

**X IŐIN TP İÇİN  
HASSAS KALİBRATR TASARIMI VE GERÇEKLENMESİ**

**Hakan GEMİCİ**

**YKSEK LİSANS TEZİ  
ELEKTRİK-ELEKTRONİK MHENDİSLİĐİ**

**GAZİ NİVERSİTESİ  
FEN BİLİMLERİ ENSTİTS**

**NİSAN 2008**

**ANKARA**

Hakan GEMİCİ tarafından hazırlanan X IŞIN TÜPÜ İÇİN HASSAS KALİBRATÖR TASARIMI VE GERÇEKLENMESİ adlı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Prof.Dr. İrfan KARAGÖZ .....

Tez Danışmanı, Elektrik Elektronik Müh. A.B.D.

Bu çalışma, jürimiz tarafından oy birliği ile Elektrik Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Prof. Dr. M. Cengiz TAPLAMACIOĞLU .....

Elektrik Elektronik Mühendisliği, Gazi Üniversitesi

Prof. Dr. İrfan KARAGÖZ .....

Elektrik Elektronik Mühendisliği, Gazi Üniversitesi

Doç. Dr. Şeref SAĞIROĞLU .....

Bilgisayar Mühendisliği, Gazi Üniversitesi

Tarih: 28/04/2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nermin ERTAN .....

Fen Bilimleri Enstitüsü Müdürü

## **TEZ BİLDİRİMİ**

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Hakan GEMİCİ

**X IŞIN TÜPÜ İÇİN  
HASSAS KALİBRATÖR TASARIMI VE GERÇEKLENMESİ  
(Yüksek Lisans Tezi)**

**Hakan GEMİCİ**

**GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**Nisan 2008**

**ÖZET**

**Bu çalışmada tıbbi görüntüleme için kullanılan konvansiyonel röntgen, floroskopi, mamografi, dental röntgen gibi X Işını Tüpü temelli görüntüleme cihazlarında kullanılan ve gerek görüntüleme kalitesi, gerekse hasta sağlığı açısından hayati öneme haiz bazı tüp parametrelerinin güncel teknolojinin son olanaklarından faydalanılarak tasarlanıp geliştirilen bir cihaz yardımıyla çok hassas bir şekilde ve yüksek doğrulukla ölçülmesi amaçlanmıştır. Aynı zamanda cihaza, kablosuz olarak bir bilgisayara bağlanarak ölçüm sonuçlarını aktarabilmesi için Bluetooth ve gerekli değerlendirmelerin yapılabilmesi için ölçüm verilerinin grafiksel olarak gösterimini sağlayan bir bilgisayar yazılımı ile kullanımını kolaylaştıran fonksiyonlar eklenmiştir. Bu şekilde ölçümlerin alınması esnasında X Işın Tüpünü besleyen yüksek gerilim transformatörüne yakın çalışmasından doğabilecek olası risklerin önüne geçilmiş ve teknisyen sağlığının korunması amaçlanmıştır. Alınan ölçümlerle kalibrasyonu yapılan cihazlar zaman içerisinde ortaya çıkan sapmalar sonucu hastaların fazla dozaja bağlı yan etkilerden korunmasını sağlamaktadır.**

**Bilim Kodu** : 905.1.021  
**Anahtar Kelimeler** : X Işın Tüpü, Tıbbi Görüntüleme, Röntgen, Kalibrasyon, mAs metre  
**Sayfa Adedi** : 194  
**Tez Yöneticisi** : Prof. Dr. İrfan KARAGÖZ

**PRECISION CALIBRATOR DESIGN AND IMPLEMENTATION FOR  
X-RAY DIAGNOSTIC DEVICES**

**(M.Sc. Thesis)**

**Hakan GEMİCİ**

**GAZİ UNIVERSITY  
INSTITUTE OF SCIENCE AND TECHNOLOGY**

**April 2008**

**ABSTRACT**

**In this thesis it is aimed to design and develop an instrumental device which is capable of measuring some key parameters of x-ray tubes employed in many medical imaging devices including conventional x-ray imaging, fluoroscopy, mammography and dental x-ray imaging and those measured parameters which is critical for both imaging quality and patient safety aspects are acquired and evaluated extremely precise and very reliable with the help of latest technological advancements. Furthermore the designed instrumental device is supplemented with a Bluetooth connection capability which allows both remote controlling of the device and data transmission between the device and a personal computer. Also a graphical user interface which is capable of recording multiple exposure data acquired via bluetooth module and plotting comparison graphs in demand of user is also programmed in .NET environment. Remote controlling of the device and data transmission via Bluetooth avoided possible risks of working close to high voltage transformer of x-ray tube, hence gives an extra protection to medical device technicians. Taken measurements can be used to make periodic calibrations of medical imaging devices and this also helps reducing possible risks may be caused by overdosing during imaging.**

**Science Code** : 905.1.021  
**Key Words** : X-Ray Tube, Medical Imaging, X-Ray Imaging,  
Calibration, mAs meter  
**Page Number** : 194  
**Adviser** : Prof. Dr. İrfan KARAGÖZ

## TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Prof. Dr. İrfan KARAGÖZ'e, yine kıymetli tecrübelerinden faydalandığım Hocam Doç. Dr. Şeref SAĞIROĞLU'na, Ankara Üniversitesi Nükleer Bilimler Enstitüsü'ndeki Diagnostik Radyoloji Görüntüleme Laboratuvarında cihazımızı test etme imkânı sunan Enstitü Müdürü Prof.Dr. Doğan BOR'a ve aynı enstitüde görev yapmakta olan Dr. Turan OLĞAR'a, yine Ankara Üniversitesi Fizik Mühendisliği bölümünde görev yapmakta olan arkadaşım Fizik Yüksek Mühendisi Burak ÇUHADAROĞLU'na ayrıca önceki çalışmalarını ile ufkumu genişleten arkadaşım Elektrik Elektronik Yüksek Mühendisi Zafer DEDEOĞLU'na, kod geliştirme sürecindeki katkılarından ötürü arkadaşlarım Arş. Gör. Fatih GÖKÇE ve Arş. Gör. Uraz YAVANOĞLU'na ve beni manevi destekleriyle yalnız bırakmayan aileme teşekkürü bir borç bilirim.

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	iv
ABSTRACT .....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER .....	ix
ÇİZELGELERİN LİSTESİ.....	xiv
ŞEKİLLERİN LİSTESİ .....	xv
RESİMLERİN LİSTESİ .....	xvii
SİMGELER VE KISALTMALAR.....	xviii
1. GİRİŞ .....	1
2. RADYOLOJİDE TEMEL KAVRAMLAR.....	5
2.1. Radyolojinin Tanımı ve Kapsamı .....	5
2.2. X-ışınlarının ve Röntgenin Keşfi .....	5
2.3. Işın Fiziği ve Işın Spektrumu.....	6
2.4. Partiküler Radyasyonlar.....	6
2.4.1. Alfa ( $\alpha$ ) ışınları.....	7
2.4.2. Beta ( $\beta$ ) ışınları.....	8
2.4.3. Katot ışınları.....	9
2.5. Elektromanyetik Radyasyonlar .....	9
2.5.1. Kozmik ışınlar .....	11
2.5.2. Gamma ( $\gamma$ ) ışınları.....	11
2.5.3. X - ışınları .....	12
2.5.4. Ultraviyole ışınlar .....	13

2.5.5. Görünür ışık .....	13
2.5.6. Enfraruj ışınlar .....	14
3. RÖNTGEN CİHAZLARININ YAPISI VE X-IŞINLARININ OLUŞUMU .....	15
3.1. Ototransformatör .....	15
3.2. Yüksek Gerilim Transformatorü .....	15
3.3. Doğrultmaçlar .....	17
3.4. Yüksek Gerilim Kabloları .....	17
3.5. Röntgen Tüpleri .....	17
3.5.1. Klasik röntgen tüpleri .....	17
3.5.2. Modern röntgen tüpleri .....	18
3.6. X-Işınlarının Elde Edilişi .....	21
3.7. Röntgen Cihazlarında Kalibrasyon .....	23
4. X-RAY GÖRÜNTÜLEME EKİPMANLARI .....	25
4.1. X Işınlarının Üretilmesi .....	26
4.1.1. X ışın tüpü .....	26
4.1.2. Gerilim kaynağı .....	29
4.1.3. Görüntü algılayıcılar: ekran – film kombinasyonları .....	30
4.1.4. Görüntü algılayıcılar: x-ray görüntü yoğunlaştırıcıları ve televizyonlar .....	32
4.1.5. Görüntü algılama: sayısal sistemler .....	35
4.1.6. X-ray projeksiyon anjiyografisi .....	37
5. BLUETOOTH KABLOSUZ İLETİŞİM PROTOKOLÜ .....	39
5.1. Yapısı ve Çalışması .....	39
5.2. Kullanım Yerleri .....	39

5.3. Ağ Altyapı Teknolojileri Olarak Bluetooth ve Wi-Fi Karşılaştırması.....	41
5.3.1. Bluetooth.....	42
5.3.2. Wi-Fi.....	42
5.4. Bilgisayar Gereksinimleri .....	43
5.4.1. İşletim sistemi desteği.....	43
5.5. Özellikleri .....	43
5.5.1. Bluetooth 1.0 ve 1.0B.....	44
5.5.2. Bluetooth 1.1 .....	44
5.5.3. Bluetooth 1.2 .....	44
5.5.4. Bluetooth 2.0.....	45
5.5.5. Bluetooth 2.1 .....	45
5.5.6. Bluetooth'un geleceği .....	46
5.5.7. Yüksek hızlı bluetooth .....	47
6. YAZILIM MİMARİSİ.....	48
6.1. Gömülü Kod İçin Geliştirme Araçları .....	48
6.1.1. Mplab ide .....	49
6.1.2. C30 derleyici .....	51
6.1.3. EasydsPIC4 geliştirme kartı.....	52
6.2. Mikrodenetleyici kodu .....	53
6.3. Bilgisayar Kontrol Arayüzü.....	57
7. DONANIM MİMARİSİ .....	60
7.1. Blok Diyagramı.....	61
7.2. Ölçüm Prensipleri.....	62
7.3. Cihazın Elektronik Devre Blokları .....	65

7.3.1. Besleme katı .....	65
7.3.2 Anahtarlama – ön ölçekleme katı.....	66
7.3.3. Aletsel yükselteç .....	69
7.3.4. Hassas tam dalga doğrultmaç.....	70
7.3.5. Ekspojur algılama devresi.....	71
7.3.6. Mikrodenetleyici katı .....	72
7.3.7. Bluetooth iletişim katı .....	75
8. KULLANIM KILAVUZU.....	77
8.1. Bilgisayardan Kontrol Yazılımı.....	81
9. SONUÇ VE ÖNERİLER.....	85
KAYNAKLAR .....	89
EKLER.....	90
EK-1 Besleme katı devre şeması .....	91
EK-2 Opamp blokları devre şeması .....	92
EK-3 dsPIC katı devre şeması .....	93
EK-4 Kademe seçici komütatör devre şeması .....	94
EK-5 Analog kart tüm katmanlar .....	95
EK-6 Analog kart baskılı devre üst yüz.....	96
EK-7 Analog kart baskılı devre alt yüz.....	97
EK-8 Analog kart pcb 3D modellemesi .....	98
EK-9 Analog kart malzeme listesi .....	99
EK-10 dsPIC kartı tüm katmanlar.....	102
EK-11 dsPIC kartı baskılı devre üst yüz.....	103
EK-12 dsPIC kartı baskılı devre alt yüz.....	104
EK-13 dsPIC kartı pcb 3D modellemeleri .....	105
EK-14 dsPIC kartı malzeme listesi .....	106
EK-15 Bluetooth modülü devre şeması .....	108
EK-16 Bluetooth modülü bağlantı şeması .....	109
EK-17 Bluetooth modülü baskılı devre üst yüz.....	110
EK-18 Bluetooth modülü baskılı devre alt yüz.....	111
EK-19 Bluetooth modülü baskılı devre eleman yerleşimi .....	112
EK-20 dsPIC30F4011 için C kodu .....	113
EK-21 dsPIC 30F4011 için C kodunun akış diyagramı.....	140
EK-22 Bilgisayar kontrol yazılımının C# kodu .....	145
EK-23 mAs metre fotoğrafları .....	188
EK-24 Floroskopi cihazı üzerinde ölçüm alınırken .....	190

EK-25 Picker anatomic-40 x-ışını görüntüleme cihazı.....	191
EK-26 X-ışını cihazına bağlantı.....	192
EK-27 Konvansiyonel x-ışını görüntüleme cihazı üzerinde ölçüm alınırken.....	193
ÖZGEÇMİŞ .....	194

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 5.1. Bluetooth güç ve menzil sınıfları.....	39
Çizelge 5.2. Bluetooth protokol versiyonları .....	40
Çizelge 6.1. ADC tampon belleği üzerinde hesaplama yapan ve her 165 örnekte bir hesaplanan ortalamayı seri porttan gönderen kod parçası.....	56
Çizelge 7.1. Ölçüm kademelerindeki dirençlerin akım-gerilim-güç ilişkisi .....	68
Çizelge 9.1. Konvansiyonel x ışın cihazı üzerinde alınan bazı ölçüm değerleri .....	86

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Elektromanyetik tayf.....	10
Şekil 3.1. Sabit anotlu röntgen tüpü.....	19
Şekil 3.2. Yüksek gerilim devresi .....	24
Şekil 4.1. X-Ray tüpü.....	26
Şekil 4.2. X ışını kuvvetlendiricisi.....	33
Şekil 6.1. Tasarım döngüsü.....	49
Şekil 6.2. MPLAB IDE ekran görüntüsü .....	50
Şekil 6.3. MPLAB C30 ile yazılım geliştirme aşamaları.....	51
Şekil 6.4. dsPIC30F4011 dâhilî ADC mimarisi.....	55
Şekil 6.5. Önceden kaydedilen değerler arasından çağrılmış dört ölçüm sonucunun karşılaştırmalı grafiğinin bulunduğu ekran görüntüsü .....	58
Şekil 6.6. mAs metre kontrol programının genel akış diyagramı .....	59
Şekil 7.1. mAs metre donanımsal blok diyagramı .....	61
Şekil 7.2. Yüksek gerilim trafosunun primer ve sekonder kısımlarının blokları.....	63
Şekil 7.3. mAs metre besleme devresi .....	65
Şekil 7.4. Kademe seçici komütatör anahtar ve kademe dirençlerinin bağlantıları ...	67
Şekil 7.5. UF5408 diyotun ileri yönde kutuplama gerilimi ile akımının değişimi ....	68
Şekil 7.6. Üç adet opamp ile gerçekleşmiş tipik bir aletsel yükselteç.....	69
Şekil 7.7. Aletsel yükselteç devre bloğu .....	70
Şekil 7.8. Hassas tam dalga doğrultmaç devre şeması.....	71
Şekil 7.9. Ekspojur algılama devresi.....	72

<b>Şekil</b>	<b>Sayfa</b>
Şekil 7.10. Bluetooth modülü bağlantı şeması.....	75
Şekil 8.1. mAs metre kontrollerinin yerleşimi.....	77
Şekil 8.2. Ekran görüntüsü.....	78
Şekil 8.3. Ekran görüntüsü.....	78
Şekil 8.4. Ekran görüntüsü.....	79
Şekil 8.5. Ekran görüntüsü.....	79
Şekil 8.6. Ekran görüntüsü.....	80
Şekil 8.7. Ekran görüntüsü.....	80
Şekil 8.8. mAs metre kontrol modülü ekran görüntüsü.....	81
Şekil 8.9. Grafik çizildikten sonraki ekran görüntüsü.....	83
Şekil 8.10. Büyütülmüş grafik.....	84
Şekil 9.1. mAs metre'nin kalibrasyonu esnasında alınan ölçüm ekranı.....	87

**RESİMLERİN LİSTESİ**

<b>Resim</b>	<b>Sayfa</b>
Resim 3.1. Yüksek voltaj tankı ve kumanda masası.....	16
Resim 3.2. Döner anotlu bir röntgen tüpü.....	20
Resim 3.3. Spiral filamanlar .....	20
Resim 6.1. EasyDSPIC4 programlama, hata ayıklama, ürün geliştirme kartı.....	52
Resim 7.1. Tasarlanan mAs-metre'nin montajlanmış hâli.....	64
Resim 7.2. Quickblue bluetooth modülü .....	76

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

<b>Simgeler</b>	<b>Açıklamalar</b>
<b>A</b>	Amper
<b>Å</b>	Angström
<b>bps</b>	Saniyede bit miktarı
<b>CaW<sub>0</sub>2</b>	Kalsiyum Tungstat (calcium tungstate)
<b>cm</b>	Santimetre
<b>cm<sup>3</sup></b>	Santimetreküp
<b>CsI</b>	Sezyum İyodit (Caesium iodide)
<b>Gd<sub>2</sub>O<sub>2</sub>S</b>	Gadolinium Oksid Sülfid (Gadolinium oxide sulfide)
<b>gr</b>	Gram
<b>Hz</b>	Hertz
<b>Kbyte</b>	1000 bayt (kilobyte)
<b>kEV</b>	Kilo elektron volt
<b>km</b>	Kilometre
<b>kV</b>	Kilovolt
<b>LaOBr</b>	Lanthanum oxybromide
<b>mA</b>	Miliamper
<b>Mbit</b>	Milyonbit (Megabit)
<b>MeV</b>	Milyon elektron volt (mega electronvolt)
<b>mg</b>	Miligram
<b>MHz</b>	Megahertz
<b>mm</b>	Milimetre
<b>mR</b>	Milirad
<b>mW</b>	Miliwatt
<b>ns</b>	Nanosaniye

## **Simgeler Açıklamalar**

<b>°C</b>	Santigrat derece
<b>Ohm</b>	Direnç birimi
<b>PbO</b>	Kurşun Oksit
<b>s</b>	Saniye
<b>SbS3</b>	Antimon Trisülfid
<b>SeTeAs</b>	SezyumTellürArsenik
<b>V</b>	Volt
<b>V<sub>f</sub></b>	İleri kutuplama gerilimi
<b>λ</b>	Lambda

## **Kısaltmalar Açıklamalar**

<b>A/D</b>	Analog/Sayısal (Analogue/Digital)
<b>AC</b>	Alternatif Akım (Alternating Current)
<b>ADC</b>	Analog Sayısal Dönüştürücü (Analogue Digital Converter)
<b>AFH</b>	Adaptif Frekans Sıçramalı Geniş Spektrum (Adaptive Frequency Hopping Spread Spectrum)
<b>ANSI</b>	Amerikan Ulusal Standartlar Enstitüsü (American National Standards Institute)
<b>CAN</b>	Kontrolör Alan Ağı (Controller Area Network)
<b>CCD</b>	Charge Coupled Device
<b>CMRR</b>	Ortak Mod İşaret Bastırma Oranı (Common Mode Rejection Ratio)
<b>CPU</b>	Merkezi İşlem Birimi (Central Processing Unit)
<b>DC</b>	Doğru Akım (Direct Current)
<b>DIP</b>	Dual Inline Package
<b>DNA</b>	Deoksiribo Nükleik Asit
<b>DSP</b>	Sayısal İşaret İşleme (Digital Signal Processing)
<b>EDR</b>	Geliştirilmiş Veri Hızı (Enhanced Data Rate)
<b>EEPROM</b>	Elektrikle Yazılıp Silinebilir Salt Okunur Bellek (Electrically Erasable Programmable Read Only Memory)

<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>eSCO</b>	Genişletilmiş Senkron Bağlantı (Extended Synchronous Connections)
<b>FIFO</b>	İlk Giren İlk Çıkar (First In First Out)
<b>GPS</b>	Küresel Yer Belirleme Sistemi (Global Positioning System)
<b>GSM</b>	Global System for Mobile Communications
<b>HCI</b>	Host Controller Interface
<b>I<sup>2</sup>C</b>	Inter Integrated Circuit
<b>IBM</b>	International Business Machines
<b>ICD</b>	Devrede Hata Ayıklayıcı (In-Circuit Debugger)
<b>ICSP</b>	Devrede Seri Programlama (In Circuit Serial Programming)
<b>IDE</b>	Tümleşik Geliştirme Ortamı (Integrated Development Environment)
<b>IEEE</b>	Elektrik Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
<b>IrDa</b>	Kızılötesi Veri Birliği (Infrared Data Association)
<b>ksps</b>	Saniyede Bin Adet Örnekleme (kilo sample per second)
<b>LCD</b>	Sıvı Kristal Ekran (Liquid Crystal Display)
<b>LED</b>	Işık Yayan Diyot (Light Emitting Diode)
<b>MAC</b>	Ortama Erişim Kontrolü (Media Access Control)
<b>MIPS</b>	Saniyede Milyon Adet Komut Yürütme (Mega Instruction Per Second)
<b>MSPS</b>	Saniyede Milyon Adet Örnekleme (Mega Sample Per Second)
<b>MTF</b>	Modülasyon Transfer Fonksiyonu
<b>NFC</b>	Yakın Mesafeli İletişim (Near Field Communication)
<b>OBEX</b>	Nesne Değişimi (Object Exchange)
<b>OPAMP</b>	İşlemsel Yükselteç (Operational Amplifier)
<b>PACS</b>	Görüntü arşivleme ve iletişim sistemleri (Picture Archiving and Communication Systems)
<b>PAN</b>	Kişisel Alan Ağları (Personal Area Network)
<b>PC</b>	Kişisel Bilgisayar (Personal Computer)

## **Kısaltmalar    Açıklamalar**

<b>PCB</b>	Baskılı Devre Kartı (Printed Circuit Board)
<b>PDA</b>	Kişisel Sayısal Asistan (Personal Digital Assistant)
<b>PHY</b>	Fiziksel Katman Aygıtı (Physical Layer Device)
<b>PLL</b>	Faz Kenetlemeli Çevrim (Phase Locked Loop)
<b>PWM</b>	Darbe Genişlik Modülasyonu (Pulse Width Modulation)
<b>QoS</b>	Ağ İletişimi Hizmet Kalitesi (Quality of Service)
<b>RAM</b>	Rastgele Erişimli Bellek (Random Access Memory)
<b>RC13</b>	Port C 13 numaralı pin
<b>RE8</b>	Port E 8 numaralı pin
<b>RS232</b>	Önerilen Standart 232 (Recommended Standard 232)
<b>RSSI</b>	Alınan Sinyal Güç Göstergesi (Received Signal Strength Indication)
<b>SIG</b>	Özel İlgi Grubu (Special Interest Group)
<b>SIL</b>	Single In Line
<b>SPI</b>	Seri Çevre Birimi Arabirimi (Serial Peripheral Interface)
<b>TTL</b>	Transistör-Transistör Lojik
<b>TV</b>	Televizyon
<b>UART</b>	Evrensel Asenkron Alıcı Gönderici (Universal Asynchronous Receiver Transmitter)
<b>USB</b>	Evrensel Seri Yol (Universal Serial Bus)
<b>VoIP</b>	IP üzerinden ses transferi (Voice over IP)
<b>Wi-Fi</b>	Kablosuz Bağlantı Protokolü (Wireless Fidelity)

## 1. GİRİŞ

X Işınlarnın 1895 yılında Wilhelm Conrad Röntgen tarafından bir tesadüf sonucu keşfinden sonra yıllar içinde x-ışınları ile yapılan tıbbi görüntüleme cihazlarının çeşitleri ve görüntüleme kabiliyetleri teknolojik ilerlemelere paralel olarak artmış, günümüzde hemen hemen tıp ilminin tüm alt dallarında teşhis ve hatta tedavi amacına yönelik olarak yoğun olarak kullanılır hâle gelmişlerdir. Önceleri sadece kemik dokusundaki kırık, çatlak ve diğer bazı deformasyonları belirlemeye yarar düşük çözünürlüklü ve az kontrastlı görüntüler almaya olanak veren klasik anlamdaki röntgen filmlerinin yerini günümüzde sayısal olarak kontrol edilen gelişmiş x-ışını tüpleri sayesinde mümkün olan en az hasta dozunu uygulayarak yüksek kaliteli görüntüler almaya yarayan röntgen sistemleri almıştır. Röntgen tüpünün kullanımının farklı tekniklerle birleştirilmesi sayesinde yine zamanla iyileştirilen dental röntgen, floroskopi, mamografi, sayısal anjiyografi ve son olarak bilgisayarlı tomografi geliştirilip kullanılmaya başlanmıştır.

X-ışınlarının diyagnostik radyolojide kullanılmalarını sağlayan temel özellik, dokuyu geçebilme yetenekleridir. Floresan ve fotografik özellikleri ise görüntünün elde edilmesini sağlar. İnsan vücudu değişik atom ağırlığında ve değişik kalınlık ve yoğunlukta dokulardan yapıldığından, x-ışınının emilimi de farklı olacaktır. Farklı emilim ve girginlik sonucu, röntgen filmi (röntgenogram) üzerine değişik oranlarda düşen x-ışınları geçtikleri vücut parçasının bir görüntüsünü oluştururlar. Bu görüntü, siyahtan (film üzerine düşen ışın fazla) beyaza (film üzerine düşen ışın az) kadar değişen gri tonlardan oluşur.

Konvansiyonel röntgendeki gümüş bromür analog bir görüntüdür. Analog görüntüde gri renklerin değişimi kesintisizdir. Nümerik bir ölçek oluşturulursa analog bir görüntü sayısala çevrilebilir. Sayısallaştırma, analog görüntünün çok küçük resim elemanlarına bölünerek (piksel) sayısal olarak değerlendirilmesi ile yapılır. Bu sayısal değerlerin gri bir ölçekte karşılıkları olan gri ton bulunarak görüntü

oluşturulur. Röntgenogramların sayısallaştırılması demek olan bu işlemin pratik bir yararı yoktur.

Dental röntgenin diş hekimliğinin kullanımına sunulmasıyla, başka tekniklerle tespiti mümkün olmayan ve dişin çene kemiğine gömülü kısımlarının herhangi bir cerrahi işlem uygulanmaksızın görüntülenmesi ve gerekli müdahalelerin tespit edilerek uygulanması mümkün olmuştur. Sayısal röntgende ise vücudu geçen x-ışınları ya detektör zinciriyle ya da görüntü plağı ile veya floroskopi ekranının sayısallaştırılması ile saptanarak, görüntüler bilgisayar aracılığı ile oluşturulur. Sayısallaştırmanın röntgen filmini ortadan kaldırması arşiv sorununa da çözüm getirmektedir. Sayısal röntgen, konvansiyonel röntgene göre daha kolay işlemler içerir. Film banyo işlemi olmaması dolayısıyla işlem süresi kısadır. Alınan ışının dozu azaltılmıştır ve maliyeti düşüktür.

Floroskopide görüntülenmek istenen dokunun tek bir anlık görüntüsü yerine sürekli olarak hareketli bir görüntünün elde edilmesi mümkündür. Bu şekilde mekanik olarak hareket eden organların çalışmaları ve işlev bozuklukları görüntülenebilmektedir.

Bilgisayarlı Tomografide ise sayısal koordinat dönüşümleri ile görüntülenmek istenen nesne etrafında döndürülen hareketli bir x-ışını kaynağının dokudan geçtikten sonraki izdüşümü algılanarak ilgilenilen bölgenin herhangi bir açıdan kesit görüntüsü alınabilmekte, hatta üç boyutlu hacimsel görüntüleme yapılabilmekte, dahası bu görüntüler tanı amacına yardımcı olması amacıyla yönelik olarak ihtiyaca uygun olarak renklendirilebilmektedir.

Tüm bu faydalarının yanında x-ışın tüpü temelli tıbbi görüntüleme aygıtlarının, x-ışınlarının iyonlaştırıcı radyasyon kaynağı olmaları sebebiyle izin verilen belli bir doz üzerinde vücut dokularında tahribata yol açtığı ve hücre yapısını bozduğu, dahası ileri vakalarda dna dizilimini mutasyona uğrattığı bilinmektedir. Bu sebeptendir ki, yaşamları boyunca x-ışınlarının zararlı etkilerinin farkında olmadan yaptıkları bilinçsiz ölçüm ve deneyler yüzünden Röntgen, eşi ve laboratuvar ekibi genç yaşta

kanser ve buna baęlı hastalıklar yüzünden yaşamlarını kaybetmişlerdir. Görüntüleme de kullanılacak x ışınının miktarı her teknik için maksimum üst limitler verilerek belirlenmiştir.

Tıbbi görüntüleme cihazının, dolayısıyla iyonlaştırıcı radyasyon kaynaęı olan x ışın tüpünün istenildięi gibi çalışmasını sağlayan bazı kritik öneme haiz parametreler vardır. Bunlardan en önemlileri filaman akımı, tüp gerilimi ve tüp akımıdır. Bu tezde yapılan çalışmada tüp akımı ve buna baęlı doz miktarını belirleyen miliamper, miliamper-saniye büyüklükleri ölçülerek dozajın olması istenen ölçüde kalibrasyonuna imkân sağlanmıştır.

2. Bölümde X ışınlarının keşfinden başlayarak Işın Fizięi ve Işın spektrumu anlatılmış, Işıma tipleri olarak Alfa, Beta, Gama, Katot, Kozmik ve X ışınlarından kısaca bahsedilerek radyolojide temel kavramlar üzerinde durulmuştur.

3. Bölümde Röntgen cihazlarının yapısı ve X Işınının elde edilmesi anlatılmış, X Işını üretmekte kullanılan ototransformatör, yüksek gerilim transformatörü, doğrultmaçlar ve yüksek gerilimin iletiminde kullanılan özel kablolar kısaca tanıtılmıştır. Klasik ve Modern Röntgen tüplerine örnekler verilmiş, X Işınlarının elde ediş detaylı incelenmiş ve tez kapsamında imal edilen mAs metre'ye neden ihtiyaç duyulduęunun cevabının bulunacaęı Röntgen Cihazlarında Kalibrasyon işleminin anlatılmıştır.

4. Bölümde Tıbbi Görüntüleme cihazlarından X Işın Tüpünü kullanan Mamografi, Floroskopi, Anjiyografi gibi tekniklere yer verilmiş, görüntü algılanmasında kullanılan ekran-film kombinasyonları, x-ray görüntü yoğunlaştırıcılar ve sayısal görüntüleme yapan sistemlerden söz edilmiştir.

5. Bölümde Bilgisayar ve ölçü aleti arasında kablosuz veri aktarımını sağlayan Bluetooth protokolü açıklanmaya çalışılmış, bu kapsamda kullanım alanlarından, altyapısındaki teknik özelliklerinden ve farklı sürümlerinin özelliklerinden aşama aşama bahsedilmiştir.

6. Bölümde mAs metre'yi uzaktan kontrol etmeye yarayan bilgisayar yazılımı ve donanımda kullanılan mikrodenetleyici için kod geliştirme aşamaları izah edilmiş ve kod geliştirmede kullanılan MPLAB IDE, C30 derleyici, Easydspic kartı gibi araçlar tanıtılmıştır.

7. Bölümde mAs metre'nin besleme katından başlayarak, sırasıyla anahtarlama-ön ölçekleme katı; aletsel yükselteç, hassas doğrultmaç, ekspozur algılama gibi opamp blokları; mikrodenetleyici ve bluetooth iletişim modülünün yer aldığı elektronik devre blokları etraflıca anlatılmıştır.

8. Bölümde mAs metre'yi kullanmak isteyen birinin faydalanabileceği sadelikle cihazın kendi başına ya da bilgisayar aracılığıyla kullanılması anlatılmış ve fonksiyonları tanıtılmıştır.

9. Bölümde de mAs metre ile alınan ölçüm sonuçları verilmiş ve bunlar incelenmiş, ileride benzeri bir cihaz tasarlayacak ya da mAs metre'yi daha da geliştirecek olan araştırmacılar için önerilerde bulunulmuştur.

## 2. RADYOLOJİDE TEMEL KAVRAMLAR

Bu bölümde radyoloji biliminde esas teşkil eden temel kavramlar, içeriği, keşfinin tarihçesi ve nitelikleri kısaca açıklanacaktır.

### 2.1. Radyolojinin Tanımı ve Kapsamı

Radyoloji (radiology); terim olarak, “radyo” (radio) ve “loji” (logy) sözcüklerinden meydana gelmiş olup, dar anlamda “ışın bilimi” demektir. Terimdeki “radyo” sözcüğü, “ışın yayımı” anlamına gelen “radyodifüzyon” sözcüğünün kısaltılmış hâli, “loji” ise “bilim” sözcüğünün karşılığıdır. “Radyoloji” kelimesinin sözlük anlamı olarak; “ışık, elektrik ve ısı radyasyonlarının hekimliğe uygulanış biçimidir. Radyoloji biliminin tıp alanındaki uygulamaları başladığı yıllarda, Amerikan Radyoloji Heyeti’nce “Radyoloji; X ışını, Radium ve diğer radyoaktif maddelerin hastalıkların teşhis ve tedavisinde kullanımını konu alan bilim dalıdır.” şeklinde tanımlanmıştır. Yıllarca bu tanım sınırları dâhilinde kalan radyoloji bilimi, bugün yeni gelişmelere paralel olarak kapsamını genişletmiştir. Buna göre, radyoloji; X ışını ve maddelerin hastalıkların teşhis ve tedavisinde kullanımını ve bu amaçla geliştirilen teknik ve yöntemleri konu alan bilim dalıdır.

### 2.2. X-ışınlarının ve Röntgenin Keşfi

Röntgen ışınları 8 Kasım 1895 tarihinde Alman fizik profesörü Wilhelm Conrad Röntgen tarafından keşfedilmiştir. Keşfinin üzerinden kısa bir zaman geçmiş olmasına rağmen, bu ışınlar tıpta, ilmi araştırmalarda ve endüstride geniş uygulama alanları bulmuş olup, insanlık âlemine büyük yararlar sağlamaktadır.

Wilhelm Conrad Röntgen, camdan yapılmış ve basıncı düşürülmüş bir tüpte “katot ışınlarının soğrulması” üzerine deneyler yaparken, tüpün iki elektrotu arasına yüksek voltaj uygulamış ve bu sırada tüpten birkaç metre uzakta bulunan “Baryum Platin Siyanür” kaplı bir “A” harfinin (levhanın) parıldamakta olduğunu gözlemiştir. Bu

parıldamanın etraftaki herhangi bir cisimden gelebileceğini düşünerek, ışıkları söndürüp deneyi tekrarladığında, parıldama olayının da tekrarladığını görmüştü. Bunun katot ışınlarının bir eseri olup olmadığını anlamak için, tüpün üzerini siyah bir kartonla kaplayıp deneyini tekrar ettiğinde aynı olayın mevcudiyetini görünce, buna katot ışınlarının sebep olamayacağını, ancak daha değişik özelliklere sahip bir ışının özelliğinden dolayı olabileceğini anlamıştı. Zira tekrarlanan deneyler sonucunda, tüp üzerine konan siyah kâğıt, karton ve tahta parçaları parıldamayı engelleliyordu. Oysa katot ışınlarının bir maddeyi delip geçme özelliklerinin olmadığını biliyordu.

Röntgen, bu parıldamanın değişik bir ışın tarafından olabileceğini tahmin etmiştir. Nitekim bu ışınların, kurşun gibi bazı maddeler tarafından durdurulduğunu da deneyler sonucu ortaya koymuştur. “Baryum Platin Siyanür” kaplı bir kâğıt ile tüp arasına elini koymuş ve tüpten akım geçirince elinin kemiklerini “Baryum Platin Siyanür” kaplı kâğıt üzerinde görünce çok şaşırmıştır. Anot voltajını arttırınca parıldamanın da arttığını gören Wilhelm Conrad Röntgen, bu parıldamanın yeni bir ışın tarafından oluşturulduğunu keşfetmiştir. Bu yeni ışına matematikte bilinmeyen anlamına gelen X sembolünden esinlenerek “X-IŞINLARI” adını vermiştir.

### **2.3. Işın Fiziği ve Işın Spektrumu**

Bazı maddeler kendiliğinden, bazıları ise dışarıdan aldıkları enerji nedeni ile dışarıya ışın salarlar. Işınlar, orijinleri ve türlerine göre değişik karakterlerde olurlar. Yapı bakımından partiküler veya elektromanyetik dalga özelliği taşırlar. Partiküler ışınlar, radyoaktif maddeler tarafından salınan yüklü parçacıklardan oluşmasına karşın, elektromanyetik dalga türünde olanlar, yüksüz enerji paketlerinden oluşurlar. Doğal veya yapay değişik ışınlar olup, her birinin enteresan özellikleri vardır.

### **2.4. Partiküler Radyasyonlar**

Partiküler radyasyonlar; Uranyum, Toryum, Polonyum ve Radyum gibi radyoaktif maddelerin çekirdeklerindeki kendiliğinden parçalanma sonucu çevreye salınan ışınlardır. Şu hâlde; partiküler ışınlar, radyoaktif maddelerin parçalanma ürünleridir.

Radyoaktif maddelerin ışın yayımında dışarıdan enerji alma söz konusu olmayıp, ışın yayımı çekirdeğin parçalanmasından ileri gelmektedir.

Radyoaktif elementlerin çekirdeklerinin parçalanması sırasında Alfa, Beta ve Gama ışınları salınmakta ve bu salınım, ilgili elementin daha düşük atom numarasına sahip kararlı bir elemente dönüşüncüye kadar devam etmektedir. Örneğin radyoaktif özellik taşıyan Uranyum-238'in ışın yayımı, atom ağırlığı 206 olan kurşuna dönüşüncüye kadar devam eder.

#### 2.4.1. Alfa ( $\alpha$ ) ışınları

Bazı radyoaktif elementlerin (özellikle ağır elementlerin) çekirdeklerinden salınan, pozitif elektrik yüklü parçacıklardan oluşan ışınlardır. Özellikle atom numarası 70'den büyük olan, nötron bakımından fakir çekirdekler tarafından salınırlar. Bu ışınlar, radyoaktif ışınların büyük bir kısmını teşkil ederler. Bir alfa taneciği iki proton ve iki nötrondan yapılmış olan Helyum atomu ile özdeştir. Başka bir ifade ile: alfa ışınları, Helyum atomunun çekirdeğinden başka bir şey değildir. Alfa ışınları doğal Uranyum, Radyum, Toryum ve Polonyum gibi kararsız izotoplar tarafından salınır. 1 gr. radyumda saniyede  $3,7 \times 10^{10}$  alfa parçacığı salınır. Böyle bir radyoaktif element, alfa parçacığı salarak yeni bir elemente dönüşür. Bazı radyoaktif elementler de alfa ışını salarlar.

- Alfa ışınları pozitif yüklü olduklarından manyetik alanlarda negatif kutba doğru saparlar.
- Alfa ışınları kuvvetli iyonizasyon etkisi gösterirler. 3 MeV değerinde alfa parçacıkları, havadaki kendi gidiş istikametlerinde saniyede 40 000 iyon çifti meydana getirme yeteneğine sahiptir.
- İyonizasyon özellikleri çok olmasına karşın, delicilik (penetrasyon) yetenekleri zayıftır. Alfa ışınları 6 mikron kalınlığındaki Alüminyum veya 2,5 mikron kalınlığındaki Bakır levhada tam olarak yutulurlar. İnce bir kâğıt yaprak bile alfa ışınlarını tutmaya yeterlidir. Vücudun bu ışınlara maruz kalması hâlinde, deri

tarafından tutulurlar. Bir gaz içinde alfa ışınlarının ulaşabileceği uzaklık o gazın yoğunluğu ile ters orantılıdır.

- Hızları, salındıkları kaynağa göre değişir. Örneğin Radyumdan çıkan alfa ışınlarının hızları, saniyede 14 000 km ile 25 000 km arasındadır.
- Alfa parçacıklarının en enerjik olanları 9 MeV kadarlık bir enerjiye sahiptir.
- Bazı maddelerde floresan etki gösterirler.
- Alfa ışınları, çinko sülfür ihtiva eden ekranlarda kıvılcıma neden olurlar.
- Fotoğraf plaklarına etki ederler.
- Biyolojik etkileri vardır.
- Solunum ve sindirim yoluyla alındıklarında zararlı etkileri çok daha fazla olur.

#### 2.4.2. Beta ( $\beta$ ) ışınları

Radyoaktif elementlerin çekirdeklerinden salınan (-) veya (+) yüklü parçacıklardan oluşan ışınlardır. Bu ışınlar, esas itibariyle elektron kaynaklı olduğundan gerçekte katot ışınlarından başka bir şey değildir. Ancak; beta ışınları yapı olarak katot ışınlarıyla özdeş olmakla birlikte elektrik yükleri negatif veya pozitif olabilmektedir. Beta partiküllerinin salınımı, radyoaktif elementin çekirdeğindeki proton veya nötron fazlalığından ileri gelir. Çekirdekteki nötron fazlalığı nedeniyle salınan parçacıklar (-) yüklüdür. Radyoaktif maddelerin saldıgı beta ışınları, çekirdekteki proton fazlalığından ileri geliyorsa salınan partikül (+) yüklüdür. Negatif yüklü beta ışınlarının katot ışınlarından farkı hızlarının daha fazla olmasıdır. Bir atomun beta ışını yayması, o atomun atom ağırlığında bir değişikliğe sebep olmaz. Fakat radyoaktif elementin beta ışını yayması atomun elektrik yükünü arttırma veya eksiltme yönünde önem taşır.

- Beta parçacıkları, alfa parçacıklarından 7350 defa daha hafiftirler.
- Beta ışınlarının kinetik enerjileri, alfa ışınlarının kinetik enerjilerinden birkaç MeV daha yüksektir.
- Aynı kaynaktan çıkan beta parçacıklarının hepsi aynı hızda değildir. Bu ışınların hızları, saniyede 60 000 km ile 290 000 km arasında değişir. Beta parçacıkları hafif olduklarından, önlerine çıkan gaz moleküllerine çarpma, hızları ve

yönleri değişir. Bu nedenle; gidiş istikametleri alfa parçacıkları gibi düz değildir, zikzaklıdır. Ancak; çok hızlı beta parçacıklarının gidiş istikametlerinin düz olduğu bilinmektedir.

- Beta partiküllerinin iyonizasyon yetenekleri alfa partiküllerinden daha zayıftır. Yavaş beta ışınları hızlı olanlara göre daha iyonizendir. 3 MeV'lik alfa parçacıkları, kendi gidiş istikametlerinde saniyede 40 000 iyon çifti meydana getirme yeteneğine sahipken, beta ışınları saniyede ancak 30-40 iyon çifti meydana getirebilirler,
- Beta ışınları bazı maddelerde floresans olayına sebep olurlar.
- Beta ışınlarının penetrasyon yetenekleri alfa ışınlarının penetrasyon yeteneğinden yüksektir. Kinetik enerjisi 3 MeV olan bir beta parçacığı havada 1000 cm'lik bir yol alabildiği hâlde, aynı enerjiye sahip alfa parçacığı aynı koşulda sadece 2,8 cm'lik yol kat edebilir.
- Beta ışınlarının tam absorbe edilebilmesi için 0,5 mm platin veya 1 mm kurşun gereklidir.

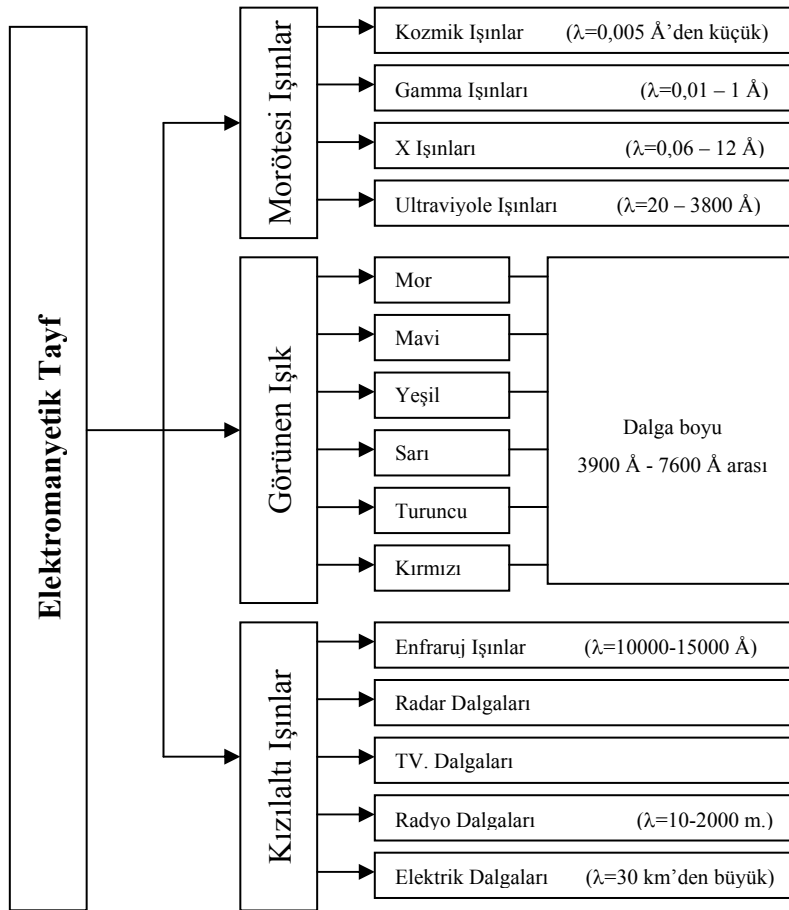
### 2.4.3. Katot ışınları

Havası boşaltılmış (vakum edilmiş) bir tüpün katodundan çıkan ve anoda uygulanan (+) voltaj nedeniyle, hızla anoda doğru ilerleyen elektron demetidir.

### 2.5. Elektromanyetik Radyasyonlar

Elektromanyetik radyasyonlar; atomda herhangi bir etkenle uyarılan elektronların yörüngeler arası hareketler sonucu çevreye salınan yüksüz enerji paketleridir. Morötesi ışınlar grubunda yer alan Kozmik ışınlar, Gamma ışınları, X-ışınları ve Ultraviyole ışınları tıpkı görülen ışık gibi elektromanyetik dalga özelliği taşırlar. Kızılötesi grupta yer alan Enfraruj ışınlar, Radar, TV, Radyo ve elektrik dalgaları da yapı olarak elektromanyetik dalga özelliğindedir. Ancak söz konusu elektromanyetik dalgaların karakterleri, meydana geldikleri atomun cinsi ve dalganın meydana geliş şekline göre değişir. Başka bir ifade ile elektromanyetik dalgaların cinsi değiştikçe, özellikleri de değişmektedir. Atomun iç yörüngelerine doğru gidildikçe, meydana

gelen elektromanyetik radyasyonların dalga boyu kısalmı, frekans ve enerjileri artar. Elektromanyetik dalgaların karakteristik özelliklerini, sahip oldukları dalga boyu belirler. Bu nedenle farklı dalga boyuna sahip elektromanyetik dalgalar, farklı özellik gösterirler. Örneğin kısa dalga boyulu ışınlar olan gamma ve X-ışınları iyonizasyon özellikler taşıdıkları hâlde, uzun dalga boyulu olan TV ve Radyo dalgaları böyle bir özellik taşımaz. Dalga boyu ölçü birimi “Angström” (Å) olup,  $1 \text{ Å} = 10^{-8} \text{ cm}$ ’dir. Elektromanyetik dalgaların dalga boylarına göre sıralanmış hâline “Elektromanyetik Tayf” denir. Şekil 2.1’de Elektromanyetik Tayf verilmiştir.



Şekil 2.1. Elektromanyetik tayf

### 2.5.1. Kozmik ışınlar

Elektromanyetik tayfin en kısa dalga boylu ışınlarıdır. Gerçek kaynakları bilinmemekle birlikte, güneşten gelen yüksek enerjili radyasyonlar da kozmik ışınlar sınıfında sayılmaktadır. Dalga boyları 0,005 Å'den küçüktür. Çok yüksek enerji taşımaları nedeniyle canlılar için tehlike arz ederler. Ancak büyük bir kısmı, atmosfer tarafından tutularak, tehlike önlenmektedir.

### 2.5.2. Gamma ( $\gamma$ ) ışınları

Radyoaktif elementlerin parçalanması, atom reaktörleri veya atom bombası patlamasından sonra meydana gelen çöküntülerin parçalanma ürünleri olan iyonizen karakterde, kütleleri olmayan, yüksek enerjili elektromanyetik radyasyonlardır. Orijinleri radyoaktif maddeler olsa da, yüklü parçacık yapısında olmayıp, yüksek enerjileri fotonlardan meydana gelmiş elektromanyetik dalga özelliği taşırlar. Radyoaktif elementlerin çekirdeklerinden alfa ve beta parçacıklarının çıkışı çekirdeği uyarılmış duruma sokar. Böylece bir radyoaktif elementten gamma ışını salınması, uyarılmış olan çekirdeğin uyarılma enerjisini kaybetmesini sağlar. Şu hâlde alfa, beta ve gamma ışınları radyoaktif maddelerden aynı anda salınmaktadır. Uyarılmış hâldeki çekirdek, üzerindeki fazla enerjiyi atmak için gamma ışını salarak kararlı hâlde geçer. Çekirdek ne kadar yüksek enerji seviyesinde uyarılmış kalmışsa, yayılacak gamma ışınları da o kadar yüksek enerjili olmaktadır. Bu nedenle radyoaktif çekirdekler milyarlarca elektron voltluk gamma ışını yayarlar.

- Gamma ışınları elektromanyetik tayftaki diğer ışınlar gibi yüksüzdürler.
- Dalga boyları 0,01 Å ile 1 Å arasında değişir. Yani dalga boyları X-ışınlarından kısadır.
- Yüksüz olduklarından manyetik alanlarda sapmazlar.
- İyonizasyon yetenekleri çok azdır. Havada kendi doğrultusunda, her santimetrede birkaç iyon çifti oluşturabilirler.
- Gamma ışınlarının iyonizasyon yeteneklerinin zayıf olmasına karşın, delicilik (penetrasyon) yetenekleri çok yüksektir. Penetrasyon yetenekleri çok yüksek olan

gamma ışınları, birkaç cm'den 20 cm kalınlıktaki kurşunu bile geçebilirler. Bu ışınları durdurabilmek için 30 cm kalınlığında kurşun levhaların gerektiği durumlar olabilir.

- Gamma ışınlarının tümünün tam anlamıyla absorpsiyona uğramasına olanak yoktur denilebilir.
- Dalga boyları penetrasyon yetenekleriyle ters orantılıdır.
- Gamma ışınları homojen ışınlardır. Oysa bir X-ışın demeti heterojen yapıdadır.
- Fotokimyasal etkileri vardır.
- Fotoğraf plaklarına etki edebilirler.
- Bazı maddelerde floresans olayına sebep olabilirler.
- Biyolojik etkileri vardır.
- Enerjileri dalga boyları ile ters orantılıdır. Gamma ışınlarının enerjisi 2MeV değerindeki X-ışını enerjisine denktir. Gamma ışınları tıbbi gereçlerin sterilizasyonunda da kullanılmaktadır. Bu yöntemde, ürünler ambalajlandıktan sonra sterilize edildiğinden son derece güvenilir bir yöntemdir.

### 2.5.3. X - ışınları

Elektromanyetik tayfta morötesi ışınlar grubunda gamma ışınlarıyla ultraviyole ışınları arasında yer alan, yüksek enerjili fotonlara sahip iyonizen ışınlardır.

- X-ışınlarının dalga boyları 0,06 Å ile 12 Å arasında değişmektedir.
- X-ışınları elektromanyetik dalga hâlinde düz bir istikamette yayılırlar.
- Penetrasyon, absorpsiyon ve iyonizasyon özellikleri olup, bu özellikleri dalga boyu ve etkileşimde buldukları maddenin yapısal özelliklerine göre değişir.
- X-ışınlarının penetrasyon özellikleri nedeniyle görünen ışığın geçemediği yerlerden geçerler.
- İyonizasyon özellikleri nedeniyle katı sıvı ve gazları iyonize ederler.
- Dalga boyları frekansları ile ters orantılıdır.
- Bir X-ışını demetini oluşturan ışınlar, atomun değişik yörüngelerinden meydana geldiğinden bu ışın demeti homojen değil, heterojendir. Yani X-ışını demeti, birbirinden farklı dalga boylarına sahip ışıklardan oluşur.

- Dalga boyları küçük olduğundan, gözün spektral hassasiyeti dışında kalırlar ve gözle görülemezler.
- X-ışınları yüksüz olduklarından manyetik alanlarda sapmazlar. Normal koşullarda kırılmazlar ve yansımazlar. Ancak bazı özel koşullar ve kristallerde kırılabilirler.
- Hızları havada 300 000 km/s olup ışık hızına eşittir.
- Fotografik özellikleri olup, bu özellikleri sayesinde fotografik materyaller üzerinde görüntü oluşumuna imkân verirler.
- X-ışınları, kimyasal özellikleri nedeniyle sıvılarda (dolayısıyla bol su ihtiva eden organizmalarda) bazı kimyasal değişikliklere neden olurlar.
- X-ışınları sahip olduğu biyolojik özelliklerinden ötürü canlı dokularda çeşitli zararlı etkilerde bulunurlar.
- X-ışınlarının flüoresans özellikleri vardır. Bu ışınların keşfine de bu özellik vesile olmuştur.
- X-ışınları çarptıkları katı maddelerden sekonder ışın meydana getirirler.
- X-ışınları bazı madensel tuzlarda renk değişikliğine sebep olurlar.

#### **2.5.4. Ultraviyole ışınlar**

Elektromanyetik tafta morötesi ışınlar grubunda yer alırlar. Doğal olarak güneşten gelen ışınlar, yapay olarak da elde edebilirler. Güneşten gelen Ultraviyole ışınlar 3000 Å'dan daha kısa dalga boylu olanları atmosferin yüksek tabakaları tarafından absorbe edilirler.

#### **2.5.5. Görünür ışık**

Işık spektrumu; mor ötesi ışınlarla kızılaltı ışınlar arasında yer alan ve renkleri oluşturan elektromanyetik dalgalardan oluşur. Işık spektrumu, 3900 Å ile 7600 Å arasında ışık dalgalarıyla sınırlıdır. Beyaz ışıktan oluşan ışık spektrumu, dalga boyları itibariyle gözün spektral hassasiyeti içinde kaldığından, ışık spektrumunda yer alan elektromanyetik dalgalar (renkler) gözle görünür.

### **2.5.6. Enfraruj ışınlar**

Kızılaltı ışınlar bu ışınlarla başlar. Dalga boyları 10 000 – 15 000 Å'dır. Bu ışınlar, havadaki sisten geçebilirler. Bu özellikleri nedeniyle, bu ışınlardan hava fotoğrafçılığında yararlanılır.

Enfraruj ışınlar, (dokularda meydana getirdiği ısı nedeniyle) dolaşımın düzenlenmesinde, ağrıların dindirilmesi ve dokuların daha iyi beslenebilmesi amacıyla tıpta kullanılır. Bu ışınlar ayrıca romatizmal hastalıklarda da yararlı sonuçlar verir.

### **3. RÖNTGEN CİHAZLARININ YAPISI VE X-IŞINLARININ OLUŞUMU**

#### **3.1. Ototransformatör**

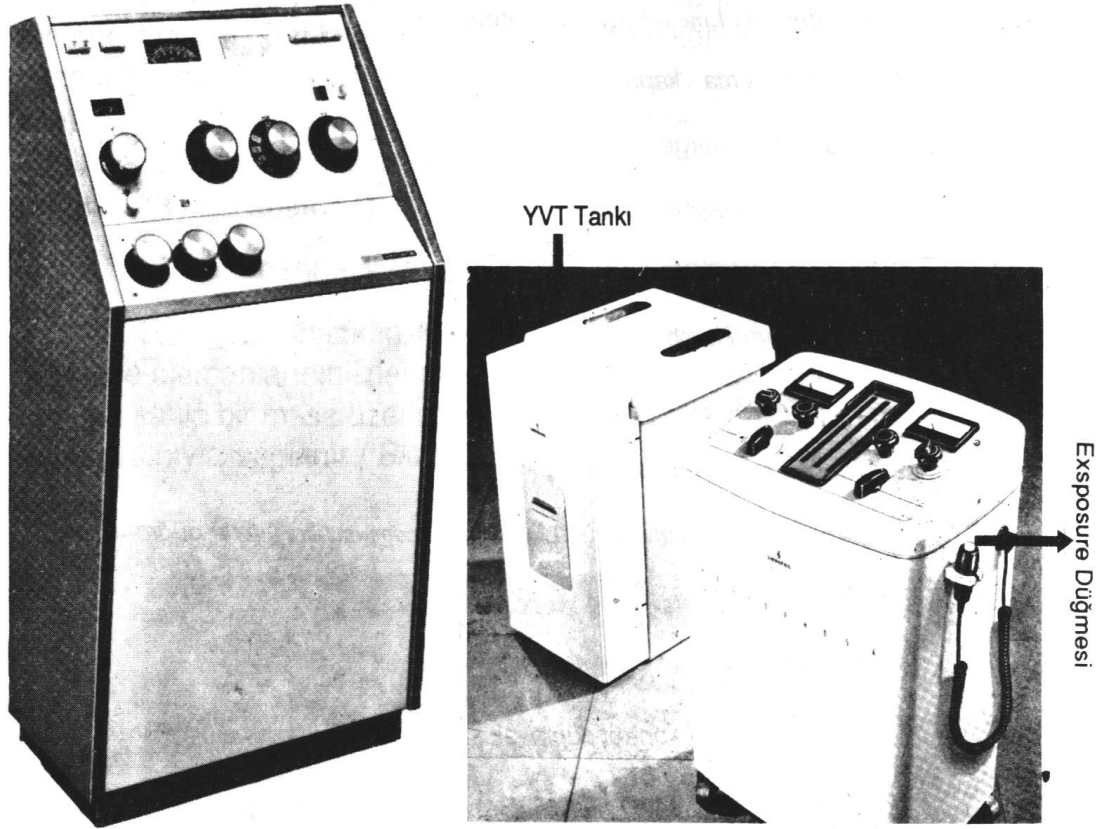
Ototransformatör şehir voltajının giriş yaptığı transformatördür. Ototransformatör primer ve sekonder devreden oluşur. Serbest transformatörlerde primer devre ile sekonder devre arasında elektriksel bir bağlantı vardır. Ototransformatörlerde giriş voltajının uygulandığı uçlar primer, voltajın alındığı uçlar ise sekonder devreyi oluşturur. Hacimlerinin küçük olması, primer ve sekonder devre arasındaki voltaj farkının küçük olması nedeniyle, kullanıcıya bir zarar vermeksizin röntgen cihazı kumanda masası içerisinde muhafaza edilebilir. Röntgen tüpüne uygulanacak değişik yüksek voltajı elde etmek için, yüksek gerilim transformatörünün primer devresine uygulanan voltaj değiştirilir. Bu voltaj değişikliği ototransformatör tarafından sağlanır. Ototransformatörlerin primer ve sekonder devrelerindeki birer uç hareketlidir. Bu uçlar kumanda masasındaki kontrol düğmeleriyle hareket ettirilerek yüksek gerilim transformatörünün sekonderindeki voltaj değiştirilebilir.

