

**T.C.
YOZGAT BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM
DALI**

Yüksek Lisans Tezi

**YENİDOĞANLAR İÇİN KUVÖZ OTOMASYONU VE
TIBBİ TERMAL GÖRÜNTÜ KONTROL SİSTEMİ
TASARIMI**

Kadir Caner KILIÇARSLAN

**Tez Danışmanı
Dr. Öğretim Üyesi Mustafa YAZ**

Yozgat 2019

**T.C.
YOZGAT BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM
DALI**

Yüksek Lisans Tezi

**YENİDOĞANLAR İÇİN KUVÖZ OTOMASYONU VE
TIBBİ TERMAL GÖRÜNTÜ KONTROL SİSTEMİ
TASARIMI**

Kadir Caner KILIÇARSLAN

**Tez Danışmanı
Dr. Öğretim Üyesi Mustafa YAZ**

Yozgat 2019

T.C.
BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEZ ONAYI

Enstitümüzün Elektrik – Elektronik Mühendisliği An Bilim Dalı 70111514003 numaralı öğrencisi Kadir Caner KILIÇARSLAN' nın hazırladığı “ **Yeni Doğanlar için Kuvöz Otomasyonu ve Tıbbi Termal Görüntü Kontrol Sistemi Tasarımı**” başlıklı YÜKSEK LİSANS tezi ile ilgili TEZ SAVUNMA SINAVI, Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği uyarınca 03/09/2019 günü saat 13.00’ de yapılmış, tezin onayına OY ÇOKLUĞU / OY BİRLİĞİYLE karar verilmiştir.

Başkan : Dr. Öğr. Ü. Yavuz TÜRKAY

Üye : Dr. Öğr. Ü. Mustafa YAZ

Üye : Dr. Öğr. Ü. Muhammet Serdar BAŞÇIL

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulu'nun 10.../10.../2019.. tarih ve 48. sayılı kararı ile onaylanmıştır.

10.10.2019
Prof. Dr. Mustafa SAGMACI
Yozgat Bozok Üniversitesi
Fen Bilimleri Enstitüsü
Müdürü
(Ünvanı, Adı Soyadı)
Müdür

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR	v
TABLolar LİSTESİ	vi
ŞEKİLLER LİSTESİ	vii
KISALTMALAR LİSTESİ	xi
1. GİRİŞ	1
2. GENEL BİLGİLER	4
2.1. Kuvöz.....	4
2.1.1. Kuvözler Hakkında Genel Bilgiler	4
2.1.2. Kuvözler Çeşitleri.....	5
2.1.3. Kuvözlerde Bulunan Temel Kontrol Birimleri.....	6
2.2 Prematüre Vital (Yaşamsal) Parametreleri.....	6
2.2.1 Biyolojik İşaretler ve Oluşumu.....	7
2.2.1.1 Elektrik Kökenli Biyolojik İşaretler.....	8
2.2.1.2 Elektrik Kökenli Olmayan Biyolojik İşaretler.....	9
2.2.1.3 Biyolojik İşaretlerin Algılanması ve Anlamlı Hale Getirilmesi.....	9
2.2.2 Kalp ve Elektrokardiyografi (EKG).....	12
2.2.2.1 Kalbin Çalışması.....	13
2.2.2.2 Kalbin Elektriksel İletimi ve EKG Oluşumu.....	14
2.2.2.3 EKG Ölçümü.....	18
2.2.3 Oksijen Satürasyonu (SpO ₂) ve Pulse Oksimetre.....	19
2.2.3.1 Pulse Oksimetre Kullanım Endikasyonları.....	22
2.2.3.2 Pulse Oksimetre Kullanımının Kontrendikasyonları.....	22
2.2.3.3 Pulse Oksimetre Probenin Uygulandığı Yerler.....	23
2.2.4 Nabız.....	23
2.2.5 Vücut Sıcaklığı.....	25
2.3 Termal Görüntüleme.....	25
2.3.1 Termal Kameralar ve Çalışma Prensipleri.....	26

2.3.2 Medikal Termografi.....	28
2.4 Görüntü İşleme.....	30
2.5 Microsoft Visual C#.....	33
2.5.1 Microsoft Visual C# Özellikleri.....	34
2.5.2 Microsoft Visual C# ile Neler Yapılabilir.....	35
2.6 Aforge.Net.....	36
3. ORTAM VE VİTAL PARAMETRELERİN ÖLÇÜM VE KONTROL YÖNTEMİ.....	37
3.1 Sistemin Genel Çalışma Yapısı.....	37
3.2 Arduino.....	39
3.3 Kuvöz Ortam Parametreleri.....	39
3.3.1 Sıcaklık ve Nem Sensörü.....	39
3.3.2 Oksijen Konsantrasyonu Sensörü.....	41
3.4 Prematüre Vital Parametreleri.....	42
3.4.1 Nabız ve SpO2 Ölçümü.....	42
3.4.2 EKG Ölçümü.....	44
3.4.3 Vücut Sıcaklığı Ölçümü.....	46
3.5 Ortam ve Vital Parametreler Kullanıcı Ara Yüz Programı.....	46
4. PREMATÜRE TERMOGRAM ANALİZİ ARA YÜZÜ	52
4.1 Sistemin Genel Çalışma Yapısı.....	52
4.2 Programı Kullanma Adımları.....	54
5. BULGULAR.....	67
5.1 1. Çalışma.....	67
5.2 2. Çalışma.....	72
5.3 3. Çalışma.....	81
5.4 Vital ve Çevresel Parametre Bulguları.....	86
6. SONUÇ VE ÖNERİLER.....	89
KAYNAKLAR.....	92
EKLER.....	97
ÖZGEÇMİŞ.....	153

ÖZET

YENİDOĞANLAR İÇİN KUVÖZ OTOMASYONU VE TIBBİ TERMAL GÖRÜNTÜ KONTROL SİSTEMİ TASARIMI

Kadir Caner KILIÇARSLAN

**Yozgat Bozok Üniversitesi
Fen Bilimleri Enstitüsü
Elektik-Elektronik Mühendisliği Ana Bilim Dalı
Yüksek Lisans Tezi**

2019; Sayfa: 153

Tez Danışmanı: Dr. Öğretim Üyesi Mustafa YAZ

Bu çalışmada, klasik kuvöz kontrol sistemlerinde olan ortam parametrelerinin (sıcaklık, nem ve O₂ konsantrasyonu) ölçülmesi ve istenilen seviye getirilmesi işlevlerine ilave olarak prematürenin vital (yaşamsal) parametrelerinin de (vücut sıcaklığı, nabız, SpO₂, EKG) ölçülmesi ve izlenmesi sağlanmıştır. Bütün ölçüm ve kontrol sistemleri tek bir kullanıcı ara yüzüne toplanmıştır. Ayrıca bu ara yüz programında son yıllarda tıpta kullanımı yaygınlaşmaya başlayan termal görüntüleme yöntemiyle elde edilen prematüre termogramları üzerinde vücut ısı analizleri yapılmasını sağlayan bir görüntü işleme programı da tasarlanmıştır. Bu şekilde tek bir ara yüz programı ile bahsedilen tüm sistem ve fonksiyonların izlenmesi ve kontrol edilmesine olanak sağlayan bütünlük bir sistem oluşturulması ve sağlık personellerinin daha etkin ve verimli çalışması amaçlanmaktadır.

Anahtar Kelimeler: Kuvöz kontrol sistemi, vital parametreler ölçümü, görüntü işleme, termal görüntüleme

ABSTRACT

INCUBATOR AUTOMATION AND MEDICAL THERMAL IMAGE CONTROL SYSTEM DESIGN FOR NEWBORNS

Kadir Caner KILIÇARSLAN

**Yozgat Bozok University
Graduate School of Natural and Applied Sciences
Department of Electrical-Electronic Engineering
Master of Science Thesis**

2019; Pages: 153

Thesis Supervisor: Assist. Prof. Dr. Mustafa YAZ

In this study, it is provided to measure and monitor the vital parameters (body temperature, pulse, SpO₂, ECG) of premature in addition to the functions of measure and controlling the ambient parameters (temperature, humidity and O₂ concentration) in the classical incubator control systems. All measuring and control systems are integrated into a single user interface. In addition, in this interface program, an image processing program has been designed which enables body temperature analysis on premature thermograms obtained by thermal imaging method which has become widespread in medicine in recent years. In this way, it is aimed to provide healthcare personnel to work more effectively and efficiently by creating an integrated system that allows monitoring and control of these systems and functions with a single interface program.

Keywords: Incubator control system, measuring vital parameters, image processing, thermal imaging

TEŐEKKÜR

Tez alıőmam sırasında deneyimleri ve bilgi birikimiyle beni ynlendiren ve alıőmama katkıda bulunan tez danıőmanım ve deęerli hocam Bozok niversitesi Mhendislik Fakltesi Elektrik-Elektronik Mhendislięi Ana Bilim Dalı Baőkanı Dr. đretim yesi Mustafa YAZ' a en iten teőekkrlerimi sunarım.

Ayrıca tm tez alıőmam boyunca bana maddi ve manevi desteklerini esirgemeyen baőta eőim Buket KILIARSLAN olmak zere tm aileme ve varlıęıyla bana g veren, yeni dnyaya gelmiő oęlum Mehmet Kerem' e teőekkr bor bilirim.

TABLolar LİSTESİ

	<u>Sayfa</u>
Tablo 2.1: Yaşa Göre Oksijen Satürasyon Değerleri.....	20
Tablo 2.2: Sağlıklı 2000 Gr Altında Prematürelere KB Değerleri.....	24
Tablo 2.3: Nabız Atım Sayıları.....	24
Tablo 2.4: Termal Kamera Kullanım Alanları.....	27
Tablo 2.5: RGB Renk Uzayı Desimal ve HEX Kod Bilgileri.....	31
Tablo 3.1: Ad8232 Ölçüm Kablosu Pinleri ile Vücut Elektrot Bağlantısı	45
Tablo 4.1: Örnek Renk Tonları ve RGB Değerleri	52
Tablo 4.2: Örnek Gri Tonları ve Desimal Değerleri.....	53

ŞEKİLLER LİSTESİ

	<u>Sayfa</u>
Şekil 2.1: Kuvöz.....	4
Şekil 2.2: İnsan-Enstrümantasyon Sistemi Blok Diyagramı	7
Şekil 2.3: Bazı Biyolojik İşaretler.....	8
Şekil 2.4: EKG İşaretlerinin Algılanması.....	8
Şekil 2.5: Yüzey Elektrot Çeşitleri.....	10
Şekil 2.6: Biyolojik İşaretleri Ölçümü Genel Blok Yapısı.....	11
Şekil 2.7: Kalbin Önden Görünüşü.....	12
Şekil 2.8: Kalbin Elektriksel İletimi Depolarizasyon ve Repolarizasyon.....	14
Şekil 2.9: Kalp Kasında Aksiyon Potansiyeli.....	15
Şekil 2.10: Kalpte Elektriksel Uyarı Akışı.....	15
Şekil 2.11: Örnek EKG İşareti	16
Şekil 2.12: Kalbin Fiziksel Hareketi Sonucu Oluşan Elektriksel Sinyaller	17
Şekil 2.13: Rutin Olmayan EKG Grafiği	17
Şekil 2.14: Yenidoğan EKG Görüntüsü	18
Şekil 2.15: Kardiyak Vektörünün İz Düşürüldüğü Eksenlerin Bulunduğu Düzlemler.....	18
Şekil 2.16: Frontal Düzlemde Einthoven Üçgeni	19
Şekil 2.17: Hb Ve Hbo2 Işık Dalga Boyu Eğrileri	21
Şekil 2.18: Pulse Oksimetre Ölçüm Mekanizması	21
Şekil 2.19: Pulse Oksimetre Çeşitleri	23
Şekil 2.20: Elektromanyetik Spekturum	26
Şekil 2.21: Termal Kamera Genel Yapısı	27
Şekil 2.22: Resim Piksel Matrisi	30
Şekil 2.23: Görüntü İşleme Sistem Yapısı.....	32
Şekil 2.24: Microsoft Visual C# Tasarım ve Form görüntüsü	34
Şekil 3.1: Kuvöz Kontrol Kartı Baskı Devre Şeması.....	38
Şekil 3.2: Kuvöz Kontrol Devre Kartı	38
Şekil 3.3: Dht22 Sensörü.....	40

Şekil 3.4:	Arduino Nano Kartı ve Dht22 Bağlantısı	40
Şekil 3.5:	itg/M-04 Medikal Oksijen Sensörü	41
Şekil 3.6:	Arduino Nano Kartı ve Itg/M-04 Sensörü Bağlantısı	42
Şekil 3.7:	Max30100 Entegresi	43
Şekil 3.8:	Max30100 Entegresi ile Parmak Ucundan Ölçüm Yapılması	43
Şekil 3.9:	Arduino Nano kartı ile Max30100 Entegresi Bağlantı Şeması.....	44
Şekil 3.10:	Ad8232 Entegresi.....	44
Şekil 3.11:	Ad8232 ve Vücut-Elektrot Bağlantısı	45
Şekil 3.12:	Arduino Nano Kartı ile Ad8232 Bağlantısı	46
Şekil 3.13:	Ortam ve Vital Parametreleri Programının Akış Diyagramı	47
Şekil 3.14:	Ara Yüz Programında Ortam ve Vital Parametrelerin Görütülenip Kontrol Edildiği Ekran.....	48
Şekil 3.15:	Ortam ve Vital Parametreleri Formu Ekran Görüntüsü	48
Şekil 3.16:	Ara Yüz Programının Arduino Kartından Ölçüm Değerleri Almasını Sağlayan Algoritma.....	49
Şekil 3.17:	Ortam Parametreleri Sınır Değer Aralıklarının Girildiği Form Ekran Görüntüsü.....	49
Şekil 3.18:	Ara Yüz Programının Görsel Uyarı Vermesi Ekran Görüntüsü.....	50
Şekil 3.19:	Ortam ve Vital Parametreler Kullanıcı Ara Yüzü Ekran Görüntüsü.....	51
Şekil 4.1:	Tasarlanan Programdaki Sıcaklık-Renk Ölçeği	53
Şekil 4.2:	Termogram Analizi Ara Yüz Programı Akış Diyagramı.....	54
Şekil 4.3:	Termaogram Analizi Ara Yüzü Açılış Ekran Görüntüsü.....	54
Şekil 4.4:	Termogram Analizi Ara Yüze Ana Resim Yüklenmesi	55
Şekil 4.5:	Renk Geçişleri ve Renk Bilgisi Desimal Değerleri.....	56
Şekil 4.6:	TrackBar Nesnesi ile RGB Renk Bilgisini Eşleştiren Algoritma.....	57
Şekil 4.7:	Ana Resim RGB Filtreleme Ekran Görüntüsü.....	58
Şekil 4.8:	RGB Filtreleme İşlemi Algoritması.....	59
Şekil 4.9:	Ana Resim Gri Format Dönüşümü Ekran Görüntüsü.....	60
Şekil 4.10:	Gri Formata Dönüştürme Algoritması.....	60
Şekil 4.11:	TrackBar2 Nesnesi ile Gri Format Renk Bilgisini Eşleştiren Algoritma.....	61
Şekil 4.12:	Gri Format Filtreleme İşlemi Ekran Görüntüsü.....	62
Şekil 4.13:	Gri Format Filtreleme İşlemi Algoritması.....	63

Şekil 4.14:	Blob Counter İşlemi Ekran Görüntüsü.....	64
Şekil 4.15:	BlobCounter İşlemi Algoritmaları.....	65
Şekil 4.16:	Seçilmiş Alanlar Bölümüne Resim Yükleme.....	66
Şekil 5.1	1. Çalışmaya ait Ana Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	68
Şekil 5.2	35,4°C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	69
Şekil 5.3	1. Çalışmaya ait Ana Görüntü, Gri Formata Çevrilmiş Görüntü ve Gri Format Resim Renk Dağılımı Grafiği.....	70
Şekil 5.4	35,4°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	71
Şekil 5.5	2.Çalışmaya ait Ana Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	73
Şekil 5.6	34,8 °C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	74
Şekil 5.7	33,2 °C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	75
Şekil 5.8	32,3 °C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	76
Şekil 5.9	2. Çalışmaya ait Ana Görüntü, Gri Formata Çevrilmiş Görüntü ve Gri Format Resim Renk Dağılımı Grafiği.....	77
Şekil 5.10	34,8°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	78
Şekil 5.11	33,2°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	79
Şekil 5.12	32,3°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	80
Şekil 5.13	3. Çalışmaya ait Ana Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	82
Şekil 5.14	31,6°C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	83
Şekil 5.15	3. Çalışmaya ait Ana Görüntü, Gri Formata Çevrilmiş Görüntü ve Gri Format Resim Renk Dağılımı Grafiği.....	84
Şekil 5.16	31,6 °C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik.....	85

Şekil 5.17	Vital Parametre Bulguları.....	86
Şekil 5.18	Ortam ve Vital Parametreler Ekranı Görsel Uyarılar Ekran Görüntüsü.....	87
Şekil 5.19	Arduino Nano Kartından Verilen Komutlar.....	88



KISALTMALAR LİSTESİ

A	: Amper
ADC	: Analog-Dijital Çevirici
Ag	: Gümüş Element Sembolü
ALC	: Ortam Işığı Kesici (Ambient Light Cancellation)
AV	: Atrioventriküler Düğüm
Ca	: Kalsiyum
dB	: Desibel
DC	: Doğru Akım
DEC	: Onluk Sayı Sistemi
DLL	: Dinamik Bağlantı Kitaplığı (Dynamic Link Library)
Dr.	: Doktor
EEPROM	: Küçük boyuttaki verileri kalıcı olarak saklamak için bilgisayar ya da diğer cihazlarda kullanılan bir yongadır (Electrically Erasable Programmable Read-Only Memory)
EKG	: Elektrokardiyografi
FIFO	: İlk giren ilk çıkar (First in First Out)
Gr	: Gram
GUI	: Grafiksel Kullanıcı Arayüzü (Graphical User Interface)
Hb	: Hemoglobin
HbO₂	: Oksihemoglobin
HEX	: Onaltılık Sayı Sistemi
Hg	: Civa
http	: Hiper Metin Transfer Protokolü (Hypet Text Transfer Protocol)
Hz	: Hertz
I² C	: Senkron Haberleşme
ICSP	: Devre Üzerinden Programlama (In-circuit serial programming)

IDE	: Tümüleşik Geliřtirme Ortamı (Integrated Development Environment)
IR	: Kızılötesi
K	: Potasyum
KB	: Kilo Bayt
LED	: Iřık Yayan Diyot (Light Emitting Diode)
M	: Mega
mA	: Mili Amper
MCU	: Mikro Denetleyici
MHz	: Mega Hertz
mm	: Milimetre
mV	: Mili Volt
Na	: Sodyum
nF	: Nano Farad
Ni	: Nikel Element Sembolü
O₂	: Oksijen Gazı
°C	: Santigrad Derece
°K	: Kelvin Derece
PaO₂	: Arter Kan Gazı Örneğindeki Kısmi Oksijen Basıncı
pH	: Bir çözeltinin asitlik veya bazlık derecesini tarif eden ölçü birimidir (Power of Hydrogen)
PWM	: Darbe Geniřliğı Modülasyonu (Pulse Width Modulation)
RH	: Bağıl Nem
RLD	: Sağ Bacak Sürücüsü
SA	: Sinoatrial Düğüm
SCL	: Seri Clock Hattı
SDA	: Seri Data Hattı
Sn	: Saniye
SpO₂	: Oksijen Satürasyonu Seviyesi
SRAM	: Durağan Rastgele Eriřimli Bellek (Static Random Access Memory)
us	: Miro saniye

V	: Volt
V_{cc}	: Kollektör Besleme Voltajı (Collector Supply Voltage)
V_{dd}	: Drain Besleme Voltajı (Drain Supply Voltage)
WPF	: .NET Framework 3.0'ın grafik altyapısını oluşturan XAML tabanlı sistemdir (Windows Presentation Foundation)
XML	: Genişletilebilir İşaretleme Dili
.NET	: Microsoft'un Windows işletim sistemi üzerinde çalışabilecek yazılımlar üretilebilmesi amacıyla geliştirdiği yazılım geliştirme ortamı ve framework'üdür
μ	: Mikro
μm	: Mikro metre
μV	: Mikro Volt

1.GİRİŞ

1891 yılında Dr. Alexandre LİON tarafından Fransa'da geliştirilen ilk bebek kuvözünün ardından, kuvözler uzun yıllardır erken ya da sorunlu olarak dünyaya gelen bebeklerin (prematüre) hayata tutunmasına yardımcı olmaktadır.

Dünyada ülkelere göre değişmekle beraber tüm gebeliklerin yaklaşık %10' nu 2500 gr ve altında doğan düşük doğum ağırlıklı (DDA) bebekler oluşturur. Düşük doğum ağırlıklı yeni doğanların ise % 70-75'i prematüre doğumlardır [1].

Doğum eyleminin 37. gebelik haftası tamamlanmadan başlamasına erken doğum tehdidi, eylemin bebeğin doğumuyla tamamlanmasına preterm doğum ya da erken doğum adı verilir. Zamanından önce doğan bebek prematüre (olgunlaşmamış) olarak adlandırılır [1].

Prematüre bebekler, sağlıklı bebeklere göre farklı fizyolojik ve anatomik bir yapıya sahiptir. Prematürelerin normal vücut ısılarını sağlama kabiliyetleri azdır ve birçok tıbbi sorunla karşı karşıya kalabilmektedirler. Bu yüzden özel bir bakım ve tedaviye ihtiyaç duyarlar.

Kuvözler, prematürelerin hayatlarına devam edebilmeleri için ısıtma, havalandırma, nemlendirme koşullarını optimum seviyede tutarak, şeffaf ve steril bir ortamda korunmasını sağlar.

Günümüzde, özellikle medikal teknolojinin hızla gelişmesiyle bebek kuvözlerin yapısı ve özellikleri de hızla değişmekte ve gelişmektedir. Kuvözler, prematüreler için en ideal ortam gereksinimlerini oldukça hassas düzeylerde gerçekleştirebilmektedirler. Bu sayede prematürelerin yeni ortama adaptasyonu sağlanmakta, tıbbi sorunları tedavi edilebilmekte, enfeksiyon vb. problemler engellenebilmektedir. Kuvözler sayesinde vücut direnci yükselen prematüreler hayata tutunabilmektedir.

Genellikle sağlık sorunlarıyla dünyaya gelen prematürelerin vital (yaşamsal) parametrelerinin de ölçülmesi ve izlenmesi gerekmektedir. Vital bulgulardan olan

vücut sıcaklığı, nabız, kan basıncı, EKG, SpO₂ prematürenin takibi için çok önemlidir. Prematürenin fizyolojik işlevleri hakkında bilgi verir. Fizyolojik işlevlerindeki bozukluk vital parametrelerin monitörizasyonu ile tespit edilebilir. Bu da prematürenin gözlenmesi ve hastalıklarının teşhis ve tedavi edilmesinde önemli rol oynar.

Vital değerler, prematürenin vücut yüzeyine yerleştirilen sensör ve elektrotlar vasıtasıyla, kuvöz dışında harici aygıtlar olan hasta başı monitörler ile ölçülür ve görüntülenir. Bununla birlikte prematüreye ilaç veya solüsyon vermeye yarayan enjektomatlar ya da infzyon pompaları, sarılık tedavisinde kullanılan fototerapi cihazları, solunum desteği veren ventilatörler gibi birçok cihaz ve sistem prematürenin hastalık teşhis ve tedavi işlemlerinde kuvöz dışında kullanılır. Bütün bunlar göz önüne alındığında görevli sağlık personelleri birçok sistem ve cihazı ayrı ayrı izlemek ve takip etmek zorunda kalmaktadır

Tasarlanan kuvöz kontrol modelinde özellikle Yeni Doğan Yoğun Bakım Ünitelerinde sıklıkla kullanılan kuvözlerde, prematürelere için uygun çevre koşullarının kontrol edilmesine ilave olarak fizyolojik ve biyolojik sinyaller de (EKG, SpO₂, nabız ve vücut ısısı) izlenebilmektedir.

Bütün ölçüm ve kontrol sistemi Microsoft Visual Studio ortamındaki C# programında tasarlanan kullanıcı ara yüz programına entegre edilmiştir.

Tasarlanan bu modelde mikro denetleyici kartı, kuvözün ortam parametrelerini ve prematürenin vital değerlerini sensörler vasıtasıyla ölçerek kullanıcı ara yüzüne göndermek ve ara yüzden gelen komutları değerlendirerek ilgili birimi (fan, ısıtıcı vb.) harekete geçirmek üzere programlanmıştır.

Ayrıca tasarlanan programda son yıllarda tıpta kullanımı yaygınlaşmaya başlayan termal görüntüleme metodu ile elde edilen prematüre termogramları üzerinde vücut ısısı analizleri yapan bir ara yüz de tasarlanmıştır. C# programında, Aforge.Net kütüphanesi kullanılarak tasarlanan görüntü işleme programı ile prematürelere termal kamera görüntüleri üzerinde renk filtreleme teknikleri kullanılarak, vücut ısı değişimlerinin analizi yapılabilmektedir. Bu sayede vücut iç kanama, dolaşım sistemi

ve hastalıkları, nöromuskuler sistem ve hastalıkları, kas, eklem ve iskelet sistemi hastalıkları vb. rahatsızlıklar için ön teşhis konularak, daha erken yapılacak radyolojik ve ultrasonografik tetkiklerle daha hızlı teşhis ve tedavi işlemlerinin yapılabileceği düşünülmektedir.

Bu çalışmanın amacı bahsedilen tüm sistemler ile bu sistemlerin kontrol ve monitörizasyonunu sağlayan kolay ve kullanışlı bir kullanıcı ara yüzü tasarlayarak, görevli doktor ve hemşirelerin daha rahat ve verimli çalışmasını sağlamaktır.



2. GENEL BİLGİLER

2.1.Kuvöz

2.1.1 Kuvözler Hakkında Genel Bilgiler

Erken ve/veya sağlık sorunları ile doğan bebekler, vücut sıcaklıklarını gerekli düzeyde ve kararlı tutamazlar. Vücut savunma sistemleri yeterli düzeyde olmadığından steril bir ortamda sürekli bakım ve tedaviye ihtiyaç duyarlar. Kuvözler, prematürelere sağlık koşullarının giderek kötüleşmemesi amacıyla sabit sıcaklık, belli bir oranda nem ve anti bakteriyel filtreden geçirilerek arındırılmış taze hava sağlayarak, uykularının bölünmeyeceği sessiz ve steril bir ortam oluştururlar [2].

Bebek kuvözleri genel olarak prematürelere izlenmesi için gözlem pencereleri olan şeffaf kabin bölümü, ısıtma, nemlendirme ve havalandırma faaliyetlerini yürüten mikro denetleyici tabanlı kontrol birimi ve gövde bölümünden oluşur. Ayrıca kuvözlerde prematürelere ağırlığını ölçen terazi, prematürelere röntgen filmi çekmek için kızaklı yapıda röntgen kasedi yerleştirme bölümü, O₂ tüpü yerleştirme bölümü, yükseklik ayarı yapmaya veya prematureyi hareket ettirmeye yarayan mekanik yapıda düzenekler de bulunmalıdır [2]. Şekil 2.1’ de klasik bir kuvöz ve bazı bölümleri görülmektedir.



Şekil 2.1 Kuvöz

Kuvözlerde genel olarak;

- Solunum sorunlarıyla doğan bebeklerin bakımı, takibi ve tedavisi
- Şeker hastalığı, böbrek yetmezliği, kan uyuşmazlığı, yüksek tansiyon gibi sağlık sorunları olan annelerin riskli gebeliklerinden dünyaya gelen bebeklerin bakımı, takibi ve tedavisi
- Doğum esnasında oksijensiz kalan ya da yeterli oksijen alamayan (asfiktik) bebeklerin takip ve tedavisi
- Prematüre olarak doğan ve henüz akciğerleri yeterli düzeyde olgunlaşmamış bebeklerin bakımı, takibi, başta solunum desteği olmak üzere her türlü tedavi işlemleri
- Enfeksiyonlu bebeklerin takip ve tedavisi
- Genç yaş ya da ileri yaş gebeliklerden doğan bebeklerinin takip ve tedavisi
- Sarılık rahatsızlığı olan bebeklerin takibi, fototerapi ve kan değişimi ihtiyaçları olan prematürelerin tedavi işlemleri yapılır [2].

2.1.2 Kuvöz Çeşitleri

Kuvözler kullanım amaçları ya da fonksiyonlarına göre sınıflandırılabilir. Hastanelerde sabit kuvözler kullanılırken, prematüre nakil işlemlerinde ambulans, helikopter vb. araçlarla taşıma işlemleri için transport kuvözler kullanılır. Ayrıca fonksiyonlarına göre koruma, bakım, izleme ve tedavi kuvözleri olarak çeşitlere ayrılmaktadır [2].

- **Koruma:** Çok fonksiyonlu değildir. Prematüreyi dış etkenlerden korumak için steril bir ortam sağlar.
- **Bakım:** Prematürelerin bakım ihtiyaçları ve gelişimleri sağlar.
- **İzleme:** Prematürelerin vital (hayati) parametreleri (nabız, vücut ısısı, solunum, kan basıncı gibi) kuvöze entegre edilmiş harici cihaz ve aygıtlarla

ölçülerek, kullanıcılara sürekli izleme, kaydetme ve gerekli hallerde müdahale etme imkanı sağlar.

- **Tedavi:** Genel olarak bu kuvözlerde sarılıklı bebeklerin fototerapi tedavileri yapılır ve bebeklere solunum desteği verilir [2].

2.1.3 Kuvözde Bulunan Temel Kontrol Birimleri

Kuvöz kontrol birimleri, kuvözdeki ısı, nem ve oksijen miktarını sensörler vasıtasıyla ölçerek; fan, ısıtıcı, nemlendirici ve O₂ kaynağına komut göndererek ortam parametrelerini istenilen düzeye getirmek ve sabit tutmak işlevlerini yerine getirmek üzere tasarlanırlar. Kuvöz kontrol birimi, kabin içi hava sirkülasyonunu sağlar ve kontrol eder. Bununla birlikte kuvöz kontrol birimi bebeğin cilt sıcaklığını da ölçerek, prematürenin vücut sıcaklığını kararlı tutulmasını sağlamak için elektrikli ısıtıcıyı da kumanda eder.

Ayrıca kuvöz kontrol birimleri kuvözün uygun şekilde çalışmadığı ve olumsuz olayların oluştuğu durumlarda kullanıcılara alarm ve uyarılar vermek üzere tasarlanmalıdır.

Üst düzey kuvöz kontrol birimleri yukarıda bahsedilen temel işlev ve görevleri yerine getirmenin yanında veri toplama, kayıt ve raporlama işlemlerini de yapabilmektedir.

2.2 Prematüre Vital (Yaşamsal) Parametreleri

Vital bulgular vücudun fizyolojik faaliyetleri hakkında bilgi verir. Eğer fizyolojik faaliyetlerde bir olumsuzluk olursa, bu vital bulguların değerlerine yansır. Vital parametreler; vücut ısısı, nabız, oksijen saturasyonu, EKG ve kan basıncı vücudu etkileyen anormal durumları gösterdiğinden hastanın izlenmesi, hastalıkların teşhis edilmesi ve tedavi işlemleri için büyük önem taşır [3].

Vital parametreler; canlı vücudundaki biyolojik işaretlerin alınması, filtrelenmesi, büyütülmesi, işlenmesi sonucu anlamlı hale getirilir.

2.2.1 Biyolojik İşaretler ve Oluşumu

Biyolojik işaret, canlı vücudundan elektrotlar veya dönüştürücüler aracılığıyla algılanan, elektrik kökenli olan veya elektrik kökenli olmayan işaretlerdir [4]. Biyolojik işaretler, insan vücudundaki sinir sistemi, beyin, kalp ve kas gibi çeşitli organların fizyolojik faaliyetleri sırasında oluşurlar. Elektriksel biyolojik işaretler hücrelerde meydana gelen elektrokimyasal olaylar temeline dayanır. Elektriksel olmayan biyolojik işaretler de yine vücuttaki organların fizyolojik faaliyetleri sonucu meydana gelen fiziksel ve kimyasal değişimler sonucu oluşurlar. Biyolojik işaretler oldukça zayıf ve karmaşık yapıda oldukları için tek başlarına bir anlam ifade etmezler. Bu işaretlerin algılayıcılar, dönüştürücüler, filtreler ve kuvvetlendiriciler vasıtasıyla bir takım işlemlerden geçirilerek anlaşılabilir hale getirilmesi gerekmektedir. Şekil 2.2' de insan vücudundan algılanan sinyaller ve bu sinyalleri anlamlı hale getiren cihaz ve sistemlerin (İnsan-Enstrumantasyon Sistemi) blok diyagramı gösterilmektedir. [4]



Şekil 2.2 İnsan-Enstrumantasyon Sistemi Blok Diyagramı [5]

Anlamlı hale getirilerek görüntülenen bazı elektriksel kökenli ve elektriksel kökenli olmayan biyolojik işaretler Şekil 2.3' de görülmektedir. Vücudun fizyolojik faaliyetlerinde meydana gelen olumsuzluklar bu işaretlerin değer ve şekline yansyarak, olağan dışında sinyaller oluşumuna sebep olurlar. Bu sayede birçok hastalık tespit edilebilir, hastanın takibi ve tedavi işlemleri yürütülebilir.

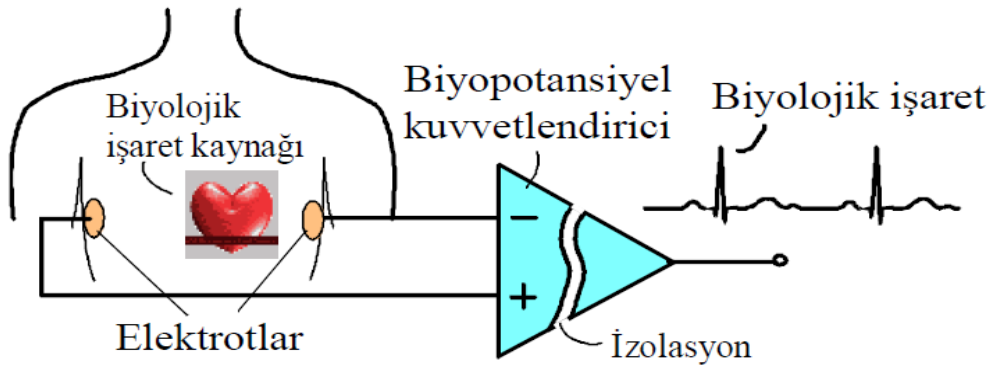


Şekil 2.3 Bazı Biyolojik İşaretler [4]

2.2.1.1 Elektrik Kökenli Biyolojik İşaretlerin Özellikleri

- Elektrotlar aracılığıyla canlı vücutundan algılanırlar, yalıtım önemlidir.
- Genlikleri küçüktür. ($100 \mu\text{V} \sim 1 \text{ mV}$)
- Spektrumu alçak frekanslar bölgesindedir. ($0,1 \text{ Hz} \sim 2000 \text{ Hz}$)
- Fark işareti şeklinde bulunurlar.
- Gürültülü işaretlerdir. Temel gürültü kaynakları; ortak mod şeklindeki 50/60 Hz'lik şebeke gürültüleri, fark işaret şeklinde bulunan diğer biyolojik işaret kaynakları ve elektronik eleman gürültüleridir [5].

Şekil 2.4' de elektriksel işaretlerden olan EKG' nin, insan vücutundan tespit edilerek görüntülenmesi işleminin basit bir şeması görülmektedir.



Şekil 2.4 EKG İşaretlerinin Algılanması [5]

2.2.1.2 Elektrik Kökenli Olmayan Biyolojik İşaretler

Elektrik kökenli olmayan işaretlere; basınç dönüştürücüleri aracılığıyla kan basınçlarının ölçümü, fotometrik dönüştürücüler vasıtasıyla nabız ve kandaki oksijen saturasyonu ölçme, ultrasonik ve elektromanyetik ile kan akış hızını ölçme, solunum hacmini ölçme, vücudun veya organların sıcaklık dönüştürücüleri, pH metre ölçümleri, kalp sesleri dönüştürücüleri, kanın veya havanın kimyasal dönüştürücüleri örnek verilebilir [6].

2.2.1.3 Biyolojik İşaretlerin Algılanması ve Anamlı Hale Getirilmesi

İnsan vücudundaki hücreler çeşitli dış ve iç etmenler sonucu, bir takım faaliyetleri yürütmek amacıyla uyarılır. Hücre zarının iç ve dış bölgeleri arasında belirli bir düzeyde elektriksel potansiyel farkı mevcuttur. Uyarılan hücrelerde kısa süreli ve anlık olarak bu potansiyel fark değişime uğrar. Hücrelerde meydana gelen bu potansiyel fark değişim olayına aksiyon potansiyeli adı verilir. Biyoelektriksel işaretler de hücrelerde meydana gelen aksiyon potansiyelleri sonucu oluşurlar.

Elektriksel biyolojik işaretler vücuda yerleştirilen elektrotlar vasıtasıyla algılanırlar. İnsan vücudunda ölçülmek istenen sistem ve aktiviteye göre vücudun çeşitli bölgelerine en az iki elektrot yerleştirilir ve elektrotların yerleştirildiği noktalardaki elektrik potansiyelleri ölçülür. Elektrot yüzeyindeki iletken maddeler, (gümüş, platin, altın, tantal, tungsten vb.) vücutta oluşan biyoeletrik sinyalleri ilgili cihaz ya da sisteme gönderir. Elektrotlar vücuda yerleştirilme şekline, bileşimindeki iletkenlere, bağlantı elemanlarına ve mekanik dayanımlara göre farklılık göstermektedir. Yüzeysel elektrot çeşitleri aşağıda tanıtılmış ve Şekil 2.5' de gösterilmiştir.

a) Metal plaka elektrot: Ni-Ag (Ag-AgCl) alaşımı kullanılır. Deri ile arasına pasta (jel) sürülür. Özel lastik veya kayışla tutturulur. EKG, EMG ve EEG sinyallerinin tespitinde kullanılır. Yüzeysel alanı büyük, bu nedenle empedansı ise küçüktür.

b) Gezici tipten elektrot: Elektrot şapka muhafazanın içinde bulunur. Şapkanın içinde elektrolit jel bulunur ve yapışkan bantla tutturulur. Şapka hareket

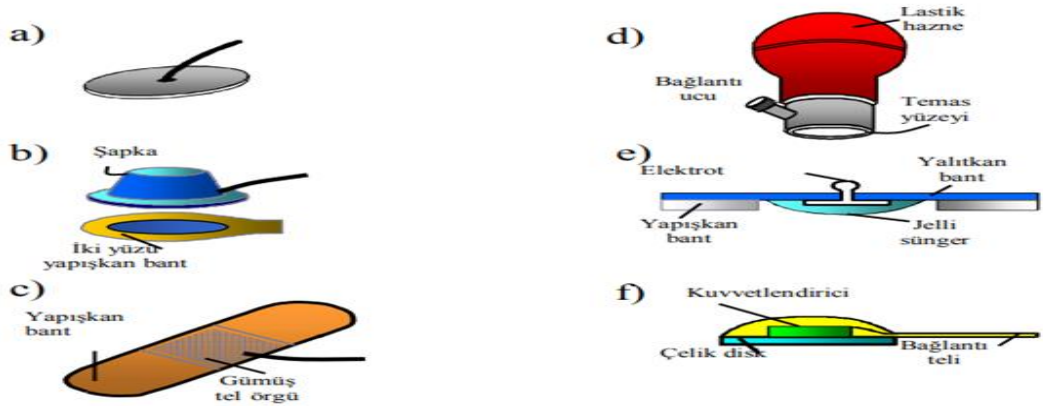
etse dahi elektrot, elektrolite göre hareket edemez. Bu tip elektrotlar kullanıldığında siyallerdeki gürültüler az olur. Ag-AgCl alaşımlar kullanılır. Tek kullanımlıktır.

c) Bükülebilir elektrot: Bir yüzü yapışkan bant şeklindedir ve yapıştırılacak vücut yüzeyinin şekline göre bükülebilir. Küçük çocuklarda EKG çekimleri için kullanılır. Ag-AgCl alaşımlar kullanılır.

d) Emici düzenli elektrot: Temas silindirik boruyla olur. Diğer uca vakum pompası vardır. Herhangi bir yapışkan veya bağlama bandı gerektirmez. Genellikle EKG işaretlerinin tespiti için kullanılan göğüs elektrodudur. Yüzey küçük, empedansı ise büyüktür.

e) Tümüyle atılır elektrot: EKG sinaylleri tespiti için göğüs elektrodudur. Ag-AgCl alaşımları kullanılır. Elektrot tabanına yapışık jel emdirilmiş süngeri vardır. Tek kullanımlıktır.

f) Kuru elektrot: Pasta (jel) gerektirmeksizin deri üzerine direkt uygulanır. Bu nedenle kapasitif kuplaj oluşur. Üst deri, elektrot ile iletkenliği daha iyi olan dermis arasında yalıtkan durumundadır ve bu yüzden burada bir kapasite oluşur ki dermis ile elektrod bu kapasitenin plakaları, dış deri ise dielektrik malzeme durumundadır. Bağlandığı kuvvetlendiricinin giriş empedansı büyük olmalıdır. Bu elektrotlarda gürültü problemi fazla olduğu için, elektrot kuvvetlendiricinin yakınına kurulus (izleyici) ve gürültü azaltılır. Si teknolojisi kullanılır. Tabanda SiO₂ yalıtkanı bulunur [4].



Şekil 2.5 Yüze Elektrot Çeşitleri [4]

Biyoelektrik sinyaller, ölçüm düzenekleri tarafından, vücut yüzeyinden yukarıda tanımlanan elektrotlar vasıtasıyla algılanırlar. Ancak Bölüm 2.2.1.1’ de de bahsedildiği üzere elde edilen bu biyoelektrik sinyallerin genlikleri küçük, spektrumları alçak frekans bölgesinde ve oldukça gürültülü işaretlerdir. Bu şekilde bir anlam ifade etmeyen biyoelektrik işaretlerin birtakım işlemlerden geçirilerek yorum ve analize uygun veriler haline dönüştürülmesi gerekmektedir. Biyolojik işaretler üzerinde filtreleme işlemleri yapılarak gürültüden arındırılması, biyopotansiyel kuvvetlendiriciler kullanarak genliklerinin büyütülmesi, alçak geçiren süzgeçler kullanarak istenilen sinyallerin diğerlerinden ayrıştırılması işaretleri anlamlı hale getiren temel işlemler olarak değerlendirilebilir. Bu şekilde işlenerek görüntülenen veriler hastalıkların teşhis ve tedavi işlemlerinde önemli rol oynamaktadır.

Elektriksel olmayan biyolojik işaretlerde algılayıcılar ve dönüştürücüler vasıtasıyla tespit edilirler. Algılayıcı olarak da adlandırılan giriş dönüştürücüsünde, ışık şiddeti, yer değişimi, sıcaklık, manyetik alan veya pH değerleri gibi fiziksel veya kimyasal birimler elektrik veya elektronik sinyallerine dönüştürülür. Yani bir giriş dönüştürücüsü de elektriksel olmayan sinyal biçiminden, elektriksel biçime dönüştürülür [7]. Şekil 2.6’ da her iki türdeki biyolojik işaretlerin genel olarak ölçüm blok yapısı görülmektedir.



Şekil 2.6 Biyolojik İşaretleri Ölçümü Genel Blok Yapısı [4]

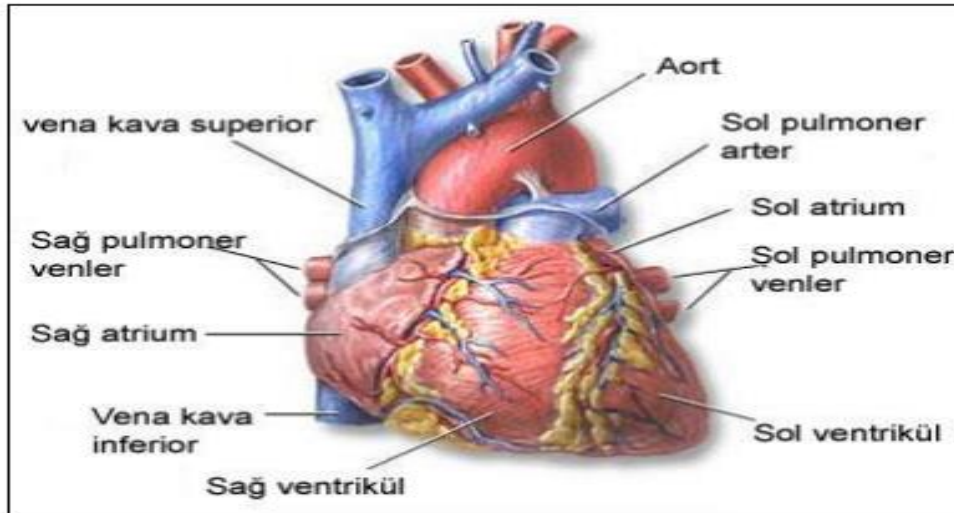
Yaptığım bu çalışmada kullanılan biyolojik işaretler, oluşumu ve özellikleri aşağıda tanıtılmıştır.

2.2.2 Kalp ve Elektrokardiyografi (EKG)

İnsan yaşamının temel yapıtaşlarından olan kalp tüm vücuda kan pompalayan motor organdır. Temel işi kanı pompalamak olan kalp, hücrelerin ihtiyacı olan besin, O₂ vb. temel ihtiyaçları taşıyıp, metabolizma sonucu oluşan artık maddeleri uzaklaştırmak, vücut ısısını ve bağışıklık sistemini düzenlemek gibi birçok hayati faaliyetin gerçekleşmesini sağlar.

Kalp çizgili kas yapısına sahiptir. Çizgili kas yapısına sahip olmasına rağmen isteğimiz dışında, otonom sinir sistemine bağlı olarak çalışır. Güçlü kas dokusuyla ve içinin boş olması sayesinde sürekli kasılıp gevşer ve kanın damarlar içinde dolaşımını sağlar. Kalp vücudun ihtiyaçları doğrultusunda dakikada 5 – 35 litre arasında kan pompalayabilir. Kalp ortalama insan ömründe yaklaşık olarak 300 milyon litre kan pompalar [8].

Kalp göğüs kafesinin içerisinde, iki akciğer arasında, diyafram kasının üzerinde bulunur. Kalbin büyüklüğü yaşa, cinsiyete ve kişiye bağlı olarak değişir. Bireylerin kalbi, yaklaşık olarak kendi yumruğu büyüklüğündedir. Yetişkin bir kadında ortalama 200 – 280 gram, erkekte 250 -390 gram ağırlığındadır [8]. Şekil 2.7’ de kalbin önden görünüşü görülmektedir.



Şekil 2.7 Kalbin Önden Görünüşü [8]

Kalp, iki karıncık ve iki kulakçık olmak üzere dört tane odacığa sahiptir. Ancak oksijen yönünden az olan kanı vücuttan toplayan ve akciğerlerde oksijen bakımından zenginleştirilen kanı vücuda pompalayan bir yapıya sahip olduğu için iki bölümden oluştuğu söylenebilir.

Vücuttan taşınan kirli (oksijen bakımından fakir) kan, alt ve üst toplardamara bağlı olan sağ kulakçıktan kalbe dönmektedir. Sağ karıncık kirli kanı temizlemek (oksijen bakımından zenginleştirmek) için akciğere gönderir. Akciğerde temizlenen kan, sol kulakçıktan kalbe döner. Sol karıncık ise sol kulakçığa gelen temiz kanı vücuttaki organ ve dokulara gönderir [9].

2.2.2.1 Kalbin Çalışması

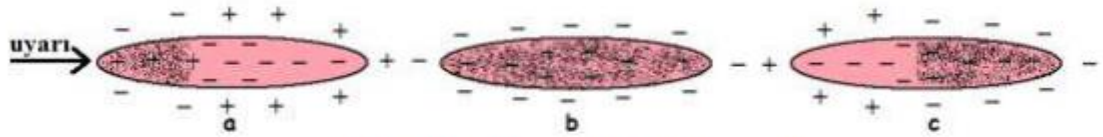
Kalp, sinirsel impulslara (uyarılar) gerek duymadan, kendi uyarılarını oluşturabilen bir kas yapısına sahiptir. Otonom sinir sisteminin kontrolünde çalışır. Kalbin ritmik olarak yaptığı kasılma ve gevşeme hareketlerini, vücudun ihtiyacına göre sempatik sinirler hızlandırırken, parasempatik sinirler yavaşlatır [8].

Kalp, içi boş bir yapıya sahip olması vasıtasıyla, sürekli ve ritmik olarak kasılıp gevşeme hareketleriyle çalışır. Kalbin kasılma hareketine “sistol”, gevşeme hareketine “diastol” denir. Kalpte bulunan atriumlar (kulakçıklar) ve ventriküller (karıncıklar) aynı anda ve birbirlerine zıt olarak kasılır ve gevşerler. Yani atriumların her ikisi aynı anda kasılırken (sistol durumu), ventriküller gevşer (diastol durumu). Tam tersi olarak da atriumların her ikisi aynı anda gevşerken, ventriküller de kasılma hareketi yaparlar. Bu da itici ve çekici bir güç oluşturarak kanın vücuda dağılmasını ve toplanmasını sağlar. Kalbin bir kasılma ve gevşeme hareketine (sistol ve diastol) bir kalp atışı denir. Kalp atışı yetişkin bir insanda dakikada 60–80 ortalama 70’dir, çocuklarda bu sayı dakikada 90–140 arasındadır [9].

