
printf(LCD_PUTC,"\fVuru Sayisi %lu",sagVuruSayisi/2);
onay=getc();
    if(onay=='y'){
        printf("%ld",sagVuruSayisi/2);
    }
for(ll=0;ll<sagVuruSayisi;)
{
mSeconds=0;
maksimumYukseklik=0;
mSeconds=read_ext_eeprom(kk*2000+ll+20);
mSeconds<=8;
mSeconds|=read_ext_eeprom(kk*2000+ll+21);
    do{
onay=getc();
if((ll/2)%4==0)
{kont='a';}
if((ll/2)%4==1)
{kont='b';}
if((ll/2)%4==2)

```

```

{kont='c';}
if((ll/2)%4==3)
{kont='d';}
    if(onay==kont)
    {printf("%lu",mSeconds);
    }
    }while (onay!=kont) ;
ll+=2;
}for(ll=0;ll<sagVuruSayisi;)
}if(testSekli==0)

//*****TEK PARMAK*****
//*****ÇİFT PARMAK*****

if(testSekli==1){
sagVuruSayisi=read_ext_eeprom(kk*2000+10);
sagVuruSayisi<=8;
sagVuruSayisi|=read_ext_eeprom(kk*2000+11);
solVuruSayisi=read_ext_eeprom(kk*2000+12);
solVuruSayisi<=8;
solVuruSayisi|=read_ext_eeprom(kk*2000+13);
printf(LCD_PUTC,"%fSol %lu \nSag %lu",solVuruSayisi/2,sagVuruSayisi/2);
onay=getc();
    if(onay=='y'){
        printf("%ld",sagVuruSayisi/2);delay_ms(500);printf("%ld",solVuruSayisi/2);
    }
for(ll=0;ll<sagVuruSayisi;)
    {
mSecondSag=0;
mSecondSag=read_ext_eeprom(kk*2000+ll+20);
mSecondSag<=8;
mSecondSag|=read_ext_eeprom(kk*2000+ll+21);
        do{
onay=getc();
if((ll/2)%4==0)
{kont='a';}
if((ll/2)%4==1)
{kont='b';}
if((ll/2)%4==2)
{kont='c';}
if((ll/2)%4==3)

```

```

{kont='d';}
    if(onay==kont)
        {printf("%lu",mSecondSag);
          }
    }while (onay!=kont) ;
ll+=2;
}for(ll=0;ll<sagVuruSayisi;)
for(ll=0;ll<solVuruSayisi;)
{
mSecondSol=0;
mSecondSol=read_ext_eeprom(kk*2000+ll+1000+20);
mSecondSol<=8;
mSecondSol=read_ext_eeprom(kk*2000+ll+1000+21);
    do{
onay=getc();
if((ll/2)%4==0)
{kont='a';}
if((ll/2)%4==1)
{kont='b';}
if((ll/2)%4==2)
{kont='c';}
if((ll/2)%4==3)
{kont='d';}
    if(onay==kont)
        { printf("%lu",mSecondSol);
          }
    }while (onay!=kont) ;
ll+=2;
}for(ll=0;ll<solVuruSayisi;)
}if(testSekli==2)
//*****ÇİFT PARMAK*****
}if(menuFlag==2)
}if(islemFlag==0)
}while}

```

Matlab Programı

```
*Tek parmak için*
s = serial('COM5'); % Com portunu belirle
set(s,'BaudRate',1200); % Baud rate veri hızını belirle
set(s, 'Parity', 'none') ; % Parity bitinin olup olmayacağını belirle
set(s, 'Databits', 8) ; % Data bit sayısını belirle
set(s, 'StopBits', 1) ; % Stop bit sayısını belirle
%s.Terminator='CR/LF';
%set(s, 'OutputBufferSize', 5000) ; % Seri portla gönderilecek verinin
boyutunu belirle
%set(s, 'inputBufferSize', 5000) ; % Seri portla alınacak verinin boyutunu
belirle

s.Timeout=0.4; % Seri portun veri maksimum veri alma süresini belirler
fopen(s); % Seri portu veri alma vermeye aktif hle getirir
gon=input('gir','s'); %Pic'e veri almak için komut yollar
fprintf(s,'%s',gon)
bilgi = fscanf(s,'%d'); %Picten veri sayısı bilgisini alır

for l=1:bilgi
if mod(l,4) == 1
fprintf(s,'a')
data(l) = fscanf(s,'%d');

end
if mod(l,4) == 2
fprintf(s,'b')
data(l) = fscanf(s,'%d');

end
if mod(l,4) == 3
fprintf(s,'c')
data(l) = fscanf(s,'%d');

end
if mod(l,4) == 0
fprintf(s,'d')
data(l) = fscanf(s,'%d');
end
end %for
fclose(s); %seri portu veri alma ve vermeye kapatır
t=1:bilgi; % grafik için zaman boyutuna vuruş sayısını atar
sigma=std(data);
I_T_I=mean(data);
cv=(sigma/I_T_I);
SUBPLOT(2,1,1),

plot(1:bilgi,data,'*');grid;xlabel(['Vuru Sayısı:
',num2str(bilgi)]);ylabel('vuru süresi (ms) '); %vuruş sayısına göre
vuruşlar arasındaki geçen sürenin grafiğini çizdirir
ylim([0 1000])
title(['Tek Parmak Vuru Testi ' ' \sigma_I_T_I: ',num2str(sigma) '
_I_T_I:',num2str(I_T_I) ' CV: ',num2str(cv)])