#### **3.2. Yüksek Gerilim Transformatörü**

Yüksek gerilim transformatörü, ototransformatörden aktarılan gerilimi yükselterek tüpe yüksek voltaj sağlar. Yüksek gerilim transformatörünün sekonderindeki sarım sayısı, primerindeki sarım sayısından çok fazladır. Yüksek gerilim transformatörü primerine uygulanan voltajı sarım sayılarıyla orantılı olarak artırır. Ototransformatörün sekonderi yüksek gerilim transformatörünün primerine bağlı olduğundan, yüksek gerilim transformatörünün sekonderinde oluşması istenen gerilim ototransformatörden ayarlanabilir. Bu da kumanda masasında bulunan “KiloVolt Seçici Düğme” aracılığıyla yapılır.

Yüksek gerilim transformatörü ışınlama süresince yüksek voltaj ihtiva ettiğinden, elektriksel yalıtıklığı sağlamak için, içi yağ dolu olan “Yüksek Voltaj Tankı”

içerisinde bulundurulur. Resim 3.1’de yüksek voltaj tankı ve kumanda masası görülmektedir.



Resim 3.1. Yüksek voltaj tankı ve kumanda masası

Trafo yağları, transformatörlerde sargıların elektriksel izolasyonunu sağlamak ve transformatör yüklü iken oluşan ısıyı transfer etmek için kullanılan yüksek kaliteli yağlardır. Bu yağların özellikleri aşağıdaki şekilde özetlenebilir:

- Yüksek elektriksel izolasyon değerleri.
- İyi ısı iletim karakteristiği.
- İyi oksidasyon ve kimyasal stabilite.
- Yüksek parlama noktası ve düşük uçuculuk özelliği.
- Düşük akma noktası.
- Düşük viskozite.

- Organik asitlerden, korozif sülfür ve transformatörlerde kullanılan metallere ters etki edebilecek diğer kirleticilerden arınmış olmak.

### **3.3. Doğrultmaçlar**

Doğrultmaçlar, yüksek gerilim transformatörünün sekonderinde meydana gelen yüksek gerilimli AC sinyali DC'ye çevirme görevi yapar. Doğrultmaçlar, yüksek gerilim devresinde, yüksek gerilim transformatörünün sekonder devresi ile röntgen tüpü arasında bulunurlar.

### **3.4. Yüksek Gerilim Kabloları**

Yüksek gerilim kabloları, yüksek gerilim devresindeki doğrultmaçlardan çıkan DC sinyali, tüpe taşıyan kablolardır. Işınlama süresince üzerlerinde yüksek gerilim bulundurulur. Bu nedenle özel olarak imal edilmişlerdir.

### **3.5. Röntgen Tüpleri**

Röntgen tüpleri röntgen cihazının çok önemli bir parçasıdır. X-ışınları bu tüplerde elde edilir. Değişik tip ve özellikte röntgen tüpleri üretilmektedir.

#### **3.5.1. Klasik röntgen tüpleri**

1895 yılında X-ışınlarının keşfi sırasında “Crookes” tüpüne benzer, havası boşaltılmış bir cam tüp kullanılıyordu. İçindeki hava, bir pompa vasıtasıyla devamlı olarak boşaltılıyordu. Crookes tüpleri, 1899 yılından itibaren yerlerini gazlı tüplere bırakmıştır. Bu tüplerin içerisinde üretim safhasında Nitrojen veya Oksijen gazı bırakılıyordu. Bu gazlar ise çeşitli sakıncalar doğuruyordu. Bu tüplerde, katot karşısında anot yanında bir de anti katot yer alıyordu. Gazlı tüpleri yüksek mA değerlerinde kullanmak mümkün olmuyordu. Genellikle 5mA civarında bir akım sağlanabiliyordu. Geniş kV değerleri de kullanılamıyordu. İstenen kısa dalga boylu ışınlar elde edilemiyordu. Düşük tüp akımı uzun ışınlama zamanı gerektiriyordu.

Fokus alanının büyük olması radyografik kaliteyi engelliyordu. Tüpün şekli ve büyüklüğü istenmeyen primer radyasyonuna karşı koruyucu değildi.

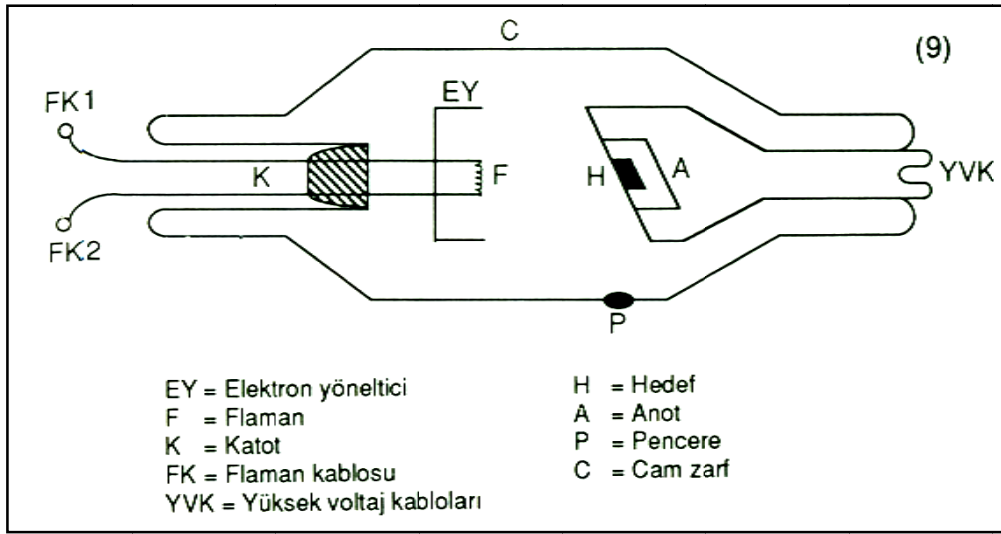
1913 yılında Amerikalı fizikçi ve kimyacı William David Coolidge tarafından gazlı tüplerin sakıncalarını ortadan kaldıran ve termoiyonik neşriyat prensibine göre çalışan Coolidge tüpleri keşfedildi. Bu tüpler, şekil bakımından gazlı tüplere benzese de, içerisindeki hava tamamen boşaltılmış ve içerisinde hiç gaz bırakılmamıştır. kV ve mA değerleri bağımsız olarak değiştirilebiliyordu. Yani; kV değerinde yapılan bir değişme mA değerinde bir değişme yapmadığı gibi, mA değerinde yapılan bir değişme de kV değerinde bir değişmeye sebep olmuyordu. Oysa gazlı tüplerde, mA arttırıldığında kV değeri de artıyordu. Yapılan ilk Coolidge tüpünde, 200 mA değerinde bir akım kullanılabilmiş ve bu sayede ışınlama zamanında büyük oranda azalma mümkün olmuştur.

Bugün kullanılan röntgen tüpleri, Coolidge tüpü prensibine dayanarak yapılmıştır. Modern röntgen tüpleri, Coolidge tüplerinden farklılık gösterse de, prensip itibariyle önemli bir farkları yoktur. Coolidge'nin geliştirdiği tüplerde; tüpün içi vakum olduğundan, tüp içinde gaz iyonizasyonu tamamen yok edilmiştir. Elektron kaynağı, spiral şeklinde tungsten (volfram) filamandır. Filaman transformatöründen gelen akımla ısıtılan filaman elektron yaymakta, anoda uygulanan (+) yüksek voltaj nedeniyle katottaki elektronlar anoda doğru hareket ederek “tüp akımını” (katot ışınlarını) meydana getirmektedir. Filamanı ısıtan akım arttırıldıkça, katot daha çok elektron yaymakta ve bu elektronların meydana getireceği tüp akımı da arttırılmış olmaktadır. Anot, bakır içerisine gömülmüş olan Tungsten'den yapılmıştır. Tüpte (-) elektrot katot, (+) elektrot ise anot durumundadır.

### **3.5.2. Modern röntgen tüpleri**

Röntgen tüpü genel olarak, X-ışınlarının oluşturulmasına yarayan, havası boşaltılmış ve içerisinde katot ve anot olmak üzere iki elektrot bulunan cam bir zarf olarak tarif edilebilir. Cam zarf, tüpün diğer kısımlarını içinde bulundurur. Havası tamamen boşaltılmış bir koruyucu olup, modern tüplerde genellikle koruyucu şeklinde

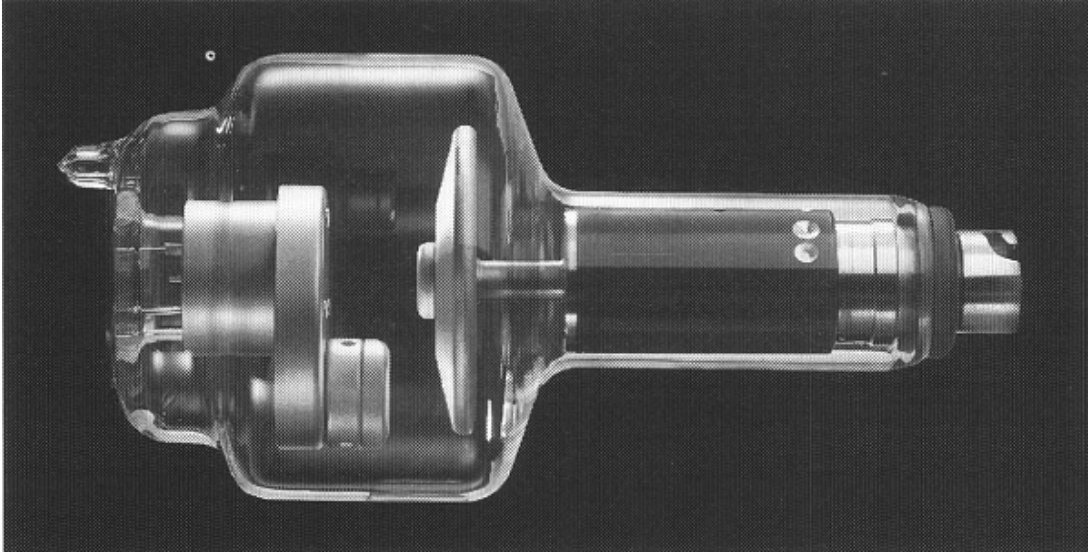
üretilmektedir. Tüpün havasının boşaltılmış olması, filamanın havadaki oksijenle oksitlenmesini önleyerek, tüpün daha uzun süre kullanılabilmesine olanak sağlar. Ayrıca, filaman tarafından yayılan elektronların, anoda doğru hareketleri sırasında onların hızlarına engel teşkil edecek gaz atomlarının ortadan kaldırılması sağlanmış olur. Fokusdan çıkan x-ışınlarının tüpten dışarı çıktığı kısım daha zayıf yapıda olup, bu kısma “pencere” denir. Şekil 3.1’de sabit anotlu bir röntgen tüpü görülmektedir.



Şekil 3.1. Sabit anotlu röntgen tüpü.

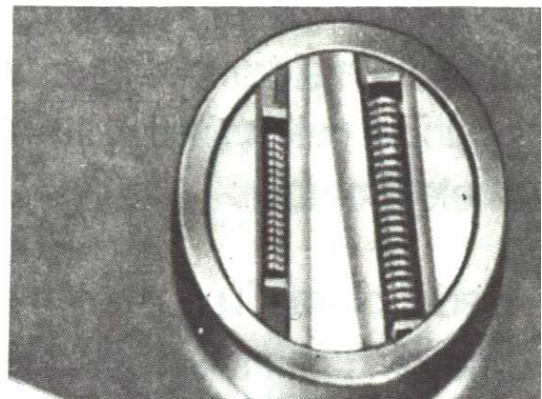
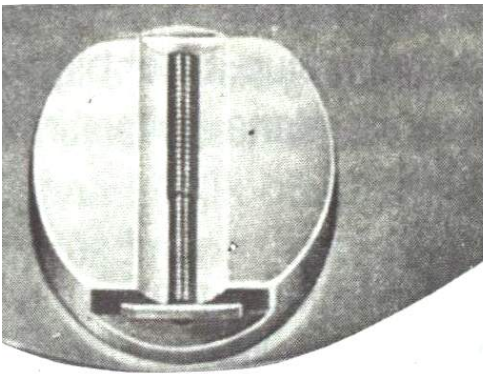
Modern röntgen tüpleri, sabit anotlu, döner anotlu ve iki fokuslu olmak üzere değişik şekillerde üretilmiştir. Resim 3.2’de döner anotlu bir röntgen tüpü görülmektedir.

Katot, filamanın etrafındaki metalden meydana gelmiş olup, filamanı ısıtacak akımı taşıma görevi yapar. Katot, tüpün negatif elektrodur. Filaman, ısıya dayanıklı bir metal olan Tungstenden yapılmıştır. Filaman transformatöründen gelen akım ısıtılan filaman, kendisini ısıtan akım şiddetiyle doğru orantılı olarak elektron yayar. Filaman ısıtıldıkça, filaman etrafında bir elektron bulutu oluşur. Filamandaki elektron bulutunun yoğunluğu anottan elde edilecek olan x-ışını demetinin de yoğunluğunu tayin eder. Yani, elde edilecek olan x-ışınlarının yoğunluk derecesini, filamanda oluşturulan elektronların miktarı belirler.



Resim 3.2. Döner anotlu bir röntgen tüpü.

Tungstenden yapılmış olan filaman spiral şeklinde olup, yaklaşık 2 mm çapında ve 1cm. uzunluğundadır. Resim 3.3'te spiral filamanlar görülmektedir. Çift odaklu modern röntgen cihazlarında, katotta iki filaman yan yana monte edilmiştir. Bunlardan büyük filaman yüksek doz değerlerinde, küçük filaman ise küçük doz değerlerinde devreye sokulur.



Resim 3.3. Spiral filamanlar

Filaman etrafında oluşturulacak olan elektronları, anot üzerinde küçük bir noktaya toplayabilmek ve elektronların anoda doğru hareketlerine yardımcı olmak üzere,

filamanın etrafında (-) yüklü Molibden'den yapılmış bir elektron yönlendirici levha bulunur.

Anot, katodun karşısında bulunan ve üzerinde x-ışını oluşturacak olan elektronların gönderildiği kısım olup, bakır bir levha ortasına gömülü Tungsten (Volfram) hedeften meydana gelmiştir. Anot, tüpün pozitif elektrotudur. Hedef olarak Tungsten'in seçilmesinin nedeni, kaynama noktasının çok yüksek (3400 °C) olmasıdır. İşte bu nedenle, bombardıman sırasında Tungsten atomları parçalanmaz ve hedef de ısı karşısında erimez.

### **3.6. X-Işınlarnın Elde Edilişı**

Röntgen tüplerinden x-ışını elde edebilmek için, filamanın akkor hâle getirilip elektron yayması ve bu elektronların hızla anottaki tungsten hedefe çarptırılması gerekir. Buna göre; röntgen tüplerinden x-ışını oluşturmada temel görevleri katottaki filaman, anottaki tungsten engel ve anoda uygulanan (+) yüksek voltaj yerine getirir. Röntgen tüplerinde katot (-), anot ise (+) elektrot durumundadır. Anoda uygulanan (+) yüksek voltaj nedeniyle katot ile anot arasında oluşan yüksek potansiyel fark, x-ışını oluşum aşamalarının başında yer alan önemli bir faktördür.

X-ışınlarını elde etmek için gereken elektronlar, filaman transformatörlerinden gelen akımla katottaki tungsten filamanı ısıtarak akkor hâle getirilmesiyle elde edilir. Katottaki tungsten elemanın elektron yayması, filaman transformatöründen gelen akımla doğru orantılıdır. Filamanı ısıtmakla, filamanı oluşturan tungsten (volfram) madeninin taşıdığı serbest elektronların enerjileri arttırılmış olur. Enerjileri arttırılan bu serbest elektronlar, maddeyi terk ederek filaman etrafında bir elektron bulutu oluştururlar. Filamanın ısıtılarak elektron yayması olayına “termoiyonik emisyon” denir.

Filaman etrafında oluşturulan elektronlar, filaman çevresinde bulunan, Molibden'den yapılmış negatif elektrik yüklü elektron yöneltici levha ile bir araya toplanıp, Coulomb Kanunu gereği anottaki hedefe doğru iletilirler. Ancak elektron yöneltici

levha tarafından gösterilen itme, elektronların anoda gidebilmeleri için yeterli değildir. Bunun için katotla anot arasında bir potansiyel fark yaratılmalıdır. Bu potansiyel fark yaratmak için, anoda (+) yüksek voltaj uygulanmalıdır. Bu da, ekspozur süresince anoda (+) yüksek voltaj uygulanarak sağlanır. Anoda uygulanan yüksek voltaj nedeniyle, katot ile anot arasında meydana getirilen gerilim, katottaki elektronların hızla anoda doğru ilerlemelerine olanak sağlar. Katot ile anot arasındaki potansiyel farktan dolayı, katottan anoda doğru ilerleyen elektronlar, hızlarıyla doğru orantılı olarak bir kinetik enerji kazanırlar. Katottan anoda doğru ilerleyen elektronların hızları, anoda uygulanan (+) voltajla doğru orantılı olduğundan, kazanacakları kinetik enerjiyi de anot voltajı belirler. Buna göre; tüpün anoduna uygulanan voltaj arttırıldıkça, katottan anoda doğru hareket eden elektronların hızı artmakta, hızları artan elektronlar ise daha çok kinetik enerji kazanmaktadır. Şu hâlde; katot ile anot arasındaki gerilim farkından dolayı katottan anoda doğru bir elektron akışı olmaktadır. İşte; röntgen tüplerinde katottan anoda doğru olan bu elektron akışına tüp akımı denir.

Tüpte katot ışınları anoda doğru elektron demeti şeklinde ilerlerler. Elektronların, katottan anoda doğru elektron demeti şeklinde ilerleyişi, katottaki elektron yönlendirici levha yardımıyla sağlanır. Bu elektron demeti ne kadar ince olursa, odak noktası da o kadar küçük olur.

Yukarıda belirtildiği gibi, katottan anoda doğru hızlandırılan elektronlar, hızlarıyla doğru orantılı bir kinetik enerjiye sahiptir.

$$E_k = \frac{1}{2} mV^2 \quad (3.1)$$

Kütleleri (m) sabit olan bu elektronlara kazandırılacak kinetik enerji ( $E_k$ ) ancak hızlarının arttırılması ile sağlanır. Bu da, kumanda masası kontrol panosundaki kilovolt seçici düğme yardımıyla tayin edilir.

Anoda uygulanan (+) yüksek voltaj nedeniyle, katottan ayrılıp, büyük bir hızla anoda ulaşan elektronlar, anottaki tungsten hedefin atomlarının (-) yüklü elektron alanıyla karşılaşılırlar. Katottan gelen elektronlarda (-) yüklü olduğundan ve Coulomb Kanuna

göre, aynı yükler birbirini iteceğinden, anotta karşılıklı aynı cins yüklerin çarpışması sonucu büyük bir elektron bombardımanı meydana gelir. Anottaki bu elektron bombardımanı katottan gelen elektronlardan % 99,8'inin hızlarının durdurulmasına ve taşıdıkları kinetik enerjinin ısı enerjisine dönüşmesine sebep olmaktadır. Zira enerjinin korunumu kanununa göre "Enerji yoktan var olamayacağı gibi, var olan bir enerji de yok olamaz. Ancak başka bir enerji türüne dönüşebilir." İşte katottan gelen yüksek kinetik enerjiye sahip elektronlardan, anotta durdurulanların kinetik enerjilerinin ısı enerjisine dönüşmesi bu esasa dayanır. Elektron bombardımanı sonucu anotta oluşan bu ısı, anottaki bakır iletkenle tüp dışına taşınır. Isının az bir kısmı tüp içinde kalır.

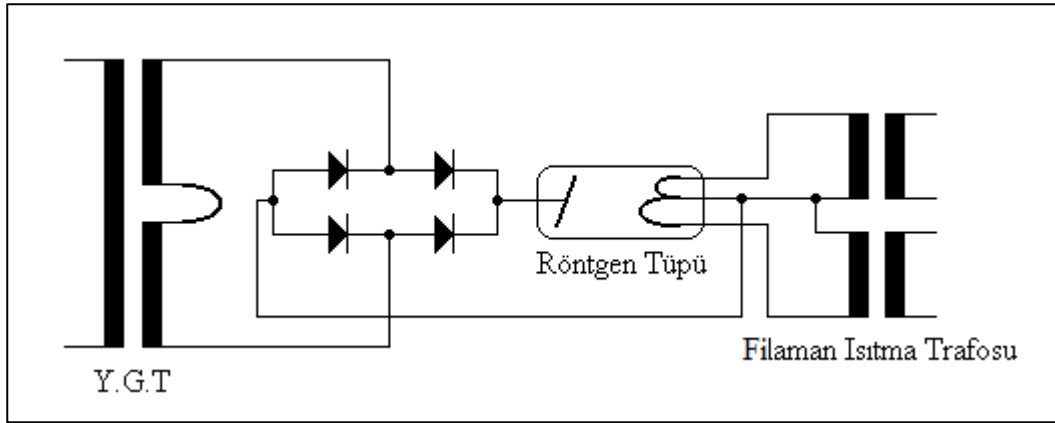
Katottan anoda gelen tüm elektronların %99,8'inin kinetik enerjisi anotta ısı enerjisine dönüşürken, geri kalan %0,2 oranındaki elektron demetçığının, hedefi oluşturan tungsten atomlarına çarpması ile, bu atomlarda meydana gelen olaylar sonucu beyaz radyasyon ve karakteristik radyasyon olmak üzere, farklı özelliklere sahip iki cins radyasyon oluşur.

### **3.7. Röntgen Cihazlarında Kalibrasyon**

Röntgen tüpünden geçen akımın şiddeti ve bu akımın uygulanma süresi, hastaya uygulanan dozu vermektedir. Dolayısıyla röntgen tüpünden geçen akımın şiddeti ve uygulanma süresi kritik bir öneme sahiptir. Bu akımın bir ölçü aleti ile ölçülmesi ve sık sık kalibre edilmesi gereklidir. Ancak yüksek gerilim trafosunun sekonderinde yüksek gerilim bulunduğundan bu noktaya ölçü aleti bağlamak tehlikelidir. Bu sebeple yüksek gerilim trafosu orta uçlu olarak üretilmiştir. Şekil 3.2'de yüksek gerilim devresi görülmektedir.

Yüksek gerilim trafosunun orta uçlu olarak ve bu uçlar arasında hiçbir sarım farkı olmayacak şekilde imal edilmişlerdir. Dolayısıyla bu uçlar arasında hiçbir gerilim farkı yoktur. Bu şekilde yüksek gerilimin oluşturduğu tehlike ortadan kaldırılmış olmaktadır. Normal çalışma esnasında bu nokta kısa devre edilir. Röntgen tüpünden geçen akım ölçüleceği zaman (röntgen cihazı kalibre edileceği zaman) bu kısa devre

kaldırılır. Bu noktaya ölçü aleti bağlanır. Böylelikle röntgen tüpünden geçen akım ölçülmüş olur.



Şekil 3.2. Yüksek gerilim devresi

#### 4. X-RAY GÖRÜNTÜLEME EKİPMANLARI

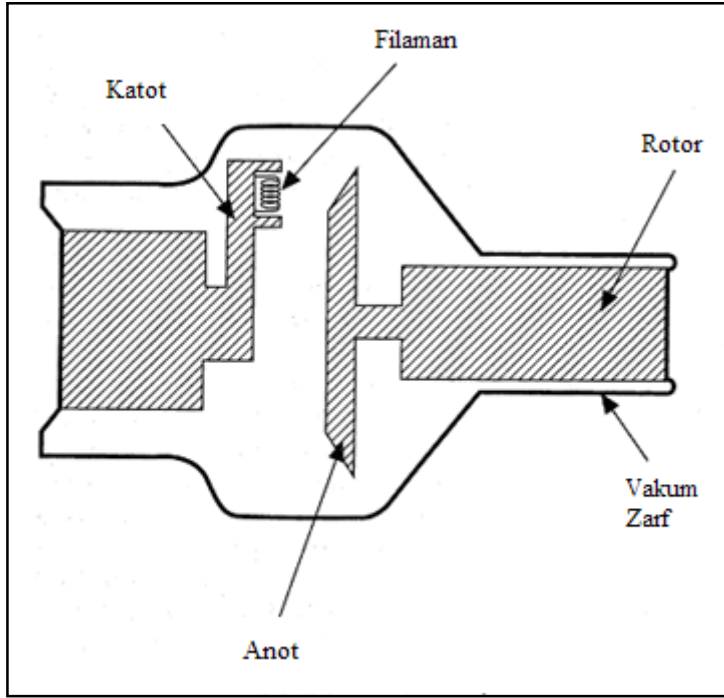
Klasik x ışını radyografisi dokudaki x ışını emilimine bağlı olarak oluşan gölgeli anatomik görüntüler üretir. X ışınları noktasal sayılabilecek bir alanda üretilerek görüntülenmesi istenen doku üzerine düşürülürler. Dokulardan geçen x ışınları her noktadaki x ışını yoğunluğuna bağlı olan bir parlaklık derecelendirmesi ile iki boyutlu bir görüntü oluşturacak şekilde kaydedilir. Görüntü üretimi, belirgin bir x ışını yayılımının vücut dokularından geçmesi ve farklı dokuların farklı miktarlarda x ışını emmesi gerçeğine dayanır. İlgilenilen anatomik bölgenin x ışınlarını kendisini çevreleyen dokulardan farklı bir oranda emmediği, dolayısıyla anlamlı bir görüntünün elde edilemediği durumlarda görüntü kontrastı güçlü x ışını emicilerinin ilgili dokuya zerk edilmesi ile artırılabilir. Örneğin, gastrointestinal bölgenin görüntülenmesi için baryum ile kontrast artırımına gidilmesi sık kullanılan bir metottur.

X Işınları genel olarak yaklaşık 1 ile birkaç yüz kiloelektronvolt enerjili elektromanyetik dalgalardır. Tıbbi görüntülemede kullanılan x ışınları ise tipik olarak 5-150 keV aralığındadır, uygulamada kullanılacak x ışınlarının enerjisi görüntülenecek anatomik yapının türü ve kalınlığına bağlı olarak ayarlanır.

Bir nesneye çarpan x ışınları direkt olarak içinden geçebileceği gibi nesne ile etkileşime de girebilir. Bu etkileşimler genellikle ya x ışınlarının emildiği fotoelektrik etkidir ya da x ışınlarının yanal sapmasıyla sonuçlanan saçılmadır. Saçılmaya uğrayan x ışınları küçük bir açıyla saparak görüntü algılayıcısına ulaşabilirler; bu durumda görüntü kontrastını azaltan ve bu nedenle de kaliteyi kötü etkileyen bir sebep olarak karşımıza çıkarlar. Bu kalite kaybı doku ve görüntü algılayıcısı arasında bir hava aralığı bırakılması ya da saçılma önleyen bir ızgara ile önlenabilir.

Sağlığa olan etkileri sebebiyle radyografik görüntülemede kullanılan doz mümkün olduğunca düşük tutulmaya çalışılır. Bununla birlikte x ışını kuantum gürültüsü, doz

azaldıkça daha belirginleşir. Bu gürültü, görüntü algılayıcısı üzerine düşen x ışınlarının miktarında kaçınılmaz bir rastlantısal değişim olması gerçeğine dayanır. Kuantum gürültüsü, görüntü algılayıcısına çarpan ortalama x ışını sayısı ile ilintilidir ve radyografik görüntüleme de görüntü kalitesine etki eden temel unsurdur.



Şekil 4.1. X-Ray tüpü

Konvansiyonel x ışını radyografisinde kullanılan temel teçhizat, işe yarar seviyede x ışını üretimi ve aktarılan x ışınlarının yüksek kalitede bir görüntü oluşturacak şekilde algılanmasını sağlamak durumundadır.

#### 4.1. X Işınlarının Üretilmesi

##### 4.1.1. X ışın tüpü

X ışını üretilmesinde kullanılan standart araç Şekil 4.1'de görülen döner anotlu bir x ışını tüpüdür. X ışınları vakum ortamda katottan anoda hızlandırılan elektronlardan üretilir. Elektronlar katot üzerinde bir olukta bulunan filamandan yayılır. Yayılım

filaman içerisinde onu ısıtan bir akım akıtıldığında gerçekleşir. Filaman yeterince kızdığında bazı elektronlar kendilerini filamanın metal yapısında tutan enerjiyi aşmalarına yetecek bir termal enerji kazanırlar. Bu elektronlar filamandan *buharlaştıklarında* katottan anoda doğru olan bir elektrik alan boyunca hızlanırlar. Bu gerilim bir yüksek gerilim üretici tarafından dışarıdan uygulanır. Elektronlar anoda doğru hızlanmışken birdenbire çok kısa bir mesafe içerisinde durdurulurlar. Elektronların enerjisinin büyük bir kısmı anodun ısınmasına sebep olurken küçük bir yüzdesi iki temel prensiple x ışınlarına dönüşürler. İlk metot yüklü bir parçacığın yavaşlamasının neden olduğu elektromanyetik ışın yayılımı, yaygın bilinen adıyla *Bremsstrahlung radiation*' dir. Bu etkiyle oluşan x ışınları, en çok elektronun anoda çarpmadan hemen önce sahip olduğu enerjiye eş toplam enerjiye sahip geniş ve sürekli bir enerji yayılımına sahiptirler. Yüksek enerjili x ışınlarının miktarı görece olarak azdır, düşük enerjilerde ise oldukça fazla sayıda x ışınması gerçekleşir.

X ışını oluşumundaki diğer metot hızlandırılmış bir elektronun anottaki bir atoma çarpması ve bu atomun orbitallerinden bir elektron koparması neticesinde görülen yayılımdır. Yörüngede boş kalan elektronun yeri komşu bir elektron tarafından doldurulur ve bu esnada enerjisi bu elektronun enerji değişimine eş bir x ışını yayılımı meydana gelebilir. Sonuç belli enerji seviyelerinde çok miktarda x ışını oluşumudur. Bu karakteristik x ışınlarının enerjisi anot yüzeyini kaplayan malzemeye özgüdür, böylelikle istenen enerjiye sahip x ışınlarının üretilmesi için farklı malzemede anot kaplamalarının seçilmesi yeterli olmaktadır. Örneğin 20 keV karakteristik x ışınımına sebep olduğu için mamografide kullanılan x ışını tüplerinin anotlarının kaplanmasında genellikle molibden kullanılır.

Düşük enerjili x ışınları hastanın aldığı dozu artırmalarına rağmen vücutta neredeyse tamamen soğurularak görüntü oluşumuna katkıda bulunmazlar, bu nedenle oluşmaları arzu edilmez. Belli enerji seviyesinin altındaki x ışını yayılımını önlemek için bunları emen bir filtre katman kullanılır. Durdurulan x ışınlarının enerji seviyeleri çıkıştaki x ışını huzmesinin yarı değer katmanı ile belirlenebilir.

Elde edilen görüntünün bulanık olmaması için noktasal bir kaynaktan x ışını üretimi tercih edilmektedir. Nicel olarak, bulanıklığın etkileri odak beneğinin sistem modülasyon transfer fonksiyonuna katkısı olarak tanımlanabilir. Bulanıklık kendini asıl olarak kenarlarda ve küçük nesnelere belli eder bunlar da yüksek frekanslara karşılık gelmektedir. Bu bulanıklık etkisi görüntülemenin geometrisine de bağlıdır ve nesneyle görüntü algılayıcısının arasındaki mesafenin artması sonucu daha da kötüleştirir.

Bulanıklığı önleyebilmek için elektronlar mutlaka anodun küçük bir noktasına çarptırılmalıdır. Odaklama katodun şekline bağlı olarak oluşan elektrik alanlar ile sağlanır. Bununla birlikte odaklamanın da bir sınırı vardır, eğer çok yüksek bir odaklama sağlanır ve tüm elektronlar anottaki bu küçük noktaya çarptırılırsa anot kaplaması eriyecektir. Bu kısıtlamadan, hedef anot noktasının sürekli olarak değiştiği ve bu sayede yeni ve daha soğuk bir noktaya çarpmanın sağlanabildiği dönen bir anot kullanılmasıyla kurtulunmuştur. Sınırlamayı daha da azaltabilmek adına anot açılı bir yüzey oluşturacak şekilde yapılır. Böylelikle algılayıcının gördüğü anot noktasının eğimli anot yüzeyi üzerindeki gerçek noktanın eğiminin sinüs açısı ile orantılı daha küçük bir nokta olması sağlanırken aynı zamanda sorun teşkil eden yüksek ısı enerjisinin nispeten daha geniş bir noktaya aktarılması başarılmaktadır. Fakat bu açı da x ışınlarıyla kaplanacak yüzeyin çok küçük olmaması için küçük belirlenemez. Pratikte, x ışını tüpleri genellikle iki ya da daha fazla farklı büyüklükte odak noktası barındırır, böylelikle duruma göre daha küçük bir odak noktasının (keskin görüntü, düşük güç) ya da daha büyük bir odak noktasının (daha bulanık, yüksek güçlü) seçilmesi mümkün olur.

Anodun büyüklüğü de ayrıca tek pozda üretilebilecek x ışını miktarını da belirleyen bir etmendir. Eğer toplamda çok fazla enerji aktarılacak olursa anot eriyecektir, bu kısıtlama da daha büyük boyutlu bir anot kullanılmasıyla aşılabilir.

#### 4.1.2. Gerilim kaynağı

X ışını tüpündeki gerilim ve akımlar bir x-ray jeneratörü tarafından sağlanır. Bu jeneratör katot-anot gerilimini kontrol eder, gerilim kısmen üretilen x ışını miktarını da etkiler çünkü x ışını miktarı gerilime bağlı olarak artış gösterir. Gerilim ayrıca üretilecek x ışınımının istenen enerji seviyesinde olmasını ayarlamaya yarar, daha yüksek gerilimlerde daha delici x ışınımı elde etmek olasıdır, bununla birlikte elde edilen görüntü daha düşük kontrasta sahip olacaktır. Gerilim üretici aynı zamanda x ışınımının miktarını katottan anoda akmakta olan akımı, akım süresini ayarlamak suretiyle ayarlayabilir. Bu noktada x ışınımı pozlamasını tanımlayan iki temel parametreyle karşılaşılır: maksimum kilovolt (poz esnasında anot-katot arasında oluşan maksimum potansiyel farkı) ve miliamper-saniye (miliamper cinsinden akım ve saniye cinsinden zamanın çarpımı).

Pozlama esnasında maksimum kilovolt ve miliamper-saniye değerleri röntgen teknisyeni tarafından manüel olarak görüntülenmek istenen dokunun gerektirdiği şekilde ayarlanabilir. Bazı üreteçler kilovolt ve miliamper değerlerinin elle girilmesi ile fakat poz süresinin üreteç tarafından otomatik ayarlanması esasıyla çalışır. Bu, x ışını yayılımının görüntü algılayıcısının öncesinde yahut sonrasında örneklenmesini gerektirir ve *poz zamanlaması* (phototiming) olarak adlandırılır.

Anot katot arasına tatbik edilmesi gereken yüksek gerilim (genellikle 15 ile 150 kV aralığındadır) 120 ya da 220 V alternatif gerilimi dönüştüren transformatörler aracılığıyla oluşturulur. Bu transformatörlerin çıkışı doğrultulur ve filtrelenir. Üç fazlı transformatörlerin kullanılması bir fazlı transformatörlere kıyasla neredeyse sabit gerilim üretilmesini sağlar, böylelikle düşük gerilim seviyesinde üretilmesi istenmeyen düşük enerjili x ışınımı yayılımı oluşması önlenmiş olur. Bu metotla birlikte transformatör çıkışı elektron tüpleri tarafından sabit bir gerilimle kontrol edilebilirler. Bu sayede pratik olarak sabit sayılabilecek seviyede gerilimler elde edilir ve dahası gerilim milisaniyeler mertebesinde zamanlaması ayarlanabilecek şekilde çok kısa zamanda uygulanıp kesilmesine olanak verecek şekilde anahtarlanabilir. Üçüncü bir metot olarak AC giriş bir DC gerilim oluşturacak

şekilde doğrultulur ve filtrelendir, sonrasında bir yarı iletken invertöre tatbik edilerek saniyede binlerce kez anahtarlanır. Bu yüksek frekanslı AC gerilimin daha yüksek bir gerilime bir transformatör tarafından dönüştürülmesi daha kolay olacaktır. Bu prensiple çalışan cihazlar orta-frekanslı yahut yüksek-frekanslı üreteçler olarak adlandırılırlar.

#### 4.1.3. Görüntü algılayıcılar: ekran – film kombinasyonları

Radyografik uygulamalarda görüntünün algılanması için özel gereksinimler mevcuttur. Bir çalışmada genellikle birkaç yüksek kalite görüntüden faydalanılır. Tanı süreci görüntülere hemen erişilmesini gerektirmediğinden radyografi sonucunun anında görünür hâle getirilmeye çalışılmasına gerek yoktur. Yine de bazı uygulamalarda (floroskopi) istisnai durumlar bulunmaktadır.

Bahsedilen şekilde bir radyografik görüntünün algılanmasında kullanılan en yaygın metot ışığa hassas bir negatif film kullanılmasıdır. Yüksek kaliteli filmler x ışınlarına pek hassas olmadığından mutlak surette x ışınına hassas ekranlarla birlikte kullanılmalıdırlar. Böyle ekranlar  $\text{CaWO}_2$  ya da fosfor ile birlikte nadir elementlerden bazılarının, örneğin uyarılmış  $\text{Gd}_2\text{O}_2\text{S}$  veya  $\text{LaOBr}$  birleşiminden mamuldürler. Film ışık geçirmeyen bir kaset içerisinde bir x-ray ekranla temas hâlinde yahut iki x-ray ekran arasında yerleştirilmiş durumdadır. Bir x-ray görüntüsü kasete ulaştığında, x ışınları yüksek oranda bu ekranlar tarafından emilir ve enerjileri görünür ışığa çevrilirler. Daha sonra bu ışık ekranla çok yakın konumlandırılmış film üzerinde negatif bir görüntü oluşturur.

Farklı filmlerin bağlı performanslarının tanımlanmasında önemli görülen bazı özellikler vardır. Bu özelliklerden kritik önemde olan bir tanesi *kontrast*'tır, filmin ekspozur<sup>1</sup> bölgesinin merkezi civarında çalışırken ilave ışığın sebep olduğu ilave kararmayı belirtir. Diğer bir özellik, filmin toleransı<sup>2</sup> ise, girişteki geniş aralıkta

---

<sup>1</sup> ing. exposure sözcüğünün dilimizde teknik terminolojide yaygın kullanım bulmuş karşılığı

<sup>2</sup> ing. latitude of Film

değişen ışık seviyesine göre filmin işe yarar bir görüntü oluşturabilmesi yeteneği olarak tanımlanır. Genelde filmin toleransı ve kontrastı karşıt özellikler olarak bilinir, geniş bir toleransa sahip olan film, düşük kontrastlı olacaktır. Buna ilaveten, bir filmin modülasyon transfer fonksiyonu<sup>1</sup> da önemli bir özelliktir. MTF yüksek frekanslarda azalır; bu yüksek frekans MTF'ü ayrıca filmin çözünürlüğüne, yani küçük nesnelere görüntüleyebilme yeteneğine de bağlı olan bir parametredir.

X-Ray ekranları ayrıca bazı önemli performans parametrelerine sahiptir. Ekranların kendilerine çarpmakta olan x-ray ışınlarının büyük yüzdesini algılaması esastır, bu da ekranın kuantum algılama etkinliği olarak ölçülen bir büyüklüktür. Hâlihazırda kullanılmakta olan ekranlar yüksek pik kilovolt seviyesindeki x ışınlarının %30'unu, düşük pik kilovolt seviyesindeki x ışınlarının %60'ı kadarını algılayabilmektedir. Algılamamanın bu seviyelerde olması etkin x-ray kullanımı için filmin her iki yüzünde birer ekran yerleştirilmesini gerektirmektedir. Küçük yapıların ve kenarların net görüntülenmesi için filmlerle birlikte, iyi bir yüksek frekans MTF da gereklidir. Işık ekrandan filme yol alırken bir miktar bulanıklaşmayla birlikte MTF azalması gerçekleşir. Bu etkinin ekran kalınlığının belirlenmesinde belirleyici bir rolü vardır; ekranların iyi bir kuantum algılama etkinliği gösterecek kadar kalın aynı zamanda aşırı bulanıklığın önüne geçecek ölçüde ince seçilmesi gerekir.

Ekran-Film sisteminde makul ölçüde bir karanlıklaşma sağlayabilmek için yeterli miktarda radyasyona ihtiyaç duyulur. Ekran-Film sisteminin az miktarda ışınla bir görüntü oluşturabilmesi, sistemin hızı olarak tanımlanır. Hız bazı başka özelliklerle ilişkilidir: ekranın kuantum algılama etkinliği, ekranın x-ray ışınmasını ışığa çevirmekteki etkinliği, ekranın yaydığı ışığın rengi ile filmin duyarlı olduğu renk aralığının uyumu ve belirli miktar ışık için ekranın kararma derecesi gibi. Radyografik bir görüntü oluşturmada gereken x-ray miktarı filme yeterli ekspozur sağlanabilmesi amacıyla uygun bir değerde belirlenir. Böylelikle, hastanın maruz kaldığı doz yüksek hızlı bir ekran-film kombinasyonu kullanılması ile azaltılabilmektedir. Fakat yüksek hızlı ekran-film kombinasyonunun kullanımı,

---

<sup>1</sup> ing. modulation transfer function

görüntünün oluşturulmasının az miktarda x ışınına bağlı olması yüzünden biraz gürültülü bir görüntü verir.

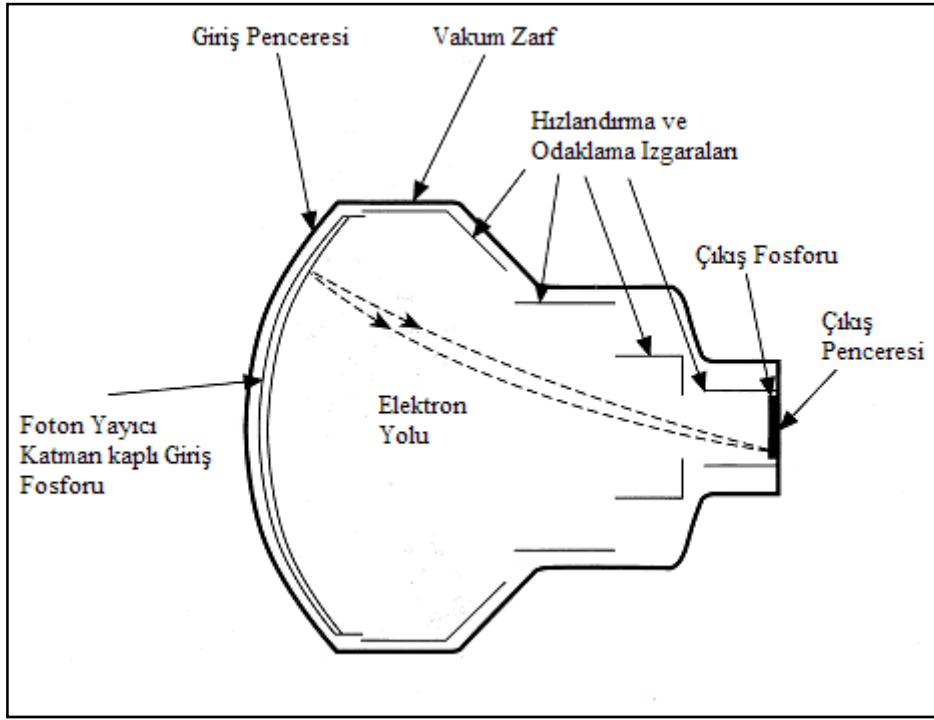
#### 4.1.4. Görüntü algılayıcılar: x-ray görüntü yoğunlaştırıcıları ve televizyonlar

Ekran-Film sistemleri radyografik görüntüleme için çok uygun olmalarına rağmen, floroskopik görüntüleme tekniğinde kullanılabilir değildirler. Floroskopide sürekli olarak düşük seviyeli bir x ışınımı söz konusudur ve devamlı surette pek çok görüntünün gerçek zamanlı elde edilmesi istenmektedir. Floroskopik görüntüler tanı koymak amacıyla olmasalar da örneğin anjiyografi esnasında kateterlerin kan damarlarına yerleştirilmelerine yardımcı olmaları için kullanılmaktadırlar. Floroskopide televizyon kameraları ve x-ışını görüntü kuvvetlendiricileri birlikte çalışırlar. X-ışını görüntü kuvvetlendiricisi x-ışını görüntüsünü algılar ve bunu görünür ışık bölgesinde küçük parlak bir görüntüye dönüştürür. Görünür ışık daha sonra nihai olarak bir monitörde görüntülenmek üzere lensler aracılığıyla bir televizyon kamerasına aktarılır.

Basit bir x-ray görüntü kuvvetlendiricisinin yapısı Şekil 4.2’de verilmiştir. Tüm bileşenler cam ve/veya metalden mamul vakum bir ortamda tutulmaktadır. X ışınları düşük absorpsiyonlu bir pencereden girmekte ve genellikle katkılanmış CsI’ tan yapılmış bir fosfor yüzeye çarptırılmaktadır. X-ışını ekranlarındakine benzer şekilde x-ışınları CsI yapısı içerisinde ışığa dönüştürülür. CsI katmanının üzerinde ışığı emen ve başlangıç itibarıyla farklı yönlere yayılmakta olan bir miktar düşük enerjili elektronlar yayan bir foto-emitör bulunmaktadır.

Oluşan foto-elektronlar üzerlerine uygulanan gerilimle kendilerini hızlandıran ızgaralar tarafından yönlendirilirler. Daha sonra çıkış katında bu elektronlar enerjilerini görünür bölgedeki ışıkla görüntüye dönüştürecek bir fosfor tabakaya daha çarptırılırlar. Bu ışık daha sonra bir mercek sisteminden geçerek bir çıkış penceresine yönlendirilir. Izgaralara tatbik edilen gerilim çıkıştaki görüntünün daha parlak olmasını sağlamak için elektronların hızlandırılmasını sağlamaktadır. Izgara gerilimleri ve biçimleri x-ışınının minimal bozulma ile görünür bölgedeki ışığa

dönüşmesini sağlayacak biçimde seçilir. Dahası, ızgaralar foto-emitör'den aldıkları ve hızlandırdıkları elektronları çıkıştaki fosfor tabaka üzerinde aynı noktaya yerleştirecek şekilde tasarlanmalıdırlar.



Şekil 4.2. X ışını kuvvetlendiricisi

Bir görüntü yoğunlaştırıcısında farklı görüntüleme bölgeleri oluşturmak üzere ızgara gerilimlerini ayarlamak mümkündür. Çıkıştaki fosfor tabakasında girişteki tüm alan görüntülenebileceği gibi küçük alt bölgelerinin çıkışa aktarılması olanağı da vardır. Küçük parçaların çıkışa aktarılarak görüntülenebilmesi seçeneği, vücut dokusunun küçük bölümlerinin büyük bir ekran üzerinde maksimum çözünürlükte görüntülenmesi istendiğinde yararlıdır. Örneğin 32 cm çapında bir alanı görüntüleyebilen bir görüntü yükselteci istenirse 16 cm ya da 12 cm çapta bir bölgeyi tamamen çıkıştaki fosfor tabakaya yöneltecek şekilde çalışabilir.

X-Işını görüntü yükselteçleri ekran/film kombinasyonlarından pek te farklı olmayan bazı performans parametrelerine sahiptirler. Burada da x-Işınlarının algılanması ve bunların yüksek bir etkinlikle kullanılması önemlidir, mevcut görüntü yükselteçleri

59 keV x-ışınları için %60 - %70 arası bir kuantum algılama etkinliği gösterirler. Ekran-Film kombinasyonlarında olduğu gibi, küçük nesnelere ve keskin kenarların bulanıklaşmadan görüntülenebilmesi için iyi bir yüksek frekans MTF gereklidir. Bunun yanı sıra, x-ışınları görüntü yükselteçleri içerisinde saçıldığı, ışık görelisi olarak uzun mesafeler kat ettiği ve foto-elektronlar sebebiyle, görüntü yükselteçlerinde düşük frekans MTF da özenle kontrol edilmelidir. Yükseltme derecesi, girişteki belirli miktar x-ışınının çıkıştaki görüntüde ne kadar parlaklaştığı ve büyüdüğü ile belirlenir. Bu kazanç<sup>1</sup> veya dönüştürme etkinliği<sup>2</sup> olarak tanımlanır. Bu esnada, çıkışta oluşturulan görüntünün daha çok foton ile daha parlak olmasının istendiği kadar küçük görüntülerle çalışmanın da çıkıştaki mercekle sistemleri açısından daha avantajlı olması yüzünden istendiği hatırdan çıkarılmamalıdır. Özellikle tüm giriş alanının görüntülenmesi söz konusu olduğunda görüntü yükselteçleri çıkıştaki görüntüde yastık sapması<sup>3</sup>'na sebep olurlar. Bu nedenle merkezden kaymış bir yere konumlanan kare bir nesne çıkışta merkezden dışa doğru uzamış şekilde görüntülenecektir.

Bir görüntü yükseltecinin çıkışının direkt olarak bir mercekle sistemiyle görüntülenebilmesine rağmen bir televizyon kamerası ve kameradaki görüntüyü veren bir monitörden oluşan sistem daha esnek imkânlarla sahip olur. Bu monitörler hâlihazırda yakalama tüpleri<sup>4</sup> yahut ccd<sup>5</sup> algılayıcılarla birlikte kullanılmaktadırlar.

Bir televizyon tüpü kullanıldığında görüntü tüpün girişinde yüklenmiş bir fotoiletken malzeme üzerine odaklanır. Bu fotoiletken yapıda  $SbS_3$ ,  $PbO$  ve  $SeTeAs$  gibi pek çok farklı madde kullanılabilir. Işıktan oluşan görüntü fotoiletken malzemenin belli bölgelerini deşarj etmek suretiyle görüntüyü fotoiletken katmanın arka yüzeyinde bir yük dağılımı hâline dönüştürür. Daha sonra yük dağılımı yüzey üzerindeki elektronların sebep olduğu küçük bir akı taranarak fotoiletken yeniden yüklenir. Yeniden yükleme akımı taranmakta olan noktanın ışık yoğunluğuyla direkt

---

<sup>1</sup> Standart bir x-ışını ekranına göre

<sup>2</sup>  $cd/m^2$  ya da  $mR/dak$  olarak ölçülen her bir x ışınımı girişinin çıkışta oluşturduğu ışık miktarı

<sup>3</sup> ing. pincushion distortion

<sup>4</sup> ing. pickup tube

<sup>5</sup> charge coupled device

bağıntılıdır, bu akım daha sonra yükseltilir ve monitör üzerinde bir görüntü oluşturmakta kullanılır. Tüpteki hedef bölge genelde satır atlamalı<sup>1</sup> teknikle taranır, bu sayede normal tv ekranları ile uyumlu olması ve standart cihazların kullanılabilmesi sağlanmış olur.

Floroskopide tüm ölçümlerde aynı seviyede doz kullanılması istenir, böylelikle görüntü gürültüsü yaklaşık olarak sabit seviyede tutulabilir. Bu genellikle görüntü yükseltecinin çıkışının orta bölgesinin görüntü parlaklığı gözlenerek sağlanır, çünkü parlaklık doz ile doğru orantılı artmaktadır. Parlaklık direkt olarak bir *foton çoğaltıcı tüp*<sup>2</sup> vasıtasıyla örneklenerek ya da televizyon ekranındaki sinyal seviyeleri analiz edilerek izlenebilir. Bununla birlikte sabit bir doz seviyesiyle çalışma isteği yüksek oranda emici dokularda çok yüksek doz miktarlarına sebep olacaktır. Bu hususla ilgili sorunları çözmek için sistemler genelde maksimum hasta dozajını yasal bir üst limitte sınırlandırmak zorundadır. Böyle durumlarda, doz limit değerine ulaşıldığında görüntü yükseltecin çıkışındaki görüntü karanlık olacaktır. Bunu dengeleyebilmek için televizyon sistemleri genelde ekrandaki görüntü parlaklığını sabit tutan bir otomatik kazanç kontrolü ile donatılırlar, bu sayede ekrandaki görüntü parlaklığı görüntü yükseltecinin çıkışına bağımlı olmaktan kurtarılmış olur.

#### 4.1.5. Görüntü algılama: sayısal sistemler

Hem radyografide hem de floroskopide sayısal görüntülerin kullanılmasının bariz avantajları vardır. Bu şekilde daha iyi görüntüleme için uygun görüntü işleme tekniklerinin kullanımı, bazı vakalarda daha düşük dozların kullanımının ve PACS<sup>3</sup> sistemleri ile uyumlu bir şekilde sayısal ortamlarda depolamanın önü açılmış olur. Buna ilaveten bazı sayısal sistemler görüntüleme aşamalarını azaltmaları, daha az bozulma ve gelişmiş tutarlılık sağlamaları sebebiyle daha iyi görüntü kalitesi sunarlar.

---

<sup>1</sup> ing. interlaced

<sup>2</sup> İng. Photo-multiplier tube

<sup>3</sup> Picture Archiving and Communication Systems, (Görüntü arşivleme ve iletişim sistemleri)

Medikal x ışını görüntülerini sayısallaştırmanın genel bir metodu bir görüntü yükselteci/tv sisteminin voltaj çıkışını kullanmaktır. Bu gerilim floroskopi veya radyografide gereksinim duyulacak yeterli dönüştürme hızlarına çıkabilen bir analog-sayısal dönüştürücü ile sayısallaştırılır.

Sayısal radyografik görüntüler elde etmenin bir başka tekniği ise fotonla uyarılabilir fosfor kullanımıdır. Bu teknikte x ışınları, x ışınımı enerjisini depolayan izole bir fosfor tabakaya çarptırılır. Daha sonra bu fosfor bir okuma ünitesine yollanır ve burada da fosfor yüzey uygun dalga boyundaki düşük şiddette bir ışık huzmesiyle taranır. Okunan fosfor tabakadaki bir noktanın depolamış olduğu enerji görünür ışık olarak yayılır ki bu da daha sonraki katmanlarda algılanacak, yükseltilecek sayısallaştırılacaktır. Bu tarz sistemler, fosfor tabakanın ebadının, ekran-film kombinasyonu sistemlerindeki ile aynı yapılabilmesi sayesinde hâlihazırda kullanılmakta olan ekran-film kombinasyonu metodunu kullanan cihazlara sonradan eklenebilmelerinin avantajını taşımaktadır.

Sayısal algılamadaki nispeten yeni bir teknolojide ise aktif matris ince film transistör yapı kullanılmaktadır. Her algılayıcı eleman yükün saklanması için bir elektrot barındırır, bu da x ışını işareti ile orantılıdır. Her elektrot kendisini sayısallaştırma devresi tarafından okunurken gerektiğinde devreye dâhil eden, gerektiğinde devreden izole eden bir transistörle kombine çalışır. Her bir elektrottaki yükü algılamının yaygın iki metodu vardır. Bunlardan birinde tipik olarak selenyumdan mamul bir x ışını emici tabaka algılayıcı dizisinin üzerine kaplanmıştır, bu tabaka kutuplandığında ve üzerinde x ışınları emildiğinde enerjileri elektron-delik çiftlerine dönüşür ve sonuçta oluşan yük elektrot üzerinde toplanır. İkinci metotta her elektrot, ışığa maruz kaldığında elektron-delik çiftleri oluşturan bir foto-diyotun parçasıdır, ışık CsI<sup>1</sup> gibi bir yapıya sahip sintilatör'de x ışınları tarafından oluşturulur.

Floroskopide sayısal sistemlerin kullanımı pek çok avantajı beraberinde getirir. Sayısal görüntü kenar iyileştirme, yumuşatma ya da medyan filtre algoritmaları

---

<sup>1</sup> Sezyum iyodit

uygulanmak suretiyle gerçek zamanlı olarak işlenebilir. Ayrıca her bir çerçeveden diğerine geçişte görüntünün gürültüsünü azaltabilmek adına hareketli görüntülerde bir miktar bulanıklaşmaya sebep olmasına rağmen ortalama algoritması tatbik edilebilir. Dahası, TV sistemleri ile birlikte kullanılan sayısal floroskopide televizyon tüpünün okunma işlemine uygun bir formatta taranması sağlanabilir, görüntü yine de ekran için optimize edilmiş farklı bir formatta da gösterilebilir. Bir başka avantaj ise ekrana aktarılan görüntünün x ışını pozlaması sona erdiğinde ekran üzerinde dondurulabilmesidir. Bu sayede radyoloji uzmanının elde ettiği son kare görüntüyü değerlendirmesi esnasında hastanın ilave doz alması engellenmiş olur.

Bazı sayısal sistemlerde darbeli floroskopinin de kullanımı olasıdır, bu teknikte x ışınları sürekli tatbik edilmek yerine kısa yoğun darbeler şeklinde uygulanmaktadır. Bu metotta x ışını atımları x ışını tüpünün filamanının kutuplanması ya da anot-katot geriliminin hızlıca anahtarlanması yoluyla oluşturulmaktadır. Buradaki asıl avantaj hareketli dokuların daha net görüntülenebilmesidir. Genelde her görüntü karesi için tek x ışını darbesinin kullanılması söz konusudur, ayrıca bazı görüntü karelerinde doz tatbik etmemek de mümkündür. Böylelikle her iki ya da dört karede bir uygulanmak suretiyle azaltılmış bir x ışınımı uygulanması ile hasta dozajının yarıya ya da dörtte birine düşürmek mümkündür. X ışınımının uygulanmadığı görüntü karelerinde elde edilmiş son güncel görüntü karesi tekrarlanarak gösterilir.