Yenidoğanlarda ise kalp hızı doğumdan itibaren 1. aya kadar artar, daha sonra azalmaya başlar. Uyanık haldeki yenidoğanlarda kalp atımı ilk hafta 91-166/dakika, 1. ayın sonunda 107-180/ dakika, 1. Yılın sonunda ise 89-151/dakikadır [10]. Uyku halinde ise 85-90/dakika olabilmektedir [11].

2.2.2.2 Kalbin Elektriksel İletimi ve Ekg Oluşumu

İnsan vücudundaki diğer sistem ve fizyolojik aktivitelerdeki uyarım mekanizmalarında olduğu gibi kalbin çalışması da hücrelerde meydana gelen elektriksel potansiyel değişimi yani aksiyon potansiyelleri sayesinde olur. Kalpteki aksiyon potansiyeli depolarizasyon ve repolarizasyon dalgalarıyla sağlanır. Kalp kası hücreleri uyarım olmadığı, dinlenme zamanlarında hücre içindeki negatif iyonlar hücre dışına göre daha fazladır. Bu durumdaki hücelere polarize hücre denir. Kalp kası hücrelerinin uyarılmasıyla hücre içinde, hücre dışına göre pozitif yüklerin sayısı artar ve depolarizasyon dalgası oluşur. Bu şekilde uyarılan hücelere de depolarize hücre denir. Uyarı bittikten sonra hücrelerin eski haline dönmesi, yani hücre içinin, dışına göre negatif elektrik yükü ile yüklenmesi olayına da repolarizasyon işlemi, bu durumdaki hücreye de repolarize hücre denir [12]. Hücrelerde meydana gelen depolarizasyon ve repolarizasyon işlemi Şekil 2.8’ de görülmektedir.



Şekil 1.2: Depolarizasyon ve repolarizasyon

a) Uyarılan hücre b) Tam depolarize hücre c) Repolarize hücre

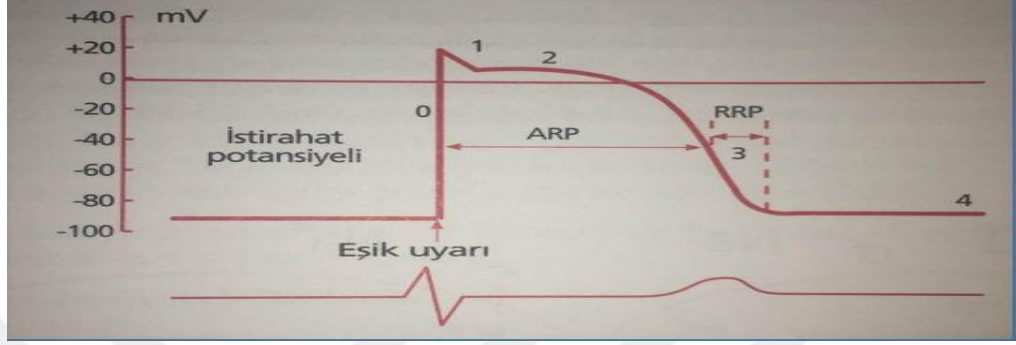
Şekil 2.8 Kalbin elektriksel iletimi depolarizasyon ve repolarizasyon [12]

Kalpte meydana gelen aksiyon potansiyelinin aşamaları aşağıda anlatılmıştır.

- **Faz 0:** Voltage-gated Na^+ (sodyum) kanalları açılır Na^+ girişi artar ve depolarizasyon gerçekleşir.
- **Faz 1:** Aksiyon potansiyeli en yüksek değere ulaştığında (+20 mV), voltage-gated Na^+ kanalları kapanır, K^+ (potasyum) kanallarının açılması ile K^+ çıkışı artar. Hızlı depolarizasyon gerçekleşir.
- **Faz 2:** (Plato Fazı) Bu fazda, Ca^{2+} (kalsiyum) kanallarının açılması ile hızlı K^+ kanalları kapanır ve K^+ çıkışı azalır, Ca^{2+} permeabilitesi artar.
- **Faz 3:** Geç repolarizasyon Ca^{2+} kanallarının kapanması ve yavaş K^+ kanallarının açılması ile sona erer.

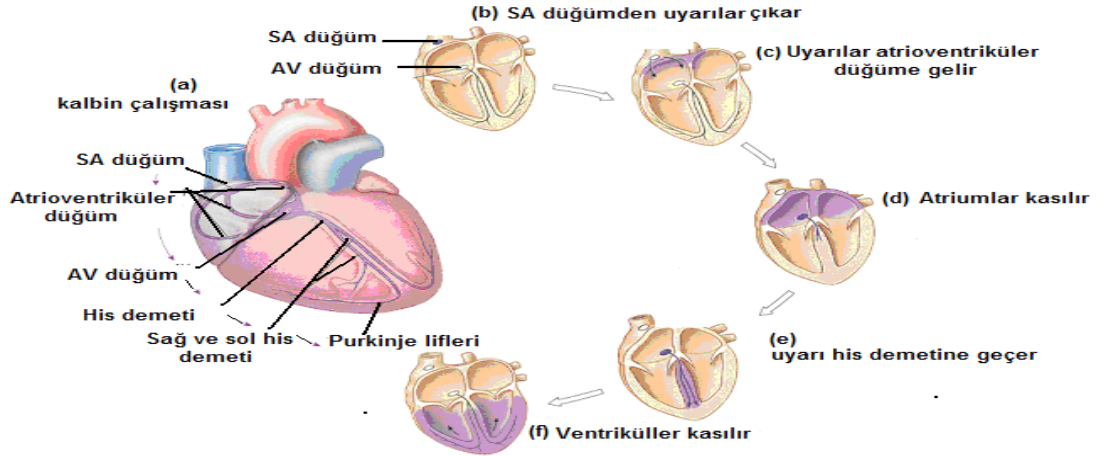
- **Faz 4:** K⁺'un dışarı çıkması ile membran potansiyeli -90 mV düzeyine geri döner [13].

Şekil 2.9' da kalp kasında aksiyon potansiyeli diyagramı gösterilmiştir.



Şekil 2.9 Kalp Kasında Aksiyon Potansiyeli [13]

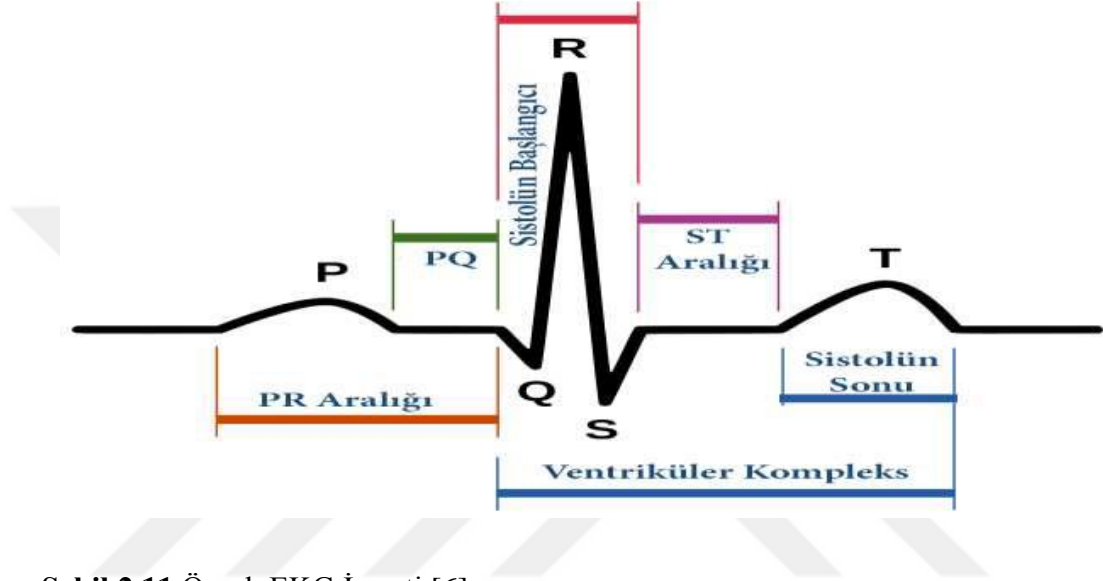
Kalpte sinoatrial düğümde (SA) oluşan uyarılar, düğümler arası yollarla atrioventriküler düğümüne (AV) gelir. Buradan da his demetleri ve Purkinje lifleri vasıtasıyla iletilerek kasılma ve gevşeme faaliyetleri yürütülür [8]. Şekil 2.10' da kalpte meydana gelen iletim ve uyarı akışı gösterilmiştir.



Şekil 2.10 Kalpte Elektriksel Uyarı Akışı [8]

Kalbin aksiyon potansiyelleri sonucu yaptığı sistol ve diastol hareketleri sırasında vücuttaki elektrik potansiyeli değişimlerinin, vücudun iletken yapısı kullanılarak tespit edilmesi, zamana karşı çizdirilip, kaydedilmesi ve grafik oluşturulması işlemine elektrokardiyogram (EKG) denir. EKG görüntülerini oluşturan gelişmiş

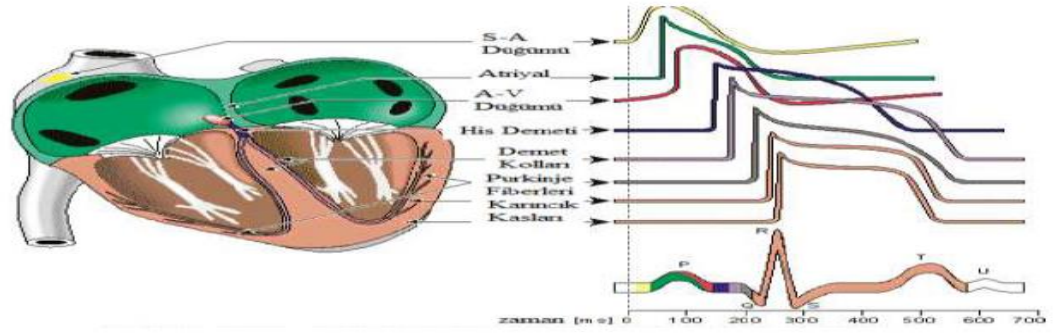
yapıdaki galvanometre cihazlara da elektrokardiyograf denir. Elektrokardiyogram üzerinde, kalbin elektriksel potansiyeline ilişkin yön, genlik ve zaman içinde değişim gibi bilgiler yer alır. Normal elektrokardiyogram, taban çizgisi üzerinde sıralanan ve P, Q, R, S, T, U adları verilen dalgalardan oluşur [14]. Kalpte oluşan bu dalgalar elektrotlar vasıtasıyla algılanır. Şekil 2.11’ de örnek bir EKG işareti görülmektedir.



Şekil 2.11 Örnek EKG İşareti [6]

- P dalgası, SA düğümünden AV düğümüne gelen depolarizasyon dalgalarının sonucudur. Kulakçıklar P dalgası başladıktan 0,1sn sonra kasılır.
- QRS kompleks ventriküler depolarizasyonun sonucudur p dalgasından 0,15 s sonra görülür ve ardından ventriküler kasılma oluşur.
- Ventriküler repolarizasyon sonucu T dalgası oluşur.
- Atrial repolarizasyonu, QRS kompleks tarafından örtülür/belirsizleştirir [8].

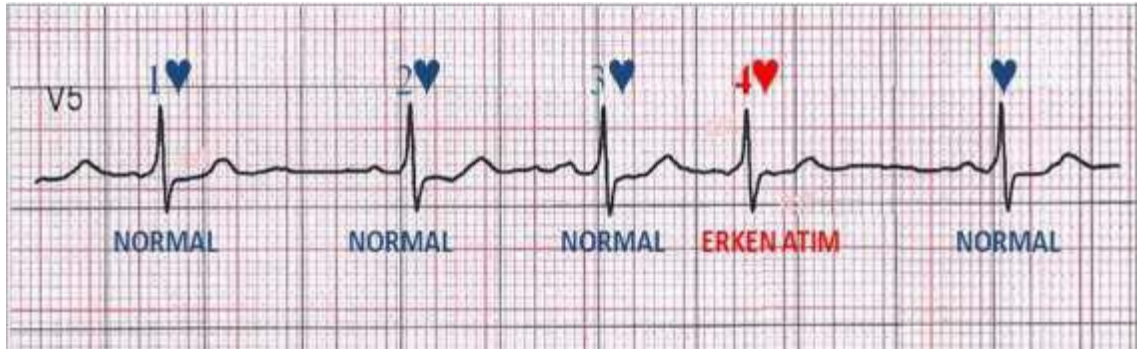
Şekil 2.12’ de kalpteki dokularda meydana gelen elektriksel aktivitelere karşılık gelen EKG işareti içerisindeki dalgalar görülmektedir.



Şekil 2.12 Kalbin Elektriksel Sinyaller Sonucu Oluşturduğu Fiziksel Hareketler

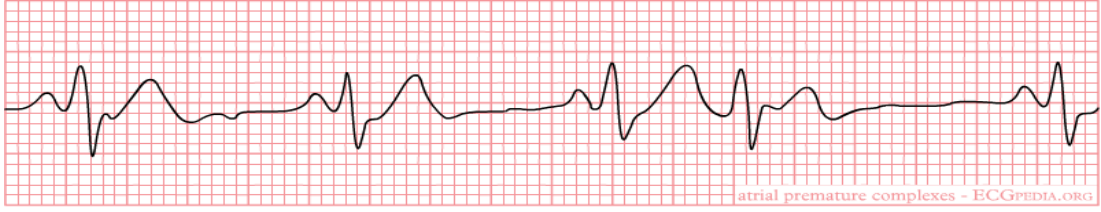
Yeni doğanlarda iletim aralıkları, (PR aralığı, QRS süresi) kalp boyutlarının erişkinlere göre daha küçük boyutta olması sebebiyle, daha kısadır [16].

Herhangi bir kalp rahatsızlığı olmayan kişinin EKG işaretleri düzenli, benzer ve birbirini izleyen bir yapı sergilerken, birbirine benzemeyen ve düzenli olmayan EKG işaretlerinin oluşumu bireylerde kalp rahatsızlıklarının göstergesidir. Şekil 2.13’ de görüldüğü gibi 1, 2 ve 3 numaralı kalp atımları normal sürede ve birbirine benzer yapıda gerçekleşirken, 4. atım beklenenden önce gelmiştir; “erken” atımdır, “ekstrasistol” dır [6].



Şekil 2.13 Rutin Olmayan EKG Grafiği [13]

Yenidoğanlar doğumda, sağ ventrikül sol ventrikülden daha geniş ve kalın olup, sağ ventrikül üzerinde in utero daha büyük fizyolojik stres varlığını göstermektedir. Bu durum yetişkindeki sağ ventrikül hipertrofisi görüntüsünü andıran bir EKG görüntüsü meydana getirir [16]. Şekil 2.17’ de örnek bir yeni doğan EKG görüntüsü görülmektedir.

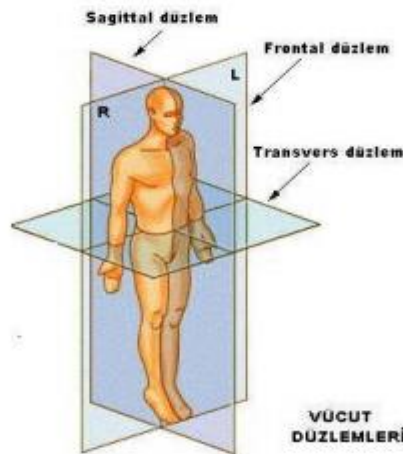


Şekil 2.14 Yenidoğan EKG Görüntüsü [17]

Yenidoğan ve prematürelde kalp hızı normal bireylere göre daha yüksektir. Çocuk büyüdükçe kalp hızı da yavaşlamaktadır [16].

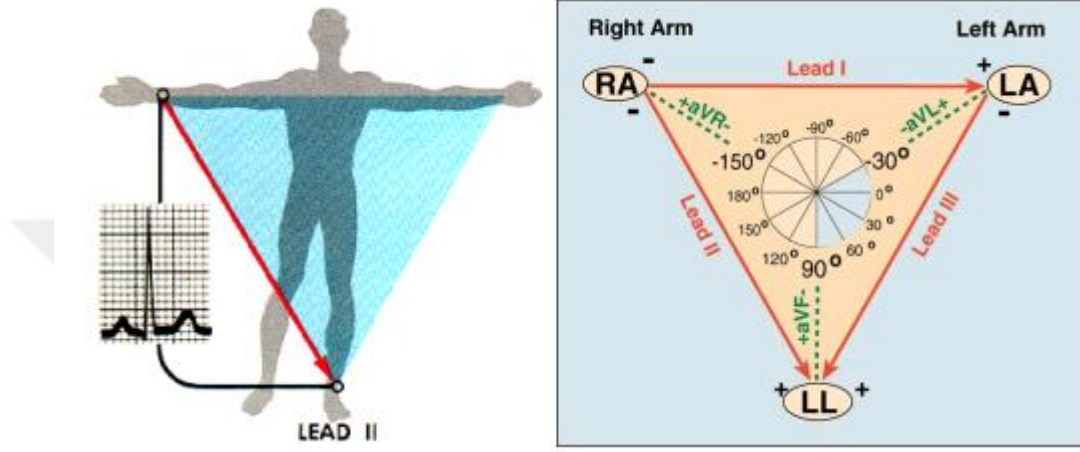
2.2.2.3 EKG Ölçümü

Kalp, gövde içerisinde bir batarya gibi düşünülebilir. Bu batarya vücut içerisinde kapalı bir yapıda olduğu için, doğrudan ölçüm ancak cerrahi operasyon sonucu gövdenin açılıp kalbe ulaşılması ile mümkün olur. EKG' de, hacimsel bir iletken olan gövdenin yüzeyindeki bazı noktalar arasında yapılan potansiyel farkı ölçümleri yardımıyla, (vücut yüzeyine yerleştirilen elektrotlar vasıtasıyla) kalbin çalışması izlenebilir. Kalpte oluşan bu potansiyel farklar insan vücudunun ön tarafına çizdirilirse kardiyak vektörleri oluşur. Bu vektörlerin yönü ve büyüklüğü zamanla değişmektedir. Bu vektörlerin çeşitli düzlemlerdeki eksenler boyunca iz düşümü yapılabilir. Şekil 2.15' de, uygulamada referans düzlem olarak alınan frontal, transverse ve sagittal düzlemler gösterilmiştir [18].



Şekil 2.15 Kardiyak Vektörünün İz Düşürüldüğü Eksenlerin Bulunduğu Düzlemler

EKG ölçüm tekniğinde frontal düzlemindeki kardiyak vektörü iz düşümünün belirlenmesi birbirleriyle 60°'lik açılar yapan üç eksen üzerindeki iz düşümlerinin ölçülmesiyle yapılmaktadır. Bu eksenlerin belirlediği üçgen "Einthoven Üçgeni" adını alır. Şekil 2.16' da bu üçgen, frontal düzlemdeki kardiyak vektörü bileşeni ve bunun diğer eksenler üzerindeki iz düşümleri gösterilmiştir [18].



Şekil 2.16 Frontal Düzlemde Einthoven Üçgeni [18]

2.2.3 Oksijen Satürasyonu (SpO₂) ve Pulse Oksimetre

Oksijen ışığı geçiren, renksiz, tatsız, kokusuz bir gazdır ve insan hayatının vazgeçilmez bir parçasıdır. Oksijen, vücudumuzda hücrelerin enerji ihtiyacını karşılayan yüksek enerjili fosfat bağlarını (ATP) oluşturmak için gereklidir [19].

Oksijen, az bir kısmı erimiş halde olmasına rağmen büyük oranda hemoglobine bağlanmış olarak hücrelere taşınır. Oksijenin kanda hemoglobine bağlanmış olarak taşınan miktarına oksijen satürasyonu (SpO₂) denir [20].

Elektriksel olmayan biyolojik işaretlerden olan SpO₂, önemli vital parametreler arasında yer almaktadır. Çünkü vücuttaki doku ve organlar yeterli düzeyde oksijen alamaz ise geri dönüşümü mümkün olmayan büyük hasarlar görürler.

Sağlıklı erişkinlerin normal vücut sıcaklığında yaklaşık olarak 15 gram hemoglobin bulunduğuna göre bu değerdeki hemoglobin 20.1 ml oksijen bağlayabilir. PaO₂ ' si 95 mmHg olan normal sağlıklı bir kişide SpO₂ yaklaşık olarak %97'dir [20].

Normal sađlıklı yeni dođanlarda ise SpO₂ deđereri ise %93' den bŸyŸk olmalıdır. Ancak yařa gŸre satŸrasyon deđerleri deđiřebilir. Tablo 2.1' de yařa gŸre oksijen satŸrasyonu deđerleri gŸrŸlmektedir.

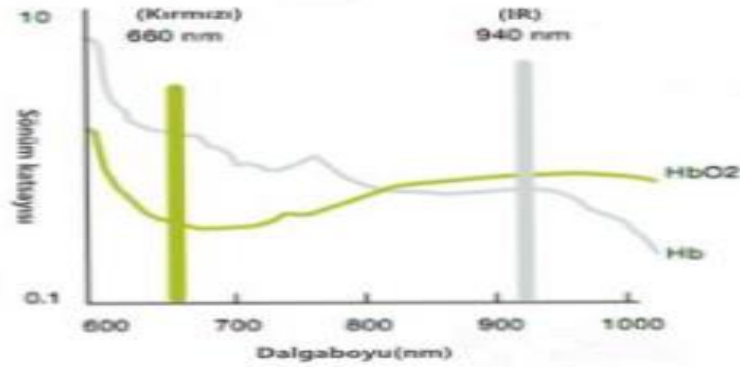
Tablo 2.1 Yařa gŸre oksijen satŸrasyon deđerleri [20]

KATEGORİ	OKSİJEN SATŸRASYONU
PrematŸre	85-95
0-3 ay	95-100
3-6 ay	95-100
6-12 ay	95-100
1-3 yař	95-100
3-6 yař	96-100
6-12 yař	96-100
12 yař ŸstŸ	96-100

Dokuların fizyolojik dŸzeyin altında oksijensiz kalmasına hipoksi denir. Arteriyel hipoksemi ise yeterli hemogloblin miktarı olmasına rađmen, PaO₂' nin olması gereken (yaklařık 80 mmHg altı) dŸzeyin altında olmasıdır. Anoksi durumu ise dokuların tamamen oksijensiz kalmasıdır [19].

Pulse oksimetre, kandaki oksijen miktarını (doygunluđunu) Ÿlçen non-invaziv bir cihazdır. Hemogloblin ve deoksihemogloblin tespiti bazında kandaki salınmıř oksijenin miktarını Ÿlçer [21]. Pulse oksimetre cihazları ile aynı zamanda dakikadaki kalp atım sayısı (nabız) ŸlçŸmŸ de yapılır.

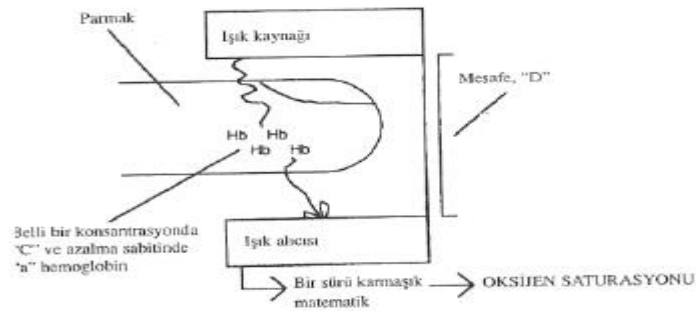
Pulse oksimetre, oksijene bađlı hemogloblin (oksihemogloblin) ve hiç ya da yeterli oksijen bulundurmayan hemogloblinin (deoksihemogloblinin) kırmızı ve kıızılŸtesi ışınları farklı oranlarda absorbe etmeleri prensibine (Lambert-Beer Kanunu) dayanır [22]. Őekil 2.17' de Hb ve HbO₂' nin kırmızı ve IR ışık dalga boylarına gŸre absorbe (sŸnŸm) oranı gŸrŸlmektedir.



Şekil 2.17 Hb Ve HbO₂ Işık Dalga Boyu Eğrileri

Lambert kanununa göre, içinden geçilen maddenin yol uzunluğu arttıkça maddeden geçen ışığın şiddeti azalmaktadır [22]. Beer kanununa göre, maddenin yoğunluğu arttıkça maddeden geçen ışığın şiddeti azalmaktadır [23].

Oksihemoglobin (HbO₂) kızılötesi ışığa karşı duyarlı olup fazla miktarda kızılötesi ışık absorbe etmektedir [24]. Deoksihemoglobin (Hb) ise kırmızı ışığa duyarlıdır ve yüksek miktarda kırmızı ışık absorbe etmektedir. Işık kaynağı ve fotodiyot, parmak ucu, kulak lobu veya perfüze olan, translüminasyon (ışık geçirgenliği iyi) özelliği gösteren dokulara yerleştirilmektedir. Fotodiyota gelen ışığın şiddeti ölçülerek arteriyal pulsasyon hesaplanmaktadır. Bu arteriyal pulsasyon oranı ise bir mikrodenetleyici tarafından kaydedilip ölçülmektedir [25]. Şekil 2.18' de el parmak ucundan yapılan oksijen saturasyonu ölçümü gösterilmektedir.



Şekil 2.18 Pulse Oksimetre Ölçüm Mekanizması [26]

Fonksiyonel hemoglobinler oksijene bağlanabilme özelliğine sahiptirler. Fonksiyonel olmayan hemoglobinler ise bu özelliğe sahip değildirler. Fonksiyonel hemoglobin dört oksijen molekülü ile bağ kurduğu zaman, oksijenli hemoglobin olarak kabul edilir (HbO₂). Dörtten daha az oksijen molekülü taşıdığı zaman bu seyrelmiş hemoglobin olarak adlandırılır (Hb). Fonksiyonel oksijen saturasyonu ölçümü genellikle SpO₂ olarak adlandırılır. Bunun sebebi kanda bulunan HbO₂ ve Hb miktarının periferik ölçümlere dayalı tahmin ve varsayım yöntemiyle bulunmasıdır [27]. (2.1)' de bu durum formulize edilmiştir.

$$SpO_2 = \frac{HbO_2}{Hb+HbO_2} \quad (2.1)$$

2.2.3.1 Pulse Oksimetre Kullanım Endikasyonları

Pulse Oksimetreler yenidoğanların ilk dakikalarının değerlendirilmesinde, pediatrik ve yenidoğan yoğun bakım ünitelerinde, prematürelere ya da çocuklara ameliyat, biyopsi veya endoskopi gibi invazif işlemler yapılması esnasında, kuvözdeki bebeklerin, serviste yatan ve stabil olmayan çocukların gözlemlenmesi ve tedavi işlemlerinde önemli bir rol üstlenmektedir [28].

Prematüre retinopatisi olanlarda kandaki oksijen düzeyinin sürekli takip edilmesi gerekmektedir. Hastalıkların tedavisinde kullanılan ilaç ya da solüsyonların kandaki oksijen düzeyinde değişikliklerine neden olabildiğinden bu değişiklikler pulse oksimetreler ile hızlı bir şekilde tespit edilebilmektedir. Yoğun bakım gerektiren hastalarda ortaya çıkabilecek hipoksi hızlı şekilde belirlenerek hipokseminin ciddi sonuçlarının azaltılmasını, düzeltilmesini sağlamaktadır [26].

2.2.3.2 Pulse Oksimetre Kullanımının Kontrendikasyonları

Pulse oksimetrelerin bilinen herhangi bir kontrendikasyonu yoktur. Pulse oksimetrede kullanılan bazı proplar uzun süre kullanıma bağlı olarak, propun ışık kaynağının sıcaklığından ya da doku üzerine yaptığı basınçtan dolayı rahatsız edici olabilmektedir. Isı şiddetli olması durumunda hasta da yanıklar meydana gelebilir [26].

2.2.3.3 Pulse Oksimetre Probenun Uygulandığı Yerler

Hastaların oksijen satürasyonunun ölçülmesi ve değerlendirilmesi amacıyla iyi perfüze olan;

- El parmağı
- Ayak parmağı
- Burun bölgesi
- Alın bölgesi
- Kulak memesi dokuları kullanılmaktadır [26].

Şekil 2.20' de pulse oksimetre prob örnekleri görülmektedir.



Şekil 2.19 Pulse Oksimetre Çeşitleri [26]

2.2.4 Nabız

Kanın vücuda dağılırken damar duvarlarına yaptığı basınca tansiyon denir. Kalbin kasılması (sistol hareketi) sırasında oluşan basınca büyük tansiyon, kalbin dinlenmesi (diastol hareketi) sırasında oluşan basınca da küçük tansiyon denir. Kan basıncının tespitinde, cıva basıncına göre belirlenen standart birim kullanılır [29].

Normal bir erişkin insanda sistolik basınç (büyük tansiyon) ortalama olarak 120 mmHg, diastolik basınç (küçük tansiyon) da ortalama 80 mmHg olarak bilinir [29].

Tablo 2.2 Sağlıklı 2000 gr Altında Prematürelere KB Değerleri (mm/Hg) [30]

Tablo II. Sağlıklı 2000gr altında prematürelere KB değerleri (mm/Hg)

Günler	Sistolik		Diastolik	
	Min.	Max.	Min.	Max.
1	48±9	63±12	25±7	35±10
4	57±10	71±11	32±8	45±10
7	61±7	74±12	34±9	46±10

Nabız, kalbin sistolik ve diastolik basınç farkından dolayı damarlar üzerinde hissedilen basınç dalgasıdır. Nabız kalbin hızı, ritmi ve düzenli çalışıp, çalışmadığı bilgisini veren önemli bir vital parametredir. Nabız dalgasının şekli, kalış süresi ve ne kadar belirgin olduğu birçok kalp ve kalp kapakçıkları hastalıklarının teşhis ve tedavi işlemlerinde kullanılan önemli bir değerdir. [31]. Tablo 2.3’ de yaş gruplarına göre ortalama nabız atım sayıları listelenmiştir.

Tablo 2.3 Nabız Atım Sayıları [32]

Normal nabız atım sayısı

Yaş grubu	Nabız atım sayısı (atım/dk)
Süt çocuğu	120-160
1-3 yaş	90-140
3-5 yaş	80-110
5-10 yaş	75-100
10-18 yaş	60-100

2.2.5 Vücut Sıcaklığı

Yenidoğan bebekte vücut yüzeyinin vücut ağırlığına oranı erişkinlere göre fazladır. Prematüre bebeklerde ise bu oran daha da fazladır [33]. Sağlıklı bebekler ekstremitelerini gövdelerine yaklaştırarak pozisyonlarını ayarlarlar ve böylece vücut yüzey alanlarını azaltarak, vücut sıcaklığını korumaya çalışırlar. Prematüre bebekler ise yeterli gelişimleri olmadıkları için pozisyonlarını düzenleyemezler ve daha fazla ısı kaybederler [33-34]. Bu da hipotermiye yol açarak prematüre için birçok rahatsızlığa neden olur. Bu yüzden prematürelere vücut ısı takibi oldukça önemlidir. Bebeklerde vücut ısı 36-37,5 °C arasında normal kabul edilir.

Bebeklerin vücut sıcaklığı, küvöz içinde ısıya duyarlı cilt probu ya da kulaktan timpanik ölçüm ile tespit edilmesi tercih edilir.

2.3 Termal Görüntüleme

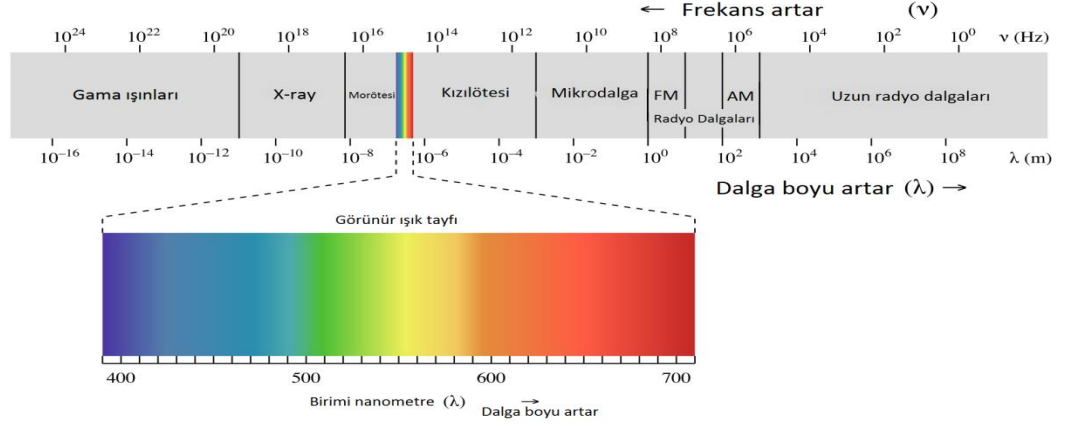
1940' lı yıllarda gelişimi başlayan kızılötesi (infrared) görüntüleme sistemi tıp ve endüstri alanlarında kullanılmaya başlanmıştır. 1960 ve 70'li yıllar boyunca da Amerika, Avrupa ve Japonya kızılötesi görüntüleme sistemlerini daha da geliştirerek yaygın şekilde kullanmaya başlamışlardır [36].

Sıcaklığı mutlak sıfır (-273°K) noktasının üzerindeki bütün cisimler elektromanyetik spektrum bölgesinin kızılötesi dalga boyunda ısı enerjisi yayarlar. Cismin sıcaklığı, atomlarının titreşimi ile doğru orantılıdır. Atomların titreşim ve rotasyon hareketleri ışıma yol açar. Bu titreşim ve rotasyon hareketlerindeki artış ışıma miktarını da artırır [37].

Termografi kavramı “sıcaklık resmi” anlamına gelen kelimelerin köklerinden elde edilmiştir. Kızılötesi termografi ışınımı ve bununla ilişkili olarak yüzey sıcaklığını algılamak ve ölçmek üzere, elektronik optik cihazlar kullanma bilimidir [38].

Modern termal görüntüleme yöntemi; canlı ya da nesnelerin ısısına bağlı olarak yaydığı kızılötesi ışınımı algılamak, ölçmek ve yüzey sıcaklığını tespit etmek üzere optik cihaz ve sistemler kullanılarak uygulanır. Termografi, termal görüntüleme veya termal video, kızılötesi görüntülemenin bir çeşididir. Termografik kameralar

elektromanyetik spektrumun kızılötesi bölümündeki (yaklaşık 900-14000 nanometre dalga boyunda) elektromanyetik ışınımı tespit ederler ve bu ışınımdan insan görsel spektrumunu tarafından anlaşılabilir, görünür ışık bölümünde (yaklaşık 400-700 nanometre dalga boyunda) resimler oluştururlar [38]. Böylece bu cihaz ve sistemler sayesinde canlı ve nesnelerin yüzey sıcaklığı ve sıcaklık değişimleri tespit edilebilir.

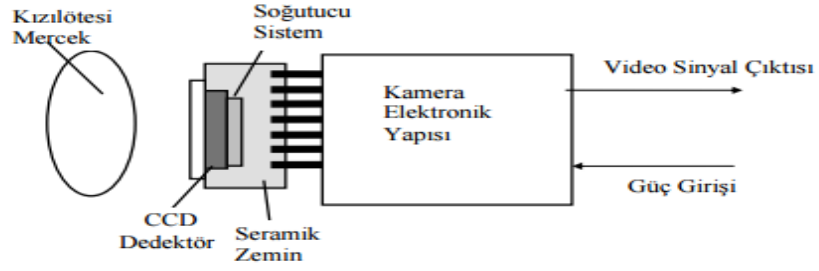


Şekil 2.20 Elektromanyetik Spektrum

2.3.1 Termal Kameralar ve Çalışma Prensibi

Termal kameralar yukarıda da bahsedildiği üzere doğrudan bir temas gerçekleştirilmeden, nesne ya da canlıların ısılarına bağlı olarak yaydıkları IR dalgaları tespit eder ve bu dalgalara bağlı olarak insan gözünün görebileceği resimler oluştururlar. Normal kameralar görüntüyü ışık sayesinde oluştururken termal kameralar görüntüyü ısı sayesinde oluştururlar [38]. Normal bir kameranın algıladığı dalga boyu yaklaşık 450–750 nanometre (görünür bölge) aralığındayken, kızılötesi kameralar 7,5–13 μm dalga boyunda çalışmaktadır [39].

İnsan beyni ve gözü görüntüyü oluşturmada renkleri kullanır. Işığı kullanırken de renk farklılıklarından yararlanır. Termal kameralar, kullanılan sensör ve objektifler ile çok küçük sıcaklık farklarını yakalayabilen (0.01°C gibi) ve bu farklılıktan görüntü oluşturabilen özelliktedirler [38]. Termal kameralar yazılım ve algoritmalarına göre, her sıcaklık değerinde farklı IR yayılımları olması sayesinde meydana getirdikleri resimlerde renk ve ton farklarını oluştururlar. Şekil 2.21’ de termal kameraların genel yapısı görülmektedir.



Şekil 2.21 Termal Kamera Genel Yapısı [41]

Termal kameralar, nesnenin ortama yaydığı kızılötesi enerjiyi, kameranın lensi tarafından (Bu lenslerin birçoğu germanyumdan yapılmıştır) algılayarak kamera içindeki kızılötesi detektöre gönderir. Detektör ise resmin işlenmesi için bilgileri sensöre gönderir. Bu işlemlerin sonucunda bilgiler bizim görebileceğimiz bir resme dönüştürülür [42].

Termal kameralar yukarıda bahsedilen özelliklerinden dolayı birçok alanda kullanılmaktadır. Tablo 2.4' de termal kameraların kullanım alanları görülmektedir.

Tablo 2.4 Termal Kamera Kullanım Alanları [38]

Savunma	Nesne tanımlama, keşif yapma vb.
Sağlık	Damar tıkanıklıklarının tespiti, mamografi, kırık bölge tespiti vb.
Elektrik	Motorlar, dağıtım merkezleri, trafolar vb.
İnşaat	Çatı rutubeti kontrolü, hava sızıntısı vb.
Çevre	Yer altı kaynaklarının bulunması, kirlilik kontrolü vb.
Sivil	Yangınla mücadele, suçla mücadele vb.

2.3.2 Medikal Termografi

Medikal uygulamalarda; Termografi, hastanın deri yüzeyindeki ısı paternlerini arařtıran, kaydeden ve termogram adı verilen görüntüsünü oluřturan tanısai bir görüntüleme yöntemidir [37].

Termal görüntüleme yöntemi, hastaya herhangi bir katater uygulamadan, radyasyon ya da kontrast madde verilmeden, hastanın cilt yüzeyindeki sıcaklık deęerlerini ölçen non-invazif bir uygulamadır [43].

Termal görüntülerin saęlık alanında kullanımına 1959 yılında bařlanmıřtır [36]. Vücut yüzey sıcaklıęı yař, cinsiyet, kilo, metabolizma, topografya ve damarlardan akan kanın miktarı gibi birçok parametreye baęlı olarak deęiřiklik gösterir [44, 45, 46]. Vücutta aęrıyan veya anormallik gösteren bölgelerdeki çok küçük ısı farkları, cilt yüzeyinden yayılan kızılötesi dalgaların deęerlerine yansır. Bu deęerler 8-14 μ 'a duyarlı termal kamera dedektörleri vasıtasıyla algılanarak elektriksel sinyale dönüřtürülür [43, 35, 47]. Bu sinyallerdeki farklılıklar renk skalasındaki farklılıklar ile eřleřtirilerek renkli dijital görüntüleme çevrilir ve sonuçta insan vücudunun termogram adı verilen termal resmi elde edilir [48]. Kızılötesi kamera ile cilt yüzeyinden yaklaşık 6 mm derinlięe kadar ısı yayılımı algılanıp sıcaklık resmi elde edilebilir [45]. Bu resimlerdeki renk farklılıkları vücut yüzeyinden yayılan kızılötesi ışınların miktarındaki deęiřim ile oluřur. Vücudun normal sıcaklık haritası simetrik bir yapıya sahip olduęundan asimetrik sıcaklık deęerleri kolaylıkla fark edilir. Vücuttaki simetrik bölgeler arasında sıcaklık farklılıęı (anamlı bir deęiřim için en az 10°C fark oluřmalı) söz konusu ise bu farklılıklar termal görüntüleme ile belirlenerek aęrının, fizyolojik ve fonksiyonel bozukluęun gerçek merkezi tespit edilebilir [35, 49].

Termal görüntüleme yöntemi ile hastalık teřhis yönteminde iki tür uygulama vardır. Bunlar sabit çevre sıcaklıęında yapılan ve termal strese sokulan hastalar üzerinde yapılan çalıřmalardır. Her iki yöntemde de sıcaklık řiddetinin ölçümü deęil, sıcaklık deęiřimleri deęerlendirilir ve tıbbi uygulamalarda sıklıkla kullanılır [46, 50].

Termografinin uygulandıđı klinik uygulamalar;

- Meme kanseri ve hastalıkları,
- Dolařım sistemi ve hastalıkları,
 - i) Periferik hastalıkların teřhisi,
- Nöromuskuler sistem ve hastalıkları,
 - i) Akut ve kronik sırt ağrıları, disk hernileri,
- Santral ve periferik sinir sistemi hastalıkları,
 - i) Otonom sinir sistemi hastalıkları, migren, yüz felci, spinal sinir yaralanmaları, karpal tünel sendromu,
- Kas, eklem ve iskelet sistemi hastalıkları,
- Romatizma, eklem iltihapları, tendinit, gut hastalığı, kas yaralanmalarıdır [50, 51].

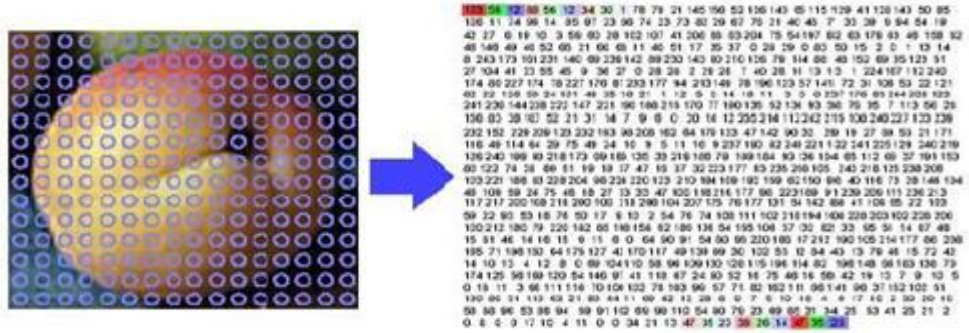
Termal görüntüleme sisteminin, genel olarak diđer metotlara göre avantajları:

- IR görüntülemenin insan vücuduna hiçbir zararı yoktur.
- Vücuda doğrudan bir teması olmayıp, non-invazifdir.
- Pasif bir cihazdır. Çalışma yapılırken vücuda zararlı hiç bir X, gama vb. radyoaktif ışın, sinyal, ilaç ya da solüsyon verilmez. Yalnızca vücuttan yayılan IR ışınları tespit eder.
- Çalışma yapılırken vücutta herhangi bir ağrı, sızı, sancı vb. durumlar yaşanmaz.
- Vücuttaki fizyolojik ve fonksiyonel bozukluklarının tespitinde ve teřhis edilen hastalıkların tedavi sürecinin öncesinde ve sonrasında uygulanabilir.

- Isı hassasiyetleri çok düşüktür ve çok küçük ısı farklılıklarını (yaklaşık 0,1°C) algılayabileceğinden, çeşitli hastalıklara erken tanı konabilir.
- Subjektif bir bilgi olan ağrının yeri ve yoğunluğunu gösteren tek tanı yöntemidir [44, 45].

2.4 Görüntü İşleme

Bilgisayar ortamında her bir görüntü sayısal verilerden oluşur ve $I(x,y)$ gibi bir fonksiyon ile temsil edilir. Burada x, y koordinat düzlemi konumunda yer alan veri, bilgisayar ortamındaki en küçük eleman olan piksel olarak adlandırılır. Her piksel için o noktadaki parlaklık ve renk bilgisi söz konusudur [52]. Şekil 2.22’ de bilgisayar ortamında saklanan sayısal bir resim ve bu resmin piksellerine ait parlaklık ve renk bilgilerini ifade eden decimal kod verileri görülmektedir.



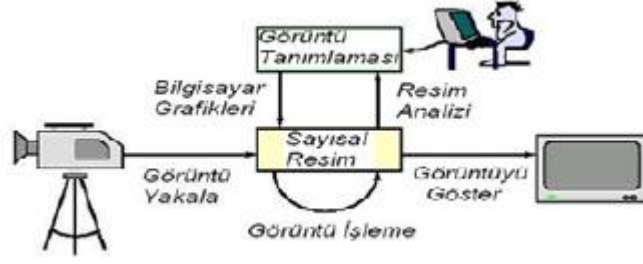
Şekil 2.22 Resim Piksel Matrisi [53]

Doğadaki tüm renkler RGB (Red, Green, Blue) kırmızı, yeşil ve mavi renk kanallarının farklı tonlarda bir araya gelmesiyle oluşur. Bilgisayar dilinde ise bu ton farkları HEX ya da Desimal kodu ile ifade edilir. Tablo 2.5’ de dijital resimlerdeki renklerin, HEX ve Desimal formatta renk bilgileri görülmektedir.

Tablo 2.5 RGB Renk Uzayı Desimal ve HEX Kod Bilgileri

RENK	HEX KODU	DECİMAL KODU
Kırmızı	#FF0000	255,0,0
Yeşil	#00FF00	0,255,0
Mavi	#0000FF	0,0,255
Sarı	#FFFF00	255,255,0
Cyan	#00FFFF	0,255,255
Mor	#FF00FF	255,0,255
Beyaz	#FFFFFF	255,255,255
Siyah	#000000	0,0,0

Görüntü işleme, sayısal görüntüler üzerinde amaca yönelik olarak birtakım işlemlerin gerçekleştirilmesi ile yeni görüntülerin ve çeşitli bilgilerin elde edilmesini amaçlayan önemli bir çalışma alanıdır. Tıp, mühendislik, fizik, biyoloji, grafik, astronomi, biyomedikal, robotik, hukuk, sanat, iletişim ve bilgisayar bilimleri gibi bir çok alanda uygulama imkanı bulabilmekte ve teknolojik gelişmelerle birlikte hızlı bir biçimde gerçekleştirilebilmektedir [52]. Şekil 2.23' de görüntü işlemenin genel olarak sistem yapısı görülmektedir.



Şekil 2.23 Görüntü İşleme Sistem Yapısı [53]

Görüntü işleme teknikleri sayısal hale getirilerek dijitalleştirilmiş resim, video, fotoğraf, animasyon vb. her türlü görüntü üzerinde uygulanabilir. Görüntü işleme metotları farklı birçok programlama dili platformunda tasarlanabilecek algoritma ve yazılımlarla, açık kaynak kütüphanelerle ya da hazır paket bilgisayar yazılımlarıyla sayısal görüntüler üzerinde işlem yapmaya ve kullanım amacına uygun hale getirmeye olanak tanır. Görüntü işleme teknikleriyle sayısal görüntülerin tamamı üzerinde işlemler yapılabilirken noktasal ya da lokal çalışmalar da yapılabilir.

Görüntü işleme metotlarıyla sayısal resimler üzerinde birçok işlem yapılabilir. Bu işlemleri üç temel başlık altında toplamak mümkündür. Bunlar alt seviye, orta seviye ve yüksek seviye görüntü işleme uygulamaları olarak adlandırılabilir. Alt seviye uygulamalarda, görüntüler üzerinde gürültüyü azaltma, keskinleştirme, bulanıklaştırma vb. filtre uygulamaları uygulanır. Bu uygulama ile görüntüler, insan gözüyle daha net anlaşılabilir ve istenilen amaca göre yorum, analiz vb. işlemler yapmaya uygun hale getirilir. Ayrıca orta ve yüksek seviyeli işlemlere hazırlık için ön işleme tekniğidir. Orta seviye uygulamalarda ise görüntüyü sınıflandırma, nesne ayrıştırma, bölütleme vb. işlemler uygulanır. Yani bu uygulamada görüntü içerisindeki nesnelere (nesne bulma, çıkarma, sınıflandırma, kenar bulma vb.) işlem yapılabilmektedir. Yüksek seviyeli uygulamalarda ise görüntüler üzerinden anlam çıkararak işlem ve çalışmalar yapılabilir. Elektronik kontrol sistemleri, parmak izi ve yüz tanıma, insansız hava araçları vb. uygulamalar yüksek seviye görüntü işleme uygulamaları olarak örnek verilebilir.

Görüntü işleme teknikleri yapılabilecek temel uygulamalar aşağıda belirtilmiştir.

- Görüntüleri yeniden boyutlandırma ve ölçeklendirme, eksenler üzerinde döndürme işlemleri
- Görüntü üzerindeki parazit ve gürültüleri filtreleyerek, görüntüyü netleştirme
- Parlaklaştırma, keskinleştirme vb. işlemlerle renk ayarı yapma
- Görüntüler üzerinde renk analizi, renk ve ton değişimleri yapma
- Görüntüler üzerinde birleştirme ve ayrıştırma işlemleri yapma,
- Görüntüler üzerinde nesne bulma, çıkarma, sayma, sınıflandırma vb. işlemler
- İstatistiksel bilgiler edinme
- Matematiksel ve morfolojik (biçimsel) işlemler yapma

Görüntü işleme teknikleri hızla gelişen teknolojik yeniliklerle beraber birçok alanda kullanılmak üzere kendine yer bulmuştur. Bunlardan bazıları aşağıda özetlenmiştir.

