



**T.C.  
DÜZCE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ANDROİD TABANLI MOBİL SAĞLIK İZLEME UYGULAMASI**

**EBRU GÜNDOĞDU**

**YÜKSEK LİSANS TEZİ**

**ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ**

**ANABİLİM DALI**

**DANIŞMAN  
DOÇ. DR. ALİ ÇALHAN**

**DÜZCE, 2018**

**T.C.**  
**DÜZCE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ANDROİD TABANLI MOBİL SAĞLIK İZLEME UYGULAMASI**

Ebru Gündođdu tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Doç. Dr. Ali ÇALHAN

Düzce Üniversitesi

**Jüri Üyeleri**

Doç. Dr. Ali ÇALHAN

Düzce Üniversitesi

Dr. Öğr. Üyesi Mehmet ŞİMŞEK

Düzce Üniversitesi

Doç. Dr. Kerem KÜÇÜK

Kocaeli Üniversitesi

Tez Savunma Tarihi: 29/06/2018

## BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

29 Haziran 2018

Ebru Gündođdu

## TEŐEKKÜR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Doç. Dr. Ali ÇALHAN' a en içten dileklerle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve eşime sonsuz teşekkürlerimi sunarım.

**29 Haziran 2018**

**Ebru GÜNDOĞDU**

# İÇİNDEKİLER

	<u>Sayfa No</u>
<b>ŞEKİL LİSTESİ</b> .....	<b>VII</b>
<b>ÇİZELGE LİSTESİ</b> .....	<b>VIII</b>
<b>KISALTMALAR</b> .....	<b>IX</b>
<b>SİMGELER</b> .....	<b>X</b>
<b>ÖZET</b> .....	<b>XI</b>
<b>ABSTRACT</b> .....	<b>XII</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. MATERYAL VE YÖNTEM</b> .....	<b>3</b>
<b>2.1. TASARLANAN SİSTEM HAKKINDA GENEL BİLGİLER</b> .....	<b>3</b>
<b>2.1.1. Arduino Elektronik Kartı</b> .....	<b>3</b>
2.1.1.1. <i>Arduino Elektronik Kartının Bileşenleri</i> .....	3
2.1.1.2. <i>Arduino Elektronik Kartı İle Yapılabilecekler</i> .....	4
2.1.1.3. <i>Arduino Elektronik Kartının Kullanımı</i> .....	4
<b>2.1.2. Uno Elektronik Kartı</b> .....	<b>6</b>
2.1.2.1. <i>Arduino Uno Elektronik Kartının Teknik Yapısı</i> .....	8
2.1.2.2. <i>Güç</i> .....	8
2.1.2.3. <i>Giriş ve Çıkışlar</i> .....	8
2.1.2.4. <i>İletişim</i> .....	9
2.1.2.5. <i>Programlama</i> .....	9
2.1.2.6. <i>USB Aşırı Akım Koruması</i> .....	10
<b>2.1.3. Android Tasarımı</b> .....	<b>10</b>
2.1.3.1. <i>Vektörel Varlıklar</i> .....	11
2.1.3.2. <i>Tema Editörü</i> .....	12
2.1.3.3. <i>Proje Şablonu</i> .....	13
2.1.3.4. <i>Performans İzleyiciler</i> .....	14
2.1.3.5. <i>Geliştirici Hizmetleri</i> .....	15
<b>2.1.4. Vital Bulguların Ölçülmesi</b> .....	<b>16</b>

2.1.5. Kablosuz Veri Transferi.....	17
2.1.6. Vücut Alan Ağları.....	17
2.2. TASARLANAN SİSTEMİN ÇALIŞMASI .....	18
3. BULGULAR VE TARTIŞMA .....	21
4. SONUÇLAR VE ÖNERİLER .....	26
5. KAYNAKLAR.....	27
6. EKLER.....	30
6.1. EK 1: ANDROID STUDIO KODLARI.....	30
ÖZGEÇMİŞ .....	45



## ŞEKİL LİSTESİ

	<b><u>Sayfa No</u></b>
Şekil 2.1. Arduino / Genuino logosu .....	3
Şekil 2.2. Fritzing programı .....	5
Şekil 2.3. Arduino Uno / Genuino Uno görünümü .....	6
Şekil 2.4. Uno görünümü .....	7
Şekil 2.5. Arduino programı board seçimi .....	10
Şekil 2.6. Android Studio vektörel varlıklar penceresi .....	11
Şekil 2.7. Android Studio vektörel varlık ekleme .....	12
Şekil 2.8. Android Studio tema editörü .....	13
Şekil 2.9. Android Studio proje şablonu .....	13
Şekil 2.10. Android Studio kayan menü örneği .....	14
Şekil 2.11. Android Studio grafik işlemci izleyicisi .....	14
Şekil 2.12. Android Studio ağ izleyicisi .....	15
Şekil 2.13. Android Studio firebase veri depolama .....	15
Şekil 2.14. Vital bulguların görüntülediği ekran .....	16
Şekil 2.15. Vücut alan ağı katmanlı yapısı .....	17
Şekil 2.16. Sistemin blok diyagramı .....	18
Şekil 2.17. Tasarlanan sistem .....	19
Şekil 2.18. Tasarlanan mobil program .....	20
Şekil 3.1. EKG ölçüm sensörü .....	21
Şekil 3.2. Nabız ve SpO2 ölçüm sensörü .....	21
Şekil 3.3. Sistemin besleme ünitesi .....	22
Şekil 3.4. Koordinatör düğüm ve iletişim ünitesi .....	22
Şekil 3.5. EKG ilk test ölçüm sonuçları .....	23
Şekil 3.6. EKG son test ölçüm sonuçları .....	24
Şekil 3.7. Sistem için tasarlanan android programı .....	25
Şekil 3.8. Nabız, SpO2 ölçülmesi ve EKG problemlerinin bağlantıları .....	25

## ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 2.1. Mega ve Uno Kartı Özellikleri .....	5





## KISALTMALAR

AC	Alternatif Akım
DC	Dođru Akım
EKG	Elektrokardiyografi
G/Ç	Giriş / Çıkış
GPU	Grafik İşleme İzleyicisi
ICSP	Devre Üzerinde Seri Programlama
IDE	Geliştirme Ortamı
CAA	Kablosuz Algılayıcı Ağ
KVAA	Kablosuz Vücut Alan Ađı
PWM	Sinyal Genişlik Modülasyonu
SMD	Yüzey Montaj Devre Elemanı
SpO2	Kandaki Oksijen Miktarı
USB	Evrensel Seri Veri Yolu

## SİMGELER

A	Akım
Hz	Frekans Birimi
mm	Uzunluk Birimi
s	Saniye
V	Volt
%	Yüzde
$\Omega$	Ohm



## ÖZET

### ANDROİD TABANLI MOBİL SAĞLIK İZLEME UYGULAMASI

Ebru GÜNDOĞDU

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı  
Yüksek Lisans Tezi

Danışman: Doç. Dr. Ali ÇALHAN

Haziran 2018, 43 sayfa

Her yıl hasta sayısındaki artış miktarı, cihaz ebatlarının çok büyük olması, hasta doktor arasındaki iletişimin eksik olması ve hastanelerde yatak sayısının yetersiz olması gibi birçok neden sonucunda hastalara zamanında teşhis konulamamaktadır. Bu nedenlerden dolayı ülkemizde ve dünyada bu problemleri ortadan kaldırmaya yönelik birçok çalışmalar yapılmıştır. Bu çalışmaların başında hasta sağlık izleme sistemlerinin geliştirilmesi gelmektedir. Teknolojinin gelişmesine paralel olarak, hasta sağlık izleme sistemleri kolay taşınabilir şekilde tasarlanmaya başlanmıştır. Değişik yazılım ve donanım yöntemlerini kullanarak tasarlanan sağlık izleme sistemlerinde amaç, hastanın algılayıcılar sayesinde kendi kendine sağlık durumunu gözlemleyebilmesi ve hastanın durumunu doktorun uzaktan takip edebilmesi gelmektedir. Bu çalışmada, kablosuz vücut alan ağları ile insanın vital bulgularından olan nabız, kandaki oksijen miktarı ve EKG değerleri kablosuz olarak android mobil taşınabilir cihaz üzerinde gözlemlenebilmektedir.

**Anahtar sözcükler:** Hasta izleme, Kablosuz vücut alan ağları, Mobil sağlık izleme.

## ABSTRACT

### ANDROID BASED MOBILE HEALTH MONITORING APPLICATION

Ebru GÜNDOĞDU

Düzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronics  
and Computer Engineering,

Master's Thesis

Supervisor: Assoc. Prof. Dr. Ali ÇALHAN

June 2018, 43 pages

Illness cannot be diagnosed on time for some reasons such as the increase in the number of patients, lack of communication between patients and doctors, inadequate number of beds in hospitals, and expensive-large size devices. For these reasons, many studies have been carried out in our country and in the world to eliminate these problems. The development of patient health monitoring systems is crucial among these studies. The patient health monitoring systems have begun to be designed to be portable correspondingly to the development of technology. The purpose of health monitoring systems designed by using different software and hardware methods is that the patient can observe his / her health status thanks to the sensors and can monitor the situation of the patient from a distance. In this study, heart rate, amount of oxygen and ECG values from wireless body area networks and human vital findings can be observed wirelessly on Android mobile portable device.