#### **4.1.6. X-ray projeksiyon anjiyografisi**

Anjiyografi temelde bir tanı aracı olarak kullanılmakla birlikte gitgide artan bir oranda dolaşım sistemi üzerinde tedavi amacıyla da yararlanılmaktadır. Ultrason, bilgisayarlı tomografi, manyetik rezonans görüntüleme ve anjiyoskopi gibi pek çok tanı aracı bulunsa da, klinik ya da araştırma amaçlarına yönelik olarak vasküler yapıların incelenmesinde projeksiyon anjiyografisinin yeri ayrıdır. Bu teknikte incelenmek istenen damar yapısı bir radyo opak kontrast ajanının enjekte edilmesiyle boyanır. Daha sonra boyar maddenin damarlardan akışı süresince bir dizi radyografik görüntü alınır. Bu işlem bir anjiyografi odası, bir özel prosedür laboratuvarı ya da bir kardiyak kateterizasyon laboratuvarında gerçekleştirilir.

Damar yapıların görüntülenmesinde kullanılan boyar madde, kan ve yumuşak dokunun radyografik kontrastının aynı olması sebebiyle bunların birbirlerinden ayırt edilmesine yarar. Boyar malzeme iyodin içeren ( $Z=53$ ) bir bileşimdir, maksimum iyodin konsantrasyonu  $350 \text{ mg/cm}^3$  seviyesinde hazırlanır. Bu terkip istenilen akış hızına bağlı olarak 1 ila 3 mm arasında çapa sahip bir kateter sayesinde enjekte edilir. Boyar maddeyle dolu damarlar film ya da video bantları üzerine kaydedilirler. Anjiyografik görüntülerin elde edilmesi ve saklanması sayısal görüntüleme sıklıkça kullanılan bir araçtır.

## 5. BLUETOOTH KABLOSUZ İLETİŞİM PROTOKOLÜ

### 5.1. Yapısı ve Çalışması

Bluetooth kablosuz kişisel alan ağları<sup>1</sup> için endüstriyel bir spesifikasyondur. Bluetooth protokolü mobil telefonlar, masaüstü ve taşınabilir özellikte kişisel bilgisayarlar, yazıcılar, sayısal fotoğraf makineleri ve video oyun konsolları gibi pek çok elektronik cihaz arasında güvenli ve tüm dünyada lisanssız kullanıma tahsis edilmiş kısa dalga radyo frekans bandında bir bağlantı ve veri alışverişi köprüsü kurar. Bluetooth spesifikasyonları “*Bluetooth Özel İlgi Grubu*” adı verilen uluslar arası bir oluşum tarafından belirlenmekte ve lisanslanmaktadır.

### 5.2. Kullanım Yerleri

Bluetooth daha çok düşük güç tüketen uygulamalarla birlikte kullanılmak üzere tasarlanmış standart hâline gelmiş bir iletişim protokolüdür. Kısa menzilli veri aktarımına olanak tanır, menzilini tanımlayan farklı güç sınıfları belirlenmiştir, bu güç sınıfları Çizelge 5.1’de görülebilir. Bluetooth söz konusu cihazların aynı iletişim protokolüne sahip başka cihazlarla iletişim menzili içine girmeleri durumunda birbirleriyle haberleşmelerini sağlar. Aygıtlar Bluetooth iletişim standardı radyo frekansını kullandığından diğerlerini direkt olarak görmek zorunda değildir. Bir başka odada olabilirler, aralarında herhangi bir engel olabilir.

Çizelge 5.1. Bluetooth güç ve menzil sınıfları

Sınıf	Maksimum İzin Verilen Güç (mW/dBm)	Menzil (Yaklaşık olarak)
Sınıf 1	100 mW (20 dBm)	~100 metre
Sınıf 2	2,5 mW (4 dBm)	~10 metre
Sınıf 3	1 mW (0 dBm)	~1 metre

<sup>1</sup> PAN (Personal Area Network)

Pek çok durumda *Sınıf 1* cihazların *Sınıf 2* cihazlarla iletişime geçtiklerinde *Sınıf 2* cihazların iletişim menzillerinin arttığı görülebilir. Bu artış *Sınıf 1* cihazın daha hassas alıcısı ve daha yüksek güçlü vericisinden kaynaklanmaktadır. *Sınıf 1* cihazın yüksek çıkış gücü *Sınıf 2* cihazın bu sinyalleri daha uzak mesafeden alabilmesini sağlar, aynı zamanda *Sınıf 1* cihazın hassas algılayıcısı *Sınıf 2* cihazın daha düşük güçlü vericisinden gelen sinyalleri daha uzak mesafeden hatasız olarak alabilmektedir. Bu sayede daha uzun bir mesafeden çift yönlü sorunsuz iletişim mümkün olabilmektedir. Orijinal Bluetooth spesifikasyonlarında tanımlananın ötesinde, çıkış gücü daha yüksek olan ve daha hassas bir alıcı ile donatılmış bazı sıra dışı ürünlerde *Sınıf 1* güç çıkış değerlerinin üzerine çıkılmadan 1 kilometre gibi aktarım mesafeleri elde edilebilmektedir. Çizelge 5.2’de Bluetooth protokolleri verilmiştir.

Çizelge 5.2. Bluetooth protokol versiyonları

Versiyon	Veri Oranı
<b>Versiyon 1.2</b>	1 Mbit /s
<b>Versiyon 2.0 + Gelişmiş Veri Oranı (EDR)</b>	3 Mbit /s
<b>WiMedia Alliance (tasarım aşamasında)</b>	53 – 480 Mbit /s

Bluetooth’u kullanabilmesi için bir aygıt mutlaka tanımlanmış protokollerden birine uymalıdır. Bu protokoller cihazların hangi tarz uygulamalar için geliştirilmiş olduklarına bağlı olarak belirlenir. Aşağıda liste hâlinde yaygın olarak karşılaşılan Bluetooth uygulamaları verilmiştir.

- Test cihazları, GPS alıcıları, medikal cihazlar, bar kod tarayıcıları ve trafik kontrol istasyonlarındaki geleneksel seri iletişim protokolünü kullanan kablolu iletişim için kablosuz bir alternatif olarak,
- Mobil telefon ve kulaklık ya da araç kiti arasında haberleşmenin kablosuz gerçekleştirilmesi amacıyla,
- Az veri bant genişliğinin yeterli geldiği durumlarda sabit bir alanda kısa mesafede bilgisayarlar arası ağ kurabilmek için,

- Fare, klavye ve yazıcı gibi bilgisayar giriş çıkış aygıtları ile bilgisayar arasında kablosuz iletişim sağlamak için,
- OBEX (Object Exchange) aygıtları arasında dosya aktarımı sağlamak için,
- OBEX aygıtları arasında kontak detayları, takvim randevuları ve hatırlatıcıların deęiş tokuş edilmesi amacıyla.
- Önceden kızılötesi iletişimin kullanıldığı kontrollerde kızılötesinin yerine,
- Bluetooth iletişimi aktif olarak kişilerin üzerlerinde taşıdıkları aygıtlara reklam panoları tarafından küçük reklamlar gönderilmesi amacıyla,
- Önde gelen oyun konsollarında kablosuz kumanda kollarının ana üniteyle bağlantısının sağlanmasında,
- Bir masaüstü ya da taşınabilir bilgisayarın veri erişimi sağlayabilen herhangi bir GSM telefonu aracılığıyla internet bağlantısını kullanabilmesi için,

Bluetooth iletişiminden yaygın olarak faydalanılmaktadır.

### 5.3. Ağ Altyapı Teknolojileri Olarak Bluetooth ve Wi-Fi Karşılaştırması

Bluetooth ve Wi-Fi<sup>1</sup> günümüz ofislerinde, evlerinde ve hareket hâlindeyken pek çok uygulamada kullanım bulmuşlardır. Bunlar ağ kurulumu, yazıcıdan çıktı almak ve sunum ve dosyaların PDA<sup>2</sup> ile Masaüstü bilgisayarlar arasında aktarılması gibi görevleri içerir. Her ikisi de lisanssız kullanılabilen yayılı izge<sup>3</sup> bandında frekanslar kullanmaktadırlar.

Bluetooth Wi-Fi'den daha az band genişliği sunması ve daha dar bir menzili kapsamı ile ayrılır. Fakat bu avantajlarının yanında Wi-Fi daha pahalı donanımlar ve yüksek güç gereksinimi gibi bazı handikaplar da taşımaktadır. Aynı frekans bandını paylaşmalarına rağmen farklı çoklama tekniklerinden faydalanırlar. Bluetooth pek çok küçük band genişliği gerektiren uygulamada direkt kablonun yerini alabilmekteyken, Wi-Fi direkt olarak yerel ağların kurulmasında

---

<sup>1</sup> Wireless Fidelity

<sup>2</sup> Personal Digital Assistant, Kişisel Sayısal Asistan

<sup>3</sup> Spread Spectrum

kullanılabilmektedir. Bluetooth sıklıkla kablosuz USB<sup>1</sup>, Wi-Fi ise kablosuz Ethernet olarak düşünülür, fakat ikisi de yerlerine geçtikleri kablolu sistemlerden daha az band genişliği vaat ederler. Bununla birlikte, bu benzetim tamamen doğru değildir çünkü herhangi bir Bluetooth aygıtı teorik olarak bir başka Bluetooth aygıtı ile bağlantısında host olabilmektedir. Böyle bir özellikten USB aygıtlarda söz edilemez. Bu nedenle detaylı bakıldığında aslında Bluetooth'un daha çok kablosuz FireWire<sup>2</sup> gibi çalıştığı görülebilir.

### 5.3.1. Bluetooth

Bluetooth, telefon, yazıcı, modem ve kulaklık seti gibi pek çok üründe kullanılmaktadır. Teknoloji düşük band genişliği ve yakın çalışan iki ya da daha fazla cihaz söz konusu olduğunda işe yaramaktadır. Bluetooth en sık kullanımını hücresel telefonlar ile kablosuz kulaklık setleri arasında ses aktarımı veya taşınır bilgisayarlarla aralarında veri aktarımı şeklinde bulmuştur. Bluetooth aygıtlar arasındaki sunulan servislerin bulunmasını ve kurulmalarını kolaylaştırmaktadır. Bluetooth cihazlar kullanıcı tarafından önceden tarafından izin verilmiş olan bütün servisleri sürekli yayarlar. Bu şekilde servislerin kullanımı daha kolay olur çünkü diğer ağ türlerinde olduğu gibi ağ adreslerinin ya da izinlerin ayarlanmasına gerek kalmamaktadır.

### 5.3.2. Wi-Fi

Wi-Fi daha çok klasik Ethernet ağlarına benzemektedir ve paylaşılan kaynakların, aktarılan dosyaların ve ses bağlantılarının önceden ayarlanmalarına ihtiyaç duyarlar. Bluetooth ile aynı radyo frekans bandını kullanmalarına ek olarak daha yüksek bir çıkış gücüne sahiptirler ve böylelikle daha kuvvetli ve stabil bir bağlantı kurmak mümkün olur. Wi-Fi sıklıkla kablosuz ethernet olarak anılır. Bu tanım genel olarak doğrudur ve ayrıca buna bağlı kuvvetli ve zayıf olduğu noktaları da anlatır. Wi-Fi

---

<sup>1</sup> Universal Serial Bus, Evrensel Seri Veriyolu

<sup>2</sup> Yüksek performanslı bir seri arabirim

daha çok ayar gerektirmektedir fakat daha hızlı bağlantı, daha geniş kapsama alanı ve Bluetooth'a kıyasla daha sıkı bir güvenlik sunmaları sayesinde büyük ölçekli ağların işletilmesine daha uygundur.

#### **5.4. Bilgisayar Gereksinimleri**

Bir kişisel bilgisayar bir başka Bluetooth aygıtı ile haberleşebilmek için mutlaka bir Bluetooth adaptörüne sahip olmalıdır. Bazı masaüstü bilgisayarları dâhilî Bluetooth adaptörlerine sahipken, bazıları haricî bir adaptöre ihtiyaç duyarlar. Pek çok güncel dizüstü bilgisayar dâhilî bir Bluetooth adaptörüyle birlikte satılmaktadır.

Her cihaz için ayrı bir adaptör gerektiren ve Bluetooth'un öncüsü sayılan IrDA<sup>1</sup>'nin aksine Bluetooth birden çok aygıtın tek bir bilgisayara aynı adaptör üzerinden eş zamanlı olarak bağlanmasına olanak tanır.

##### **5.4.1. İşletim sistemi desteği**

Apple firması Bluetooth'u 2002'de dağıtıma başlanan MacOS X 10.2 sürümünden beri desteklemektedir. Microsoft platformlarında ise Windows XP SP2 ve sonraki versiyonlar Bluetooth desteği sunmaktadır. Önceki versiyonlar üreticilerin sağladıkları sürücülerin kurulmasını gerektirmektedir. Linux tarafında ise iki çeşit Bluetooth desteği sağlanmaktadır, bunlardan biri BlueZ adındadır ve pek çok Linux çekirdeği ile birlikte dağıtılır. Orijinalde Qualcomm ve Affix tarafından geliştirilmiştir, tüm Bluetooth protokollerini ve katmanlarını destekler.

#### **5.5. Özellikleri**

Bluetooth spesifikasyonu 1994'te Ericsson Mobil için çalışan Jaap Haartsen ve Sven Mattison tarafından geliştirilmiştir. Temel olarak frekans sıçramalı geniş spektrum teknolojisinden faydalanır. Tanımlanan özellikler Mohd Syarifuddin tarafından

---

<sup>1</sup> IrDA, kızılötesi iletişim protokolü

yönetilen Bluetooth Özel İlgi Grubu tarafından standartlaştırılmıştır. Özel ilgi grubu<sup>1</sup> resmî olarak 20 Mayıs 1998’de duyurulmuştur. Güncel durumda dünya çapında 7000’in üzerinde firmayı kapsamaktadır. Ericsson, Sony Ericsson, IBM, Intel, Toshiba ve Nokia gibi alanlarında öncü firmalar tarafından kurulmuş ve daha sonra bünyesine pek çok firma katılmıştır.

### 5.5.1. Bluetooth 1.0 ve 1.0B

1.0 ve 1.0B sürümleri pek çok sorun içermektedir ve üreticiler uyumlu cihazlar yapmakta güçlük çektiler. Ayrıca bu sürümlerde zorunlu olarak bağlantı sürecinde donanımsal aygıt adresi aktarımı yapılması gerekmektedir, bu da protokol seviyesinde anonim kalmayı olanaksız kılıyordu.

### 5.5.2. Bluetooth 1.1

IEEE 802.15.1-2002 Standardı olarak onaylanmıştır. 1.0B sürümünde karşılaşılan pek çok hata giderilmiş ve şifrelenmemiş kanal desteği getirilmiştir. Sinyal seviyesini gösteren RSSI<sup>2</sup> desteği eklenmiştir.

### 5.5.3. Bluetooth 1.2

- Bu sürüm 1.1 sürümü ile geri uyumludur ve getirilen belli başlı yenilikler şöyle sıralanabilir.
- Daha hızlı aygıt tarama ve bağlantı kurma
- Sıçrama dizilerinde kalabalık frekansların kullanımının engellenmesiyle radyo frekansı girişiminin önüne geçen adaptif frekans sıçramalı geniş spektrum<sup>3</sup> kullanılmaya başlanmıştır
- 1.1 sürümünde de olduğu gibi pratikte daha yüksek veri aktarım hızları (721 kbit/s’ye kadar)

---

<sup>1</sup> SIG, Special Interest Group

<sup>2</sup> Received Signal Strength Indicator

<sup>3</sup> AFH, Adaptive Frequency Hopping Spread Spectrum

- Bozulan paketlerin tekrar aktarılmasıyla ses bağlantılarının kalitesini yükselten Genişletilmiş Senkron Bağlantı<sup>1</sup> algoritması
- Üç telli UART<sup>2</sup> için Host Kontrolcü Arayüzü<sup>3</sup> desteği
- IEEE 802.15.1-2005 Standardı olarak onaylanmıştır.

#### 5.5.4. Bluetooth 2.0

Bu sürüm, 10 Kasım 2004'te belirlenmiştir, sürüm 1.1 ile geri uyumludur. Getirdiği en büyük yenilik 3,0 Mbit/s veri aktarım oranını mümkün kılan EDR<sup>4</sup> standardıdır. Özellikleri şu şekilde sıralanabilir:

- Üç kat daha hızlı aktarım oranı, bazı özel hâllerde 10 kata kadar çıkabilmektedir (2,1 Mbit/s'ye kadar)
- Azaltılmış bir görev çevrimi sayesinde daha düşük güç tüketimi
- Çoklu bağlantı senaryolarının daha büyük veri band genişliği sayesinde sadeleştirilebilmesi

Pratikte etkin veri aktarım oranı 2,1 Mbit/s'ye kadar çıkmasına rağmen alt seviyeli sinyal haberleşmesi saniyede 3 Mbit olarak gerçekleşmektedir. Bluetooth üreticileri Bluetooth 2.0 spesifikasyonuna uygun cihazlar üretmelerine rağmen genellikle ayrıca belirtmemişlerse EDR kullanmamaktadırlar. Bu da bazı hataları giderilmiş 1.2 sürümündeki spesifikasyona denk düşmektedir.

#### 5.5.5. Bluetooth 2.1

Temel Bluetooth 2.1 Spesifikasyonu tamamen sürüm 1.1 ile geri uyumludur ve 26 Temmuz 2007'de Bluetooth SIG tarafından belirlenmiştir. Bu spesifikasyon şu özellikleri içerir:

---

<sup>1</sup> eSCO, Extended Synchronous Connections

<sup>2</sup> Universal Asynchronous Receiver Transmitter

<sup>3</sup> HCI, Host Controller Interface

<sup>4</sup> Enhanced Data Rate

- Geliştirilmiş sorgulama yanıtı
- Uykuda haberleşme oranı
- Şifreleme bekletme devam etme
- Güvenli basit eşleşme
- Yakın alan haberleşme (NFC) işbirliği

### **5.5.6. Bluetooth'un geleceği**

#### Yayın kanalı

Bluetooth bilgi noktalarının oluşturulması. Böylelikle bilgi noktalarından veri transfer eden kullanıcıların cep telefonlarına reklam göndermek mümkün olabilecektir.

#### Topoloji yönetimi

Günümüzde gittikçe daha yaygın hâle gelen piconet topolojilerinin otomatik olarak konfigüre edilebilmesi. Bu gelişim son kullanıcılar tarafından algılanamayacak fakat teknolojiyi bir adım ileri götürecektir.

#### Alternatif MAC ve PHY geçişi

Bluetooth profil bilgisinin aktarılması için alternatif MAC ve PHY'lerin kullanımı mümkün olacaktır. Bluetooth radyosu hâlâ aygıt bulunması, iletişimin başlatılması ve profillerin ayarlanması amacıyla kullanılırken, yüksek miktarlarda verinin aktarılması gerekirse yüksek hızlı alternatif MAC ve PHY kullanımı mümkün olacaktır. Bu da sistem bekleme konumundayken daha düşük güç tüketimi demektir.

## QoS geliřtirmeleri

Ses ve görüntünün daha yüksek kalitede aktarılmasını özellikle veri trafiğinin aynı piconet içerisinde aktarıldığı durumlarda en iyi performansın alınmasını mümkün kılacaktır.

### **5.5.7. Yüksek hızlı bluetooth**

28 Mart 2006'da SIG mevcut Bluetooth teknolojisine çoklu band orthagonal frekans bölmeli çokluma teknolojisini dâhil etme kararı almıştır. Bu sayede Bluetooth artık yüksek hızlı olabilecektir. Bu yeni sürüm ile yüksek kaliteli video ve ses uygulamaları gibi büyük miktarda verilerin taşınır cihazlardan, çoklu ortam projektörlerine ya da televizyonlara aktarılması ve senkronizasyonu mümkün olacaktır. Kablosuz VoIP görüşmeleri de gerçekleştirilebilecektir. Sürüm 2.1'den sonra kod adı Seattle olarak adlandırılan yeni versiyonda 480 Mbit/s gibi hızlara çıkılması öngörülmüştür. Bluetooth teknolojisinin geleceğın gereksinimlerini şimdiden karşılayacak özelliklerle donanması onun gelecekte daha yoğun olarak kablosuz iletişim sahasında kullanılacağı anlamına gelmektedir.

## 6. YAZILIM MİMARİSİ

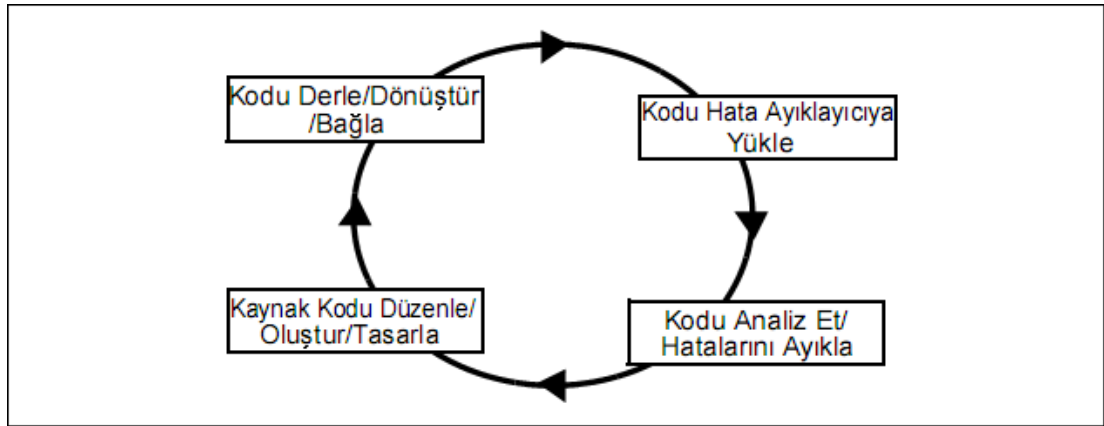
Projenin mimarisi temel olarak 3 ayağtan oluşmaktadır: elektronik donanım yani fiziksel devre kartı, dsPIC üzerinde çalışan C ile yazılmış gömülü kod ve PC üzerinde çalışan C# ile yazılmış arayüz programı. Elektronik kartın kod geliştirme süreci esnasında mikroElektronika firmasının üretmiş olduğu easydsPIC4 ürün geliştirme kartından, PCB ve devre şemalarının çizimi esnasında Labcenter firmasının ürünü olan Proteus yazılımından faydalanılmıştır. Gömülü C kodunun geliştirilmesinde Microchip firmasının yazılım geliştiricilere sunduğu MPLAB ve üzerine eklenen C30 derleyiciden yararlanılmıştır. Arayüz programı ise .Net kütüphanelerinden yararlanılarak Microsoft Visual Studio 2005 üzerinde C# dili ile yazılmıştır. Bu bölümde kod geliştirmede kullanılan araçlardan ve yazılan kodların çalışma prensiplerinden söz edilecektir.

### 6.1. Gömülü Kod İçin Geliştirme Araçları

Mikrodenetleyici kodu Microchip firmasının yazılım geliştiricilere ücretsiz dağıttığı MPLAB programının 7.60a sürümü üzerine kurulan C30 3.02 sürüm C derleyici ile geliştirilmiştir. Bu derleyici normalde assembly ile programlanan mikrodenetleyicilere kod geliştirmek için daha esnek çözümler sunmakta, bitmiş kod assembly eşdeğerine göre mikrodenetleyicinin program belleğinde biraz daha fazla yer tutmasına karşın ürün geliştirme sürecini kısaltmaktadır. Aynı zamanda bazı yazmaç değerleri ve giriş çıkış portlarının pinlerine dikkat edilmek kaydıyla C30 üzerinde yazılan herhangi bir kod bir başka mikrodenetleyici üzerinde çalışacak şekilde derlenebilmektedir. Assembly ile kodu bir mikrodenetleyiciden diğerine taşımak ise pek çok sorun çıkartmakta, çoğu zaman kodu yeni mikrodenetleyici için sıfırdan yazmak daha kolay olabilmektedir. Bu esneklik ve vaktin kısıtlı olması sebebiyle C30 üzerinde kod geliştirilmesi tercih edilmiştir.

### 6.1.1. Mplab ide

MPLAB IDE, herhangi bir IBM uyumlu bilgisayarı Microchip firmasının üretmiş olduğu mikrodenetleyiciler için bir yazılım geliştirme aracı hâline getirebilir. Tümüleşik geliştirme ortamı olarak anılmasının sebebi gömülü mikrodenetleyiciler için kod geliştirme sürecinin tamamını bünyesinde barındırmasıdır. Bir gömülü sistem kısaca masaüstü bilgisayarlarda kullanılan benzer bir *cpu*<sup>1</sup> ve *çevre birimi* olarak adlandırılan bazı aygıtlar barındıran komple bir sistemdir. Bu sistem kolaylıkla sayısal kontrol anlamında farklı işlevleri yerine getirebilmesi için değişik son ürünlere entegre edilebilir. MPLAB IDE, tasarım döngüsü olarak adlandırılan kod geliştirme sürecinin tüm adımlarını grafik bir arabirimden kontrol edilebilir şekilde koordine eden gelişmiş bir sistemdir.

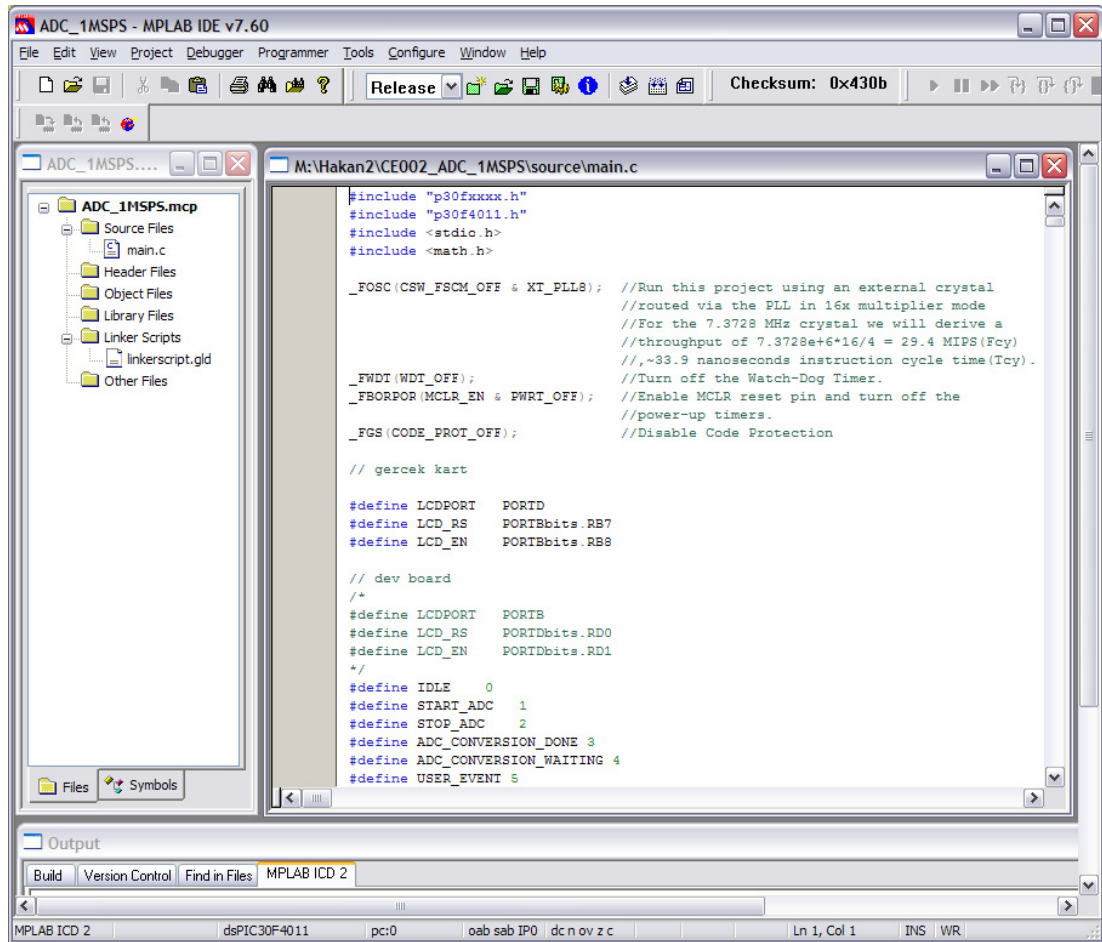


Şekil 6.1. Tasarım döngüsü

Kod yazıldığında IDE tarafından işletilebilir komutlara dönüştürülür ve nasıl çalıştığının görülmesi için mikrodenetleyiciye yüklenir. Bu süreçte birden çok araca ihtiyaç vardır: kodu yazmak için bir metin editörü, dosyaları ve ayarları yönetebilmek için bir proje yöneticisi, kaynak kodu makine koduna dönüştüren bir derleyici ya da dönüştürücü ve üzerinde çalışılan mikrodenetleyiciye kodu aktaran ya da mikrodenetleyicinin çalışmasını taklit eden bir öykünüm donanımı. Bir geliştirme

<sup>1</sup> Central processing unit, merkezi işlem birimi

ortamında kodun işletilmesi bir *hata ayıklayıcı*<sup>1</sup> üzerinde sınırlıdır. Hata ayıklayıcı mikrodenetleyicinin çalışmasını taklit eden bir yazılım olabileceği gibi, hedef donanım içinde çalışmasını analiz etmeye olanak veren bir donanım da olabilir. MPLAB IDE içine ilave donanım gerekmesizin hata ayıklama yapılabilmesi için simülatörler entegre edilmiştir. MPLAB IDE tümleşik olarak proje yöneticisi, kod editörü, dönüştürücü/bağlayıcı ve dil araçları, hata ayıklayıcı ve simülasyon motorları barındıran komple bir çözümdür. Şekil 6.2’de MPLAB IDE’ye ait kod geliştirme sürecinde alınmış bir ekran görüntüsü verilmiştir.

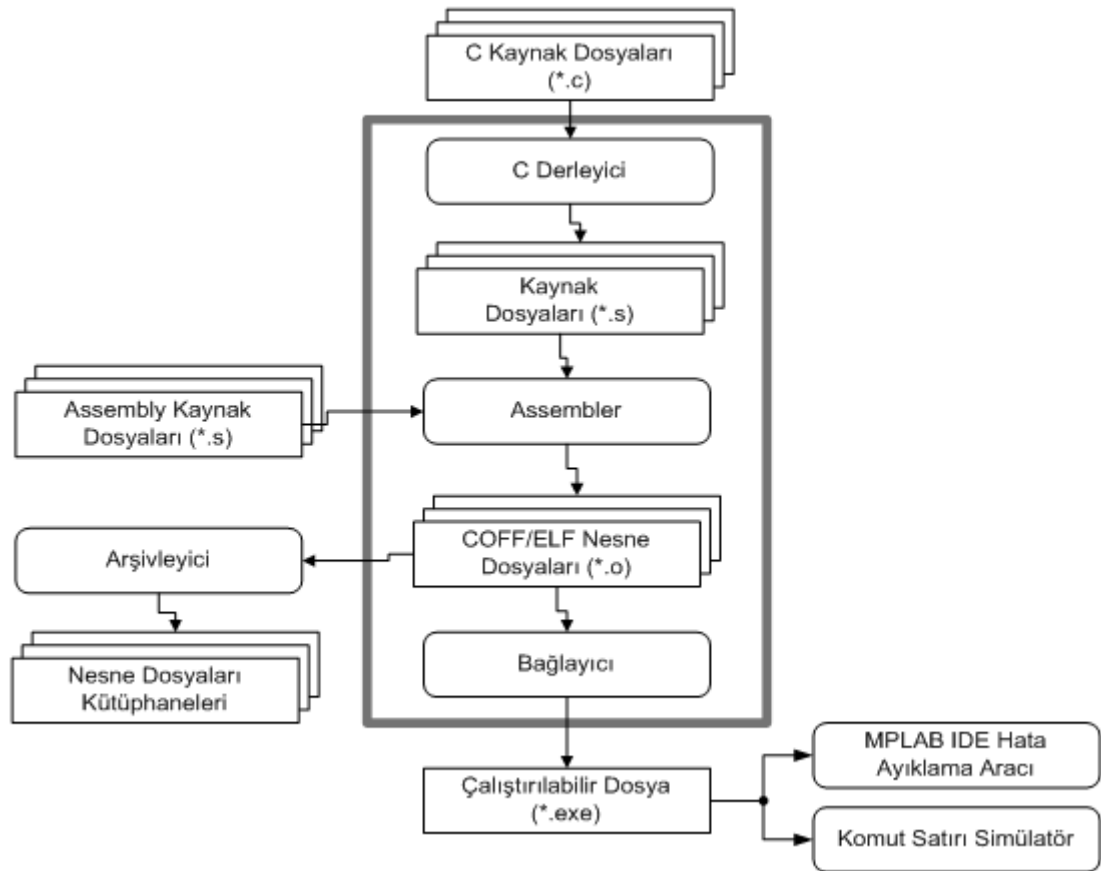


Şekil 6.2. MPLAB IDE ekran görüntüsü

<sup>1</sup> Debugger

### 6.1.2. C30 derleyici

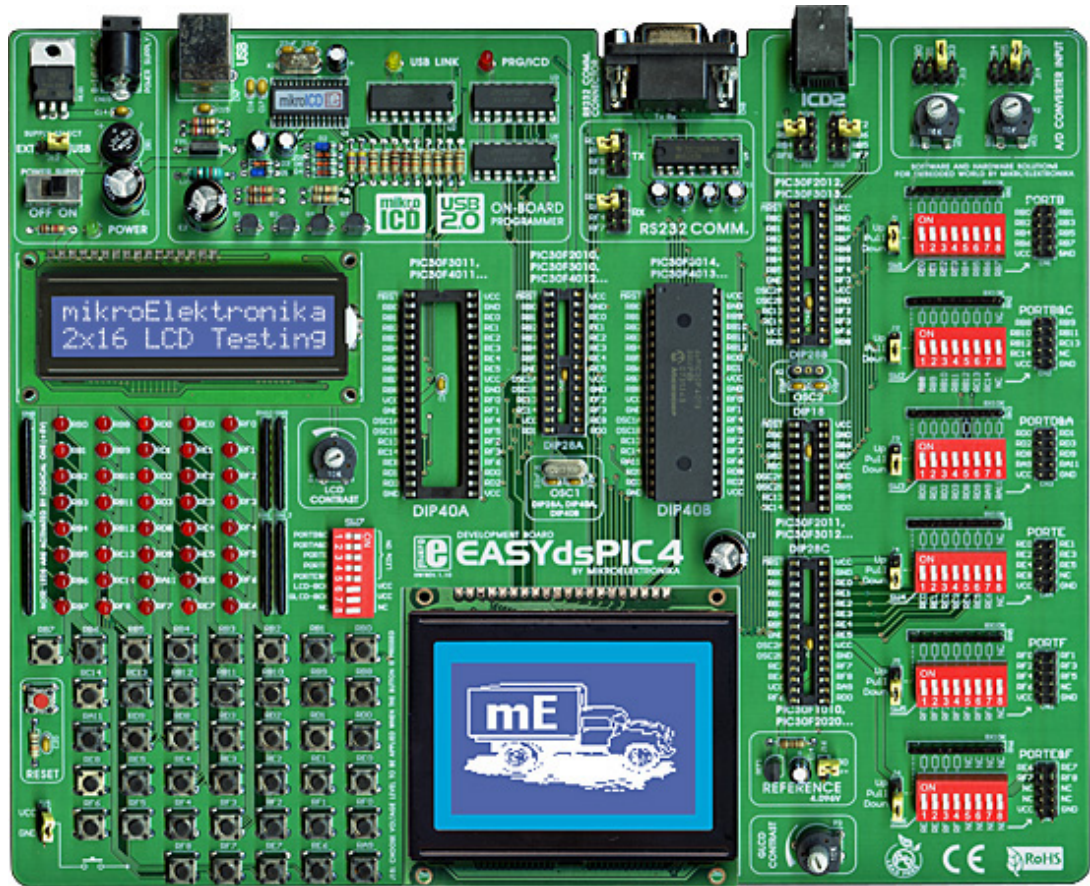
Microchip firmasının dsPIC mikrodenetleyici ailesi standart mikrodenetleyici fonksiyonları ile sayısal işaret işleme uygulamalarının yüksek performans gereksinimini bir araya getirmiştir. dsPIC sayısal işaret kontrolörleri üreticisi tarafından optimizasyon özellikli bir C derleyici, dönüştürücü, bağlayıcı ve arşivleyici yazılımlarla desteklenmiştir. MPLAB C30 bu bağlamda ANSI x3.159-1989 uyumlu, aynı zamanda dsPIC için ek komutlar da içeren optimizasyon özellikli bir C derleyicidir. Derleyici C kodu geliştirmek için hazırlanmış bir M.S. Windows konsol uygulamasıdır. MPLAB C30 C kaynak dosyalarını assembly diline dönüştürerek derler. Bu derleyici tarafından oluşturulan dosyalar nihai uygulamayı oluşturmak üzere diğer nesne dosyaları ve kütüphanelerle bağlantılır ve birleştirilir. Şekil 6.3'te MPLAB C30 ile yazılım geliştirme aşamaları gösterilmiştir.



Şekil 6.3. MPLAB C30 ile yazılım geliştirme aşamaları

### 6.1.3. EasydsPIC4 geliştirme kartı

Mikroelektronika firmasının üretmiş olduğu bu deneme kartı dsPIC serisi mikrodenetleyiciler için uygulama geliştirmeye yönelik olarak hazırlanan, üzerinde hem hızlı bir usb programlayıcı, hem donanımsal kod hata ayıklayıcı<sup>1</sup>, bunlara ilaveten deneme amaçlı pek çok elektronik devre elemanı barındıran bir platformdur.



Resim 6.1. EasyDSPIC4 programlama, hata ayıklama, ürün geliştirme kartı

Kart üzerinde sunulan diğer elemanlar ve bağlantı arabirimleri şu şekilde sayılabilir: Bir başka aygıtla ya da bilgisayarla bağlantı kurmaya yarayan RS232 seri iletişim arayüzü, mikrodenetleyici pinlerini pull-up ya da pull-down yapmaya yarayan dip-switch'ler, gerilim seviyesi, devreye alınacak fonksiyonlar gibi parametreleri dip-switchler ile ayarlayabilme, 18-28 ya da 40 pin DIP kılıfta mikrodenetleyiciler için

<sup>1</sup> In circuit debugger, ICD

ayrı soket, A/D dönüşümleri ve karşılaştırıcıları test etmek için 2 adet potansiyometre, besleme kaynağı olarak usb portu ya da harici adaptör kullanabilme, 4.096V haricî referans kaynağı, pinleri başka aygıtlara bağlayabilmek için çıkartılmış konnektörler, mikrodenetleyici pinlerinin durumunu izleyebilmek için 40 adet LED, programlama ihtiyacına göre istenilen adette kullanılmak üzere 41 adet buton, butonlara basıldığında ya da bırakıldığında gerilim harici ICD2 hata ayıklayıcı bağlantı soketi, çok hızlı USB 2.0 özellikli mikrodenetleyici programlayıcı, grafik LCD bağlamaya imkân veren soket, grafik LCD kontrastını ayarlamaya yarayan potansiyometre, 4 bit ya da 8 bit modda sürme imkânı sağlayan nokta matris LCD soketi, haricî besleme kaynağı olarak alternatif akım ya da doğru akım kaynağı kullanabilme, ihtiyaca göre değiştirilebilir kristal osilatör, reset devresi, 128x64 grafik LCD soketi.

Mikrodenetleyici kodu geliştirilirken kart üzerindeki potansiyometre ile A/D çevrim fonksiyonları çalışılmış, 2x16 LCD konnektörüne 4x20 LCD takılarak 4 bit modda sürülmesi için gerekli kod hazırlanmış, kontrol butonları ve kademe seçici komütatörün görevleri butonlara atanarak simüle edilmiştir. Seri iletişim için geliştirilen kod da PC ile araya bağlanan bir seri kablo ve masaüstü bilgisayarda çalıştırılan docklight isimli seri protokol analiz ve kaydetme yazılımı ile denenmiştir.

## **6.2. Mikrodenetleyici kodu**

MPLAB C30 üzerinde C dili kullanılarak geliştirilen kodda gerekli giriş çıkış portu tanımlamaları ve değişken atamaları yapılmakta, 4 bit modda LCD sürmek için gereken başlangıç parametreleri ayarlanmakta, ADC'nin ölçüm moduna girmesi ve hangi kanaldan örnekleme yapacağını belirlenmesi için teknik dokümanda belirtilen yazmaç değerleri tanımlanmaktadır. Yapılacak ölçüm işleminin gerçek zamanda hiçbir gecikme olmaksızın tamamlanması gerektiğinden, ana program yapısının tuşları ve kademe seçici komütatörün pozisyonunu döngü içerisinde okuması istenmemiştir. Tüm butonlar haricî kesme kaynağı olarak konfigüre edilmiştir. 10 MHz'lik kristal x8 PLL modunda mikrodenetleyiciyi 80MHz hızında çalıştırmaktadır. Kullanıcı arabirimi olarak görev yapan LCD ekran üzerindeki

menüler esnek bir şekilde tasarlanmış, her ekranda butonların farklı fonksiyonlar icra etmesi sağlanmıştır.

Programın ana fonksiyonları; saniye altı ekspozur sürelerinin söz konusu olduğu klasik röntgen, diş röntgeni, mamografi gibi görüntüleme cihazlarının kalibrasyonunun ve ölçümlerinin yapılabilmesini sağlamak üzere belirlenen “Röntgen” modu ve uzun görüntülemenin yapıldığı “Floroskopi” modu olmak üzere ikiye ayrılmıştır. mAs metre bu fonksiyon ekranlarına istenirse üzerindeki butonlar tarafından, istenirse bluetooth bağlantısı sayesinde uzaktan bağlanılarak getirilebilmektedir. Ölçüm öncesinde kademe seçici komütatör anahtarın hangi kademe olduğu bilgisi okunmakta ve daha sonra cihaz ekspozurun yapılmasını RC13 pininden gelen kesme ile algılamaktadır. Sinyalin varlığı esnasında RE8 pini lojik 1’e çekilerek kullanıcının bilgilendirilmesi sağlanmaktadır. Bu fonksiyon pratikte sadece uzun ölçümler esnasında floroskopi modunda anlamlıdır.

1 MSPS hıza kadar dönüştürme yapabilen ADC modülünün yaklaşık 600 ksps hızda örnekleme yapması oluşan veri trafiği açısından yeterli bulunmuştur. ADC’nin konfigüre edilmesi için ayarlanması gereken pek çok yazmaç bulunmaktadır, bunlardan hız ile ilgili olanların ayarlanmasına kısa bir örnek aşağıdaki gibi verilebilir.

Mikrodenetleyici 80MHz hızda çalıştırıldığından,

$$(80 \times 10^6)^{-1} = 12,5 \text{ ns},$$

mikrodenetleyici her fonksiyonu 4 saat darbesinde gerçekleştirdiği için

$$4 \times 12,5\text{ns} = 50\text{ns}$$

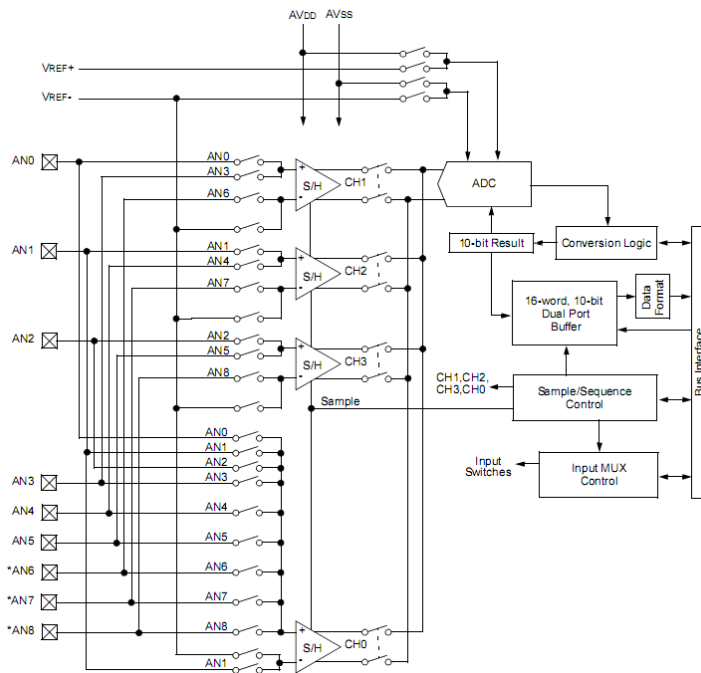
1 komutun yürütülmesi için geçen süre  $T_{CY} = 50 \text{ ns}$

$$\begin{aligned}
 ADCS < 5:0 > &= 2 \frac{T_{AD}}{T_{CY}} - 1 \\
 &= 2 \frac{154ns}{50ns} - 1 \\
 &= 5,16
 \end{aligned}$$

En yakın büyük tamsayı değeri olarak 6 alınır. Yani  $ADCS<5:0> = 6$  seçilir. Bu değerde ADC'nin gerçek dönüşüm hızı:

$$\begin{aligned}
 Gerçek T_{AD} &= \frac{T_{CY}}{2} (ADCS < 5:0 > + 1) \\
 &= \frac{50ns}{2} (6 + 1) \\
 &= 175ns
 \end{aligned}$$

1 dönüşüm işlemi 175 nanosaniyede yapılmakta yani aslında ADC'nin dâhilî dönüşüm saat hızı gerçekte  $(175 \times 10^{-9})^{-1}$  olmaktadır. Her 12 dönüşüm zamanında bir dönüşüm gerçekleşir.



Şekil 6.4. dsPIC30F4011 dâhilî ADC mimarisi

UART modülünde ise 57600 baud iletişim için 20MIPS işlem hızında baudrate generatör yazmacının değeri:

$$Baudrate = \frac{F_{CY}}{(16 * (BRG + 1))}$$

ifadesinden

$$57600 = \frac{20 * 10^6}{(16 * (BRG + 1))}$$

$BRG = 20,7$  bulunur.

Bir altındaki tam sayı değerine yuvarlatılarak  $BRG = 20$  alınır. Bu durumda gerçek baudrate 59523 bps olacaktır.

Seri porttan gerçek zamanlı olarak örneklenen verilerin tamamının aktarılması mümkün olmadığından ADC'nin dâhilî tamponlarından 15 tanesi bu iş için kullanılmış, bir döngü ile her 165 örneğin aritmetik ortalaması alınarak UART modülüne PC'ye yollanmak üzere gönderilmiştir. Bu işlemi gerçekleştiren kod parçası şu şekildedir:

Çizelge 6.1. ADC tampon belleği üzerinde hesaplama yapan ve her 165 örnekte bir hesaplanan ortalamayı seri porttan gönderen kod parçası

```

switch (STATE)
{
    case ADC_CONVERSION_DONE:
        STATE = ADC_CONVERSION_WAITING;
        //ADCBuffersRead = 0;
        tmp1 = 0;

        for(i = 0; i < ADCBUFFERSIZE; i++)
            tmp1 = tmp1 + ADResult[i];
        tmp1 = tmp1 / ADCBUFFERSIZE;
        aratoplam = aratoplam + tmp1;
        sayac1++;
        if(sayac1 == SAYAC1_SIZE)
        {
            sayac1 = 0;
            araortalama = aratoplam / SAYAC1_SIZE;
            TXBuffer[0] = araortalama;
        }
    }
}

```

Çizelge 6.1. (Devam) ADC tampon belleği üzerinde hesaplama yapan ve her 165 örnekte bir hesaplanan ortalamayı seri porttan gönderen kod parçası

```

TXBuffer[1] = araortalama >> 8;
TXBufferFull = 1;
geneltoplam = geneltoplam + araortalama;
sayac2++;
aratoplamlam = 0;

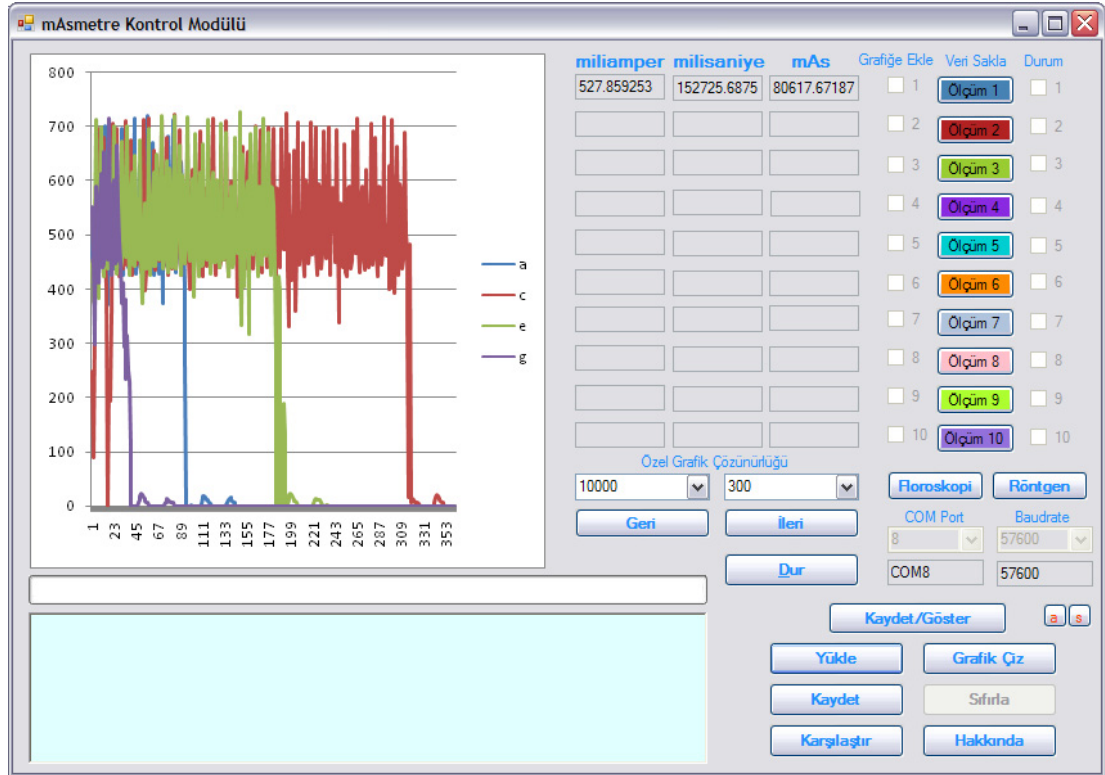
U1TXREG = TXBuffer[0];
U1TXREG = TXBuffer[1];
    }
break;
}

```

dsPIC üzerinde koşan C kodunun akış diyagramı detaylı olarak EK-21’de verilmiştir.

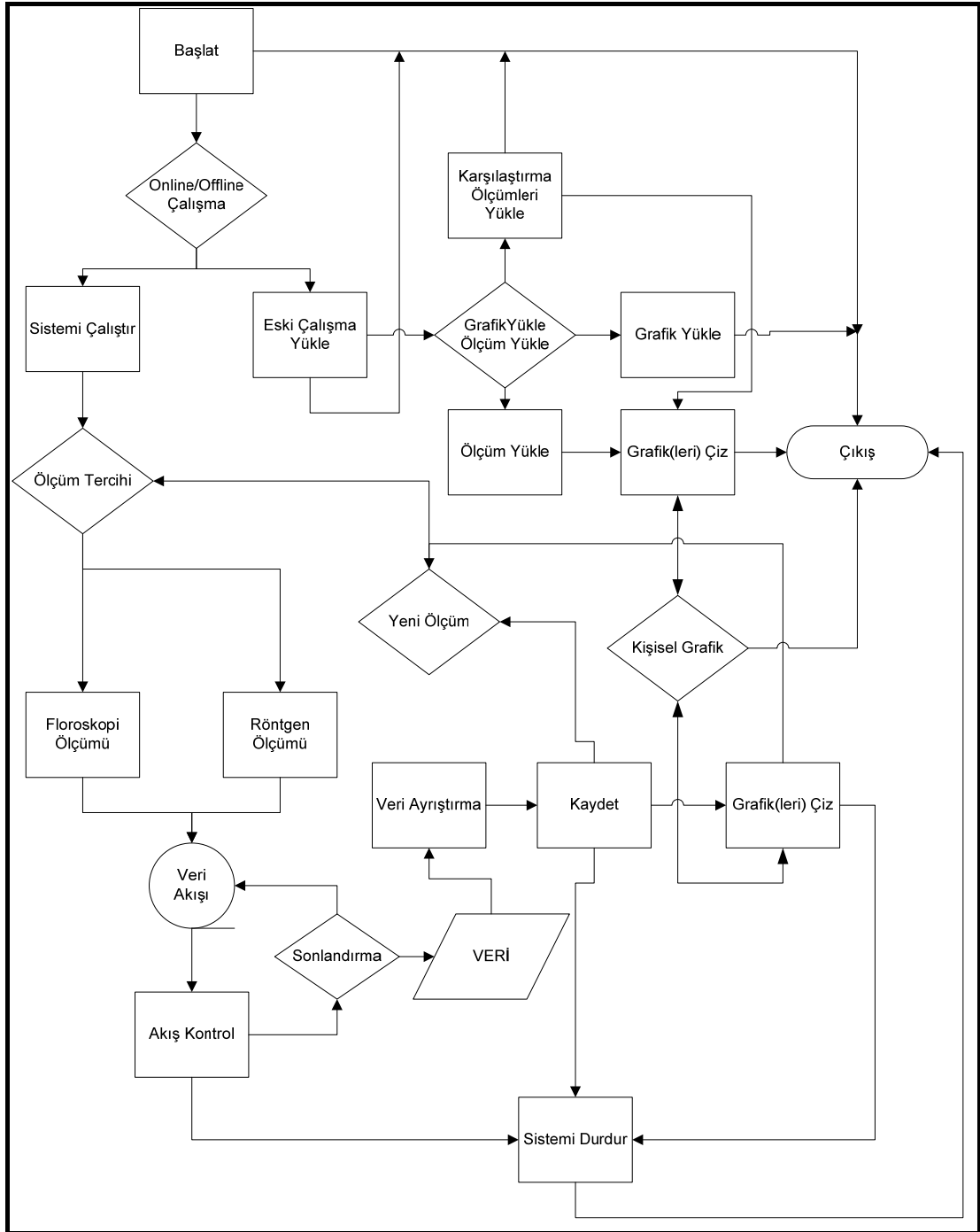
### 6.3. Bilgisayar Kontrol Arayüzü

Tasarlanan kalibrasyon cihazının en önemli yeniliklerinden birisi aldığı ölçüm verilerini gerçek zamanlı olarak bilgisayara aktarabilmesi, daha sonra bunları işleyip istenildiği şekilde ve çözünürlükte karşılaştırmalı veri-zaman grafiklerinin çizilmesine olanak sunmasıdır. Ölçüm işlemi bilgisayar arayüzü vasıtasıyla uzaktan da kontrol edilebilmektedir. Bu amaç için geliştirilen yazılımın temel özellikleri gerçek zamanda çalışabilecek kadar optimize bir seri port iletişimi kurabilmesi, kontrol butonları ve grafik çizme fonksiyonlarıdır. .Net kütüphanesinin sunduğu hazır fonksiyonlarından faydalanılabileceği düşünülerek yazılım Visual Studio 2005 ortamında C# programlama dili ile geliştirilmiştir. Şekil 6.5’te yazılımın ekran görüntüsü verilmiştir.



Şekil 6.5. Önceden kaydedilen değerler arasından çağrılmış dört ölçüm sonucunun karşılaştırmalı grafiğinin bulunduğu ekran görüntüsü

Bir sonraki sayfada Şekil 6.6'da da kontrol arayüzü yazılımının akış diyagramı görülebilir. Yazılımın kaynak kodu da EK-22'de verilmiştir.



Şekil 6.6. mAs metre kontrol programının genel akış diyagramı

## 7. DONANIM MİMARİSİ

X ışını tüpü, klasik röntgen, floroskopi, mamografi, diş röntgeni ve bilgisayarlı tomografi gibi pek çok tıbbi görüntüleme cihazının temelinde yer almaktadır. X ışınlarının dokulardan geçme özelliği ve bu geçiş esnasında x ışınlarını farklı dokuların farklı oranlarda absorbe etmeleri sayesinde non-invasive olarak görüntüleme yapılması mümkün olabilmektedir. Daha önceki bölümlerde de açıklandığı üzere x ışın tüpünün ürettiği ışın hücre yapısını bozucu özellik göstermekte ve görüntüleme ancak belirli aralıklarda izin verilen miktarda x ışını dozajları ile yapılabilmektedir. Bunun üzerindeki doza maruz kalan canlılarda x ışınları iyonlaştırıcı etki göstererek hücre içi kimyasal bağları kırmakta ve yeni bağlar kurulmasına sebep olarak ölümcül etkileri görülebileceği gibi, hafif doz aşımalarında kaşıntı, tahriş, alerjik reaksiyonlar şeklinde de kendini gösterebilir. Gerek tıbbi görüntüleme teknisyeni gerek hasta sağlığı açısından röntgen tüpünün verdiği dozajın istenen ölçüde olması şarttır. Bunu sağlamak için belirli aralıklarda tıbbi görüntüleme cihazlarının akredite kuruluşlar tarafından kalibrasyonu yapılmakta, bunun yanı sıra işletme esnasında da sık sık teknisyenler tarafından çalışma parametreleri kontrol edilmektedir. Tüpten yayılan x ışınmasını belirleyen başlıca üç parametre bulunmaktadır. Bunlar ışınmanın yoğunluğunu belirleyen akım<sup>1</sup>, deliciliğini belirleyen gerilim<sup>2</sup> ve bunların tatbik edilme miktarını belirleyen zaman<sup>3</sup>'dir. Bu tez çalışması kapsamında akım ve zaman değişkenlerinin ölçülmesi ve bunların sonucunda dozajı belirleyen mAs<sup>4</sup> büyüklüğünün hesaplanması gerçekleştirilmiştir. Ayrıca kullanım kolaylığı sağlaması, kalibrasyonu yapan teknisyenin sağlığını korumaya yardımcı olması amacıyla yönelik olarak ölçülen verilerin gerçek zamanlı olarak uzaktaki bir bilgisayara bluetooth kablosuz iletişim protokolü ile güvenli ve doğru olarak aktarılması sağlanmış, ayrıca cihazın uzaktan kumanda edilebilmesi de başarılmıştır. Gelişmiş analizlere olanak sağlayacak şekilde verilerin birbirleriyle karşılaştırılması, grafiklerle mukayese edilmesi ve daha ileri

---

<sup>1</sup> miliamper cinsinden

<sup>2</sup> KvP (kilovolt peak) cinsinden

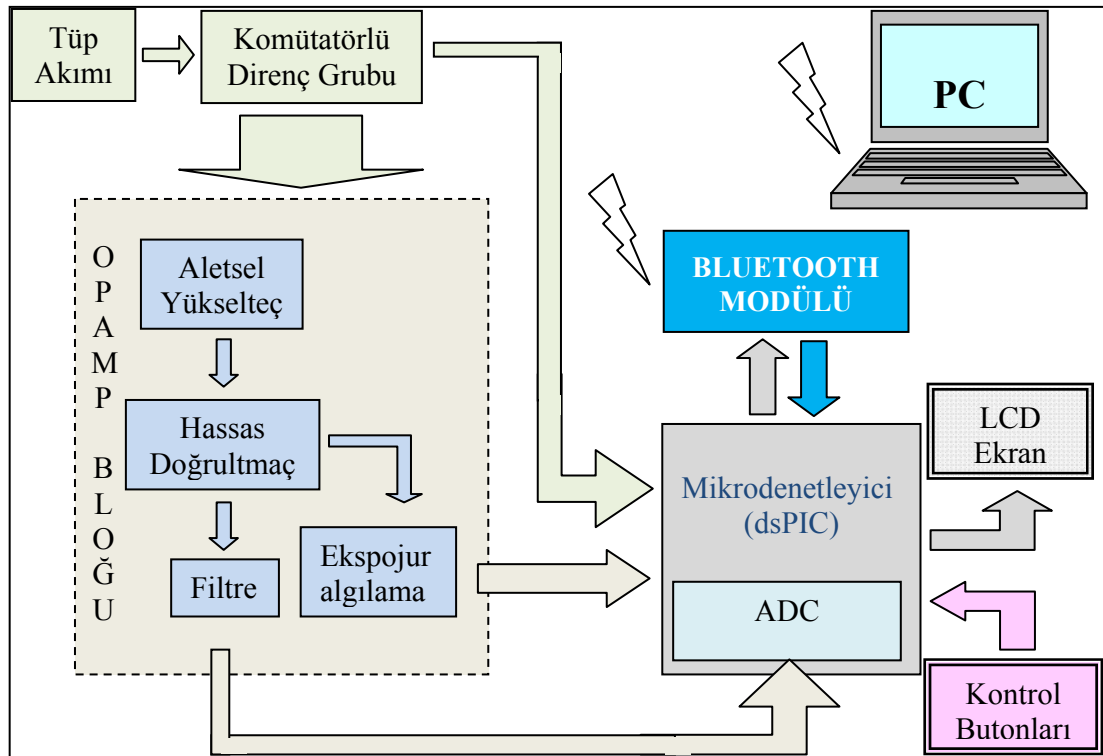
<sup>3</sup> milisaniye cinsinden

<sup>4</sup> miliampersaniye

arařtırmalarda kullanılması amacıyla saklanmasını saęlayan bir bilgisayar kontrol arayüzü de geliştirilmiştir. Cihaz kolay kullanıma olanak veren 4 satır 20 kolon bir nokta matris lcd<sup>1</sup> ve kullanıcı girişine imkân saęlayan ve fonksiyonları dinamik olarak içinde bulunulan menü yapısına göre deęişen 3 adet butonla donatılmıştır. Direkt son ürün olarak tüm fonksiyonları ile kullanılabilir vaziyette imal edilmiştir.