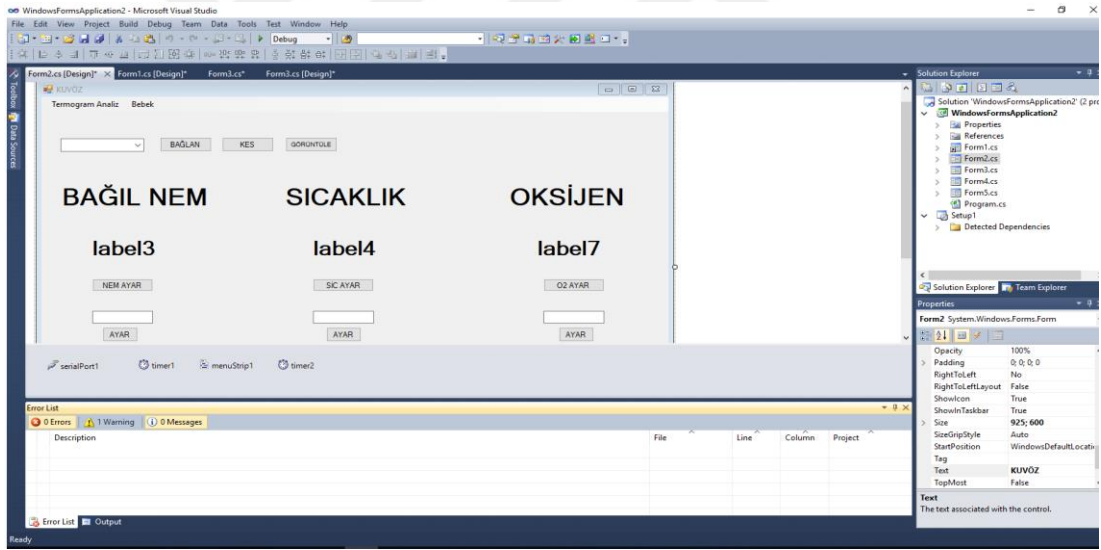
- Tıp alanında; Tümör tespiti ve boyutlandırılması, damar yollarının belirginleştirilmesi ve tıkanıklığının tespit edilmesi, kemik, kas ve doku tespiti, kanserli hücre tespiti vb.
- Biyometrik tanıma alanı; Parmak izi ve yüz tespiti
- Askeri alanlarda; Füze, uçak, silah vb. donanımlar ve araçlar tespiti, yeraltı ve yer üstü görüntüleme
- İnsansız araç ve robotik sistemler; İnsansız hava, kara ve deniz araçları yön ve istikamet tespiti, robotik uygulamalar
- Trafik, radar, astronomi alanları; Araç plaka tanıma, gökyüzü cisimleri ile ilgili işlemler, hava, kara, deniz cisim tespiti
- Geoteknik, Jeoloji alanları; Maden arama ve tespit işlemleri
- İmalat sanayi; İmal edilen ürünlerde hata tespiti, kalite kontrol çalışmaları vb.

2.5 Microsoft Visual C#

C#, Microsoft tarafından geliştirilmiş, yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiş, nesne yönelimli, esnek, güçlü, basit ve kullanımı kolay bir programlama dilidir. Ayrıca C#, ortak platformlarda taşınabilir bir (portable language) programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır. C# programının en önemli özelliği ise .Net Framework

platformu için hazırlanmış ve tamamen nesne yönelimli bir yazılım dilidir. Yani nesnelere önceden sınıflar halinde yazılır. Programcı, yalnızca nesneyi istediği alana taşıyarak, amaca göre nesneyi çalıştıracak kodları yazar ve programlarını oluşturur [54].

Microsoft tarafından Anders Hejlsberg öncülüğünde tasarlanan C# dili orta düzey programlama dilleri arasında girmiştir. Java, C++ ve Visual Basic dillerinde yer alan eksiklik ve tutarsızlıkları ortadan kaldırdığı için kısa zamanda geniş bir kullanıcı kitlesine sahip olmuştur. C# dili Windows form uygulamaları ve araçları sayesinde kullanım kolaylığı ve hızlı çalışma imkânı sunarak, masaüstü, web ortamı, veri tabanı vb. birçok uygulama ile birlikte görsel kullanıcı ara yüzü geliştirilmesine olanak tanır. Şekil 2.24’ de örnek bir Microsoft Visual C# tasarım ve form görüntüsü görülmektedir.



Şekil 2.24 Microsoft Visual C# Tasarım ve Form Görüntüsü

Ayrıca C#’ da bulunan derleyici (debugger) ile programdaki hatalar bulunmakta ve kullanıcı ara yüzünde bir pencereye listelenmektedir. Böylece mevcut hatalar kullanıcılar tarafından kolayca tespit edilip düzeltilebilmektedir.

2.5.1 Microsoft Visual C# Özellikleri

Microsoft Visual C# programlama dilinin özellikleri aşağıda özetlenmiştir.

- Windows form uygulamaları ve form araçları ile nesne tabanlı bir programlama dili olduğu için öğrenilmesi ve kullanılması kolay bir dildir. Kısa zamanda ve hızlı bir şekilde projeler yapılabilir.
- Derleyici (debugger) seçeneği ile programda hata tespiti ve düzeltilmesi kolaydır.
- Kullanımı kolay olmasına rağmen hızlı ve güçlü programlar geliştirilebilir.
- XML desteğine sahiptir.
- Masaüstü, Web, veritabanı ve mobil uygulamalar geliştirmek için uygundur.
- Hazır kütüphane seçenekleri birçok uygulamada kullanım kolaylığı sağlamaktadır.
- Öğrenilmesi kolaydır.
- .Net desteği olan diller için güçlü, hızlı ve esnek DLL (Dynamic Link Library) bileşenleri yazma özelliğine sahiptir.
- WPF uygulamaları geliştirilebilmektedir [55].

2.5.2 Microsoft Visual C# İle Neler Yapılabilir

- **Windows Form Uygulamaları (Ado.Net)** : Grafikselsel kullanıcı ara yüzü (GUI—Graphical User Interface) olarak tanımlanır. Windows form uygulamalarını, konsol uygulamalarından ayıran en önemli özelliği de budur. Windows Form uygulaması oluşturmak için Toolbox adı verilen araç kutusundan nesnel araçlar sürükleyip bırak yöntemi ile kolayca oluşturulabilir. Windows Form Application seçeneği ile windows ve kullanıcı ara yüzü uygulamaları yapılabilir. Ayrıca grafikselsel olarak gelişmiş çalışmalar da yapmak mümkündür. Bununla birlikte form uygulamaları Ado.Net uygulaması ile veri tabanı bağlantıları yapılabilir [54].
- **Web Form Uygulaması (Asp.Net):** Asp. NET uygulaması ile web uygulamaları yapılabilir. Windows form uygulamalarından farklı olarak web tarayıcısı ile çalışmaktadır [54].
- **Mobil Programlama:** Kısaca veri iletimi için kullanılan sistemlerdir diyebiliriz. Evrensel veri transfer metodu sayılan XML ve http kuralları ile internet ağı ile dünyanın her hangi bir yerine veri taşıyan sistemlerdir [54].

- **DLL yazma:** DLL (Dinamic Link Library) yani dinamik kütüphaneler oluşturulabilir.

2.6 A.Forge.Net

Aforge.NET, bilgisayarlı görü ve yapay zeka (resim işleme, yapay sinir ağları, genetik algoritma gibi) konularıyla ilgilenen geliştiriciler ve araştırmacılar için tasarlanmış bir C# kütüphanesidir [56].

Bu kütüphanelerin toplandığı ana başlıklar şöyledir;

- **Aforge.Imaging** – Resim işleme rutinleri ve filtreleri kütüphaneleri
- **Aforge.Vision** – Bilgisayarlı görü kütüphaneleri
- **Aforge.Video** – Video işleme kütüphaneleri
- **Aforge.Neuro** – Yapay sinir ağları kütüphaneleri
- **Aforge.Genetic**–Genetik algoritma ve evrimsel programlama kütüphaneleri
- **Aforge.Fuzzy** – Bulanık hesaplamalar kütüphanesi
- **Aforge.Robotics** – Bazı robot kitlerini destekleyen kütüphane;
- **Aforge.MachineLearning** – Makine öğrenme kütüphanesi, vb. [56]

Bu framework (çerçeve) 2006 yılında Andrew Kirillov tarafından geliştirilmiş ve hizmete sunulmuştur. Açık kaynaklı olması nedeniyle iyileştirilmesi ve geliştirilmesi sürekli devam etmektedir. Aforge.NET çerçevesi tarafından sağlanan kütüphaneler ve sınıflar doğrudan C# programında uygulanabilmektedir. Bu sayede satırlarca kod yazmak yerine tek bir sınıf kullanarak etkili çıktılar elde edilebilir. Bu da kullanıcıların çalışmalarını hızlı ve kolay yapabilmelerine imkân sunar.

3. ORTAM ve VİTAL PARAMETRELERİN ÖLÇÜM VE KONTROL YÖNTEMİ

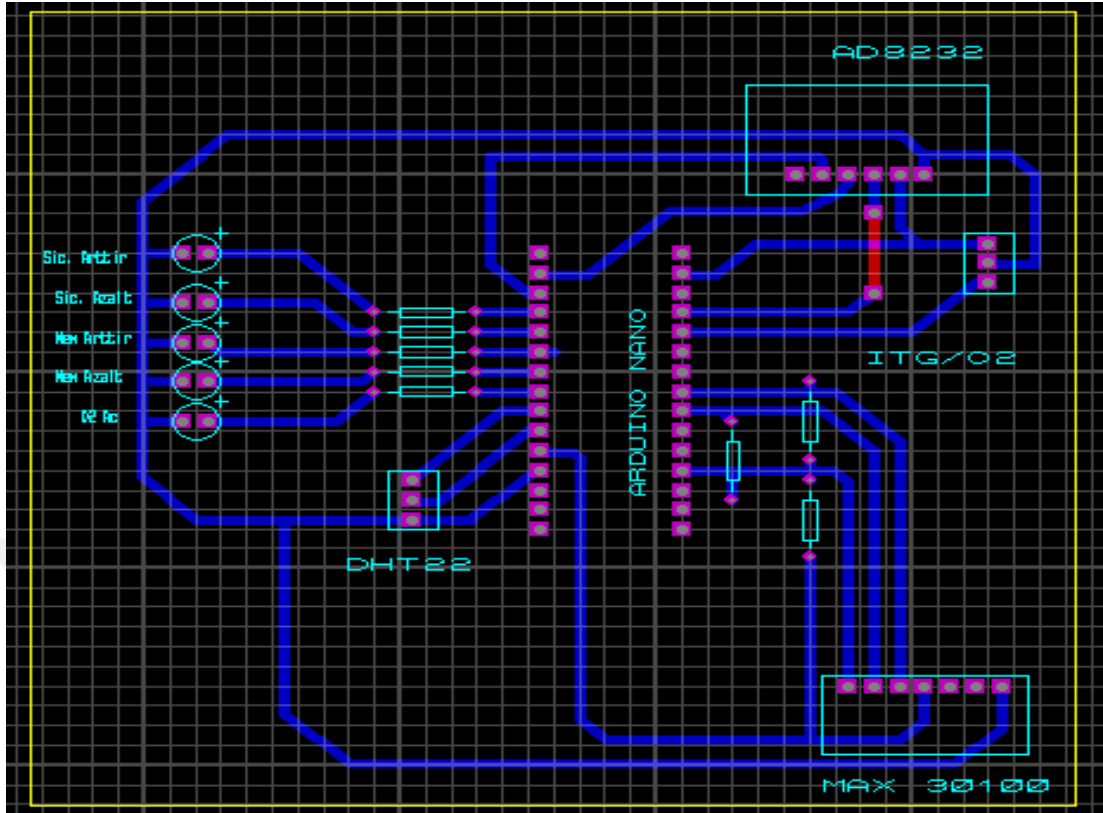
3.1 Sistemin Genel Çalışma Yapısı

Tasarladığım kuvöz kontrol sisteminde ısıtma, havalandırma ve nemlendirme parametrelerine ilave olarak bebek vücudundaki vital (yaşamsal) parametrelerden olan nabız, EKG ve SpO₂ değerleri, vücut yüzeyine yerleştirilen sensörler ve elektrotlar vasıtasıyla ölçülmektedir. Bütün ölçüm ve kontrol sistemi Microsoft Visual C# ortamında tasarlanan kullanıcı ara yüz programına entegre edilmiştir. Bu ara yüz programında prematürenin vital parametreleri görüntülenmekte, kuvözün çevresel parametreleri ise görüntülenip, kontrol edilebilmektedir.

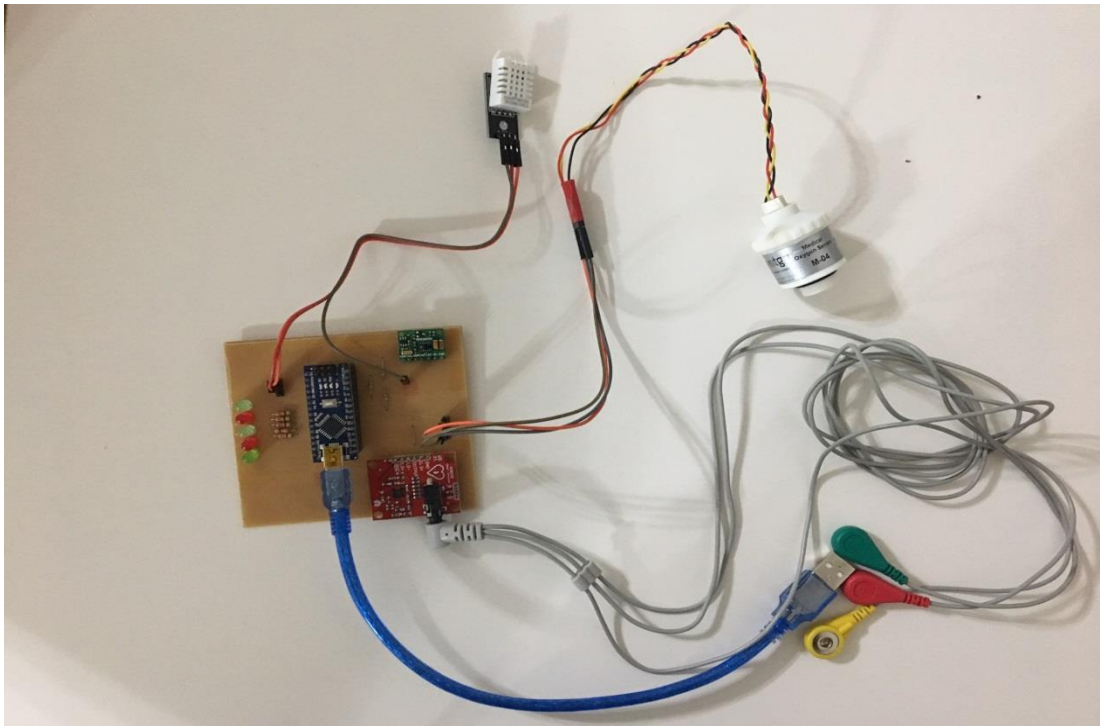
Mikrodenetleyici kartı olarak Arduino Nano kullanılmıştır. Arduino Nano kartı ilgili sensörler vasıtasıyla kuvözün çevresel parametreleri olan sıcaklık, nem ve O₂ konsantrasyonu değerleri ile prematürenin vital parametre değerlerini (nabız, EKG ve SpO₂) ölçerek kullanıcı ara yüzüne gönderme ve kullanıcı ara yüzünden gelen komutları ilgili birimlere (fan, ısıtıcı, nemlendirici, O₂ kaynağı vb.) göndererek ortam değerlerini kontrol etmek üzerine programlanmıştır.

Tasarladığım sistemde kullanılan Arduino Nano mikrodenetleyici kartı, sensörler, ara yüz programı ve bunların birbirleriyle bağlantıları bu bölümde anlatılmıştır.

Tasarlanan sistemin baskı devre şeması Şekil 3.1, devre kartı Şekil 3.2' de görülmektedir.



Şekil 3.1 Kuvöz Kontrol Kartı Baskı Devre Şeması



Şekil 3.2 Kuvöz Kontrol Devre Kartı

3.2 Arduino

Arduino, İtalyan elektronik mühendisleri tarafından açık kaynak kodlu olarak geliştirilen bir mikrodenetleyici kartı ve geliştirme ortamıdır. İsteyen herkesin devre şemalarını temin ederek kendi devrelerini yapabilecekleri, isterlerse devre elemanları yerleştirilmiş hazır paket olarak satın alabilecekleri, esnek, kolay kullanımlı donanım ve yazılım tabanlı bir fiziksel programlama platformudur [57].

Yaptığım çalışmada maliyetinin ucuz olması, farklı oranlarda besleme voltajı sağlayabilmesi (3,3 ve 5 V), paket yapıda olması, kolay programlanabilir olması ve çok sayıda açık kaynak kütüphanesi bulunması nedeniyle Arduino kartı kullanılmıştır. Boyutlarının küçük olması, üzerinde bulunan bağlantı elemanları (header) sayesinde bakır plakete kolayca lehimlenebilmesi özellikleri sebebiyle de Arduino modellerinden Arduino Nano kartı kullanılmıştır. Arduino kartları ve Arduino Nano modeli ile ilgili detaylı bilgi Ek-1' de verilmiştir.

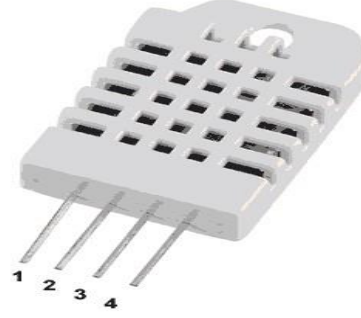
3.3 Kuvöz Ortam Parametreleri

Kuvözler, prematürelerin hayatlarına devam edebilmeleri için ısıtma, havalandırma, nemlendirme koşullarını optimum seviyede tutarak, şeffaf ve steril bir ortamda korunmasını sağlar. Bu nedenle kuvözlerdeki ortam parametrelerinin sürekli ölçülerek, ideal seviyede tutulması büyük önem arz etmektedir. Tasarladığım sistemde, kuvözdeki sıcaklık, nem ve oksijen konsantrasyonu değerlerini ölçen sensörler ve Arduino Nano kartına bağlantıları aşağıda anlatılmıştır.

3.3.1 Sıcaklık ve Nem Sensörü

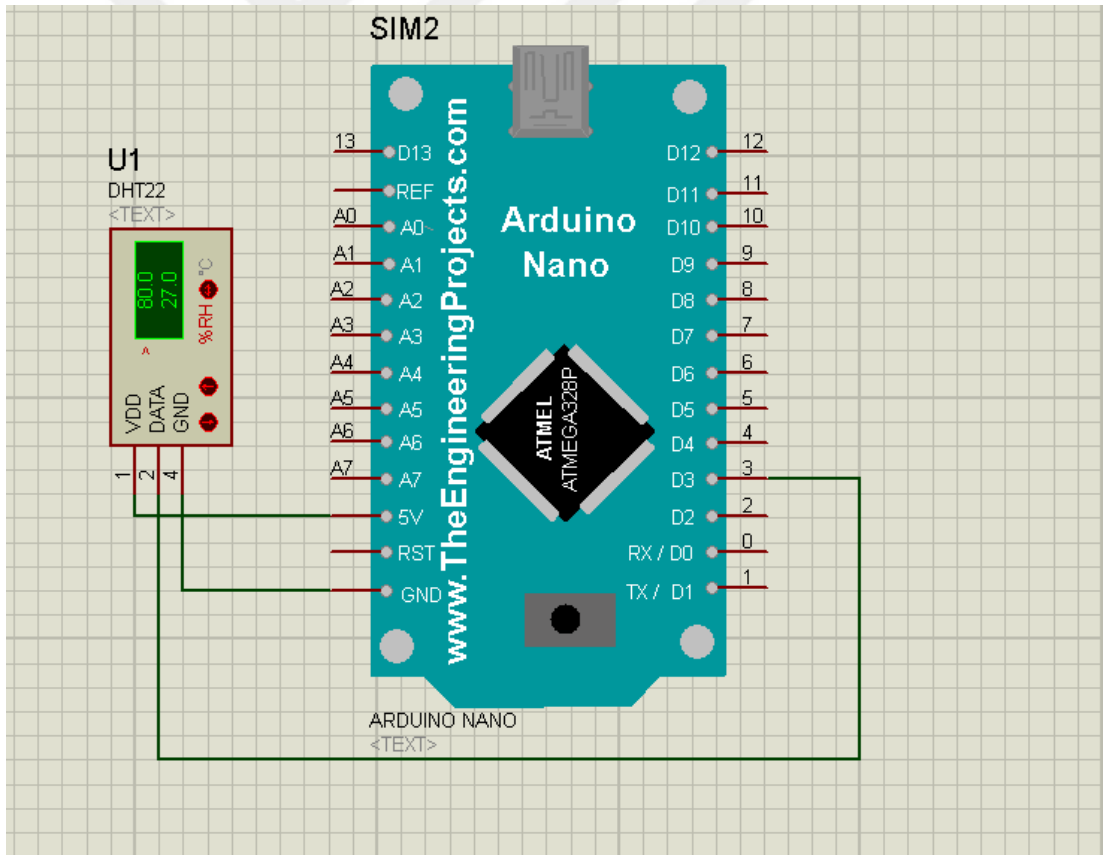
Kuvözdeki sıcaklık ve nem miktarını ölçmek için Dht22 entegresi kullanılmıştır. Tek bir entegre ile kuvözdeki hem sıcaklık hem de nem değerleri ölçülebilmektedir. Bu sayede kullanım kolaylığı ve maliyet tasarrufu sağlanmaktadır. Sensör 4 pinli bir yapıya sahiptir (1-Vcc 2- Data 3- Boş 4- Gnd). Sensör Şekil 3.3' de görülmektedir. Dht22 sensörü hakkında detaylı bilgi Ek-2' de verilmiştir.

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Şekil 3.3 Dht22 Sensörü [59]

Şekil 3.4' de Arduino Nano kartı ile Dht22 sensörünün bağlantı şeması görülmektedir.



Şekil 3.4 Arduino Nano Kartı ve Dht22 Bağlantısı

Yapılan çalışmada Dht22' ye Arduino Nano kartından 5V besleme voltajı ve toprak hattı sağlanır. Sensörün data hattı ise Arduino kartının D3 pinine bağlanır. Dht22 data hattından sıcaklık ve nem bilgisini, bit paketleri şeklinde Arduino Nano kartına

gönderir. Arduino kartı, sensörün açık kaynak kütüphanelerinden de yararlanarak sıcaklık ve nem bilgilerini decimal formata çevirerek seri haberleşme yoluyla kullanıcı ara yüzüne göndermek üzere programlanmıştır.

TS EN 60601-2-19 standartına göre kuvözlerde sıcaklık ölçüm toleransı $\pm 0,5^{\circ} \text{C}$, nem ölçüm toleransı ise $\pm 10 \% \text{ RH}$ olarak tespit edilmiştir. Tasarlanan modelde kullanılan Dht22 entegresi $\pm 0,5^{\circ} \text{C}$ değerinde sıcaklık, $\pm 5 \% \text{ RH}$ değerinde ise bağıl nem ölçüm doğruluk oranında çalıştığı için ilgili standarta uygundur.

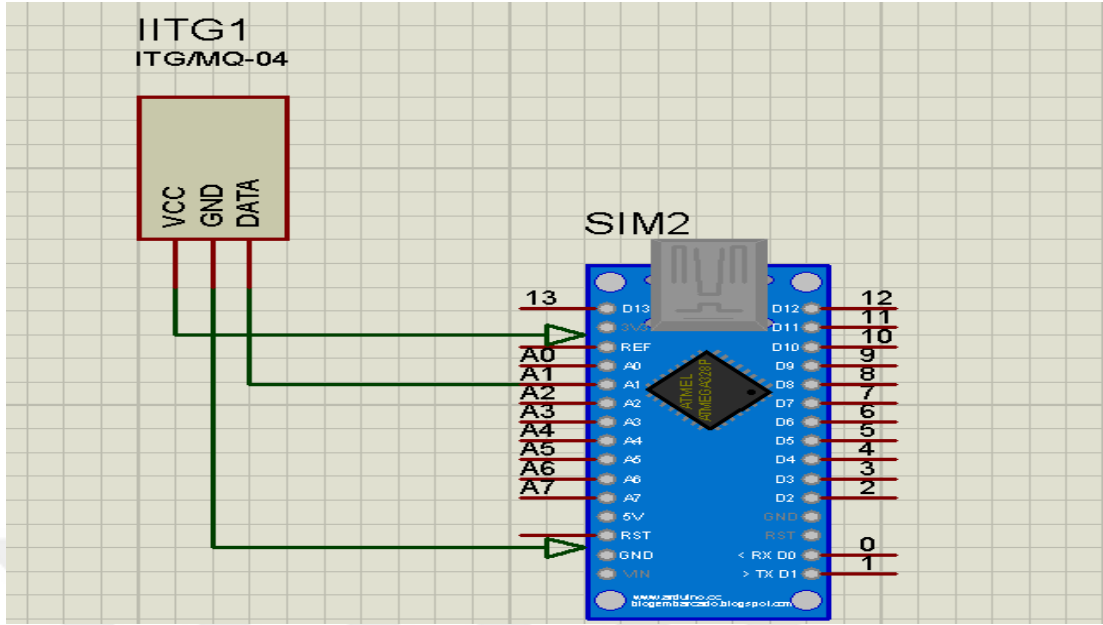
3.3.2 Oksijen Konsantrasyonu Sensörü

Kuvözdeki oksijen konsantrasyonunu ölçmek için itg/M-04 medikal oksijen sensörü kullanılmıştır. Sensör 3 Pinli (VDD, Toprak, Analog çıkış pini) yapıya sahiptir ve ortamdaki oksijen miktarına göre 12,5 – 16,5 mV aralığında analog çıkış gerilimi üretir. itg/M-04 medikal oksijen sensörü Şekil 3.5’ de görülmektedir. Sensör hakkında detaylı bilgi Ek-3’ de verilmiştir.



Şekil 3.5 itg/M-04 Medikal Oksijen Sensörü [60]

Yapılan çalışmada, sensörün çalışması için ihtiyaç duyduğu 3,3V besleme voltajı ve toprak hattı Arduino Nano kartından sağlanmıştır. Sensörün data pini Arduino kartının analog pinine (A1) bağlanmıştır. Şekil 3.6’ de itg/M-04 sensörü ile Arduino kartının bağlantısı görülmektedir.



Şekil 3.6 Arduino Nano Kartı ve Itg/M-04 Sensörü Bağlantısı

Yapılan çalışmada Arduino kartı sensörden gelen gerilim miktarına göre ortamdaki O₂ oranını hesaplamak üzere programlanmıştır. Hesapladığı değerleri de seri haberleşme yoluyla kullanıcı ara yüzüne gönderir.

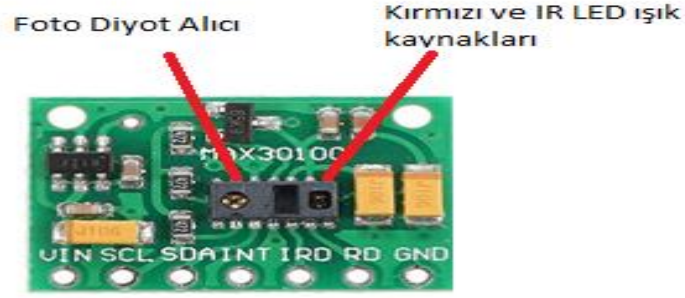
TS EN 60601-2-19 standartına göre O₂ ölçüm toleransı +/- 0.5 olarak verilmiştir. Tasarlanan sistemde <= +/- 0.3 doğruluk oranında çalışan itg/M-04 medikal oksijen sensörü kullanılmıştır.

3.4 Prematüre Vital Parametreleri

Bu bölümde prematürenin yaşamsal fonksiyonlarını gösteren vital parametrelerinin ölçümünde kullanılan sensör ve entegreler tanıtılmıştır.

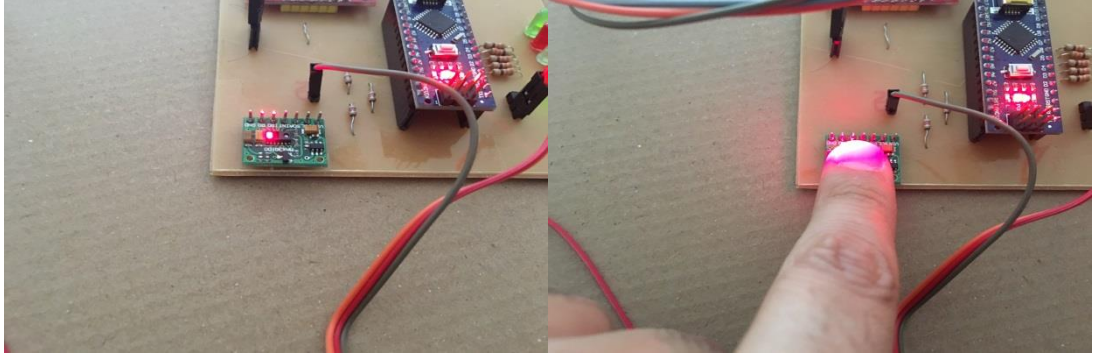
3.4.1 Nabız ve SpO₂ Ölçümü

Prematürenin nabız ve SpO₂ değerlerini ölçmek için Max30100 entegresi kullanılmıştır. Entegre içeriğinde bulunan kırmızı ve kızılötesi LED ışık kaynaklarından yayılan ışığın parmak ucundan yansımaları ve sönmelerinden ölçüm olarak çalışır. Max30100 entegresi ile ilgili detaylı bilgi Ek-4' de verilmiştir. Şekil 3.7' de Max30100 entegresi görülmektedir.



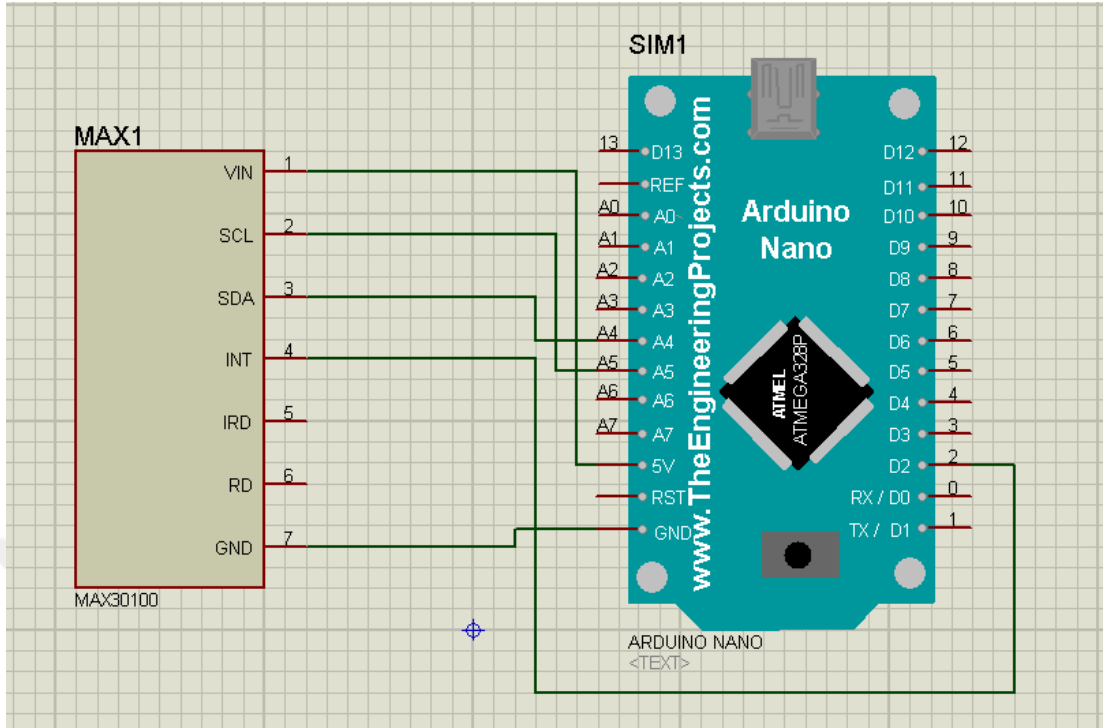
Şekil 3.7 Max30100 Entegresi [61]

Bu entegre ile pramatürenin el parmak ucundan ölçüm yapılması planlanmıştır. Şekil 3.8’ de bu ölçüm şekli görülmektedir.



Şekil 3.8 Max30100 Entegresi ile Parmak Ucundan Ölçüm Yapılması

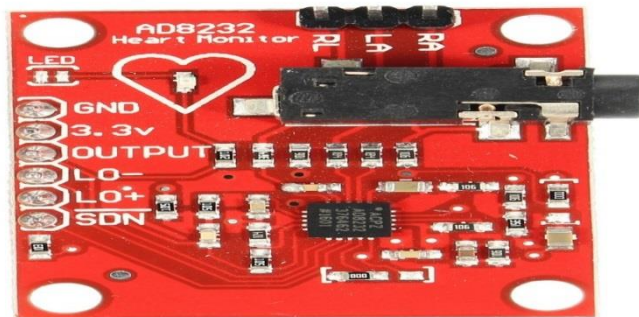
Max30100 entegresi I²C haberleşme metodu ile Arduino kartına veri aktarır. Entegrenin SDA pini Arduino Nano kartındaki SDA pini olan A4 pinine, SCL pini de Arduino Nano kartındaki SCL pini olan A5 pinine bağlanır. Entegre üzerindeki INT pini de Arduino Nano kartında D2 pinine bağlanmıştır. Entegre 5V besleme voltajı ve toprak hattı ihtiyacını da Aruino Nano kartından temin etmek üzere tasarlanmıştır. Şekil 3.9’ da Arduino Nano kartı ile Max30100 bağlantı şeması görülmektedir.



Şekil 3.9 Arduino Nano kartı ile Max30100 Entegresi Bağlantı Şeması

3.4.2 EKG Ölçümü

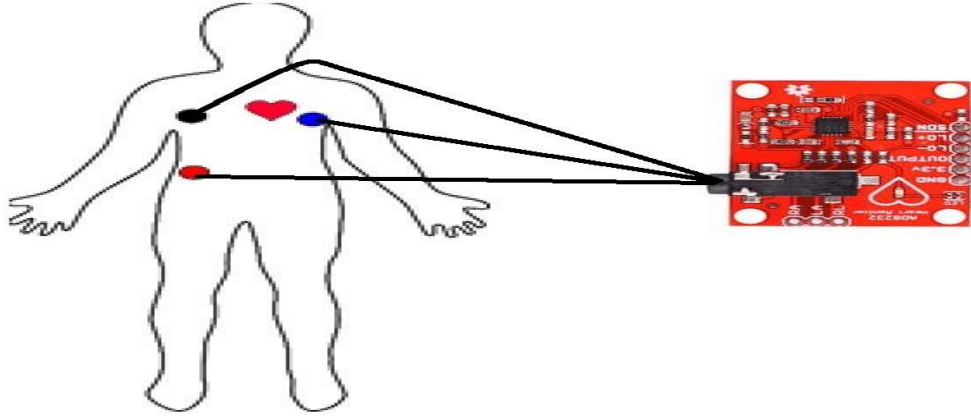
Tasarladığım sistemde EKG ölçümleri için Ad3232 entegresi kullanılmıştır. Bu entegre küçük biyopotansiyel sinyalleri gürültüden arındırır, büyütür ve filtreler. Ad8232 entegresi ile ilgili detaylı bilgi Ek-5’ de anlatılmıştır. Şekil 3.10’ da Ad8232 entegresi görülmektedir.



Şekil 3.10 Ad8232 Entegresi [62]

Ad8232 entegresi, üzerinde bulunan 3.5 mm TRS dişi ses fişine ölçüm kablosu bağlanarak çalışır. Ölçüm kablolarındaki pinler vücuda yerleştirilen elektrotlara

bağlanarak kalp aktiviteleri sonucu elde edilen analog verileri entegreye gönderir. Yaptığım çalışmada kalp sinyallerini vücut yüzeyinden tespiti için tek kullanımlık AgCl (Gümüş Klorür) elektrotlar kullanılmıştır. Ad8232 ile vücut bağlantısı Şekil 3.11’ de görülmektedir.



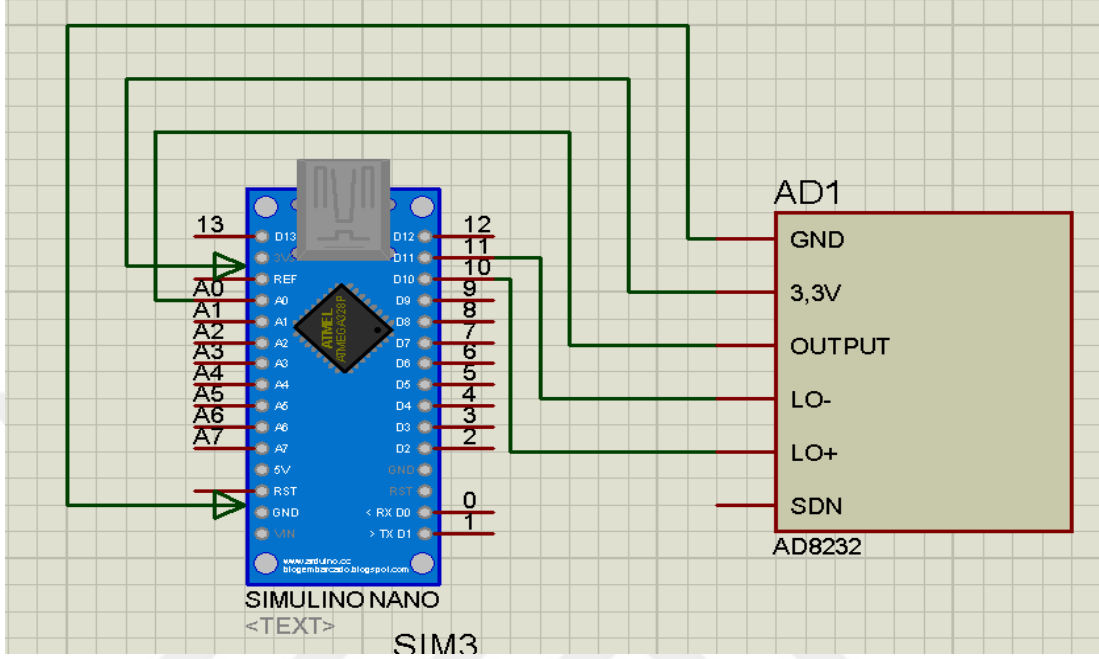
Şekil 3.11 Ad8232 ve Vücut-Elektrot Bağlantısı

Tablo 3.1 Ad8232 Ölçüm Kablosu Pinleri ile Vücut Elektrot Bağlantısı

Kablo Rengi	Sinyal
Siyah	RA (Sağ Kol)
Mavi	LA (Sol Kol)
Kırmızı	RL (Sağ Ayak)

Ad8232 elde ettiği EKG verilerini işlemden geçirerek anlamlı hale getirdikten sonra OUTPUT pini vasıtasıyla Arduino Nano kartına gönderir. Entegre LO- ve LO+ pinleri vasıtasıyla Arduino Nano kartına sinyal göndererek, EKG verilerinin güvenilirliğini sağlar. Ad8232 entresinin çalışması için ihtiyaç duyduğu 3,3V besleme voltajı ve toprak hattı bağlantısı yine Arduino Nano kartından sağlanmıştır.

Arduino Nano kartı ile Ad8232 entegresinin bağlantı pinleri Şekil 3.12’ de görülmektedir.



Şekil 3.12 Arduino Nano Kartı ile Ad8232 Bağlantısı

3.4.3 Vücut Sıcaklığı Ölçümü

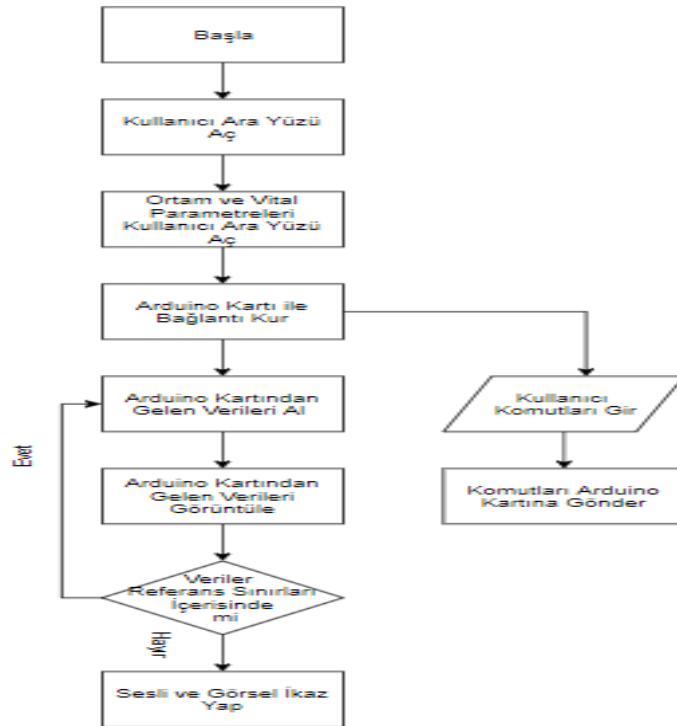
Tasarlanan sistemde termal kameralar ile elde edilen prematüre termogramları üzerinde ısı analizi işlemleri yapılabileceği için vücut sıcaklığı tespitinin termal görüntülerden elde edilmesi uygun bulunmuştur. Aksi takdirde prematürenin vücut yüzeyinden sensörler vasıtasıyla ölçüm yapılması sonucu elde edilecek veriler, termogramlardan elde edilen verilerle uyuşmaması halinde (termal kameralardan elde edilecek ölçümler sıcaklık sensörlerine göre çok daha hassas verilerdir.) kafa karışıklığına ve sağlık personellerinin yanlış karar vermesine neden olabilir.

3.5 Ortam ve Vital Parametreler Kullanıcı Ara Yüz Programı

Yapılan çalışmada, Arduino Nano kartı yukarıda bahsedilen sensörler vasıtasıyla hem kuvözün çevresel parametre hem de prematürenin vital parametre verilerini olarak Microsoft Visual C# programında tasarlanan kullanıcı ara yüzüne gönderir. Bu sayede kullanıcılar kuvözün ısı, nem ve O₂ konsantrasyonu verilerini ve prematürenin EKG, SpO₂ ve nabız değerlerini tek bir ara yüzde görüntüleme imkânı

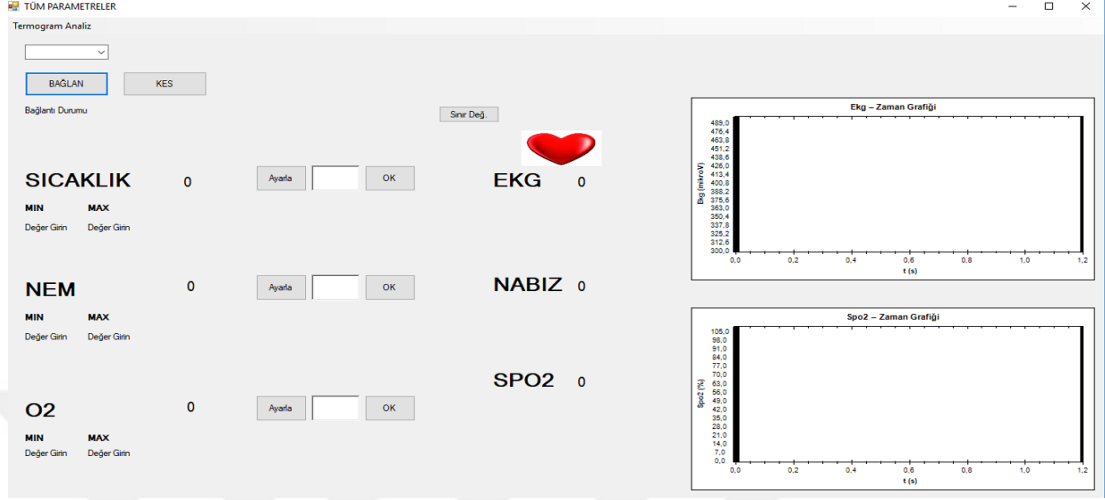
bulur. Arduino Nano kartının yukarıda bahsedilen sensörlerden ölçüm verilerini alması ve ara yüze göndermesi işlemleri için tasarlanmış ve kartın içine gömülmüş program Ek-6’ da verilmiştir.

Arduino Nano kartı ile ara yüz programı arasındaki veri akışı seri haberleşme vasıtasıyla, 115200 kb/sn bant genişliğinde yapılır. Kuvözdeki ortam parametreleri kullanıcı ara yüzü vasıtasıyla kontrol edilir. Ara yüzden alınan kullanıcı komutları Arduino kartına gönderilir. Arduino, gelen komut verilerini değerlendirir ve kuvöz ortam parametrelerini, istenilen değerlere ulaşmasını sağlamak için ilgili birime (fan, ısıtıcı, O₂ kaynağı) komut göndermek ve kontrol etmek üzere programlanmıştır. Ayrıca ara yüz yazılımının ortam ve vital parametreleri ekranı, kuvözdeki çevresel parametrelerin ayarlanan referans aralığının dışına çıkması, vital parametrelerin ise sağlık sınırları dışına çıkması durumunda program hem görsel, hem de işitsel uyarılar vermek üzere tasarlanmıştır. Şekil 3.13’ de programın akış diyagramı görülmektedir.



Şekil 3.13 Ortam ve Vital Parametreleri Programının Akış Diyagramı

Ara yüz programında ortam ve vital parametrelerinin görüntülediği ekran çıktısı Şekil 3.14’ de görülmektedir.



Şekil 3.14 Ara Yüz Programında Ortam ve Vital Parametrelerinin Görüntülenip Kontrol Edildiği Ekran

Program açıldıktan sonra menüden Tüm Parametreler sekmesi seçilerek Ortam ve Vital Parametreler ara yüzü ekranı görüntülenir. Ara yüz üzerinde Arduino Nano kartının bağlı olduğu port seçilerek BAĞLAN butonuna basıldığında Arduino kartından veri akışı başlar. Bütün parametrelerin görüntülediği formun ekran çıktısı Şekil 3.15 ‘ de görülmektedir.



Şekil3.15 Ortam ve Vital Parametreleri Formu Ekran Görüntüsü

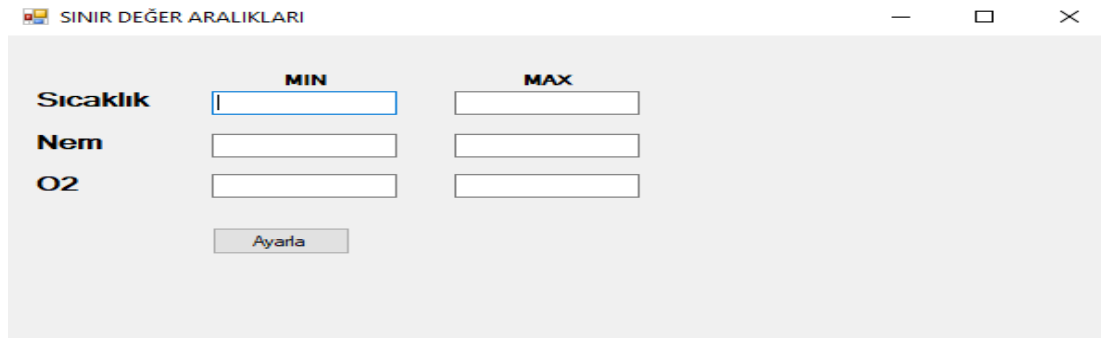
Ara yüz programı bu veri akışını Timer fonksiyonu vasıtasıyla yapar. Arduino kartından bütün ölçüm verileri paketler halinde gelir. Program gelen paketlerdeki verileri ayırarak, her bir ölçüm parametresi için oluşturulan dizilerdeki elemanlara sırasıyla atar. Bu işlem için tasarlanan algoritma Şekil 3.16’ de görülmektedir.

```
private void timer1_Tick(object sender, EventArgs e) // Seri porttan gelen verileri okuyor.//
{
    // Arduino kartı bütün ölçüm verilerini paket şeklinde gönderir ve verilerin karışmaması için * işareti konmuştur.
    //C# programı gelen verileri dizi hale getirir ve her bir veriyi * işareti vasıtasıyla ayırır.
    timer1.Interval = 1;
    string sonuc = serialPort1.ReadLine().ToString();
    string[] pot = sonuc.Split('*');
    float nem = (float)(Math.Round(Convert.ToDouble(pot[0]) / 100, 2));
    float isi = (float)(Math.Round(Convert.ToDouble(pot[1]) / 100, 2));
    float o2 = (float)(Math.Round(Convert.ToDouble(pot[2]) / 100, 2));
    float nabiz = (float)(Math.Round(Convert.ToDouble(pot[3]) / 100, 2));
    float spo2 = (float)(Math.Round(Convert.ToDouble(pot[4]), 0));
    int ekg = (int)(Math.Round(Convert.ToDecimal(pot[5]), 0));

    label7.Text = isi.ToString();
    label8.Text = nem.ToString();
    label9.Text = o2.ToString(); //Gelen veriler string formatına çevrilerek form üzerinde ilgili alanlarda görüntülenir.
    label11.Text = ekg.ToString();
    label12.Text = nabiz.ToString();
    label13.Text = spo2.ToString();
}
```

Şekil 3.16 Ara Yüz Programının Arduino Kartından Ölçüm Değerlerini Almasını Sağlayan Algoritma

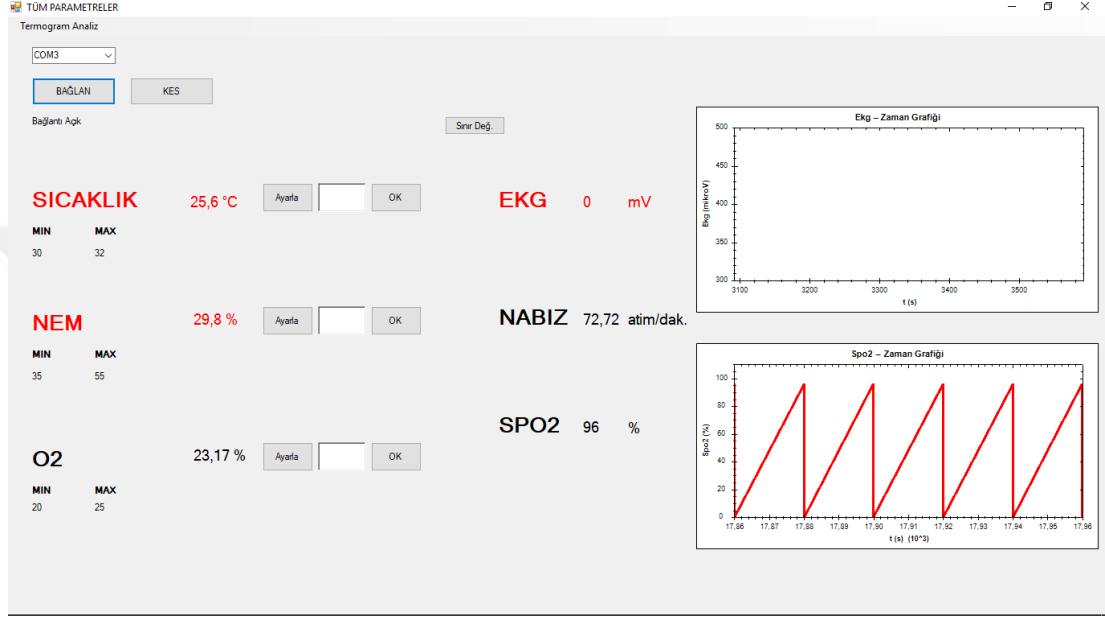
Sınır Değer Butonu basılarak, açılacak forma ortam parametreleri (Sıcaklık, nem ve O₂ konsantrasyonu) sınır değer aralıkları girilmelidir. Ortam parametre verileri Sınır değerlerin girildiği formun ekran görüntüsü Şekil 3.17’ de görülmektedir.