SUBPLOT(2,1,2),

hist(data);xlabel('vuru süreleri (ms) ');ylabel('vuru adeti');
title('Tek Parmak Vuru Histogram diagramı')
```

```

*Çift parmak için*
s = serial('COM5'); % Com portunu belirle
set(s,'BaudRate',1200); % Baud rate veri hızını belirle
set(s, 'Parity', 'none') ; % Parity bitinin olup olmayacağını belirle
set(s, 'Databits', 8) ; % Data bit sayısını belirle
set(s, 'StopBits', 1) ; % Stop bit sayısını belirle
%s.Terminator='CR/LF';
%set(s, 'OutputBufferSize', 5000) ; % Seri portla gönderilecek verinin
boyutunu belirle
%set(s, 'inputBufferSize', 5000) ; % Seri portla alınacak verinin boyutunu
belirle

s.Timeout=0.4; % Seri portun veri maksimum veri alma süresini belirler
fopen(s); % Seri portu veri alma vermeye aktif hle getirir
gon=input('gir','s'); %Pic'e veri almak için komut yollar
fprintf(s,'%s',gon)
bilgi = fscanf(s,'%d'); %Picten veri sayısı bilgisini alır
bilgi2 = fscanf(s,'%d'); %Picten veri sayısı bilgisini alır


for z=1:bilgi
if mod(z,4) == 1
fprintf(s,'a')
data(z) = fscanf(s,'%d');

end
if mod(z,4) == 2
fprintf(s,'b')
data(z) = fscanf(s,'%d');

end
if mod(z,4) == 3
fprintf(s,'c')
data(z) = fscanf(s,'%d');

end
if mod(z,4) == 0
fprintf(s,'d')
data(z) = fscanf(s,'%d');

end
end %for


for m=1:bilgi2
if mod(m,4) == 1
fprintf(s,'a')
data2(m) = fscanf(s,'%d');

end
if mod(m,4) == 2
fprintf(s,'b')
data2(m) = fscanf(s,'%d');

end
if mod(m,4) == 3
fprintf(s,'c')

```

```

data2(m) = fscanf(s,'%d');

end
if mod(m,4) == 0
fprintf(s,'d')
data2(m) = fscanf(s,'%d');

end
end %for

fclose(s); %seri portu veri alma ve vermeye kapatır
t=1:bilgi; % grafik için zaman boyutuna vuruş sayısını atar

sigma=std(data);
I_T_I=mean(data);
cv=(sigma/I_T_I);
sigma1=std(data2);
I_T_I1=mean(data2);
cv1=(sigma1/I_T_I1);

SUBPLOT(4,1,1),

plot(1:bilgi,data,'*');grid;xlabel(['Vuru Sayısı:
',num2str(bilgi)]);ylabel('vuru süresi ms '); %vuruş sayısına göre
vuruşlar arasındaki geçen sürenin grafiğini çizdirir
ylim([0 1000])
title(['Çift Parmak Vuru Testi Sol Vuru' ' \sigma_I_T_I: ',num2str(sigma)
' _I_T_I:',num2str(I_T_I) ' CV: ',num2str(cv)])

SUBPLOT(4,1,2),

hist(data);xlabel('vuru süreleri (ms) ');ylabel('vuru adeti');
title('Sol Vuru Histogram diagramı')

SUBPLOT(4,1,3),

plot(1:bilgi2,data2,'*');grid;xlabel(['Vuru Sayısı:
',num2str(bilgi2)]);ylabel('vuru süresi ms '); %vuruş sayısına göre
vuruşlar arasındaki geçen sürenin grafiğini çizdirir
ylim([0 1000])
title(['Çift Parmak Vuru Testi Sağ Vuru' ' \sigma_I_T_I: ',num2str(sigma1)
' _I_T_I:',num2str(I_T_I1) ' CV: ',num2str(cv1)])

SUBPLOT(4,1,4),

hist(data2);xlabel('vuru süreleri (ms) ');ylabel('vuru adeti');
title('Sağ Vuru Histogram diagramı')

```