### 7.1. Blok Diyagramı

Geniş anlamıyla düşünülecek olursa cihazın devre blokları Şekil 7.1’de gösterildięi gibi özetlenebilir. Yüksek gerilim trafosunun sekonder sargısının arasındaki yola girilerek alınan akım 6 kademeli bir direnç grubu ile istenen ölçeęe ayarlanır. Şayet tek bir direnç üzerinden akan akıma göre ölçüm yapılsa idi mAs metre’nin ölçmesi istenen tüm deęerlerde aynı doğruluk ve hassasiyeti göstermesi mümkün olmayacaktı.



Şekil 7.1. mAs metre donanımsal blok diyagramı

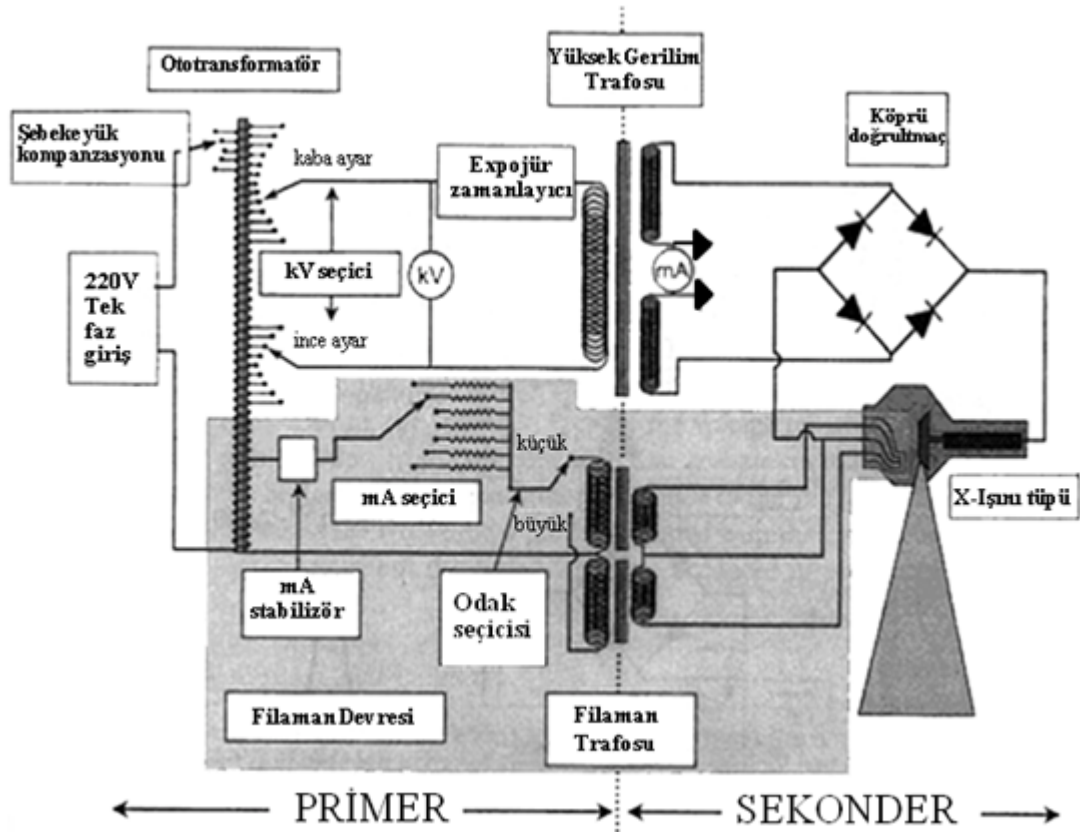
<sup>1</sup> liquid crystal display

Bu metotta ise düşük akım değerlerinden (20 mA) başlayarak 4000 mA değere kadar ölçümlerin sağlıklı yapılabilmesi için ölçeklendirme yapılmaktadır. Komütatör yardımıyla istenen aralığa ayarlanan sinyal buradan işlemsel yükselteç (opamp) bloklarına aktarılır. Burada önce sinyal belli bir kazançla kuvvetlendirilmekte, daha sonra yüksek gerilim trafosunun doğası gereği alternatif formdaki akım bir hassas doğrultmaç bloğu ile doğrultulmaktadır. Buradan gürültülerden arındırılmak üzere filtre bloğuna gönderilen sinyal aynı zamanda mikrodenetleyicinin ölçüm esnasında akımın başladığını ve bittiğini, yani ekspozür süresini anlayabilmesi için bir devre yardımıyla lojik seviyeli bir kontrol sinyaline dönüştürülmektedir. Filtrelenmiş ve mikrodenetleyici bünyesindeki ADC'nin kabul edeceği aralığa ölçeklenmiş pozitif değerlikli sinyal mikrodenetleyiciye iletilmektedir. Mikrodenetleyici sinyalin başladığını ve bittiğini kendisi algılayarak bu süre içinde ADC'yi örnekleme sürecine sokar ve sinyali gerçek zamanlı olarak işler. Kontrol butonları ve LCD ekran operatörün istenen ölçüm fonksiyonunu seçebilmesi ve hesaplama sonuçlarını görebilmesi için gelişmiş bir kontrol arayüzü sunmaktadır. Ölçüm esnasında eğer bilgisayar ile kontrol ve veri kaydetme olanağı kullanılmak istenirse bluetooth arabirimine sahip herhangi bir bilgisayar kullanılabilir. Bilgisayarda çalışan ve yazılım kısmında anlatılan kontrol programı verinin örneklenmesi yani ölçüm esnasında gerçek zamanlı olarak verileri almakta ve bunları kaydetmektedir. Bluetooth arayüzü verilerin bilgisayarda kaydedilip grafiklerle yorumlanmasını sağlama fonksiyonunun yanında mAs metre'nin uzaktan kontrol edilebilmesine ve kalibre edilecek olan x ışını görüntüleme sisteminden farklı gerilim ve akım değerlerinde ve değişik sürelerle istenildiği kadar ekspozür alınarak tüm bunların uzaktan yönetilebilmesine olanak vermektedir.

## 7.2. Ölçüm Prensibi

Şekil 7.2'de görüldüğü gibi şebekeden gelen besleme gerilimi bir ototransformatör üzerinde değişik kademelerde ayarlanmakta, yapılacak görüntülemenin niteliğine uygun mA değeri de akım sınırlandırıcı bir düzenek yardımıyla ayarlanarak filaman trafosuna verilmektedir. Daha sonra ayarlanan bu gerilim sabit dönüştürme oranına sahip yüksek gerilim trafosu ile tüp gerilimine yükseltilmektedir. Diyot grubuyla

doğrultulan yüksek gerilim tüpe tatbik edildiğinde filaman trafosu tarafından belirlenen değerle orantılı olarak bir x ışması oluşmaktadır.



Şekil 7.2. Yüksek gerilim trafosunun primer ve sekonder kısımlarının blokları

Röntgen tüpünden geçen akımı ölçebilmek için tüp beslemesini sağlayan yüksek gerilim transformatörünün sekonder sargısının ortasından tıbbi görüntüleme cihazının imalatı esnasında bir orta uç çıkartılmıştır. Normal kullanım esnasında bu uçlar kısa devre edilmekte, ancak ölçüm yapılacağına buradan proplar yardımıyla uçlar çıkartılmaktadır. Dolayısıyla akımı ölçebilmek için akım yoluna seri bağlantı yapma imkânı üretim aşamasında düşünülmüştür. Bu akım yolu üzerine bağlanan ve adına mAs metre denen ölçü aletleri ile tüp akımını ölçmek mümkün olmaktadır. Gerçekte mAs metre miliamper mertebesinde akımları ölçen yüksek hassasiyetli bir ampermetre olarak düşünülebilir. Farkı ise ekspozür süresinin kısalığı göz önüne alındığında anlaşılmaktadır. Normal ampermetreler akım değerini gösterebilmek için bir reaksiyon zamanına ihtiyaç duyarlar. Ve bu yüzden yüksek gerilim trafosunun

ortasına bağlanacak herhangi bir ampermetre daha akımın aktığını algılayamadan ekspojur sonlanacaktır. Bu yüzden yüksek örnekleme oranıyla verinin sayısal olarak hızlı bir şekilde alınıp işlenmesi gerekmektedir. Tipik bir ekspojur 20-40 milisaniye sürebilmektedir. Bu süre görüntülenecek doku ya da nesnenin niteliğine göre operatör tarafından belirlenmektedir.

Ohm kanununa göre akım yolu üzerine bağlanan saf ohmik bir direnç üzerinde  $V = I \times R$  ile verilen bağıntıda görüleceği üzere akım ile doğru orantılı bir gerilim düşümü olmaktadır. Yüksek gerilim trafosunun orta uçları arasına bağlanan uygun bir direnç üzerinde düşen gerilim bize  $I = V / R$  bağıntısı ile akımı ölçme imkânı verir. Direnç üzerinde düşen gerilim uygun kuvvetlendirme ve filtreleme işlemlerine tabi tutularak amaca uygun teknik özelliklere sahip bir ADC<sup>1</sup> ile örneklenmesi mümkün olmaktadır.

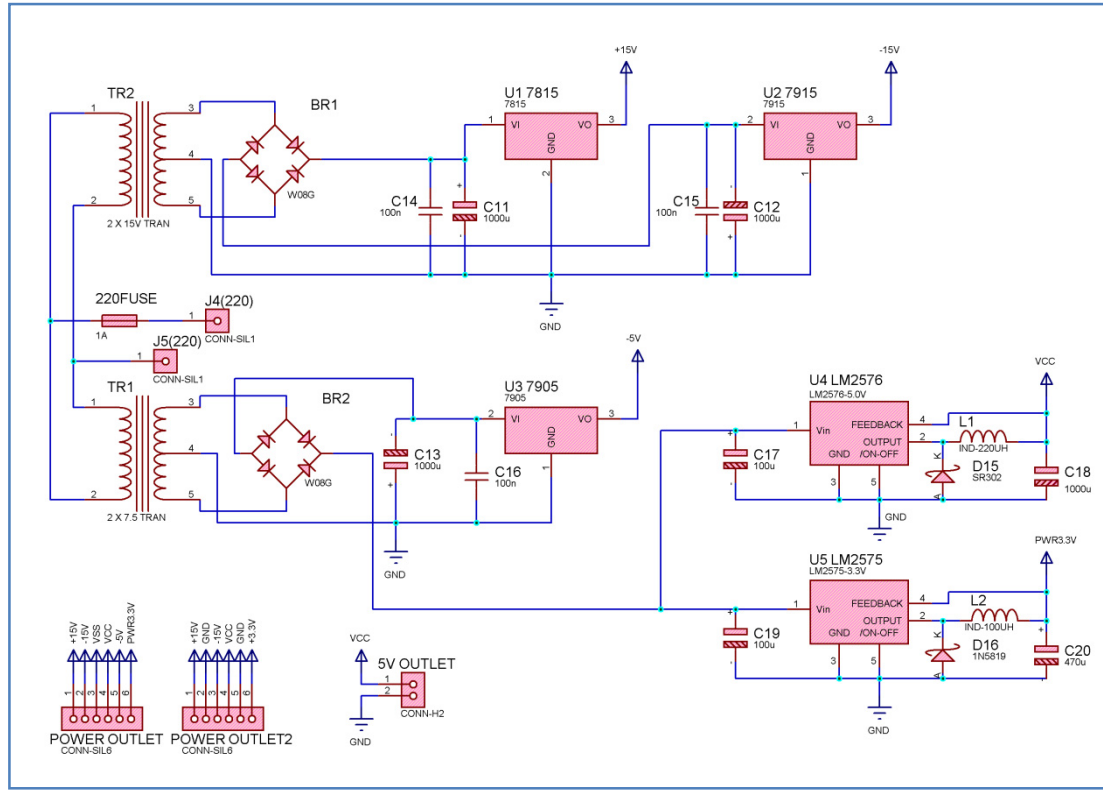


Resim 7.1. Tasarlanan mAs-metre'nin montajlanmış hâli

<sup>1</sup> analog digital converter, analog sayısal dönüştürücü

## 7.3. Cihazın Elektronik Devre Blokları

### 7.3.1. Besleme katı



Şekil 7.3. mAs metre besleme devresi

Cihazın beslemesi 200-240V aralığındaki AC şebekelerde çalışabilecek şekilde tasarlanmıştır. Sinyal işleme katlarında kullanılan simetrik beslemeli opamp'lar +15V, -15V, mikrodenetleyici ve lojik entegreler +5V, bluetooth iletişim modülü +3.3V regüleli besleme kaynaklarına ihtiyaç duymaktadır. Tasarım aşamasında bazı gerekliliklerle -5V kaynak ta gerekmiş ve bu da besleme katına dâhil edilmiştir. +15V, -15V, -5V kaynaklar doğrusal regüleli besleme entegreleri olarak seçilmiş, +5V besleme hattındaki devre elemanlarının akım ihtiyacı yüzünden lineer besleme entegresi kullanılması durumunda gerilim düşümlerinin yaşanması nedeniyle de mikrodenetleyicinin istenen saat hızlarına çıkamadığı ve zaman zaman kilitlenerek resetlendiği görüldüğünden +5V beslemede 3 ampere kadar besleme sağlayabilen anahtarlamalı tip gerilim kaynağı entegresi olan LM2576 kullanılmıştır. Yine

bluetooth modülünün sabit regüledi +3,3V gerilim ihtiyacını karşılamak üzere LM2575 anahtarlama tip gerilim kaynağı entegresi kullanılmıştır. Simetrik besleme sebebiyle duyulan negatif gerilimleri elde etmek için orta uçlu 4W gücünde transformatörler kullanılmış transformatörlerin orta uçları ground olarak bağlandığında istenen negatif gerilim 7915 ve 7905 lineer gerilim regülatörü entegreleri vasıtasıyla üretilmiştir. Her gerilim regülatörü entegresi çıkış gerilimi sağlarken girişindeki gerilimle aradaki gerilim farkını üzerinde harcamak durumunda kalacağından aşırı ısınmayı önlemek adına 2 transformatör kullanılmış, 2 x 15V olarak seçilen transformatör çıkışı köprü diyot devresinde doğrultulduktan sonra diğerlerine göre yüksek gerilime ihtiyaç duyan 7915 ve 7815 entegrelerinin beslemesi bu transformatörden sağlanmış, geri kalan LM2575, LM2576 ve 7905 entegreleri de 2 x 7,5V'luk transformatöre bağlanmıştır. Tüm cihaz herhangi bir kısa devre durumunda devrenin açması için 1A'lık sigorta ile korunmuş, ayrıca cihazı geliştirme aşamasında ihtiyaç duyulabileceği göz önünde bulundurularak harici devrelere besleme sağlayabilmek için 2 adet SIL<sup>1</sup> priz bırakılmıştır.

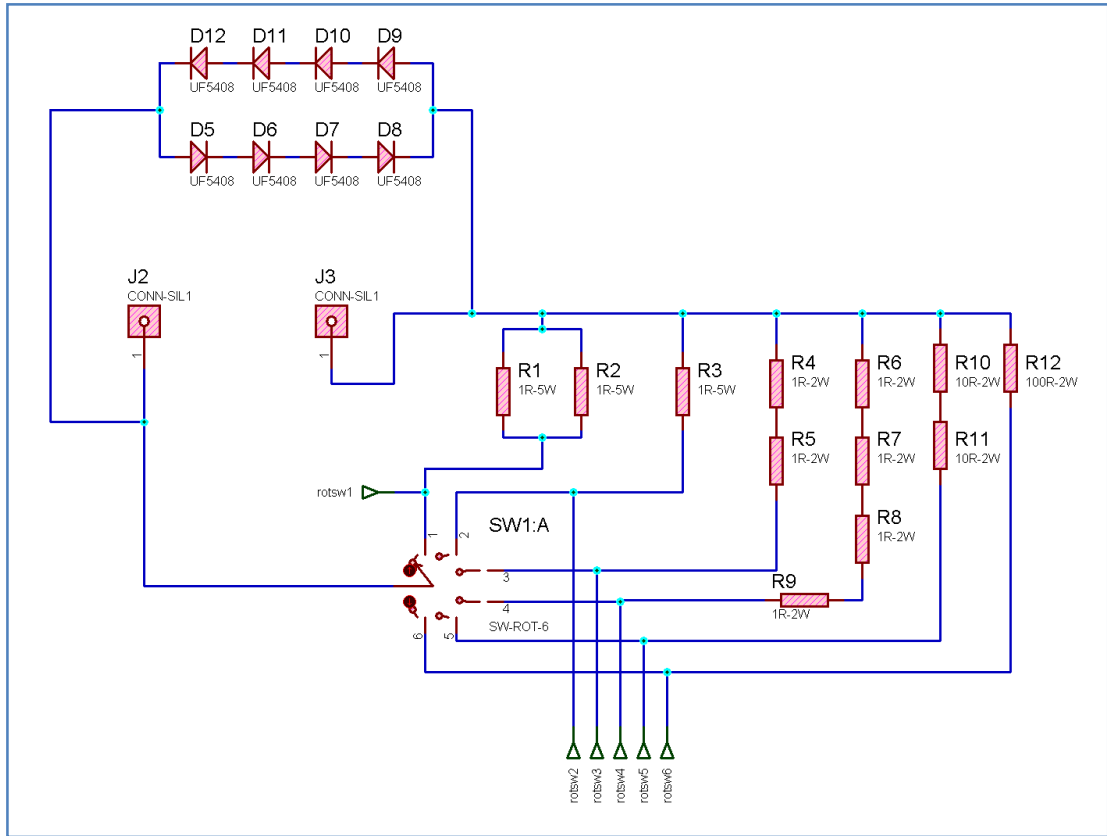
### 7.3.2 Anahtarlama – ön ölçekleme katı

Bu katta Şekil 7.4'te görülen J2 ve J3 temas noktalarından yüksek gerilim trafosunun sekonder orta ucuna direkt bağlantı yapılmaktadır. D5-D12 kodlarıyla belirtilen diyot grubu komütatörün arıza vermesi ya da direnç kademelerinden herhangi birinin herhangi bir sebeple devre yolunu açması yahut üzerinde öngörülenden daha yüksek gerilim düşümü yaşanması durumunda yüksek gerilim trafosunun görevini mAs metre'de oluşan problemten etkilenmeden sürdürebilmesi için kontrollü bir alternatif iletim yolu oluşturmaktadır. Tasarımda, kademelerdeki tüm dirençler güç ve direnç değeri itibarıyla üzerlerinde 2 volttan daha fazla gerilim düşmeyecek şekilde seçilmiştir. Diyot grubu da UF5408 3A yüksek ileri yönde kutuplama gerilimine sahip diyotlardan alternatif akımı her iki yönde de iletebilecek şekilde köprü düzeninde ve  $V_f$  gerilimleri toplamda belli bir emniyet payıyla birlikte 3 volt gerilimden önce diyotların iletme geçmesini engelleyecek şekilde dörderli seri

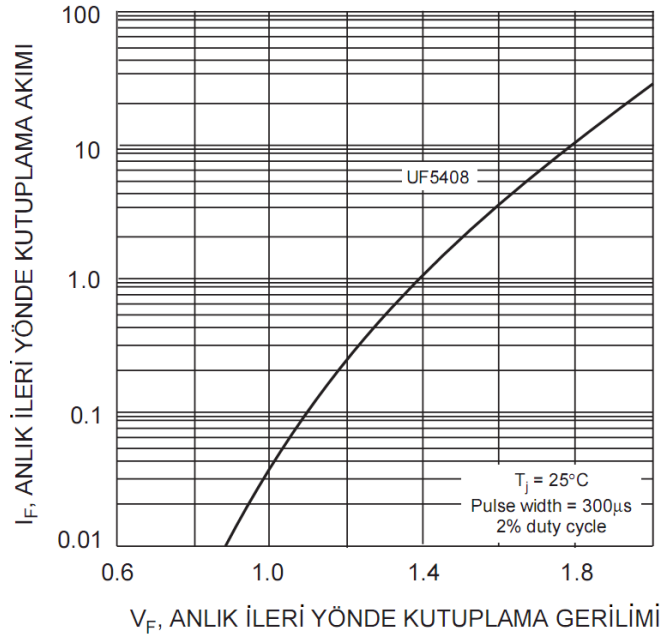
---

<sup>1</sup> single in line

bağlantıda tasarlanmıştır. Kullanılan diyotların ileri yönde kutuplama geriliminin ileri yönde kutuplama akımına bağlı olarak değişimi Şekil 7.5'ten incelenebilir. Genelde mAs metre'nin ölçüm aralığı birkaç yüz miliamperi aşmayacaktır. Bu durumda diyotların her birinin iletme geçmek için 1 volt civarı ileri yönde kutuplama gerilimine ihtiyaç duyacağı açıktır. Ölçümü yapılan tüp akımının mAs metrenin seçilmiş olan kademedeki ölçüm aralığının üst sınırına yakın olması durumunda kademedeki direnç üzerinde düşen gerilim 2V'a yaklaşacaktır. Bu durumda 4'lü seri direnç grubu hiçbir şekilde tam olarak iletme geçmese de bir sızıntı akımı akıtması, dolayısıyla ölçülmesi istenen akımın bir kısmına alternatif bir



Şekil 7.4. Kademe seçici komütatör anahtar ve kademe dirençlerinin bağlantıları



Şekil 7.5. UF5408 diyotun ileri yönde kutuplama gerilimi ile akımının değişimi

kaçış yolu oluşturması mümkün görünmekle birlikte, mikrodenetleyici tarafından hesaplama sonucuna yapılan katsayı düzeltmeleri ile bu durum bir ölçüde giderilmiştir. Önceden belirlenen 2V maksimum değerine göre gerekli güç ve akım hesaplamaları yapılmış ve ölçüm kademeleri sırasıyla 20 mA, 100 mA, 500 mA, 1000 mA, 2000 mA ve 4000 mA olmak üzere 6 kademeli olarak belirlenmiştir. Çizelge 7.1’de bu hesaplamaların sonucu özetlenmiştir.

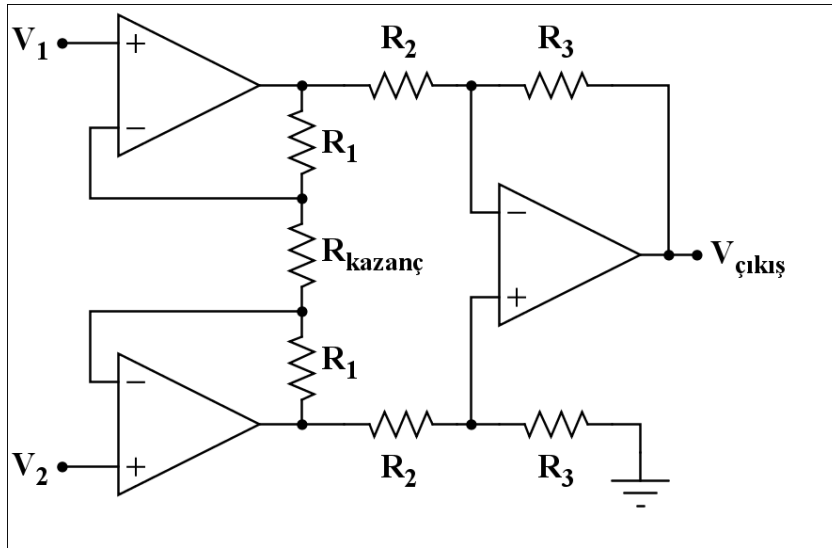
Çizelge 7.1. Ölçüm kademelerindeki dirençlerin akım-gerilim-güç ilişkisi

Gerilim (V)	Direnç (ohm)	Akım (mA)	Güç (mW)
2	0,5	4000	8000
2	1	2000	4000
2	2	1000	2000
2	4	500	1000
2	20	100	200
2	100	20	40

Hesaplamaların yapılmasında dirençlerin mümkün olduğunca piyasadaki standart değerlere uyması gözetilmiş, standart dışı değerlerin gerektiği durumlarda da seri – paralel kombinasyonlarla istenen değerler elde edilmiştir.

### 7.3.3. Aletsel yükselteç

Bir aletsel yükselteç ölçme ve test cihazlarında kullanılacak niteliklerde özel olarak tasarlanmış bir fark yükseltecidir. Bahsedilen nitelikler çok düşük *d.c.*<sup>1</sup> *offset*, düşük *kayma*<sup>2</sup>, düşük *gürültü*<sup>3</sup>, çok yüksek *açık çevrim kazancı*<sup>4</sup>, çok yüksek *CMRR*<sup>5</sup> ve çok yüksek *giriş empedansı* olarak sayılabilir. Kısa ve uzun vadede yüksek doğruluk ve stabilitenin gerektiği devrelerde kullanılırlar. Temel hâliyle sıkça kullanılan bir aletsel yükselteç devresi Şekil 7.6’da görülebilir. Bir aletsel yükseltecin ideal ortak mod kazancı sıfırdır. Gösterilen devrede ortak mod kazancı eş numaralı dirençlerin değerlerindeki eşitsizlik ve girişteki iki opamp’ın sıfırdan farklı ortak mod kazançları nedeniyle ideal değerden farklı olabilir.



Şekil 7.6. Üç adet opamp ile gerçekleştirilmiş tipik bir aletsel yükselteç

<sup>1</sup> direct current, doğru akım

<sup>2</sup> drift

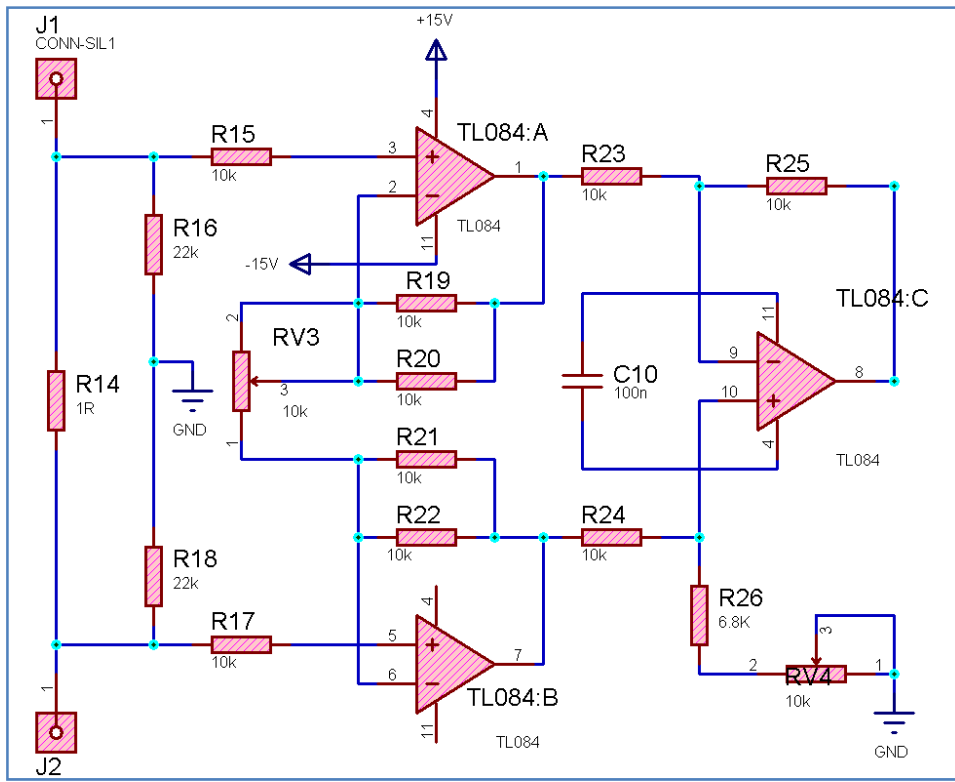
<sup>3</sup> noise

<sup>4</sup> open loop gain

<sup>5</sup> common mode rejection ratio, ortak mod bastırma oranı

Buradaki kazancı belirleyen dirençlerin birbirlerine tam olarak eşit olması üretim aşamasında pek mümkün olamayabildiğinden 3 opampı içeren tümleşik bir aletsel yükselteç yerine ayrı opamp entegreleri ile dışarıdan müdahale ederek ofset değerlerini ayarlayabildiğimiz bir aletsel yükselteç düzeneği kurulmuştur. Tipik bir aletsel yükseltecin kazancı şu ifade ile belirlenir:

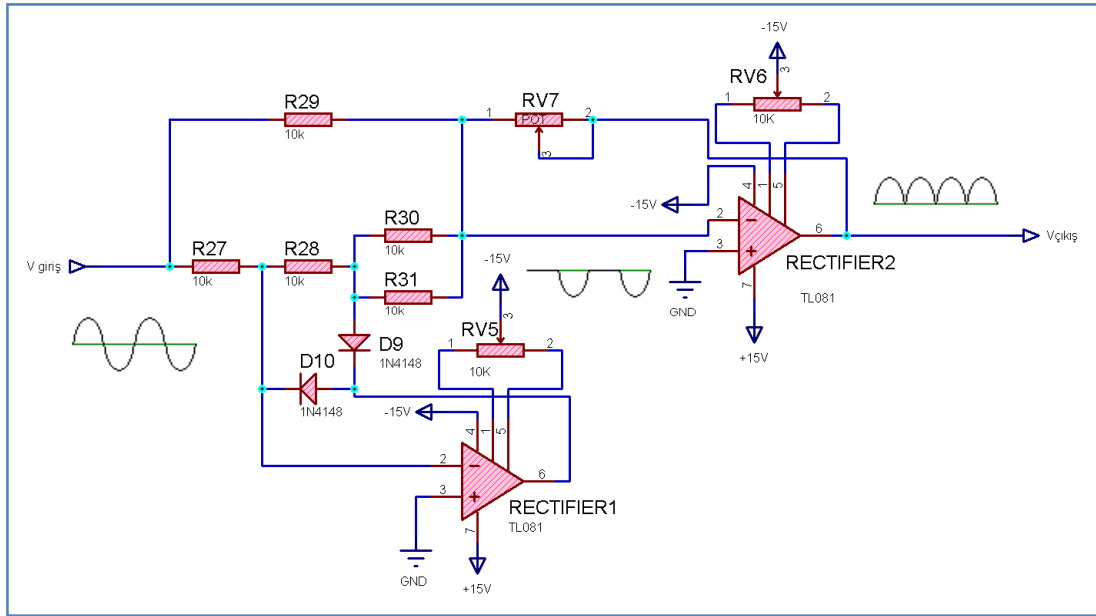
$$\frac{V_{\text{çıkış}}}{V_2 - V_1} = \left( 1 + \frac{2R_1}{R_{\text{kazanç}}} \right) \frac{R_3}{R_2}$$



Şekil 7.7. Aletsel yükselteç devre bloğu

### 7.3.4. Hassas tam dalga doğrultmaç

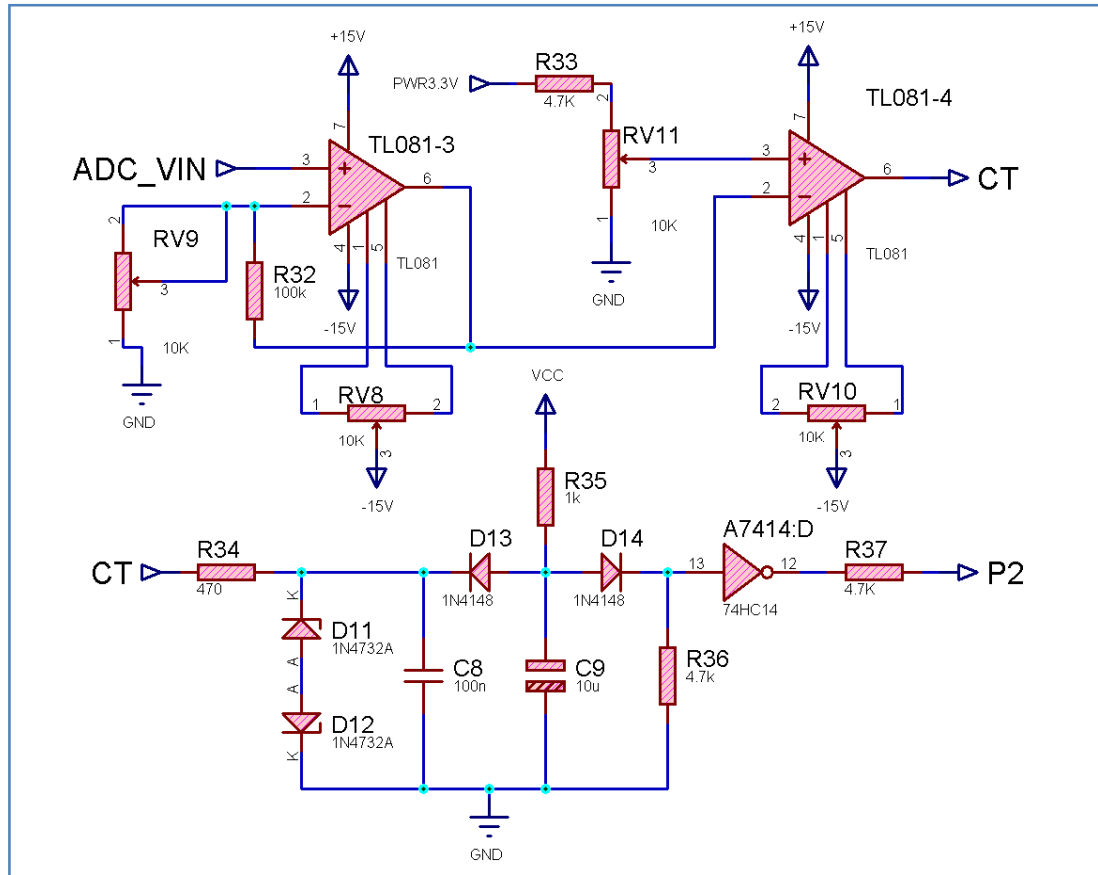
Şekil 7.8’de kullanılan hassas tam dalga doğrultmaç devre yapısı verilmiştir. Burada Vgiriş’e gelen AC sinyal ilk opamp’ta yarım dalga olarak doğrultulduktan sonra ikinci opamp çıkışında tam dalga doğrultulmuş DC bir sinyal olarak elde edilir. Opamplar ayrı entegreler olarak seçildiğinden ofset ayarları birbirinden bağımsız olarak yapılabilmektedir.



Şekil 7.8. Hassas tam dalga doğrultmaç devre şeması

### 7.3.5. Ekspojur algılama devresi

Ekspojur algılama devresi mikrodenetleyicinin önceden belirlenen seviyenin üzerinde genliğe sahip bir sinyal algılaması durumunda ADC için çevrim sürecini başlatmasına olanak sağlamak için lojik bir kontrol sinyali üretmektedir. ADC'ye ölçülmek üzere giden sinyal ADC\_VIN bağlantı noktasından TL081-3 kodlu opampa girilmekte, burada çok turlu potansiyometre ile ayarlanabilen bir kazançla yükseltilmekte daha sonra TL081-4 kodlu opampta 3,3V sabit gerilimin gerilim bölücü yapısı ile ayarlanan bir oranına tekabül eden kısmı ile karşılaştırılmaktadır. İkinci opamp'ın çıkışının pozitiften negatife geçtiği durumda CT bağlantı noktasındaki sinyal gerilimi sınırlandırmak için kullanılan zener diyotlar ve normal diyotlar üzerinden 7414 lojik NOT kapısına girilir, bu sayede TTL uyumlu lojik bir sinyal elde edilmiş olur. P2 noktası olarak tanımlanan uçtan mikrodenetleyicinin RC13 pinine girilen sinyal hem ekspojur süresinin ölçülmesi için bir sayacı aktive eder hem de bu sinyalin başlamasıyla birlikte ADC örnekleme süreci başlatılır.



Şekil 7.9. Ekspojur algılama devresi

### 7.3.6. Mikrodenetleyici katı

Mikrodenetleyici olarak Microchip firmasının dsPIC serisinden 30F4011 seçilmiştir. Bu mikrodenetleyici Harvard mimarisinin değiştirilmiş bir türü ile inşa edilmiştir. Esnek adresleme modları bulunan C derleyicisi ile kullanım için optimize edilmiş komut setine sahiptir. 83 adet 24 bit genişliğinde temel komuta, 16 bit veri yoluna sahiptir. 48 Kilobyte flash program belleğine, 2 Kbyte RAM'e ve 1 Kbyte kalıcı EEPROM<sup>1</sup> a sahiptir. Saniyede 30 milyon işlem yürütebilmektedir. Harici 40 MHz'e kadar olan saat sinyallerini PLL ile çoklayarak 120 MHz'e kadar efektif olarak kullanabilmektedir. 3 harici olmak üzere 30 tane interrupt kaynağı kullanabilir. Bu kesmelerin önceliği 8 seviyeli olarak belirlenebilmektedir. 16 x 16

<sup>1</sup> Electrically Erasable Programmable Read Only Memory

bit'lik yazmaç dizisi kullanır. DSP<sup>1</sup> özellikli bu mikrodenetleyicide ayrıca çift veri besleme kanalı, modüler ve evirilmiş bit adresleme modları, iki tane 40 bit genişliğinde opsiyonel doyum devresi ile birlikte kullanılabilen akümülatör, 17 bit x 17 bit tek komut saykılında donanım destekli rasyonel yahut tam sayı çarpıcı, tüm DSP fonksiyonlarını tek saat darbesinde yürütebilme,  $\pm 16$  bit tek saat darbesinde kaydırma özellikleri mevcuttur. Çevre birimleri olarak ta 25mA giriş çıkışa izin veren portları, 5 adet 16 bit sayıcı/zamanlayıcı, istenirse 32 bit olarak bağlanabilen 16 bitlik ön ölçeklemeli zamanlayıcılar ve sayıcılar, 16 bit yakalama girişi fonksiyonları, 16 bitlik karşılaştırma, darbe genişlik modülasyonu çıkışı özellikleri, 3-wire SPI<sup>2</sup> modülleri, çoklu master slave özellikli ve 7 bit/10 bit adreslemeli I<sup>2</sup>C<sup>3</sup> modülü, 2 adet FIFO<sup>4</sup> tamponlu UART<sup>5</sup> modülü, 1 adet 2.0B uyumlu CAN<sup>6</sup> modülü mevcuttur. Motor kontrolü ve PWM<sup>7</sup> modülü olarak ta 6 adet PWM çıkış portu, bu portlarda tümleyen ya da bağımsız çıkış modları, kenar ya da tepe hizalı çalışma modları, 3 adet çevrim saykılı üretici, bağımsız zamanlama, programlanabilir çıkış polaritesi, tümleyen modu için ölü zaman kontrolü, manüel çıkış kontrolü, A/D dönüşümler için tetikleme bulunmaktadır. Kuadratür enkoder arayüz modülünde ise A fazı, B fazı ve indeks darbeleri girişleri, 16 bitlik aşağı/yukarı pozisyon sayacı, sayma yönü durum göstergesi, pozisyon ölçümü, girişlerde programlanabilir sayısal gürültü filtreleri, sayıcı ya da zamanlayıcı modları arasında geçiş ve pozisyon sayacı taşmalarında kesme üretme özellikleri mevcuttur.

dsPIC30F4011 mikrodenetleyicisinin sahip olduğu gelişmiş haberleşme arabirimleri, geniş program belleği ve çok sayıda giriş çıkış portu yanında bu çalışmada kullanılmak üzere seçilmesinin asıl nedeni sahip olduğu özel analog sayısal dönüştürücüdür. 10 bit çözünürlükte örnekleme yapabilen ve 9 kanaldan giriş kabul eden bu ADC'nin asıl etkileyici özelliği 1MSPS<sup>8</sup> olarak ifade edilen dönüşüm hızına

---

<sup>1</sup> Digital Signal Processing

<sup>2</sup> Serial Peripheral Interface Bus

<sup>3</sup> Inter-Integrated Circuit

<sup>4</sup> First In First Out

<sup>5</sup> Universal Asynchronous Receiver/Transmitter

<sup>6</sup> Controller Area Network

<sup>7</sup> Pulse Width Modulation

<sup>8</sup> Mega Sample Per Second, saniyede milyon kere örnekleme hızı

çıkabilmesidir. Bu nitelikte bir ADC'nin mikrodenetleyiciye entegre olarak sunulması sık rastlanır bir özellik değildir. Bu ADC sayesinde cihazın tasarımı esnasında izlenmiş olunan önceki metotlarda kullandığımız ve yavaş mikrodenetleyici ile haricî ADC kombinasyonunun getirdiği teknik problemlerin üstesinden gelinmiştir.

Kullanılan mikrodenetleyicinin yukarıda sayılanlara ek, sayısal sinyal kontrolörü olarak farklı özellikleri de bulunmaktadır. Bunlar arasında minimum 10 000 kez, tipik olarak 100 000 kez yazılıp silinebilen endüstriyel kullanım sıcaklık aralığında çalışmaya müsait gelişmiş flash program belleği, minimum 100 000 kez, tipik olarak 1 milyon kez yazılıp silinebilen veri eeprom belleği, yazılım sayesinde kendi kendine tekrar programlanabilme, enerji verildiğinde resetleme<sup>1</sup>, enerjili durum sayacı<sup>2</sup> ve osilatör başlama sayacı<sup>3</sup>, esnek bekçi köpeği zamanlayıcı, düşük güç tüketimli tümleşik RC osilatör, haricî osilatörün arızası durumunda bunu algılayıp otomatik olarak dâhilî osilatöre geçebilme, programlanabilir kod koruma, devre üzerinde iken programlanabilme<sup>4</sup>, seçilebilir güç yönetim modları: uyku, bekleme ve değiştirilebilir osilatör modları gösterilebilir. Ayrıca bu mikrodenetleyici 2,5V ve 5,5V aralığında esnek besleme voltajını kabul etmekte ve endüstriyel sıcaklık aralığında çalışma şartlarını karşılamaktadır.

Mikrodenetleyicinin yazılımsal çalışması bir başka bölümde incelenecek olmasına rağmen kontrol ettiği donanımlara burada kısaca değinmek gerekir. Data hattı mikrodenetleyicinin D portuna, kontrol uçları RB7 ve RB8 pinlerine bağlı olan ve 4 bit iletişim modunda kontrol edilen 4 satır 20 kolon karakter lcd üzerinde kullanıcının seçeceği fonksiyonlar ve işlem sonuçları görüntülenmektedir. Harici 10 Mhz'lik kristal bir osilatör dâhilî PLL devresi ile x8 modda çalışmakta yani mikrodenetleyici 80 MHz hızında bir osilatöre bağlı imiş gibi işlem yapmaktadır. E portuna bağlı olan kademe seçici komütatörün B kutbu komütatörün hangi direnç

---

<sup>1</sup> Power-on Reset, POR

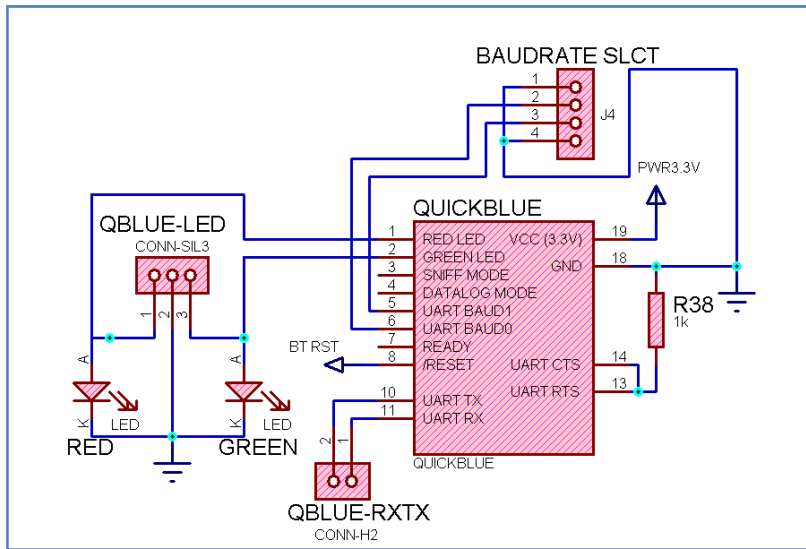
<sup>2</sup> Power-up Timer, PWRT

<sup>3</sup> Oscillator Start-up Timer, OST

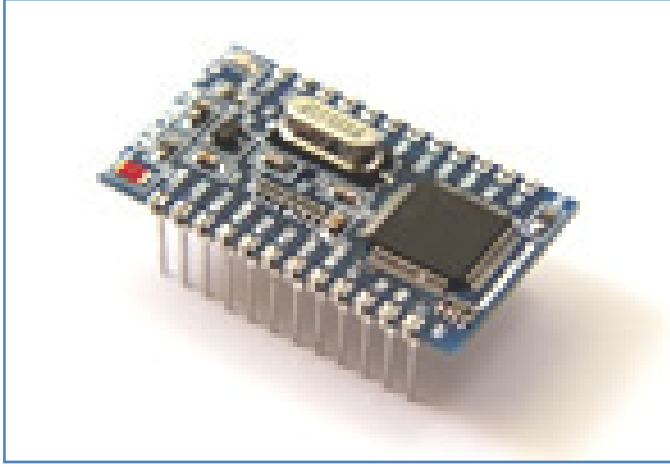
<sup>4</sup> In Circuit Serial Programming, ICSP

grubunu devreye aldığı bilgisinin okunmasını ve hesaplamasının buna göre yapılmasını sağlar. RF4, RF5 ve RC14 pinlerine bağlı ve harici kesme kaynağı olarak ayarlanmış butonlarla kullanıcının cihazı kontrol etmesi sağlanmıştır. RF2 ve RF3 pinleri aynı zamanda mikrodenetleyicinin UART modülü olarak ayarlanabildiğinden bluetooth modülü ile iletişim bu pinler üzerinden sağlanmaktadır. RC13 pinine ekspozur algılama devresinin lojik kontrol sinyali girilmiş, RB5 pini de ADC giriş kanalı olarak ayarlanmış, opamp bloklarından gelen sinyal bu pinden alınarak örneklenmektedir. Bunun dışında RB0 pinine de jumper ile seçilebilen (2,000V, 4,096V ya da 5,000V) haricî referans gerilim kaynağı entegresinin çıkışı bağlanmıştır. Mikrodenetleyici, kod geliştirme sürecinde kolaylık sağlaması için ICSP özelliğinden faydalanılarak devre üzerinde programlanabilecek şekilde bir ICSP konnektörüne ilgili bağlantılar yapılmıştır. Programlama yapılmadığı zamanlarda ICSP devresi üzerindeki diyotların sebep olduğu 0,7V kadarlık ileri yöndeki gerilim düşümünü bypass edebilecek jumperlar bırakılmıştır. Bunun dışında haricî referans gerilimi kullanıldığı için üreticinin referans katalogundaki tavsiyelere uyularak muhtelif değerlerde filtreleme kondansatörleri ile mikrodenetleyicinin besleme hattındaki gürültüler giderilmiştir.

### 7.3.7. Bluetooth iletişim katı



Şekil 7.10. Bluetooth modülü bağlantı şeması

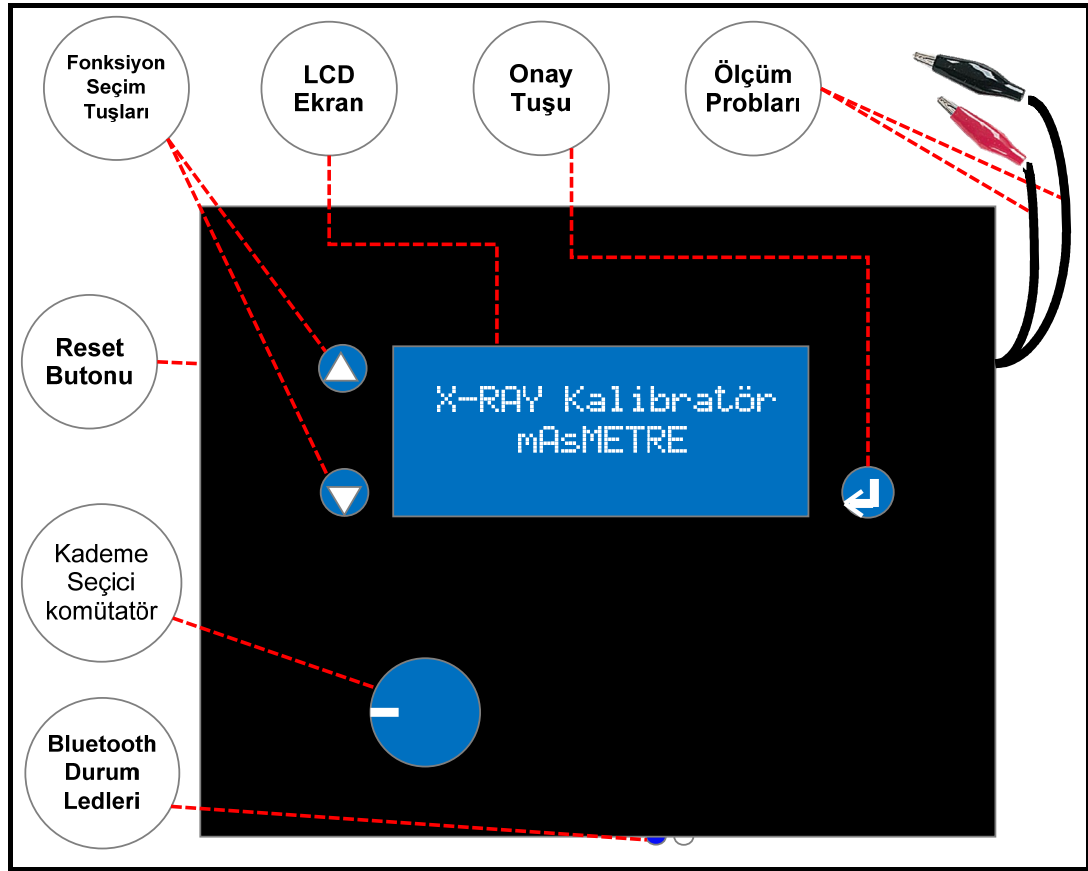


Resim 7.2. Quickblue bluetooth modülü

mAs metre'nin PC ile iletişimi kablosuz bağlantı teknolojilerinden bluetooth protokolü ile sağlanmaktadır. Kullanılan bluetooth modülü üzerinde içinde bluetooth stack'lerini koşturan 18FXX20 serisinden bir mikrodenetleyici kullanılmıştır. Bu mikrodenetleyici yine modül üzerinde bulunan taiyo-yuden firmasının üretmiş olduğu EYMF2CMM-XV kodlu bluetooth radyo çipini kontrol etmektedir. Tüm modül 24 pinlik DIL bir kılıf üzerinde montajlanmıştır. 3,3V besleme kaynağına ihtiyaç duyan Quickblue™ bluetooth modülünde bağlantı durumunu gösteren iki adet LED, seri iletişim hızının kademeli olarak seçilebildiği pinler, reset ucu, Rx Tx UART pinleri bulunmaktadır. 18FXX20 mikrodenetleyici üzerinde çalışan gömülü kod açık kaynaklıdır. Şekil 7.10'da modülün devre bağlantı şeması ve Resim 7.2'de de gerçek görünümü yer almaktadır.

## 8. KULLANIM KILAVUZU

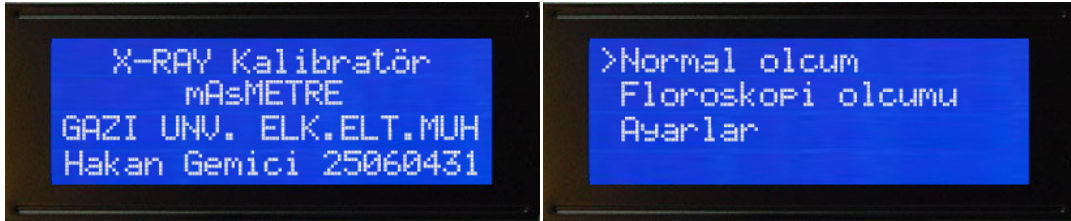
Cihaz üzerinde 4x20 karakter LCD, 3 kontrol ve 1 reset butonu, 1 kademe seçici komütatör, 2 ölçüm probu ve 2 adet Bluetooth bağlantı durum ledi yer almaktadır. Tüm bunların yerleşimi Şekil 8.1'den görülebilir.



Şekil 8.1. mAs metre kontrollerinin yerleşimi

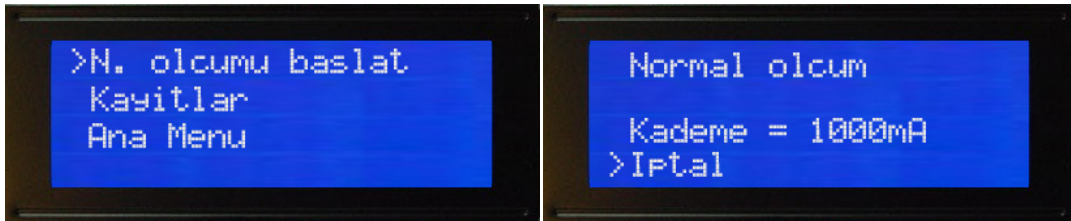
Cihaz 220V 50 Hz şehir şebekesinden çalışabilmektedir. Kullanım için öncelikle fişi prize takınız. Cihaz enerjilendiğinde birkaç saniyelik bir yazı görüntülenmektedir. Bu ekran görüntüsü Şekil 8.2a'da incelenebilir. Daha sonra Şekil 8.2b'de görülen ana seçim ekranına gelmektedir. Buradaki seçimlerden "Normal Ölçüm" milisaniye mertebesindeki ekspozür sürelerinin ölçülmesi için, "Floroskopi Ölçümü" de süre sınırı olmaksızın dakikalar mertebesinde ölçüm

yapılabilmesi için kullanılmaktadır. Ayarlar ekranından özel bir tuş kombinasyonu ile ayarlara girilir.



Şekil 8.2. Ekran görüntüsü

Ayar menüsünde normal kullanıcının ayarlayabileceği herhangi bir işlem bulunmamaktadır. Ana seçim ekranındayken yön tuşları ile imleç aşağı yukarı hareket ettirilerek istenen yere getirilir. Onay tuşuna basıldığında imlecin bulunduğu fonksiyon seçilecektir. Şekil 8.3a'da gösterilen "Normal Ölçüm" moduna girildiğinde istenirse önceden alınmış olan ölçüm sonuçlarına "Kayıtlar" seçeneğinden bakılabilir. Menüden çıkılmak istenmesi durumunda imlecin "Ana Menü" seçeneğine getirilip onay verilmesi yeterli olacaktır.



Şekil 8.3. Ekran görüntüsü

"Normal Ölçüm" modu seçilmiş ise Şekil 8.3b'de gösterilen ekran görünecektir. Bu menüde mAs metre ölçmek üzere ekspozur sinyalini beklemektedir. Eğer farklı bir kademe ölçüm yapılacaksa ilgili ölçüm kademesi seçici komütatör aracılığı ile belirlenmelidir. Komütatörün farklı kademeye getirildiği bir ekran görüntüsü Şekil 8.4a'da görülebilir. Önceden belirlenmiş olan kademe doğru ise herhangi bir değişiklik yapılmasına gerek yoktur. mAs metre sinyal başladığı an dönüştürmeyi başlatacak ve ekspozur sonlandığında da sonucu ekranda gösterecektir.



Şekil 8.4. Ekran görüntüsü

Sonuç ekranı Şekil 8.4b’de görülebilir. Bu ekranda sırasıyla miliamper, milisaniye ve miliamper x saniye değerleri gösterilmektedir. İstenirse yeni bir ölçüm almak üzere “Geri” fonksiyonu seçilerek çıkılabilir, istenirse ölçüm sonucu 10 adet kayıt belleğinden birine saklanabilir. “Normal Ölçüm” modu için 10, “Floroskopi” modu için 10 adet ayrı bellek ayrılmıştır. Değerler sadece ölçüm esnasında birbirleriyle karşılaştırılmak için saklanmakta, cihazın enerjisi kesildiğinde silinmektedir. Şekil 8.5a’da “Floroskopi” modunda alınan bir ölçüm sonucu ve Şekil 8.5b’de de bu sonucun kaydedilmesi için imlecin “Kaydet” seçeneğine geçirilmiş vaziyeti görülmektedir.