	MIN	MAX
Sıcaklık	<input type="text"/>	<input type="text"/>
Nem	<input type="text"/>	<input type="text"/>
O2	<input type="text"/>	<input type="text"/>

Şekil 3.17 Ortam Parametreleri Sınır Değer Aralıklarının Girildiği Form Ekran Görüntüsü

Ara yüz programı, vital parametrelerin insan sağlığını tehdit eden sınırların altına düştüğünde, ortam parametrelerinin de programda kullanıcıların ayarlandığı sınır değerlerin dışına çıktığında görsel ve işitsel uyarılar vermek üzere tasarlanmıştır. Şekil 3.18’ de ara yüz programında bazı ölçüm değerlerinin eşik sınırlar dışına çıktığında verdiği görsel uyarıların ekran görüntüsü görülmektedir.

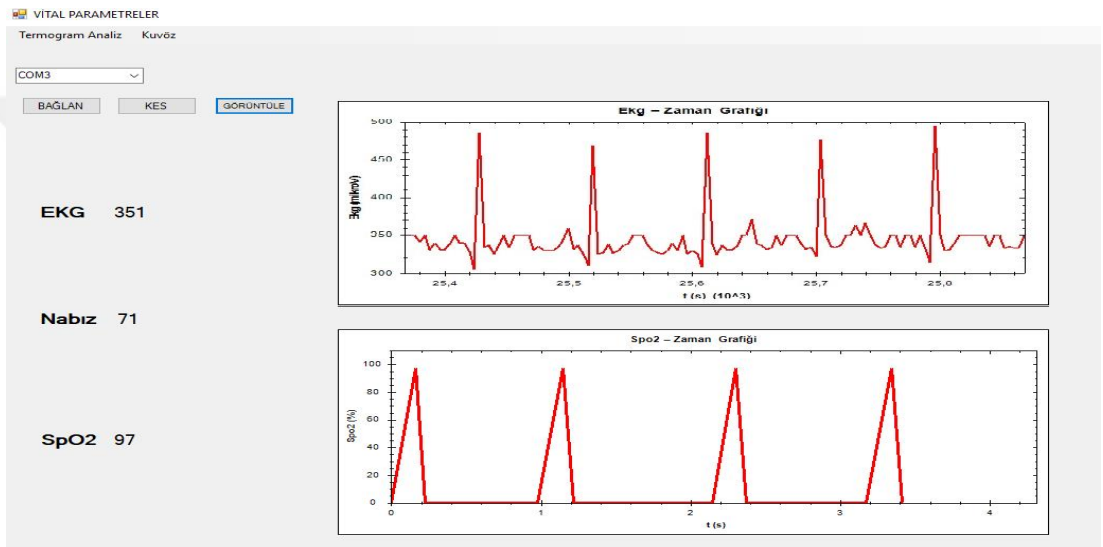
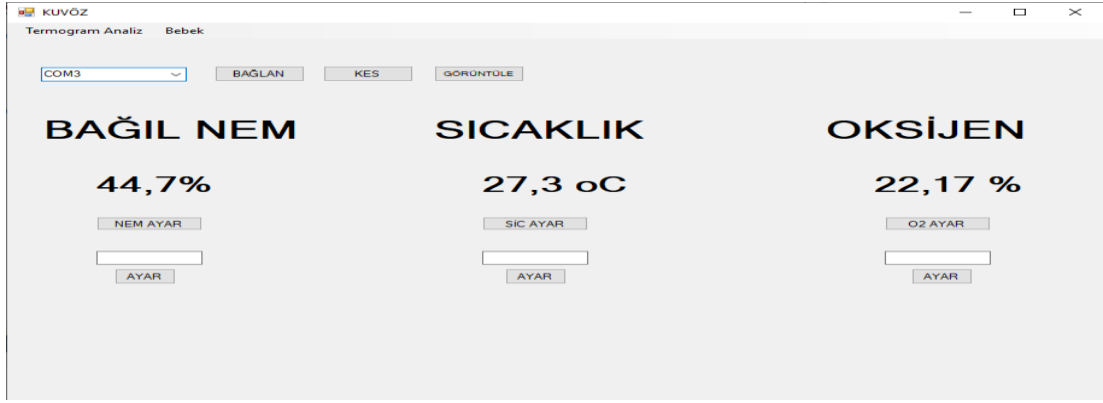


Şekil 3.18 Ara Yüz Programının Görsel Uyarı Vermesi Ekran Görüntüsü

Kullanıcılar ortam parametrelerinde değişiklik yapmak istediklerinde; değiştirilmek istenen parametrenin yanında bulunan Ayarla Butonu bastıktan sonra, bu butonun yanında bulunan richTextBox nesnesine ayarlanmak istenen değer girilir ve en son olarak OK butonuna basıldığında program Arduino kartına komut göndererek, ilgili birimleri (fan, ısıtıcı, nemlendirici, O₂ kaynağı vb.) harekete geçirir.

Ara yüz programın, yukarıda bahsedilen bütün fonksiyonları ve işlevleri için tasarlanmış algoritması Ek - 8’ de incelenebilir.

Ayrıca kuvözdeki ortam parametreleri ve prematürenin vital parametreleri ayrı ekranlarda da izlenebilir. Böylece yalnızca ortam ya da vital parametreleri takip etmek isteyen kullanıcılar daha rahat çalışabilirler. Şekil 3.19’ da ayrı ekranlarda ortam ve vital parametre verilerinin görüntülediği ekran çıktıları görülmektedir.



Şekil 3.19 Ortam ve Vital Parametreler Kullanıcı Ara Yüzü Ekran Görüntüsü



4. PREMATÜRE TERMOGRAM ANALİZİ ARA YÜZÜ

4.1 Sistemin Genel Çalışma Yapısı

Bölüm 2.7.1 Termal Kameralar ve Çalışma Prensipleri bölümünde de anlatıldığı üzere termal kameralar, nesnelere ya da canlılardaki sıcaklık farklarını göstermek için insan gözünün algılayabileceği RGB renkleri ve bu renklerin ton farklarını kullanırlar. Bu sayede nesne yada canlıların sıcaklık resimlerini dijital ortamda görüntülerler.



2.8 Görüntü işleme bölümünde bahsedildiği üzere bilgisayar ortamında her bir görüntü piksellerden oluşur. Her bir piksel için de o noktadaki parlaklık ve renk bilgisi söz konusudur. Bilgisayar ortamındaki renkli resimlere her bir piksel için, RGB renkleri ve bu renklerin yoğunluklarını gösteren 8' er bitlik, toplamda 24 bit değer atanır. Yani her bir pikselde kırmızı, yeşil ve mavi renk bilgilerini ifade eden 3 adet 8 bitlik değerler bulunur. Bu 8 bitlik veriler decimal sayı sistemine çevrildiğinde (0-255) aralığında değerler alırlar. Örneğin koyu mavi renge sahip bir pikseldeki renk bilgisi kırmızı = 5, yeşil = 10, mavi = 216 gibi değerler olabilir. Bahsi geçen bu piksel, bu üç rengin birleşiminin değer aralığı RGB = (5, 10, 216) biçiminde gösterilir. Açık kırmızı renge sahip bir pikseldeki renk bilgisi de RGB = (250, 50, 40) şeklinde gösterilebilir. Tablo 4.1' de, bu örnek renk tonları ve RGB değerleri gösterilmiştir.

Tablo 4.1 Örnek Renk Tonları ve RGB Değerleri

RENK	KIRMIZI	YEŞİL	MAVİ
	5	10	216
	250	50	40

Gri tonlu resimler ise bilgisayar ortamında 8 bitlik veriler şeklinde saklanırlar. Decimal sayı sisteminde de 0 – 255 aralığında değerler alır. 0 siyah rengi gösterirken, 255 beyaz rengi gösterir. Tablo 4.2' de örnek gri ton renkleri ve decimal değerleri gösterilmiştir.

Tablo 4.2 Örnek Gri Tonları ve Desimal Değerleri

RENK	GRİ TON DEĞERİ
	134
	46

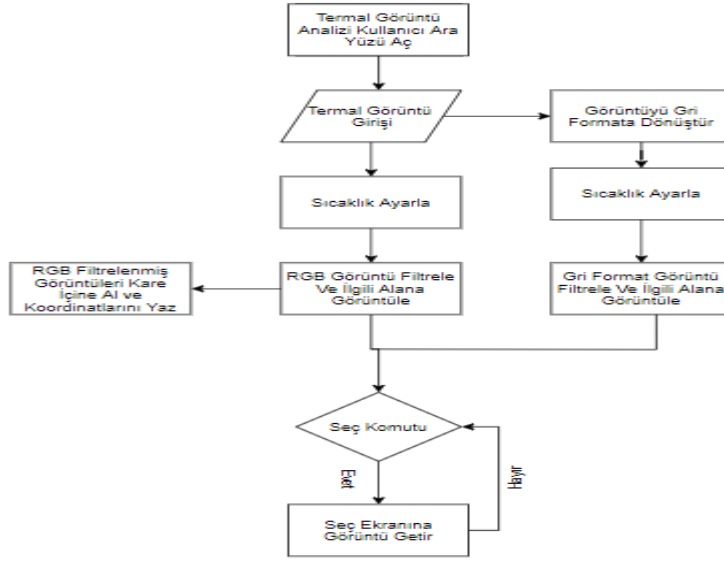
Tasarlanan bu program, termal kameralardan elde edilen görüntüler üzerinde sıcaklık farklarını ifade eden renk ve tonlarını, görüntü işleme yöntemlerinden renk filtreleme tekniklerini kullanarak, ayarlanan sıcaklık değerlerindeki lokasyonları filtreler ve diğer bölgeleri karartır. Böylece prematüre üzerinde ayarlanan sıcaklık değerindeki bölgeler tespit edilebilir.

Termal kameralar genellikle sıcaklığı düşük bölgeleri koyu, sıcaklığı yüksek bölgeleri ise açık renklerde görüntüler. Ancak termal kameralar yazılım farklılıklarına göre sıcaklık değerlerini farklı renk spektrumlarında gösterebilir. Örneğin sıcaklık ölçüm aralığı 10 - 40 °C olan termal kamera için 10 °C sıcaklığındaki bölge en koyu, 40 °C sıcaklığındaki bölge en açık renklerde görüntülenir yada sıcaklık değerlerini farklı renk tonlarında ölçeklendirebilir. Kullanılacak termal kameranın renk sıcaklık ölçeğine göre çalışmamdaki renk sıcaklık eşleşmeleri ve filtreleme algoritmaları güncellenebilir. Yapılan bu çalışmada referans alınan renk - sıcaklık ölçeği Şekil 4.1' de görülmektedir.



Şekil 4.1 Tasarlanan Programdaki Sıcaklık-Renk Ölçeği

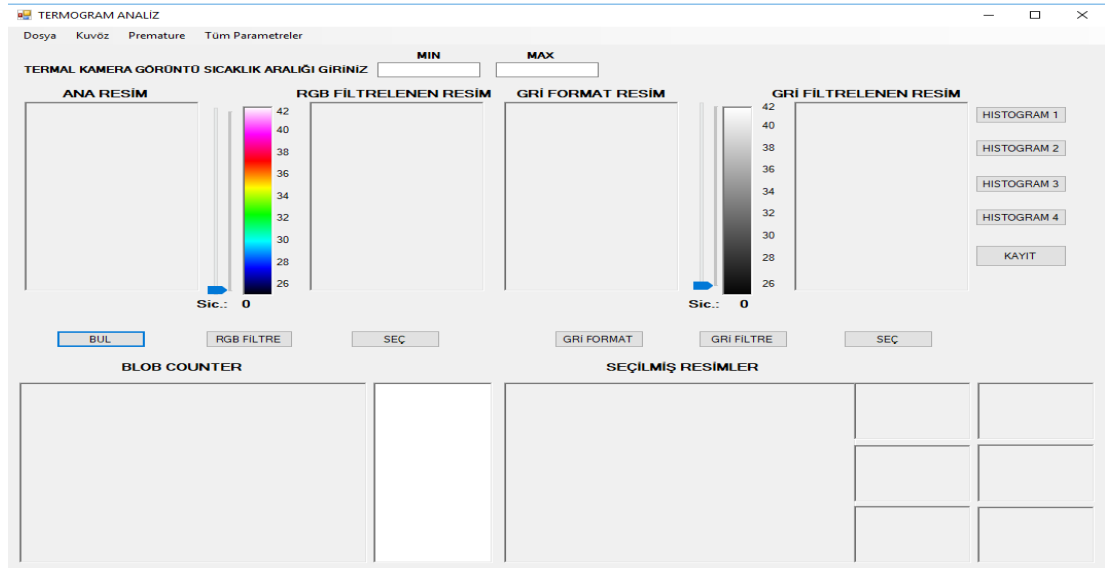
Bütün bu işlemler için C# form araçları ve C# için tasarlanmış Aforge.Net kütüphanesi kullanılmıştır. Programın işlem adımlarını gösteren akış diyagramı Şekil 4.2' de görülmektedir.



Şekil 4.2 Termogram Analizi Ara Yüz Programı Akış Diyagramı

4.2 Programı Kullanma Adımları

Program açılışındaki ekran görüntüsü Şekil 4.3’ de görülmektedir.

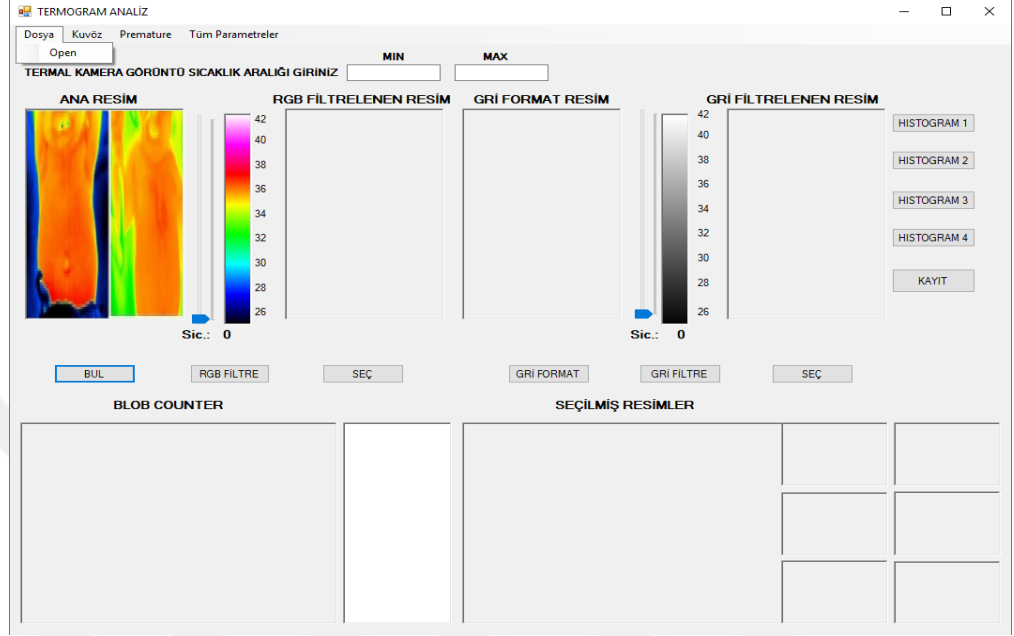


Şekil 4.3 Termaogram Analizi Ara Yüzü Açılış Ekran Görüntüsü

Programın açılışından itibaren kullanılması adım adım aşağıda anlatılmıştır.

- Program açıldıktan sonra dosya sekmesinden, bilgisayarda kayıtlı olan ve üzerinde çalışılmak istenen termal görüntü seçilerek Ana Resim bölümüne

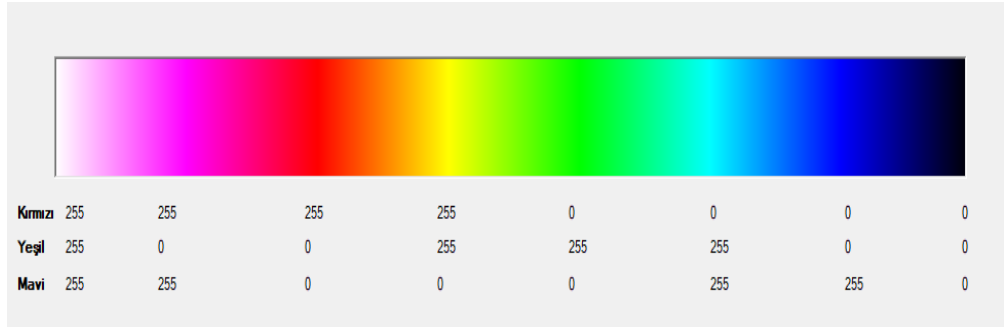
görüntülenir. Bu işlem için C# form araçlarından OpenFileDialog fonksiyonu kullanılmıştır. Şekil 4.4’ de bu işlem sonucu oluşan ekran görüntüsü görülmektedir.



Şekil 4.4 Termogram Analizi Ara Yüze Ana Resim Yüklenmesi

- Üzerinde çalışılmak istenen termal görüntü Ana Resim bölümüne görüntüledikten sonra Termal Kamera Görüntü Sıcaklık Aralığı bölümünden minimum ve maksimum değerleri seçilmelidir. Aksi takdirde program hata verecektir. Bu değerler görüntünün elde edildiği termal kameranın görüntülediği en düşük ve en yüksek sıcaklık değerleridir. Program bu değer aralıklarında renk-sıcaklık eşleşmesi yapar.

Pikseller ana ve ara renk geçişlerinde RGB = (0-255, 0-255, 0-255) aralıklarında değişen değerler alırlar. Örnek vericek olursak yeşil renge sahip bir pikselin renk bilgisi RGB =(0, 255, 0) iken sarı renge sahip pikselin renk bilgisi RGB = (255, 255, 0), kırmızı renge sahip pikselin renk bilgisi (255, 0, 0)’ dir. Buradan da anlaşılacağı üzere bütün renk geçiş ve tonlamalarında RGB renkleri 0-255 aralığında değerler almaktadır. Bu durum Şekil 4.5’ de görülmektedir.



Şekil 4.5 Renk Geçişleri ve Renk Bilgisi Desimal Değerleri

Tasarlanan algoritmada sıcaklık değerlerinin seçildiği trackBar nesnesi, bu üç rengin (kırmızı, yeşil, mavi) renk geçişlerinde aldığı 0-255 aralığındaki değerler ile eşleşmiştir. Yani trackBar nesnesindeki; 0 değeri RGB (0, 0, 0) siyah tonlu renk bilgisine, 255 değeri RGB (0, 0, 255) mavi tonlu renk bilgisine, 510 değeri RGB (0, 255, 255) cyan tonlu renk bilgisine, 765 değeri RGB (0, 255, 0) yeşil tonlu renk bilgisine, 1020 değeri RGB (255, 255, 0) sarı tonlu renk bilgisine, 1275 değeri RGB (255, 0, 0) kırmızı tonlu renk bilgisine, 1530 değeri RGB (255, 0, 255) magenta tonlu renk bilgisine, 1785 değeri ise RGB (255, 255, 255) beyaz tonlu renk bilgisine ve 0 - 1785 aralığındaki diğer değerlerde bu renk geçişlerinin farklı yoğunluklardaki renk bilgileri ile eşleşmektedir. Örneğin trackBar nesnesindeki 1350 değeri RGB (255, 0, 75) renk bilgisi ile eşleşmektedir. Şekil 4.6' da trackBar nesnesindeki değerler ile RGB renk bilgisini eşleştiren algoritma görülmektedir.

```

private void trackBar_Scroll(object sender, EventArgs e) //Sıcaklık değerinin ayarlandığı trackBar nesnesi
{
    a = trackBar7.Value; // trackBar nesnesinden seçilen değer a değişkenine atanır.

    label137.Text = a.ToString(); //trackBar nesnesinin değerini gösterir.

    if (textBox1.Text == "")
    {
        MessageBox.Show("Sıcaklık Aralığı Giriniz");
    }

    else if (textBox1.Text != "")
    {
        min = (float)(Math.Round(Convert.ToDouble(textBox1.Text)));
        max = (float)(Math.Round(Convert.ToDouble(textBox2.Text))); // textBoxlara girilen termal kameraların en düşük ve en yüksek görüntülediği sıcaklık değerleri
        sic = (1785 * min + (max - min) * a)/1785; // Sıcaklık - trackBar nesnesinin değerinin eşleştiği fonksiyon
        ort = (max - min)/8;
        label117.Text = textBox1.Text;
        label133.Text = textBox1.Text;
        label19.Text = textBox2.Text;
        label131.Text = textBox2.Text;
        label116.Text = (min + ort).ToString();
        label11.Text = (min + ort).ToString();
        label115.Text = (min + 2*ort).ToString();
        label125.Text = (min + 2 * ort).ToString();
        label114.Text = (min + 3*ort).ToString();
        label126.Text = (min + 3 * ort).ToString();
        label113.Text = (min + 4 * ort).ToString();
        label127.Text = (min + 4 * ort).ToString(); // textBoxlara girilen min ve mak. sıcaklık değerlerine göre ölçek aralığının yeniden hesaplanması
        label112.Text = (min + 5*ort).ToString();
        label128.Text = (min + 5 * ort).ToString();
        label111.Text = (min + 6*ort).ToString();
        label129.Text = (min + 6 * ort).ToString();
        label110.Text = (min + 7*ort).ToString();
        label130.Text = (min + 7 * ort).ToString();

        label123.Text = sic.ToString("N1") + " oC";
        // q = mavi renk yoğunluğunu
        // u = yeşil renk yoğunluğunu
        // w = kırmızı renk yoğunluğunu gösterir.

        if (a <= 255)
        {
            q = a;
            u = 0;
            w = 0;
        }
        else if (a >= 256 && a <= 510)
        {
            q = 255;
            u = a-255;
            w = 0;
        }
        else if (a >= 511 && a <= 765)
        {
            q = 765-a;
            u = 255;
            w = 0;
        }
        else if (a > 765 && a <= 1020)
        {
            q = 0;
            u = 255;
            w = a-765;
        }
        else if (a > 1020 && a <= 1275)
        {
            q = 0;
            u = 1275-a;
            w = 255;
        }
        else if (a > 1275 && a <= 1530)
        {
            q = a-1275;
            u = 0;
            w = 255;
        }
        else if (a > 1530 && a <= 1785)
        {
            q = 255;
            u = a-1530;
            w = 255;
        }
    }
}

```

Şekil 4.6 TrackBar Nesnesi ile RGB Renk Bilgisini Eşleştiren Algoritma

Termal kameralar sıcaklık artışlarını koyu renklere açık renklere doğru geçiş yaparak görüntüledikleri için, renk – sıcaklık eşleşmeleri doğrusal bir fonksiyonla yapılmıştır. Sıcaklık değerini gösteren P(x) fonksiyonu (4.1)' de görülmektedir.

$$P(x) = \frac{(1785 \times \min) + (\max - \min) \times a}{1785} \quad (4.1)$$

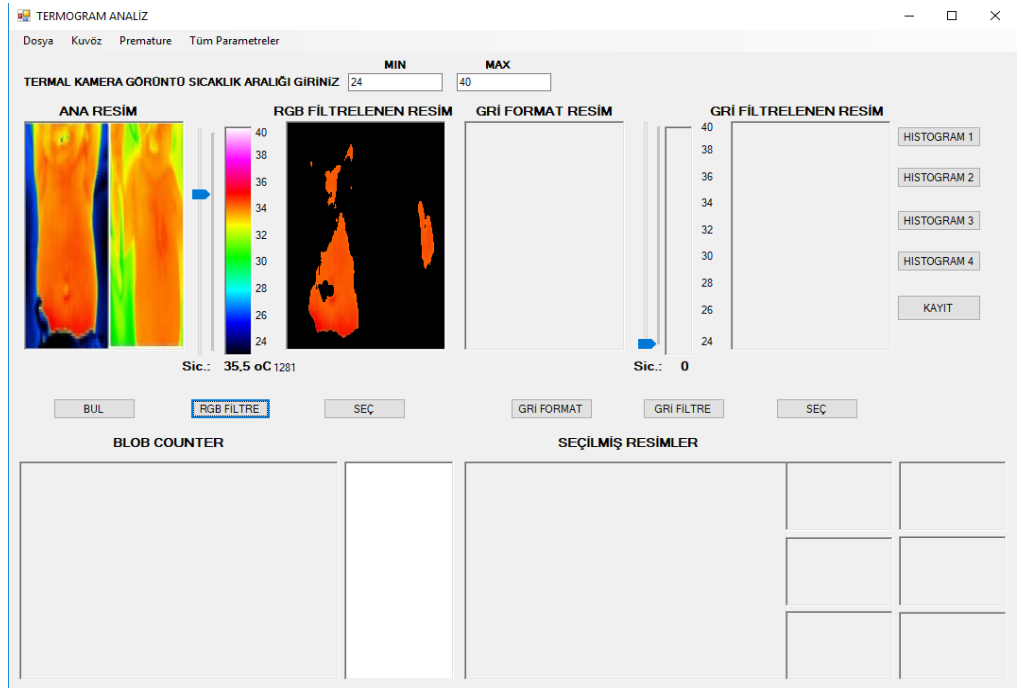
Min : textBox' a girilen termal kamera en düşük sıcaklık değeri

Max : textBox' a girilen termal kamera en düşük sıcaklık değeri

a : trackBar nesnesindeki değeri

Bu fonksiyon sayesinde, trackBar nesnesindeki 0 - 1785 aralığındaki değerler ile programa girilen termal kamera görüntü sıcaklık aralığındaki min ve max değişkenine göre sıcaklık değerleri eşleşir. Örnek verecek olursak 24 – 40 °C görüntü sıcaklık aralığında, trackBar nesnesinden 35,6 °C sıcaklık değeri seçildiğinde P(x) fonksiyonuna göre a değişkeni 1289 değerini alır. Bu da RGB (255, 0, 14) renk bilgisine sahip pikselli ifade etmektedir.

- Ana resim bölümünde görüntülenen resimler trackBar nesnesinden istenilen sıcaklık ayarı yapıldıktan sonra RGB Filtre butonuna basıldığında program, seçilen sıcaklık değerindeki renk bilgisini merkez alarak, ayarlanan yarıçap aralığındaki renk bilgilerine sahip pikselli filtreleyerek diğer pikselli karartır. Bu işlem Aforge.Net kütüphanesindeki EuclideanColorFiltering sınıfı kullanılarak yapılır. Yapılan bu işlemin ekran görüntüsü Şekil 4.7' de görülmektedir.



Şekil 4.7 Ana Resim RGB Filtreleme Ekran Görüntüsü

RGB filtreleme işlemi için tasarlanan algoritma Şekil 4.8' de görülmektedir.

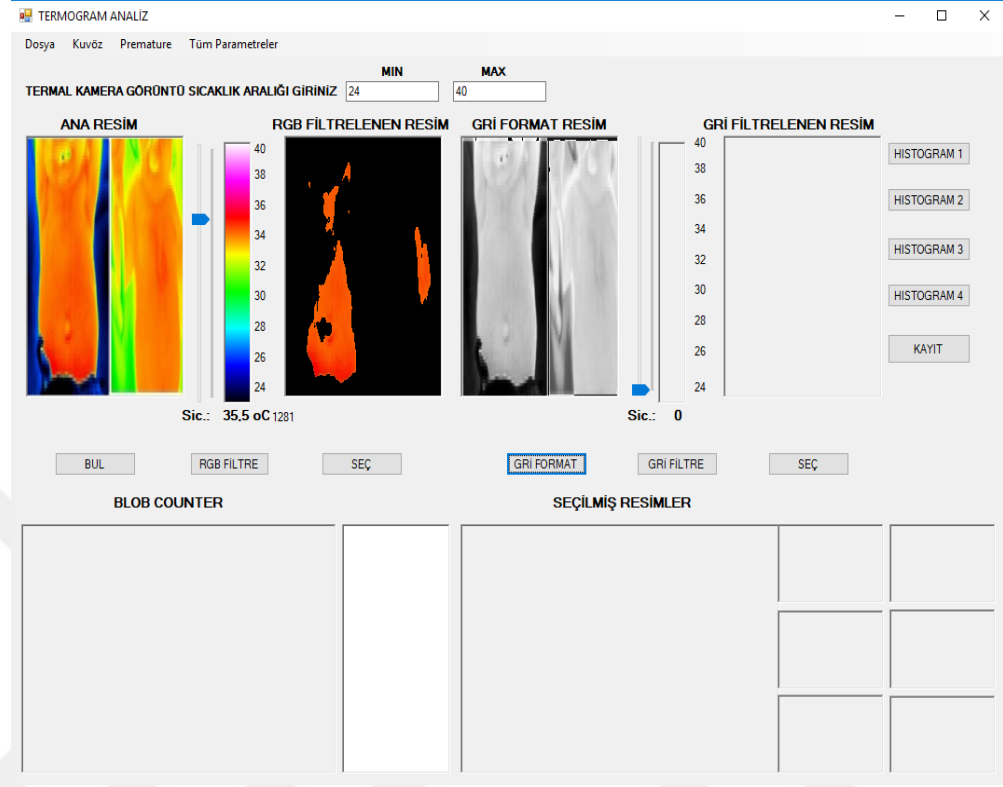
```
private void button2_Click(object sender, EventArgs e) //RGB Filtrele Butonu
{
    if ((Bitmap)pictureBox1.Image != null)
    {
        // filtre oluşturuluyor
        EuclideanColorFiltering filtre1 = new EuclideanColorFiltering();
        // merkez alınıp filtrelenecek renk bilgisindeki pikseller ayarlanıyor.
        filtre1.CenterColor = new RGB(Color.FromArgb(w, u, q));
        // filtrelenecek piksel küresinin yarıçapı
        filtre1.Radius = 100;
        // filtre uygulanıyor.
        pictureBox2.Image = filtre1.Apply((Bitmap)pictureBox1.Image);
    }
}
```

Şekil 4.8 RGB Filtreleme İşlemi Algoritması

Bu algorithmada da görüldüğü üzere Ana Resim bölümüne resim yüklenmediğinde program hata verecektir. Merkez alınacak renk değeri seçildikten sonra RGB renk küresindeki 100 birim yarıçap olarak ayarlanmıştır.

- Termal kameralar tarafından nesne ve canlılaraki sıcaklık farklarıyla oluşturulmuş RGB formatındaki renkli resimler gri formata çevrilerek sıcaklık analizleri siyah – beyaz tonlarında yapılabilir. Daha önceden bahsedildiği üzere, bilgisayar ortamında renkli resimler 24 bit renk bilgisine sahipken, gri formattaki resimler 8 bit renk bilgisinde saklanır. Renkli bir resmi gri formata dönüştürmek için her bir pikseldeki kırmızı, yeşil ve mavi renk değerlerinin belli bir oranda ölçeklendirilip, toplanmasıyla 8 bit yani 0 – 255 decimal aralığında renk bilgisi elde edilmelidir. Örnek verecek olursak bu üç rengin toplamını üçe bölecek olursak gri ton resim elde edebiliriz. Ancak bu ölçeklendirme resimdeki farklı renkleri ayırt etme hassasiyetini ortadan kaldırır. Bu yüzden kullanım amacına göre renkli resimleri gri tona çevirmek için farklı formüller kullanılır. Yapılan çalışmada mavi renkten kırmızı renge geçiş sıcaklık artışını ifade ettiği için $RGB = (0,6 \times \text{Kırmızı} + 0,3 \times \text{Yeşil} + 0,1 \times \text{Mavi})$ ölçeklendirmesi kullanılmıştır. Ana Resim bölümündeki görüntüyü gri formata dönüştürmek için Aforge.Net kütüphanesinden Grayscale sınıfı kullanılmıştır. Bu sınıf için de ölçek aralığı $RGB = (0,6 \ 0,3 \ 0,1)$ girilmiştir. Gri formata çevrilen resim Gri Format Resim

bölümüne görüntülenir. Bu işlemin ekran görüntüsü Şekil 4.9’ da görülmektedir.



Şekil 4.9 Ana Resim Gri Format Dönüşümü Ekran Görüntüsü

Ana Resim bölümüne resim yüklenmediği takdirde program hata verecektir. Gri Formata Dönüştürme işleme ait algoritma Şekil 4.10’ da görülmektedir.

```
private void button7_Click(object sender, EventArgs e) // Gri Formata Çevir Butonu
{
    if ((Bitmap)pictureBox1.Image != null)
    {
        // filtre oluşturuluyor.
        Grayscale filter1 = new Grayscale(0.60, 0.30, 0.10); // Renk ölçek oranları
        // filtre uygulanıyor.
        pictureBox12.Image = filter1.Apply((Bitmap)pictureBox1.Image);
    }
    else MessageBox.Show("Resim Yükleyiniz");
}
```

Şekil 4.10 Gri Formata Dönüştürme Algoritması

Gri formata dönüştürülen termal görüntülerde siyah yoğunluklu tonlardan beyaz yoğunluklu tonlara geçişte sıcaklık artar. Tasarlanan programda 0 – 255 değer aralığındaki trackBar2 nesnesi ile 0 – 255 aralığındaki gri renk tonları eşleşmiştir. Termal Kamera görüntü sıcaklık aralığı en düşük ve en

yüksek sıcaklık değerleri girildiğinde program bu aralıktaki sıcaklık değerleri ile trackBar2 nesnesindeki 0 – 255 ölçek aralığını eşleştirir. Yapılan bu işlem RGB filtreleme işleminde olduğu gibi doğrusal bir fonksiyonla yapılır. Sıcaklık bilgisini görüntüleyen P(x) fonksiyonu Denklem (4.2)' de görülmektedir.

$$P(x) = \frac{(255 \times \min) + (\max - \min) \times b}{255} \quad (4.2)$$

Min : textBox' a girilen termal kamera en düşük sıcaklık değeri

Max : textBox' a girilen termal kamera en yüksek sıcaklık değeri

b : trackBar2 nesnesindeki değer

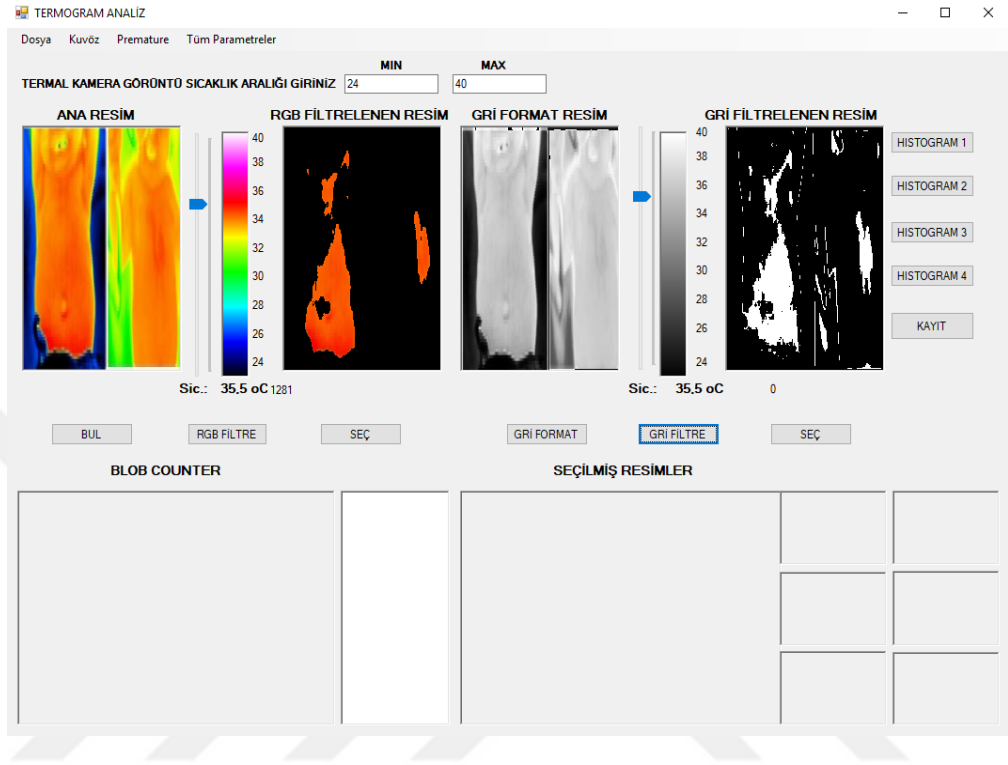
Örnek verecek olursak termal kamera görüntü sıcaklık aralığı 24 – 40 °C girildiğinde, 35,5°C değeri ayarlanırsa, trackBar2 nesnesi 184 değerinde olacaktır. Bu durumda 184 gri renk tonundaki pikseller belirli bir tolerans aralığında filtrelenecektir. Bu işleme ait programdaki algoritmalar Şekil 4.11' de görülmektedir.

```
private void trackBar2_Scroll1(object sender, EventArgs e) // Gri Format Sıcaklık değerinin
seçildiği trackbar nesnesi
{
    min = (float)(Math.Round(Convert.ToDouble(textBox1.Text)));
    max = (float)(Math.Round(Convert.ToDouble(textBox2.Text))); // textBoxlara girilen
termal kameraların en düşük ve en yüksek görüntülediği sıcaklık değerleri
    int b = trackBar2.Value; // trackBar2 nesnesindeki ayarlanan değer
    sic1 = ((max-min)*b+(255*min))/255; // Sıcaklık trackBar2 nesnesinin eşleştiği fonksiyon
    label141.Text = b.ToString();
    label132.Text = sic1.ToString("M1") + " oC";
}
```

Şekil 4.11 TrackBar2 Nesnesi ile Gri Format Renk Bilgisini Eşleştiren Algoritma

- Gri Format Resim bölümünde görüntülenen resimler trackBar2 nesnesinden istenilen sıcaklık ayarı yapıldıktan sonra Gri Filtre butonuna basıldığında, program seçilen sıcaklık değerindeki renk bilgisine (belirli bir tolerans aralığında) sahip pikselleri beyaz, yani 255 renk değerine dönüştürerek, diğer pikselleri siyah, yani 0 renk değerine çevirerek karartır. Böylece renk bilgisi seçilen pikseller Gri Filtrelenen Resim alanına görüntülenir. Yapılan bu

işlemin ekran görüntüsü Şekil 4.12’ de ve algoritmaları Şekil 4.13’ de görülmektedir.



Şekil 4.12 Gri Format Filtreleme İşlemi Ekran Görüntüsü

```

private void button6_Click(object sender, EventArgs e) //Gri Format Filtrele butonu
{
    if ((Bitmap)pictureBox12.Image != null) //Gri Format Resim yüklendi ise
        filtreleme işlemleri yapılabilir. Aksi takdirde program hata verecektir.
        {
            int b = trackBar2.Value; // trackBar2' den seçilen eşik değeri
            Color OkunanRenk, DonusenRenk;
            int R = 0, G = 0, B = 0;
            Bitmap resim = new Bitmap(pictureBox12.Image);
            int i = 0, j = 0;
            for (int x = 0; x < resim.Width; x++)
            {
                j = 0;
                for (int y = 0; y < resim.Height; y++)
                {
                    OkunanRenk = resim.GetPixel(x, y);
                    if (b - 10 <= OkunanRenk.R && OkunanRenk.R <= b + 10)
                        R = 255;
                    else
                        R = 0;
                    if (OkunanRenk.G >= b-10 && OkunanRenk.G <= b+10) //trackBar2
                        nesnesinden seçilen eşik değere göre, belli bir tolerans aralığında bütün
                        pikseller aranarak filtreleme işlemi yapılıyor.
                        G = 255;
                    else
                        G = 0;
                    if (OkunanRenk.B >= b-10 && OkunanRenk.B <= b+10)
                        B = 255;
                    else
                        B = 0;
                    DonusenRenk = Color.FromArgb(R,G,B);
                    resim.SetPixel(i, j, DonusenRenk);

                    j++;
                }
                i++;
            }

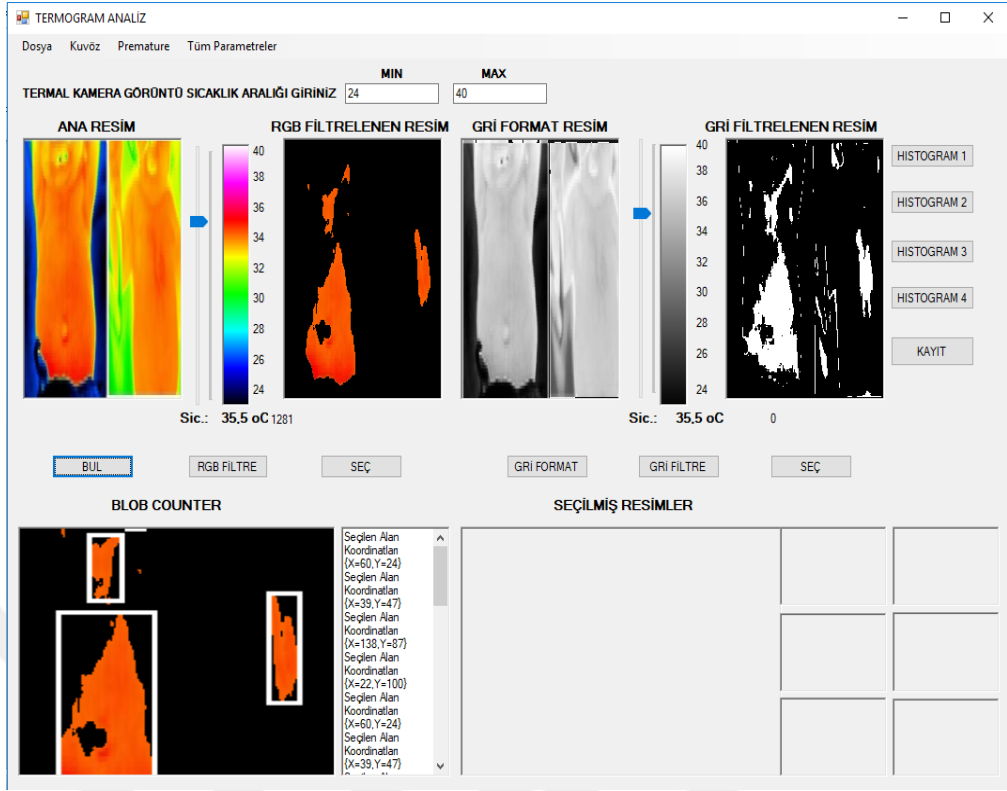
            pictureBox14.Image = resim; Filtrelenen resim Gri
            Filtrelenmiş Resim Bölümüne görüntülenir.
        }
        else MessageBox.Show("Resim Yükleyiniz");
    }
}

```

Şekil 4.13 Gri Format Filtreleme İşlemi Algoritması

- Tasarlanan programda “BUL” butonu ile RGB Filtrenen Resim bölümündeki filtrelenmiş pikseller kare içine alınarak, Blob Counter bölümüne görüntülenir. Yapılan bu işlemde Aforge.Net kütüphanesindeki BlobCounter fonksiyonu kullanılmıştır. Kare içine alınan bu bölümlerin koordinatları da richTextBox bölümüne yazdırılır. Böylece prematüre termogramları üzerinde aranan sıcaklık değerlerinin sınırları ve lokasyonları tespit edilebilir. Bu işlemin termogramlar üzerinde analiz ve raporlama kolaylığı sağlanacağı düşünülmektedir.

Yapılan bu işlemin ekran görüntüsü Şekil 4.14’ de görülmektedir.



Şekil 4.14 Blob Counter İşlemi Ekran Görüntüsü

RGB Filtrelenmiş Resim Bölümüne görüntü yüklenmediği takdirde program hata verecektir.

Yapılan bu işlem için yazılmış algoritmalar Şekil 4.15' de görülmektedir.

```

private void button1_Click(object sender, EventArgs e) //Bul butonuna basılıyor.
{
    if ((Bitmap)pictureBox2.Image != null) //RGB Filtre bölümünde görüntü olmadıkça program
    hata verecektir.
    {
        //Ana Resim Bölümündeki görüntü filtrelenerek Blob Counter alanına da görüntülenir.

        // filtre oluşturuluyor.
        EuclideanColorFiltering filtre = new EuclideanColorFiltering();
        // Merkez pikseller ve RGB küresi yarıçapı belirleniyor.
        filtre.CenterColor = new RGB(Color.FromArgb(w, u, q));
        filtre.Radius = 100;
        // filtre uygulanıyor.
        pictureBox2.Image = filtre.Apply((Bitmap)pictureBox1.Image);
        pictureBox11.Image = filtre.Apply((Bitmap)pictureBox1.Image);

        //Blob Counter işlemi için algoritmalar
        BlobCounter blobCounter = new BlobCounter();

        blobCounter.MinWidth = 5;
        blobCounter.MinHeight = 5;
        blobCounter.FilterBlobs = true;
        blobCounter.ObjectsOrder = ObjectsOrder.Size;
        blobCounter.ProcessImage((Bitmap)pictureBox2.Image);
        Rectangle[] rects = blobCounter.GetObjectsRectangles();
        foreach (Rectangle recs in rects)
            if (rects.Length > 0)
            {
                foreach (Rectangle objectRect in rects)
                {
                    Graphics g = Graphics.FromImage((Bitmap)pictureBox11.Image);
                    using (Pen pen = new Pen(Color.FromArgb(255, 255, 255), 3))
                    {
                        g.DrawRectangle(pen, objectRect);
                    }

                    int objectX = objectRect.X + (objectRect.Width / 2);
                    int objectY = objectRect.Y + (objectRect.Height / 2);

                    g.Dispose();

                    //Filtrelenmiş alanların koordinat bilgileri tespit edilerek richTextBox nesnesine yazdırılıyor.

                    if (button1.Enabled)
                    {
                        this.Invoke((MethodInvoker)delegate
                        {
                            richTextBox1.Text = "Seçilen Alan Koordinatları" + "\n" +
                            objectRect.Location.ToString() + "\n" + richTextBox1.Text + "\n";
                        });
                    }
                }
            }

        //RGB filtrelenmiş ve kare içine alınmış görüntü, BlobCounter bölümündeki pictureBox nesnesinin
        ölçülerine uygun hale getiriliyor.

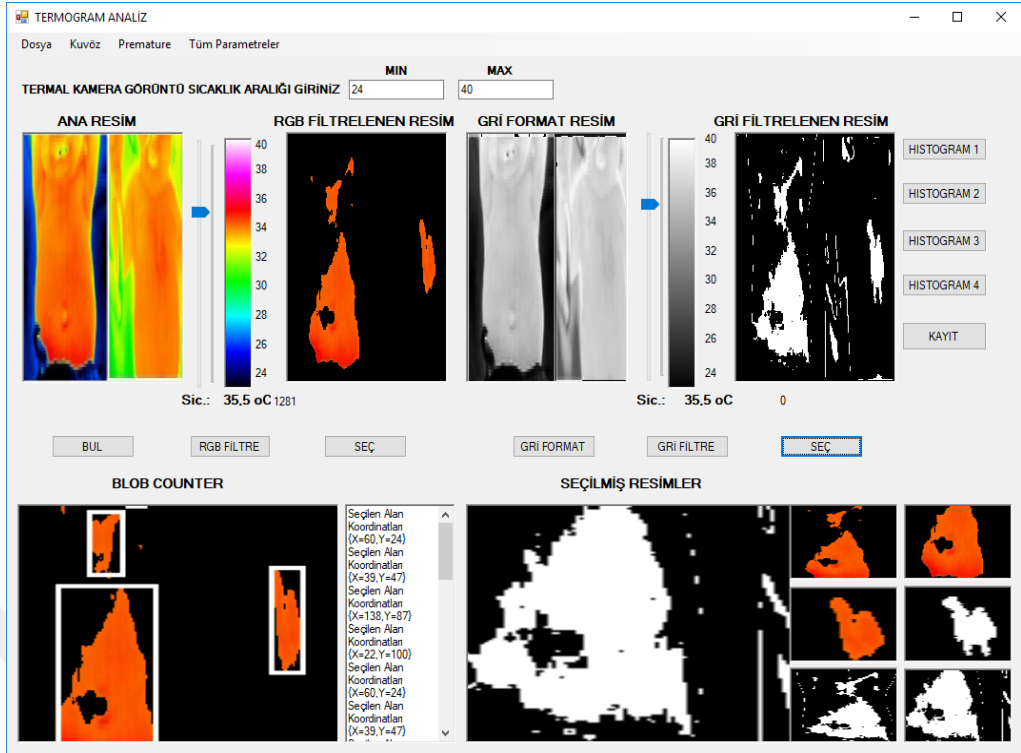
        pictureBox11.SizeMode = PictureBoxSizeMode.StretchImage;
        Bitmap orjinall = new Bitmap(pictureBox11.Image, 338, 362);
        pictureBox11.Image = orjinall;
        pictureBox11.SizeMode = PictureBoxSizeMode.CenterImage;
    }

    else MessageBox.Show("Resim Yükleyiniz");
}

```

Şekil 4.15 BlobCounter İşlemi Algoritmaları

- RGB ve Gri formatta filtrelenmiş görüntüler üzerinde çalışılacak veya not alınacak lokasyonalar, bu görüntüler üzerinde seçilerek “SEÇ” butonu ile Seçilmiş Resimler alanındaki küçük kutucuklarda görüntülenir. Bu küçük kutucuklardaki resimlerin üzerine tıklandığında da, bu kutucuktaki görüntü büyütülerek yan taraftaki büyük kutucukta görüntülenir. Bu sayede kullanıcılara analiz ve çalışma kolaylığı sağlanması amaçlanmaktadır. Bu işleme ait ekran görüntüsü Şekil 4.16’ da görülmektedir.