```

*Hassas Ölçüm için*
s = serial('COM5'); % Com portunu belirle
set(s, 'BaudRate', 1200); % Baud rate veri hızını belirle
set(s, 'Parity', 'none') ; % Parity bitinin olup olmayacağını belirle
set(s, 'Databits', 8) ; % Data bit sayısını belirle
set(s, 'StopBits', 1) ; % Stop bit sayısını belirle
%s.Terminator='CR/LF';
%set(s, 'OutputBufferSize', 5000) ; % Seri portla gönderilecek verinin
boyutunu belirle
%set(s, 'InputBufferSize', 5000) ; % Seri portla alınacak verinin boyutunu
belirle

s.Timeout=0.4; % Seri portun veri maksimum veri alma süresini belirler
fopen(s); % Seri portu veri alma vermeye aktif hle getirir
gon=input('gir','s'); %Pic'e veri almak için komut yollar
fprintf(s,'%s',gon)
bilgi = fscanf(s,'%d'); %Picten veri sayısı bilgisini alır

for i=1:bilgi
if mod(i,4) == 1
fprintf(s,'a')
data(i) = fscanf(s,'%d');

end
if mod(i,4) == 2
fprintf(s,'b')
data(i) = fscanf(s,'%d');

end
if mod(i,4) == 3
fprintf(s,'c')
data(i) = fscanf(s,'%d');

end
if mod(i,4) == 0
fprintf(s,'d')
data(i) = fscanf(s,'%d');

end
end %for

%data=250-data;

data=60.6*exp(-3.5*5/255*data) + 20.04*exp(-0.6*5/255*data)-3.2;

fclose(s); %seri portu veri alma ve vermeye kapatır
t=1:bilgi; % grafik için zaman boyutuna vuruş sayısını atar
plot(1:bilgi,data,'-');grid;xlabel('Vuru Sayısı 116 ');ylabel('Vuru
Yüksekliği (cm)'); %vuruş sayısına göre vuruşlar arasındaki geçen sürenin
grafliğini çizdirir
ylim([0 5])
title('Hassas Ölçüm Testi ')

```


EK AÇIKLAMALAR B
CİHAZIN FOTOĞRAFLARI



Foto 1 Genel Görünüş



Foto 2 Yan görünüş

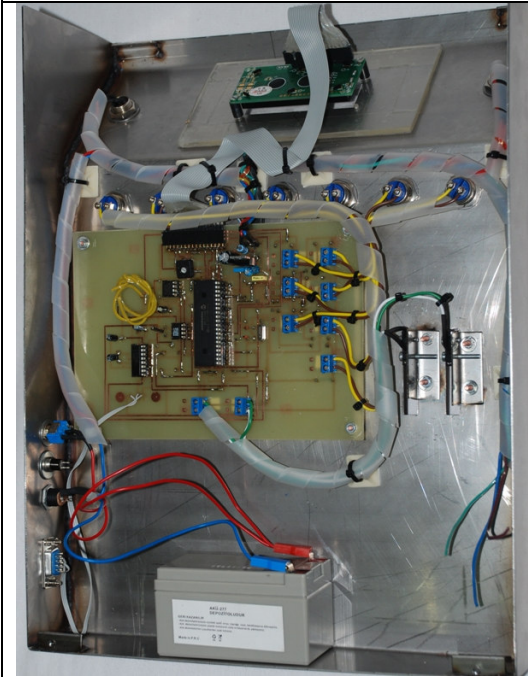


Foto 4 ayrıntılı (içyapısı) görünüş1

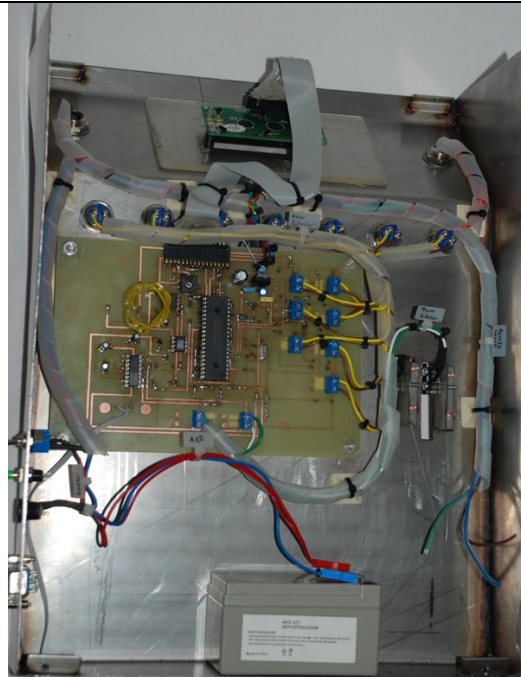


Foto 4 ayrıntılı (içyapısı) görünüş2

Şekil B.1 Cihazın fotoğrafları

ÖZGEÇMİŞ

Hüseyin YETER 1982’de İstanbulda doğdu; ilk ve orta öğretimini tamamladıktan sonra Şişli Teknik lisesinden mezun olup Marmara Üniversitesi T.B.M.Y. O Elektrik bölümünü bitirdi. Dikey Geçiş sınavı ile Zonguldak Karaelmas Üniversitesi Elektrik-Elektronik Mühendisliği Bölümüne girmiştir.2006 senesinde iyi derece ile mezun olup Sensormatic Güvenlik Sistemleri AŞ. Şirketinde Teknik Destek Koordinatörü olarak göreve başlamıştır. Halen 2006 Senesinde girmiş olduğu ZKÜ Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda yüksek lisans programını sürdürmektedir

ADRES BİLGİLERİ

Adres: Zafer Mah.Köyüm Sok.3/5 Y.Bosna
34520 Bahçelievler İSTANBUL

Tel: (0212) 552 52 99

E-posta: yeterhuseyin@yahoo.com

Hüseyin YETER