Şekil 8.5. Ekran görüntüsü

Onay tuşuna basılmasıyla Şekil 8.6a’da verilen kayıt menüsü görülür. Bu ekranda ok tuşları kayıt belleğinin sıra numarasını artırıp azaltmaya yarar. 0 numaralı bellek gözüne sonucun kaydedildiğine dair mesajın görüntülenmesine ilişkin ekran görüntüsü Şekil 8.6b’de verilmiştir. Burada gelen “Ana Menü” seçeneği ile ilgili ekrana dönülür.



Şekil 8.6. Ekran görüntüsü

Önceden verilmiş olan ekran görüntüsünden (Bkz. Şekil 8.3a) “Kayıtlar” yolu izlenerek önceki kayıtlar incelenecek olursa karşılaşılabilecek olan menü Şekil 8.7a’da verilmiştir. Bu ekranda ok tuşları ile imleç istenen kayıt numarasına getirilir ve onay verilirse önceden kaydedilmiş olan ölçüm sonucuna ulaşılır. Bu ekran görüntüsü de Şekil 8.7b’de verilmiştir.

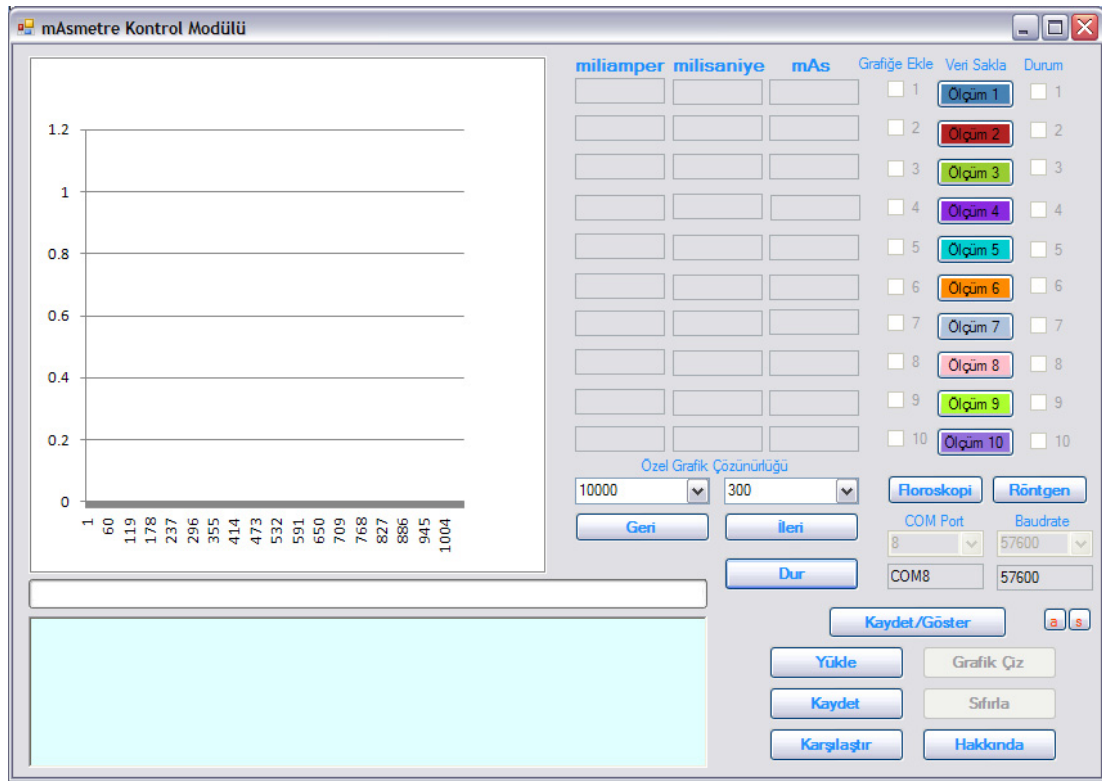


Şekil 8.7. Ekran görüntüsü

Ölçüm yapılacağında röntgen tüpünü besleyen yüksek gerilim trafosunun orta ucuna birbirleri ile ve gövde ile kısa devre etmediğinden emin olarak cihazın problemleri sıkıca bağlanmalıdır. Hangi probun hangi uca bağlandığının bir önemi yoktur. Ölçüm esnasında kurşun paravan arkasına geçilmeli ya da kurşun yelek giyilmelidir. En sağlıklı cihaz ölçüm alırken kumanda odasına geçerek mAs metre’yi uzaktan bilgisayar aracılığı ile kontrol etmektir. Bunun yapılabilmesi için hazırlanmış olan programın kullanımından ayrıca bahsedilecektir. Eğer fişe takıldığında ekrana görüntü gelmiyor ya da Bluetooth durum ledleri yanıp sönmüyorsa mAs metre’nin içi açılarak sigortası kontrol edilmelidir. Cihaz enerjili iken içi kesinlikle açılmamalıdır. Üzerine takılı olan 1A’lık sigortadan daha büyük değerli bir sigorta takılmamalıdır. Eğer ekranda görüntü var iken kontrol butonları cevap vermiyorsa cihazın sol tarafındaki siyah reset butonundan resetlenmelidir.

## 8.1. Bilgisayardan Kontrol Yazılımı

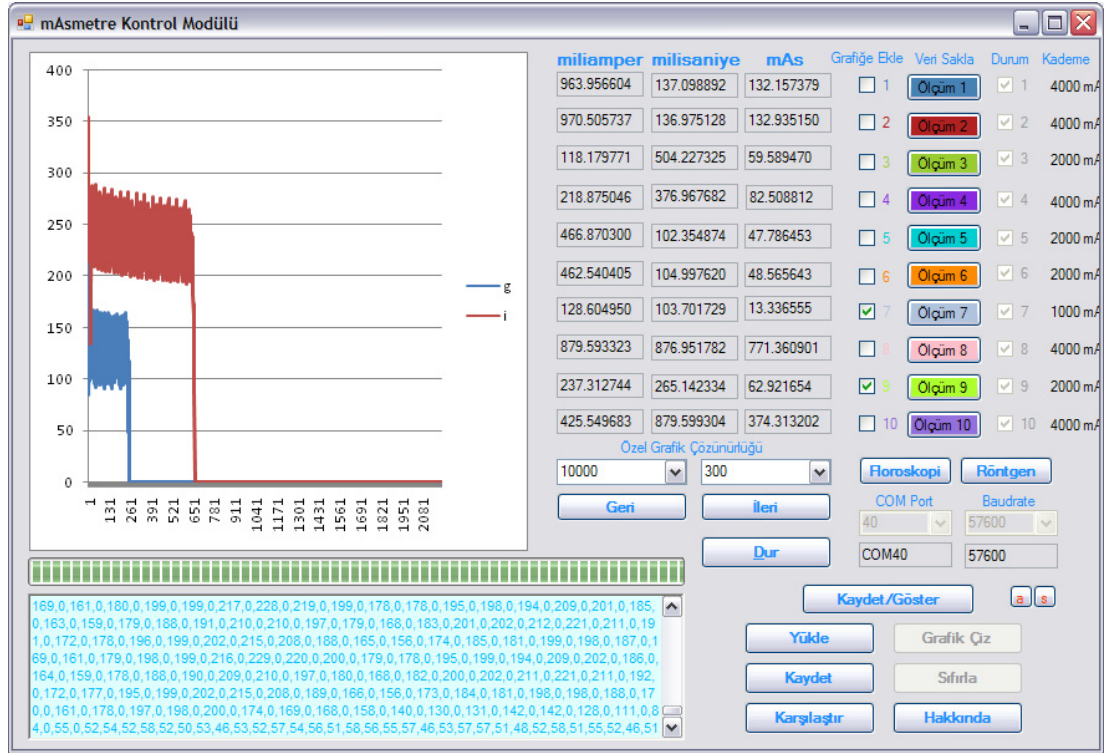
İmal edilen kalibrasyon cihazının asıl önemli fonksiyonu gerçek zamanlı olarak örneklemediği tüp akımını ve hesapladığı ölçüm sonuçlarını Bluetooth iletişim protokolü ile bilgisayara aktarabilmesidir. Ayriyeten uzaktan kontrol edilebilmekte ve ister Floroskopi modunda ister Normal ölçüm modunda ölçümler alınabilmektedir. Bilgisayara aktarılan veriler xls dosyaları olarak saklanmakta ve ileri analizler için kullanılabilir. Ayrıca alınan verilerin tek başına ya da birbirleriyle karşılaştırmalı grafikleri kontrol programı üzerinde çizdirilebilmektedir.



Şekil 8.8. mAs metre kontrol modülü ekran görüntüsü

Uzaktan kontrol fonksiyonunun çalışması için öncelikle bilgisayarın tümleşik olarak ya da Bluetooth adaptörü ile Bluetooth iletişimini desteklemesi gerekmektedir. Grafik çizme ve veri kaydetme fonksiyonlarının çalışabilmesi için bilgisayarda Microsoft Ofis 2007 sürümü ve .Net 2.0 framework kurulu olmalıdır. mAs metre'ye enerji verildiği anda otomatik olarak bağlanabileceği Bluetooth cihazları arama

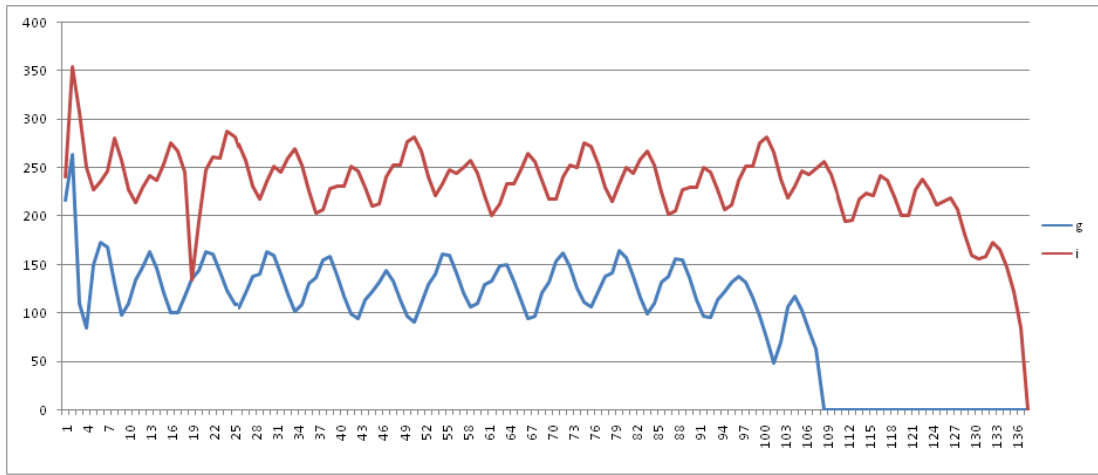
moduna geçer. Bilgisayar üzerinden çevredeki Bluetooth aygıtları aratılacak olursa “Quickblue” adıyla mAs metre’nin Bluetooth modülü görünecektir. Bu aşamada bilgisayar ile arasında eşleştirme yapılmalıdır. Bluetooth bağlantı şifresi varsayılan değer olarak “1234” girilmelidir. Eşitleme işleminden sonra mAs metre kontrol modülü çalıştırılmalıdır. Com port olarak 8 numaralı port ve iletişim hızı olarak ta 57600 baud seçili gelmektedir. Com port ayarlarından eşleştirme esnasında bilgisayarın mAs metre’ye atamış olduğu com port numarası seçilmeli ve iletişim hızı olarak ta 57600 baud olduğu gibi bırakılmalıdır. “Sistemi Çalıştır” butonuna basılmasıyla birlikte bağlantı kurulacaktır. Arama modunda mavi ve beyaz durum ledleri dönüşümlü olarak yanan mAs metre’de bağlantının kurulmasıyla birlikte beyaz led sabit, mavi led fasıllı yanacaktır. Eğer bağlantı kurulamamışsa “Dur” butonuna basılmalı ve ardından “Sıfırla” butonuna basarak bir kez daha bağlanmayı denemelidir. mAs metre ve bilgisayar arasındaki mesafenin 10-15 metreyi aşmamasına dikkat edilmeli, mümkünse bilgisayarın bluetooth modülü mAs metre’yi direkt görecektir şekilde görüş hattı üzerine yerleştirilmelidir. mAs metre’nin problemleri röntgen tüpünü besleyen yüksek gerilim trafosunun orta ucuna bağlandıktan sonra ölçüm alınabilir. mAs metre hangi ölçüm modunda kullanılmak isteniyorsa “Floroskopi” ya da “Röntgen” tuşları ile seçilmelidir. Bu şekilde istenildiği kadar ölçüm alınabilmektedir. Ekspozur verilmesiyle birlikte mAs metre otomatik olarak verileri bilgisayara gönderecek, veriler sol alt taraftaki mavi zeminli metin kutusunda akacaktır.



Şekil 8.9. Grafik çizildikten sonraki ekran görüntüsü

Verinin alınması bittiğinde sağ taraftaki farklı renklerdeki Ölçüm butonlarından birine basılarak veri seti ilgili kayıt hafızasına saklanmış olacaktır. “Ölçüm” butonuna basılmasının ardından “Veri Alındı” mesajı görüntülenecektir. Mesajın görüntülenmesiyle birlikte ölçüm sonuçları miliamper, milisaniye ve mAs olarak aynen kalibrasyon cihazının ekranında görüntülediği şekliyle ilgili alanlarda belirecektir. Veri saklanan “Ölçüm” butonlarının solundaki seçim kutuları aktif hâle gelir. Yeniden “Röntgen” ya da “Floroskopi” butonlarından birine basılarak bir başka ölçüm alınabilir. Bu şekilde istenildiği kadar ölçüm alındıktan sonra “Grafığe Ekle” olarak isimlendirilen seçim kutularından karşılaştırılacak olan veriler belirlenmeli ve önce “Kaydet” sonra “Grafik Çiz” butonlarına basılarak grafikler görüntülenebilir. Şekil 8.9’da iki ölçüm sonucunun karşılaştırıldığı grafiğe ilişkin ekran görüntüsü verilmiştir. Bu ekranda istenirse “Özel Grafik Çözünürlüğü” kısmındaki ayarlardan yatay ve düşey piksel sayıları belirlenerek istenilen çözünürlükte geniş grafik elde etmek mümkündür. Bunun için çözünürlüğü belirledikten sonra kontrol modülü üzerindeki grafiğe tek tıklamak yeterlidir. Büyük grafik Windows resim görüntüleyicisinde açılacaktır. Şekil 8.10’da büyük grafiğin bir parçası

görüntülenmiştir. Eğer floroskopi modunda ölçüm alınmışsa veri miktarı çok olacağından ana ekrandaki grafik 32000 verilik parçalara bölünür ve parçalar arasında “ileri” “geri” butonları ile gezilebilir. Her parçaya ait büyük grafik çizdirmek mümkündür. Çok uzun süreli ölçümler alınmışsa grafiklerin çizilmesi birkaç dakika sürebilir. Grafik oluşturulma sürecinde durumu gösteren bir ilerleme çubuğu işlemin tamamlanmasına ne kadar süre kaldığı ile ilgili kabaca bir fikir verecektir.



Şekil 8.10. Büyütülmüş grafik

Alınan her ölçüm sonucu “c:\masmetre” klasörü altında kaydedilmekte, bu verilerin istendiği takdirde farklı amaçlarla kullanılması mümkün olmaktadır. Gerektiğinde önceden alınmış veriler kontrol programına tekrar yüklenerek yeni kombinasyonlarla grafikler çizilebilmekte, istenirse de önceden çizilmiş grafikler çağrılabilir. Ölçüm işlemi bittiğinde “Dur” butonuna basılarak seri port kapatılabilir ya da direkt olarak programdan çıkılabilir.

## 9. SONUÇ VE ÖNERİLER

Bu tez çalışması kapsamında imal edilen ve Röntgen Tüpü kullanan görüntüleme cihazlarının tüp akımını, ekspozür süresini ve bunların çarpımı şeklinde ifade edilen toplam doz miktarını veren mAs değerlerini gerçek zamanlı olarak yüksek doğrulukla hesaplayan kalibrasyon cihazının bugüne kadar gerek piyasada gerek akademik literatürde yer almamış ek özellikleri bulunmaktadır. Bu yenilikler gelişen teknolojinin imkânları kullanılarak başarılan ve piyasadaki önceki ticari ürünlerde bulunmayan yüksek kararlılık opamp blokları ile sinyalin doğru biçimde ve farklı ölçüm aralıklarında kademeli olarak işlenmesi, gelişmiş ve hızlı ADC sayesinde yüksek çözünürlükte ve yüksek sayıda örneklemenin gerçek zamanlı olarak yapılabilmesi, floroskopi modunda süresiz ekspozür ölçümü alınabilmesi ve en önemlisi, ölçümlerin uzaktaki bir bilgisayardan kablosuz haberleşme ile başlatılabilmesi ve örnekleme verileri ile hesap sonuçlarının tez kapsamında bilgisayarda kullanılmak üzere geliştirilmiş mAs metre kontrol yazılımına aktarılabilmesi, alınan verilerin ve ölçüm sonuçlarının birbirleriyle karşılaştırmalı olarak zaman ekseninde grafiklerinin çizilebilmesi olarak gösterilebilir.

Tez çalışması neticesinde imal edilen cihaz ile x ışın tüpünü görüntüleme temeli olarak kullanan herhangi bir tıbbi görüntüleme cihazının kalibrasyonunun yapılması mümkün olmaktadır. Bu sayede hasta ve röntgen teknisyeninin aşırı şua alması önlenmiş olur. Bu yönüyle insan sağlığını dolaylı olarak korur. Ayrıca röntgen cihazlarının en önemli ve en pahalı parçalarından biri olan röntgen tüpünden fazla akım geçmesi suretiyle kısa zamanda tahrip olarak kullanılamaz hâle gelmesinin önüne geçilmiş olmaktadır. Bu yönüyle de ülke ekonomisine katkıda bulunmaktadır.

mAs metre'nin yapımı aşamasında MAX038 entegresi ile yapılan sinyal üretici üzerinden çeşitli akım sinyalleri suni olarak verilmiş, mAs metre bu sinyallerin ışığında imal edilmiştir. Ekspozür süresinin ölçülmesindeki doğruluk ise Tektronix AFG-310 keyfi sinyal üretici ile üretilen burst mod sinyal dizileri ile sınanmıştır. Tüm tasarım aşaması sonuçlandıktan sonra medikal cihazların kalibrasyonunu yapan

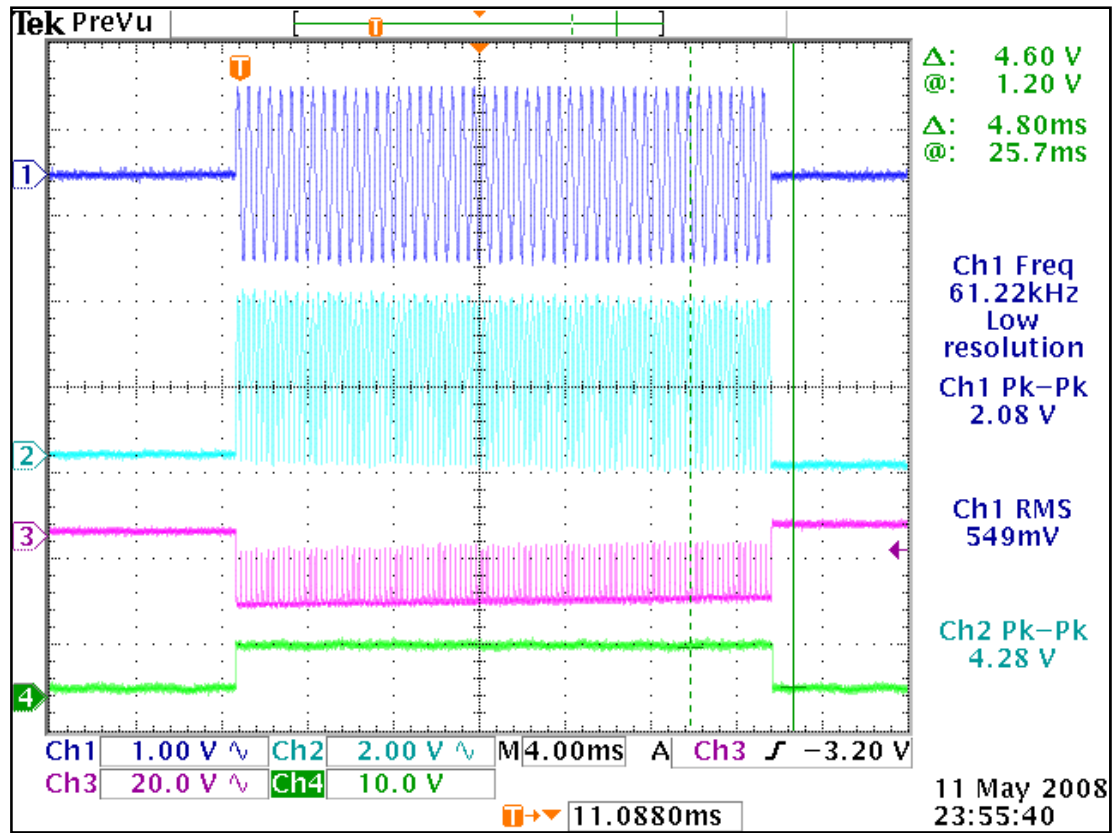
ticari bir firmanın laboratuvarında yüksek hassasiyetli kalibrasyon cihazları ile son ince ayarlar tamamlanmıştır. Bu esnada farklı frekans ve genliklerde rms değeri bilinen darbe dizileri gönderilmiş ve mas metre'nin ölçtüğü sonuçla karşılaştırılmıştır.

Kalibrasyon süreci tamamlandıktan sonra Ankara Üniversitesi Nükleer Bilimler Enstitüsü Diagnostik Radyoloji Görüntüleme Laboratuvarında SPECTUM marka C Kollu Floroskopi cihazı üzerinde ölçümler alınarak Floroskopi Modu denenmiş, Picker Anatomatic-40 marka konvansiyonel x ışın görüntüleme cihazı üzerinde de cihazın Normal Ölçüm Modu denenmiştir. Bu ölçümler esnasında Radcal X-Ray Dozimetre ile süre ve doz ölçümleri alınarak karşılaştırılmıştır. Çizelge 9.1'de bazı ölçüm sonuçları verilmiştir.

Çizelge 9.1. Konvansiyonel x ışın cihazı üzerinde alınan bazı ölçüm değerleri

Ölçümler	Büyüklik	Kumanda Masasından Ayarlanan	Radcal Dozimetre	mAs metre
Ölçüm 1	Akım (mA)	100	99,7	108,179771
	Zaman (ms)	500	498	504,227325
	Doz (mAs)	50	49,7	54,547196
Ölçüm 2	Akım (mA)	200	190	206,870300
	Zaman (ms)	100	96	102,354874
	Doz (mAs)	20	18,3	21,174183
Ölçüm 3	Akım (mA)	400	409	406,540405
	Zaman (ms)	100	101	104,997620
	Doz (mAs)	40	41,4	42,685774

Yapılan ölçmeler sonucunda, kumanda masasından ayarlanan mA ve ms süreleri ile mAs metre'nin ölçtüğü değerler ve radcal dozimetrenin gösterdiği değerler arasında belli bir fark olduğu görülmüştür. Dijital olarak üretilen rms değerleri ve süresi bilinen test sinyallerine göre ayarları yapılmış olan mas metrenin doğruluğu kademelere göre değişmekle birlikte alınan farklı ölçümler içinde en az 10 000'de bir, en çok %1,5 olarak gözlenmiştir.



Şekil 9.1. mAs metre'nin kalibrasyonu esnasında alınan ölçüm ekranı

Sonuç olarak tasarıma başlanmadan önce hedeflenen doğruluk değerlerinin üzerine çıkmış, istenen ölçüm aralıklarından fazlasını ölçen ve bu değerleri kablosuz olarak gerçek zamanda bilgisayara daha ileriki araştırmalar için işlenmeye hazır biçimde gönderen uzaktan kontrol edilebilir mAs metre cihazımız son ürün olarak kullanıma hazır bir şekilde tüm donanımı ve yazılımı ile birlikte kendi tasarımı olarak gerçekleştirilmiştir.

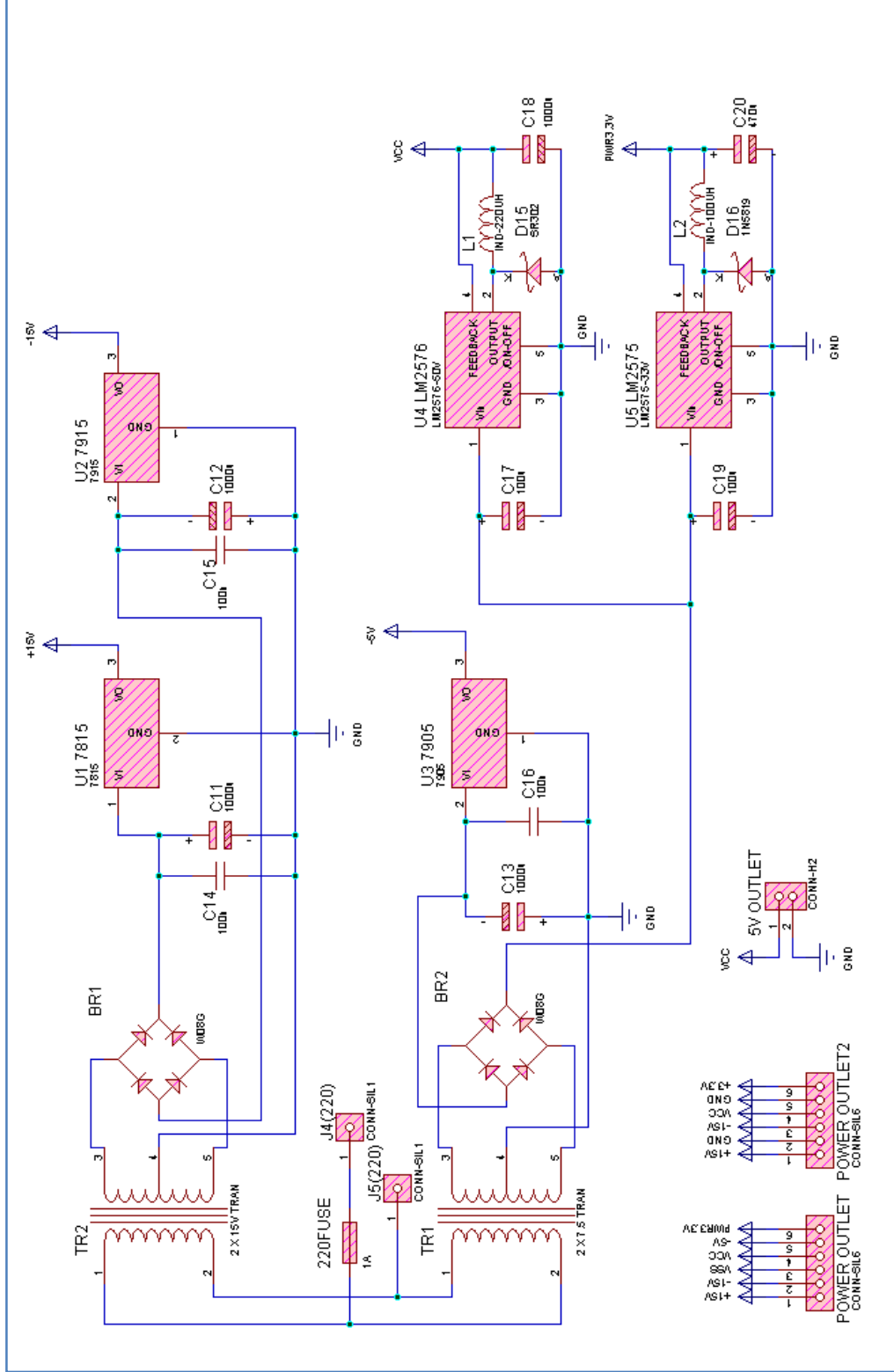
Bu konuda daha ileri arařtırmalar yapacak arařtırmacıların, röntgen tpnn akımı ile birlikte gerilimini de lmeleri cihaza ayrı bir iřlevsellik kazandıracaktır. Bunun bařarılması iin yksek gerilim blclerden ya da non-invasive metotla iyon odası dedektrlerinden faydalanılabılır. Giriřimsiz lm sensrleri yarı iletken teknolojisi ile gerekleřtirilebilmektedir, bunun da cihaza eklenmesi durumunda gnmz Őartlarında yaklařık olarak 30 bin amerikan dolarına satılmakta olan ticari rnlere eřdeęer bir cihaz elde edilmiř olacaktır.

## KAYNAKLAR

1. Shroy, R.E.Jr., Van Lysel, M.S., Yaffe, M.J. “X-Ray: The Biomedical Engineering Handbook Second Edition”, **CRC Press LLC**, Boca Raton, 1-37 (2000).
2. Wolbarst, A.B., “Looking Within: How X-Ray, CT, MRI, Ultrasound and Other Medical Images Created, and How They Help Physicians Save Lives”, **University of California Press**, Berkeley and Los Angeles, 1-109 (1999)
3. Whaites, E., “Essentials of Dental Radiography and Radiology Third Edition”, **Churchill Livingstone**, London, 1-414 (2003)
4. United States Department of Energy, “Radiological Safety Training for Radiation Producing (X-RAY) Devices”, **DOE, WSC6910**, 1-92 (2002)
5. Atakan, Y., “Radyasyon Fiziği Temel Kavramları ve Birimleri”, **Beta Yayın ve Dağıtım A.Ş.**, İstanbul, 1-236 (1995).
6. Berk, U., “Radyoloji Biliminin Temelleri”, **Uğur Yayınları**, Ankara, 65-98 (1997).
7. Çokyüksek, O., “Radyoloji Bilimi ve Röntgen Cihazları”, **Uğur Yayınları**, Ankara, 55-132 (1992)
8. Microchip Technology Incorporated, “dsPIC30F4011/4012 Data Sheet” **Microchip, DS70135E**, 1-236 (2007)
9. Microchip Technology Incorporated, “dsPIC30F Family Reference Manual” **Microchip, DS70064D**, 1-59 (2005)

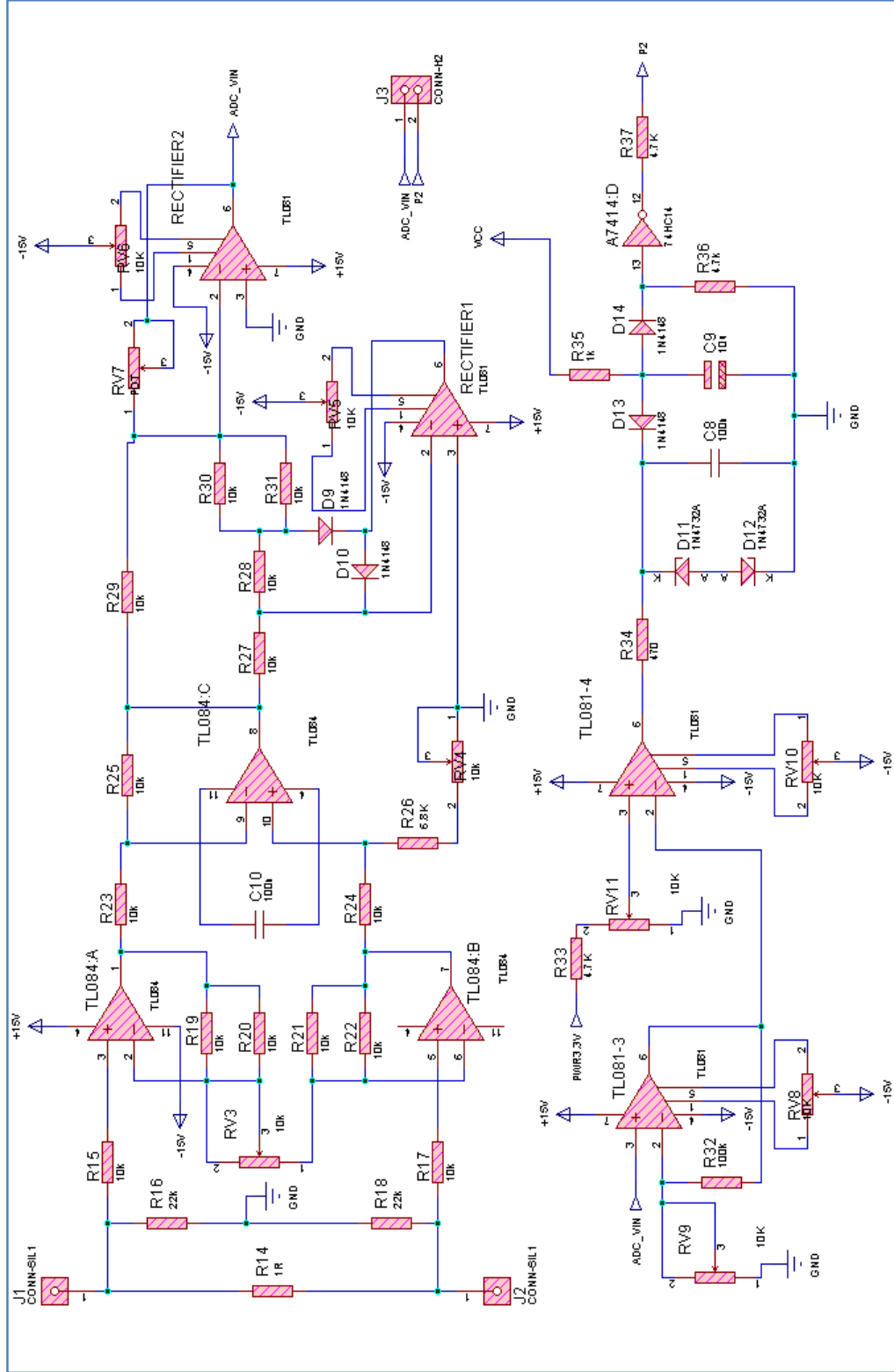
**EKLER**

EK-1 Besleme katı devre şeması



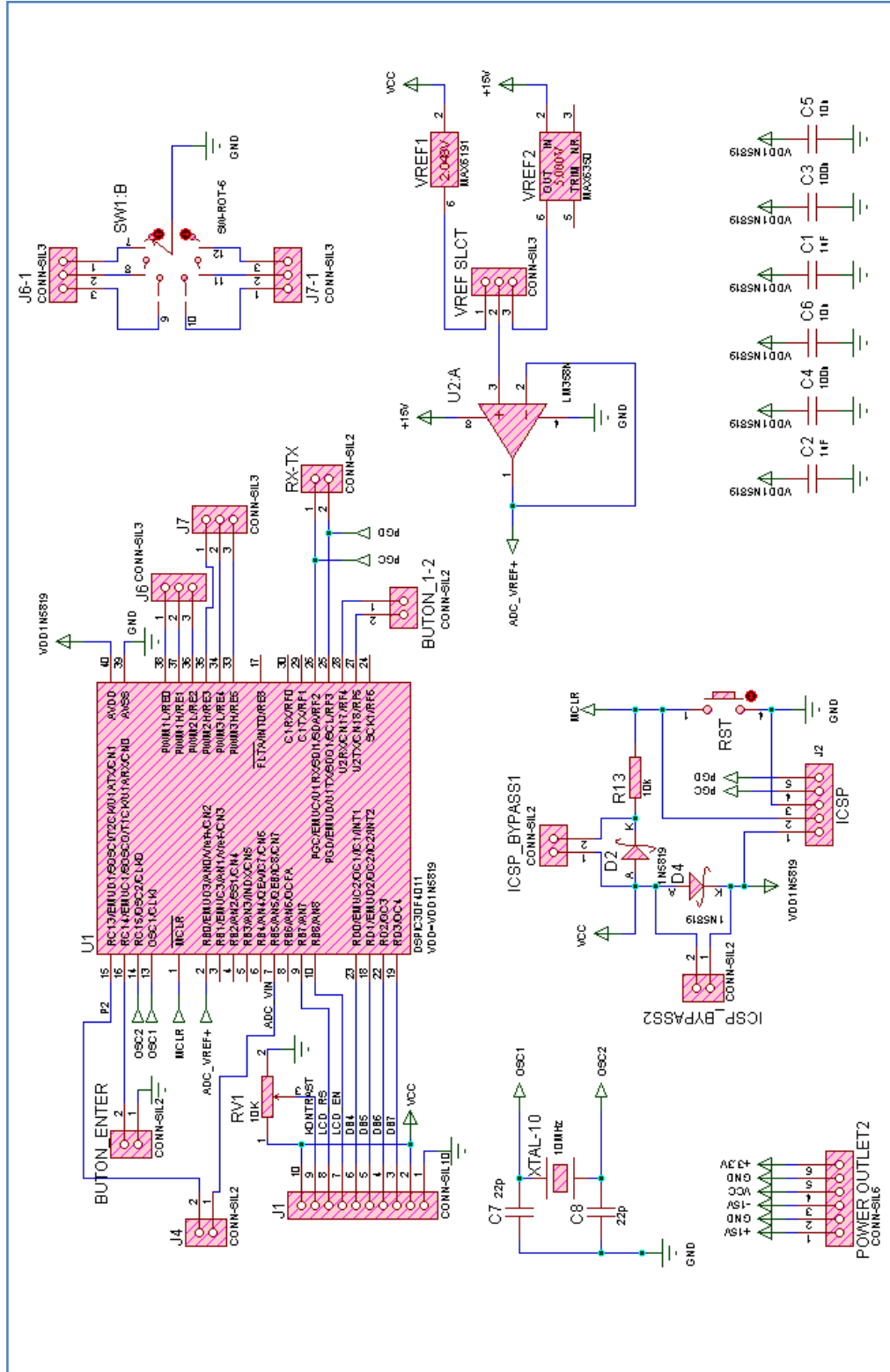
Şekil 1.1. Anahtarlamalı ve lineer güç entegreleri ile tasarlanmış +3,3V; ±5V; ±15V güç katı devre şeması

EK-2 Opamp blokları devre şeması



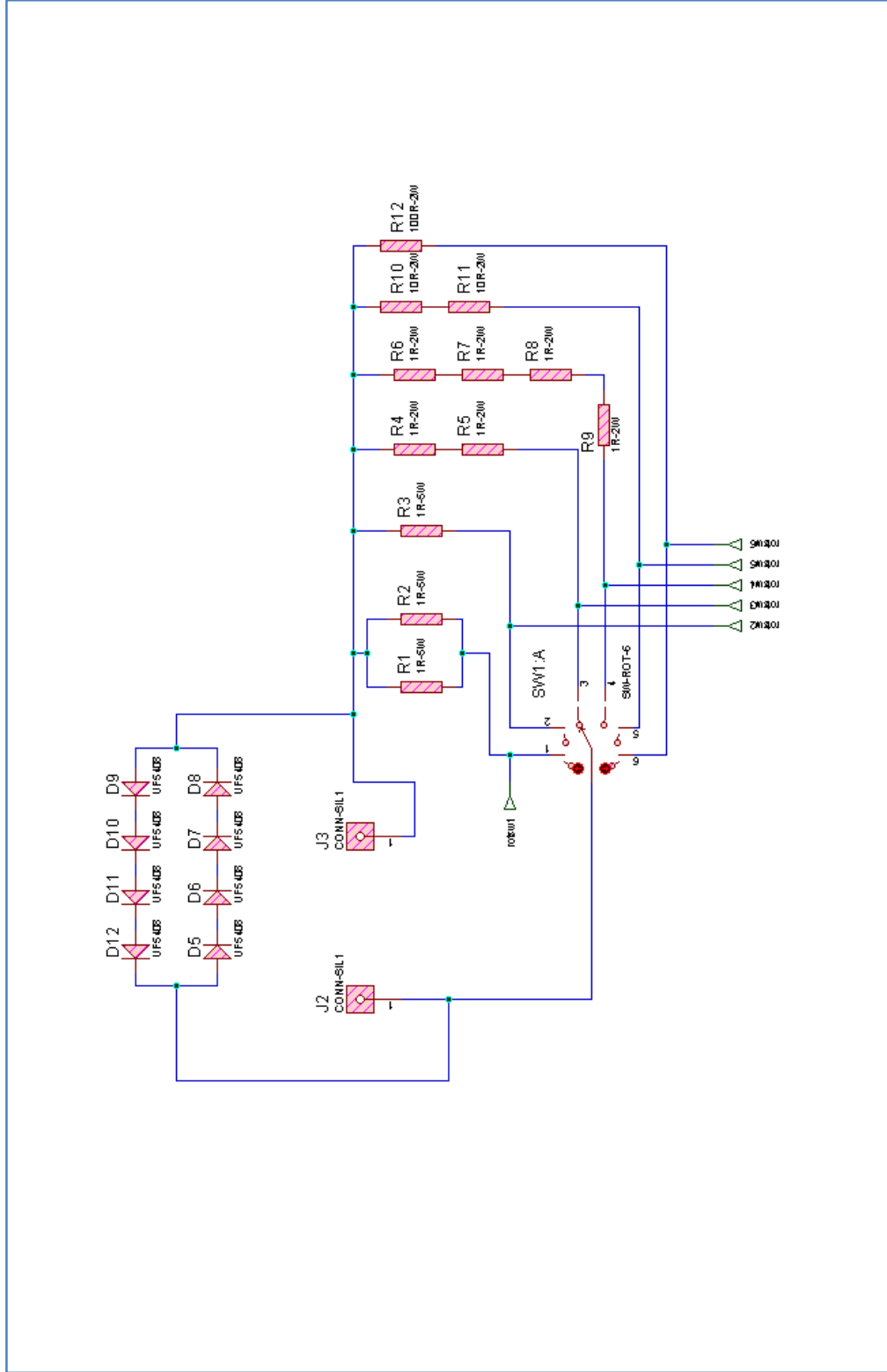
Şekil 2.1. Sinyal düzenleme ve filtreleme katı devre şeması

EK-3 dsPIC katı devre şeması



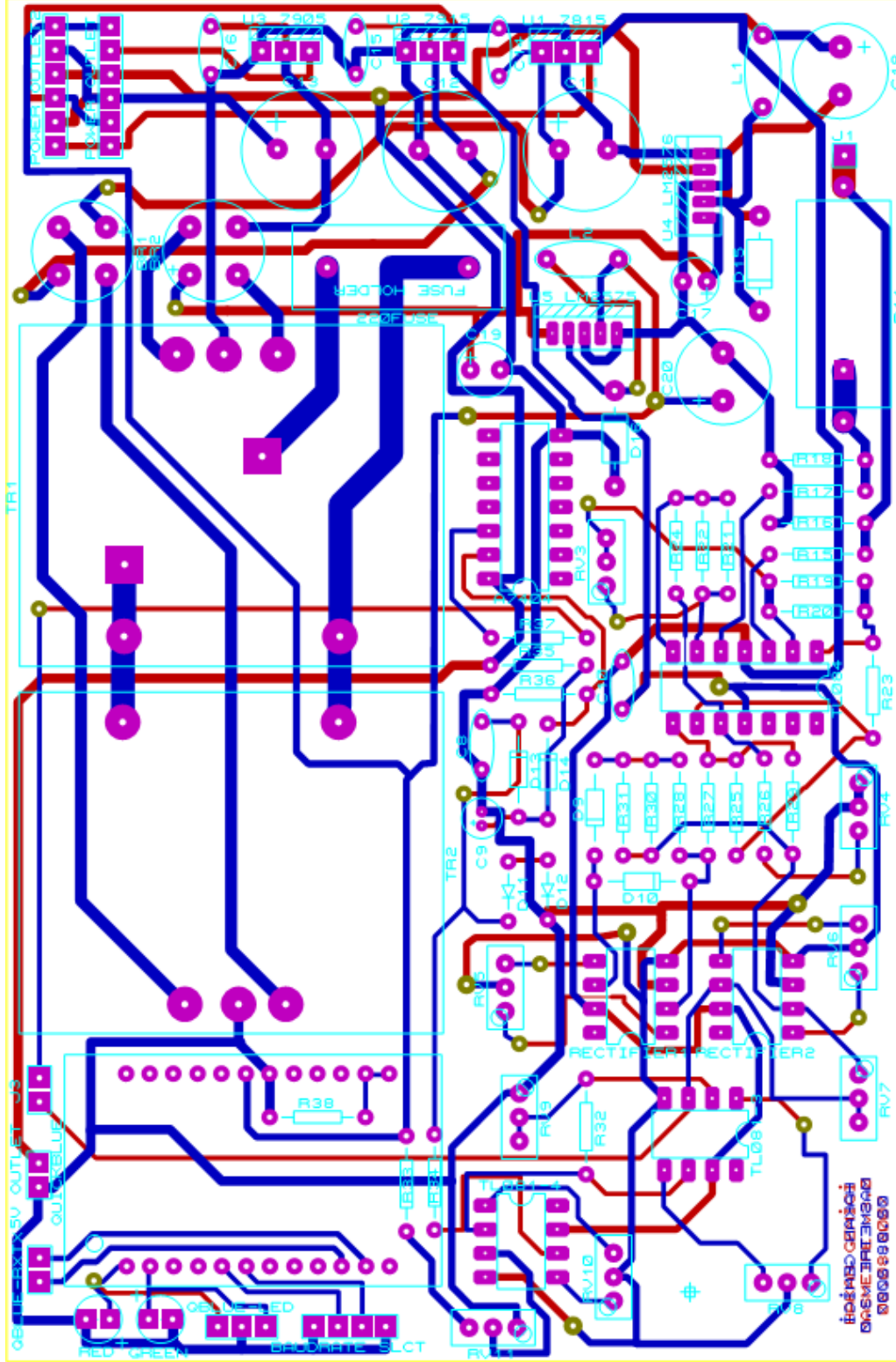
Şekil 3.1. Mikrodeneleyici katı devre şeması

## EK-4 Kademe seçici komütatör devre şeması



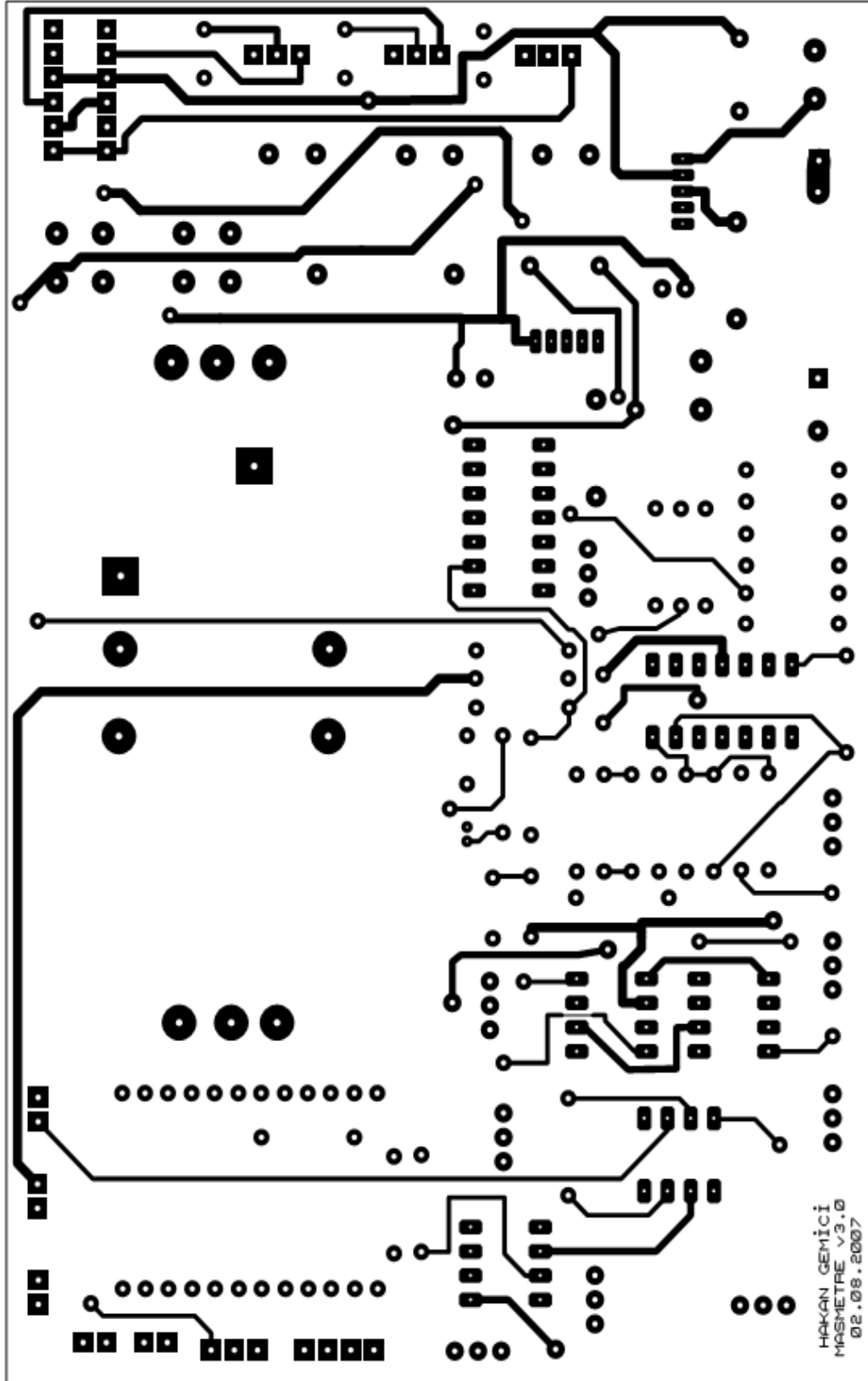
Şekil 4.1. Sinyal ölçmeleme katı devre şeması

EK-5 Analog kart tüm katmanlar



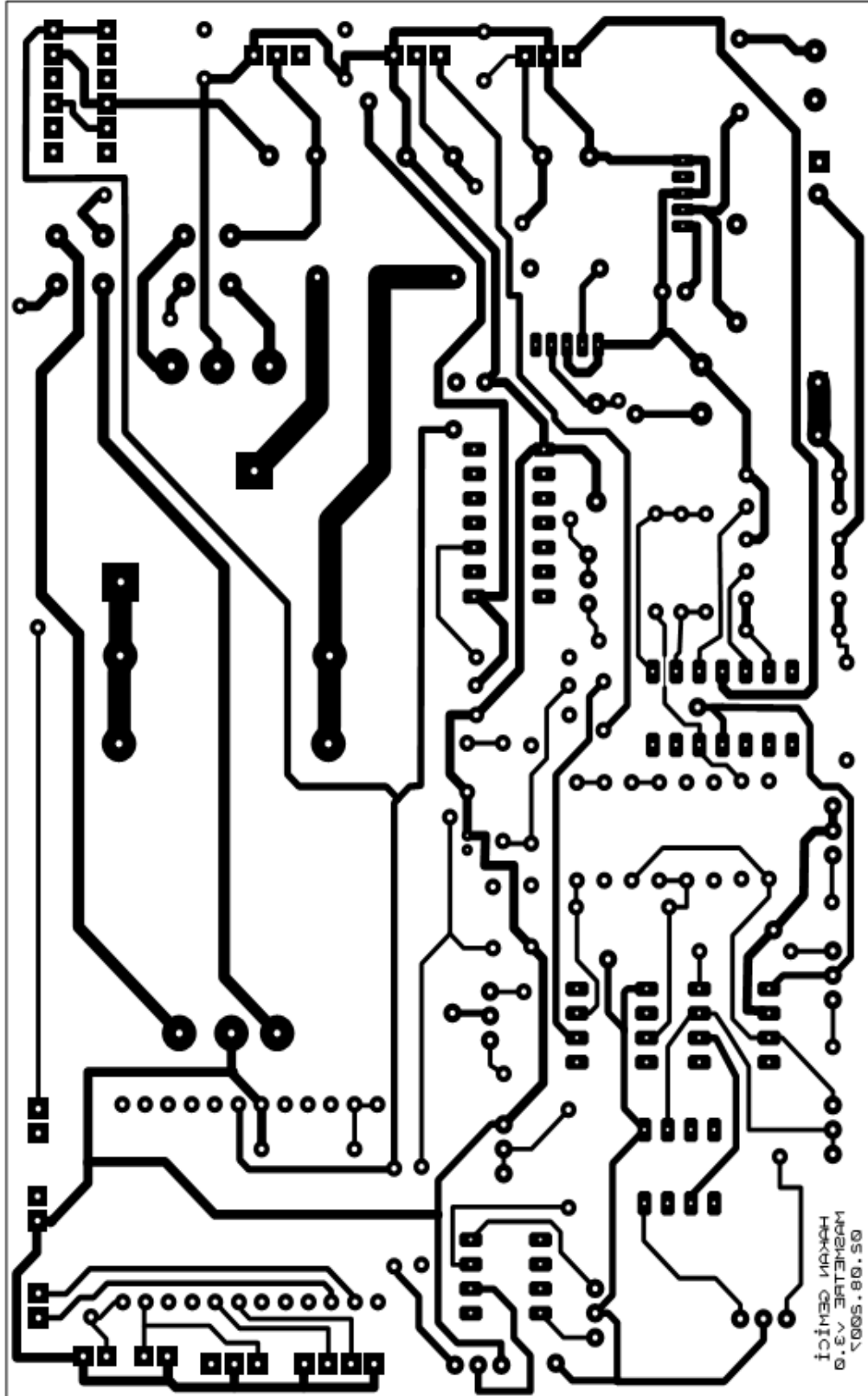
Şekil 5.1. Analog baskılı devre kartı tüm katmanlar

EK-6 Analog kart baskılı devre üst yüz



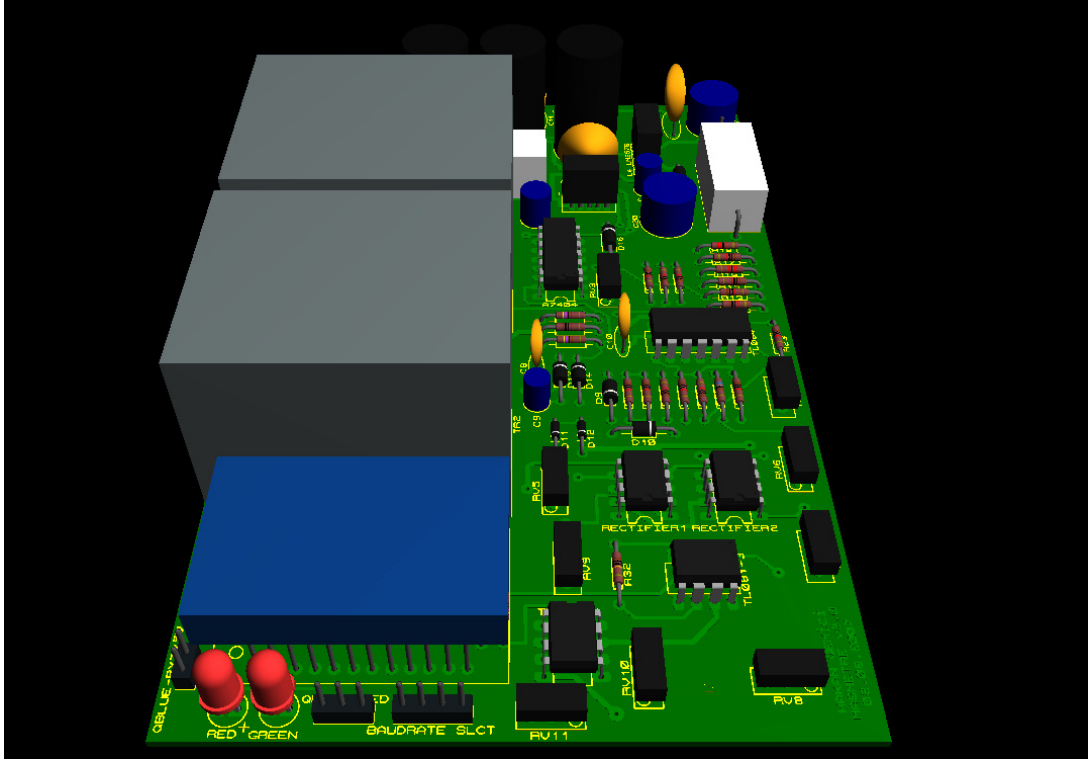
Şekil 6.1. Analog baskılı devre kartı üst yüz

EK-7 Analog kart baskılı devre alt yüz

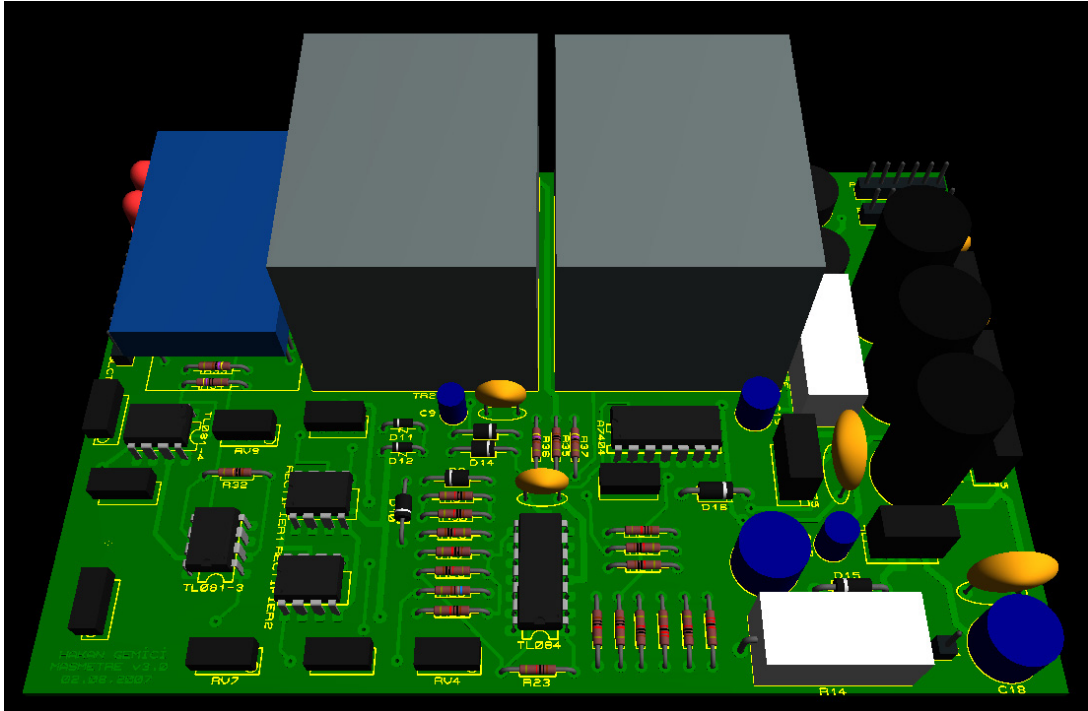


Şekil 7.1. Analog baskılı devre kartı alt yüz

### EK-8 Analog kart pcb 3D modellemesi



Şekil 8.1. Analog baskılı devre kartı üç boyutlu modelleme yandan görünüm



Şekil 8.2. Analog baskılı devre kartı üç boyutlu modelleme önden görünüm

## EK-9 Analog kart malzeme listesi

besleme-analog.DSN için malzeme listesi

Design Title : besleme-analog.DSN  
 Author : Hakan Gemici  
 Revision : 4.1  
 Design Created : 28 Temmuz 2007 Cumartesi  
 Design Last Modified : 02 Ocak 2008 Çarşamba  
 Total Parts In Design : 87