Şekil 4.16 Seçilmiş Alanlar Bölümüne Resim Yükleme

- Tasarlanan programda “Kayıt” butonu ile çalışılmış (filtrelenmiş) görüntüler bilgisayara kaydedilebilir. Ayrıca “Histogram” butonları ile programda görüntülenen Ana ve Filtrelenmiş resimlerin renk ve ton dağılımları grafikleştirilir.

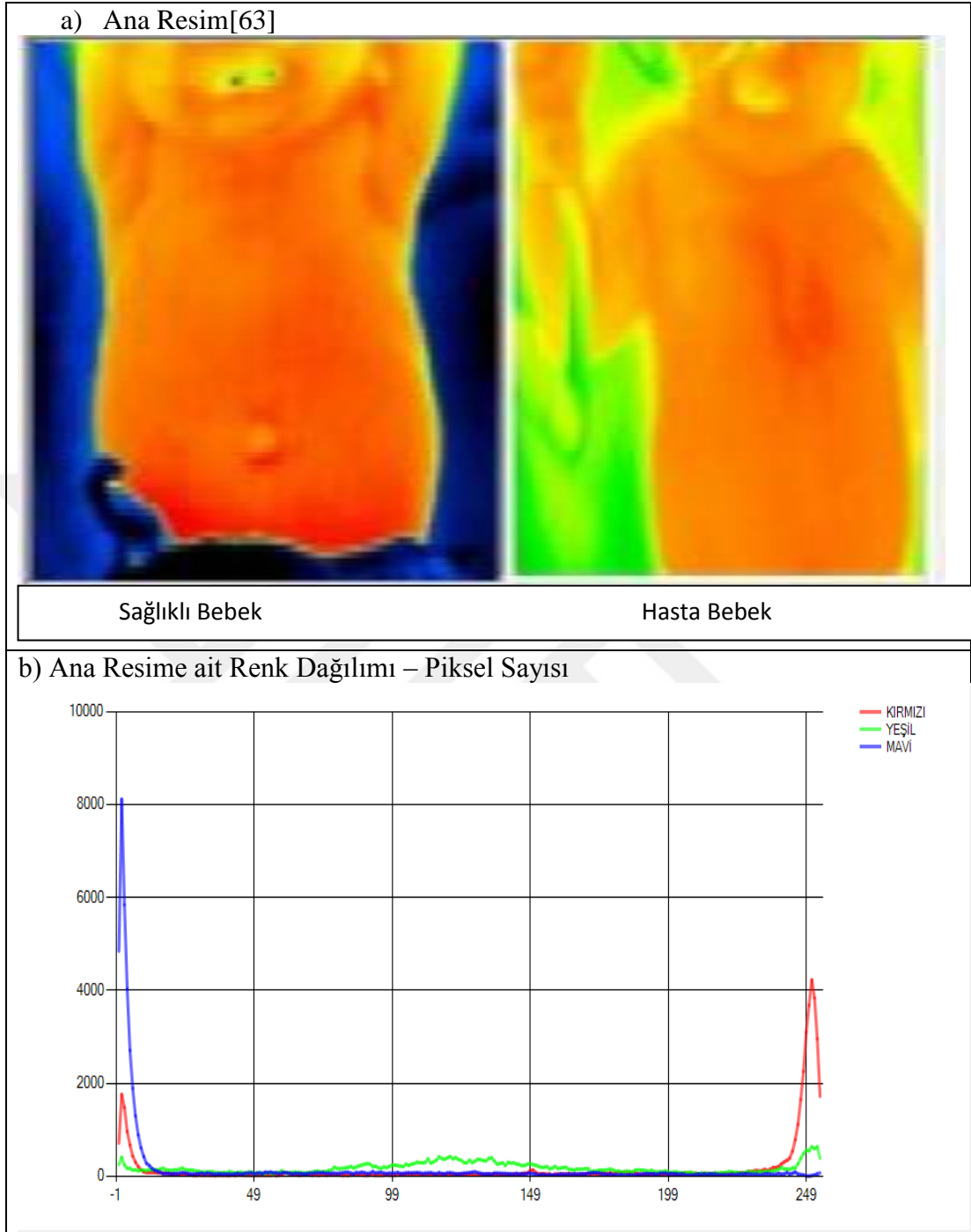
Premature Termogram Analizi programına ait bütün algoritmalar Ek-7’ de incelenebilir.

5.BULGULAR

Tasarlanan termogram sıcaklık analizi ara yüz programında, bu alanda daha önce yapılmış çalışmalarda kullanılan hem prematüre hem de yetişkin termal görüntüleri üzerinde çalışıldığında başarılı sonuçlar elde edildi. Termogramlar üzerinde ayarlanan renk-sıcaklık ölçeğine göre, istenilen sıcaklık değerlerinin bulunduğu lokasyonların filtreleme işlemleri sonucu ortaya çıktığı görüldü. Çalışılan bazı termal görüntüler, bu görüntüler üzerinden elde edilen filtrelenmiş görüntüler ve bu görüntülere ait renk ton - piksel sayısı dağılımlarını gösteren histogram grafikleri aşağıda gösterilmiştir.

5.1 1. Çalışma

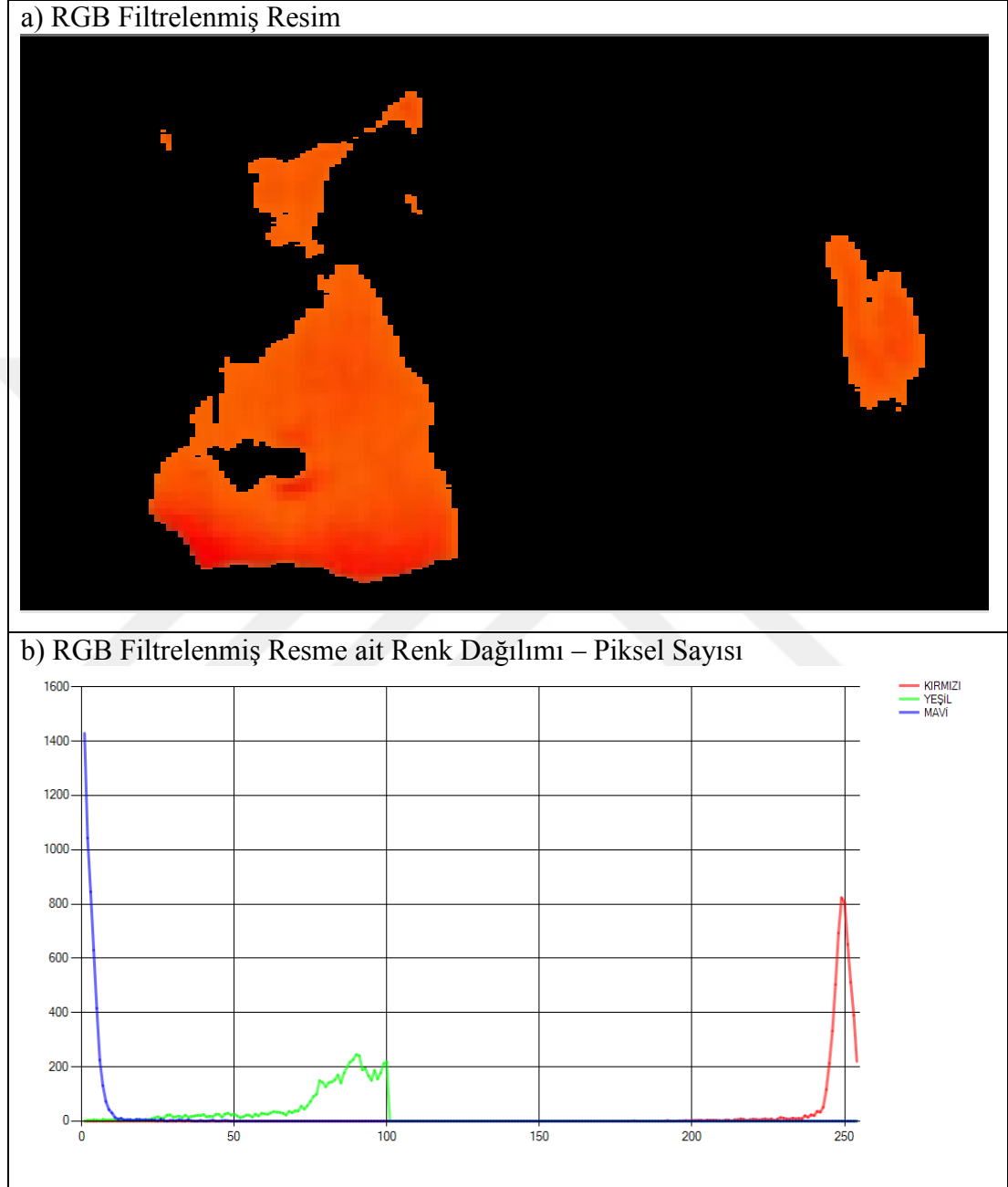
Duygu SAVAŞÇI ve Murat CEYLAN tarafından hazırlanan “Yenidoğan Yoğun Bakım Ünitelerinde Termal Görüntüleme Analizi” makalesinden alınan prematüre termal görüntüsü üzerinde çalışılmıştır [63]. Sol taraftaki bebek sağlıklı, diğer bebek kalp hastasıdır. Programa, bu çalışmada kullanılan termal kameranın görüntü sıcaklık aralığı (24 – 40 °C) girilmiştir. Şekil 5.1’ de Programda kullanılan ana resim ve bu resme ait renk dağılımları görülmektedir.



Şekil 5.1 1. Çalışmaya ait Ana Görüntü ve Renk Dağılımlarını Gösteren Grafik

Bu resim üzerinde, 35,4°C sıcaklık değerinde programın RGB Filtreleme fonksiyonu çalıştırıldığında, ana görüntü üzerinde bu değere karşılık gelen renk değerine sahip pikseller merkez alınarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerine sahip pikseller filtrelenmiştir. Program bu yarıçap aralığı dışında kalan diğer pikselleri karartarak elde edilen görüntüyü RGB Filtrelenen

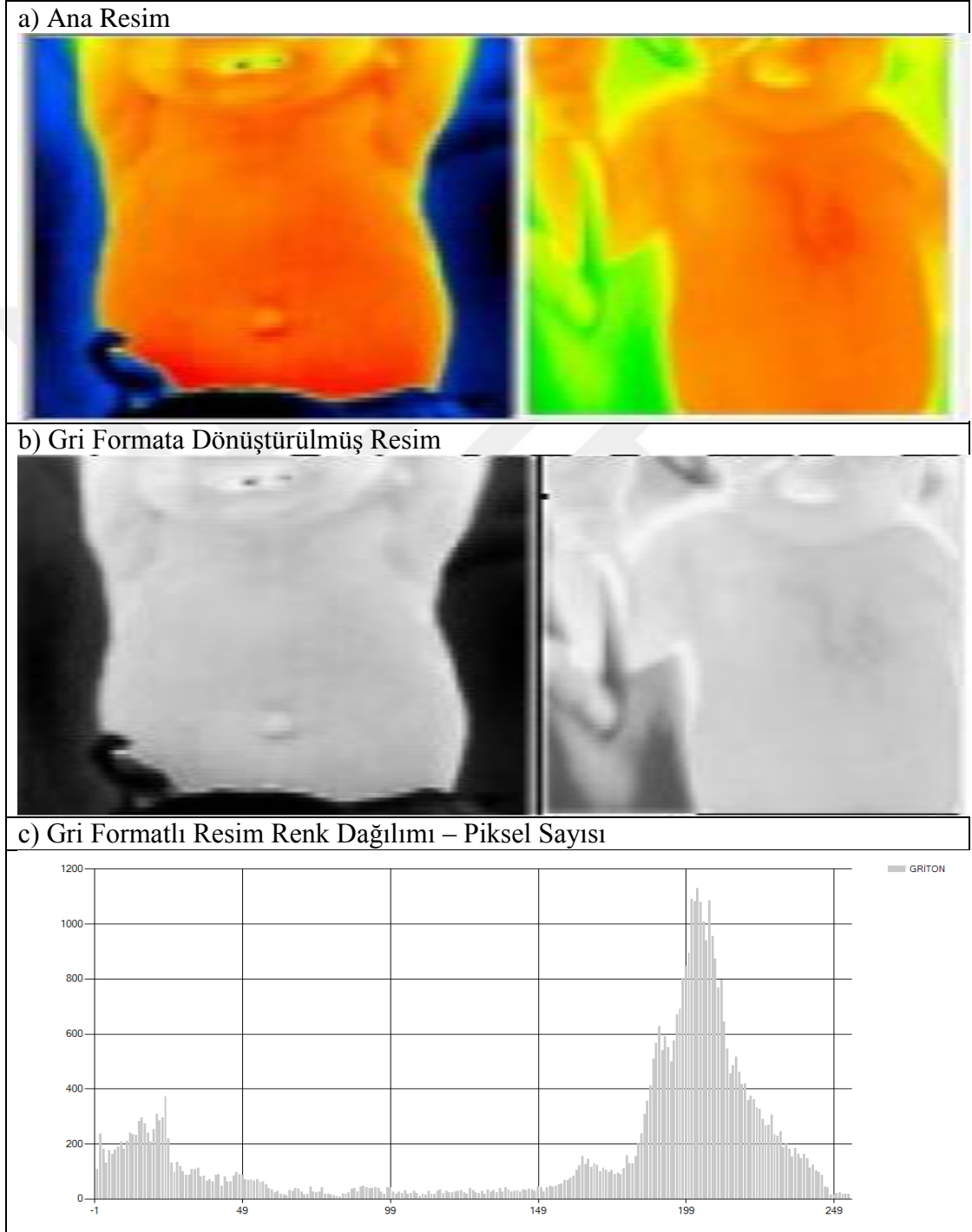
Resim alanına getirir. Programın RGB Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.2' de görülmektedir.



Şekil 5.2 35,4°C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

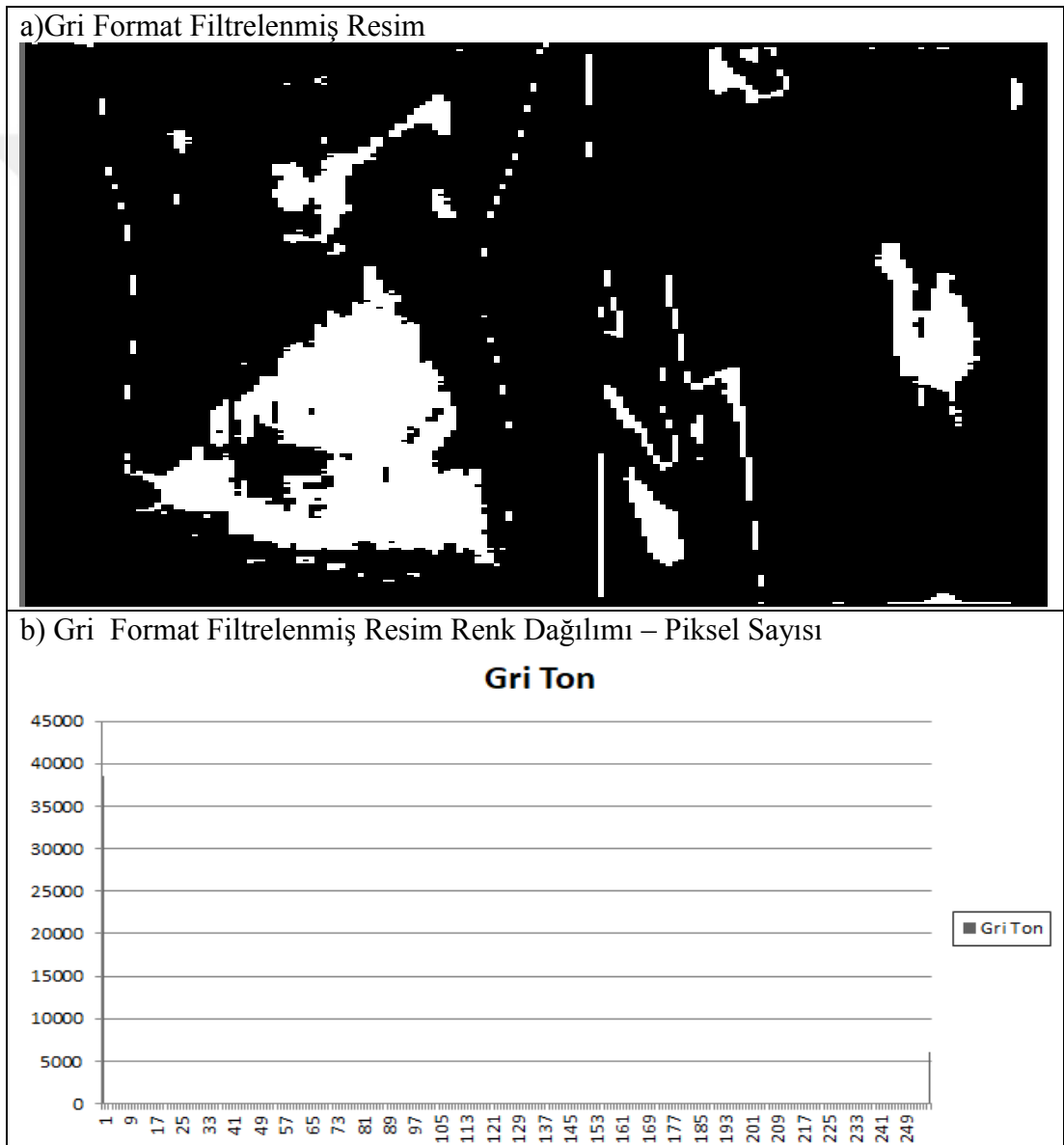
Şekil 5.2' de renk dağılım grafiği incelendiğinde, programın 35,4°C sıcaklık değerine karşılık gelen renk değerine sahip pikselleri merkez alarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerlerine sahip pikselleri filtrelediği, diğer pikselleri kararttığı görülmektedir.

Şekil 5.3’ de ana görüntünün, program üzerinde RGB (0.6, 0.3, 0.1) ölçek oranlarında gri formata dönüştürülerek, Gri Format Resim Alanında görüntülenen resmin ekran görüntüsü ve renk dağılımı görülmektedir.



Şekil 5.3 1. Çalışmaya ait Ana Görüntü, Gri Formata Çevrilmiş Görüntü ve Gri Format Resim Renk Dağılımı Grafiği

Elde edilen gri format görüntü üzerinde program, renk – ton (0 – 255) aralığında sıcaklık eşleştirmesi yapar. Gri formatlı görüntü üzerinde 35,4 °C sıcaklık değerinde Gri Format Filtreleme işlemi yapıldığında, program ana görüntü üzerinde bu değere karşılık gelen renk bilgisine sahip pikselleri filtreleyerek, (+/- 10 birim tolerans aralığında) diğer pikselleri karartır ve Gri Format Filtrelenen Resim alanına görüntüler. Programın Gri Format Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.4’ de görülmektedir.



Şekil 5.4 35,4°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

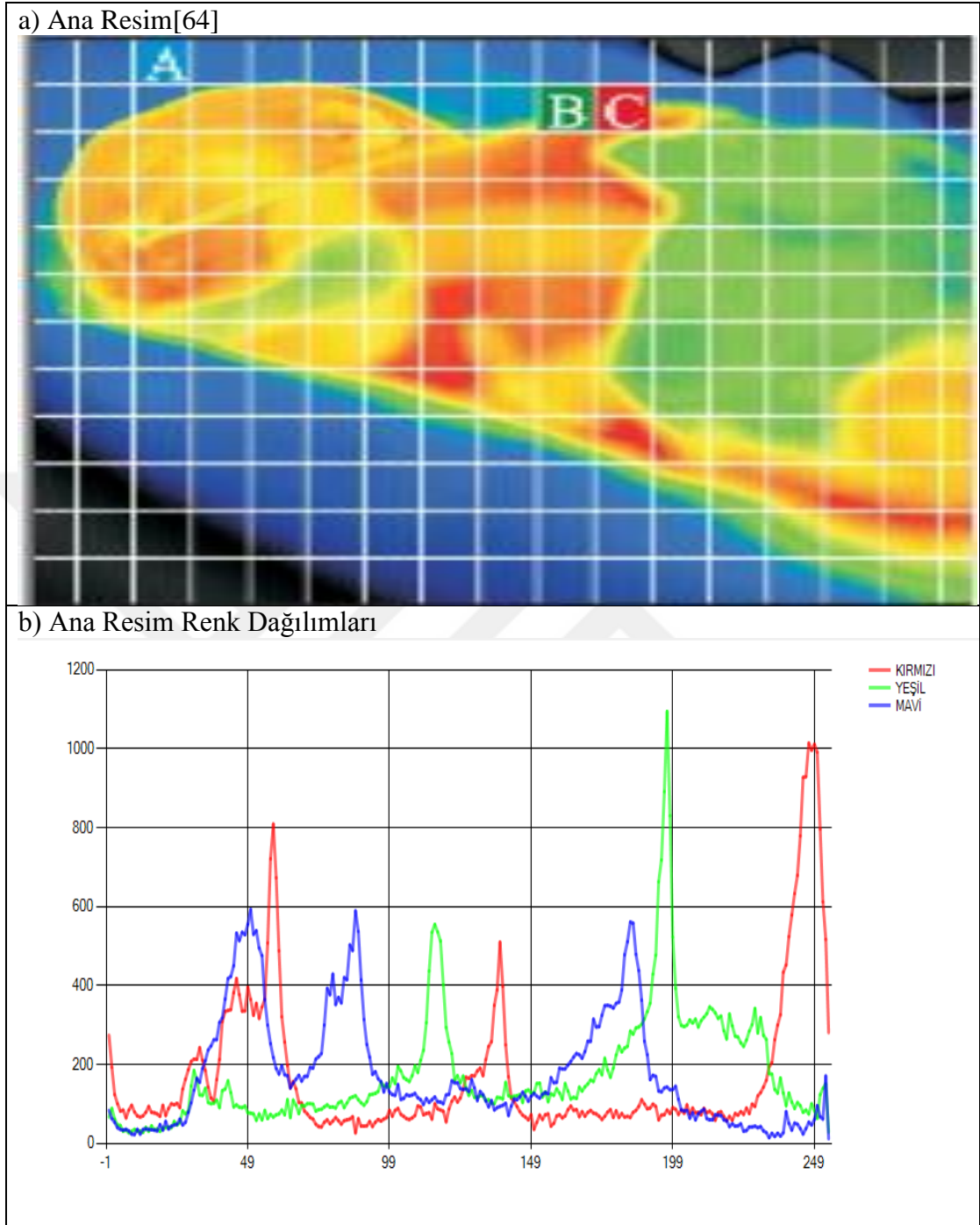
Şekil 5.4' de filtrelenen görüntünün renk dağılımlarını gösteren grafik incelendiğinde 35,4°C sıcaklık değerine karşılık gelen piksellerin 255 renk değeri, yani beyaz renkte görüntülediği diğer piksellerin ise 0 renk değeri yani, siyah renkte görüntülediği görülmektedir.

Hem renkli hem de gri format ekran çıktıları incelendiğinde, aynı sıcaklık değerinde sağlıklı bebekte termal simetri görülürken, hasta bebekte termal simetriye rastlanmamıştır.

5.2 2.Çalışma

Carina Barbosa PEREIRA ve arkadaşları tarafından hazırlanan “Noncontact Monitoring of Respiratory Rate in Newborn Infants Using Thermal Imaging” isimli makalede, prematürelerin solunum hızının vücuda yerleştirilen sensör ya da elektrotlar dışında termal video ile temazsız ölçümü uygulaması yapılmıştır. Tasarlanan Premaüre Termogram Analizi programında bu makalede verilen termal görüntü üzerine çalışılmıştır[64].

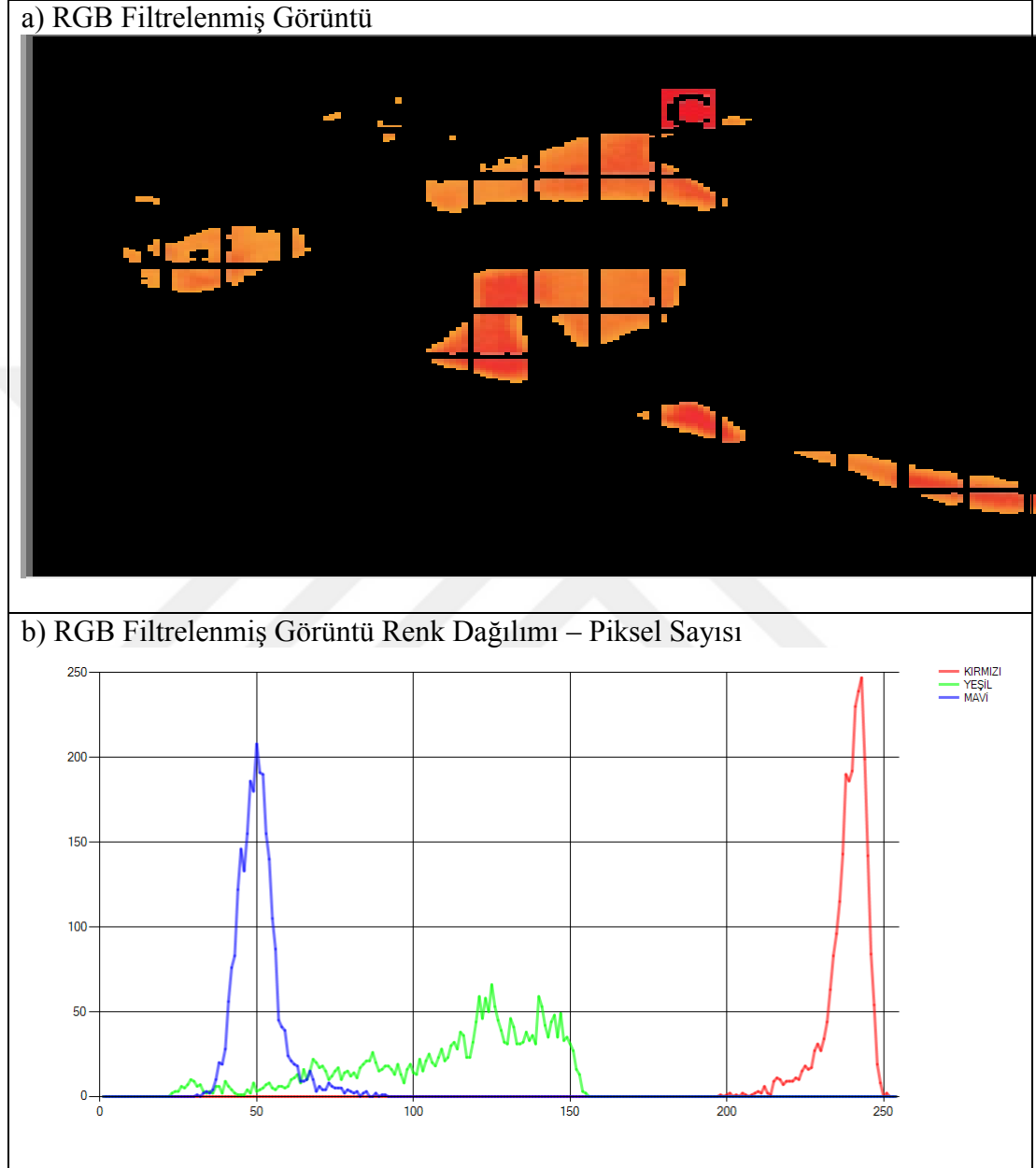
Programa termal kamera görüntü sıcaklık aralığı 24 – 40 °C olarak girilmiştir. Şekil 5.5' de Ana Resim ve bu resme ait renk dağılımları görülmektedir.



Şekil 5.5 2. Çalışmaya ait Ana Görüntü ve Renk Dağılımlarını Gösteren Grafik

Bu resim üzerinde, 34,8°C sıcaklık değerinde programın RGB Filtreleme fonksiyonu çalıştırıldığında, ana görüntü üzerinde bu değere karşılık gelen pikseller merkez alınarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) pikseller filtrelenmiştir. Program, bu yarıçap aralığı dışında kalan diğer pikselleri karartarak elde edilen görüntüyü RGB Filtrelenmiş Resim alanına getirir. Programın RGB

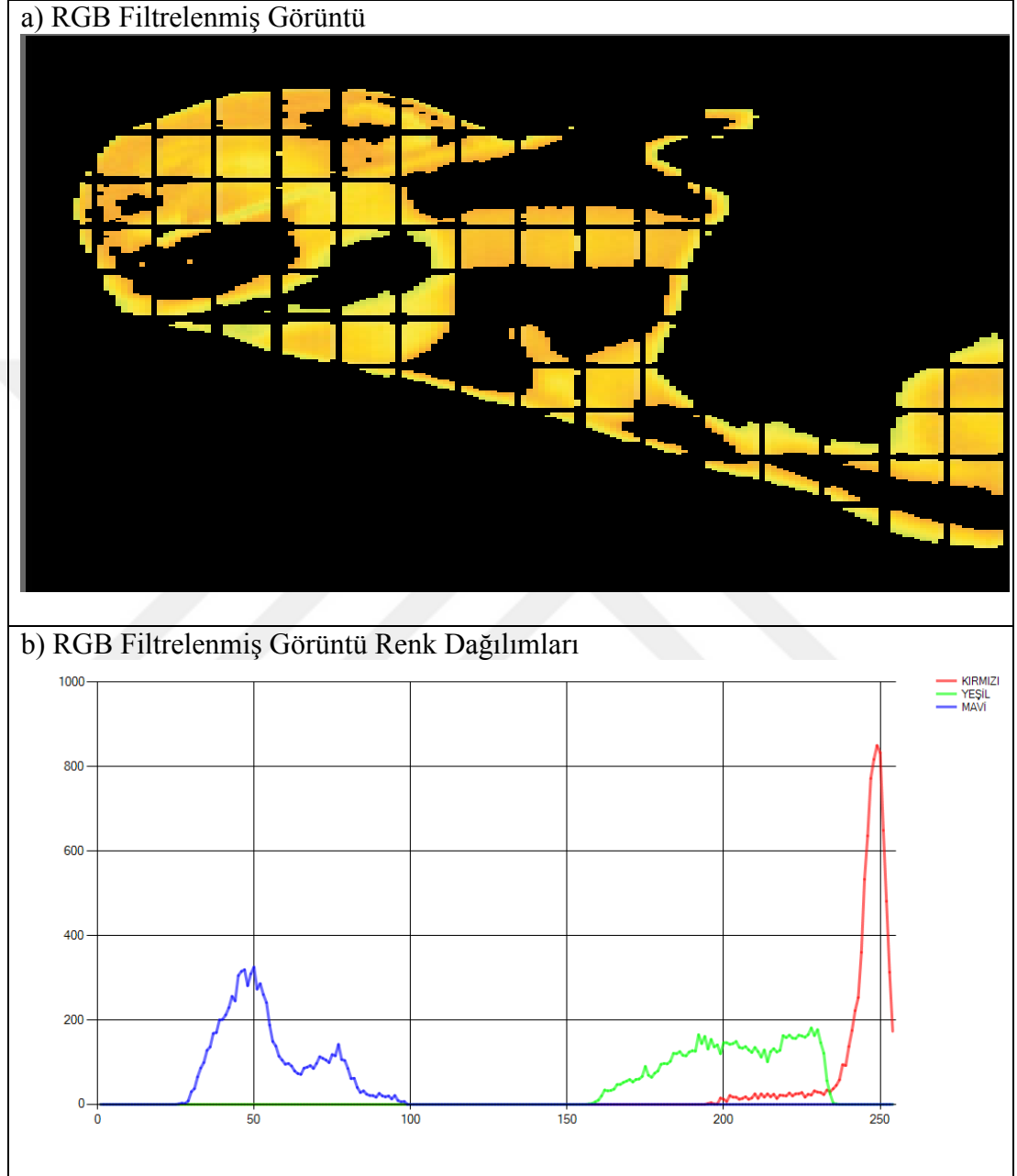
Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.6' da görülmektedir.



Şekil 5.6 34,8 °C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.6' da renk dağılım grafiği incelendiğinde, programın 34,8°C sıcaklık değerine karşılık gelen renk değerine sahip pikselleri merkez alarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerlerine sahip pikselleri filtrelediği, diğer pikselleri kararttığı görülmektedir.

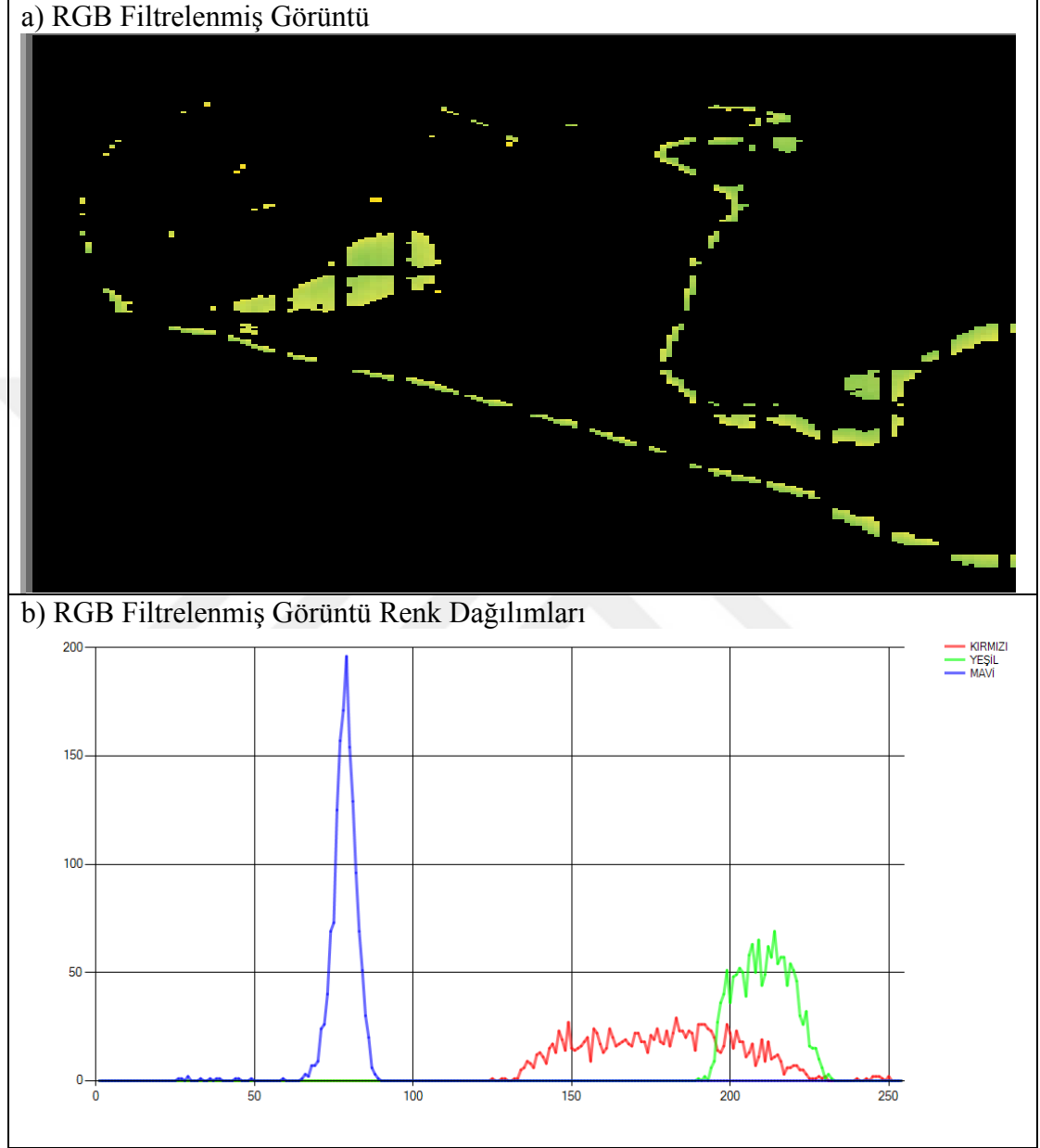
Şekil 5.7' de ana resim üzerinde, programın 33,2 °C sıcaklık değerlerinde elde ettiği RGB filtrelenmiş görüntüsü ve bu görüntüye ait renk dağılımları görülmektedir.



Şekil 5.7 33,2 °C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.7' de renk dağılım grafiği incelendiğinde, programın 33,2°C sıcaklık değerine karşılık gelen renk değerine sahip pikselleri merkez alarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerlerine sahip pikselleri filtrelediği, diğer pikselleri kararttığı görülmektedir.

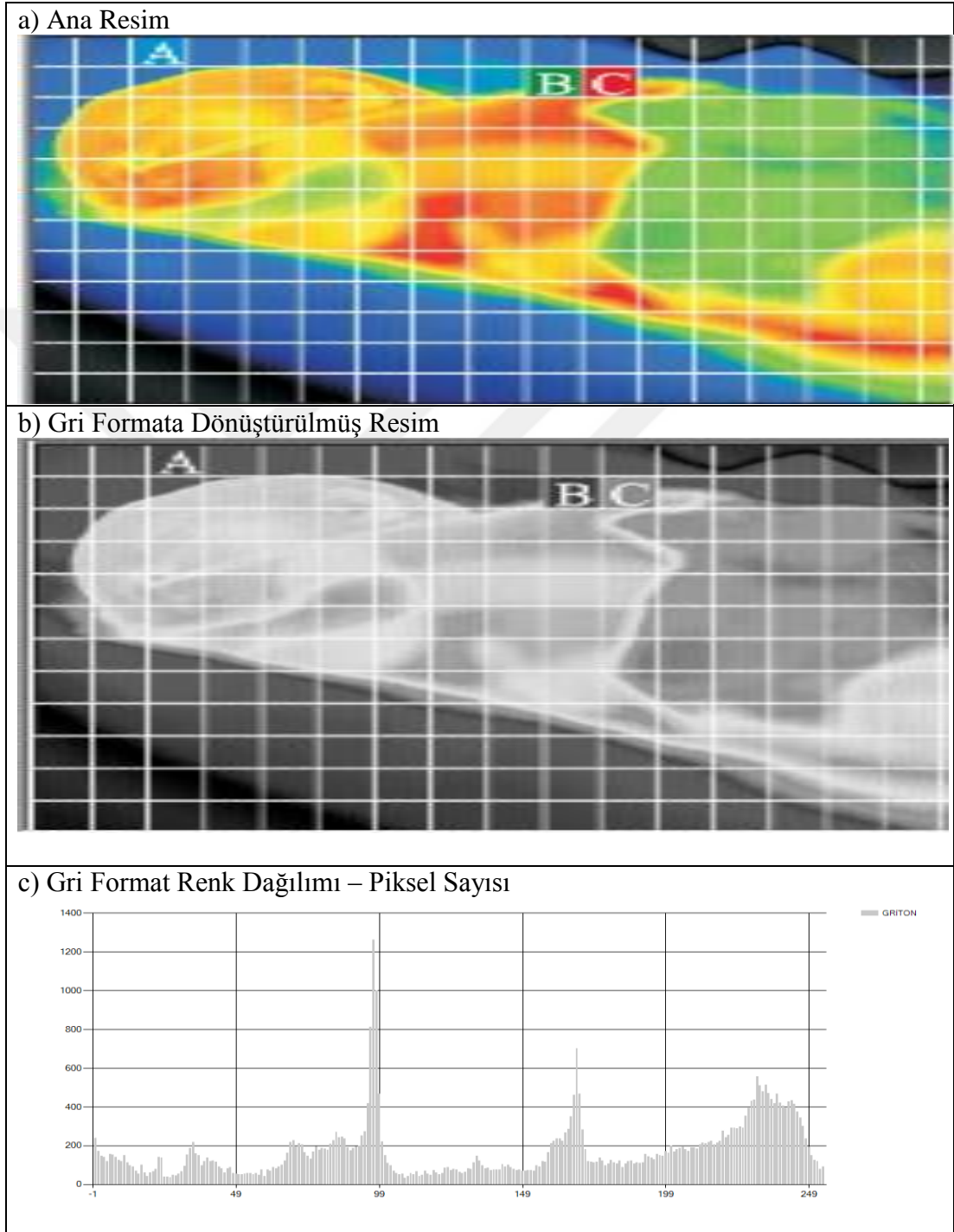
Şekil 5.8 incelendiğinde ana resim üzerinde, 32,3°C sıcaklık değerine RGB filtreleme sonucu elde edilen görüntünün ekran çıktısı ve renk dağılımları görülmektedir.



Şekil 5.8 33,2 °C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

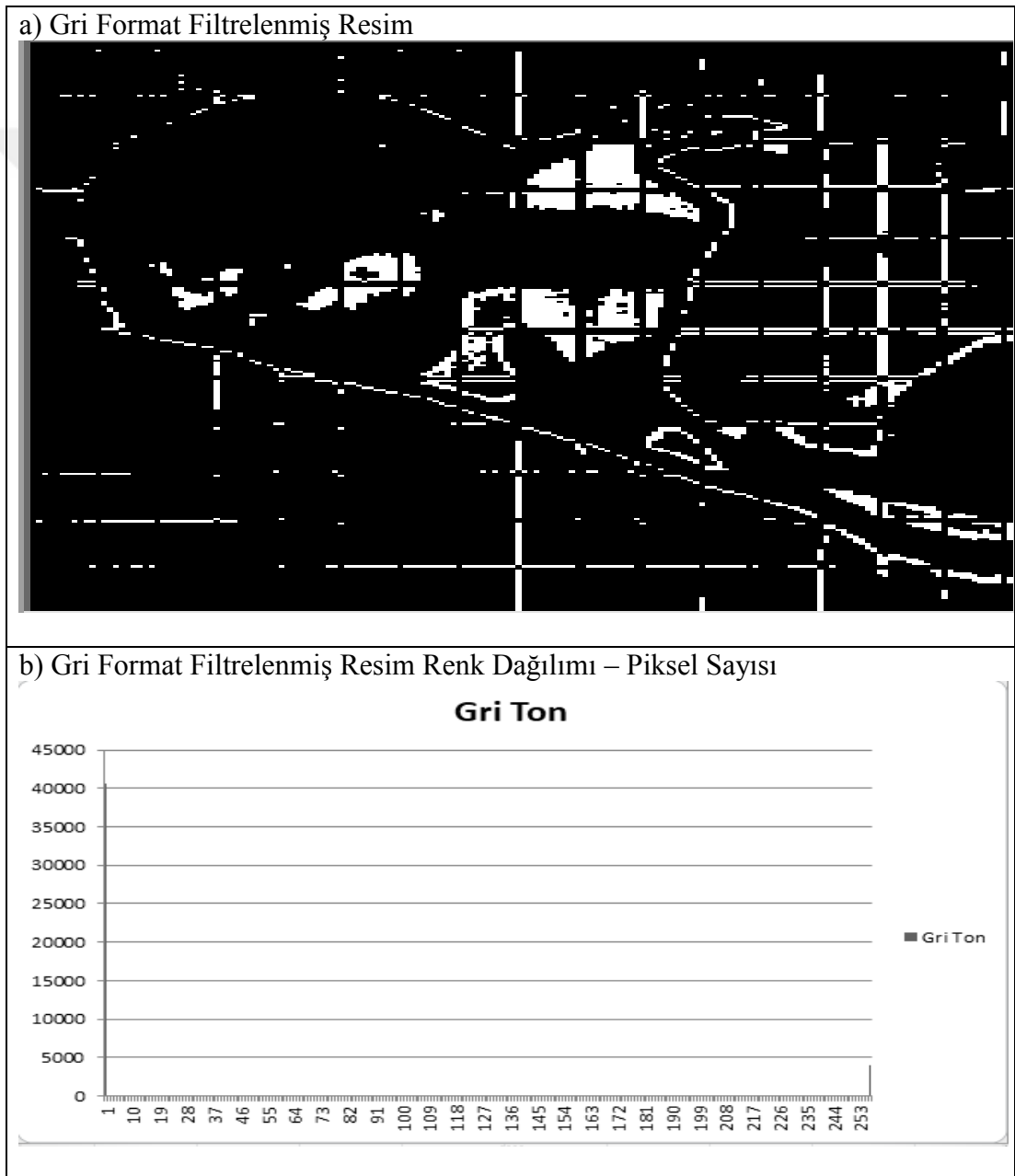
Şekil 5.8' de renk dağılım grafiği incelendiğinde, programın 32,3°C sıcaklık değerine karşılık gelen renk değerine sahip pikselleri merkez alarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerlerine sahip pikselleri filtrelediği, diğer pikselleri kararttığı görülmektedir.

Şekil 5.9’ da ana görüntünün, program üzerinde RGB (0.6, 0.3, 0.1) ölçek oranlarında gri formata dönüştürülerek, Gri Format Resim Alanında görüntülenen resmin ekran görüntüsü ve renk dağılımı görülmektedir.



Şekil 5.9 2. Çalışmaya ait Ana Görüntü, Gri Formata Çevrilmiş Görüntü ve Gri Format Resim Renk Dağılımı Grafiği

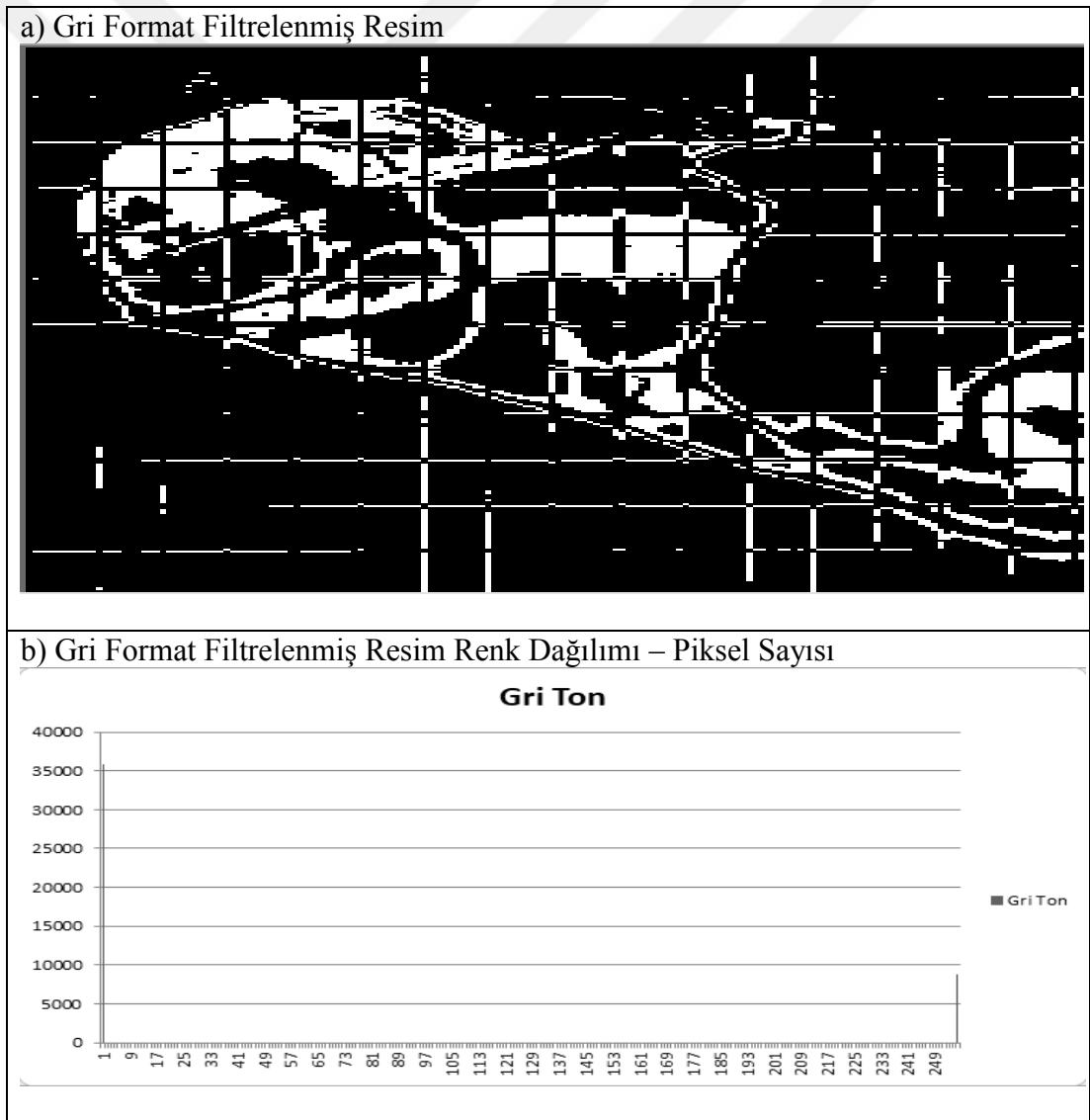
Elde edilen gri format görüntü üzerinde, program renk – ton (0 – 255) aralığında sıcaklık eşleştirmesi yapar. Gri formatlı görüntü üzerinde 34,8 °C sıcaklık değerinde Gri Format Filtreleme işlemi yapıldığında, program ana görüntü üzerinde bu değere karşılık gelen renk bilgisine sahip pikselleri filtreleyerek, (+/- 10 birim tolerans aralığında) diğer pikselleri karartır ve Gri Format Filtrelenen Resim alanına görüntüler. Programın Gri Format Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.10’ de görülmektedir.



Şekil 5.10 34,8°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.10 renk dağılımlarını gösteren grafik incelendiğinde 34,8 °C sıcaklık değerine karşılık gelen piksellerin 255 renk değeri, yani beyaz renkte görüntülediği diğer piksellerin ise 0 renk değeri, yani siyah renkte görüntülediği görülmektedir.

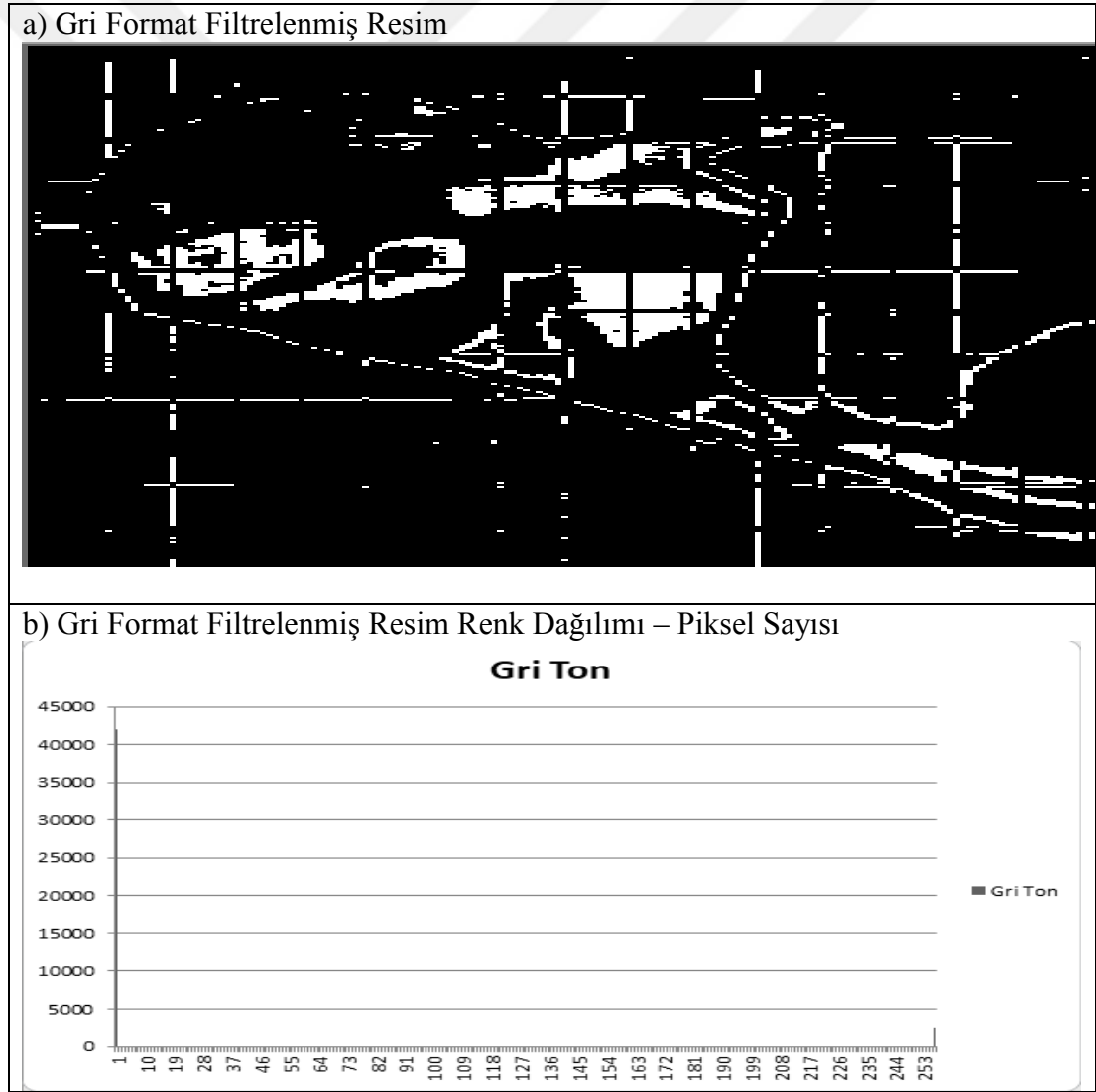
Gri formatlı görüntü üzerinde 33,2 °C sıcaklık değerinde Gri Format Filtreleme işlemi yapıldığında, program ana görüntü üzerinde bu değere karşılık gelen renk bilgisine sahip pikselleri filtreleyerek, (+/- 10 birim tolerans aralığında) diğer pikselleri karartır ve Gri Format Filtrelenen Resim alanına görüntüler. Programın Gri Format Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.11’ de görülmektedir



Şekil 5.11 33,2°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.11’ de renk dağılımlarını gösteren grafik incelendiğinde 33,2°C sıcaklık değerine karşılık gelen piksellerin 255 renk değeri, yani beyaz renkte görüntülediği diğer piksellerin ise 0 renk değeri, yani siyah renkte görüntülediği görülmektedir.

Gri formatlı görüntü üzerinde 32,3 °C sıcaklık değerinde Gri Format Filtreleme işlemi yapıldığında, program ana görüntü üzerinde bu değere karşılık gelen renk bilgisine sahip pikselleri filtreleyerek, (+/- 10 birim tolerans aralığında) diğer pikselleri karartır ve Gri Format Filtrelenen Resim alanına görüntüler. Programın Gri Format Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.12’ de görülmektedir



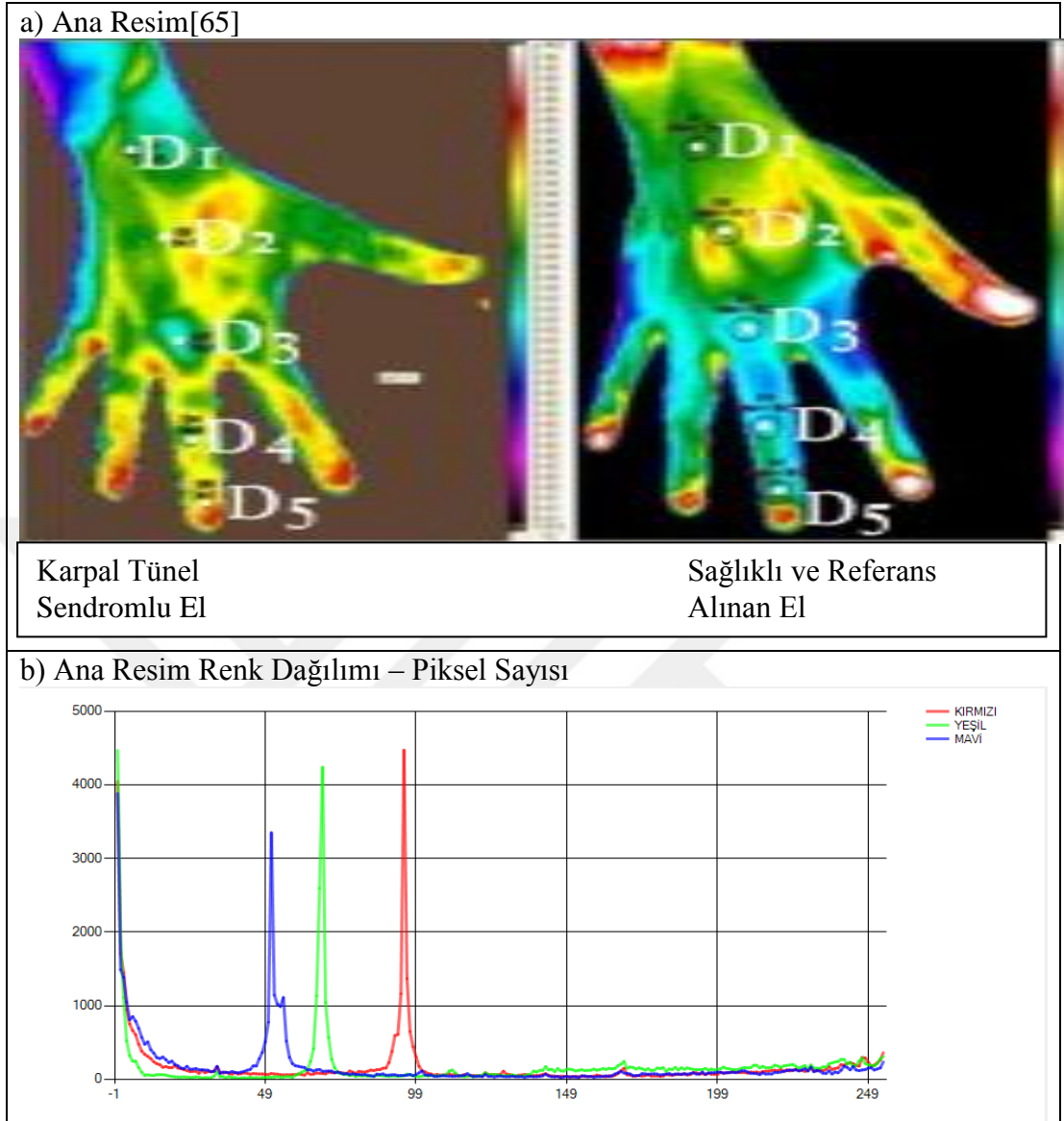
Şekil 5.12 32,3°C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.12 renk dağılımlarını gösteren grafik incelendiğinde 32,3 °C sıcaklık değerine karşılık gelen piksellerin 255 renk değeri, yani beyaz renkte görüntülediği diğer piksellerin ise 0 renk değeri, yani siyah renkte görüntülediği görülmektedir.

5.3 3. Çalışma

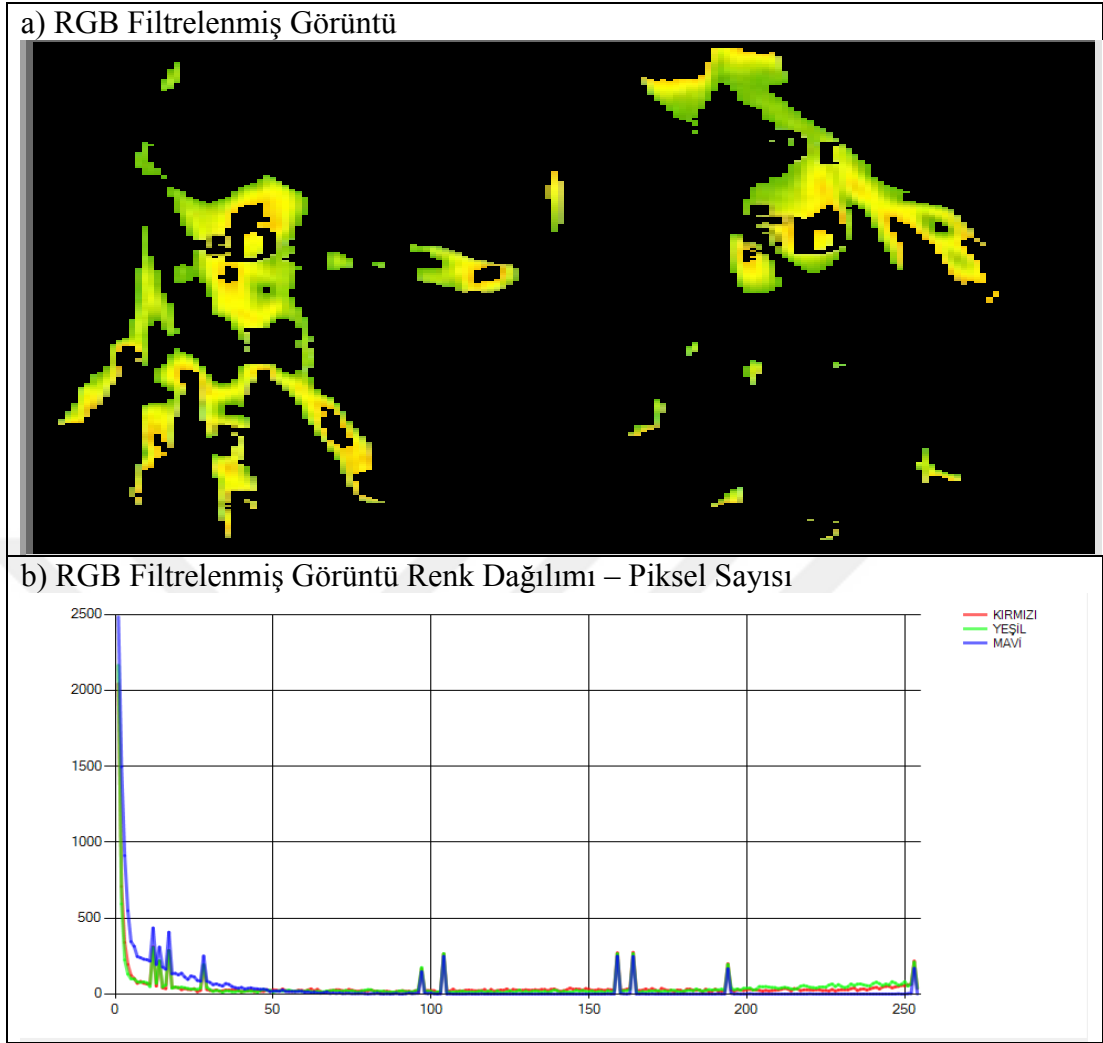
M. TKACOVA ve arkadaşları tarafından hazırlanan, “Infrared Diagnostics of the Human Hand” isimli makalede kullanılan, insan elindeki karpal tünel sendromu hastlığının kızılötesi kameralar kullanılarak teşhis edilmesi çalışmasındaki insan eli termogramları üzerinde çalışılmıştır[65].

Termogram Analizi Programına 25 – 35 °C termal kamera sıcaklık ölçüm aralığı girilmiştir. Ayrıca çalışmada kullanılan termal kameranın renk – sıcaklık ölçeğine göre programın yazılımı güncellenmiştir. Şekil 5.13’ de kullanılan Ana Resim ve Renk Dağılımı görülmektedir.



Şekil 5.13 3. Çalışmaya ait Ana Görüntü ve Renk Dağılımlarını Gösteren Grafik

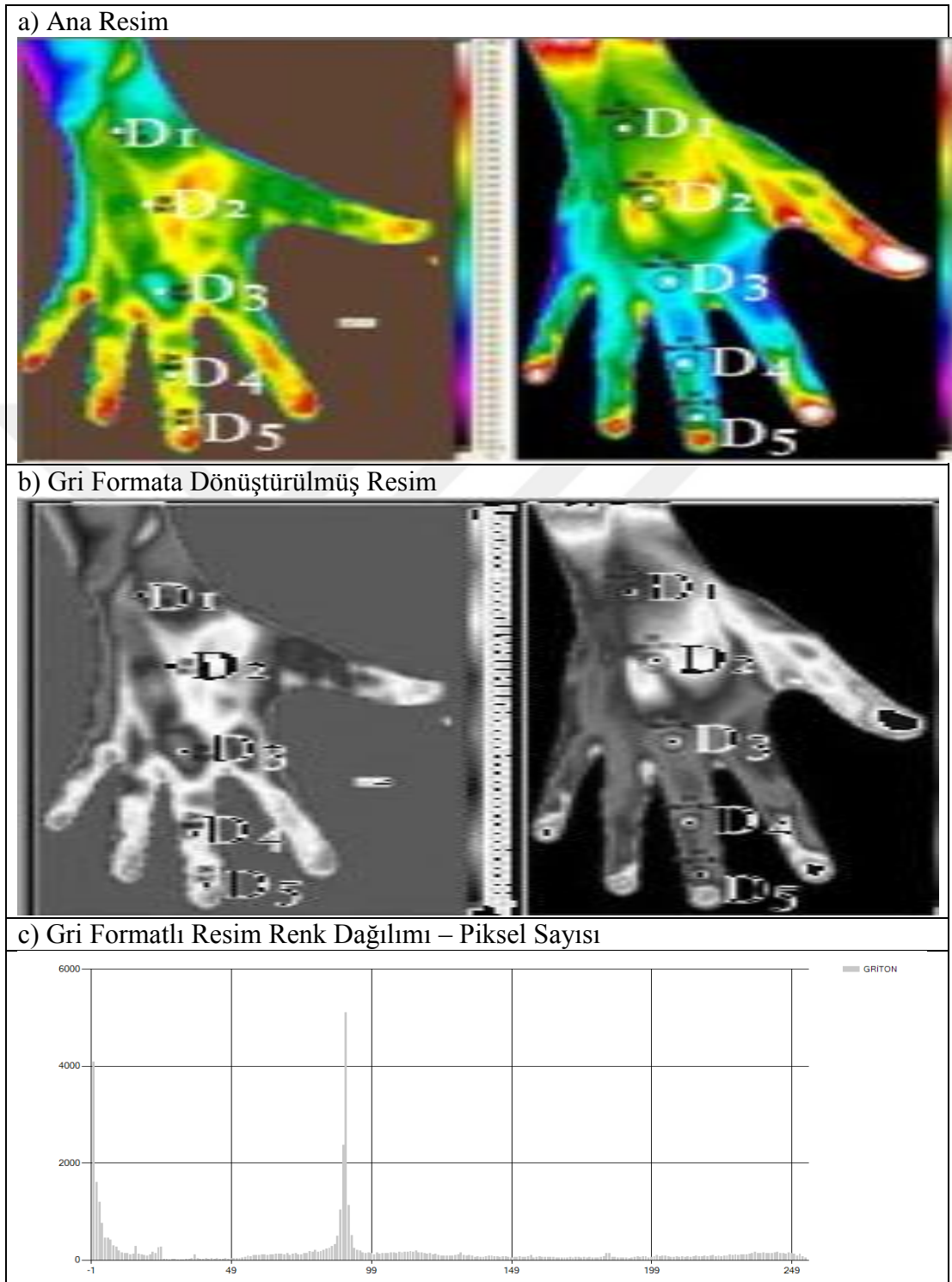
Bu resim üzerinde, $31,6^{\circ}\text{C}$ sıcaklık değerinde programın RGB Filtreleme fonksiyonu çalıştırıldığında, ana görüntü üzerinde bu değere karşılık gelen renk değerine sahip pikseller merkez alınarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerine sahip pikseller filtrelenmiştir. Program bu yarıçap aralığı dışında kalan diğer pikselleri karartarak elde edilen görüntüyü RGB Filtrelenmiş Resim alanına getirir. Programın RGB filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.14' de görülmektedir.



Şekil 5.14 31,6°C RGB Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.14’ de renk dağılım grafiği incelendiğinde, programın 31,6°C sıcaklık değerine karşılık gelen renk değerine sahip pikselleri merkez alarak, ayarlanan RGB küresi yarıçapındaki (yarıçap aralığı = 100 birim) renk değerlerine sahip pikselleri filtrelediği, diğer pikselleri kararttığı görülmektedir.

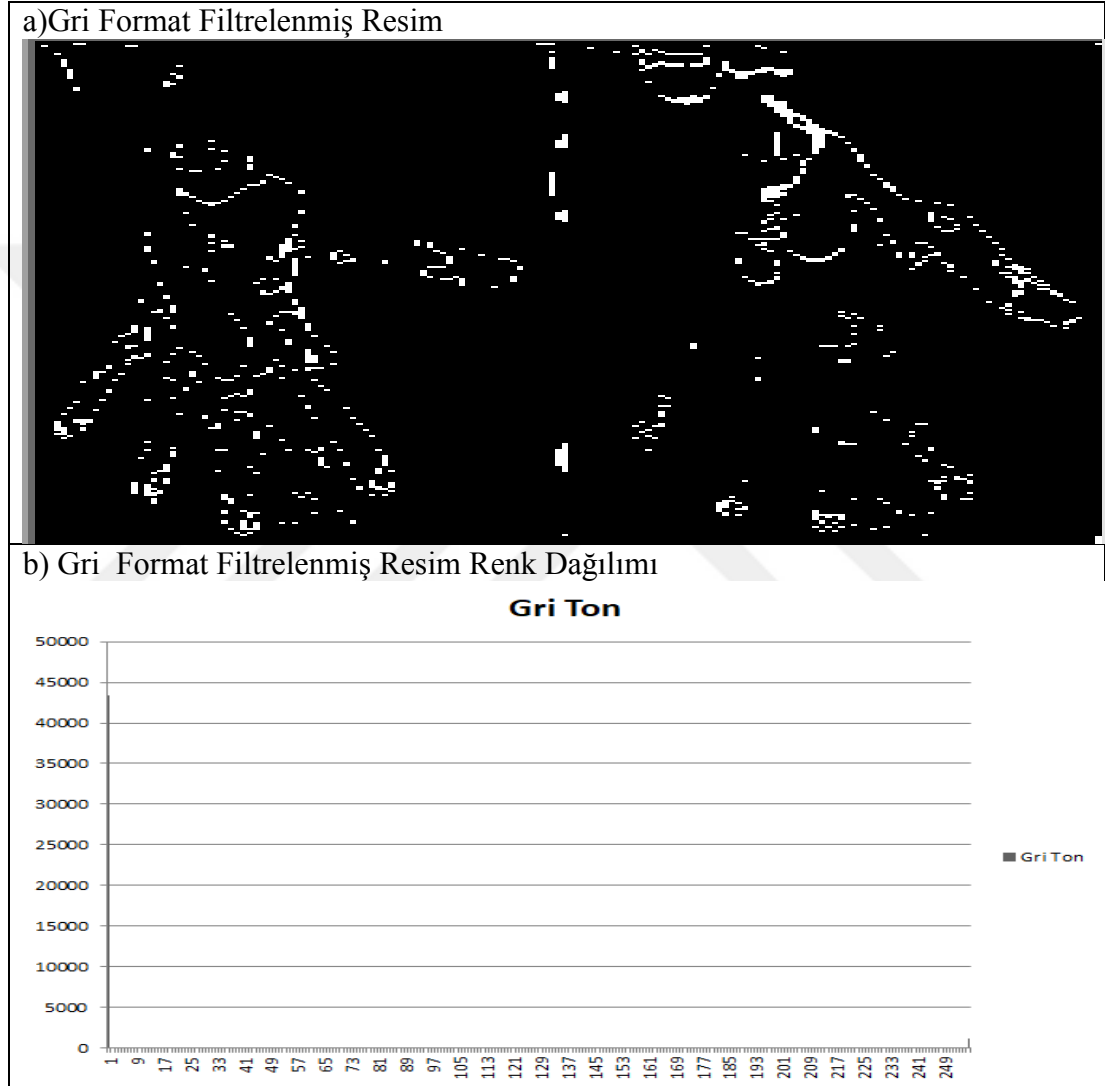
Şekil 5.15’ de ana görüntünün, program üzerinde RGB (0.6, 0.3, 0.1) ölçek oranlarında gri formata dönüştürülerek, Gri Format Resim Alanında görüntülenen resmin ekran görüntüsü ve renk dağılımı görülmektedir.



Şekil 5.15 3. Çalışmaya ait Ana Görüntü, Gri Formata Çevrilmiş Görüntü ve Gri Format Resim Renk Dağılımı Grafiği

Elde edilen gri format görüntü üzerinde program, renk – ton (0 – 255) aralığında sıcaklık eşleştirmesi yapar. Gri formatlı görüntü üzerinde 31,6 °C sıcaklık değerinde

Gri Format Filtreleme işlemi yapıldığında, program ana görüntü üzerinde bu değere karşılık gelen renk bilgisine sahip pikselleri filtreleyerek, (+/- 10 birim tolerans aralığında) diğer pikselleri karartır ve Gri Format Filtrelenen Resim alanına görüntüler. Programın Gri Format Filtreleme fonksiyonu çalıştırıldığında elde edilen görüntü ve renk dağılımı Şekil 5.16' da görülmektedir



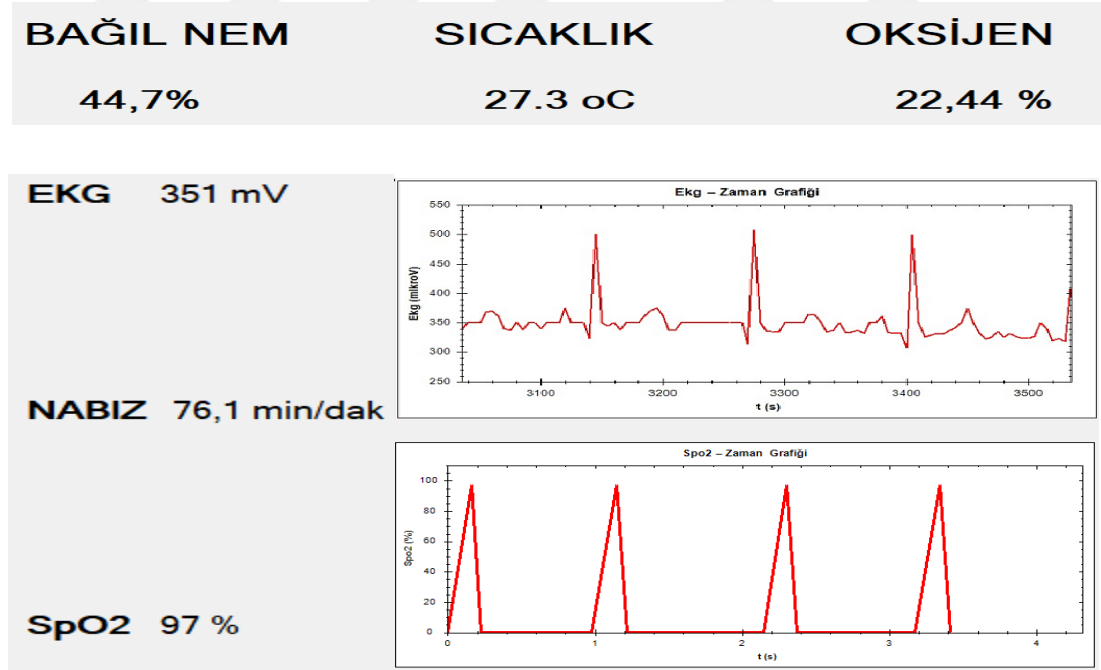
Şekil 5.16 31,6 °C Gri Format Filtrelenmiş Görüntü ve Renk Dağılımlarını Gösteren Grafik

Şekil 5.15 renk dağılımlarını gösteren grafik incelendiğinde 31,6 °C sıcaklık değerine karşılık gelen piksellerin 255 renk değeri, yani beyaz renkte görüntülediği diğer piksellerin ise 0 renk değeri, yani siyah renkte görüntülediği görülmektedir.

Bu ekran çıktıları üzerinde, referans alınan sağlıklı ele göre karpal tünel sendromlu elin sıcaklık dağılımlarının farklı ve daha yüksek olduğu tespit edilmiştir.

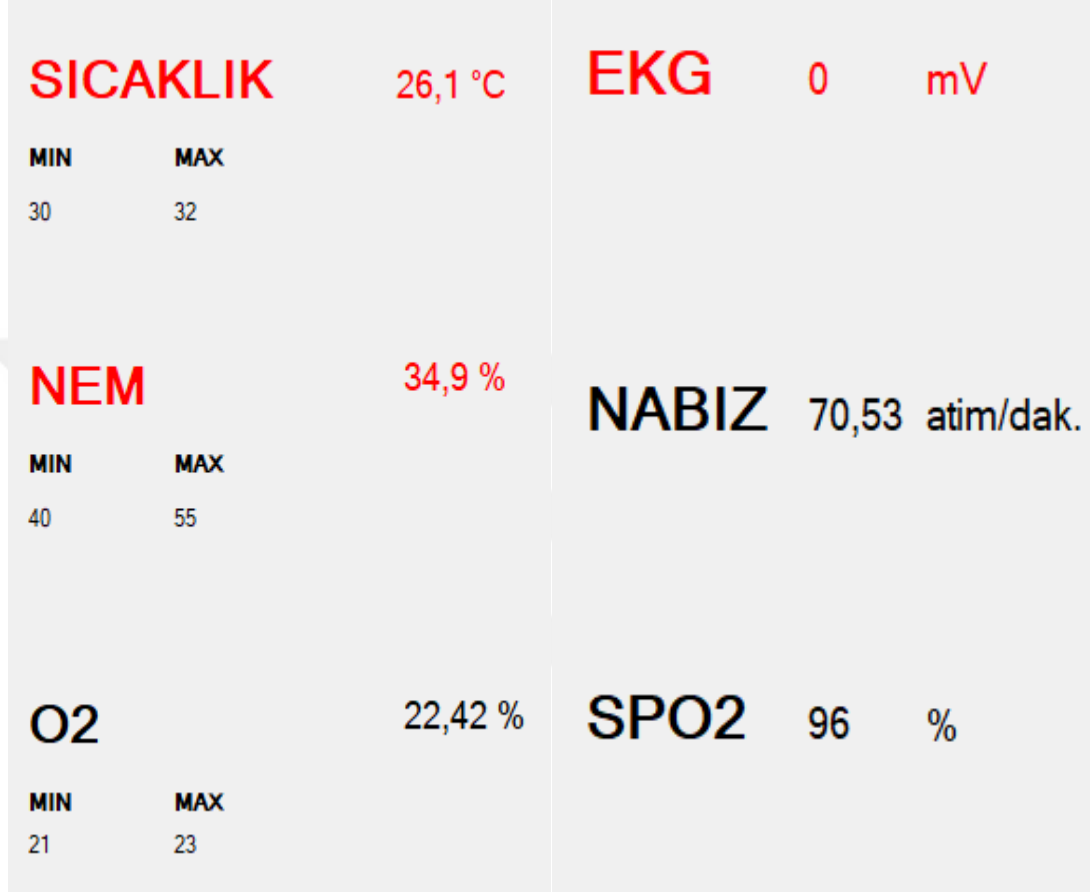
5.4 Vital ve Çevresel Parametre Bulguları

Çevresel ve vital parametrelerin sensörler vasıtasıyla ölçülüp C#' da tasarlanan kullanıcı ara yüzüne aktarımı başarılı bir şekilde yapıldığı görüldü. Tasarlanan bu sistem sağlıklı bireyler üzerinde uygulandığında ölçülen vital parametre değerlerinden nabız, SpO₂ ve EKG verilerinin sağlıklı bireylerde bulunması gereken değerlerde olduğu gözlemlenmiştir. Bununla birlikte çevresel parametre değerlerinden olan sıcaklık ve bağıl nem miktarlarının diğer ölçüm aletlerinden elde edilen verilerle büyük benzerlik taşıdığı, (aradaki farklar sıcaklık ve nem sensörlerinin hassasiyet farkından dolayı olduğu tespit edildi.) O₂ oranında ölçüm yapılan yerdeki ortalama miktarlarla uyum sağladığı tespit edilmiştir. Şekil 5.17' de Arduino kartından kullanıcı ara yüzüne aktarılan vital ve çevresel parametre verilerinin ekran görüntüsü görülmektedir.



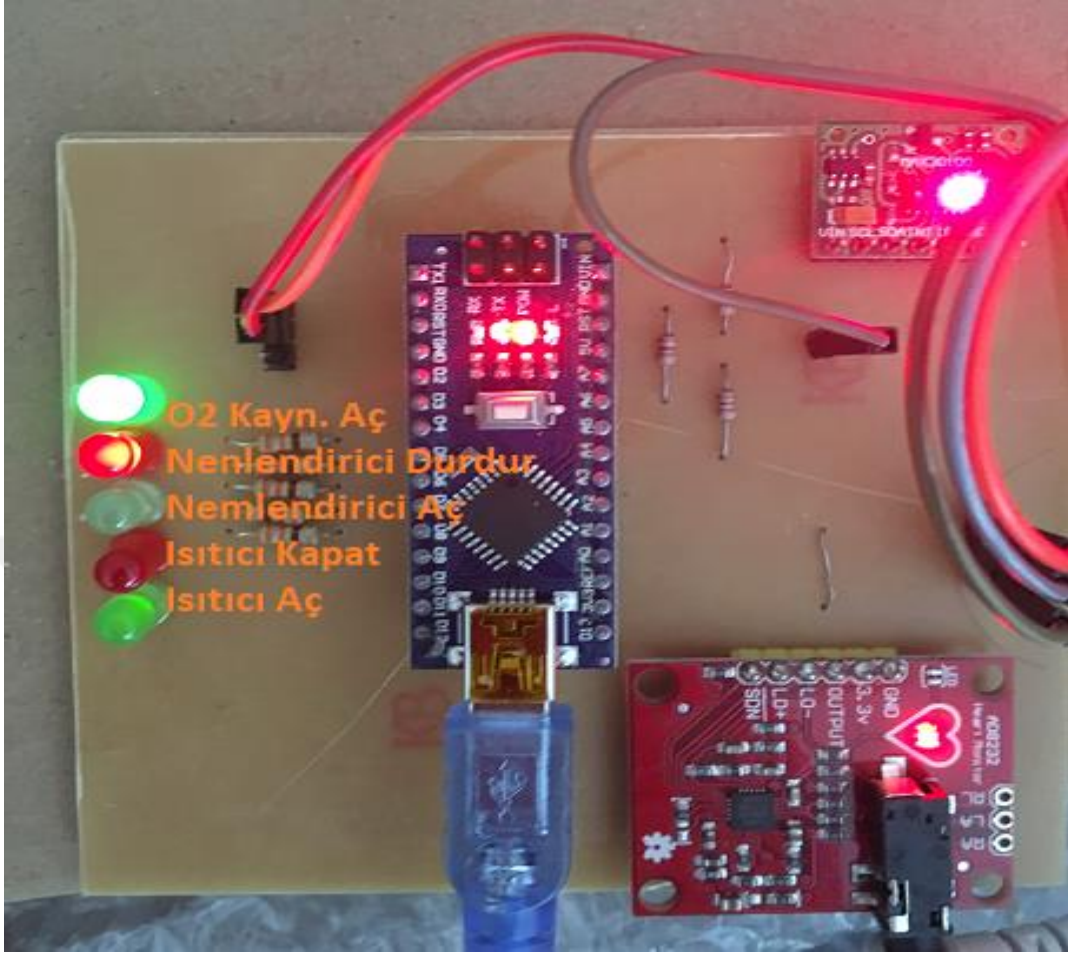
Şekil 5.17 Vital Parametre Bulguları

Ölçülen vital ve çevresel parametre değerlerinin, ayarlanan referans değerlerin dışına çıktığında ara yüz programının görsel ve işitsel uyarıları verebildiği de tespit edilmiştir. Şekil 5.18’ de programın uyarı veren ekran görüntüsü görülmektedir.



Şekil 5.18 Ortam ve Vital Parametreler Ekranı Görsel Uyarılar Ekran Görüntüsü

Ara yüzden verilen kullanıcı komutları ise Arduino kartı tarafından değerlendirilip uygun kaynakları (fan, ısıtıcı, oksijen kaynağı vb.) harekete geçirerek, kontrol edebildiği de saptanmıştır. Aşağıda Şekil 5.19’ da görüldüğü üzere kullanıcı ara yüzünden kuvödeki sıcaklığın arttırılması, nem oranının sabit kalması ve O₂ oranının arttırılması komutları verildiğinde, bu kaynakları ifade eden Led’ lerin yandığı görülmektedir.



Şekil 5.19 Arduino Nano Kartından Verilen Komutlar

5.SONUÇ ve ÖNERİLER

Yaptığım bu çalışmada, özellikle son yıllarda tıpta kullanımı yaygınlaşmaya başlayan termal görüntüleme yöntemiyle elde edilmiş termogramlar üzerinde, prematürenin vücut yüzeyi sıcaklık analizleri yapılmasını sağlayan bir program tasarlanmıştır. Tasarlanan modelde, bu alanda daha önce yapılmış çalışmalarda kullanılan termogramlar üzerinde çalışılmış ve görüntünün alındığı termal kameranın renk ölçek skalasına göre uyarlanan algoritmalar sonucu ayarlanan sıcaklıktaki lokasyonları elde edilmiştir. Bu sayede prematürelerin sürekli termografik muayene altında olabileceği düşünülmektedir. Bu sebepten ötürü prematürenin vücut yüzeyinde tespit edilen termal asimetri ve olması gereken dışındaki sıcaklık değerlerinin tespiti ile birçok hastalık için ön tanı, tanı ve tedavi işlemlerinin yürütülebileceği düşünülmektedir.

Prematüre Termogram Analizi ara yüz programına, çalışılan termal kameranın görüntü sıcaklık aralığı girildiğinde, program güncel renk – sıcaklık ilişkisini oluşturabilmektedir. Ancak termal kameraların yazılım farklılıklarına göre kullandığı renk – sıcaklık ilişkisi farklı olabilmektedir. Örneğin A marka termal kamera 35,7 °C' yi RGB (20, 50, 120) renk değerinde görüntüleyebilirken, B marka termal kamera RGB (70, 180, 10) renk değerlerinde görüntüleyebilir. Bu durumda çalışılacak termal kameranın renk – sıcaklık ilişkisine göre yazılımın güncellenmesi gerekir.

Prematüre Termogram Analizi ara yüzünde, Ana Resim gri formata çevrilerek siyah – beyaz ton aralığında renk – sıcaklık ilişkisi prematüre vücut yüzey sıcaklık analizlerinde daha elverişli olabilir. Gri Format Filtrele fonksiyonu ile seçilen eşik değer belirli bir tolerans aralığında beyaz renkte görüntülenir. Bu sayede kullanıcılar daha rahat yorum ve analiz yapabilir. Ancak ana resmi gri formata dönüştürürken RGB renk tonlarının belirli bir oranda toplanıp yeni renk değeri elde edildiği için yine termal kameranın renk – sıcaklık ölçeğine göre bu oranlar girilmelidir.

Küvözlerde prematürenin ihtiyaç duyduğu optimum çevre koşullarının (sıcaklık, nem ve O₂ oranı) ölçümü ve kontrolü tasarlanan küvöz kontrol kartı ile sağlanmaktadır. Elde edilen verilerin güvenilir ve stabil olduğu saptanmıştır. Bununla birlikte bu

verilerin seri haberleşme yoluyla C# programında tasarlanan ara yüz programına başarılı bir şekilde aktarımının yapıldığı görülmüştür. Ara yüzden gelen komutların da Arduino Nano kartı tarafından değerlendirme ve ilgili birimi (ısıtıcı, fan, nemlendirici, O₂ kaynağı) harekete geçirme işlevini yerine getirdiği tespit edilmiştir. Ayrıca ölçülen ortam parametreleri değerlerinin, programa girilen sınır değerler dışına çıkması durumunda ara yüz programı görsel ve işitsel uyarılar da vermektedir. Böylece görevli personeller ortaya çıkan sorun ve arızaları erken fark edip müdahale edebilirler.

Genellikle sağlık sorunlarıyla dünyaya gelen prematürelerin, hem bakım hem de tedavilerinde vital (yaşamsal) göstergelerinin tespiti ve takibi çok önemlidir. Bu yüzden tasarladığım modelde prematürenin bazı vital parametreleri de ölçülüp görüntülenmektedir.

Prematürenin vital parametrelerinin de kuvöz kontrol kartı tarafından ölçülüp ara yüz programına aktarımı başarılı bir şekilde yapılmıştır. Ara yüzde görüntülenen vital parametre verilerinin prematürenin sağlık sınırlarını tehdit eden noktaları aşması durumunda yine programın görsel ve işitsel uyarıları başarılı bir şekilde yaptığı görülmüştür. Bu sayede görevli sağlık personelleri prematürede meydana gelebilecek sağlık sorunlarını tespit edip müdahale edebilir.

Vital parametreleri oluşturan biyolojik işaretler çok küçük genlikli ve gürültülü sinyaller olduğu için, algılayıcılar ve dönüştürücülerin oldukça hassas oranda çalışması gerekmektedir. Çalışmamda elde ettiğim EKG grafiklerinde küçük gürültüler ortaya çıkabilmektedir. Bunun sebebi olarak, elektrotların ve sinyalleri taşıyan kabloların gürültülere karşı tam bir izolasyon sağlayamadığı düşünülmektedir. Ancak EKG verilerindeki P-Q-R-S-T dalgaları anlamlı olduğu için bu durum ihmal edilmiştir.

Prematürenin hem vital parametreleri (nabız, vücut sıcaklığı, EKG, SpO₂) hem de ortam parametreleri (sıcaklık, nem ve oksijen konsantrasyonu) uygun sensörler ile ölçülerek tek bir kullanıcı ara yüzünde hem görüntülenebilmekte hem de kontrol edilebilmektedir. Bu şekilde tek bir ara yüz programı ile sağlık personelleri,

prematürelerin kontrol, teşhis ve tedavi işlerini daha kolay, hızlı ve etkili bir şekilde yapabileceği düşünülmektedir.

Çalışmanın ileri safhası olarak tasarlanan bu modele prematürenin tedavi işlemlerinde kullanılan fototerapi, infüzyon pompası, enjektomat, ventilatör vb. cihazları da kontrol edebilen algoritma ve sistemler tasarlanarak, entegre edilebileceği düşünülmektedir. Bununla birlikte yapay zekâ teknikleri kullanılarak bütün bu cihaz ve sistemlerin kullanımı ve kontrolü, belirli bir referans aralığında, tasarlanan modelin kontrol sistemi vasıtasıyla yürütülebilir. Bu sayede yapay zekâ algoritmalarının ürettiği oldukça hassas değerlerdeki çözüm ve kararlar çok hızlı bir şekilde prematüreye uygulanabilir. Bu da prematürelerin hayata tutunmasını, daha kısa sürede ve daha etkin tedavi alarak sağlık sorunlarının daha hızlı giderilebilmesini sağlayabilir.