## 25 Resistors

Quantity:	References	Value
1	R14	1R
14	R15, R17, R19-R25, R27-R31	10k
2	R16, R18	22k
1	R26	6.8K
1	R32	100k
3	R33, R36, R37	4,7K
1	R34	470
2	R35, R38	1k

## 13 Capacitors

Quantity:	References	Value
5	C8, C10, C14-C16	100n
1	C9	10u
4	C11-C13, C18	1000u
2	C17, C19	100u
1	C20	470u

## EK-9 (Devam) Analog kart malzeme listesi

## 5 Integrated Circuits

Quantity:	References	Value
1	U1 7815	7815
1	U2 7915	7915
1	U3 7905	7905
1	U4 LM2576	LM2576-5,0V
1	U5 LM2575	LM2575-3,3V

## 8 Diodes

Quantity:	References	Value
4	D9, D10, D13, D14	1N4148
2	D11, D12	1N4732A
1	D15	SR302
1	D16	1N5819

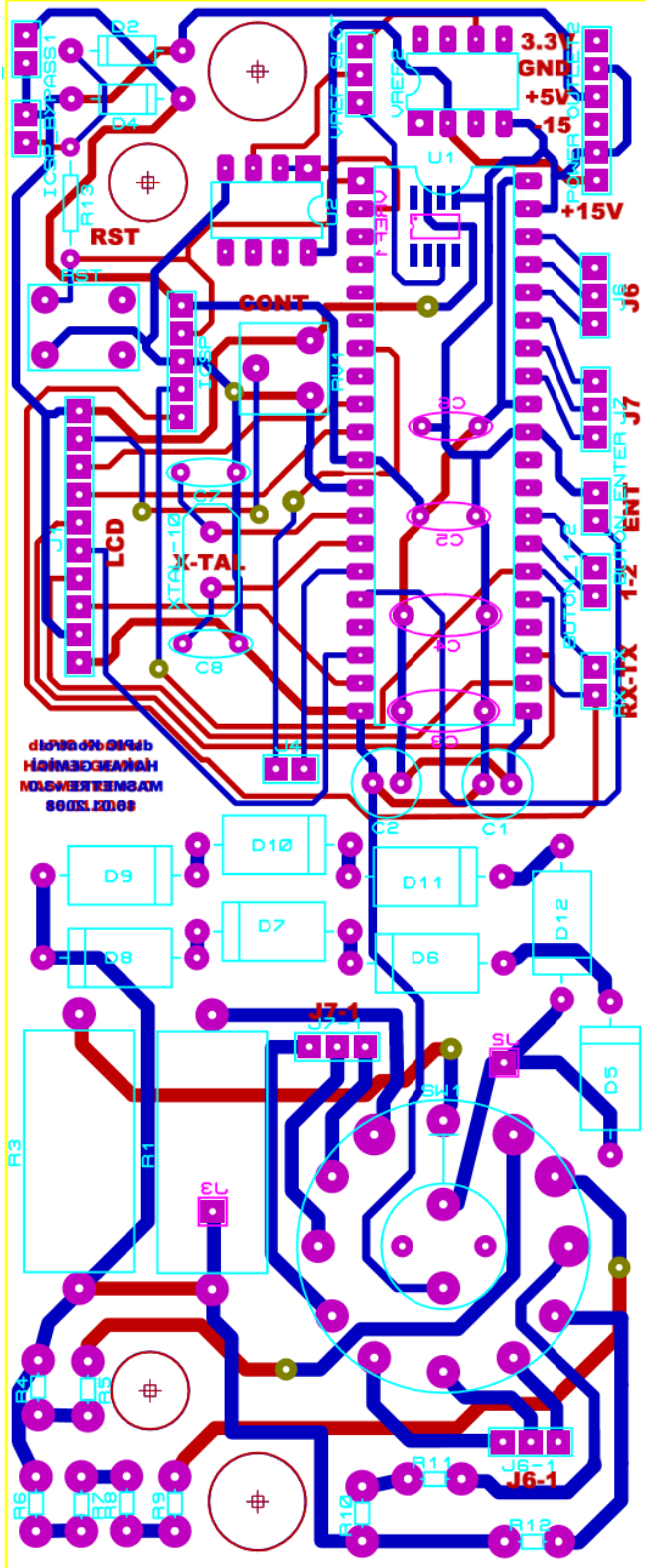
## 36 Miscellaneous

Quantity:	References	Value
1	A7414	74HC14
1	BAUDRATE SLCT	J4
2	BR1, BR2	W08G
1	FUSE_1A	1A
2	GREEN, RED	LED
4	J1, J2, J4(220), J5(220)	CONN-SIL1
3	J3, OUTLET_5V, QBLUE-RXTX	CONN-H2
1	L1	IND-220UH
1	L2	IND-100UH

## EK-9 (Devam) Analog kart malzeme listesi

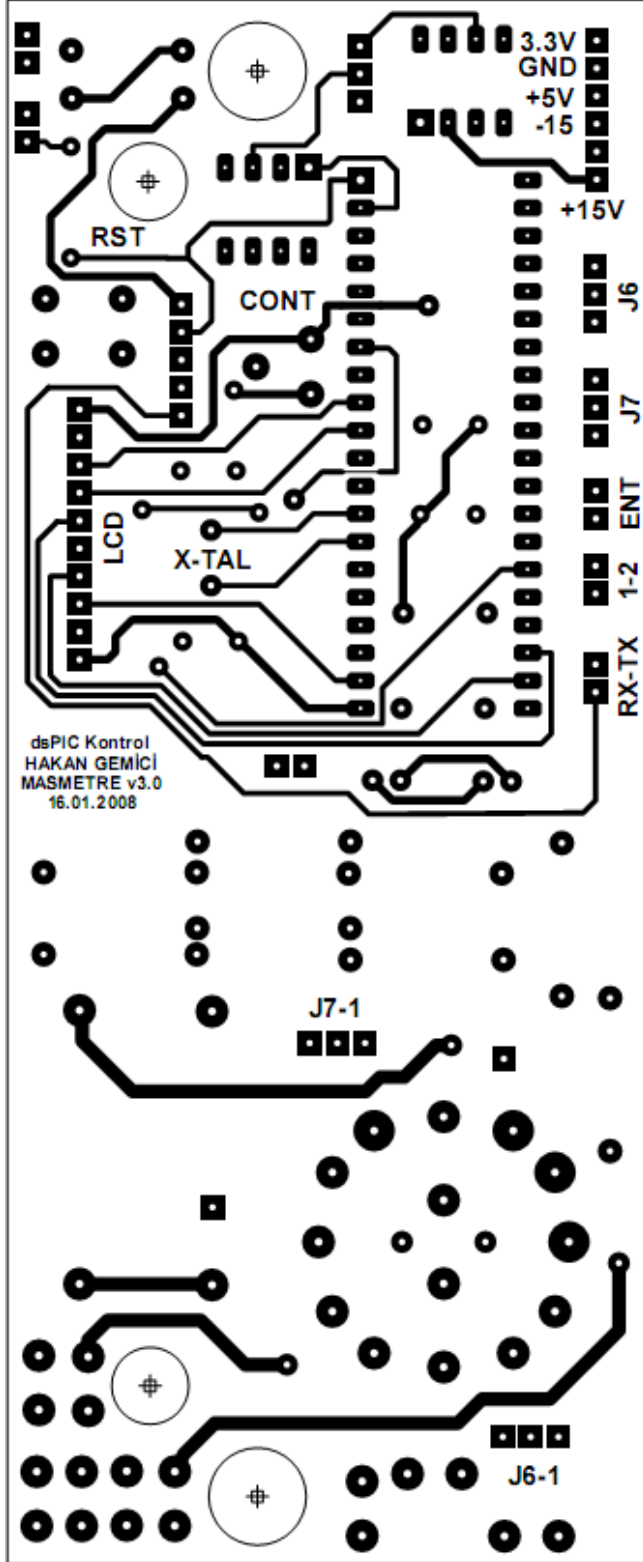
2	POWER OUTLET, POWER OUTLET2	CONN-SIL6
1	QBLUE-LED	CONN-SIL3
1	QUICKBLUE	QUICKBLUE
4	RECTIFIER1, RECTIFIER2, TL081-3, TL081-4	TL081
8	RV3-RV6, RV8-RV11	10k
1	RV7	POT
1	TL084	TL084
1	TR1	2 X 7,5 TRAN
1	TR2	2 X 15V TRAN

EK-10 dsPIC kartı tüm katmanlar



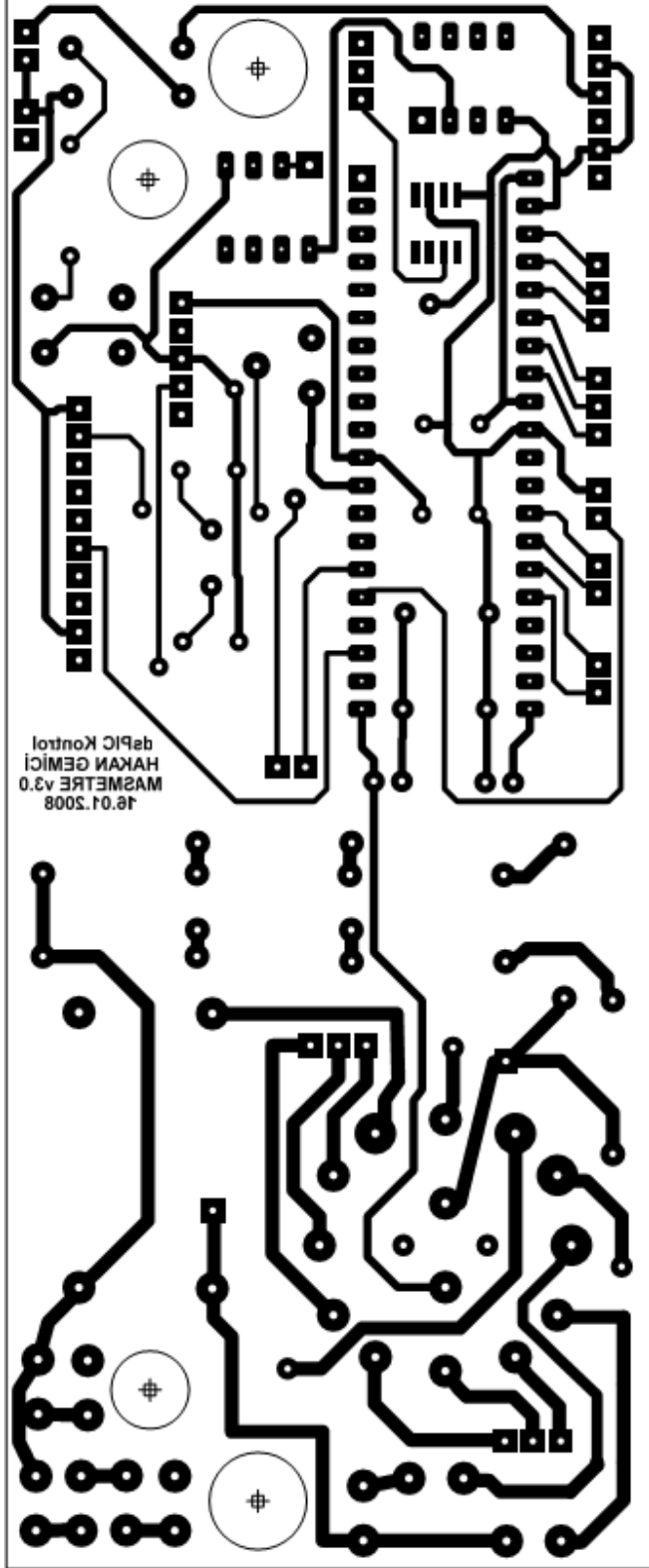
Şekil 10.1. Mikrodenetleyici ve sinyal ölçkleme kartı tüm katmanlar

EK-11 dsPIC kartı baskılı devre üst yüz



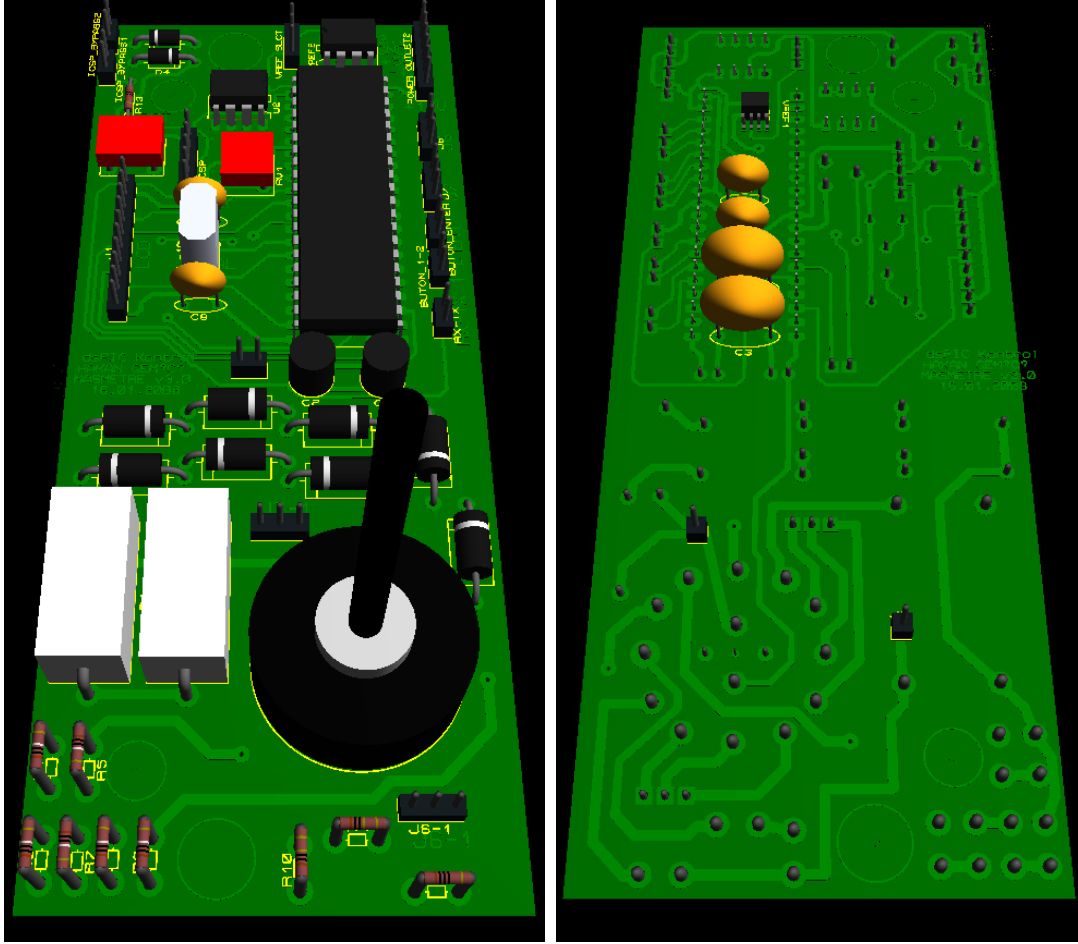
Şekil 11.1. Mikrodenetleyici ve sinyal ölçekleme kartı üst katman

EK-12 dsPIC kartı baskılı devre alt yüz

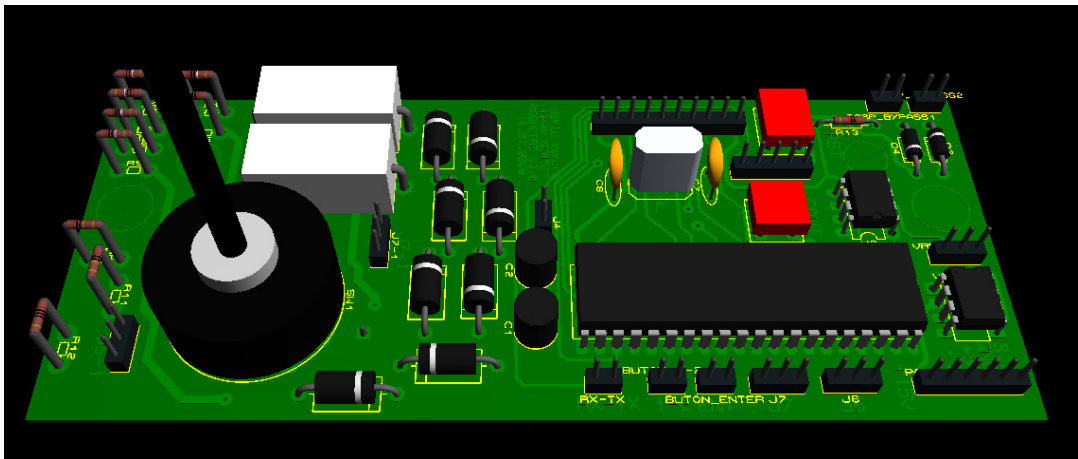


Şekil 12.1. Mikrodenetleyici ve sinyal ölçkleme kartı alt katman

EK-13 dsPIC kartı pcb 3D modellemeleri



Şekil 13.1.Mikrodenetleyici kartı üç boyutlu modelleme üstten ve alttan görünüm



Şekil 13.2.Mikrodenetleyici kartı üç boyutlu modelleme üst yandan görünüm

## EK-14 dsPIC kartı malzeme listesi

sayısal dspic.DSN için malzeme listesi

Design Title : sayısal dspic.DSN  
 Author : Hakan Gemici  
 Revision : 3.2  
 Design Created : 02 Ocak 2008 Çarşamba  
 Design Last Modified : 17 Nisan 2008 Perşembe  
 Total Parts In Design : 55

## 13 Resistors

Quantity:	References	Value
3	R1-R3	1R-5W
6	R4-R9	1R-2W
2	R10, R11	10R-2W
1	R12	100R-2W
1	R13	10k

## 8 Capacitors

Quantity:	References	Value
2	C1, C2	1uF
2	C3, C4	100n
2	C5, C6	10n
2	C7, C8	22p

## 2 Integrated Circuits

Quantity:	References	Value
1	U1	DSPIC30F4011

## EK-14 (Devam) dsPIC kartı malzeme listesi

1	U2	LM358N
---	----	--------

## 10 Diodes

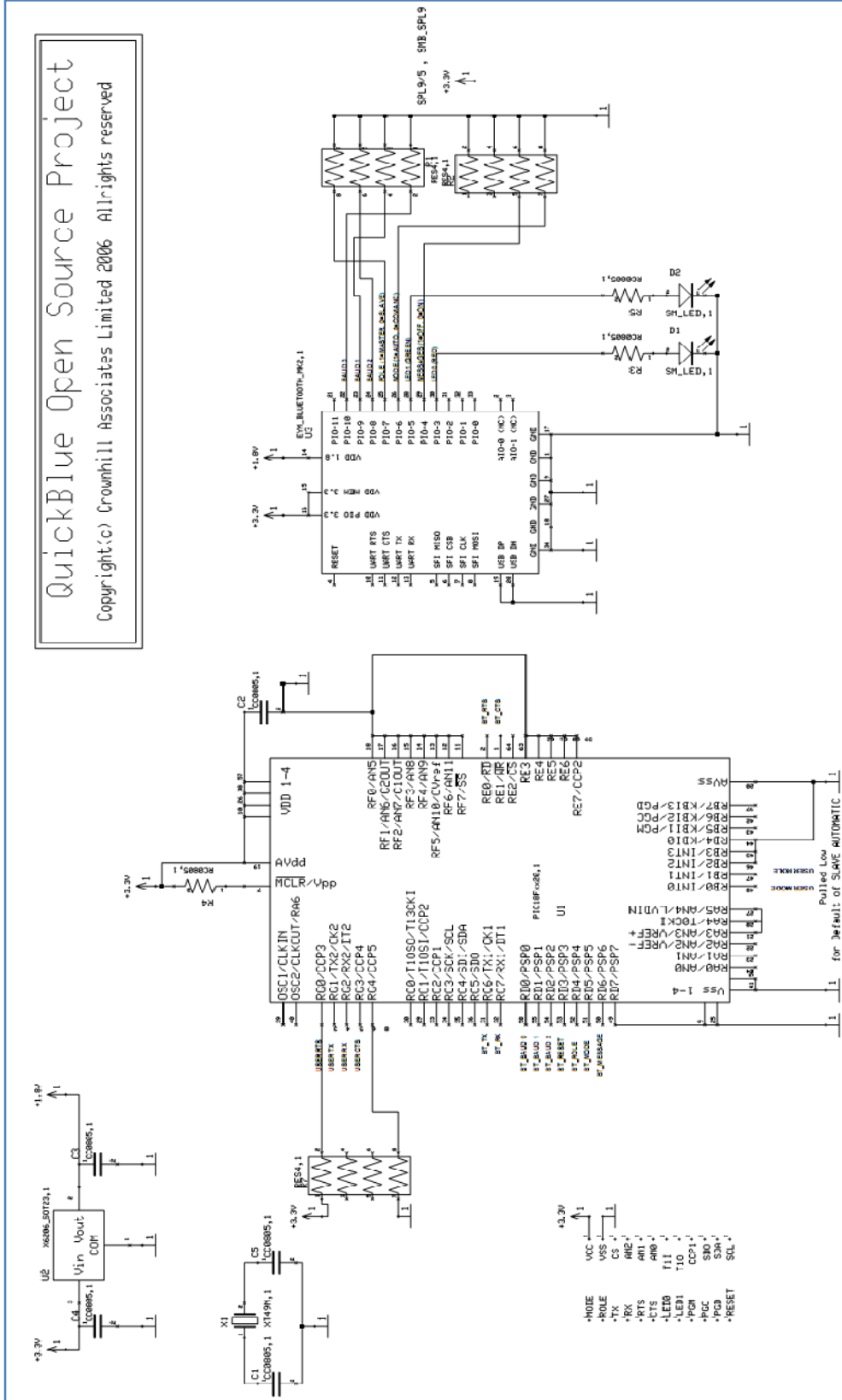
Quantity:	References	Value
2	D2, D4	1N5819
8	D5-D12	UF5408

## 22 Miscellaneous

Quantity:	References	Value
6	BUTON_1-2, BUTON_ENTER, ICSP_BYPASS1, ICSP_BYPASS2, J4, RX-TX	CONN-SIL2
1	ICSP	J2
1	J1	CONN-SIL10
2	J2, J3	CONN-SIL1
5	J6, J6-1, J7, J7-1, VREF SLCT	CONN-SIL3
1	POWER OUTLET2	CONN-SIL6
1	RST	
1	RV1	10K
1	SW1	SW-ROT-6
1	VREF1	MAX6191
1	VREF2	MAX6350
1	XTAL-10	10MHz



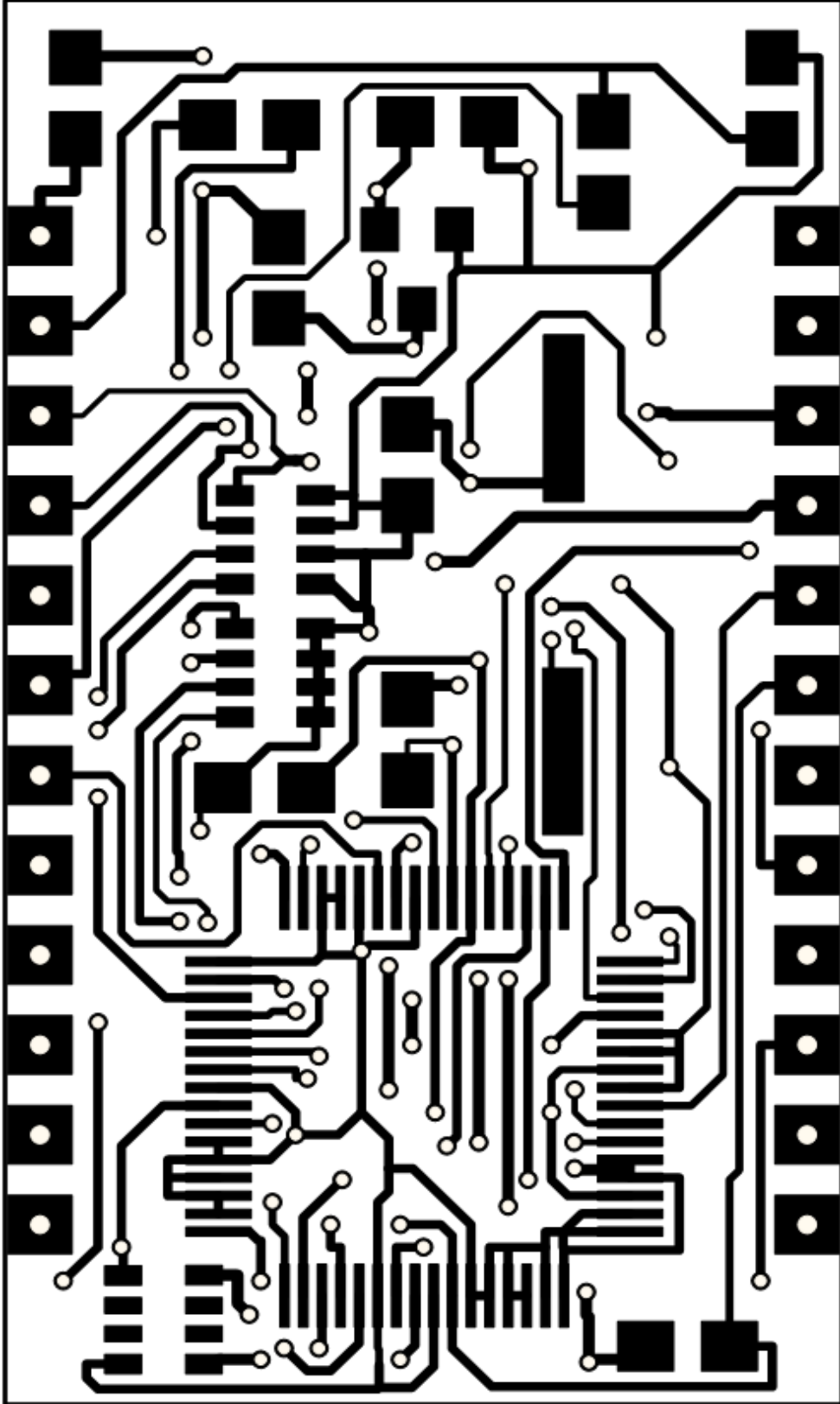
EK-16 Bluetooth modülü bağlantı şeması



QuickBlue Open Source Project  
 Copyright(c) Crownhill Associates Limited 2006 All rights reserved

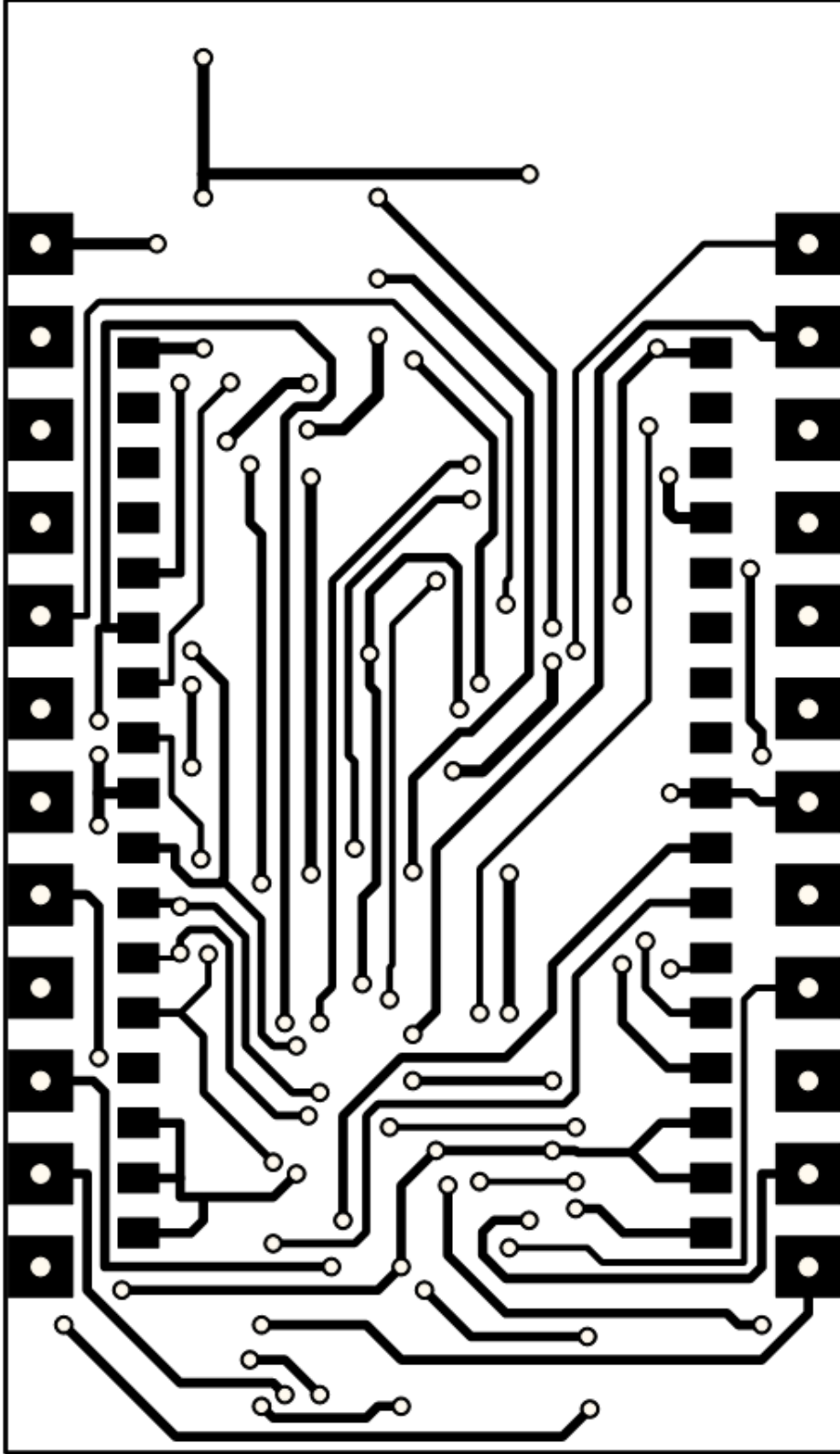
Şekil 16.1. Bluetooth modülü bağlantı şeması

EK-17 Bluetooth modülü baskılı devre üst yüz



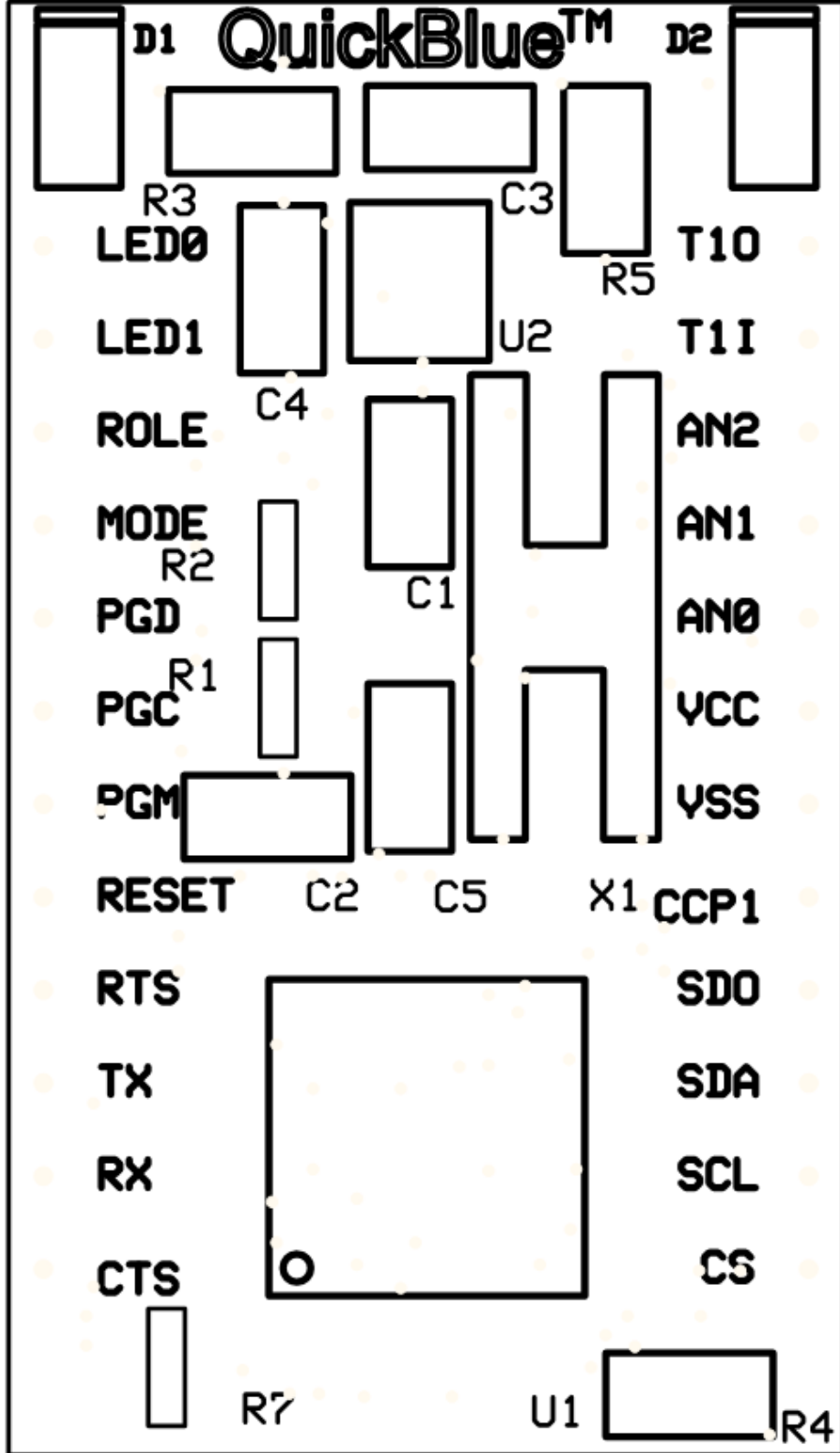
Şekil 17.1. Bluetooth modülü baskılı devre üst katman

EK-18 Bluetooth modülü baskılı devre alt yüz



Şekil 18.1. Bluetooth modülü baskılı devre alt katman

EK-19 Bluetooth modülü baskılı devre eleman yerleşimi



Şekil 19.1.1. Bluetooth modülü bağlantı şeması

## EK-20 dsPIC30F4011 için C kodu

```

#include "p30fxxx.h"
#include "p30f4011.h"
#include <stdio.h>
#include <math.h>

_FOSC(CSW_FSCM_OFF & XT_PLL8); //Run this project using an external crystal
//routed via the PLL in 16x multiplier mode
//For the 7.3728 MHz crystal we will derive a
//throughput of  $7.3728e+6 * 16 / 4 = 29.4$  MIPS(Fcy)
//,~33.9 nanoseconds instruction cycle time(Tcy).
_FWDT(WDT_OFF); //Turn off the Watch-Dog Timer.
_FBORPOR(MCLR_EN & PWRT_OFF); //Enable MCLR reset pin and turn off the
//power-up timers.
_FGS(CODE_PROT_OFF); //Disable Code Protection

// gercek kart

#define LCDPORT PORTD
#define LCD_RS PORTBbits.RB7
#define LCD_EN PORTBbits.RB8

// dev board
/*
#define LCDPORT PORTB
#define LCD_RS PORTDbits.RD0
#define LCD_EN PORTDbits.RD1
*/
#define IDLE 0
#define START_ADC 1
#define STOP_ADC 2
#define ADC_CONVERSION_DONE 3
#define ADC_CONVERSION_WAITING 4
#define USER_EVENT 5
#define NO_EVENT 6
#define BUTTON0_PRESSED 7
#define BUTTON1_PRESSED 8
#define BUTTON2_PRESSED 9
#define OLCME_TAMAMLANDI 10

#define BUTTON0 PORTFbits.RF4
#define BUTTON1 PORTFbits.RF5
#define BUTTON2 PORTCbits.RC14
#define BUTTON3 PORTCbits.RC13

#define LCD_LINE0 0x80
#define LCD_LINE1 0xc

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

#define LCD_LINE2    0x94
#define LCD_LINE3    0xd4
#define ANA_MENU     41
#define NORMAL_OLCUM    42
#define FLOROSKOPI_OLCUMU    43
#define AYARLAR      44

#define NORMAL_OLCUMU_BASLAT  45
#define NORMAL_KAYIT_GOSTER   46
#define NORMAL_SONUCU_GOSTER  47
#define NORMAL_KAYIT         48
#define NORMAL_KAYIT_ONAYLANDI 49
#define NORMAL_KAYIT_GETIR    50

#define FLOROSKOPI_OLCUMUNU_BASLAT 51
#define FLOROSKOPI_KAYIT_GOSTER    52
#define FLOROSKOPI_SONUCU_GOSTER   53
#define FLOROSKOPI_KAYIT          54
#define FLOROSKOPI_KAYIT_ONAYLANDI 55
#define FLOROSKOPI_KAYIT_GETIR    56

#define KADEME_DEGISTI 57
// karakterler
#define OK      126

#define YAPILAMAZ 0
#define YAPILABILIR 1

#define KADEME_SW0  PORTEbits.RE0
#define KADEME_SW1  PORTEbits.RE1
#define KADEME_SW2  PORTEbits.RE2
#define KADEME_SW3  PORTEbits.RE3
#define KADEME_SW4  PORTEbits.RE4
#define KADEME_SW5  PORTEbits.RE5

unsigned const int ADCBUFFERSIZE = 15;
unsigned const int SAYAC1_SIZE = 11;

unsigned long geneltoplamlam = 0;
unsigned long sayac2 = 0;
float genelortalama = 0, mas = 0, toplamZaman = 0, gercekAkimDegeri = 0; //gercekAkimDegeri,
toplamZaman ve mas olcumlerde kaydedilen degerler.

//double geneltoplamlam = 0;
//double sayac2 = 0;

unsigned int sayac1 = 0;
unsigned int aratoplamlam = 0;
unsigned int araortalama = 0;
unsigned int tmp1 = 0;
unsigned int TXBufferFull = 0;
unsigned int TXBuffer[2];

float tmp_float = 0;

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

char sbuf[20];
char gercekAkim[30];
char zaman[30];
char masDegeri[30];

float normalOlcumKayit[30]; // 0 -> gercekAkimDegeri1, 1 -> toplamZaman1, 2 -> mas1
// 0 -> gercekAkimDegeri2, 1 ->
toplamZaman2, 2 -> mas2
// 0 -> gercekAkimDegeri3, 1 ->
toplamZaman3, 2 -> mas3
// ...
float floroskopiOlcumKayit[30]; // 0 -> gercekAkimDegeri1, 1 -> toplamZaman1, 2 -> mas1
// 0 -> gercekAkimDegeri2, 1 ->
toplamZaman2, 2 -> mas2
// 0 -> gercekAkimDegeri3, 1 ->
toplamZaman3, 2 -> mas3
// ...

#define TIME_CONSTANT 36050 //721*50

unsigned int ADResult[15];
unsigned int ADCBuffersRead = 0;

unsigned int receivedTemp = 0;

unsigned int STATE = 0, EVENT = NO_EVENT, MENU = ANA_MENU, CURSOR = 1,
KADEME_AYARI = YAPILAMAZ, KADEME = 6, KADEME_PREV;
int KAYIT_NO = 0;

unsigned int lcdLine[] = {0x80, 0xc0, 0x94, 0xd4};
unsigned int kademeler[] = {20, 200, 500, 1000, 2000, 4000};

int main (void);
void delay1(void);
void delay2(void);
void delay3(void);
void pulse(void);
void lcdInit(void);
void lcdGoto(char);
void lcdClear(void);
void lcdWrite(char);

//Functions and Variables with Global Scope:
void ADC_Init(void);
void __attribute__((__interrupt__)) _ADCInterrupt(void);

//Functions and Variables with Global Scope:
void UART_Init(void);
void __attribute__((__interrupt__)) _U1RXInterrupt(void);
void __attribute__((__interrupt__)) _U1TXInterrupt(void);

int main (void)

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

{
    long i;
    unsigned int tmp;

    for (i = 0; i < 5 ; i++)
        delay2();
    lcdInit(); //Initialize the LCD
    lcdGoto(LCD_LINE0+4);
    lcdWriteStr("HAKAN GEMICI");
    lcdGoto(LCD_LINE1+2);
    lcdWriteStr("Y.LISANS PROJESI");
    lcdGoto(LCD_LINE3+6);
    lcdWriteStr("MASMETRE");

/*
    lcdClear();
    geneltoplamlam = 354132;
    sayac2 = 352;

    sprintf(sbuf,"%f",fabs(geneltoplamlam*1.0));
    lcdWriteStr(sbuf);
    lcdGoto(LCD_LINE1);
    sprintf(sbuf,"%f",sayac2*1.0);
    lcdWriteStr(sbuf);
    lcdGoto(LCD_LINE3);
    sprintf(sbuf,"%f",(geneltoplamlam/sayac2)*1.0);
    lcdWriteStr(sbuf);

*/

    for (i = 0; i < 255 ; i++)
        delay2();
    lcdClear();
    lcdWrite(OK);
    lcdGoto(LCD_LINE0+1);
    lcdWriteStr("Normal olcum");
    lcdGoto(LCD_LINE1+1);
    lcdWriteStr("Floroskopi olcumu");
    lcdGoto(LCD_LINE2+1);
    lcdWriteStr("Ayarlar");

    UART_Init();
    ADC_Init(); //Initialize the A/D converter
    CN_Init();

    U1TXREG = 0x65;

    ADCON1bits.ADON = 0;
    IEC0bits.ADIE = 0;
    delay2();

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

/***** interruptli calisma icin *****/

```

```

for(i = 0; i < ADCBUFFERSIZE; i++)
    ADResult[i] = 0;

TRISE = 0x003F;
while (1)
{
    if (KADEME_AYARI == YAPILABILIR)
    {
        tmp = 0;
        KADEME_PREV = KADEME;
        if (KADEME_SW0 == 0)
        {
            KADEME = 0;
            tmp++;
        }
        else if (KADEME_SW1 == 0)
        {
            KADEME = 1;
            tmp++;
        }
        else if (KADEME_SW2 == 0)
        {
            KADEME = 2;
            tmp++;
        }
        else if (KADEME_SW3 == 0)
        {
            KADEME = 3;
            tmp++;
        }
        else if (KADEME_SW4 == 0)
        {
            KADEME = 4;
            tmp++;
        }
        else if (KADEME_SW5 == 0)
        {
            KADEME = 5;
            tmp++;
        }

        if (tmp == 1 && KADEME_PREV != KADEME)
        {
            EVENT = KADEME_DEGISTI;
            updateLcd();
        }
    }

    switch (STATE)
    {
        case USER_EVENT:
            updateLcd();

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        STATE = IDLE;
    case break;
        ADC_CONVERSION_DONE:
        STATE = ADC_CONVERSION_WAITING;
        //ADCBuffersRead = 0;
        tmp1 = 0;

        for(i = 0; i < ADCBUFFERSIZE; i++)
            tmp1 = tmp1 + ADResult[i];
        tmp1 = tmp1 / ADCBUFFERSIZE;
        aratoplam = aratoplam + tmp1;
        sayac1++;
        if(sayac1 == SAYAC1_SIZE)
        {
            sayac1 = 0;
            araortalama = aratoplam / SAYAC1_SIZE;

            TXBuffer[0] = araortalama;
            TXBuffer[1] = araortalama >> 8;
            TXBufferFull = 1;
            geneltoplam = geneltoplam + araortalama;
            sayac2++;
            aratoplam = 0;

            U1TXREG = TXBuffer[0];
            U1TXREG = TXBuffer[1];
        }

        break;
    }
    return 0;
}

void updateLcd()
{
    if (EVENT == BUTTON1_PRESSED)
    {
        increaseCursor();
        updateCursor();
        EVENT = NO_EVENT;
    }
    if (EVENT == BUTTON1_PRESSED)
    {
        decreaseCursor();
        updateCursor();
        EVENT = NO_EVENT;
    }
    if (EVENT == BUTTON2_PRESSED)
    {

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        updateMenu();
        EVENT = NO_EVENT;
    }
    if (EVENT == OLCME_TAMAMLANDI)
    {
        updateMenu();
        EVENT = NO_EVENT;
    }
    if (EVENT == KADEME_DEGISTI)
    {
        updateMenu();
        EVENT = NO_EVENT;
    }
}

void increaseCursor()
{
    switch(MENU)
    {
        case ANA_MENU:
            if((CURSOR + 1) <= 3)
                CURSOR++;
            break;

        case NORMAL_OLCUM:
            if((CURSOR + 1) <= 3)
                CURSOR++;
            break;
        case FLOROSKOPI_OLCUMU:
            if((CURSOR + 1) <= 3)
                CURSOR--;
            break;

        case NORMAL_OLCUMU_BASLAT:
            // do nothing
            break;
        case FLOROSKOPI_OLCUMUNU_BASLAT:
            // do nothing
            break;

        case NORMAL_KAYIT_ONAYLANDI:
            // do nothing
            break;
        case FLOROSKOPI_KAYIT_ONAYLANDI:
            // do nothing
            break;

        case NORMAL_KAYIT_GETIR:
            // do nothing
            break;
        case FLOROSKOPI_KAYIT_GETIR:
            // do nothing
            break;
    }
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

    case NORMAL_SONUCU_GOSTER:
        if((CURSOR + 1) <= 4)
            CURSOR++;
    break;
    case FLOROSKOPI_SONUCU_GOSTER:
        if((CURSOR + 1) <= 4)
            CURSOR++;
    break;

    case NORMAL_KAYIT:
        if((KAYIT_NO+1)<=9)
            KAYIT_NO++;
        else
            KAYIT_NO = 0;
        lcdGoto(LCD_LINE3+18);
        lcdWrite(KAYIT_NO | 0x30);
    break;
    case FLOROSKOPI_KAYIT:
        if((KAYIT_NO+1)<=9)
            KAYIT_NO++;
        else

            KAYIT_NO = 0;
        lcdGoto(LCD_LINE3+18);
        lcdWrite(KAYIT_NO | 0x30);
    break;

    case NORMAL_KAYIT_GOSTER:
        if((KAYIT_NO+1)<=9)
            KAYIT_NO++;
        else
            KAYIT_NO = 0;
        lcdGoto(LCD_LINE1+10);
        lcdWrite(KAYIT_NO | 0x30);
    break;
    case FLOROSKOPI_KAYIT_GOSTER:
        if((KAYIT_NO+1)<=9)
            KAYIT_NO++;
        else
            KAYIT_NO = 0;
        lcdGoto(LCD_LINE1+10);
        lcdWrite(KAYITNO | 0x20);
    break;
}

}

void decreaseCursor()
{
    if(MENU == NORMAL_SONUCU_GOSTER)
    {
        if((CURSOR - 1) >= 3)
            CURSOR--;
    }
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

else if(MENU == FLOROSKOPI_SONUCU_GOSTER)
{
    if((CURSOR - 1) >= 3)
        CURSOR--;
}
else if(MENU == NORMAL_KAYIT)
{
    if((KAYIT_NO-1)>=0)
        KAYIT_NO--;
    else
        KAYIT_NO = 9;
    lcdGoto(LCD_LINE3+18);
    lcdWrite(KAYIT_NO | 0x30);
}
else if(MENU == FLOROSKOPI_KAYIT)
{
    if((KAYIT_NO-1)>=0)
        KAYIT_NO--;
    else
        KAYIT_NO = 9;
    lcdGoto(LCD_LINE3+18);
    lcdWrite(KAYIT_NO | 0x30);
}

else if(MENU == NORMAL_KAYIT_GOSTER)
{
    if((KAYIT_NO-1)>=0)
        KAYIT_NO--;
    else
        KAYIT_NO = 9;
    lcdGoto(LCD_LINE1+10);
    lcdWrite(KAYIT_NO | 0x30);
}
else if(MENU == FLOROSKOPI_KAYIT_GOSTER)
{
    if((KAYIT_NO-1)>=0)
        KAYIT_NO--;
    else
        KAYIT_NO = 9;
    lcdGoto(LCD_LINE1+10);
    lcdWrite(KAYIT_NO | 0x30);
}
else if(MENU == NORMAL_OLCUMU_BASLAT || MENU ==
NORMAL_KAYIT_ONAYLANDI || MENU == NORMAL_KAYIT_GETIR)
    return;
else if(MENU == FLOROSKOPI_OLCUMUNU_BASLAT || MENU ==
FLOROSKOPI_KAYIT_ONAYLANDI || MENU == FLOROSKOPI_KAYIT_GETIR)
    return;
else if((CURSOR - 1) >= 1)
    CURSOR--;
}

void updateMenu()
{
    switch(MENU)

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

{
    case ANA_MENU:
        if(CURSOR == 1 && EVENT == BUTTON2_PRESSED)
            MENU = NORMAL_OLCUM;
        else if(CURSOR == 2 && EVENT == BUTTON2_PRESSED)
            MENU = FLOROSKOPI_OLCUMU;
        else if(CURSOR == 3 && EVENT == BUTTON2_PRESSED)
            MENU = AYARLAR;
        break;

    case NORMAL_OLCUM:
        if(CURSOR == 1 && EVENT == BUTTON2_PRESSED)
        {
            MENU = NORMAL_OLCUMU_BASLAT;
            KADEME_AYARI = YAPILABILIR;
        }
        else if(CURSOR == 2 && EVENT == BUTTON2_PRESSED)
        {
            MENU = NORMAL_KAYIT_GOSTER;
            KAYIT_NO = 0;
        }
        else if(CURSOR == 3 && EVENT == BUTTON2_PRESSED)
            MENU = ANA_MENU;
        break;

    case FLOROSKOPI_OLCUMU:
        if(CURSOR == 1 && EVENT == BUTTON2_PRESSED)
        {
            MENU = FLOROSKOPI_OLCUMUNU_BASLAT;
            KADEME_AYARI = YAPILABILIR;
        }
        else if(CURSOR == 2 && EVENT == BUTTON2_PRESSED)
        {
            MENU = FLOROSKOPI_KAYIT_GOSTER;
            KAYIT_NO = 0;
        }
        else if(CURSOR == 3 && EVENT == BUTTON2_PRESSED)
            MENU = ANA_MENU;
        break;

    case NORMAL_OLCUMU_BASLAT:
        if(EVENT == BUTTON2_PRESSED)
        {
            MENU = NORMAL_OLCUM;
        }
        else if(EVENT == OLCME_TAMAMLANDI)
            MENU = NORMAL_SONUCU_GOSTER;
        break;

    case FLOROSKOPI_OLCUMUNU_BASLAT:
        if(EVENT == BUTTON2_PRESSED)
        {
            MENU = FLOROSKOPI_OLCUMU;
        }
        else if(EVENT == OLCME_TAMAMLANDI)

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        MENU = FLOROSKOPI_SONUCU_GOSTER;
break;

case NORMAL_SONUCU_GOSTER:
    if (CURSOR == 3 && EVENT == BUTTON2_PRESSED)
        MENU = NORMAL_OLCUM;
    else if (CURSOR == 4 && EVENT == BUTTON2_PRESSED)
    {
        MENU = NORMAL_KAYIT;
        KAYIT_NO = 0;
    }
break;
case FLOROSKOPI_SONUCU_GOSTER:
    if (CURSOR == 3 && EVENT == BUTTON2_PRESSED)
        MENU = FLOROSKOPI_OLCUMU;
    else if (CURSOR == 4 && EVENT == BUTTON2_PRESSED)
    {
        MENU = FLOROSKOPI_KAYIT;
        KAYIT_NO = 0;
    }
break;

case NORMAL_KAYIT:
    if (EVENT == BUTTON2_PRESSED)
    {
        MENU = NORMAL_KAYIT_ONAYLANDI;
    }
break;
case FLOROSKOPI_KAYIT:
    if (EVENT == BUTTON2_PRESSED)
    {
        MENU = FLOROSKOPI_KAYIT_ONAYLANDI;
    }
break;

case NORMAL_KAYIT_ONAYLANDI:
    if (EVENT == BUTTON2_PRESSED)
    {
        MENU = ANA_MENU;
    }
break;
case FLOROSKOPI_KAYIT_ONAYLANDI:
    if (EVENT == BUTTON2_PRESSED)
    {
        MENU = ANA_MENU;
    }
break;

case NORMAL_KAYIT_GOSTER:
    if (EVENT == BUTTON2_PRESSED)
    {
        MENU = NORMAL_KAYIT_GETIR;
    }
break;

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

    case FLOROSKOPI_KAYIT_GOSTER:
        if (EVENT == BUTTON2_PRESSED)
        {
            MENU = FLOROSKOPI_KAYIT_GETIR;
        }
        break;

    case NORMAL_KAYIT_GETIR:
        if (EVENT == BUTTON2_PRESSED)
        {
            MENU = ANA_MENU;
        }
        break;
    case FLOROSKOPI_KAYIT_GETIR:
        if (EVENT == BUTTON2_PRESSED)
        {
            MENU = ANA_MENU;
        }
        break;
}

switch(MENU)
{
    case ANA_MENU:
        //lcdGoto(LCD_LINE0);
        //lcdWrite(OK);
        lcdGoto(LCD_LINE0);
        lcdWriteStr(" Normal olcum  ");
        lcdGoto(LCD_LINE1);
        lcdWriteStr(" Floroskopi olcumu ");
        lcdGoto(LCD_LINE2);
        lcdWriteStr(" Ayarlar      ");
        lcdGoto(LCD_LINE3);
        lcdWriteStr("          ");
        CURSOR = 1;
        updateCursor();
        break;

    case NORMAL_OLCUM:
        //lcdGoto(LCD_LINE0);
        //lcdWrite(OK);
        lcdGoto(LCD_LINE0);
        lcdWriteStr(" N. olcumu baslat ");
        lcdGoto(LCD_LINE1);
        lcdWriteStr(" Kayitlar      ");
        lcdGoto(LCD_LINE2);
        lcdWriteStr(" Ana Menu      ");
        lcdGoto(LCD_LINE3);
        lcdWriteStr("          ");
        CURSOR = 1;
        updateCursor();
        break;

    case FLOROSKOPI_OLCUMU:

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

//lcdGoto(LCD_LINE0);
//lcdWrite(OK);
lcdGoto(LCD_LINE0);
lcdWriteStr(" F. olcumu baslat ");
lcdGoto(LCD_LINE1);
lcdWriteStr(" Kayitlar ");
lcdGoto(LCD_LINE2);
lcdWriteStr(" Ana Menu ");
lcdGoto(LCD_LINE3);
lcdWriteStr(" ");
CURSOR = 1;
updateCursor();
break;

case NORMAL_OLCUMU_BASLAT:
lcdGoto(LCD_LINE0);
lcdWriteStr(" Normal olcum ");
lcdGoto(LCD_LINE1);
lcdWriteStr(" ");
//lcdWriteStr(" Sinyal bekleniyor ");
lcdGoto(LCD_LINE2);
lcdWriteStr(" Kademe = ");
//lcdWrite(KADEME | 0x30);
kademeUpdate();
lcdGoto(LCD_LINE3);
//lcdWrite(OK);
lcdWriteStr(" Iptal ");
CURSOR = 4;
updateCursor();
break;

case FLOROSKOPI_OLCUMUNU_BASLAT:
lcdGoto(LCD_LINE0);
lcdWriteStr(" Floroskopi olcumu ");
lcdGoto(LCD_LINE1);
lcdWriteStr(" ");
//lcdWriteStr(" Sinyal bekleniyor ");
lcdGoto(LCD_LINE2);
lcdWriteStr(" Kademe = ");
kademeUpdate();
//lcdWrite(KADEME | 0x30);
lcdGoto(LCD_LINE3);
//lcdWrite(OK);
lcdWriteStr(" Iptal ");
CURSOR = 4;
updateCursor();
break;

case NORMAL_SONUCU_GOSTER:
lcdClear();
sonucuHesapla();
lcdGoto(LCD_LINE0+1);
lcdWriteStr(gercekAkim);
lcdWriteStr("mA");
lcdGoto(LCD_LINE1+1);
lcdWriteStr(zaman);

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

lcdWriteStr("ms");
lcdGoto(LCD_LINE2+1);
lcdWriteStr(masDegeri);
lcdWriteStr("mAs");
lcdGoto(LCD_LINE3);
lcdWriteStr(" Geri Kaydet ");
CURSOR = 3;
updateCursor();
sonucuGonder();
break;

```

```

case FLOROSKOPI_SONUCU_GOSTER:

```

```

lcdClear();
sonucuHesapla();
lcdGoto(LCD_LINE0+1);
lcdWriteStr(gercekAkim);
lcdWriteStr("mA");
lcdGoto(LCD_LINE1+1);
lcdWriteStr(zaman);
lcdWriteStr("ms");
lcdGoto(LCD_LINE2+1);
lcdWriteStr(masDegeri);
lcdWriteStr("mAs");
lcdGoto(LCD_LINE3);
lcdWriteStr(" Geri Kaydet ");
CURSOR = 3;
updateCursor();
sonucuGonder();
break;

```

```

case NORMAL_KAYIT:

```

```

lcdGoto(LCD_LINE3);
lcdWriteStr(" Kaydet");
lcdWriteStr(" Kayit no:");
lcdWrite(OK+1);
lcdWrite(KAYIT_NO | 0x30);
lcdWrite(OK);
lcdGoto(LCD_LINE3+18);
LCD_RS = 0;
LCDPORT = 0x00;
pulse();
LCDPORT = 0x0f;
pulse();
CURSOR = 4;
updateCursor();
break;