KAYNAKLAR

1. Bayram N., Riskli Pretermlerde Transport Edilen Ve Edilmeyen Grupların Morbidite Ve Mortalite Yönünden Karşılaştırılması, Uzmanlık Tezi, Sağlık Bakanlığı Bakırköy Kadın Doğum Ve Çocuk Hastalıkları Eğitim ve Araştırma Hastanesi, İstanbul, 2006
2. Megep, Biyomedikal Cihaz Teknolojileri, Kuvözler, Ankara, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/K%C3%BCvezler.pdf, 2012
3. Megep, Anestezi Ve Reanimasyon, Vital Bulgular, Ankara, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Ya%C5%9Fam%20Bulgular%C4%B1.pdf, 2011
4. https://web.itu.edu.tr/~dokur/EMG_Giris/EMG_07.pdf, Erişim Tarihi: 17.03.2019
5. www.yildiz.edu.tr/~nicoskun/tipelektronigi.ppt, Erişim Tarihi: 17.03.2019
6. Ekmekci H.D., E-Health Kalkan Ve Arduino Kullanılarak Çoklu Fizyolojik İşaretlerin Bilgisayar Ortamında Görüntülenmesi, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi, Afyon, 2017
7. Megep, Biyomedikal Cihaz Teknolojileri, Biyomedikal Algılayıcı Ve Dönüştürücüler, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Biyomedikal%20Alg%C4%B1lay%C4%B1c%C4%B1%20Ve%20D%C3%B6n%C5%BCt%C5%9Ft%C3%BCr%C3%BCc%C3%BCler.pdf, Ankara, 2012
8. Megep, Alanlar Ortak, Dolaşım Sistemi, http://megep.meb.gov.tr/mte_program_modul/moduller_pdf/Dola%C5%9F%C4%B1m%20sistemi.pdf, Erişim Tarihi 16.03.2019, Ankara, 2012
9. Meriç, M., Kalp ve Damar Hastalıklarında Semiyoloji, 1. Baskı, İstanbul Medikal Yayıncılık, İstanbul, 2007
10. <http://www.metealpaslan.com/ekg/pedekg.htm>, Erişim Tarihi 25.03.2019
11. Zenciroğlu A., ve ark., Temel Yenidoğan Bakımı, Türkiye Halk Sağlığı Kurumu, Çocuk Ve Ergen Sağlığı Daire Başkanlığı, Ocak, 2015
12. Megep, Acil Sağlık Hizmetleri, Ekg (Elektrokardiyografi), Ankara, [http://megep.meb.gov.tr/mte_program_modul/moduller_pdf/Ekg%20\(elektrokardiografi\).pdf](http://megep.meb.gov.tr/mte_program_modul/moduller_pdf/Ekg%20(elektrokardiografi).pdf), 2011
13. http://file.atuder.org/tr/_atuder.org/fileUpload/BGShK13Ktjai.pdf, Erişim Tarihi: 25.03.2019

14. Nizam A., Ekg İşaretlerini Gerçek Zamanda Bilgisayara Aktaran Ve İşleyen Arayüz Programı Tasarımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, İstanbul, 2000
15. <https://web.itu.edu.tr/~gulmezt/BIO102/bio102chapt12-TR.pdf>, Erişim Tarihi: 25.03.2019
16. <https://www.acilci.net/pediyatrik-ekg>, Erişim Tarihi: 26.03.2019
17. https://www.wikidoc.org/index.php/Premature_atrial_contraction_electrocardiogram, Erişim Tarihi: 26.03.2019
18. Megep, Biyomedikal Cihaz Teknolojileri, Elektrokardiyografi Ölçümleri, <http://www.cahilim.com/pdf/medikal/elektrokardiyografi-olcumleri.pdf>, Ankara, 2008
19. Topal F.E., İzmir Katip Çelebi Üniversitesi Tıp Fak. Acil Tıp Bölümü <http://file.atuder.org.tr/atuder.org/fileUpload/pXarvTK1Ujxr.pdf>, Erişim Tarihi: 27.03.2019
20. Kaya Y., Acıbadem Kayseri Hastanesi Eğitim ve Gelişim Hemşiresi, 2016, <http://www.acibademhemsirelik.com/e-dergi/91/docs/ug-2.pdf>, Erişim Tarihi: 27.03.2019
21. Vegfors M et al, Assessment of peripheral blood flow using a pulse oximeter, J Clin Monit, 6:1-4, 1990
22. Szocik JF., Barker SJ., Tremper KK., Fundamental principles of monitoring instrumentation. In. Miller's Anesthesia. Ed, Miller RD. Pennsylvania: Elsevier, p: 1191-1226, 2005
23. Öncel T., Puls Oksimetre, Türk Yoğun Bakım Dergisi, Cilt:4, Sayı:2, 96-106, 2006
24. Ateş G., Polat K., Bulanık Mantık ve Hareketli Ortalamaya Dayanan Puls Oksimetre ile Kandaki Oksijen Saturasyonunun Ölçülmesi , TMMOB EMO Ankara Şubesi Haber Bülteni 2012/2
25. Altuğ M. E., Gönenci R., Pulse Oksimetre ile Arteriyel Oksijenasyonun İzlenmesi, Veteriner Cerrahi Dergisi, 9 (3-4), 58-62, 2003
26. Megep, Anestezi Ve Reanimasyon, Monitörize Etme, Ankara, http://megep.meb.gov.tr/mte_program_modul/moduller_pdf/Monit%C3%B6rize%20Etme.pdf, 2011
27. Acartürk E., Koah Hastalarındaki Oksijen Saturasyonunun Pulse Oksimetre ile Tesbitinin Arter Kan Gazı Tetkiki ile Korelasyonu ve Bu Korelasyonu Etkileyen Faktörler, Tıpta Uzmanlık Tezi, Sağlık Bakanlığı Süreyyapaşa Göğüs Hastalıkları ve Göğüs Cerrahisi Eğitim ve Araştırma Hastanesi, İstanbul, 2009

28. Bülbul A. ve ark., Pediatri Alanında Hizmet Sunan Sağlık Personelinin Pulse Oksimetre Kullanımı İle İlgili Bilgi Düzeyleri, Şişli Etfal Hastanesi Tıp Bülteni, Cilt: 48, Sayı: 4, syf 303-307, Mart, 2014
29. <http://aysegulyildirimkaptanoglu.com/rsmlr/dosya/dolasimsistemi.pdf>, Erişim Tarihi: 20.03.2019
30. Tümer N., Özkaya N., Yenidoğan Döneminde Hipertansiyon Tanı Ve Tedavi, Adnan Menderes Üniversitesi Tıp Fakültesi Dergisi; 2(1) : 33 – 38, Guignard J P, Drukker A. Clinical Neonatal Nephrology. In: Barratt M, Avner E, Harmon W (Eds). Pediatric Nephrology W K. 4 Ed. Lippincott Williams &Willkins, 1051-1065, 1999
31. <http://www.ahmetalpman.com/defkonuoku.asp?id=138>, Erişim Tarihi: 01.05.2019
32. <http://kisi.deu.edu.tr/ihsan.esen/V%C4%B0TAL%20BULGULAR%20yeni%20%20Copy.pdf>, Erişim Tarihi: 01.05.2019
33. Kösa E., Çınar N., Prematüre Bebeklerde Hipoterminin Önlenmesi: Plastik Örtü Kullanım, Balıkesir Sağlık Bil. Dergisi Cilt:3, Sayı:3, Syf 161-165, Esentepe Kampüsü, Sakarya, Aralık 2014
34. Dağoğlu T., Görak G., Temel Neonatoloji ve Hemşirelik İlkeleri, 2.Baskı, s.179-193, Nobel Tıp Kitabeleri, İstanbul, 2008
35. Lillian U. Palmon et al., Ruptured or Intact: What Can Linear Echoses Silicone Breast İmplants Tell Us?, American Journal of Radiology, 168: 1595-1598, 1997
36. Ring EFJ., The Historical Development of Thermal Imaging in Medicine. Rheumatology, 43(6): 800-2, 2004
37. Düzgün D., Or M., Termal Kameraların Tıpta Veteriner Hekimlikte Kullanımı, Tubav Bilim Dergisi, Cilt:2, Sayı:4, Sayfa:468-475, İstanbul, 2009
38. Çalışan M., Türkoğlu İ., Termal Kameralar ve Uygulamaları, <http://web.firat.edu.tr/feeb/kitap/c12/109.pdf>, Erişim Tarihi: 13.12.2019
39. Durmuş, G., Görhan, G., Doğal Taş Plakaların Isıl İletkenlik Bakımdan Termografik Görüntülerinin İncelenmesi, Selçuk Üniversitesi Teknik Bilimler Meslek Yüksek Okulu Teknik Online Dergi, Cilt 8, Sayı: 1, 2009
40. Günyel, B., Bala, E., Akar, G.B., Uzun ve Orta Dalga Boylu Kızılötesi Görüntü Kaynaştırılması, Signal Processing and Communications Applications, SIU, 2007
41. Büyüksalih G., Kızılötesi Ccd Kameralar Ve Teknik Özellikleri, https://www.harita.gov.tr/images/dergi/makaleler/122_3.pdf, Erişim Tarihi: 02.04.2019

42. Demirci B., Termal Video Görüntüleme, Elektrik Mühendisliği Dergisi Sayı-458 sayfa 56-57, Eylül, 2016
43. Susam. M, Koroner Arter Cerrahisinde Termal Görüntüleme Kullanılarak Greft Değerlendirilmesi, Uzmanlık Tezi, 9-13, 2005
44. Laughmiller J. A et al, Relationship Between Mean Surface Temperature Measured by Use of Infrared Thermography and Ambient Temperature in Clinically Normal Pigs and Pigs Inoculated with Antinobacillus Pleuropneumonia. Am, J. Vwt.Res, 62: 676-681, 2001
45. Mike. R, Dunbar M.S. et al, Use of Infrared Thermography to Detect Thermographic Changes in Mule Deer (Odocoileus Hemionus) Experimentally Infected with Foot and Mouth Disease, Journal of Zoo and Wildlife Medicine, 40(2): 296–301, 2009
46. JW. Trum, FM. Gubler, R. Laan et al, The Value of Palpation, Varicoscreen Contact Thermography and Colour Doppler Ultrasound in The Diagnosis of Varicocele, Hum Reprod, 11(6): 1232-1235, 1996
47. <http://paintreatmentcenter.net/thermography-qa/2715253>, Erişim Tarihi: 02.04.2019
48. Jatteau. M, Techniques of Infrared thermography, Bibl Radiology, (6): 9-24, 1975
49. Habash R. W. Y., Electromagnetic Fields and Radiation, Human Bioeffects and Safety, New York, Marcel Dekker, 167-173, 2002
50. Çolak A. ve ark., Early Detection of Mastitis Using Infrared Thermography in Dairy Cows, Journal of Dairy Science, 91(11): 4244-8, 2008
51. Şirel. D ve ark., Synovial Hemangioma of The Knee, Official Journal of The Turkish Society of Radiology, 7(1), 131-134, 2001
52. Karakoç M., .Net Platformunda Emgu Cv Ve Aforge.Net ile Görüntü İşleme Tekniklerinin Uygulanması, İleri Teknoloji Bilimleri Dergisi, Cilt 5, Sayı 3, Akdeniz Üniversitesi, Bilgisayar Bilimleri Araştırma Ve Uygulama Merkezi (Baum), 07058, , Aralık, 2016
53. Karakoç M., Görüntü İşleme Teknikleri ve Yapay Zeka Yöntemleri Kullanarak Görüntü İçinde Görüntü Arama, Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli, 2011
54. <https://medium.com/@resulzel/c-nedir-c-ile-neler-yap%C4%B1labilir-199e8b337615>, Erişim Tarihi: 16.03.2019
55. Sharp, J., Adım Adım Microsoft Visual C# 2008. Arkadaş Yayın Evi, 696s, Ankara, 2009
56. <http://www.aforgenet.com/framework>, Erişim Tarihi: 10.11.2019

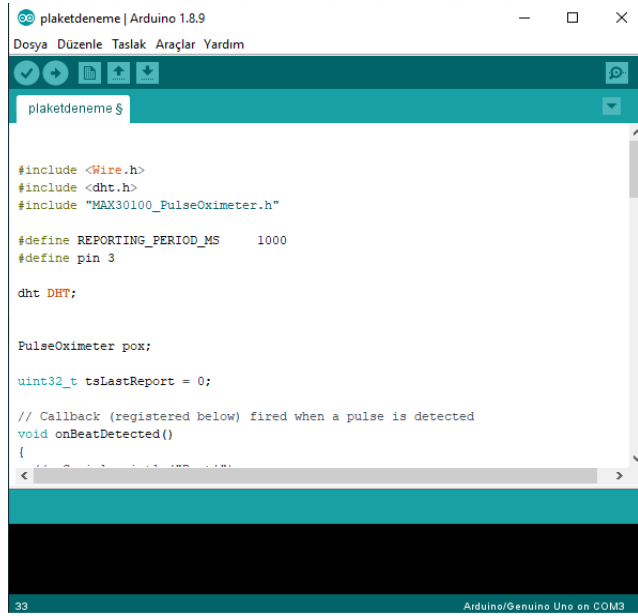
57. <https://www.nedir.com/arduino>, Erişim Tarihi: 14.03.2019
58. <http://www.elektrohobim.com/index.php/arduino/genel/30-arduino-nano>, Erişim Tarihi: 15.03.2019
59. DHT22/AM22 Datasheet, <https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>, Erişim Tarihi: 12.03.2019
60. http://www.it-wismar.de//documents/specs/M-04_spec.pdf, Erişim Tarihi: 13.03.2019
61. <https://www.robotistan.com/max30100-kalp-nabiz-sensoru>, Erişim Tarihi: 10.01.2019
62. <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>, Erişim Tarihi: 18.03.2019
63. Savaşçı D., Ceylan M. “Yenidoğan Yoğun Bakım Ünitelerinde Termal Görüntüleme Analizi (İlk Değerlendirme Sonuçları)”, 26th Signal Processing and Communications Applications Conference (SIU), İzmir, s.2.
64. Pereira C. Et al. “Noncontact Monitoring of Respiratory Rate in Newborn Infants Using Thermal Imaging”, IEEE Transactions on Biomedical Engineering, Vol. 66, No. 4, s. 1107.
65. Tkacova M. Et al. “Infrared Diagnostics of the Human Hand”, 15th International Conference on Intelligent Engineering Systems, Poprad, s. 418.
66. Pulse Oximeter and Heart-Rate Sensor IC for Wearable Health, <https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>, Erişim Tarihi: 24.03.2019
67. <https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide/all>, Erişim Tarihi: 18.03.2019
68. <https://www.analog.com/media/en/technical-documentation/tech-articles/predicting-and-finding-your-limits-ms-2385.pdf>, Erişim Tarihi: 18.03.2019

EK 1. ARDUINO

Arduino

Arduino, İtalyan elektronik mühendisleri tarafından açık kaynak kodlu olarak geliştirilen bir mikrodenetleyici kartı ve geliştirme ortamıdır. İsteyen herkesin devre şemalarını temin ederek kendi devrelerini yapabilecekleri, isterlerse devre elemanları yerleştirilmiş hazır paket olarak satın alabilecekleri, esnek, kolay kullanımlı donanım ve yazılım tabanlı bir fiziksel programlama platformudur [57].

Arduino ile bir çok sistem, devre, entegre ve sensör kontrol edilebileceği gibi bilgisayar yazılımlarına da bağlanarak beraber çalışılabilir. Arduino geliştirme kartı üzerindeki mikroişlemci (AtmegaXX) arduino programlama dili (wiring tabanlı) ile programlanır ve bu program processing tabanlı Arduino Yazılım Geliştirme Ortamı (IDE) yardımı ile karta yüklenir [57]. Ayrıca Arduino yazılımında bulunan derleyici (debugger) özelliği ile programdaki hatalar kullanıcılar tarafından rahatlıkla tespit edilebilir.



```
plaketdeneme | Arduino 1.8.9
Dosya Düzenle Taslak Araçlar Yardım

plaketdeneme §

#include <Wire.h>
#include <dht.h>
#include "MAX30100_PulseOximeter.h"

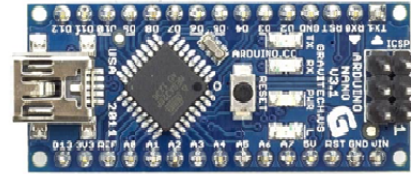
#define REPORTING_PERIOD_MS 1000
#define pin 3

dht DHT;

PulseOximeter pox;

uint32_t tsLastReport = 0;

// Callback (registered below) fired when a pulse is detected
void onBeatDetected()
{
  // ...
}
```



Arduino Kartı ve IDE Yazılım Geliştirme Ortamı

Arduino Temel Donanım Özellikleri

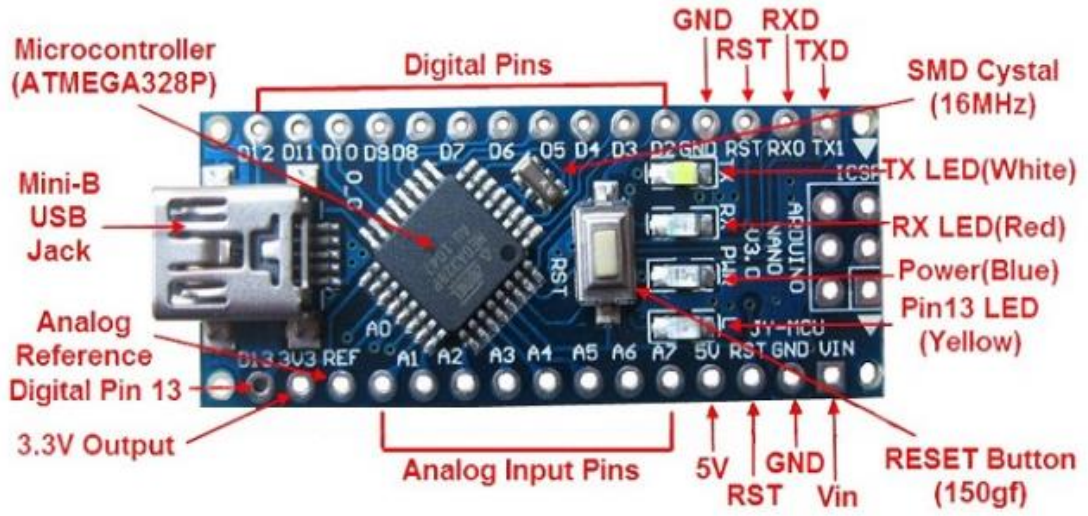
Bu donanım özellikleri board'a göre farklı özellikler gösterse de genel özellikleri aşağıda listelenmiştir. [57].

- ATmega8, ATmega168, ATmega328 mikroişlemci
- 5 voltluk regüle entegresi,
- 16MHz kristal osilator ya da seramik rezonatör,
- Flash memory,
- SRAM
- EEPROM

Arduino Nano

Arduino Nano; Atmega328 temelli bir mikrodenetleyici kartıdır. Üzerinde 14 adet dijital giriş/çıkış pini (6 tanesi PWM çıkışı olarak kullanılabilir), 8 analog giriş, 16Mhz kristal, usb soketi, ICSP konektörü ve reset tuşu bulundurmaktadır. Kart üzerinde mikrodenetleyicinin çalışması için gerekli olan tüm devre elemanları mevcuttur. Usb kablosu vasıtasıyla bilgisayara bağlanarak mikrodenetleyici içerisine tasarlanan Arduino programı gömülebilir. Ayrıca seri haberleşme yoluyla bilgisayar yazılımları ile bütünleşik çalışabilir. Arduino Nano kartı bilgisayara bağlanan usb kablosu, adaptör veya pil ile enerji ihtiyacını giderebilir [58].

Yaptığım çalışmada maliyetinin ucuz olması, farklı oranlarda besleme voltajı sağlayabilmesi (3,3 ve 5 V), paket yapıda olması, kolay programlanabilir olması ve çok sayıda açık kaynak kütüphanesi bulunması nedeniyle Arduino kartı kullanılmıştır. Boyutlarının küçük olması, üzerinde bulunan bağlantı elemanları (header) sayesinde bakır plakete kolayca lehimlenebilmesi özellikleri sebebiyle de Arduino modellerinden Arduino Nano kartı kullanılmıştır.



Arduino Nano ve Temel Birimleri [58]

Şekil 3.4’ de Arduino Nano’ nun temel birimleri, bağlantı elemanları ve özellikleri gösterilmektedir.

Arduino Nano Teknik Özellikleri [58]

Mikrodenetleyici	ATmega328
Çalışma gerilimi	+5 V DC
Tavsiye edilen besleme gerilimi	7 - 12 V DC
Besleme gerilimi limitleri	6 - 20 V
Dijital giriş / çıkış pinleri	14 tane (6 tanesi PWM çıkışını destekler)
Analog giriş pinleri	8 tane
Giriş / çıkış pini başına düşen DC akım	40 mA
3,3 V pini için akım	50 mA

Flash hafıza	32 KB (ATmega328) 2 KB kadarı bootloader tarafından kullanılmaktadır
SRAM	2 KB
EEPROM	1 KB
Saat frekansı	16 MHz
Uzunluk	45 mm
Genişlik	18 mm
Ağırlık	5 gr

EK 2. DHT22 ENTEGRESİ

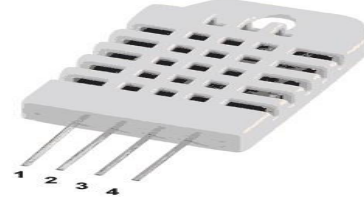
Dht22 Entegresi

Dht22 ortamın sıcaklık ve nem oranını ölçmeye yarayan, kalibre edilmiş dijital çıkış sinyali üreten bir sensördür. Özel dijital sinyal toplama tekniği ve nem algılama teknolojisini kullanarak güvenilirliğini ve kararlılığını güvence altına alır. Algılama elemanları 8 bitlik mikrodenetleyiciye bağlanır [59].

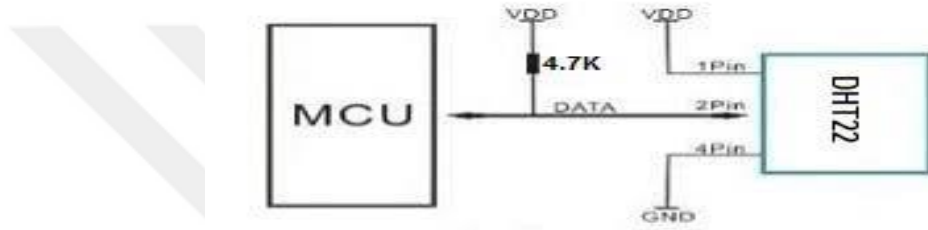
Tablo Dht22 Teknik Özellikleri [59]

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100% RH; temperature -40~80Celsius
Accuracy humidity	+2% RH(Max +-5% RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1% RH; temperature 0.1Celsius
Repeatability	humidity +-1% RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3% RH
Long-term Stability	+0.5% RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



Dht22 Entegresi



Dht22 Bağlantı Şeması

Not: 3Pin = Boş; MCU = Mikrodenetleyici

TS EN 60601-2-19 standartına göre kuvözlerde sıcaklık ölçüm toleransı +/- 0,5° C, nem ölçüm toleransı ise +/- 10 RH olarak tespit edilmiştir. Tasarlanan modelde kullanılan Dht22 entegresi +/- 0,5° C değerinde sıcaklık, +/- 5 RH değerinde ise bağıl nem ölçüm doğruluk oranında çalıştığı için ilgili standarta uygundur.

Dht22 Güç ve Pin Bağlantısı

Dht22'nin güç kaynağı 3,3-6V DC'dir. Sensöre besleme voltajı verildiğinde, sensörün uygun çalışması için 1 saniye boyunca hiçbir data gönderilmemelidir. Besleme hattı (VCC) ile toprak hattı (GND) arasına 100 nF değerinde by-pass kondansatörü kullanılması önerilir [59].

Dht22 Haberleşme İşlemi (Tek Hat, Çift Yön)

Mikrodenetleyici ile Dht22 sensörü tek data hattı üzerinden haberleşmekte ve senkronize edilmektedir [59].

Dht22 sıcaklık ve nem bilgisini 40 bitlik bir data paketinin içerisinde göndermektedir. Bu paketin yapısı şu şekildedir:

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1101</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

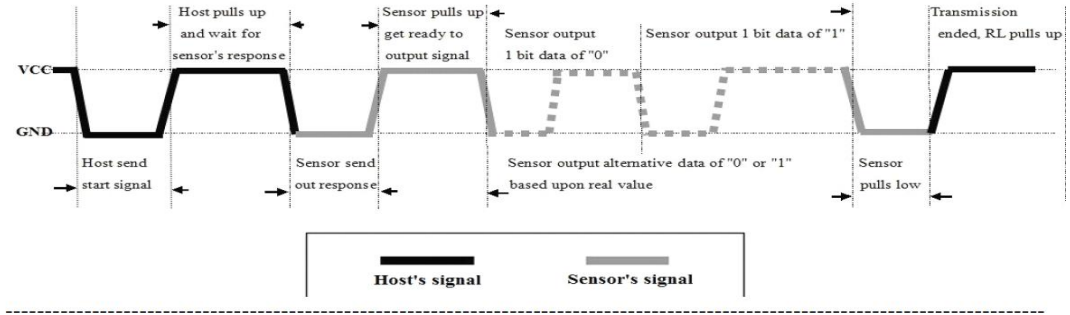
Dht22 Data Paketleri

Humidity: Nem **Temp.:**Sıcaklık

Parity bit = humidity high + humidity low + temperature high + temperature low

Burada HIGH olarak belirtilenler sıcaklık ve nem değerlerinin tam kısım, LOW olarak belirtilenler ise ondalık kısımdır. Entegreden en son gönderilen 8 bitlik data (parity bit) önce gönderilen 32 bitin toplamıdır. Gönderilen datanın doğruluğunun kontrolü için kullanılır. Entegre nem ve sıcaklık için 16 bitlik paketler kullanır.

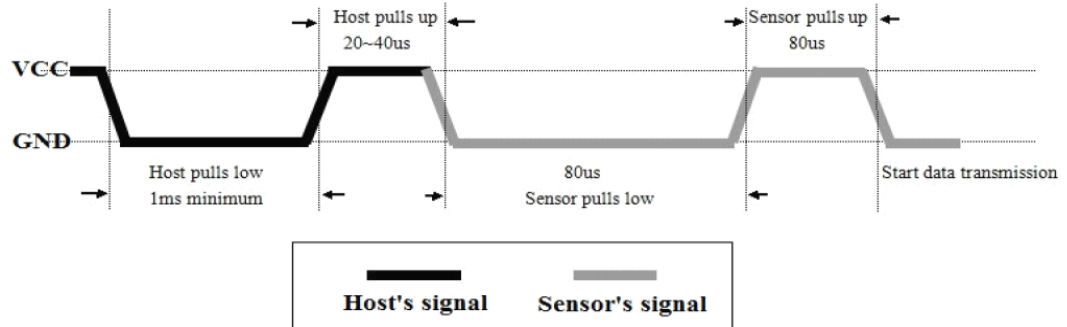
Mikrodenetleyici başlama sinyali gönderdiğinde, Dht22 data hattı HIGH konumdan LOW konuma geçer. Mikrodenetleyici başlangıç sinyalini göndermeyi tamamladığında, Dht22 bağıl nem ve sıcaklık bilgisini yansıtan 40 bitlik bir verinin yanıt sinyalini mikrodenetleyiciye gönderir. Mikrodenetleyiciden başlatma sinyali olmadan Dht22 mikrodenetleyiciye cevap sinyali vermez. Veri toplama işlemi, mikrodenetleyiciden tekrar başlatma sinyali alamazsa Dht22 data hattı HIGH konuma geçecektir [59].



Dht22 Haberleşme Diyagramı [59]

- **Mikrodenetleyicinin Sensöre Başlama Sinyali Göndermesi**

Tek data hattı başlangıçta HIGH konumundadır. Mikrodenetleyici başlama sinyali göndermeye başladığında tek data hattı LOW konumuna çekilir. Bu işlem sensörün başlama sinyali tespit edebilmesi için 1 ms devam eder. Daha sonra mikrodenetleyici data hattını HIGH konumuna çeker ve 20-40 us zaman aralığında bu konumda sensör cevabını bekler [59].

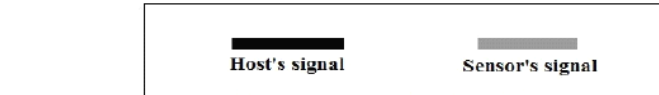
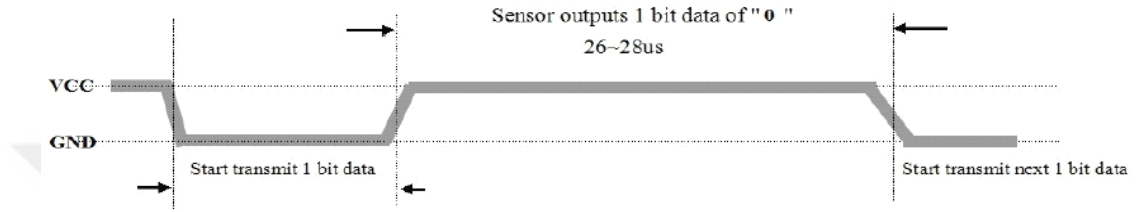


Mikrodenetleyicinin Başlama Sinyali Göndermesi [59]

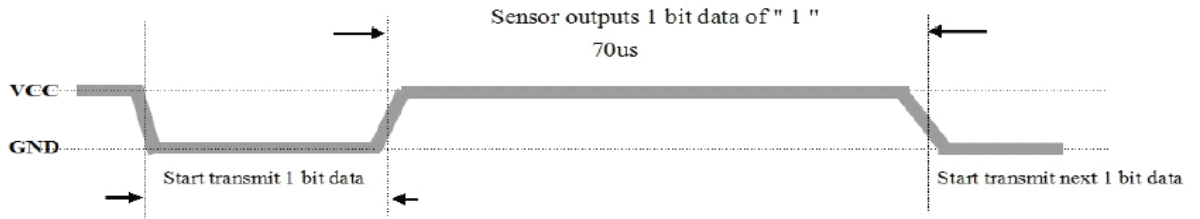
- **Sensörün Mikrodenetleyiciye Cevabı**

Dht22 başlatma sinyalini tespit ettiğinde, sensör LOW konuma geçerek cevap sinyali gönderir. Bu işlem yaklaşık 80 us zaman aralığında gerçekleşir. Daha sonra tek data hattı HIGH konumuna geçerek sensörün data gönderme işlemi hazırlıkları yapılır. Bu işlem de 80 us' de gerçekleşir.

Sensör mikrodenetleyiciye data göndermeye başladığında her bir bit gönderiminden önce tek data hattı 50 us boyunca LOW konumuna çekilir. Daha sonra her bir bit için tek data hattı HIGH konumuna getirilir. Bu konumdaki süresine göre mikrodenetleyici gelen bitin '0' ya da '1' olduğunu anlar. Bu konumdaki süre 26-28 us ise '0' biti, 70 us ise '1' biti gönderilir. Bu durum Şekil 3.10 ve 3.11' de daha rahat anlaşılmaktadır [59].



Sensörün Mikrodenetleyiciye Cevabı (0 Bitinin Gönderilmesi) [59]



Sensörün Mikrodenetleyiciye Cevabı (1 Bitinin Gönderilmesi) [59]

EK 3. ITG/M-04 MEDİKAL OKSİJEN SENSÖRÜ

ıtg/M-04 Medikal Oksijen Sensörü Teknik Özellikleri [60]

Ölçüm Aralığı	0 – 100 % Aralığı
Beklenen Çalışma Süresi	Yaklaşık 1.000.000 % h
Sensör Ömrü	< 6 Yıl Hava yapısı ve diğer uygulamalara bağlı olarak
Elektriksel Bağlantı	3 Pinli (VDD, Toprak, Analog çıkış pini)
Çıkış Sinyali	12,5 – 16,5 mV
Tepki Süresi	< 12s
Akıntı	< % 1
Çalışma Sıcaklığı	10 – 40 °C
Basınç Aralığı	700 – 1250 hPa
Doğrusallık Hatası	<= %3
Sıfır Offset Gerilimi	<= 0,3%
Tekrarlanabilirlik	+ - 1 %
Nem Etkisi	- 0,003 % her bir % RH
Tavsiye Edilen Yükleme	>= 10 kOhm

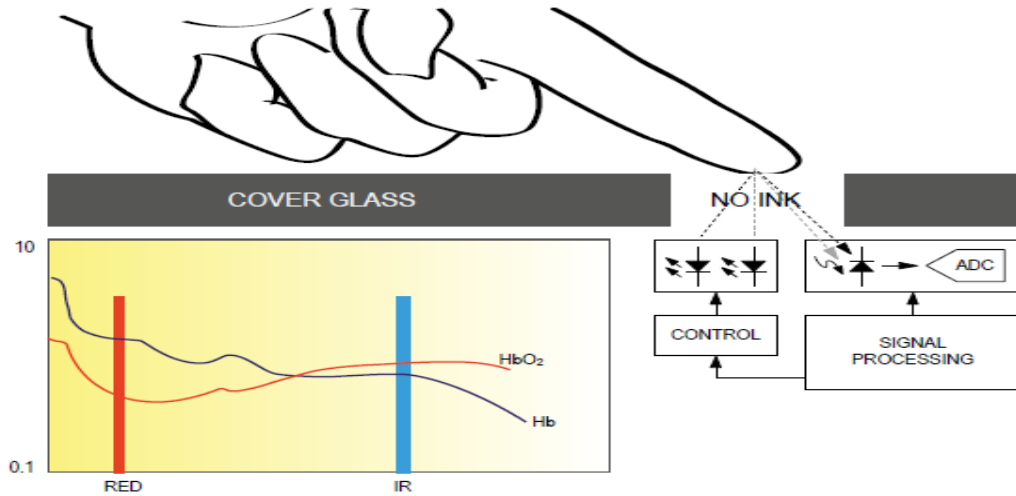
Direnci	
Sıcaklık Komponenti	NTC
Kullanılan Standart	DIN EN ISO 80601-2-55
Ağırlık	Yaklaşık 24 gr
Kullanılan Malzemeler	PA (Polyamid), PPS (Polifenilen Sülfid), PTFE (Polytetrafloretilen), Paslanmaz çelik

EK 4. MAX30100 ENTEGRESİ

Max30100 Entegresi

Max30100, pulse-oksimetre ve nabız sensörünü bir arada sunmaktadır. Bu sayede hem oksijen satürasyonu hem de nabız ölçümleri yapılabilir. İçerdiği kırmızı ve kızılötesi LED'lerden ışık yayarak ve bu ışığın yansımından ölçüm olarak çalışır [61].

Bu özel LED renk kombinasyonu, parmak ucundan verileri okumak için optimize edilmiştir. Sinyal, düşük gürültülü bir analog sinyal işleme ünitesi tarafından işlenir ve mikroBUS I²C arabirimi aracılığıyla hedef mikroişlemciye iletilir. Aşırı hareket ve sıcaklık değişimleri ölçümleri olumsuz etkileyebilir. Ayrıca entegre üzerine çok fazla basınç, kılcal kan akışını sınırlayabilir ve bu verilerin güvenilirliğini düşürebilir. Entegrede programlanabilir bir INT pini de mevcuttur [66].



Max30100 Sistem Blok Diyagramı [66]

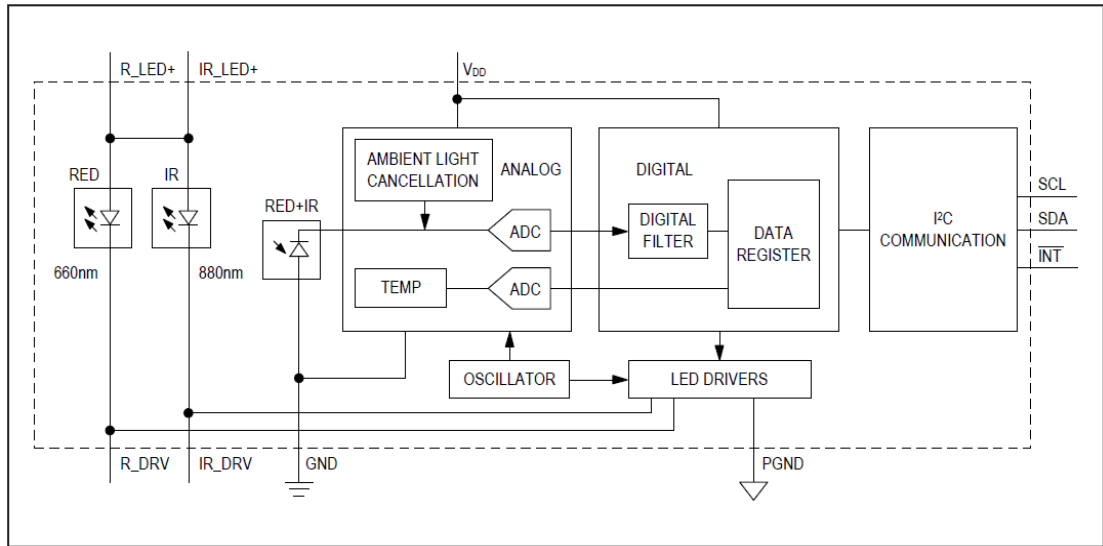
Max30100, giyilebilir cihazların zorlu gereksinimleri için tasarlanmış eksiksiz bir nabız oksimetresi ve kalp atış hızı sensörü sistemi çözümüdür. Max30100 optik veya elektriksel performanstan ödün vermeden, toplamda çok küçük bir çözüm boyutu sunar. Giyilebilir bir cihaza entegrasyon için minimum harici donanım bileşenleri gerekir [66].



Max30100 Entegresi

Max30100 yazılım kayıtları üzerinden tamamen yapılandırılabilir ve dijital çıkış verileri cihaz içindeki 16 bitlik bir FIFO registerında saklanır [66].

Max30100 Entegresi Bileşenleri



Max30100 Sistem Bileşenleri [66]

- **SpO₂ Bileşeni**

Max30100'deki SpO₂ alt sistemi, ortam ışığı kesici (ALC), 16 bit sigma delta analog dijital çevirici (ADC) ve özel ayırık zaman filtresinden oluşur [66].

SpO₂ bileşimindeki ADC, 16 bit çözünürlüğe kadar sürekli zaman aşımına uğrayan sigma delta dönüştürücüsüdür. ADC çıkış veri hızı, 50Hz ile 1kHz arasında programlanabilir. Max30100, 50Hz / 60Hz şehir şebekesi parazitini ve

düşük frekanslı ortam gürültüsünü engellemek için özel bir ayırık zaman filtresi içerir [68].

- **Sıcaklık Sensörü**

Max30100, SpO₂ alt sisteminin sıcaklık bağımlılığını (isteğe bağlı olarak) kalibre etmek için çip üzerinde bir sıcaklık sensörüne sahiptir [66].

SpO₂ algoritması, IR LED' in dalga boyuna nispeten duyarsızdır, ancak kırmızı LED' in dalga boyu, verilerin yorumunu düzeltmek için kritik öneme sahiptir. Sıcaklık sensörü verileri, SpO₂ hatasını ortam sıcaklığı değişiklikleriyle telafi etmek için kullanılabilir [66].

- **Led Sürücüsü**

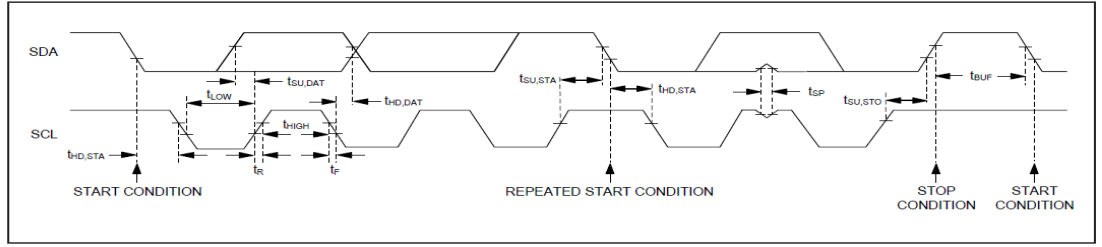
Max30100, SpO₂ ve kalp atım ölçümleri için LED palslarını çalıştıran kırmızı ve IR LED sürücülerini içerir. LED akımı, uygun besleme gerilimi ile 0 mA'dan 50 mA'ya (sadece tipik) kadar programlanabilir. LED darbe genişliği, kullanım durumlarına göre ölçüm doğruluğunu ve güç tüketimini optimize etmek için 200 mikro saniyeden 1,6 milisaniyeye kadar programlanabilir [66].

Max30100 I²C İletişimi

Max30100, seri veri hattı (SDA) ve seri clock hattından (SCL) oluşan I²C / SMBus uyumlu, 2 kablolu seri arabirime sahiptir. SDA ve SCL, Max30100 ile master arasında 400 kHz' e kadar clock darbe hızlarıyla iletişimi kolaylaştırır. Şekil 3.16' da iki hatlı ara yüz zamanlama diyagramı gösterilmektedir. Master SCL'yi üretir ve veri yolunda veri aktarımını başlatır. Master cihaz uygun slave adresini ve ardından verileri ileterek Max30100'e veri yazar. Her bir iletim dizisi, bir START (S) ya da REPEATED START (Sr) şartı ve bir STOP (P) şartı ile çerçevesizdir (frame oluşur). Max30100'e her bilgi 8 bit uzunluğunda iletilir ve bunun ardından onaylı bir clock darbesi gönderilir. Master' ın Max30100' de veri okuması uygun slave adresine dokuz SCL darbesi ardından iletilir [66].

Max30100 master tarafından üretilen SCL darbeleriyle senkronize olarak SDA' ya veri iletir. Master her veri baytının alındığını onaylar. Her okuma dizisi bir START (S) ya da REPEATED START (Sr) koşulu, onaylanmamış ve bir STOP (P) koşulu

ile çerçevenlenmiştir. SDA hem giriş hem de open-drain çıkışı olarak çalışır. SDA' da tipik olarak 500 Ω ' dan büyük bir pull-up direnci gerekir. SCL sadece bir giriş olarak çalışır [63]. Tasarlanan sistemde SDA, SCL ve INT pinlerine 4,7 k Ohm değerinde pull-up dirençleri bağlanmıştır.



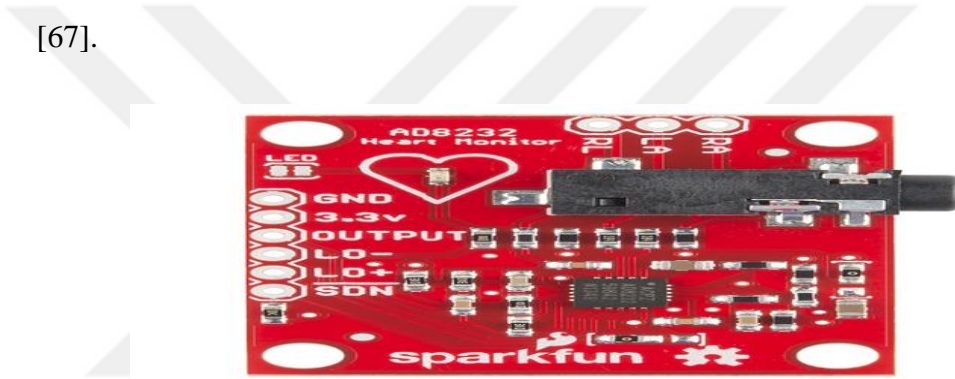
Max30100 I²C İletişimi Sistem Diyagramı [66]

EK 5. AD8232 ENTEGRESİ

Ad8232 Entegresi

Ad8232, kalbin elektriksel aktivitesini ölçmek için kullanılan zarif, küçük bir çiptir. Ad8232 entegresiyle kalpte elektriksel aktivite sonucu oluşan analog sinyaller grafik haline getirilerek EKG veya elektrokardiyogram şeklinde çıktılar elde edilebilir [67].

Ad8232 EKG ve diğer biyo-potansiyel ölçüm uygulamalarında kullanılabilir. Bu komponent küçük, gürültülü biyo-potansiyel sinyalleri çıkarmak, büyötmek ve filtrelemek için tasarlanmıştır. Sinyallerin tespitini ise elektrotlar vasıtasıyla yapar [67].



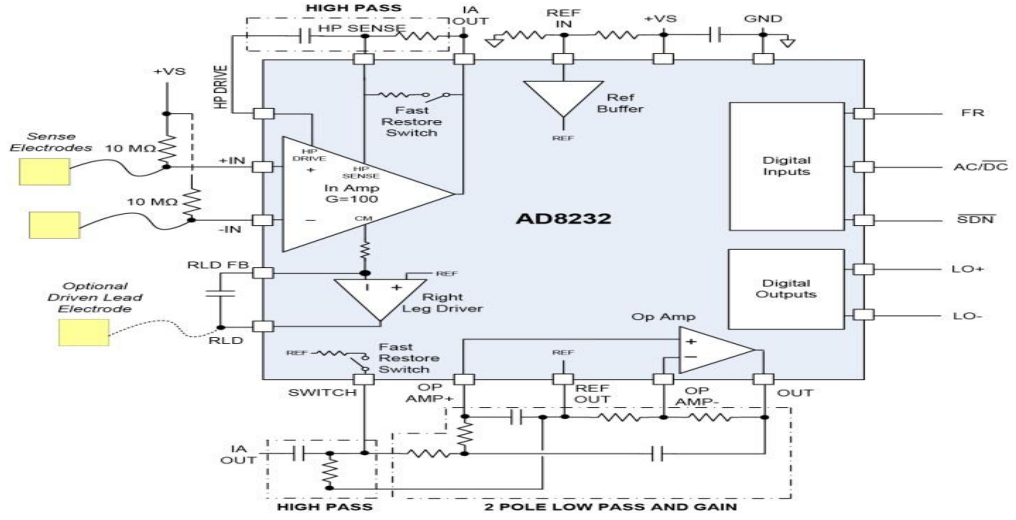
Ad8232 Kalp Atış Hızı Sensörü [62]

Ad8232 sensöründe elektrot yarı hücre potansiyelini ve sinyal gürültülerini engellemek için ikinci dereceden yüksek geçirgen filtre kullanılmıştır. Bu filtre, yükseltecin hem yüksek kazanç hem de yüksek geçişe izin vermesi için tek aşamalı filtre içerisine, enstrümantasyon mimarisıyla sıkıca birleştirilmiştir. Böylece yer ve maliyet tasarrufu sağlanmıştır [67].

Ad8232, sistemdeki hat frekanslarının ve diğer istenmeyen gürültülerin ortak mod reddini gerçekleştirmek için sağ bacak sürücü gibi sürücü yönetim yükselteçi içerir [67].

Ad8232 diğer yüksek geçişli filtrelerin aksine uzun işlem sürelerini azaltan hızlı bir geri yükleme fonksiyonu içerir. Yükselticiyi harekete geçiren ani bir sinyal değişiminden sonra Ad8232 otomatik olarak daha yüksek bir filtre kesmesine ayar

yapar. Bu özellik Ad8232' nin hızlı bir şekilde resetlenmesini ve bu sayede elektrotları vücuda bağladıktan hemen sonra geçerli ölçümler almasını sağlar [67].



Ad8232 Blok Diyagram [68]

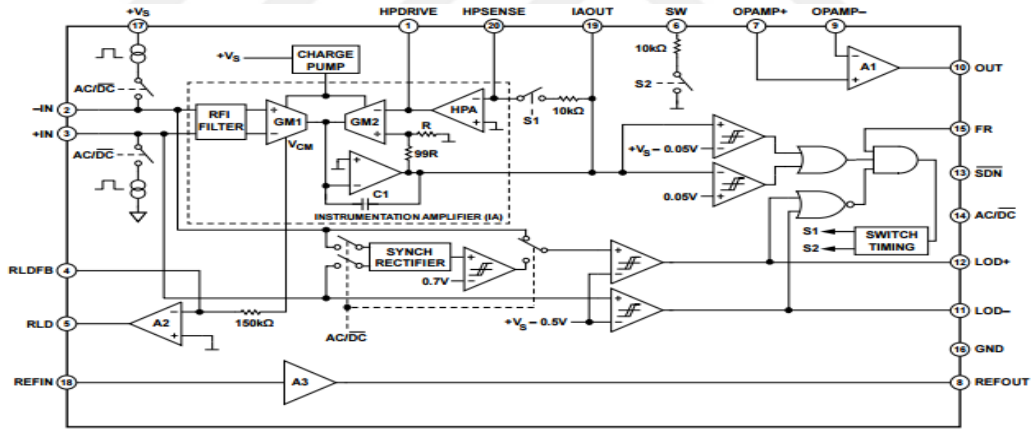


Figure 45. Simplified Schematic Diagram

Ad8232 Sistem Şematığı[67]

Ad8232 Çalışma Yapısı

Genel olarak 4 ayrı fonksiyonu olan Ad8232'nin basitleştirilmiş bir devre şeması Şekil 3.20' de gösterilmektedir. Bir enstrümantasyon yükselteci, düşük geçişli filtrelemeyi destekleyen bir yükselteç aşaması, bir sağ bacak sürücüsü (RLD) yükselteci ve bir çip üzerinde referans tamponu içerir [68].

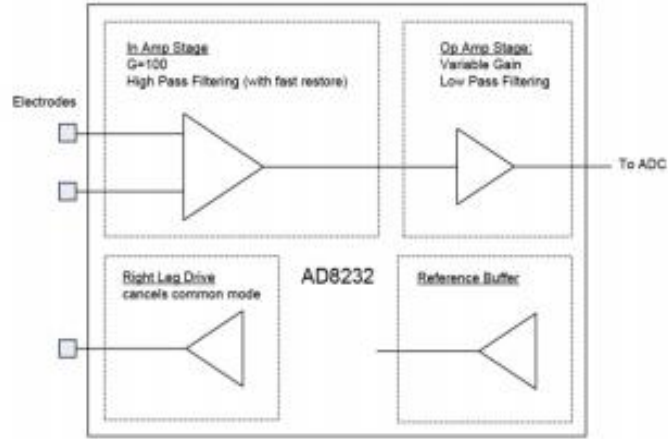


Figure 2. Simplified Block Diagram of the AD8232

Ad8232 Basit Blok Diyagramı [68]

Sinyal zincirindeki temel fonksiyon enstrümantasyon yükseltecidir. Dengeli giriş aşaması en az 80 dB olan ortak mod reddetme oranını sağlar. Vücuttaki elektrotlar doğrudan, 100 m Ω 'lık bir giriş empedansına sahip olan bu yükseltecin yüksek empedanslı giriş düğümlerine bağlanır. Girdiye uygulanan küçük sinyalleri yükseltmek için bu aşamada 100 V / V'lik sabit bir kazanç sağlayan entegre dahil edilmiştir. Bu yüksek kazanç karşı DC ofset kolayca doygun hale gelebildiğinden giriş yükselticisi, yapay voltaj değişimlerini filtrelemeye ilaveten elektrotların yarı hücre potansiyelini engelleyen 2 kutuplu bir yüksek geçiren filtre ile birleştirilmiştir [68].

İkinci aşamada, sinyale 11 V / V ek bir kazanç ekleyen, toplam kazancı 1100 V / V olarak ayarlayan ve maksimum diferansiyel giriş sinyalini 2.7 mV p-p ile sınırlayan bir işlemsel yükselteç bulunur. Bununla birlikte cihaz aşırı gerilime karşı korunur, böylece maksimum seviyeyi aşan giriş gerilimleri çıkışı bozar, ancak parçaya zarar vermez. İşlemsel yükselteçte, hat gürültüsünü ve ortak mod sinyallerini gidermek için 2 kutuplu alçak geçiren filtre kullanılır. Çıkış sinyali, Ad8232 'nin girişinde sağlanan EKG sinyalinin gürültüden arındırılmış, güçlendirilmiş bir versiyonunu temsil eder. Sinyalin nasıl işlendiğine bağlı olarak, bir “kalp atış hızı monitörü” cihazı olarak kullanılabilir [68].

Ad8232 düşük güç, tek besleme işlemi için tasarlanmıştır. Bu nedenle toprak bağlantısının, sabit bir referans seviyesine (sahte toprak), tercihen orta ölçekli veya sinyal zincirinden daha aşağıdan kullanılan ADC (Analog Dijital Çevirici)' nin ortak mod voltaj seviyesine yükseltilmesi gerekir. Entegre referans yükselteci, Ad8232 içindeki ortak referans noktasını ayarlamak için bu voltajı tamponlar. Referansı tercih edilen herhangi bir seviyeye ayarlamak için iki direnç kullanılabilir veya ADC' den gelen referans voltaj çıkışı kullanılabilir. ADC' nin dahili bir referans gerilimi yoksa, Ad8232' nin referans çıkışı ADC' ye referans sağlamak için kullanılabilir. Bu oranlı ölçüm yüksek doğruluk sağlar [68].

Ad8232 Entegresi Filtrelemenin Önemi

Vücuttan elde edilen sinyaller zayıf ve gürültülüdür. Bu nedenle filtreleme yararlı bir sinyal elde etmek için önemlidir. Giriş yükselticisi aynı anda kazanç ve yüksek geçişli filtreleme sağlar. Küçük EKG sinyallerine 100 V / V kazanç eklerken, ± 300 mV kadar yüksek olabilen elektrot offsetlerini bastırır [68].

İlk filtrenin -3 dB kesme frekansı $FHP1 -3 \text{ dB} = 100 / (2\pi R1C1)$ ' dir. Yükseltecin kazanç ve geri bildirim mimarisi nedeniyle, filtre kesmesi ondan beklenenden 100 kat daha yüksektir. Şekil 3.21' de optimum sinyal aktarımı için yükselteçlerin ve çeşitli filtre aşamalarının devre şeması gösterilmiştir [68].

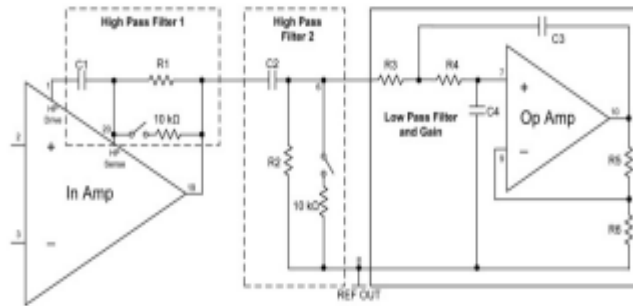


Figure 3. Filtering Section in the AD8232

Ad8232 Filtreleme İşlemi [68]

İlk aşamayı, AC-eşli ağ takip eder. Kesme normal bir birinci dereceden yüksek geçişli filtre, $FHP2 -3 \text{ dB} = 1 / (2\pi R2C2)$ olarak ayarlanmıştır. İki kombine yüksek geçişli filtre, her onluk devrede bir toplam -40 dB geri dönüş sağlar [68].

Bir sonraki filtre yapay voltaj değişimlerini kaldırmak ve temel çizgiyi düzeltmek için önemlidir. Önerilen kesim frekansı 25 Hz civarındadır. Bu yükselteç kademesinin çıktısı + VS ve topraktan 100 mV içinde salınım yapabilir. Böylece Ad8232' yi izleyen neredeyse ADC' nin tüm dinamik aralığı kullanılabilir; bu durumda eğer ADC ve Ad8232 aynı güç kaynağından besleniyorsa besleme gerilimi en üst düzeye çıkar [68].

Ad8232 Entegresi Sağ Bacak Sürücüsü (RLD)

Sağ bacak sürücüsü adı verilen bu yükselteç, üçüncü elektrotla bağlantı sağlamak için entegre edilmiştir. RLD, hasta / kullanıcı ve Ad8232 arasındaki ortak voltajı uygun hale getirmek için kullanılabilir. RLD, Ad8232' nin performansını optimize etmeye yardımcı olur, ancak zorunlu bir işlev değildir. RLD bağlantısı, uygulamayı şebekeden gelen 50 Hz / 60 Hz gürültüden, güç kaynaklarından, LED ışıklardan, flüoresan lambalardan ve diğer aydınlatma sistemlerinden gelen radyasyon vb. sinyal bozucu aktivitelerden korur [68].

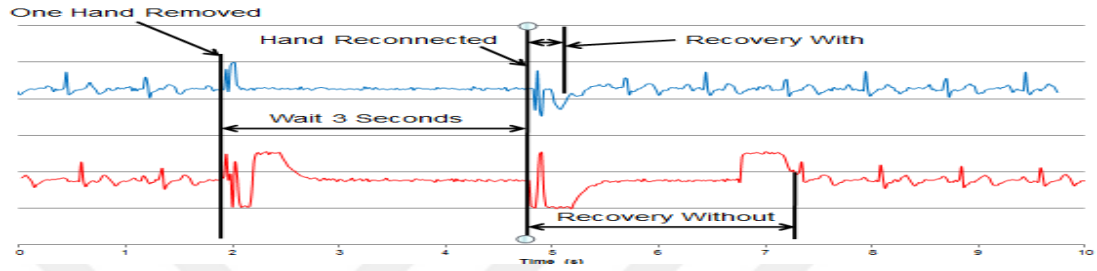
Ad8232 Çıkış Tespiti ve Hızlı Yenilenme Modu

Ad8232, bir elektrotun hastadan veya kullanıcıdan ayrıldığını izler. Kullanıcı seçenekleri DC veya AC çıkış saptama tespitine izin verir. DC çıkış algılama modunda, sistem elektrotların giriş potansiyelini algılar. Elektrot girişlerinden biri yükseldiğinde, ilgili LO- veya LO + pin operatörü veya kullanıcıyı uyararak için işaret verir. Ayrıca, Ad8232' nin sadece iki elektrotla (RLD' siz) çalıştığı durumlarda AC çıkış algılama modu seçilebilir. Cihaz, elektrotlara 100 kHz' lik küçük bir uyarım akımı elde ederek bir elektrotun ne zaman ayrıldığını tespit eder. Ancak iki elektrot kullanıldığında Ad8232 hangi elektrotun vücuttan ayrıldığını tespit edemez. AC çıkış tespiti sırasında yalnızca LO + pini işaretlenir [68].

Ad8232' deki diğer bir önemli özellik hızlı geri yükleme modudur. Bunu desteklemek için Ad8232 uçlardan birinin girişinde ani bir değişiklik olduğunda

otomatik olarak daha yüksek bir filtre kesme ayarına geçer. Bu özellik cihazın daha hızlı resetlenmesini ve geçerli ölçümler yapmasını sağlar [68].

Sinyal yerleştiğinde, filtre hareketi sistemin genel gürültü davranışını iyileştirmek için otomatik olarak değişir. Şekil 3.22’ de etkin hızlı geri yükleme modu olan ve olmayan çözüm durumlarını göstermektedir.



Ad8232'nin Hızlı Geri Yükleme Modu Olan ve Bulunmayan Etkinleştirilmesi [68]

EK 6. YAPILAN ÇALIŞMADAKİ ARDUINO KODLARI

```
#include <Wire.h>

#include <dht.h>           //Arduino' ya gerekli kütüphaneler ekleniyor.

#include <EEPROM.h>

#include "MAX30100_PulseOximeter.h"

#define REPORTING_PERIOD_MS   1000

#define pin 3

dht DHT;

PulseOximeter pox;

uint32_t tsLastReport = 0;

void onBeatDetected()

const float VRefer = 3.3;

const int pinAdc  = A1;

void setup()

{

    Serial.begin(115200); // Seri haberleşme hızı 115200 kb/sn olarak ayarlanıyor.

    if (!pox.begin()) {

        for(;;) }

    pox.setOnBeatDetectedCallback(onBeatDetected);

}

void loop()
```

```
{  
  
//Programda kullanılacak deęişkenler atanıyor.  
  
int x;  
  
int deger;  
  
int deger1;  
  
int deger2;  
  
int kayitAdresi , kaydedilecekVeri;  
int okunacakAdres, okunanVeri;  
  
int kayitAdresi1 , kaydedilecekVeri1;  
int okunacakAdres1, okunanVeri1;  
  
int kayitAdresi2, kaydedilecekVeri2;  
int okunacakAdres2, okunanVeri2;  
  
int kayitAdresi3 , kaydedilecekVeri3;  
int okunacakAdres3, okunanVeri3;  
  
int kayitAdresi4 , kaydedilecekVeri4;  
int okunacakAdres4, okunanVeri4;  
  
float Vout =0;  
  
Vout = readO2Vout();
```



```
int readData = DHT.read22(pin);
```

```
pox.update();
```

```
//Sensörlerden gelen ölçüm verileri aralarına * işareti konarak paketler şeklinde gönderiliyor
```

```
Serial.print(DHT.humidity);
```

```
Serial.print('*');
```

```
Serial.print(DHT.temperature);
```

```
Serial.print('*');
```

```
Serial.print(readConcentration()); .
```

```
Serial.print('*');
```

```
Serial.print(pox.getHeartRate());
```

```
Serial.print("*");
```

```
Serial.print(pox.getSpO2());
```

```
Serial.print("*");
```

```
// Ara yüzden gelen ortam parametre komutları parametrenin cinsine göre hafızaya kaydediliyor.
```

```
if((digitalRead(10) == 1)||((digitalRead(11) == 1)){
```

```
Serial.print('0');
```

```
}
```

```
else{
```

```
Serial.print(analogRead(A0));
```

```
}
```

```
Serial.println();
```

```

delay (1);

if (Serial.available() ) {

    x = Serial.readString().toInt();

    if(x==1 || x==2 || x==3){

        kayıtAdresi1 = 11;

        kaydedilecekVeri1 = x;

        EEPROM.write(kayıtAdresi1, kaydedilecekVeri1);

        delay(10);

    }

    if (x>3){

        okunacakAdres1 = 11;

        okunanVeri1 = EEPROM.read(okunacakAdres1); // Bağıl nem verilerinin
kaydedildiği adres

        if(okunanVeri1==1){

            deger = x;

            kayıtAdresi4 = 10;

            kaydedilecekVeri4 = deger;

            EEPROM.write(kayıtAdresi4, kaydedilecekVeri4);} // Sıcaklık verilerinin
kaydedildiği adres

        if (okunanVeri1==2){

            deger1 = x;

            kayıtAdresi2 = 9;

            kaydedilecekVeri2 = deger1;

```

```
EEPROM.write(kayitAdresi2, kaydedilecekVeri2); // O2 verilerinin  
kaydedildiği adres
```

```
}
```

```
if (okunanVeri1==3){
```

```
deger2 = x;
```

```
kayitAdresi3 = 8;
```

```
kaydedilecekVeri3 = deger2;
```

```
EEPROM.write(kayitAdresi2, kaydedilecekVeri3);
```

```
}
```

```
delay(1000);
```

```
}
```

```
}
```

```
if (Serial.available()> 0) { //Hafızadaki ortam verileri okunuyor.
```

```
okunacakAdres2 = 11;
```

```
okunanVeri2 = EEPROM.read(okunacakAdres2);
```

```
if(okunanVeri2 ==1){
```

```
okunacakAdres4 = 10; // 10 adresini okuyacağız
```

```
okunanVeri4 = EEPROM.read(okunacakAdres4);
```

```
deger = okunanVeri4; //}
```

```
if(okunanVeri2 ==2) {
```

```
okunacakAdres2 = 9; // 9 adresini okuyacağız
```

```
okunanVeri2 = EEPROM.read(okunacakAdres2);
```

```

deger1 = okunanVeri2;//}

if(okunanVeri2 ==3){

okunacakAdres3 = 8; // 8 adresini okuyacağız

okunanVeri3 = EEPROM.read(okunacakAdres2);

deger2 = okunanVeri3;//}

delay(1000);

}

// Ara Yüzden gelen veriler ve sensörlerdem gelen veriler karşılaştırılarak gerekli
çıkışlar HIGH ya da LOW yapılıyor.

if (DHT.humidity<deger)
{

digitalWrite(7,HIGH);

digitalWrite(6,LOW);

}

if (DHT.humidity>=deger){

digitalWrite(6,HIGH);

digitalWrite(7,LOW);}

if (DHT.temperature<deger1){

digitalWrite(9,HIGH);

digitalWrite(8,LOW);

}

if (DHT.temperature>=deger1){

```

```

digitalWrite(8,HIGH);

digitalWrite(9,LOW); }

if (readConcentration()<deger2){

digitalWrite (5, HIGH);}

if (readConcentration()>=deger2){

digitalWrite (5, LOW);}

}

float readO2Vout() // O2 sensöründen gelen analog verilerin sayısal hale getiren alt
program.

{

long sum = 0;

for(int i=0; i<32; i++)

{

sum += analogRead(pinAdc);

}

sum >>= 5;

float MeasuredVout = sum * (VRefer / 1023.0);

return MeasuredVout;

}

float readConcentration() // O2 konsantrasyonu hesaplayan alt program

{

float MeasuredVout = readO2Vout();

```

```
float Concentration = MeasuredVout * 0.21 / 2.0;  
  
float Concentration_Percentage=Concentration*100;  
  
return Concentration_Percentage;  
  
}
```



EK 7. TERMOGRAM ANALİZİ ARA YÜZÜ PROGRAM KODLARI

```
//Program için gerekli kütüphaneler yükleniyor.

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Windows.Forms;
using AForge;
using AForge.Imaging;
using System.Drawing.Imaging;
using AForge.Imaging.ComplexFilters;
using AForge.Imaging.ColorReduction;
using AForge.Imaging.Filters;
using AForge.Imaging.Textures;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}

// Programda kullanılacak değişkenler atanıyor.
private bool _selecting;
private Rectangle _selection;
private bool _selecting1;
private Rectangle _selection1;
bool ciz;

double sic;
double sic1;

int baslaX;
int baslaY;

int n;
int t;
int r;

int x;
int y;
int z;

int a, b;
int u;
int w;
int q;
```

```

int i, j;
int il, jl;
int dl;
int c = 255;
int d = 255;
int f = 255;
double min;
double max;
double ort;

int[] kirmizi = new int[256];
int[] yesil = new int[256];
int[] mavi = new int[256];

private void trackBar_Scroll(object sender, EventArgs e) // Sıcaklık değerinin
ayarlandığı trackBar nesnesi
{
a = trackBar7.Value; // trackBar nesnesinden seçilen değer a değişkenine
atanır.

label37.Text = a.ToString(); //trackBar nesnesinin değerini gösterir.

if (textBox1.Text == "") //Sıcaklık aralığı girilmediğinde program hata
verecektir.
{
MessageBox.Show("Sıcaklık Aralığı Giriniz");
}

else if (textBox1.Text != "")
{

min = (float)(Math.Round(Convert.ToDouble(textBox1.Text)));
max = (float)(Math.Round(Convert.ToDouble(textBox2.Text))); // textBoxlara
girilen termal kameraların en düşük ve en yüksek görüntülediği sıcaklık
değerleri
sic = (1785 * min + (max - min) * a)/1785; // Sıcaklık - trackBar nesnesinin
değerinin eşleştiği fonksiyon
ort = (max - min)/8;
label17.Text = textBox1.Text;
label33.Text = textBox1.Text;
label9.Text = textBox2.Text;
label31.Text = textBox2.Text;
label16.Text = (min + ort).ToString();
label11.Text = (min + ort).ToString();
label15.Text = (min + 2*ort).ToString();
label25.Text = (min + 2 * ort).ToString();
label14.Text = (min + 3*ort).ToString();
label26.Text = (min + 3 * ort).ToString();
label13.Text = (min + 4 * ort).ToString();
label27.Text = (min + 4 * ort).ToString(); // textBoxlara girilen min ve mak.
sıcaklık değerlerine göre ölçek aralığının yeniden hesaplanması
label12.Text = (min + 5*ort).ToString();
label28.Text = (min + 5 * ort).ToString();
label11.Text = (min + 6*ort).ToString();
label29.Text = (min + 6 * ort).ToString();
label10.Text = (min + 7*ort).ToString();
label30.Text = (min + 7 * ort).ToString();
}
}

```



```

label23.Text = sic.ToString("N1") + " oC";
// q = mavi renk yoğunluğunu
// u = yeşil renk yoğunluğunu
// w = kırmızı renk yoğunluğunu gösterir.
if (a <= 255)
{
q = a;
u = 0;
w = 0;

}
if (a >= 256 && a <= 510)
{
q = 255;
u = a-255;
w = 0;

}
if (a >= 511 && a <= 765)
{

q = 765-a;
u = 255;
w = 0;
}

if (a > 765 && a <= 1020)
{
q = 0;
u = 255;
w = a-765;
}

if (a > 1020 && a <= 1275)
{
q = 0;
u = 1275-a;
w = 255;
}

if (a > 1275 && a <= 1530)
{
q = a-1275;
u = 0;
w = 255;
}

if (a > 1530 && a <= 1785)
{
q = 255;
u = a-1530;
w = 255;
}

}
private void pictureBox4_Click(object sender, EventArgs e)
//RGB renk - sıcaklık picutreBox
{

pictureBox4.Image = new Bitmap(pictureBox4.Width, pictureBox2.Height);

```

```

int i, j;
f = 255;
d = 255;
c = 255;
for (j = 0; j < 36; j++)
{

d = d - 7;

if (d < 0) { d = 0; }
if (c < 0) { c = 0; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(f, d,
c));

}}

d = 0;
f = 255;
c = 255;
for (j = 36; j < 72; j++)
{

c = c - 7;

if (d < 0) { d = 0; }
if (c < 0) { c = 0; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(f, d,
c));

}

}

f = 255;
d = 0;
c = 0;
for (j = 72; j < 108; j++)
{
d = d + 7;

if (d > 255) { d = 255; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(f, d,
0));
}
}

```

```

}}
f = 255;
c = 0;

for (j = 108; j < 144; j++)
{

f = f - 7;
if (f < 0) { f = 0; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(f,
255, c));

}}

f = 0;
d = 255;
c = 0;
for (j = 144; j < 180; j++)
{

c = c + 7;
if (c > 255) { c = 255; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(0, d,
c));

}

}
d = 255;
f = 0;
for (j = 180; j < 216; j++)
{

d = d - 7;
if (d < 0) { d = 0; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(f, d,
255));

}
}
c = 255;
f = 0;
for (j = 216; j < 255; j++)
{

```

```

c = c - 7;

if (c < 0) { c = 0; }
if (f < 0) { f = 0; }

for (i = 0; i < 30; i++)
{

((Bitmap)pictureBox4.Image).SetPixel(i, j, System.Drawing.Color.FromArgb(f, 0,
c));

}}

private void openToolStripMenuItem_Click(object sender, EventArgs e) //Termal
görüntü yükleme
{
if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
{

pictureBox1.Image =
(Bitmap)System.Drawing.Image.FromFile(openFileDialog1.FileName);

}

pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage; //Görüntü Ana Resim
Bölümünün boyutlarına getiriliyor.
Bitmap orjinal = new Bitmap(pictureBox1.Image, 175, 255);
pictureBox1.Image = orjinal;
pictureBox1.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void button1_Click(object sender, EventArgs e)
//Blob Counter Fonksiyonu
{
if ((Bitmap)pictureBox2.Image != null)
{

// filtre oluşturuluyor.
EuclideanColorFiltering filtre = new EuclideanColorFiltering();
// Merkez pikseller ve RGB küresi yarıçapı belirleniyor.
filtre.CenterColor = new RGB(Color.FromArgb(w, u, q));
filtre.Radius = 100;
// filtre uygulanıyor.
pictureBox2.Image = filtre.Apply((Bitmap)pictureBox1.Image);
pictureBox11.Image = filtre.Apply((Bitmap)pictureBox1.Image);

BlobCounter blobCounter = new BlobCounter();

blobCounter.MinWidth = 5;
blobCounter.MinHeight = 5;
blobCounter.FilterBlobs = true;
blobCounter.ObjectsOrder = ObjectsOrder.Size;
blobCounter.ProcessImage((Bitmap)pictureBox2.Image);
Rectangle[] rects = blobCounter.GetObjectsRectangles();
foreach (Rectangle recs in rects)
if (rects.Length > 0)

```

```

{
foreach (Rectangle objectRect in rects) //Seçili Alanlar kare içine alınıyor.
{
Graphics g = Graphics.FromImage((Bitmap)pictureBox11.Image);

using (Pen pen = new Pen(Color.FromArgb(255, 255, 255), 3))
{
g.DrawRectangle(pen, objectRect);
}

int objectX = objectRect.X + (objectRect.Width / 2);
int objectY = objectRect.Y + (objectRect.Height / 2);

g.Dispose();

if (button1.Enabled)
{
this.Invoke((MethodInvoker)delegate //Seçili Alanların koordinatları yazdırılıyor.
{ richTextBox1.Text = "Seçilen Alan Koordinatları" + "\n" +
objectRect.Location.ToString() + "\n" + richTextBox1.Text + "\n"; });
}}
pictureBox11.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap orjinal1 = new Bitmap(pictureBox11.Image, 338, 362); //Seçili alanlar
Blob Counter bölümünün boyutlarına uygun hale getirilerek görüntüleniyor.
pictureBox11.Image = orjinal1;
pictureBox11.SizeMode = PictureBoxSizeMode.CenterImage;
}

else MessageBox.Show("Resim Yükleyiniz"); //hata mesajı
}

private void button2_Click(object sender, EventArgs e) //RGB Filtre Fonksiyonu
{
if ((Bitmap)pictureBox1.Image != null)
{
// filtre oluşturuluyor
EuclideanColorFiltering filtre1 = new EuclideanColorFiltering();
// merkez alınıp filtrelenecek renk bilgisindeki pikseller ayarlanıyor.
filtre1.CenterColor = new RGB(Color.FromArgb(w, u, q));
// filtrelenecek piksel küresinin yarıçapı
filtre1.Radius = 100;
// filtre uygulanıyor.
pictureBox2.Image = filtre1.Apply((Bitmap)pictureBox1.Image);
}

else MessageBox.Show("Resim Yükleyiniz");
}
private void kuvözToolStripMenuItem_Click(object sender, EventArgs e)
//Formlar arası geçiş yapılıyor.
{
Form2 göster = new Form2();
göster.Show();
this.Hide();
}

```

```

}

private void bebekToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form3 göster1 = new Form3();
    göster1.Show();
    this.Hide();
}

private void denemeToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 göster2 = new Form4();
    göster2.Show();
    this.Hide();
}

private void deneme1ToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form5 göster3 = new Form5();
    göster3.Show();
    this.Hide();
}

private void birleşmeToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 göster4 = new Form4();
    göster4.Show();
    this.Hide();
}

private void button12_Click(object sender, EventArgs e) // Kayıt fonksiyonu
{

    Bitmap bit = new Bitmap(pictureBox2.Image);

    string kayit = @"C:\Users\kkilicarslan\Pictures\c#\1.png";

    bit.Save(kayit);
}

private void pictureBox2_MouseDown(object sender, MouseEventArgs e)
//Filtrelenmiş resimlerden mouse sayesinde parçalar seçiliyor.
{
    // Starting point of the selection:
    if (e.Button == MouseButtons.Left)
    {
        _selecting = true;
        _selection = new Rectangle(new System.Drawing.Point(e.X, e.Y), new Size());
    }
}

private void pictureBox2_MouseMove(object sender, MouseEventArgs e)
{
    // Update the actual size of the selection:
    if (_selecting)
    {

```

```

_selection.Width = e.X - _selection.X;
_selection.Height = e.Y - _selection.Y;

// Redraw the picturebox:
pictureBox2.Refresh();
}}

private void pictureBox2_Paint(object sender, PaintEventArgs e)
{
if (_selecting)
{
// Draw a rectangle displaying the current selection
Pen pen = Pens.GreenYellow;
e.Graphics.DrawRectangle(pen, _selection);
}}

private void pictureBox2_MouseUp(object sender, MouseEventArgs e)
{
_selecting = false;
}

private void button4_Click(object sender, EventArgs e) // RGB filtrelenmiş
görüntü Seç komutu
{

// Seç komutu ile seçilen parçalar küçük kutucuklara ve büyük kutucuğa
boyutları değiştirilerek görüntüleniyor.

Bitmap b = (Bitmap)pictureBox2.Image;
Bitmap bm = b.Clone(_selection, b.PixelFormat);
if (bm != null)
{
pictureBox3.Image = bm;
pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap bmpKucuk = new Bitmap(pictureBox3.Image, 342, 243);
pictureBox3.Image = bmpKucuk;
pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;

Bitmap v1 = (Bitmap)pictureBox5.Image;
Bitmap v2 = (Bitmap)pictureBox6.Image;
Bitmap v3 = (Bitmap)pictureBox7.Image;
Bitmap v4 = (Bitmap)pictureBox8.Image;
Bitmap v5 = (Bitmap)pictureBox9.Image;
Bitmap v6 = (Bitmap)pictureBox10.Image;

if (v1 == null)
{

pictureBox5.Image = bm;
pictureBox5.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap kucuk1 = new Bitmap(pictureBox5.Image, 114, 77);
pictureBox5.Image = kucuk1;
pictureBox5.SizeMode = PictureBoxSizeMode.CenterImage;

}

else if (v2 == null)
{

pictureBox6.Image = bm;

```

```

pictureBox6.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap kukuk2 = new Bitmap(pictureBox6.Image, 114, 77);
pictureBox6.Image = kukuk2;
pictureBox6.SizeMode = PictureBoxSizeMode.CenterImage;

}
else if (v3 == null)
{

pictureBox7.Image = bm;
pictureBox7.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap kukuk3 = new Bitmap(pictureBox7.Image, 114, 77);
pictureBox7.Image = kukuk3;
pictureBox7.SizeMode = PictureBoxSizeMode.CenterImage;

}
else if (v4 == null)
{

pictureBox8.Image = bm;
pictureBox8.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap kukuk4 = new Bitmap(pictureBox8.Image, 114, 77);
pictureBox8.Image = kukuk4;
pictureBox8.SizeMode = PictureBoxSizeMode.CenterImage;

}
else if (v5 == null)
{

pictureBox9.Image = bm;
pictureBox9.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap kukuk5 = new Bitmap(pictureBox9.Image, 114, 77);
pictureBox9.Image = kukuk5;
pictureBox9.SizeMode = PictureBoxSizeMode.CenterImage;

}
else if (v6 == null)
{

pictureBox10.Image = bm;
pictureBox10.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap kukuk6 = new Bitmap(pictureBox10.Image, 114, 77);
pictureBox10.Image = kukuk6;
pictureBox10.SizeMode = PictureBoxSizeMode.CenterImage;

}

pictureBox2.Refresh();

}

else MessageBox.Show("Seçim Yapınız");
}

private void trackBar2_Scroll(object sender, EventArgs e) //Gri Format Renk -
Sıcaklık Ölçeği
{
min = (float)(Math.Round(Convert.ToDouble(textBox1.Text)));

```



```

max = (float)(Math.Round(Convert.ToDouble(textBox2.Text))); // textBoxlara
girilen termal kameraların en düşük ve en yüksek görüntülediği sıcaklık
değerleri
int b = trackBar2.Value; // trackBar2 nesnesindeki ayarlanan değer
sic1 = ((max-min)*b+(255*min))/255; // Sıcaklık trackBar2 nesnesinin eşleştiği
fonksiyon
label41.Text = tc.ToString();
label32.Text = sic1.ToString("N1") + " oC";
}

private void pictureBox13_Click(object sender, EventArgs e) //Gri Ton Renk -
Sıcaklık pictureBox
{
pictureBox13.Image = new Bitmap(pictureBox4.Width, pictureBox2.Height);
int il, jl;
dl = 256;

for (jl = 0; jl < 255; jl++) //1
{
dl--;

for (il = 0; il < 30; il++)
{

((Bitmap)pictureBox13.Image).SetPixel(il, jl,
System.Drawing.Color.FromArgb(dl, dl, dl));

}}}

private void pictureBox14_MouseDown(object sender, MouseEventArgs e) //Gri
Format filtrelenmiş resim üzerinde mouse oynatılarak parçalar seçiliyor.
{
// Starting point of the selection:
if (e.Button == MouseButtons.Left)
{
_selecting1 = true;
_selection1 = new Rectangle(new System.Drawing.Point(e.X, e.Y), new Size());
}}

private void pictureBox14_MouseMove(object sender, MouseEventArgs e)
{
// Update the actual size of the selection:
if (_selecting1)
{
_selection1.Width = e.X - _selection1.X;
_selection1.Height = e.Y - _selection1.Y;

// Redraw the pictureBox:
pictureBox14.Refresh();
}}

private void pictureBox14_Paint(object sender, PaintEventArgs e)
{
if (_selecting1)
{
// Draw a rectangle displaying the current selection
Pen pen = Pens.GreenYellow;
e.Graphics.DrawRectangle(pen, _selection1);
}}

```

```

private void pictureBox14_MouseUp(object sender, MouseEventArgs e)
{
    _selecting1 = false;
}

private void button5_Click(object sender, EventArgs e)
// RGB filtrelenmiş görüntü Seç komutu
{
// Seç komutu ile seçilen parçalar küçük kutucuklara ve büyük kutucuğa
boyutları değiştirilerek görüntüleniyor.

Bitmap b1 = (Bitmap)pictureBox14.Image;

if (b1.Clone(_selection1, b1.PixelFormat) != null)
{
    Bitmap bm1 = b1.Clone(_selection1, b1.PixelFormat);
    pictureBox3.Image = bm1;
    pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap bmpKucuk1 = new Bitmap(pictureBox3.Image, 342, 243);
    pictureBox3.Image = bmpKucuk1;
    pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;

    Bitmap v11 = (Bitmap)pictureBox5.Image;
    Bitmap v21 = (Bitmap)pictureBox6.Image;
    Bitmap v31 = (Bitmap)pictureBox7.Image;
    Bitmap v41 = (Bitmap)pictureBox8.Image;
    Bitmap v51 = (Bitmap)pictureBox9.Image;
    Bitmap v61 = (Bitmap)pictureBox10.Image;
    if (v11 == null)
    {

        pictureBox5.Image = bm1;
        pictureBox5.SizeMode = PictureBoxSizeMode.StretchImage;
        Bitmap kucuk11 = new Bitmap(pictureBox5.Image, 114, 77);
        pictureBox5.Image = kucuk11;
        pictureBox5.SizeMode = PictureBoxSizeMode.CenterImage;

    }

    else if (v21 == null)
    {

        pictureBox6.Image = bm1;
        pictureBox6.SizeMode = PictureBoxSizeMode.StretchImage;
        Bitmap kucuk21 = new Bitmap(pictureBox6.Image, 114, 77);
        pictureBox6.Image = kucuk21;
        pictureBox6.SizeMode = PictureBoxSizeMode.CenterImage;

    }

    else if (v31 == null)
    {

        pictureBox7.Image = bm1;
        pictureBox7.SizeMode = PictureBoxSizeMode.StretchImage;
        Bitmap kucuk31 = new Bitmap(pictureBox7.Image, 114, 77);
        pictureBox7.Image = kucuk31;
        pictureBox7.SizeMode = PictureBoxSizeMode.CenterImage;

    }

    else if (v41 == null)

```

```

{
    pictureBox8.Image = bm1;
    pictureBox8.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap küçük41 = new Bitmap(pictureBox8.Image, 114, 77);
    pictureBox8.Image = küçük41;
    pictureBox8.SizeMode = PictureBoxSizeMode.CenterImage;
}
else if (v51 == null)
{
    pictureBox9.Image = bm1;
    pictureBox9.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap küçük51 = new Bitmap(pictureBox9.Image, 114, 77);
    pictureBox9.Image = küçük51;
    pictureBox9.SizeMode = PictureBoxSizeMode.CenterImage;
}
else if (v61 == null)
{
    pictureBox10.Image = bm1;
    pictureBox10.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap küçük61 = new Bitmap(pictureBox10.Image, 114, 77);
    pictureBox10.Image = küçük61;
    pictureBox10.SizeMode = PictureBoxSizeMode.CenterImage;
}
pictureBox14.Refresh();
}

else MessageBox.Show("Seçim Yapınız");
}

private void button7_Click(object sender, EventArgs e)
//Gri format dönüştürme işlemi yapılıyor.
{
    if ((Bitmap)pictureBox1.Image != null)
    {
        // filtre oluşturuluyor.
        Grayscale filter1 = new Grayscale(0.60, 0.30, 0.10);
        // filtre uygulanıyor.
        pictureBox12.Image = filter1.Apply((Bitmap)pictureBox1.Image);
    }

    else MessageBox.Show("Resim Yükleyiniz");
}

private void button6_Click(object sender, EventArgs e) //Gri format filtreleme
işlemi yapılıyor.
{
    if ((Bitmap)pictureBox12.Image != null)
    {
        int b = trackBar2.Value;

```

```

Color OkunanRenk, DonusenRenk;
int R = 0, G = 0, B = 0;
Bitmap resim = new Bitmap(pictureBox12.Image);
int i = 0, j = 0;
for (int x = 0; x < resim.Width; x++)
{
j = 0;
for (int y = 0; y < resim.Height; y++)
{
OkunanRenk = resim.GetPixel(x, y);
if (b - 10 <= OkunanRenk.R && OkunanRenk.R <= b + 10)
R = 255;
else
R = 0;
if (OkunanRenk.G >= b-10 && OkunanRenk.G <= b+10)
G = 255;
else
G = 0;
if (OkunanRenk.B >= b-10 && OkunanRenk.B <= b+10)
B = 255;
else
B = 0;
DonusenRenk = Color.FromArgb(R,G,B);
resim.SetPixel(i, j, DonusenRenk);

j++;
}
i++;
}

pictureBox14.Image = resim;
}
else MessageBox.Show("Resim Yükleyiniz");
}

private void button8_Click(object sender, EventArgs e)
// Ana resim Renk - Ton dağılımını gösteren histogramlar oluşturuluyor.
{
Form7 f7 = new Form7();
f7.Show();

Bitmap bmp = new Bitmap(pictureBox1.Image);
int[] kirmizi = new int[256];
int[] yesil = new int[256];
int[] mavi = new int[256];

for (int i = 0; i < bmp.Size.Height; i++)
for (int j = 0; j < bmp.Size.Width; j++)
{
Color renk = bmp.GetPixel(j, i);

kirmizi[renk.R]++;
yesil[renk.G]++;
mavi[renk.B]++;
}

for (int p = 0; p < 255; p++)
{

```

```

f7.chart1.Series["KIRMIZI"].Points.AddXY(p, kirmizi[p]);
f7.chart1.Series["YEŞİL"].Points.AddXY(p, yesil[p]);
f7.chart1.Series["MAVİ"].Points.AddXY(p, mavi[p]);

}
// Histogramların x ve y eksen verileri bilgisayarda seçilen alanlara
depolanır.
SaveFileDialog sfd = new SaveFileDialog();
sfd.InitialDirectory = ".";
sfd.Title = @"C:\Users\KILICARSLAN\Desktop";
if (sfd.ShowDialog() == DialogResult.OK)
{
StreamWriter fwriter =
File.CreateText(sfd.FileName);

for (int y = 0; y < 256; y++)
{
fwriter.Write(kirmizi[y] + Environment.NewLine);
fwriter.Write(yesil[y] + Environment.NewLine);
fwriter.Write(mavi[y] + Environment.NewLine);

}

fwriter.Close();

}}

private void button10_Click(object sender, EventArgs e)
//Gri Formatlı resim Renk - Ton dağılımını gösteren histogramlar
oluşturuluyor.

{
Form8 f8 = new Form8();
f8.Show();
Bitmap bmp = new Bitmap(pictureBox2.Image);
int[] kirmizi = new int[256];
int[] yesil = new int[256];
int[] mavi = new int[256];

for (int i = 0; i < bmp.Size.Height; i++)
for (int j = 0; j < bmp.Size.Width; j++)
{
Color renk = bmp.GetPixel(j, i);

kirmizi[renk.R]++;

yesil[renk.G]++;
mavi[renk.B]++;
}

for (int p = 0; p < 255; p++)
{

f8.chart1.Series["KIRMIZI"].Points.AddXY(p, kirmizi[p]);
f8.chart1.Series["YEŞİL"].Points.AddXY(p, yesil[p]);
f8.chart1.Series["MAVİ"].Points.AddXY(p, mavi[p]);

}
}

```

```

SaveFileDialog sfd = new SaveFileDialog();
sfd.InitialDirectory = ".";
sfd.Title = @"C:\Users\KILICARSLAN\Desktop";
if (sfd.ShowDialog() == DialogResult.OK)
{
    StreamWriter fwriter =
    File.CreateText(sfd.FileName);

    for (int y = 0; y < 256; y++)
    {
        fwriter.Write(kirmizi[y] + Environment.NewLine);
        fwriter.Write(yesil[y] + Environment.NewLine);
        fwriter.Write(mavi[y] + Environment.NewLine);
    }

    fwriter.Close();
}

private void button11_Click(object sender, EventArgs e)
// RGB Filtrelenmiş resim Renk - Ton dağılımını gösteren histogramlar
oluşturuluyor.

{
    Form9 f9 = new Form9();
    f9.Show();
    Bitmap bmp = new Bitmap(pictureBox12.Image);
    int[] kirmizi = new int[256];
    int[] yesil = new int[256];
    int[] mavi = new int[256];
    int griton;
    int[] gri = new int[256];

    for (int i = 0; i < bmp.Size.Height; i++)
    for (int j = 0; j < bmp.Size.Width; j++)
    {
        Color renk = bmp.GetPixel(j, i);
        griton = (renk.R + renk.G + renk.B) / 3;
        gri[griton] = gri[griton] + 1;
    }

    for (int p = 0; p < 255; p++)
    {
        f9.chart1.Series["GRİTON"].Points.AddXY(p, gri[p]);
    }

    SaveFileDialog sfd = new SaveFileDialog();
    sfd.InitialDirectory = ".";
    sfd.Title = @"C:\Users\KILICARSLAN\Desktop";
    if (sfd.ShowDialog() == DialogResult.OK)
    {
        StreamWriter fwriter =
        File.CreateText(sfd.FileName);

        for (int y = 0; y < 256; y++)
        {

```

```

fwriter.Write(gri[y] + Environment.NewLine);
}

fwriter.Close();

}}

private void button9_Click(object sender, EventArgs e)
// Gri Format Filtrelenmiş resim Renk - Ton dağılımını gösteren histogramlar
oluşturuluyor.

{
Form10 f10 = new Form10();
f10.Show();
Bitmap bmp = new Bitmap(pictureBox14.Image);
int[] kirmizi = new int[256];
int[] yesil = new int[256];
int[] mavi = new int[256];
int griton;
int[] gri = new int[256];

for (int i = 0; i < bmp.Size.Height; i++)
for (int j = 0; j < bmp.Size.Width; j++)
{
Color renk = bmp.GetPixel(j, i);
griton = (renk.R + renk.G + renk.B) / 3;
gri[griton] = gri[griton] + 1;
}
for (int p = 0; p < 256; p++)
{
f10.chart1.Series["GRITON"].Points.AddXY(p, gri[p]);
}

SaveFileDialog sfd = new SaveFileDialog();
sfd.InitialDirectory = ".";
sfd.Title = @"C:\Users\KILICARSLAN\Desktop";
if (sfd.ShowDialog() == DialogResult.OK)
{
StreamWriter fwriter =
File.CreateText(sfd.FileName);

for (int y = 0; y < 256; y++)
{
fwriter.Write(gri[y] + Environment.NewLine);
}

fwriter.Close();
}}

//Seçili Alanlar bölümündeki küçük resimlerin üzerine tıklandığında yandaki
ekrana büyütülerek getiriliyor.

private void pictureBox5_Click(object sender, EventArgs e)
{
Bitmap bmp = new Bitmap(pictureBox5.Image);

```

```

pictureBox3.Image = bmp;
pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
Bitmap bmp1 = new Bitmap(pictureBox3.Image, 342, 245);
pictureBox3.Image = bmp1;
pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void pictureBox6_Click(object sender, EventArgs e)
{
    Bitmap bmp = new Bitmap(pictureBox6.Image);
    pictureBox3.Image = bmp;
    pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap bmp1 = new Bitmap(pictureBox3.Image, 342, 245);
    pictureBox3.Image = bmp1;
    pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void pictureBox7_Click(object sender, EventArgs e)
{
    Bitmap bmp = new Bitmap(pictureBox7.Image);
    pictureBox3.Image = bmp;
    pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap bmp1 = new Bitmap(pictureBox3.Image, 342, 245);
    pictureBox3.Image = bmp1;
    pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void pictureBox8_Click(object sender, EventArgs e)
{
    Bitmap bmp = new Bitmap(pictureBox8.Image);
    pictureBox3.Image = bmp;
    pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap bmp1 = new Bitmap(pictureBox3.Image, 342, 245);
    pictureBox3.Image = bmp1;
    pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void pictureBox9_Click(object sender, EventArgs e)
{
    Bitmap bmp = new Bitmap(pictureBox9.Image);
    pictureBox3.Image = bmp;
    pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap bmp1 = new Bitmap(pictureBox3.Image, 342, 245);
    pictureBox3.Image = bmp1;
    pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void pictureBox10_Click(object sender, EventArgs e)
{
    Bitmap bmp = new Bitmap(pictureBox10.Image);
    pictureBox3.Image = bmp;
    pictureBox3.SizeMode = PictureBoxSizeMode.StretchImage;
    Bitmap bmp1 = new Bitmap(pictureBox3.Image, 342, 245);
    pictureBox3.Image = bmp1;
    pictureBox3.SizeMode = PictureBoxSizeMode.CenterImage;
}}}}

```


EK 8. Ortam ve Vital Parametreleri Ara Yüzü Program Kodları

```
//Gerekli Kütüphaneler yükleniyor.

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;
using System.Media;
using ZedGraph;

namespace WindowsFormsApplication2
{
    public partial class Form4 : Form
    {
        //Programda kullanılacak değişkenler atanıyor.

        string[] portlar = SerialPort.GetPortNames();
        int p;
        float nem;
        float isi;
        float o2;
        float nabiz;
        float ekg;
        float spo2;
        float isiset
        float nemset
        float o2set
        int sayac = 0;
        int sayac1 = 0;
        int sayac2 = 0;
        int sayac3 = 0;
        double x, y, z;
        float ek, n, s;

        GraphPane myPaneEkg = new GraphPane();
        PointPairList listPointsEkg = new PointPairList(); //ekg grafik
        değişkenleri//
        LineItem myCurveEkg;
        double zaman = 0;

        GraphPane myPaneSpo2 = new GraphPane();
        PointPairList listPointsSpo2 = new PointPairList(); //spo2 grafik
        değişkenleri//
        LineItem myCurveSpo2;
        double zaman1 = 0;

        long maksm = 30;
        long minm = 0;
        SoundPlayer ses = new SoundPlayer();
    }
}
```

```

public Form4()
{
InitializeComponent();
}

private void Form4_Load(object sender, EventArgs e)

//COMBOBOX' a seri port adlarını getiriyor.//
{
pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
timer2.Enabled = true;

// Sınır Değer formundan kullanıcılar tarafından ortam değerleri atanan sınır
değerler getiriliyor.

isiset = (float)(Math.Round(Convert.ToDouble(Form6.t)));
nemset = (float)(Math.Round(Convert.ToDouble(Form6.n)));
o2set = (float)(Math.Round(Convert.ToDouble(Form6.o2)));
isiset1 = (float)(Math.Round(Convert.ToDouble(Form6.t1)));
nemset1 = (float)(Math.Round(Convert.ToDouble(Form6.n1)));
o2set1 = (float)(Math.Round(Convert.ToDouble(Form6.o21)));

foreach (string port in portlar)
{
comboBox1.Items.Add(port);
comboBox1.SelectedIndex = 0;
}

GrafikHazirla();
GrafikHazirla1();

}
private void button1_Click(object sender, EventArgs e)
// Bağlantıyı başlatıyor.//
{

if (serialPort1.IsOpen == false)
{
if (comboBox1.Text == "")
{
MessageBox.Show("Bağlantı Yok", "HATA");
return;
}
serialPort1.BaudRate = 115200; //115200 kb/sn bant genişliği ile ara yüze veri
aktarılıyor.

serialPort1.PortName = comboBox1.Text;
try
{
serialPort1.Open();
timer1.Enabled = true;

label10.Text = "Bağlantı Açık";
button2.Enabled = true;
}

catch (Exception hata)

```

```

{
    MessageBox.Show("Hata:" + hata.Message);
}}

private void button2_Click(object sender, EventArgs e)
//Bağlantıyı kesiyor.//
{
    serialPort1.Close();
    label10.Text = "Bağlantı Kapalı";
    timer1.Enabled = false;
    button1.Enabled = true;
    label7.Text = "-----";
    label8.Text = "-----";
    label9.Text = "-----";
    label11.Text = "-----";
    label12.Text = "-----";
    label13.Text = "-----";

    return;
}
private void termogramAnalizToolStripMenuItem_Click(object sender, EventArgs e)
//Formlar Arası Geçiş//
{
    Form1 göster1 = new Form1();
    göster1.Show();
    this.Hide();
}

private void button9_Click(object sender, EventArgs e)
// Sınır değerler formuna geçiş
{
    serialPort1.Close();
    timer1.Enabled = false;
    timer2.Enabled = false;
    Form6 göster2 = new Form6();
    göster2.Show();
    this.Hide();
}

private void timer1_Tick(object sender, EventArgs e)
// Seri porttan gelen veriler okuyor.
{
    // Arduino kartı bütün ölçüm verilerini paket şeklinde gönderir ve verilerin
    // karışmaması için * işareti konmuştur.
    //C# programı gelen verileri dizi hale getirir ve her bir veriyi * işareti
    // vasıtasıyla ayırır.
    timer1.Interval = 1;
    string sonuc = serialPort1.ReadLine().ToString();
    string[] pot = sonuc.Split('*');
    float nem = (float)(Math.Round(Convert.ToDouble(pot[0]) / 100, 2));
    float isi = (float)(Math.Round(Convert.ToDouble(pot[1]) / 100, 2));
    float o2 = (float)(Math.Round(Convert.ToDouble(pot[2]) / 100, 2));
    float nabiz = (float)(Math.Round(Convert.ToDouble(pot[3]) / 100, 2));
    float spo2 = (float)(Math.Round(Convert.ToDouble(pot[4]), 0));
    int ekg = (int)(Math.Round(Convert.ToDecimal(pot[5]), 0));
}

```



```

private void button3_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
serialPort1.Write("2");
timer1.Enabled = true;
}
private void button6_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
serialPort1.Write(richTextBox1.Text);
timer1.Enabled = true;
}

// Nem set değeri Arduinoya gönderiliyor.

private void button4_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
serialPort1.Write("1");
timer1.Enabled = true;
}
private void button7_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
serialPort1.Write(richTextBox2.Text);
timer1.Enabled = true;
}

// O2 set değeri Arduinoya gönderiliyor.

private void button5_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
serialPort1.Write("3");
timer1.Enabled = true;
}

private void button8_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
serialPort1.Write(richTextBox3.Text);
timer1.Enabled = true;}

private void timer2_Tick_1(object sender, EventArgs e)
//Vital parametreler uyarı ve ikaz fonksiyonu
{

sayac++;
timer2.Interval = 500;

if (serialPort1.IsOpen == true)
{
ek = (float)(Math.Round(Convert.ToDouble(label111.Text)));
n = (float)(Math.Round(Convert.ToDouble(label112.Text)));
s = (float)(Math.Round(Convert.ToDouble(label113.Text)));

if (sayac % 1 == 0)
{

```

```

pictureBox1.Width = 0;
pictureBox1.Height = 0;
}
if (sayac % 2 == 0)
{

pictureBox1.Width = 100;
pictureBox1.Height = 50;
}

if (ek < 200) // Ekg ikaz fonksiyonu
{

if (sayac % 1 == 0)
{

this.label11.ForeColor = Color.Red;
this.label4.ForeColor = Color.Red;
this.label26.ForeColor = Color.Red;
}
if (sayac % 2 == 0)
{

this.label11.ForeColor = Color.Black;
this.label4.ForeColor = Color.Black;
this.label26.ForeColor = Color.Black;
}}

if (ek >= 200)
{
this.label11.ForeColor = Color.Black;
this.label4.ForeColor = Color.Black;
this.label26.ForeColor = Color.Black;

}

if (n < 50) // nabız ikaz fonksiyonu
{

if (sayac % 1 == 0)
{

this.label12.ForeColor = Color.Red;
this.label5.ForeColor = Color.Red;
this.label27.ForeColor = Color.Red;
}
if (sayac % 2 == 0)
{

this.label12.ForeColor = Color.Black;
this.label5.ForeColor = Color.Black;
this.label27.ForeColor = Color.Black;
}}

if (n >= 50)
{

```

```

this.label12.ForeColor = Color.Black;
this.label5.ForeColor = Color.Black;
this.label27.ForeColor = Color.Black;

}

if (s < 85) // Spo2 ikaz fonksiyonu
{

if (sayac % 1 == 0)
{

this.label13.ForeColor = Color.Red;
this.label6.ForeColor = Color.Red;
this.label28.ForeColor = Color.Red;
}
if (sayac % 2 == 0)
{

this.label13.ForeColor = Color.Black;
this.label6.ForeColor = Color.Black;
this.label28.ForeColor = Color.Black;
}}

if (s >= 85)
{
this.label13.ForeColor = Color.Black;
this.label6.ForeColor = Color.Black;
this.label28.ForeColor = Color.Black;
}

if (x < isiset || y < nemset || z < o2set || n < 50 || s < 85 || ekg < 200)
{

string dizin = Application.StartupPath + "\\Tehlike Alarm Sesi (online-audio-
converter.com).wav";
ses.SoundLocation = dizin;
ses.Play();
}}

private void timer3_Tick(object sender, EventArgs e)
//Ortam parametreleri (sıcılık) uyarı ve ikaz fonksiyonları
{
timer3.Interval = 700;
sayac1++;
if (sayac % 1 == 0)
{

this.label18.ForeColor = Color.Red;
this.label2.ForeColor = Color.Red;
}
if (sayac % 2 == 0)
{

this.label18.ForeColor = Color.Black;
this.label2.ForeColor = Color.Black;
}}

```

```

private void timer4_Tick(object sender, EventArgs e)
//Ortam parametreleri (nem) uyarı ve ikaz fonksiyonları

{
timer4.Interval = 700;
sayac2++;

if (sayac % 1 == 0)
{

this.label7.ForeColor = Color.Red;
this.label11.ForeColor = Color.Red;
}
if (sayac % 2 == 0)
{

this.label7.ForeColor = Color.Black;
this.label11.ForeColor = Color.Black;
}}

private void timer5_Tick(object sender, EventArgs e)
//Ortam parametreleri (o2) uyarı ve ikaz fonksiyonları

{
timer5.Interval = 700;
sayac3++;

if (sayac % 1 == 0)
{

this.label19.ForeColor = Color.Red;
this.label13.ForeColor = Color.Red;
label19.Text = o2 + ("%");
}
if (sayac % 2 == 0)
{

this.label19.ForeColor = Color.Black;
this.label13.ForeColor = Color.Black;
label19.Text = o2 + ("%");
}}

private void GrafikHazirla() //EKG grafik alt fonksiyonu
{

myPaneEkg = zedGraphControl1.GraphPane;
myPaneEkg.Title.Text = "Ekg - Zaman Grafiği ";
myPaneEkg.XAxis.Title.Text = " t (s) ";
myPaneEkg.YAxis.Title.Text = "Ekg (mikrov)";
myPaneEkg.YAxis.Scale.Min = 300;
myPaneEkg.YAxis.Scale.Max = 500;
myCurveEkg = myPaneEkg.AddCurve(null, listPointsEkg, Color.Red,
SymbolType.None);
myCurveEkg.Line.Width = 2;

}

private void GrafikHazirla1() //Spo2 grafik alt fonksiyonu
{

```



```
myPaneSpo2 = zedGraphControl2.GraphPane;  
myPaneSpo2.Title.Text = "Spo2 - Zaman Grafiği ";  
myPaneSpo2.XAxis.Title.Text = " t (s) ";  
myPaneSpo2.YAxis.Title.Text = "Spo2 (%)";  
myPaneSpo2.YAxis.Scale.Min = 0;  
myPaneSpo2.YAxis.Scale.Max = 110;  
myCurveSpo2 = myPaneSpo2.AddCurve(null, listPointsSpo2, Color.Red,  
SymbolType.None);  
myCurveSpo2.Line.Width = 3;  
}}}
```



ÖZGEÇMİŞ

30.03.1988 tarihinde Kayseri’ de dünyaya gelen Kadir Caner KILÇARSLAN, ilköğrenimini Ahmet Paşa İlkokulu, ortaöğrenimini Şükrü Malaz İlköğretim Okulu, Lise öğrenimini ise Nuh Mehmet Küçükçalık Anadolu Lisesinde tamamladıktan sonra 2007 yılında Erciyes Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümünü kazanmıştır. 3. Sınıfta iken Çift Anadal Programı Başvurusu kabul edilerek Elektrik-Elektronik Mühendisliği Bölümünde de öğrenimini sürdürmüş ve her iki bölümden 2012 yılında mezun olmuştur.

2014 yılında yüksek lisans öğrenimine Bozok Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Ana Bilim Dalında başlamıştır.

Kısa süreli olarak özel sektörde çalışan Kadir Caner KILIÇARSLAN, 2013 yılında Bozok Üniversitesi Yapı İşleri ve Teknik Daire Başkanlığına atanmış, burada 2 sene çalışmış, daha sonra Bozok Üniveristesi Sağlık Uygulama ve Araştırma hastanesinde de 2 sene görev yaptıktan sonra 2018 yılı içerisinde tayin olduğu Gençlik ve Spor Kayseri İl Müdürlüğünde halen çalışmaktadır. Evli ve bir çocuk sahibidir.

Adres: Gençlik ve Spor Kayseri İl Müdürlüğü

Melikgazi/KAYSERİ

Telefon: 0352 332 44 30

E-posta: k.caner.kilicarslan@gmail.com