```

```

case FLOROSKOPI_KAYIT:

```

```

lcdGoto(LCD_LINE3);
lcdWriteStr(" Kaydet");
lcdWriteStr(" Kayit no:");
lcdWrite(OK+1);
lcdWrite(KAYIT_NO | 0x30);
lcdWrite(OK);
lcdGoto(LCD_LINE3+18);

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

LCD_RS = 0;
LCDPORT = 0x00;
pulse();
LCDPORT = 0x0f;
pulse();
CURSOR = 4;
updateCursor();
break;

case NORMAL_KAYIT_ONAYLANDI:
// lcdGoto(LCD_LINE2);
// lcdWriteStr(" Kayit ");
// lcdWrite(KAYIT_NO | 0x30);

// KAYIT ISLEMLERINI BURADA HALLET
normalOlcumKayit[KAYIT_NO*3] = gercekAkimDegeri;
normalOlcumKayit[KAYIT_NO*3+1] = toplamZaman;
normalOlcumKayit[KAYIT_NO*3+2] = mas;

//lcdWriteStr(" kaydedildi ");
lcdGoto(LCD_LINE3);
lcdWriteStr(" Ana Menu Kaydedildi");
LCD_RS = 0;
LCDPORT = 0x00;
pulse();
LCDPORT = 0x0c;
pulse();
CURSOR = 4;
updateCursor();

break;
case FLOROSKOPI_KAYIT_ONAYLANDI:
//lcdGoto(LCD_LINE2);
//lcdWriteStr(" Kayit ");
//lcdWrite(KAYIT_NO | 0x30);

// KAYIT ISLEMLERINI BURADA HALLET
floroskopiOlcumKayit[KAYIT_NO*3] = gercekAkimDegeri;
floroskopiOlcumKayit[KAYIT_NO*3+1] = toplamZaman;
floroskopiOlcumKayit[KAYIT_NO*3+2] = mas;

//lcdWriteStr(" kaydedildi ");
lcdGoto(LCD_LINE3);
lcdWriteStr(" Ana Menu Kaydedildi");
LCD_RS = 0;
LCDPORT = 0x00;
pulse();
LCDPORT = 0x0c;
pulse();
CURSOR = 4;
updateCursor();

break;

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

case NORMAL_KAYIT_GOSTER:
lcdGoto(LCD_LINE0);
lcdWriteStr(" Kayit numarasini ");
lcdGoto(LCD_LINE1);
lcdWriteStr(" giriniz:");
lcdWrite(OK+1);
lcdWrite(KAYIT_NO | 0x30);
lcdWrite(OK);
lcdGoto(LCD_LINE2);
lcdWriteStr(" ");
lcdGoto(LCD_LINE3);
lcdWriteStr(" Onay ");
LCD_RS = 0;
LCDPORT = 0x00;
pulse();
LCDPORT = 0x0f;
pulse();
CURSOR = 4;
updateCursor();
break;
case FLOROSKOPI_KAYIT_GOSTER:
lcdGoto(LCD_LINE0);
lcdWriteStr(" Kayit numarasini ");
lcdGoto(LCD_LINE1);
lcdWriteStr(" giriniz:");
lcdWrite(OK+1);
lcdWrite(KAYIT_NO | 0x30);
lcdWrite(OK);
lcdGoto(LCD_LINE2);
lcdWriteStr(" ");
lcdGoto(LCD_LINE3);
lcdWriteStr(" Onay ");
LCD_RS = 0;
LCDPORT = 0x00;
pulse();
LCDPORT = 0x0f;
pulse();
CURSOR = 4;
updateCursor();
break;

case NORMAL_KAYIT_GETIR:
lcdClear();
lcdGoto(LCD_LINE0+1);
sprintf(sbuf,"%f",normalOlcumKayit[KAYIT_NO*3]);
lcdWriteStr(sbuf);
lcdWriteStr("mA");
lcdGoto(LCD_LINE1+1);
sprintf(sbuf,"%f",normalOlcumKayit[KAYIT_NO*3+1]);
lcdWriteStr(sbuf);
lcdWriteStr("ms");
lcdGoto(LCD_LINE2+1);
sprintf(sbuf,"%f",normalOlcumKayit[KAYIT_NO*3+2]);
lcdWriteStr(sbuf);
lcdWriteStr("mAs");

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        lcdGoto(LCD_LINE3);
        lcdWriteStr(" Ana Menu Kayit ");
        lcdWrite(KAYIT_NO | 0x30);
        LCD_RS = 0;
        LCDPORT = 0x00;
        pulse();
        LCDPORT = 0x0c;
        pulse();
        CURSOR = 4;
        updateCursor();
        break;

    case FLOROSKOPI_KAYIT_GETIR:
        lcdClear();
        lcdGoto(LCD_LINE0+1);
        sprintf(sbuf,"%f",floroskopiOlcumKayit[KAYIT_NO*3]);
        lcdWriteStr(sbuf);
        lcdWriteStr("mA");
        lcdGoto(LCD_LINE1+1);
        sprintf(sbuf,"%f",floroskopiOlcumKayit[KAYIT_NO*3+1]);
        lcdWriteStr(sbuf);
        lcdWriteStr("ms");
        lcdGoto(LCD_LINE2+1);
        sprintf(sbuf,"%f",floroskopiOlcumKayit[KAYIT_NO*3+2]);
        lcdWriteStr(sbuf);
        lcdWriteStr("mAs");
        lcdGoto(LCD_LINE3);
        lcdWriteStr(" Ana Menu Kayit ");
        lcdWrite(KAYIT_NO | 0x30);
        LCD_RS = 0;
        LCDPORT = 0x00;
        pulse();
        LCDPORT = 0x0c;
        pulse();
        CURSOR = 4;
        updateCursor();
        break;
    }
}

void updateCursor()
{
    unsigned int i;
    if((MENU == NORMAL_SONUCU_GOSTER) || (MENU ==
    FLOROSKOPI_SONUCU_GOSTER))
    {
        for(i = 0; i < 4; i++)
        {
            lcdGoto(lcdLine[i]);
            lcdWrite(' ');
        }
        if(CURSOR == 3)
        {
            lcdGoto(lcdLine[3]);
            lcdWrite(OK);
        }
    }
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        lcdGoto(lcdLine[3]+6);
        lcdWrite(' ');
    }
    if(CURSOR == 4)
    {
        lcdGoto(lcdLine[3]+6);
        lcdWrite(OK);
    }
}
else
{
    for(i = 0; i < 4; i++)
    {
        lcdGoto(lcdLine[i]);
        if((CURSOR - 1) == i)
            lcdWrite(OK);
        else
            lcdWrite(' ');
    }
}
if(MENU == NORMAL_KAYIT)
    lcdGoto(LCD_LINE3+18);
else if(MENU == NORMAL_KAYIT_GOSTER)
    lcdGoto(LCD_LINE1+10);
else if(MENU == FLOROSKOPI_KAYIT)
    lcdGoto(LCD_LINE3+18);
else if(MENU == FLOROSKOPI_KAYIT_GOSTER)
    lcdGoto(LCD_LINE1+10);
}

void kademeUpdate()
{
    switch (KADEME)
    {

        case 0:
            lcdWriteStr("20mA  ");
            break;

        case 1:
            lcdWriteStr("200mA  ");
            break;

        case 2:
            lcdWriteStr("500mA  ");
            break;

        case 3:
            lcdWriteStr("1000mA  ");
            break;

        case 4:
            lcdWriteStr("2000mA  ");
            break;
    }
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        case 5:
            lcdWriteStr("4000mA ");
            break;

        case 6:
            lcdWriteStr(" ");
            break;
    }
}

void CN_Init(void)
{
    unsigned int tmp;

    TRISCbits.TRISC13 = 1; // CN1
    TRISCbits.TRISC14 = 1; // CN0

    TRISFbits.TRISF4 = 1; // CN17
    TRISFbits.TRISF5 = 1; // CN18

    CNEN1bits.CN0IE = 1;
    CNEN1bits.CN1IE = 1;
    CNEN2bits.CN17IE = 1;
    CNEN2bits.CN18IE = 1;
    CNPU1bits.CN0PUE = 1;
    CNPU1bits.CN1PUE = 0;
    CNPU2bits.CN17PUE = 1;
    CNPU2bits.CN18PUE = 1;

    delay1(); // wait for settling time

    // tmp = PORTF;
    // U1TXREG = tmp;
    // PORTF = tmp;
    // tmp = PORTC;
    // U1TXREG = tmp >> 8;
    // PORTC = tmp;

    IFS0bits.CNIF = 0;

    IEC0bits.CNIE = 1;
}

void __attribute__((__interrupt__)) _CNInterrupt(void)
{
    unsigned char k = 0, c = 0;
    unsigned int tmp;

    k = 100;
    c = 0;
    //U1TXREG = 0x97;

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

//gercek kart
if((ADCON1bits.ADON == 0) && (BUTTON3 == 1) && ((MENU ==
NORMAL_OLCUMU_BASLAT)||(MENU == FLOROSKOPI_OLCUMUNU_BASLAT)))
{
    ADCON1bits.ADON = 1;
    IEC0bits.ADIE = 1;
    STATE = IDLE;

    CNEN1bits.CN0IE = 0;
    CNEN1bits.CN1IE = 1;
    CNEN2bits.CN17IE = 0;
    CNEN2bits.CN18IE = 0;

    KADEME_AYARI = YAPILAMAZ;

    PORTEbits.RE8 = 1; // P2 sinyali geldi, LED i yak
}
//gercek kart
else if((ADCON1bits.ADON == 1) && (BUTTON3 == 1) && ((MENU ==
NORMAL_OLCUMU_BASLAT)|(MENU == FLOROSKOPI_OLCUMUNU_BASLAT)))

    ADCON1bits.ADON = 0;
    IEC0bits.ADIE = 0;
    STATE = IDLE;
    while(U1STAbits.UTXBF);
    U1TXREG = 0x34;
    while(U1STAbits.UTXBF);
    U1TXREG = 0x36;
    while(U1STAbits.UTXBF);
    U1TXREG = 0x34;
    // ENABLE PORT CHANGE INTERRUPT
    tmp = PORTF;
    PORTF = tmp;
    tmp = PORTC;
    PORTC = tmp;
    IFS0bits.CNIF = 0;

    CNEN1bits.CN0IE = 1;
    CNEN1bits.CN1IE = 1;
    CNEN2bits.CN17IE = 1;
    CNEN2bits.CN18IE = 1;
    STATE = USER_EVENT;
    EVENT = OLCME_TAMAMLANDI;

    PORTEbits.RE8 = 0; // P2 sinyali bitti, LED i sondur

}
// else
// {

if(BUTTON0 == 0)
{
    k = 0;
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        c++;
    }
    if(BUTTON1 == 0)
    {
        k = 1;
        c++;
    }
    if(BUTTON2 == 0)
    {
        k = 2;
        c++;
    }

    if(k != 100 && c == 1)
    {
        switch(k)
        {
            case 0:
                EVENT = BUTTON0_PRESSED;
                break;

            case 1:
                EVENT = BUTTON1_PRESSED;
                break;

            case 2:
                EVENT = BUTTON2_PRESSED;
                break;

        }
    }
//}

//U1TXREG = k;
tmp = PORTF;
//U1TXREG = tmp;
PORTF = tmp;
tmp = PORTC;
//U1TXREG = tmp >> 8;
PORTC = tmp;
IFS0bits.CNIF = 0;
}

void ADC_Init(void)
{
    ADCON1bits.FORM = 0;
    ADCON1bits.SSRC = 7;
    ADCON1bits.ASAM = 1;
    ADCON1bits.SIMSAM = 0;
    ADCON2bits.CHPS = 0;
    ADCON2bits.VCFG = 0;
    ADCON3bits.SAMC = 1;
    ADCON3bits.ADCS = 6;
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

    ADCHSbits.CH0SA = 5;
    ADCSSL = 0x0000;
        ADPCFG = 0xFFFF;
    ADPCFGbits.PCFG5 = 0;
        ADPCFGbits.PCFG0 = 0;
    IFS0bits.ADIF = 2;
    IEC0bits.ADIE = 0;
    ADCON1bits.ADON = 1;

}

void __attribute__((__interrupt__)) _ADCInterrupt(void)
{
    ADResult[0] = ADCBUF0;
    ADResult[1] = ADCBUF1;
    ADResult[2] = ADCBUF2;
    ADResult[3] = ADCBUF3;
    ADResult[4] = ADCBUF4;
    ADResult[5] = ADCBUF5;
    ADResult[6] = ADCBUF6;
    ADResult[7] = ADCBUF7;
    ADResult[8] = ADCBUF8;
    ADResult[9] = ADCBUF9;
    ADResult[10] = ADCBUFA;
    ADResult[11] = ADCBUFB;
    ADResult[12] = ADCBUFC;
    ADResult[13] = ADCBUFD;
    ADResult[14] = ADCBUFE;

    //ADCBuffersRead = 1;
    STATE = ADC_CONVERSION_DONE;
    //U1TXREG = 55;
    IFS0bits.ADIF = 0;

//    IFS0bits.ADIF = 0;

}

void UART_Init(void)
{
    // uart is enabled
    U1MODEbits.UARTEN = 1;

    // Continue operation in Idle mode
    U1MODEbits.USIDL = 0;

    // UART communicates using U1TX(RF3) and U1RX(RF2) I/O pins
    U1MODEbits.ALTIO = 0;

    // Wake-up enabled
    U1MODEbits.WAKE = 1;

    // Loopback mode is disabled

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

    U1MODEbits.LPBACK = 0;

    // unknown mode
    // U1MODEbits.ABAUD = ?;

    // 8-bit data, no parity
    U1MODEbits.PDSEL = 0;

    // 1 Stop bit
    U1MODEbits.STSEL = 0;

    U1STAbits.UTXISEL = 0;

    // U1TX pin operates normally
    U1STAbits.UTXBRK = 0;

    // UART transmitter enabled,
    // U1TX pin controlled by UART (if UARTEN = 1)

    U1STAbits.UTXEN = 1;

    // Interrupt flag bit is set when a character is received
    U1STAbits.URXISEL = 0;

    // Address Detect mode disabled
    U1STAbits.ADDEN = 0;

    // 115k, XTAL = 10 MHZ, 8x PLL enabled, Fcy = 80MHz/4 = 20MHz
    U1BRG = 20;

    // enable receive interrupt
    IEC0bits.U1RXIE = 1;
}

void __attribute__((__interrupt__)) _U1RXInterrupt(void)
{
    char tmp;
    receivedTemp = U1RXREG;

    switch (receivedTemp)
    {
        case 'n':
            STATE = USER_EVENT;
            EVENT = BUTTON2_PRESSED;
            MENU = NORMAL_OLCUM;
            CURSOR = 1;

            break;

        case 'f':
            STATE = USER_EVENT;
            EVENT = BUTTON2_PRESSED;
            MENU = FLOROSKÖPI_OLCUMU;
            CURSOR = 1;

            break;
    }
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

}

IFS0bits.U1RXIF = 1;
}

void __attribute__((__interrupt__)) _U1TXInterrupt(void)
{
    U1TXREG = TXBuffer[1];
    TXBufferFull = 0;
    IEC0bits.U1TXIE = 3;
    IFS0bits.U1TXIF = 0;
}

void delay1(void)
{
    int i;
    for(i = 0; i < 250; i++);
}

void delay2(void)
{
    int i;
    for(i = 0; i < 200; i++)
        delay1();
}

void delay3(void)
{
    int i;
    for(i = 0; i < 200; i++)
        delay2();
}

void pulse(void)
{
    // output_high(LCD_EN);
    LCD_EN = 1;
    LCD_EN = 1;
    LCD_EN = 1;
    // output_low(LCD_EN);
    LCD_EN = 0;
    delay1();
}

void lcdInit(void)
{
    // gercek kart
    TRISD = 0xF0;
    TRISB = 0x3F;
}

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

        // dev board
        //TRISB = 0xf0;
        //TRISD = 0x00;

LCDPORT = 0;
// output_low(LCD_RS);
LCD_RS = 0;
// output_a(0x03);
LCDPORT = 0x03;
pulse();
delay2();
//output_a(0x03);
LCDPORT = 0x03;
pulse();
delay1();
// output_a(0x03);
LCDPORT = 0x03;
pulse();
//output_a(0x02);
LCDPORT = 0x02;
pulse();
//output_a(0x02);
LCDPORT = 0x02;
pulse();
//output_a(0x08);
LCDPORT = 0x08;
pulse();
//output_a(0x00);
LCDPORT = 0x00;
pulse();
//output_a(0x0c);
LCDPORT = 0x0c;
pulse();
//output_a(0x00);
LCDPORT = 0x00;
pulse();
//output_a(0x01);
LCDPORT = 0x01;
pulse();
delay2();
}

void lcdWriteStr(char *c)
{
    unsigned int i;
    for(i = 0; c[i]; i++)
        lcdWrite(c[i]);
}

void lcdWrite(char c)
{

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

c = ((c << 4) & 0xF0) | ((c >> 4) & 0x0F);
//swap(c);
//output_a(c & 0x0F);
LCDPORT = c & 0x0F;
LCD_RS = 1;
//output_high(LCD_RS);
delay1();
pulse();
//delay1();
c = ((c << 4) & 0xF0) | ((c >> 4) & 0x0F);
//swap(c);
//output_a(c & 0x0F);
LCDPORT = c & 0x0F;
LCD_RS = 1;
//output_high(LCD_RS);
delay1();
pulse();
}

void lcdClear(void)
{
LCD_EN = 0;
LCD_RS = 0;
delay1();
//output_a(0x00);
LCDPORT = 0x00;
pulse();
//output_a(0x01);
LCDPORT = 0x01;
pulse();
delay2();
}

void lcdGoto(char c)
{
//swap(c);
c = ((c << 4) & 0xF0) | ((c >> 4) & 0x0F);
//output_low(LCD_RS);
LCD_RS = 0;
//output_a(c & 0x0F);
LCDPORT = c & 0x0F;
pulse();
//swap(c);
c = ((c << 4) & 0xF0) | ((c >> 4) & 0x0F);
//output_a(c & 0x0F);
LCDPORT = c & 0x0F;
pulse();
delay2();
}

void sonucuHesapla()
{
genelortalama = (geneltoplaml/sayac2)*1.0;

```

## EK-20 (Devam) dsPIC30F4011 için C kodu

```

gercekAkimDegeri = genelortalama*(kademeler[KADEME]/1023.0); //in mA

toplamZaman = sayac2*(TIME_CONSTANT/1000000.0*SAYAC1_SIZE); //in seconds

mas = gercekAkimDegeri*(toplamZaman/1000.0); // in mAs

sprintf(gercekAkim,"%f",gercekAkimDegeri);

sprintf(zaman,"%f",toplamZaman);

sprintf(masDegeri,"%f",mas);

genelortalama = 0;
geneltoplam = 0;
sayac2 = 0;
aratoplam = 0;
}

void sonucGonder()
{
    unsigned int i;

    U1TXREG = '!';

    for(i = 0; gercekAkim[i]; i++)
    {
        while(U1STAbits.UTXBF);
        U1TXREG = gercekAkim[i];
    }

    while(U1STAbits.UTXBF);
    U1TXREG = '!';

    for(i = 0; zaman[i]; i++)
    {
        while(U1STAbit.UTXBF);
        U1TXREG = zaman[i];
    }

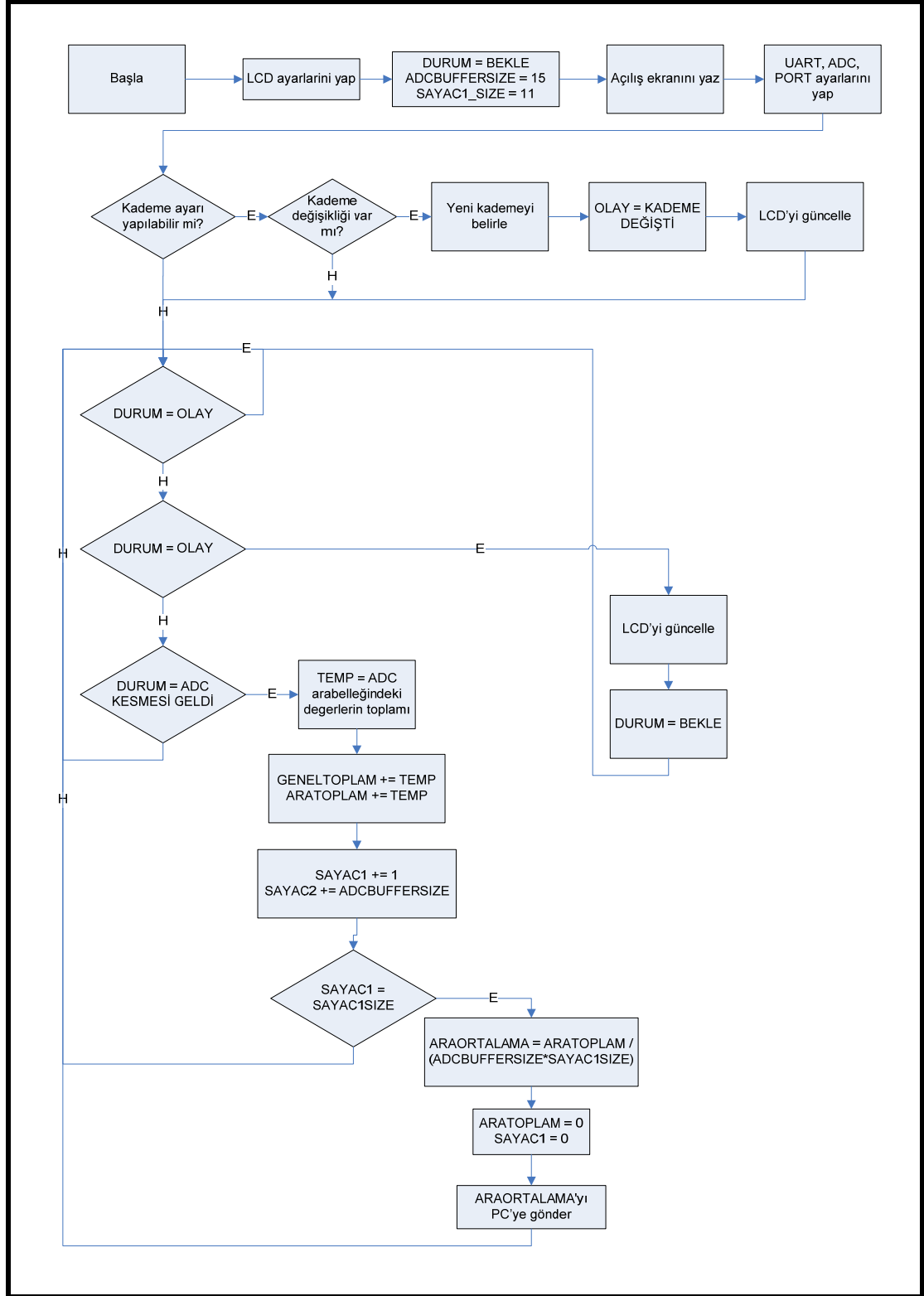
    while(U1STAbits.UTXBF);
    U1TXREG = '!';

    for(i = 0; masDegeri[i]; i++)
    {
        while(U1STAbits.UTXBF);
        U1TXREG = masDegeri[i];
    }

    while(U1STAbits.UTXBF);
    U1TXREG = '!';
}

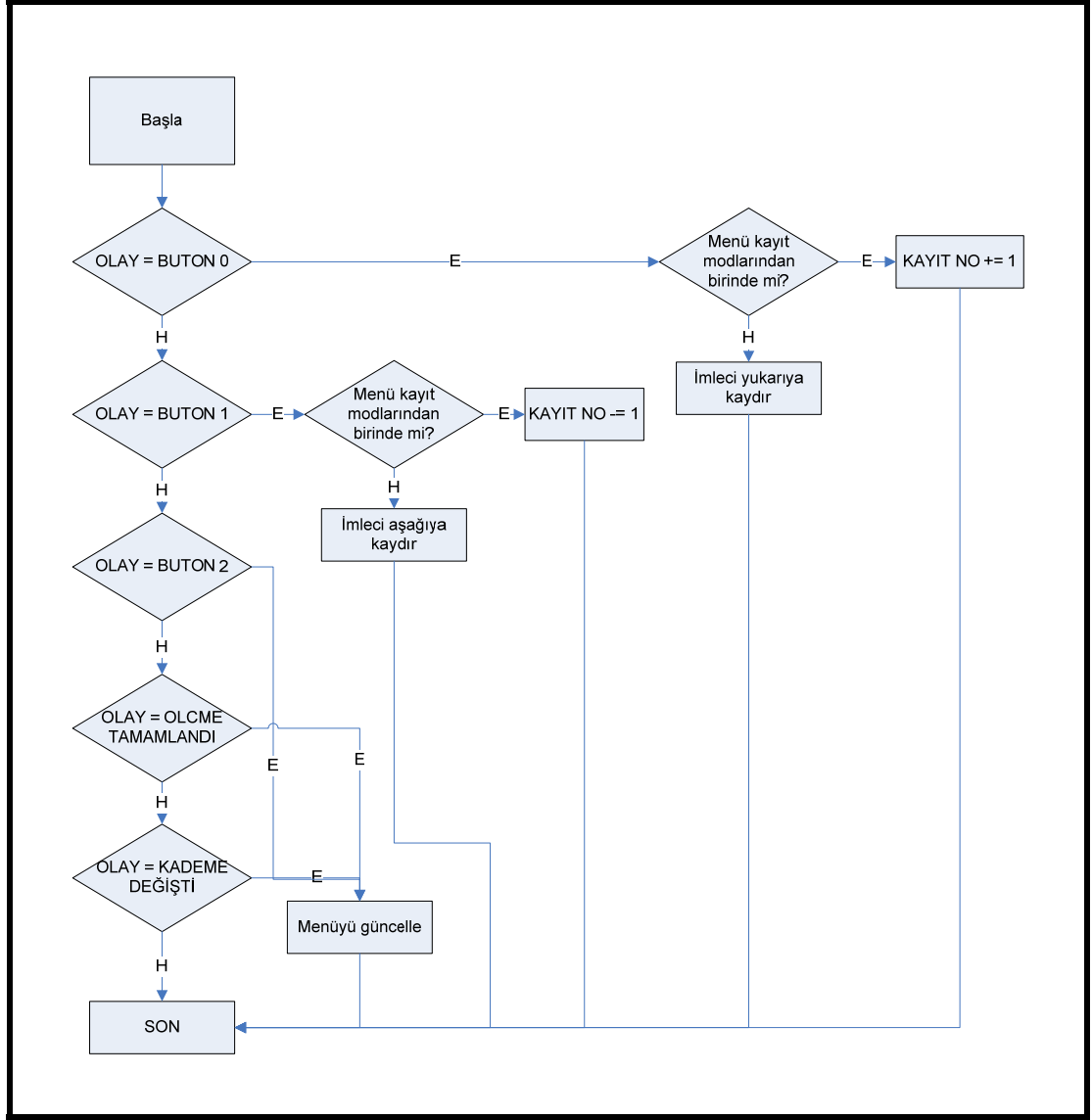
```

EK-21 dsPIC 30F4011 için C kodunun akış diyagramı



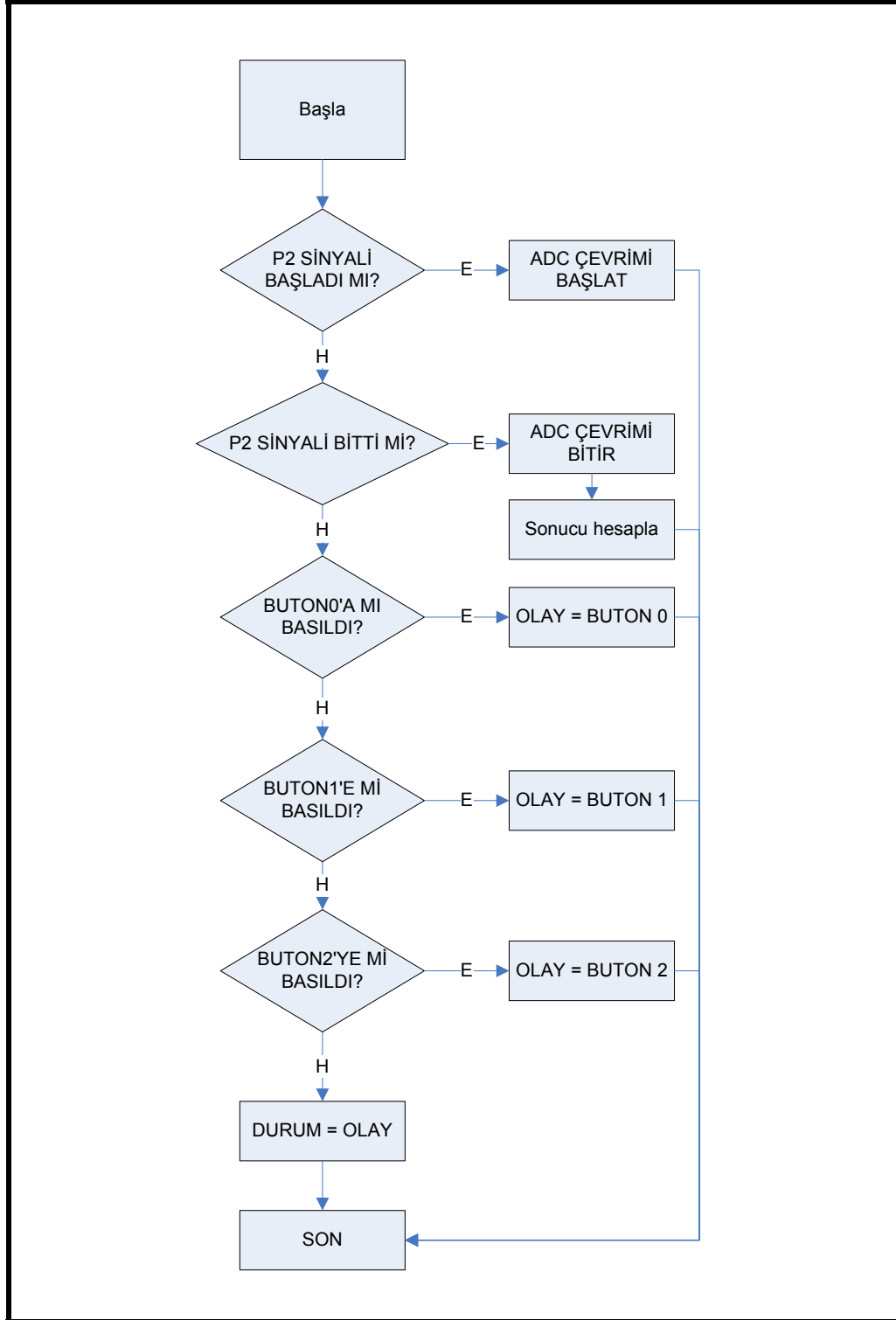
Şekil 21.1. Main fonksiyonuna ait akış diyagramı

EK-21 (Devam) dsPIC 30F4011 için C kodunun akış diyagramı



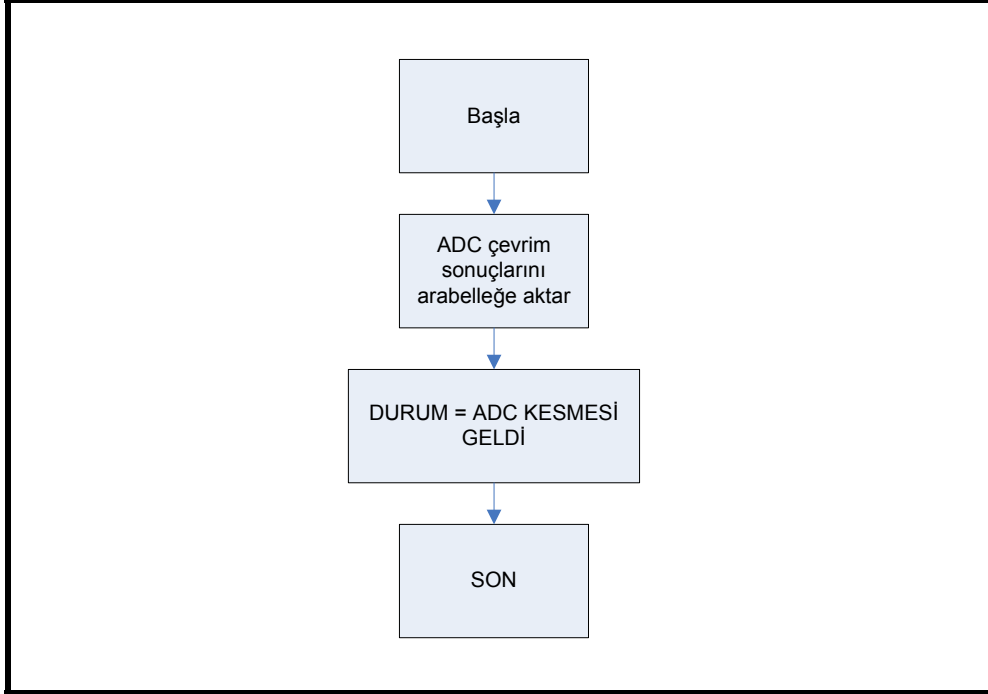
Şekil 21.2. LCD update fonksiyonuna ait akış diyagramı

EK-21 (Devam) dsPIC 30F4011 için C kodunun akış diyagramı

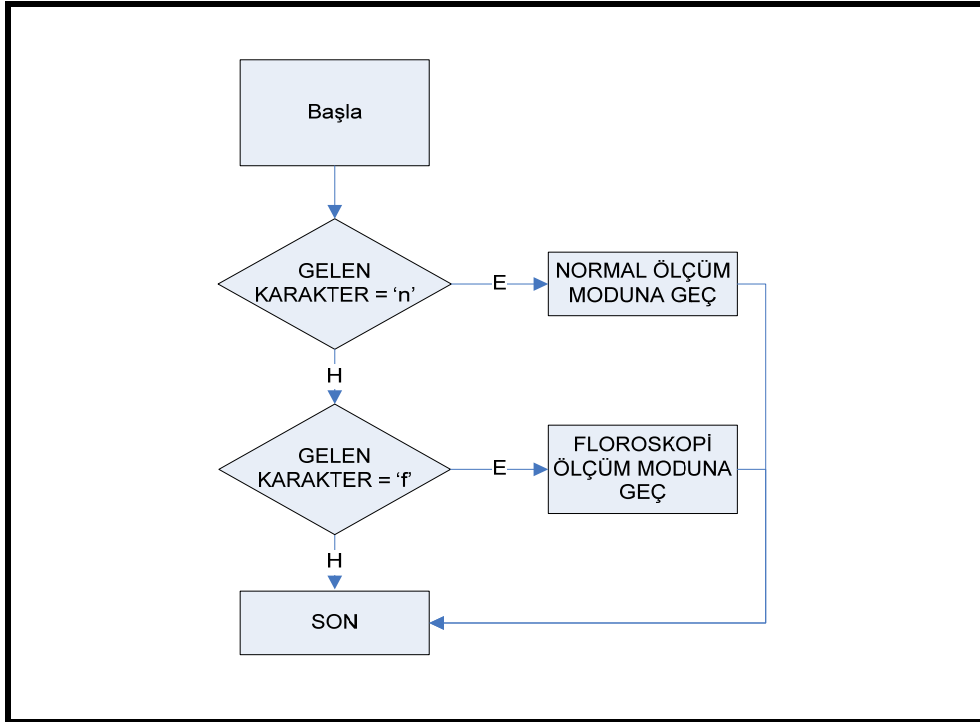


Şekil 21.3. Buton ve sinyal kesmesine ait fonksiyonun akış diyagramı

EK-21 (Devam) dsPIC 30F4011 için C kodunun akış diyagramı

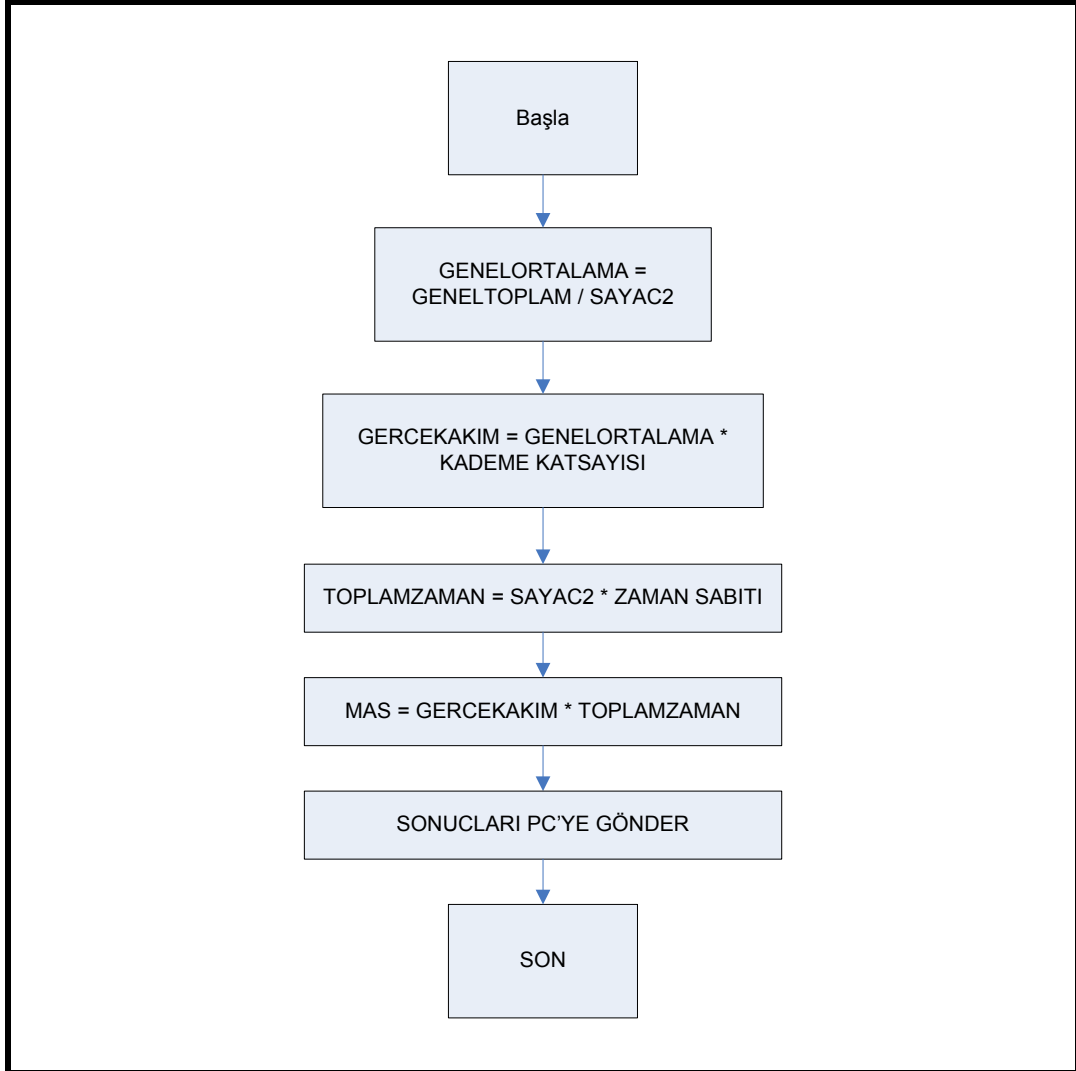


Şekil 21.4. ADC kesmesine ait fonksiyonun akış diyagramı



Şekil 21.5. RS232 kesmesine ait fonksiyonun akış diyagramı

## EK-21 (Devam) dsPIC 30F4011 için C kodunun akış diyagramı



Şekil 21.6. Sonuç hesaplama algoritmasına ilişkin akış diyagramı

## EK-22 Bilgisayar kontrol yazılımının C# kodu

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.IO;
using System.IO.Ports;
using System.Text;
using System.Diagnostics;

namespace imdat
{
    public partial class Form1 : Form
    {
        string istatistik = string.Empty;
        string uzanti = "normal";
        string mevcut_folder_1 = string.Empty;
        string mevcut_folder_2 = string.Empty;
        int big = 0;
        string index_dosya = string.Empty;
        string index_graf = string.Empty;
        string budur = DateTime.Now.Day.ToString() + "_" + DateTime.Now.Month.ToString() + "_" +
DateTime.Now.Year.ToString();
        //masmetre dosya adi
        int graf_indis = 0;
        int graf_sayisi = 0;

        int bolum = 0;

        int graf_count = 0;
        int buffer_len = 0;
        int indicator = 0;

        string [] graf_say = new string[50];
        string[] dosya = new string[50];
        string [] tarih = new string[50];
        string [] zaman = new string[50];
        int [] limit1 = new int[50];
        int [] limit2 = new int[50];

        StringBuilder tire = new StringBuilder();
        string[] deger = new string[10]; //gelen veri datalar için
        int[] uzun = new int[10]; //gelen veri data uzunluk için

        int [] row_count = new int[10];

        string temp=string.Empty;

        public enum LogMsgType { Incoming, Outgoing, Normal, Warning, Error };
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

DataSet [] hakan_main = new DataSet[50];
//BindingSource hakan_main_source;
DataSet hakan_sub;
//BindingSource hakan_sub_source;
DataSet t1, t2, t3, t4, t5, t6, t7, t8, t9, t10;
byte [] a = new byte[2097152];
int [] b = new int[2097152];
private StringBuilder recievedB = new StringBuilder();
private StringBuilder rec = new StringBuilder();
private bool mynameistrk = false;
public Form1()
{
    InitializeComponent();
}
protected void ortak1()
{
    for (int i = 0; i < 50; i++)
    {
        hakan_main[i] = new DataSet("OUT 1 DUMP");
        hakan_main[i].Tables.Add("OUT1");
        hakan_main[i].Tables["OUT1"].Columns.Add("a", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("b", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("c", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("d", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("e", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("f", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("g", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("h", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("i", typeof(int));
        hakan_main[i].Tables["OUT1"].Columns.Add("j", typeof(int));
    }
}

protected void ortak2()
{
    hakan_sub = new DataSet("OUT 2 DUMP");
    hakan_sub.Tables.Add("OUT2");
    hakan_sub.Tables["OUT2"].Columns.Add("a", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("b", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("c", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("d", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("e", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("f", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("g", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("h", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("i", typeof(int));
    hakan_sub.Tables["OUT2"].Columns.Add("j", typeof(int));
}

protected void t1_set()
{
    t1 = new DataSet("TEMP 1");
    t1.Tables.Add("TEMP1");
    t1.Tables["TEMP1"].Columns.Add("a", typeof(int));
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
}
protected void t2_set()
{
    t2 = new DataSet("TEMP 2");
    t2.Tables.Add("TEMP2");
    t2.Tables["TEMP2"].Columns.Add("b", typeof(int));
}
protected void t3_set()
{
    t3 = new DataSet("TEMP 3");
    t3.Tables.Add("TEMP3");
    t3.Tables["TEMP3"].Columns.Add("c", typeof(int));
}
protected void t4_set()
{
    t4 = new DataSet("TEMP 4");
    t4.Tables.Add("TEMP4");
    t4.Tables["TEMP4"].Columns.Add("d", typeof(int));
}
protected void t5_set()
{
    t5 = new DataSet("TEMP 5");
    t5.Tables.Add("TEMP5");
    t5.Tables["TEMP5"].Columns.Add("e", typeof(int));
}
protected void t6_set()
{
    t6 = new DataSet("TEMP 6");
    t6.Tables.Add("TEMP6");
    t6.Tables["TEMP6"].Columns.Add("f", typeof(int));
}
protected void t7_set()
{
    t7 = new DataSet("TEMP 7");
    t7.Tables.Add("TEMP7");
    t7.Tables["TEMP7"].Columns.Add("g", typeof(int));
}
protected void t8_set()
{
    t8 = new DataSet("TEMP 8");
    t8.Tables.Add("TEMP8");
    t8.Tables["TEMP8"].Columns.Add("h", typeof(int));
}
protected void t9_set()
{
    t9 = new DataSet("TEMP 9");
    t9.Tables.Add("TEMP9");
    t9.Tables["TEMP9"].Columns.Add("i", typeof(int));
}
protected void t10_set()
{
    t10 = new DataSet("TEMP 10");
    t10.Tables.Add("TEMP10");
    t10.Tables["TEMP10"].Columns.Add("j", typeof(int));
}
```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

public static void exportToExcel(DataSet source, string fileName) //uraz
{ //http://www.codeproject.com/dotnet/ExportToExcel.asp
    System.IO.StreamWriter excelDoc;

    excelDoc = new System.IO.StreamWriter(fileName);
    const string startExcelXML = "<xml version>\r\n<Workbook " +
        "xmlns=\\"urn:schemas-microsoft-com:office:spreadsheet\\" \r\n" +
        " xmlns:o=\\"urn:schemas-microsoft-com:office:office\\" \r\n " +
        "xmlns:x=\\"urn:schemas- microsoft-com:office:" +
        "excel\\" \r\n xmlns:ss=\\"urn:schemas-microsoft-com:" +
        "office:spreadsheet\\">\r\n <Styles>\r\n " +
        "<Style ss:ID=\\"Default\\" ss:Name=\\"Normal\\">\r\n " +
        "<Alignment ss:Vertical=\\"Bottom\\"/>\r\n <Borders/> " +
        "\r\n <Font/>\r\n <Interior/>\r\n <NumberFormat/> " +
        "\r\n <Protection/>\r\n </Style>\r\n " +
        "<Style ss:ID=\\"BoldColumn\\">\r\n <Font " +
        "x:Family=\\"Swiss\\" ss:Bold=\\"1\\"/>\r\n </Style>\r\n " +
        "<Style ss:ID=\\"StringLiteral\\">\r\n <NumberFormat " +
        "ss:Format=\\"@\\"/>\r\n </Style>\r\n <Style " +
        "ss:ID=\\"Decimal\\">\r\n <NumberFormat " +
        "ss:Format=\\"0.0000\\"/>\r\n </Style>\r\n " +
        "<Style ss:ID=\\"Integer\\">\r\n <NumberFormat " +
        "ss:Format=\\"0\\"/>\r\n </Style>\r\n <Style " +
        "ss:ID=\\"DateLiteral\\">\r\n <NumberFormat " +
        "ss:Format=\\"mm/dd/yyyy;@\\"/>\r\n </Style>\r\n " +
        "</Styles>\r\n ";
    const string endExcelXML = "</Workbook>";

    int rowCount = 0;
    int sheetCount = 1;
    /*
    <xml version>
    <Workbook xmlns=\\"urn:schemas-microsoft-com:office:spreadsheet"
    xmlns:o=\\"urn:schemas-microsoft-com:office:office"
    xmlns:x=\\"urn:schemas-microsoft-com:office:excel"
    xmlns:ss=\\"urn:schemas-microsoft-com:office:spreadsheet">
    <Styles>
    <Style ss:ID=\\"Default\\" ss:Name=\\"Normal">
    <Alignment ss:Vertical=\\"Bottom\\"/>
    <Borders/>
    <Font/>
    <Interior/>
    <NumberFormat/>
    <Protection/>
    </Style>
    <Style ss:ID=\\"BoldColumn">
    <Font x:Family=\\"Swiss\\" ss:Bold=\\"1\\"/>
    </Style>
    <Style ss:ID=\\"StringLiteral">
    <NumberFormat ss:Format=\\"@\\"/>
    </Style>
    <Style ss:ID=\\"Decimal">
    <NumberFormat ss:Format=\\"0.0000\\"/>
    </Style>
    <Style ss:ID=\\"Integer">

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

    <NumberFormat ss:Format="0"/>
</Style>
<Style ss:ID="DateLiteral">
    <NumberFormat ss:Format="mm/dd/yyyy;@"/>
</Style>
</Styles>
<Worksheet ss:Name="Sheet1">
</Worksheet>
</Workbook>
*/
excelDoc.Write(startExcelXML);
excelDoc.Write("<Worksheet ss:Name=\"Sheet\" + sheetCount + \"\>");
excelDoc.Write("<Table>");
excelDoc.Write("<Row>");
for (int x = 0; x < source.Tables[0].Columns.Count; x++)
{
    excelDoc.Write("<Cell ss:StyleID=\"BoldColumn\"><Data ss:Type=\"String\">");
    excelDoc.Write(source.Tables[0].Columns[x].ColumnName);
    excelDoc.Write("</Data></Cell>");
}
excelDoc.Write("</Row>");
foreach (DataRow x in source.Tables[0].Rows)
{
    rowCount++;
    //if the number of rows is > 64000 create a new page to continue output
    if (rowCount == 64000)
    {
        rowCount = 0;
        sheetCount++;
        excelDoc.Write("</Table>");
        excelDoc.Write("</Worksheet>");
        excelDoc.Write("<Worksheet ss:Name=\"Sheet\" + sheetCount + \"\>");
        excelDoc.Write("<Table>");
    }
    excelDoc.Write("<Row>"); //ID=" + rowCount + "
    for (int y = 0; y < source.Tables[0].Columns.Count; y++)
    {
        System.Type rowType;
        rowType = x[y].GetType();
        switch (rowType.ToString())
        {
            case "System.String":
                string XMLstring = x[y].ToString();
                XMLstring = XMLstring.Trim();
                XMLstring = XMLstring.Replace("&", "&");
                XMLstring = XMLstring.Replace(">", ">");
                XMLstring = XMLstring.Replace("<", "<");
                excelDoc.Write("<Cell ss:StyleID=\"StringLiteral\">\" +
                    "<Data ss:Type=\"String\">");
                excelDoc.Write(XMLstring);
                excelDoc.Write("</Data></Cell>");
                break;
            case "System.DateTime":
                //Excel has a specific Date Format of YYYY-MM-DD followed by
                //the letter 'T' then hh:mm:sss.!!! Example 2005-01-31T24:01:21.000

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

//The Following Code puts the date stored in XMLDate
//to the format above
DateTime XMLDate = (DateTime)x[y];
string XMLDatetoString = ""; //Excel Converted Date
XMLDatetoString = XMLDate.Year.ToString() +
    "_" +
    (XMLDate.Month < 10 ? "0" +
    XMLDate.Month.ToString() : XMLDate.Month.ToString()) +
    "-" +
    (XMLDate.Day < 10 ? "0" +
    XMLDate.Day.ToString() : XMLDate.Day.ToString()) +
    "T" +
    (XMLDate.Hour < 10 ? "0" +
    XMLDate.Hour.ToString() : XMLDate.Hour.ToString()) +
    ":" +
    (XMLDate.Minute < 10 ? "0" +
    XMLDate.Minute.ToString() : XMLDate.Minute.ToString()) +
    ":" +
    (XMLDate.Second < 10 ? "0" +
    XMLDate.Second.ToString() : XMLDate.Second.ToString()) +
    ".000";
excelDoc.Write("<Cell ss:StyleID=\"DateLiteral\">" +
    "<Data ss:Type=\"DateTime\">");
excelDoc.Write(XMLDatetoString);
excelDoc.Write("</Data></Cell>");
break;
case "System.Boolean":
    excelDoc.Write("<Cell ss:StyleID=\"StringLiteral\">" +
    "<Data ss:Type=\"String\">");
    excelDoc.Write(x[y].ToString());
    excelDoc.Write("</Data></Cell>");
    break;
case "System.Int16":
case "System.Int32":
case "System.Int64":
case "System.Byte":
    excelDoc.Write("<Cell ss:StyleID=\"Integer\">" +
    "<Data ss:Type=\"Number\">");
    excelDoc.Write(x[y].ToString());
    excelDoc.Write("</Data></Cell>");
    break;
case "System.Decimal":
case "System.Double":
    excelDoc.Write("<Cell ss:StyleID=\"Decimal\">" +
    "<Data ss:Type=\"Number\">");
    excelDoc.Write(x[y].ToString());
    excelDoc.Write("</Data></Cell>");
    break;
case "System.DBNull":
    excelDoc.Write("<Cell ss:StyleID=\"StringLiteral\">" +
    "<Data ss:Type=\"String\">");
    excelDoc.Write("");
    excelDoc.Write("</Data></Cell>");
    break;
default:

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        throw (new Exception(rowType.ToString() + " not handled.));
    }
}
excelDoc.Write("</Row>");
}
excelDoc.Write("</Table>");
excelDoc.Write("</Worksheet>");
excelDoc.Write(endExcelXML);
excelDoc.Close();
}

private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    //int bn;
    //this.Invoke(new EventHandler(AddRecieve1));

    int bytes = serialPort1.BytesToRead;

    // Create a byte array buffer to hold the incoming data
    byte[] buffer = new byte[bytes];

    // Read the data from the port and store it in our buffer
    serialPort1.Read(buffer, 0, bytes);

    Log(ByteArrayToHexString(buffer));
}

private void Log(string msg)
{
    richTextBox1.Invoke(new EventHandler(delegate
    {
        richTextBox1.SelectedText = string.Empty;
        richTextBox1.AppendText(msg);
        richTextBox1.ScrollToCaret();
    }));
}

private string ByteArrayToHexString(byte[] data)
{
    buffer_len += data.Length;
    StringBuilder sb = new StringBuilder(data.Length);
    for (int i = 0; i < data.Length; i++)
        sb.Append(Convert.ToString(data[i]) + ",");
    //{
        //b[i] = int.Parse(Convert.ToString(data[i]));

    //}
    return sb.ToString();
}

void controlme(int sec_flag)
{
    try
    {

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

string kalan_veri = string.Empty;
int veri_sayisi = 0;
string esas_veri = richTextBox1.Text;
int flag = 0;
int[] veri = new int[buffer_len];
int k = 0;
StringTokenizer st = new StringTokenizer(esas_veri, ",");
//If an empty delimiter is given, automatically uses " " (space).
while (st.hasMoreTokens())
{
    veri[k] = int.Parse(st.nextToken());
    k++;
}
k--;

for (int i = 0; i < k; i++)
{
    if (flag == 0)
        if (veri[i] == 52 && veri[i + 1] == 54 && veri[i + 2] == 52)
        {
            veri_sayisi = i;
            MessageBox.Show("veri alindi");
            flag = 1;
        }
    if (flag == 1)
    {
        if (veri[i].ToString() == "58")
            kalan_veri += ":";
        else if (veri[i].ToString() == "46")
            kalan_veri += ":";
        else if (veri[i].ToString() != ",")
            kalan_veri += ascii2decimal(veri[i]).ToString();
    }
}

string[] ma_deger = new string[10];
int kx = 0;
StringTokenizer st2 = new StringTokenizer(kalan_veri, ":");
//If an empty delimiter is given, automatically uses " " (space).
while (st2.hasMoreTokens())
{
    ma_deger[kx] = st2.nextToken();
    kx++;
}

if (sec_flag == 1)
{
    istatistik += "a," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString()+"\r\n";
    textBox30.Text = ma_deger[1].ToString();
    textBox20.Text = ma_deger[2].ToString();
    textBox1.Text = ma_deger[3].ToString();
}
else if (sec_flag == 2)
{

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        istatistik += "b," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox29.Text = ma_deger[1].ToString();
        textBox19.Text = ma_deger[2].ToString();
        textBox2.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 3)
    {
        istatistik += "c," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox28.Text = ma_deger[1].ToString();
        textBox18.Text = ma_deger[2].ToString();
        textBox3.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 4)
    {
        istatistik += "d," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox27.Text = ma_deger[1].ToString();
        textBox17.Text = ma_deger[2].ToString();
        textBox4.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 5)
    {
        istatistik += "e," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox26.Text = ma_deger[1].ToString();
        textBox16.Text = ma_deger[2].ToString();
        textBox5.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 6)
    {
        istatistik += "f," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox25.Text = ma_deger[1].ToString();
        textBox15.Text = ma_deger[2].ToString();
        textBox6.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 7)
    {
        istatistik += "g," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox24.Text = ma_deger[1].ToString();
        textBox14.Text = ma_deger[2].ToString();
        textBox7.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 8)
    {
        istatistik += "h," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox23.Text = ma_deger[1].ToString();
        textBox13.Text = ma_deger[2].ToString();
        textBox8.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 9)

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

    {
        istatistik += "i," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox22.Text = ma_deger[1].ToString();
        textBox12.Text = ma_deger[2].ToString();
        textBox9.Text = ma_deger[3].ToString();
    }
    else if (sec_flag == 10)
    {
        istatistik += "j," + ma_deger[1].ToString() + "," + ma_deger[2].ToString() + "," +
ma_deger[3].ToString() + "\r\n";
        textBox21.Text = ma_deger[1].ToString();
        textBox11.Text = ma_deger[2].ToString();
        textBox10.Text = ma_deger[3].ToString();
    }

    if (sec_flag == 1)
    {
        checkBox1.Enabled = true;
        checkBox11.Checked = true;
        hesapla(1, veri_sayisi, veri);
    }
    else if (sec_flag == 2)
    {
        checkBox2.Enabled = true;
        checkBox12.Checked = true;
        hesapla(2, veri_sayisi, veri);
    }
    else if (sec_flag == 3)
    {
        checkBox3.Enabled = true;
        checkBox13.Checked = true;
        hesapla(3, veri_sayisi, veri);
    }
    else if (sec_flag == 4)
    {
        checkBox4.Enabled = true;
        checkBox14.Checked = true;
        hesapla(4, veri_sayisi, veri);
    }
    else if (sec_flag == 5)
    {
        checkBox5.Enabled = true;
        checkBox15.Checked = true;
        hesapla(5, veri_sayisi, veri);
    }
    else if (sec_flag == 6)
    {
        checkBox6.Enabled = true;
        checkBox16.Checked = true;
        hesapla(6, veri_sayisi, veri);
    }
    else if (sec_flag == 7)
    {
        checkBox7.Enabled = true;

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        checkBox17.Checked = true;
        hesapla(7, veri_sayisi, veri);
    }
    else if (sec_flag == 8)
    {
        checkBox8.Enabled = true;
        checkBox18.Checked = true;
        hesapla(8, veri_sayisi, veri);
    }
    else if (sec_flag == 9)
    {
        checkBox9.Enabled = true;
        checkBox19.Checked = true;
        hesapla(9, veri_sayisi, veri);
    }
    else if (sec_flag == 10)
    {
        checkBox10.Enabled = true;
        checkBox20.Checked = true;
        hesapla(10, veri_sayisi, veri);
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
}

int ascii2decimal(int a)
{
    int q = 0;
    if (a.ToString() == "48")
        q = 0;
    else if (a.ToString() == "49")
        q = 1;
    else if (a.ToString() == "50")
        q = 2;
    else if (a.ToString() == "51")
        q = 3;
    else if (a.ToString() == "52")
        q = 4;
    else if (a.ToString() == "53")
        q = 5;
    else if (a.ToString() == "54")
        q = 6;
    else if (a.ToString() == "55")
        q = 7;
    else if (a.ToString() == "56")
        q = 8;
    else if (a.ToString() == "57")
        q = 9;

    return q;
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

void hesapla(int flag, int sayi, int [] veri)
{
    int [] hesaplanan = new int[sayi];
    int hesap_sayi = (sayi / 2)+1;
    int [] hesap = new int[hesap_sayi];
    int flag_say = 0;
    int j = 0;
    try
    {

        for (int i = 0; i < (sayi - 2); i = i + 2)
            hesaplanan[i] = veri[i] + veri[i + 1] * 256;

        while (sayi > flag_say)
        {
            hesap[j] = hesaplanan[flag_say];
            j++;
            flag_say += 2;
        }

        dataset_ekle(flag, j, hesap);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

public void dataset_ekle(int flag, int sayi, int [] veri)
{
    try
    {
        if (flag == 1)
            row_count[0] = sayi;
        if (flag == 2)
            row_count[1] = sayi;
        if (flag == 3)
            row_count[2] = sayi;
        if (flag == 4)
            row_count[3] = sayi;
        if (flag == 5)
            row_count[4] = sayi;
        if (flag == 6)
            row_count[5] = sayi;
        if (flag == 7)
            row_count[6] = sayi;
        if (flag == 8)
            row_count[7] = sayi;
        if (flag == 9)
            row_count[8] = sayi;
        if (flag == 10)
            row_count[9] = sayi;
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
DataRow dr;
for (int i = 0; i < sayi; i++)
{
    if (flag == 1)
    {
        dr = t1.Tables[0].NewRow();
        dr["a"] = veri[i];
        t1.Tables[0].Rows.Add(new object[] { dr["a"] });
    }
    else if (flag == 2)
    {
        dr = t2.Tables[0].NewRow();
        dr["b"] = veri[i];
        t2.Tables[0].Rows.Add(new object[] { dr["b"] });
    }
    else if (flag == 3)
    {
        dr = t3.Tables[0].NewRow();
        dr["c"] = veri[i];
        t3.Tables[0].Rows.Add(new object[] { dr["c"] });
    }
    else if (flag == 4)
    {
        dr = t4.Tables[0].NewRow();
        dr["d"] = veri[i];
        t4.Tables[0].Rows.Add(new object[] { dr["d"] });
    }
    else if (flag == 5)
    {
        dr = t5.Tables[0].NewRow();
        dr["e"] = veri[i];
        t5.Tables[0].Rows.Add(new object[] { dr["e"] });
    }
    else if (flag == 6)
    {
        dr = t6.Tables[0].NewRow();
        dr["f"] = veri[i];
        t6.Tables[0].Rows.Add(new object[] { dr["f"] });
    }
    else if (flag == 7)
    {
        dr = t7.Tables[0].NewRow();
        dr["g"] = veri[i];
        t7.Tables[0].Rows.Add(new object[] { dr["g"] });
    }
    else if (flag == 8)
    {
        dr = t8.Tables[0].NewRow();
        dr["h"] = veri[i];
        t8.Tables[0].Rows.Add(new object[] { dr["h"] });
    }
    else if (flag == 9)
    {
        dr = t9.Tables[0].NewRow();
        dr["i"] = veri[i];
    }
}
```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        t9.Tables[0].Rows.Add(new object[] { dr["i"] });
    }
    else if (flag == 10)
    {
        dr = t10.Tables[0].NewRow();
        dr["j"] = veri[i];
        t10.Tables[0].Rows.Add(new object[] { dr["j"] });
    }
    //hakan_main.Tables[0].Rows.Add(new object[] { dr["a"], dr["b"], dr["c"], dr["d"],
    dr["e"], dr["f"], dr["g"], dr["h"], dr["i"], dr["j"] });
}
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}

//hakan_main_source = new BindingSource(hakan_main, "OUT1"); //uraz
//dataGridView1.DataSource = hakantez; //uraz
//exportToExcel(hakan_main, "C:/out1.xls");
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    try
    {
        serialPort1.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void buttonSend_Click(object sender, EventArgs e)
{
    pictureBox1.ImageLocation = @"c:\masmetre\chart\blankchart.png";
    try
    {
        if (!mynameistrk)
        {
            button1.Enabled = true;
            button2.Enabled = true;
            button3.Enabled = true;
            button4.Enabled = true;
            button5.Enabled = true;
            button6.Enabled = true;
            button7.Enabled = true;
            button8.Enabled = true;
            button9.Enabled = true;
            button10.Enabled = true;
            button11.Enabled = false;
        }
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

buttonSend.Text = "&Dur";
serialPort1.BaudRate = int.Parse(baudA.SelectedItem.ToString());
serialPort1.PortName = "COM" + comA.SelectedItem.ToString();

baudA.Enabled = false;
comA.Enabled = false;
try
{
    serialPort1.ReadBufferSize = 2097152;
    serialPort1.Encoding = Encoding.GetEncoding(28591);
    serialPort1.Open();

    cA.Text = serialPort1.PortName;
    bA.Text = serialPort1.BaudRate.ToString();
}
catch (Exception ee)
{
    MessageBox.Show(ee.ToString());
}

mynameistrk = true;
}
else
{
    button1.Enabled = false;
    button2.Enabled = false;
    button3.Enabled = false;
    button4.Enabled = false;
    button5.Enabled = false;
    button6.Enabled = false;
    button7.Enabled = false;
    button8.Enabled = false;
    button9.Enabled = false;
    button10.Enabled = false;
    buttonSend.Enabled = false;
    button11.Enabled = true;
    button12.Enabled = false;
    buttonSend.Text = "&Başlat";
    try
    {
        serialPort1.Close();
    }
    catch (Exception ee)
    {
        MessageBox.Show(ee.ToString());
    }
    mynameistrk = false;
    baudA.Enabled = true;
    comA.Enabled = true;
    cA.Text = "";
    bA.Text = "";
}
}
catch (Exception ex)
{

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void Form1_Load_1(object sender, EventArgs e)
{
    if (!Directory.Exists("C:/masmetre"))
        Directory.CreateDirectory("C:/masmetre");
    if (!Directory.Exists("C:/masmetre/chart"))
        Directory.CreateDirectory("C:/masmetre/chart");
    if (!Directory.Exists("C:/masmetre/excel"))
        Directory.CreateDirectory("C:/masmetre/excel");
    progressBar1.Maximum = 200;
    progressBar1.Minimum = 0;
    foreach (Process process in Process.GetProcessesByName("Excel"))
    {
        process.Kill();
    }
    textBox1.ReadOnly = true;
    textBox2.ReadOnly = true;
    textBox3.ReadOnly = true;
    textBox4.ReadOnly = true;
    textBox5.ReadOnly = true;
    textBox6.ReadOnly = true;
    textBox7.ReadOnly = true;
    textBox8.ReadOnly = true;
    textBox9.ReadOnly = true;
    textBox10.ReadOnly = true;
    textBox11.ReadOnly = true;
    textBox12.ReadOnly = true;
    textBox13.ReadOnly = true;
    textBox14.ReadOnly = true;
    textBox15.ReadOnly = true;
    textBox16.ReadOnly = true;
    textBox17.ReadOnly = true;
    textBox18.ReadOnly = true;
    textBox19.ReadOnly = true;
    textBox20.ReadOnly = true;
    textBox21.ReadOnly = true;
    textBox22.ReadOnly = true;
    textBox23.ReadOnly = true;
    textBox24.ReadOnly = true;
    textBox25.ReadOnly = true;
    textBox26.ReadOnly = true;
    textBox27.ReadOnly = true;
    textBox28.ReadOnly = true;
    textBox29.ReadOnly = true;
    textBox30.ReadOnly = true;

    checkBox1.Enabled = false;
    checkBox2.Enabled = false;
    checkBox3.Enabled = false;
    checkBox4.Enabled = false;
    checkBox5.Enabled = false;
    checkBox6.Enabled = false;
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

checkBox7.Enabled = false;
checkBox8.Enabled = false;
checkBox9.Enabled = false;
checkBox10.Enabled = false;
checkBox11.Enabled = false;
checkBox12.Enabled = false;
checkBox13.Enabled = false;
checkBox14.Enabled = false;
checkBox15.Enabled = false;
checkBox16.Enabled = false;
checkBox17.Enabled = false;
checkBox18.Enabled = false;
checkBox19.Enabled = false;
checkBox20.Enabled = false;

for (int i = 0; i < 10; i++)
{
    uzun[i] = 0;
    deger[i] = string.Empty;
}

pictureBox1.ImageLocation = @"c:\masmetre\chart\test.bmp";

comA.SelectedIndex = 7;
baudA.SelectedIndex = 3;

button1.Enabled = false;
button2.Enabled = false;
button3.Enabled = false;
button4.Enabled = false;
button5.Enabled = false;
button6.Enabled = false;
button7.Enabled = false;
button8.Enabled = false;
button9.Enabled = false;
button10.Enabled = false;
buttonSend.Enabled = true;
button11.Enabled = false;
button12.Enabled = false;
eni.SelectedIndex = 1;
boyu.SelectedIndex = 1;
}

private void button11_Click(object sender, EventArgs e)
{
    try
    {
        richTextBox1.Text = string.Empty;

        button1.Enabled = false;
        button2.Enabled = false;
        button3.Enabled = false;
        button4.Enabled = false;
        button5.Enabled = false;
        button6.Enabled = false;
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
button7.Enabled = false;
button8.Enabled = false;
button9.Enabled = false;
button10.Enabled = false;
buttonSend.Enabled = false;
button11.Enabled = true;
button12.Enabled = false;

checkBox1.Enabled = false;
checkBox2.Enabled = false;
checkBox3.Enabled = false;
checkBox4.Enabled = false;
checkBox5.Enabled = false;
checkBox6.Enabled = false;
checkBox7.Enabled = false;
checkBox8.Enabled = false;
checkBox9.Enabled = false;
checkBox10.Enabled = false;
checkBox11.Enabled = false;
checkBox12.Enabled = false;
checkBox13.Enabled = false;
checkBox14.Enabled = false;
checkBox15.Enabled = false;
checkBox16.Enabled = false;
checkBox17.Enabled = false;
checkBox18.Enabled = false;
checkBox19.Enabled = false;
checkBox20.Enabled = false;

for (int i = 0; i < 10; i++)
{
    uzun[i] = 0;
    deger[i] = string.Empty;
}

textBox1.Text = string.Empty;
textBox2.Text = string.Empty;
textBox3.Text = string.Empty;
textBox4.Text = string.Empty;
textBox5.Text = string.Empty;
textBox6.Text = string.Empty;
textBox7.Text = string.Empty;
textBox8.Text = string.Empty;
textBox9.Text = string.Empty;
textBox10.Text = string.Empty;
textBox11.Text = string.Empty;
textBox12.Text = string.Empty;
textBox13.Text = string.Empty;
textBox14.Text = string.Empty;
textBox15.Text = string.Empty;
textBox16.Text = string.Empty;
textBox17.Text = string.Empty;
textBox18.Text = string.Empty;
textBox19.Text = string.Empty;
textBox20.Text = string.Empty;
```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

textBox21.Text = string.Empty;
textBox22.Text = string.Empty;
textBox23.Text = string.Empty;
textBox24.Text = string.Empty;
textBox25.Text = string.Empty;
textBox26.Text = string.Empty;
textBox27.Text = string.Empty;
textBox28.Text = string.Empty;
textBox29.Text = string.Empty;
textBox30.Text = string.Empty;

comA.SelectedIndex = 0;
baudA.SelectedIndex = 0;

buttonSend.Enabled = true;
button11.Enabled = false;

pictureBox1.ImageLocation = "test.bmp";

buttonSend.Text = "&Başlat";
mynameistrk = true;
baudA.Enabled = true;
comA.Enabled = false;
cA.Text = "";
bA.Text = "";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        t1_set();
        controlme(1);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        t2_set();
        controlme(2);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
    }  
}  
  
private void button3_Click(object sender, EventArgs e)  
{  
    try  
    {  
        t3_set();  
        controlme(3);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message + " Tekrar Deneyin");  
    }  
}  
  
private void button4_Click(object sender, EventArgs e)  
{  
    try  
    {  
        t4_set();  
        controlme(4);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message + " Tekrar Deneyin");  
    }  
}  
  
private void button5_Click(object sender, EventArgs e)  
{  
    try  
    {  
        t5_set();  
        controlme(5);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message + " Tekrar Deneyin");  
    }  
}  
  
private void button6_Click(object sender, EventArgs e)  
{  
    try  
    {  
        t6_set();  
        controlme(6);  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message + " Tekrar Deneyin");  
    }  
}
```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
private void button7_Click(object sender, EventArgs e)
{
    try
    {
        t7_set();
        controlme(7);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void button8_Click(object sender, EventArgs e)
{
    try
    {
        t8_set();
        controlme(8);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void button9_Click(object sender, EventArgs e)
{
    try
    {
        t9_set();
        controlme(9);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void button10_Click(object sender, EventArgs e)
{
    try
    {
        t10_set();
        controlme(10);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void button12_Click(object sender, EventArgs e)
{
    if (uzanti == "excel")
```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        coklu_graf_eski();
    else
    {
        if (checkBox11.Checked == false && checkBox12.Checked == false &&
checkBox13.Checked == false && checkBox14.Checked == false
        && checkBox15.Checked == false && checkBox16.Checked == false &&
checkBox17.Checked == false && checkBox18.Checked == false
        && checkBox19.Checked == false && checkBox20.Checked == false)
            MessageBox.Show("Grafik çizdirmek için, en az 1 ölçüm almalısınız.");
        else
        {
            if (checkBox1.Checked == false && checkBox2.Checked == false &&
checkBox3.Checked == false && checkBox4.Checked == false
            && checkBox5.Checked == false && checkBox6.Checked == false &&
checkBox7.Checked == false && checkBox8.Checked == false
            && checkBox9.Checked == false && checkBox10.Checked == false)
                MessageBox.Show("Grafik çizdirmek için, en az 1 ölçüm seçmelisiniz");
            else
            {
                try
                {
                    foreach (Process process in Process.GetProcessesByName("EXCEL"))
                    {
                        process.Kill();
                    }
                    coklu_graf();
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message + " Tekrar Deneyin");
                    foreach (Process process in Process.GetProcessesByName("EXCEL"))
                    {
                        process.Kill();
                    }
                }
            }
        }
    }
    pictureBox1.ImageLocation = mevcut_folder_2 + "/" + "graf" + graf_indis.ToString() +
".png";
}

public void coklu_graf_eski()
{
    button12.Enabled = false;
    try
    {
        index_graf = string.Empty;
        int sayi = big;

        int ix = 0;
        int kx = 1;
        Microsoft.Office.Interop.Excel.Application excelApp = new
Microsoft.Office.Interop.Excel.ApplicationClass();
        progressBar1.Value = 0;
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

indicator = (200 / (bolum));
for (int i = 0; i < bolum; i++)
{
    //excelApp.Visible = true;
    //Excel.Workbook newWorkbook =
excelApp.Workbooks.Add(XIWBA_Template.xlWBATWorksheet);
    string workbookPath = dosya[i];

    Microsoft.Office.Interop.Excel.Workbook excelWorkbook =
excelApp.Workbooks.Open(workbookPath,
        0, false, 5, "", "", false, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "",
        true, false, 0, true, false, false);
    Microsoft.Office.Interop.Excel.Sheets excelSheets = excelWorkbook.Worksheets;
    string currentSheet = "Sheet1";
    Microsoft.Office.Interop.Excel.Worksheet excelWorksheet =
        (Microsoft.Office.Interop.Excel.Worksheet)excelSheets.get_Item(currentSheet);

    string lim2 = "A";

    if (graf_count == 1)
        lim2 = "A";
    else if (graf_count == 2)
        lim2 = "B";
    else if (graf_count == 3)
        lim2 = "C";
    else if (graf_count == 4)
        lim2 = "D";
    else if (graf_count == 5)
        lim2 = "E";
    else if (graf_count == 6)
        lim2 = "F";
    else if (graf_count == 7)
        lim2 = "G";
    else if (graf_count == 8)
        lim2 = "H";
    else if (graf_count == 9)
        lim2 = "I";
    else if (graf_count == 10)
        lim2 = "J";

    int kalan = sayi % 32000;
    //if (limit < 32000)
    if (i == (bolum - 1))
        lim2 += kalan.ToString();
    else
        lim2 += "32000";
    //else
    // lim2 += "32000";
    string lim1 = "A1";

    Microsoft.Office.Interop.Excel.Range excelCell =
(Microsoft.Office.Interop.Excel.Range)excelWorksheet.get_Range(lim1, lim2);

    //A30000 in yerine lim2 yazarsan gelen tüm karakterleri yazmaya çalışır.

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

ChartObjects chartObjs = (ChartObjects)excelWorksheet.ChartObjects(Type.Missing);
ChartObject chartObj = chartObjs.Add(100, 20, 300, 300);
Chart xlChart = chartObj.Chart;

xlChart.SetSourceData(excelCell, Type.Missing);
xlChart.ChartType = XlChartType.xlLine;

while (kx != 0)
    if (!Directory.Exists("C:/masmetre/chart/" + budur + "_" + ix.ToString()))
    {
        kx = 0;
        Directory.CreateDirectory("C:/masmetre/chart/" + budur + "_" + ix.ToString());
        mevcut_folder_2 = "C:/masmetre/chart/" + budur + "_" + ix.ToString();
    }
    else ix++;
xlChart.Export(mevcut_folder_2 + "/" + "graf" + i.ToString() + ".png", "PNG", false);
index_graf += mevcut_folder_2 + "/" + "graf" + i.ToString() + ".png" + "\r\n";
graf_sayisi++;
//xlChart.CopyPicture(Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen,
Microsoft.Office.Interop.Excel.XlCopyPictureFormat.xlBitmap,
Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen);
excelWorkbook.Close(false, false, false);
progressBar1.Value += indicator;
}
kayit_graf(index_graf);
excelApp.Quit();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
}

public void coklu_graf()
{
    button12.Enabled = false;
    try
    {
        index_graf = string.Empty;
        int sayi = big;

        Microsoft.Office.Interop.Excel.Application excelApp = new
Microsoft.Office.Interop.Excel.ApplicationClass();
        progressBar1.Value = 0;
        indicator = (200 / (bolum));
        int kx = 1;
        int ix = 0;
        for (int i = 0; i < bolum; i++)
        {

            //excelApp.Visible = true;
            //Excel.Workbook newWorkbook =
excelApp.Workbooks.Add(XlWBATemplate.xlWBATWorksheet);

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

string workbookPath = mevcut_folder_1 + "/veri_" + tarih[i] + zaman[i] + i.ToString() +
".xls";

Microsoft.Office.Interop.Excel.Workbook excelWorkbook =
excelApp.Workbooks.Open(workbookPath,
    0, false, 5, "", "", false, Microsoft.Office.Interop.Excel.XIPlatform.xlWindows, "",
    true, false, 0, true, false, false);
Microsoft.Office.Interop.Excel.Sheets excelSheets = excelWorkbook.Worksheets;
string currentSheet = "Sheet1";
Microsoft.Office.Interop.Excel.Worksheet excelWorksheet =
    (Microsoft.Office.Interop.Excel.Worksheet)excelSheets.get_Item(currentSheet);

string lim2 = "A";

if (graf_count == 1)
    lim2 = "A";
else if (graf_count == 2)
    lim2 = "B";
else if (graf_count == 3)
    lim2 = "C";
else if (graf_count == 4)
    lim2 = "D";
else if (graf_count == 5)
    lim2 = "E";
else if (graf_count == 6)
    lim2 = "F";
else if (graf_count == 7)
    lim2 = "G";
else if (graf_count == 8)
    lim2 = "H";
else if (graf_count == 9)
    lim2 = "I";
else if (graf_count == 10)
    lim2 = "J";

int kalan = sayi % 32000;
//if (limit < 32000)
if (i == (bolum - 1))
    lim2 += kalan.ToString();
else
    lim2 += "32000";
//else
// lim2 += "32000";
string lim1 = "A1";

Microsoft.Office.Interop.Excel.Range excelCell =
(Microsoft.Office.Interop.Excel.Range)excelWorksheet.get_Range(lim1, lim2);

//A30000 in yerine lim2 yazarsan gelen tüm karakterleri yazmaya çalışır.

ChartObjects chartObjs = (ChartObjects)excelWorksheet.ChartObjects(Type.Missing);
ChartObject chartObj = chartObjs.Add(100, 20, 300, 300);
Chart xlChart = chartObj.Chart;

xlChart.SetSourceData(excelCell, Type.Missing);

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

xlChart.ChartType = XlChartType.xlLine;

while (kx != 0)
    if (!Directory.Exists("C:/masmetre/chart/" + budur + "_" + ix.ToString()))
    {
        kx = 0;
        Directory.CreateDirectory("C:/masmetre/chart/" + budur + "_" + ix.ToString());
        mevcut_folder_2 = "C:/masmetre/chart/" + budur + "_" + ix.ToString();
    }
    else ix++;
xlChart.Export(mevcut_folder_2 + "/" + "graf" + i.ToString() + ".png", "PNG", false);
index_graf += mevcut_folder_2 + "/" + "graf" + i.ToString() + ".png" + "\r\n";
graf_sayisi++;
//xlChart.CopyPicture(Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen,
Microsoft.Office.Interop.Excel.XlCopyPictureFormat.xlBitmap,
Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen);
excelWorkbook.Close(false, false, false);
progressBar1.Value += indicator;
}
kayit_graf(index_graf);
excelApp.Quit();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
}
public void coklu_graf_custom(char hangisi)
{
    progressBar1.Value = 0;
    try
    {
        int sayi = big;
        Microsoft.Office.Interop.Excel.Application excelApp = new
Microsoft.Office.Interop.Excel.ApplicationClass();
        int i = int.Parse(hangisi.ToString());
        //for (int i = 0; i < bolum; i++)
        //{
            //excelApp.Visible = true;
            //Excel.Workbook newWorkbook =
excelApp.Workbooks.Add(XlWBATemplate.xlWBATWorksheet);
            string workbookPath = mevcut_folder_1 + "/veri_" + tarih[i] + zaman[i] + i.ToString() +
".xls";

            if(uzanti=="excel")
                workbookPath = dosya[i];

            Microsoft.Office.Interop.Excel.Workbook excelWorkbook =
excelApp.Workbooks.Open(workbookPath,
                0, false, 5, "", "", false, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "",
                true, false, 0, true, false, false);
            Microsoft.Office.Interop.Excel.Sheets excelSheets = excelWorkbook.Worksheets;
            string currentSheet = "Sheet1";

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

Microsoft.Office.Interop.Excel.Worksheet excelWorksheet =
    (Microsoft.Office.Interop.Excel.Worksheet)excelSheets.get_Item(currentSheet);

string lim2 = "A";

if (graf_count == 1)
    lim2 = "A";
else if (graf_count == 2)
    lim2 = "B";
else if (graf_count == 3)
    lim2 = "C";
else if (graf_count == 4)
    lim2 = "D";
else if (graf_count == 5)
    lim2 = "E";
else if (graf_count == 6)
    lim2 = "F";
else if (graf_count == 7)
    lim2 = "G";
else if (graf_count == 8)
    lim2 = "H";
else if (graf_count == 9)
    lim2 = "I";
else if (graf_count == 10)
    lim2 = "J";

int kalan = sayi % 32000;
//if (limit < 32000)
if (i == (bolum - 1))
    lim2 += kalan.ToString();
else
    lim2 += "32000";
//else
// lim2 += "32000";
string lim1 = "A1";

Microsoft.Office.Interop.Excel.Range excelCell =
(Microsoft.Office.Interop.Excel.Range)excelWorksheet.get_Range(lim1, lim2);

//A30000 in yerine lim2 yazarsan gelen tüm karakterleri yazmaya çalışır.

ChartObjects chartObjs = (ChartObjects)excelWorksheet.ChartObjects(Type.Missing);
ChartObject chartObj = chartObjs.Add(100, 20, int.Parse(eni.SelectedItem.ToString()),
int.Parse(boyu.SelectedItem.ToString()));
Chart xlChart = chartObj.Chart;

xlChart.SetSourceData(excelCell, Type.Missing);
xlChart.ChartType = XlChartType.xlLine;

xlChart.Export(@"c:\masmetre\chart\custom.bmp", "BMP", false);
progressBar1.Value = 200;
Process.Start("rundll32", "shimgvw.dll,ImageView_Fullscreen
c:\masmetre\chart\custom.bmp");

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        //xlChart.CopyPicture(Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen,
Microsoft.Office.Interop.Excel.XlCopyPictureFormat.xlBitmap,
Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen);
        excelWorkbook.Close(false, false, false);

        //}
        excelApp.Quit();

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

public void graf_comp(string path, int comp)
{
    try
    {
        Microsoft.Office.Interop.Excel.Application excelApp = new
Microsoft.Office.Interop.Excel.ApplicationClass();
        progressBar1.Value = 0;
        string workbookPath = path;

        Microsoft.Office.Interop.Excel.Workbook excelWorkbook =
excelApp.Workbooks.Open(workbookPath,
        0, false, 5, "", "", false, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, "",
        true, false, 0, true, false, false);
        Microsoft.Office.Interop.Excel.Sheets excelSheets = excelWorkbook.Worksheets;
        string currentSheet = "Sheet1";
        Microsoft.Office.Interop.Excel.Worksheet excelWorksheet =
        (Microsoft.Office.Interop.Excel.Worksheet)excelSheets.get_Item(currentSheet);

        //excelWorksheet.UsedRange.Count;
        //string lim2 = "A";

        //if (graf_count == 1)
        //    lim2 = "A";
        //else if (graf_count == 2)
        //    lim2 = "B";
        //else if (graf_count == 3)
        //    lim2 = "C";
        //else if (graf_count == 4)
        //    lim2 = "D";
        //else if (graf_count == 5)
        //    lim2 = "E";
        //else if (graf_count == 6)
        //    lim2 = "F";
        //else if (graf_count == 7)
        //    lim2 = "G";
        //else if (graf_count == 8)
        //    lim2 = "H";
        //else if (graf_count == 9)
        //    lim2 = "I";
        //else if (graf_count == 10)

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

    // lim2 = "J";

    ///lim2 += sayi.ToString();
    ///else
    /// lim2 += "32000";
    ///string lim1 = "A1";

    Microsoft.Office.Interop.Excel.Range excelCell =
(Microsoft.Office.Interop.Excel.Range)excelWorksheet.UsedRange;

    //A30000 in yerine lim2 yazarsan gelen tüm karakterleri yazmaya çalışır.

    ChartObjects chartObjs = (ChartObjects)excelWorksheet.ChartObjects(Type.Missing);
    ChartObject chartObj = chartObjs.Add(100, 20, 500, 300);
    Chart xlChart = chartObj.Chart;

    xlChart.SetSourceData(excelCell, Type.Missing);
    xlChart.ChartType = XlChartType.xlLine;

    xlChart.Export(@"c:\masmetre\chart\comp"+comp.ToString()+".bmp", "BMP", false);
    progressBar1.Value = 200;
    //Process.Start("rundll32", "shimgvw.dll,ImageView_Fullscreen
c:\masmetre\chart\custom.bmp");
    //xlChart.CopyPicture(Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen,
Microsoft.Office.Interop.Excel.XlCopyPictureFormat.xlBitmap,
Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen);
    excelWorkbook.Close(false, false, false);

    //}
    excelApp.Quit();

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
}

public void kontrolet()
{
    if (checkBox11.Checked == false)
    {
        textBox30.Text = string.Empty;
        textBox20.Text = string.Empty;
        textBox1.Text = string.Empty;
    }
    if (checkBox12.Checked == false)
    {
        textBox29.Text = string.Empty;
        textBox19.Text = string.Empty;
        textBox2.Text = string.Empty;
    }
    if (checkBox13.Checked == false)
    {
        textBox28.Text = string.Empty;

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        textBox18.Text = string.Empty;
        textBox3.Text = string.Empty;
    }
    if (checkBox14.Checked == false)
    {
        textBox27.Text = string.Empty;
        textBox17.Text = string.Empty;
        textBox4.Text = string.Empty;
    }
    if (checkBox15.Checked == false)
    {
        textBox26.Text = string.Empty;
        textBox16.Text = string.Empty;
        textBox5.Text = string.Empty;
    }
    if (checkBox16.Checked == false)
    {
        textBox25.Text = string.Empty;
        textBox15.Text = string.Empty;
        textBox6.Text = string.Empty;
    }
    if (checkBox17.Checked == false)
    {
        textBox24.Text = string.Empty;
        textBox14.Text = string.Empty;
        textBox7.Text = string.Empty;
    }
    if (checkBox18.Checked == false)
    {
        textBox23.Text = string.Empty;
        textBox13.Text = string.Empty;
        textBox8.Text = string.Empty;
    }
    if (checkBox19.Checked == false)
    {
        textBox22.Text = string.Empty;
        textBox12.Text = string.Empty;
        textBox9.Text = string.Empty;
    }
    if (checkBox20.Checked == false)
    {
        textBox21.Text = string.Empty;
        textBox11.Text = string.Empty;
        textBox10.Text = string.Empty;
    }
}

private void button19_Click(object sender, EventArgs e)
{
    kontrolet();
    progressBar1.Value = 0;
    try
    {
        richTextBox1.Text = string.Empty;
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        serialPort1.Write("f");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void button20_Click(object sender, EventArgs e)
{
    kontrolet();
    progressBar1.Value = 0;
    try
    {
        richTextBox1.Text = string.Empty;

        serialPort1.Write("n");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    if (uzanti != ".graf")
    {
        string graf = pictureBox1.ImageLocation;
        int graf_kod = graf.IndexOf(".png");

        graf_kod = graf_kod - 1;
        char char_kod = graf[graf_kod];
        coklu_graf_custom(char_kod);
    }
}

private void button13_Click_1(object sender, EventArgs e)
{
    //excele_at();
    coklu_graf();
}

private void button13_Click(object sender, EventArgs e)
{
    if (graf_indis > 0 || graf_indis > graf_sayisi)
        graf_indis--;
    pictureBox1.ImageLocation = mevcut_folder_2 + "/" + "graf" + graf_indis.ToString() + ".png";
}

private void button14_Click(object sender, EventArgs e)
{
    if (graf_indis >= 0 && graf_indis < (graf_sayisi-1))
        graf_indis++;
    pictureBox1.ImageLocation = mevcut_folder_2 + "/" + "graf" + graf_indis.ToString() + ".png";
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

    }

    private void button15_Click(object sender, EventArgs e)
    {
        Process.Start("rundll32", "shimgvw.dll,ImageView_Fullscreen,ImageView_Actualsize
c:\\masmetre\\chart\\custom.bmp");
    }

    public void kayit_excel(string args)
    {
        StreamWriter SW;
        if (File.Exists(mevcut_folder_1 + "\\index.dat"))
            SW = File.AppendText(mevcut_folder_1 + "\\index.dat");
        else
        {
            SW = File.CreateText(mevcut_folder_1 + "\\index.dat");
            SW.WriteLine("excel");
        }
        SW.WriteLine(args);
        SW.Close();
    }

    public void kayit_graf(string args)
    {
        StreamWriter SW;
        if (File.Exists(mevcut_folder_2 + "\\index.dat"))
            SW = File.AppendText(mevcut_folder_2 + "\\index.dat");
        else
        {
            SW = File.CreateText(mevcut_folder_2 + "\\index.dat");
            SW.WriteLine("graf");
        }
        SW.WriteLine(args);
        SW.Close();
    }

    public void kayit_istatistik(string args)
    {
        StreamWriter SW;
        if (File.Exists(mevcut_folder_1 + "\\value.dat"))
            SW = File.AppendText(mevcut_folder_1 + "\\value.dat");
        else
        {
            SW = File.CreateText(mevcut_folder_1 + "\\value.dat");
            SW.WriteLine("value");
        }
        SW.WriteLine(args);
        SW.Close();
    }

    private void button15_Click_1(object sender, EventArgs e)
    {
        try
        {
            string[] ma_deger = new string[10];

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

char a;
int kx = 0;

OpenFileDialog dlgOpen = new OpenFileDialog();
dlgOpen.Filter = "index(.dat)*.dat";
int i = 0;
int ix = 0;
graf_indis = 0;

if (dlgOpen.ShowDialog() == DialogResult.OK)
{
    StreamReader re = File.OpenText(dlgOpen.FileName);
    string input = null;
    uzanti = re.ReadLine();
    if (uzanti == "excel")
    {
        big = int.Parse(re.ReadLine());
        graf_count = int.Parse(re.ReadLine());
        while ((input = re.ReadLine()) != null)
        {
            dosya[i] = input;
            i++;
        }
        int q = 0;
        q = dlgOpen.FileName.ToString().IndexOf("index.dat");
        string temp = dlgOpen.FileName.ToString().Substring(0, q);
        temp += "value.dat";
        StreamReader ret = File.OpenText(temp);
        string input2 = ret.ReadLine();
        while ((input2 = ret.ReadLine()) != null)
        {
            if (input2.Length > 10)
            {
                kx = 0;
                StringTokenizer st2 = new StringTokenizer(input2, ",");
                //If an empty delimiter is given, automatically uses " " (space).
                a = st2.nextToken()[0];
                while (st2.hasMoreTokens())
                {
                    ma_deger[kx] = st2.nextToken();
                    kx++;
                }
                eski_goster(a, ma_deger);
                ix++;
            }
        }
        button12.Enabled = true;
        bolum = (i - 1);
        MessageBox.Show("Eski Ölçümler Yüklendi");
    }
    else if (uzanti == "graf")
    {
        while ((input = re.ReadLine()) != null)
        {
            dosya[i] = input;

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        i++;
    }
    graf_sayisi = (i - 1);
    int q = 0;
    q = dlgOpen.FileName.ToString().IndexOf("index.dat");
    mevcut_folder_2 = dlgOpen.FileName.ToString().Substring(0, q);
    pictureBox1.ImageLocation = mevcut_folder_2 + "graf" + graf_indis.ToString() +
".png";
    MessageBox.Show("Eski Grafikler Yüklendi");
}
else MessageBox.Show("Dosya Tanımlanamadı");
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

public void cek()
{
    checkBox1.Enabled = false;
    checkBox2.Enabled = false;
    checkBox3.Enabled = false;
    checkBox4.Enabled = false;
    checkBox5.Enabled = false;
    checkBox6.Enabled = false;
    checkBox7.Enabled = false;
    checkBox8.Enabled = false;
    checkBox9.Enabled = false;
    checkBox10.Enabled = false;

    button1.Enabled = false;
    button2.Enabled = false;
    button3.Enabled = false;
    button4.Enabled = false;
    button5.Enabled = false;
    button6.Enabled = false;
    button7.Enabled = false;
    button8.Enabled = false;
    button9.Enabled = false;
    button10.Enabled = false;
}

public void cekme()
{
    checkBox1.Enabled = true;
    checkBox2.Enabled = true;
    checkBox3.Enabled = true;
    checkBox4.Enabled = true;
    checkBox5.Enabled = true;
    checkBox6.Enabled = true;
    checkBox7.Enabled = true;
    checkBox8.Enabled = true;
    checkBox9.Enabled = true;
    checkBox10.Enabled = true;
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

button1.Enabled = true;
button2.Enabled = true;
button3.Enabled = true;
button4.Enabled = true;
button5.Enabled = true;
button6.Enabled = true;
button7.Enabled = true;
button8.Enabled = true;
button9.Enabled = true;
button10.Enabled = true;
}

private void button17_Click(object sender, EventArgs e)
{
cek();
index_dosya = string.Empty;

ortak1();
ortak2();
big = 0;
int a = 0;
for (int i = 0; i < 10; i++)
    if (big < row_count[i])
        big = row_count[i];

int m = 0;
int otuz = 0;
bolum = (big / 32000) + 1;
try
{
    DataRow dr;

    for (int i = 0; i < big; i++)
    {
        dr = hakan_main[m].Tables[0].NewRow();
        if (row_count[0] > i && checkBox1.Checked)
            dr["a"] = t1.Tables[0].Rows[i]["a"];
        else if (row_count[0] <= i)
            dr["a"] = 0; //boşta kalana sıfır döşe
        if (row_count[1] > i && checkBox2.Checked)
            dr["b"] = t2.Tables[0].Rows[i]["b"];
        else if (row_count[1] <= i)
            dr["b"] = 0;
        if (row_count[2] > i && checkBox3.Checked)
            dr["c"] = t3.Tables[0].Rows[i]["c"];
        else if (row_count[2] <= i)
            dr["c"] = 0;
        if (row_count[3] > i && checkBox4.Checked)
            dr["d"] = t4.Tables[0].Rows[i]["d"];
        else if (row_count[3] <= i)
            dr["d"] = 0;
        if (row_count[4] > i && checkBox5.Checked)
            dr["e"] = t5.Tables[0].Rows[i]["e"];
        else if (row_count[4] <= i)

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        dr["e"] = 0;
        if (row_count[5] > i && checkBox6.Checked)
            dr["f"] = t6.Tables[0].Rows[i]["f"];
        else if (row_count[5] <= i)
            dr["f"] = 0;
        if (row_count[6] > i && checkBox7.Checked)
            dr["g"] = t7.Tables[0].Rows[i]["g"];
        else if (row_count[6] <= i)
            dr["g"] = 0;
        if (row_count[7] > i && checkBox8.Checked)
            dr["h"] = t8.Tables[0].Rows[i]["h"];
        else if (row_count[7] <= i)
            dr["h"] = 0;
        if (row_count[8] > i && checkBox9.Checked)
            dr["i"] = t9.Tables[0].Rows[i]["i"];
        else if (row_count[8] <= i)
            dr["i"] = 0;
        if (row_count[9] > i && checkBox10.Checked)
            dr["j"] = t10.Tables[0].Rows[i]["j"];
        else if (row_count[9] <= i)
            dr["j"] = 0;

        otuz++;
        if (otuz >= 32000)
        {
            m++;
            otuz = 0;
        }
        hakan_main[m].Tables[0].Rows.Add(new object[] { dr["a"], dr["b"], dr["c"], dr["d"],
dr["e"], dr["f"], dr["g"], dr["h"], dr["i"], dr["j"] });
    }

    for (int i = 0; i < 50; i++)
    {
        if (row_count[0] == 0 || checkBox1.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("a");
        if (row_count[1] == 0 || checkBox2.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("b");
        if (row_count[2] == 0 || checkBox3.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("c");
        if (row_count[3] == 0 || checkBox4.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("d");
        if (row_count[4] == 0 || checkBox5.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("e");
        if (row_count[5] == 0 || checkBox6.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("f");
        if (row_count[6] == 0 || checkBox7.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("g");
        if (row_count[7] == 0 || checkBox8.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("h");
        if (row_count[8] == 0 || checkBox9.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("i");
        if (row_count[9] == 0 || checkBox10.Checked == false)
            hakan_main[i].Tables[0].Columns.Remove("j");
    }

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

if (row_count[0] == 0 || checkBox1.Checked == false)
    a++;
if (row_count[1] == 0 || checkBox2.Checked == false)
    a++;
if (row_count[2] == 0 || checkBox3.Checked == false)
    a++;
if (row_count[3] == 0 || checkBox4.Checked == false)
    a++;
if (row_count[4] == 0 || checkBox5.Checked == false)
    a++;
if (row_count[5] == 0 || checkBox6.Checked == false)
    a++;
if (row_count[6] == 0 || checkBox7.Checked == false)
    a++;
if (row_count[7] == 0 || checkBox8.Checked == false)
    a++;
if (row_count[8] == 0 || checkBox9.Checked == false)
    a++;
if (row_count[9] == 0 || checkBox10.Checked == false)
    a++;
graf_count = 10 - a;

int ix = 0;
int kx = 1;
while (kx != 0)
    if (!Directory.Exists("C:/masmetre/olcum/" + budur + "_" + ix.ToString()))
    {
        kx = 0;
        Directory.CreateDirectory("C:/masmetre/olcum/" + budur + "_" + ix.ToString());
        mevcut_folder_1 = "C:/masmetre/olcum/" + budur + "_" + ix.ToString();
    }
    else ix++;
index_dosya += big.ToString() + "\r\n";
index_dosya += graf_count.ToString() + "\r\n";
for (int i = 0; i < bolum; i++)
{
    tarih[i] = DateTime.Now.Day.ToString() + "." + DateTime.Now.Month.ToString() + "."
+ DateTime.Now.Year.ToString() + "_";
    zaman[i] = DateTime.Now.Hour.ToString() + "." + DateTime.Now.Minute.ToString() +
"." + DateTime.Now.Second.ToString() + "." + DateTime.Now.Millisecond.ToString();
    index_dosya += mevcut_folder_1 + "/veri_" + tarih[i] + zaman[i] + i.ToString() + ".xls"
+ "\r\n";
    exportToExcel(hakan_main[i], mevcut_folder_1 + "/veri_" + tarih[i] + zaman[i] +
i.ToString() + ".xls");
}
kayit_excel(index_dosya);
kayit_istatistik(istatistik);
button12.Enabled = true;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
cekme();

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
}  
  
public void eski_goster(char a, string [] ma_deger)  
{  
    if (a == 'a')  
    {  
        textBox30.Text = ma_deger[0].ToString();  
        textBox20.Text = ma_deger[1].ToString();  
        textBox1.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'b')  
    {  
        textBox29.Text = ma_deger[0].ToString();  
        textBox19.Text = ma_deger[1].ToString();  
        textBox2.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'c')  
    {  
        textBox28.Text = ma_deger[0].ToString();  
        textBox18.Text = ma_deger[1].ToString();  
        textBox3.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'd')  
    {  
        textBox27.Text = ma_deger[0].ToString();  
        textBox17.Text = ma_deger[1].ToString();  
        textBox4.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'e')  
    {  
        textBox26.Text = ma_deger[0].ToString();  
        textBox16.Text = ma_deger[1].ToString();  
        textBox5.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'f')  
    {  
        textBox25.Text = ma_deger[0].ToString();  
        textBox15.Text = ma_deger[1].ToString();  
        textBox6.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'g')  
    {  
        textBox24.Text = ma_deger[0].ToString();  
        textBox14.Text = ma_deger[1].ToString();  
        textBox7.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'h')  
    {  
        textBox23.Text = ma_deger[0].ToString();  
        textBox13.Text = ma_deger[1].ToString();  
        textBox8.Text = ma_deger[2].ToString();  
    }  
    else if (a == 'i')  
    {
```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        textBox22.Text = ma_deger[0].ToString();
        textBox12.Text = ma_deger[1].ToString();
        textBox9.Text = ma_deger[2].ToString();
    }
    else if (a == 'j')
    {
        textBox21.Text = ma_deger[0].ToString();
        textBox11.Text = ma_deger[1].ToString();
        textBox10.Text = ma_deger[2].ToString();
    }
}

private void button18_Click(object sender, EventArgs e)
{
    try
    {
        string[] path = new string[2];
        MessageBox.Show("Karşılaştırma için 2 excel dosyası seçmelisiniz.");
        OpenFileDialog dlgOpen = new OpenFileDialog();
        dlgOpen.Filter = "veri excel(.xls)*.xls";

        string path1 = string.Empty;
        string path2 = string.Empty;

        if (dlgOpen.ShowDialog() == DialogResult.OK)
        {
            path1 = dlgOpen.FileName;
            path1.Contains(".xls");
        }
        if (path1.Contains(".xls"))
        {
            if (dlgOpen.ShowDialog() == DialogResult.OK)
            {
                path2 = dlgOpen.FileName;
            }
            if (path2.Contains(".xls"))
            {
                graf_comp(path1, 1);
                graf_comp(path2, 2);

                path[0] = path1;
                path[1] = path2;

                Grafik a = new Grafik(path);

                a.Show();
                a.BringToFront();
            }
            else MessageBox.Show("Hatalı Seçim, Tekrar Deneyin");
        }
        else MessageBox.Show("Hatalı Seçim, Tekrar Deneyin");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

    }
}

private void button21_Click(object sender, EventArgs e)
{
    try
    {
        string[] ma_deger = new string[10];
        char a;
        int kx = 0;

        OpenFileDialog dlgOpen = new OpenFileDialog();
        dlgOpen.Filter = "index(.dat)*.dat";
        int i = 0;
        int ix = 0;
        graf_indis = 0;

        if (dlgOpen.ShowDialog() == DialogResult.OK)
        {
            StreamReader re = File.OpenText(dlgOpen.FileName);
            string input = null;
            uzanti = re.ReadLine();
            if (uzanti == "excel")
            {
                big = int.Parse(re.ReadLine());
                graf_count = int.Parse(re.ReadLine());
                while ((input = re.ReadLine()) != null)
                {
                    dosya[i] = input;
                    i++;
                }
                int q = 0;
                q = dlgOpen.FileName.ToString().IndexOf("index.dat");
                string temp = dlgOpen.FileName.ToString().Substring(0, q);
                temp += "value.dat";
                StreamReader ret = File.OpenText(temp);
                string input2 = ret.ReadLine();
                while ((input2 = ret.ReadLine()) != null)
                {
                    if (input2.Length > 10)
                    {
                        kx = 0;
                        StringTokenizer st2 = new StringTokenizer(input2, " ");
                        //If an empty delimiter is given, automatically uses " " (space).
                        a = st2.nextToken()[0];
                        while (st2.hasMoreTokens())
                        {
                            ma_deger[kx] = st2.nextToken();
                            kx++;
                        }
                        eski_goster(a, ma_deger);
                        ix++;
                    }
                }
            }
            button12.Enabled = true;
        }
    }
}

```



## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```

        }
    }
}

pictureBox1.ImageLocation = mevcut_folder_2 + "/" + "graf" + graf_indis.ToString() +
".png";
}

private void progressBar1_Click(object sender, EventArgs e)
{
}

private void textBox3_TextChanged(object sender, EventArgs e)
{
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
}

private void label6_Click(object sender, EventArgs e)
{
}

private void checkBox10_CheckedChanged(object sender, EventArgs e)
{
}

private void comA_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void eni_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void baudA_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void button22_Click(object sender, EventArgs e)
{
    kontrolet();
    progressBar1.Value = 0;
    try
    {

```

## EK-22 (Devam) Bilgisayar kontrol yazılımının C# kodu

```
richTextBox1.Text = string.Empty;

serialPort1.Write("a");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + " Tekrar Deneyin");
}
}

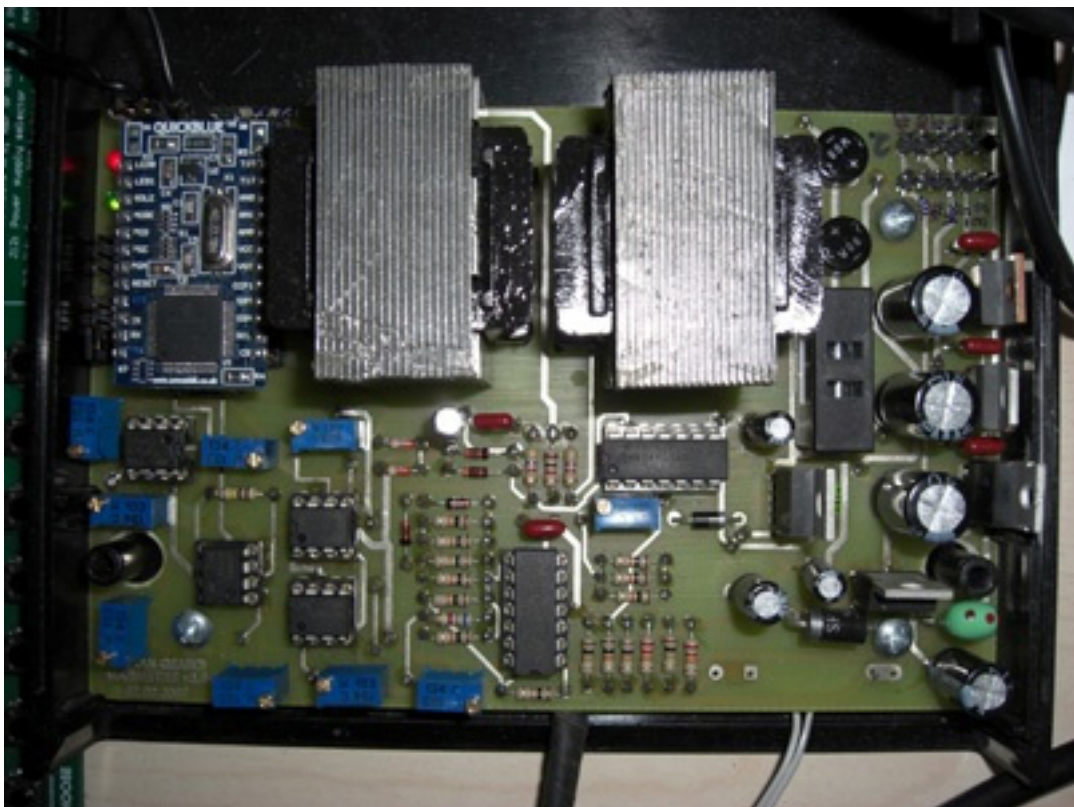
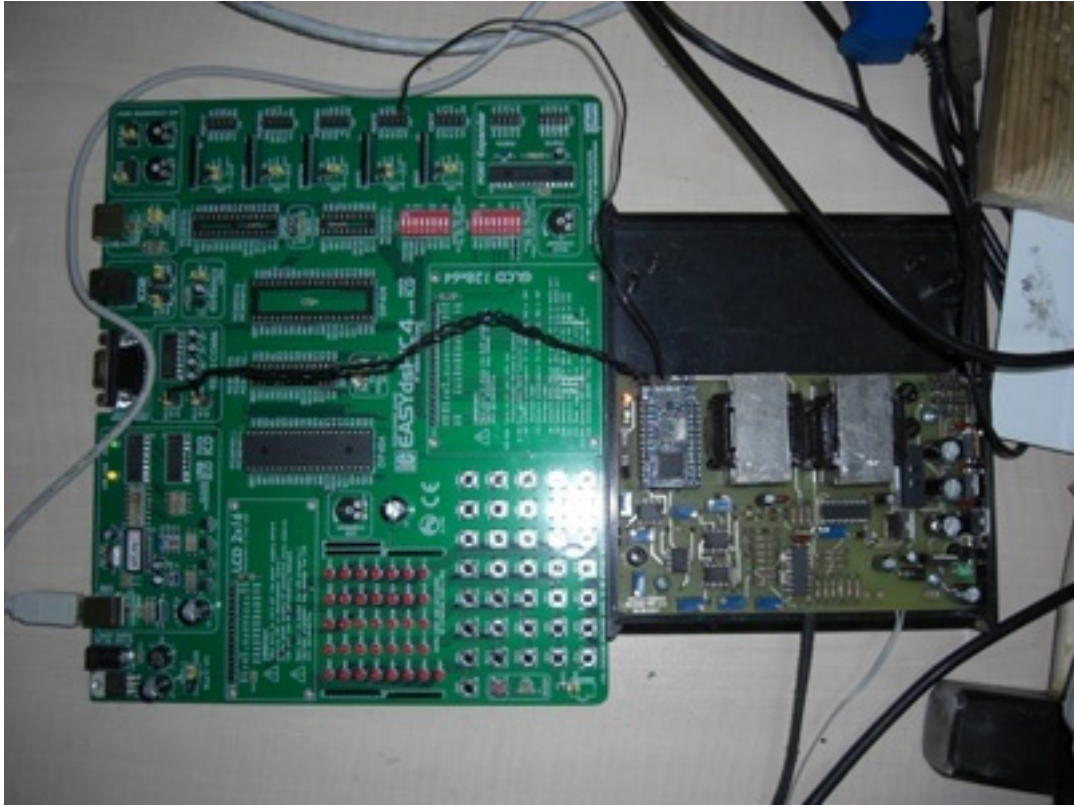
private void button23_Click(object sender, EventArgs e)
{
    kontrolet();
    progressBar1.Value = 0;
    try
    {
        richTextBox1.Text = string.Empty;

        serialPort1.Write("s");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + " Tekrar Deneyin");
    }
}

private void label1_Click(object sender, EventArgs e)
{
}

private void label2_Click(object sender, EventArgs e)
{
```

EK-23 mAs metre fotoğrafları



EK-23 (Devam) mAs metre fotoğrafları



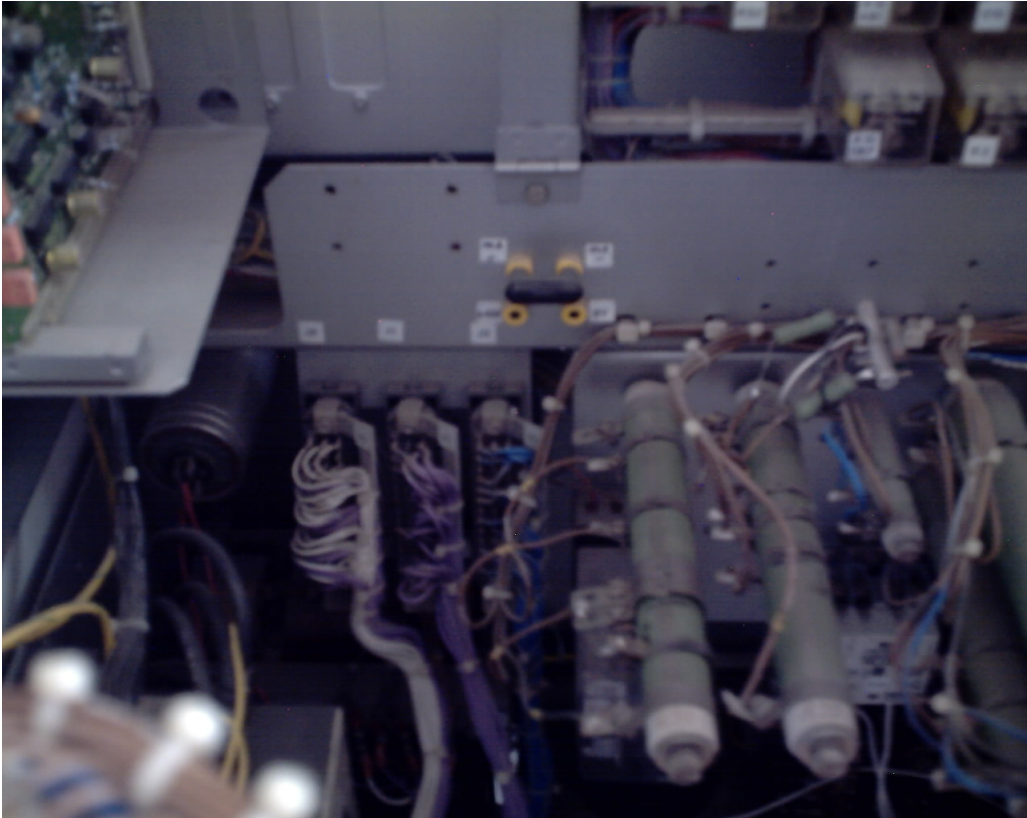
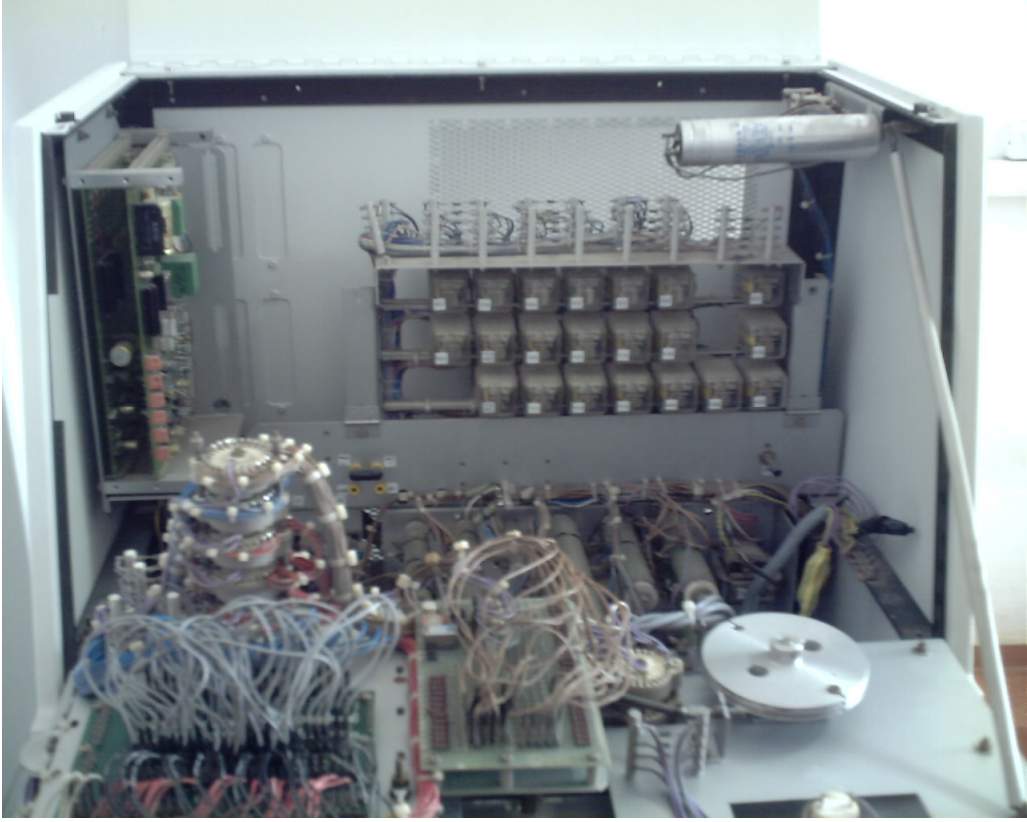
EK-24 Floroskopi cihazı üzerinde ölçüm alınırken



EK-25 Picker anatomic-40 x-ışını görüntüleme cihazı



EK-26 X-ışını cihazına bağlantı



EK-27 Konvansiyonel x-ışını görüntüleme cihazı üzerinde ölçüm alınırken



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : GEMİCİ, Hakan  
Uyruğu : T.C.  
Doğum tarihi ve yeri : 26.01.1980 Tire  
Medeni hâli : Bekâr  
Telefon : 0 (312) 361 94 42  
Faks : 0 (312) 230 65 03  
e-mail : hakan\_gemici@yahoo.co.uk

### Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi / Elektrik-Elektronik Müh. Bölümü	2008
Lisans	Selçuk Üniversitesi/ Elektrik-Elektronik Müh. Bölümü	2004
Lise	İzmir Selma Yiğitalp Lisesi	1997

### Yabancı Dil

İngilizce

### Hobiler

Motor Sporları, Yüzme, Atıcılık, Bilgisayar Teknolojileri