**Keywords:** Mobile health monitoring, Patient monitoring, Wireless body area networks.

# 1. GİRİŞ

Kablosuz teknolojinin geliřimi, kısa mesafede birbirleriyle kablosuz olarak haberleřebilen cihazların sayısını arttırmıřtır. Bu teknoloji sayesinde dūřuk gūçte ve maliyette çok fonksiyonlu cihazlar geliřtirilebilmektedir. Algılayıcı dūğüm denilen bu cihazlar hesaplama, algılama ve iletiřim kabiliyeti olan birer kablosuz ađ elemanıdır. Algılayıcı dūğümlerin topladıkları verileri gerekli birimlere göndermek amacıyla oluřturulan kablosuz ortama Kablosuz Algılayıcı Ađlar (KAA) denilmektedir [1]-[3]. KAA teknolojisi ile birçok uygulama gerçekteřtirilmektedir [4],[5]. Gerçekteřtirilen bu uygulamalardan birisi, fizyolojik sinyallerin ölçülmesinde kullanılan Kablosuz Vücut Alan Ađları (KVAA)'dır [6]. KVAA alanında birçok çalıřmalar yapılmıřtır [7]-[14]. KAA ve KVAA'ları incelediğimize, KVAA'ların daha az dūğüm sayısına, kararlı görüntülenme oranına, daha basit gūç kaynađına ve küçük çalıřma alanına sahip olduđu görülmektedir. KVAA'lar dūğümlerin üzerinde bulunan algılayıcılara göre ölçüm yapabilme kabiliyetine sahiptir. Bazı dūğümler nabız, oksijen miktarı ölçümlerken bazıları ise Elektrokardiyografi (EKG), vücut sıcaklıđı gibi deđerleri ölçümlenmektedir [15]-[18]. Bu ölçüm yapabilen algılayıcılar insan vücudunun üzerine veya içerisine yerleřtirilebilmektedir [19],[20].

Son yıllarda kabloları, büyük cihazları aradan kaldıracak ve maliyetleri azaltacak için birçok hasta izleme sistemi yapılmıřtır ve birçok deđiřik yöntem kullanılmıřtır [21]-[24]. Buna paralel olarak yakın zamanda kullanım kolaylıđı, uygun maliyet gibi çözümlerden dolayı cep telefonu ve Android tabanlı sistemleri kullanarak birçok kontrol ve görüntüleme sistemleri geliřtirilmiřtir [25]-[30]. Geliřtirilen bu mobil sistemlerin sađlık alanına uygulanmasına da çalıřılmaktadır [31]-[32].

Bu çalıřmada, KVAA'lar ve Android tabanlı sistem kullanarak hastaya bađlanacak algılayıcı dūğümler vasıtasıyla hastanın vital bulgularından olan nabız, oksijen miktarı ve EKG grafiđinin mobil sistemde görüntülenmesi amaçlanmıřtır. Bu amacı gerçekteřtirmek için algılayıcı dūğümlerde hastanın nabzını, kandaki oksijen miktarını ve EKG ölçümünü yapabilecek elektronik bir devre tasarlanmıřtır. Ölçüm yapılan sinyaller bir koordinatör dūğüme gönderilmektedir. Koordinatör dūğüm

toplamiş olduđu verileri Android mobil sistemde görüntülenebilmesi için Bluetooth 2.0 teknolojisini kullanarak verileri sisteme göndermektedir. Gönderilen bu veriler Android mobil sistem için tasarımı yapılan ara yüz programı ile mobil sağlık sisteminde görüntülenmektedir. Bu şekilde büyük pahalı cihazlar olmadan; zaman, mekân ve kablolardan bağımsız bir şekilde hastanın kolaylıkla izlenebilmesi sağlanmıştır.

## 2. MATERYAL VE YÖNTEM

### 2.1. TASARLANAN SİSTEM HAKKINDA GENEL BİLGİLER

Tez içeriğinde tasarlanan tüm sistemin elemanları başlıklar halinde açıklanmıştır.

#### 2.1.1. Arduino Elektronik Kartı

Arduino kartı, giriş/çıkış birimlerini içeren fiziksel bir geliştirme platformudur. Kartın içeriğinde ATmega32u4, ATmega2560, ATmega328 gibi Atmel AVR mikrodenetleyicilerinden biri ve diğer işlemleri gerçekleştirebilmek için devre elemanlarını bulunur. Mikrodenetleyicinin çalışabilmesi için 16MHz kristal ve 5V regüle devresi vardır. Kartı programlayabilmek için herhangi bir programlayıcıya ihtiyaç duymaz. Programlama yapabilmek için kartın içeriğinde bootloader dosyası vardır ve bu sayede kartın USB bağlantısını yaparak karta program atmak mümkündür.



Şekil 2.1. Arduino / Genuino logosu.

Arduino kartına benzer bir kart da mevcuttur. Bu kart Genuino kartıdır. Arduino'nun kurucu ortaklarından olan Massimo Banzi, David Cuartielles, Tom Igoe, ve David Mellis tarafından Genuino şirketi kurulmuştur. Kartın Arduino kartından tek farkı ismidir, diğer bütün özellikler iki kartta da aynıdır.

##### 2.1.1.1. Arduino Elektronik Kartının Bileşenleri

Arduino'da kullandığımız geliştirme platformu; Arduino bootloader, geliştirme ortamı (IDE), AVR Dude, Arduino kütüphaneleri ve derleyiciden oluşmaktadır.

Bu cihazın yazılımı, geliştirme platformu ve kütüphaneler aracılığıyla yazılır. IDE, java dilini temel alır. Arduino'da kütüphaneler C++ ve C dilinde yazılarak hazırlanmıştır. Kütüphaneler AVR Libc. Ve AVR-GCC derlenerek hazırlanmıştır. Arduino bootloader

bileşeni olarak Optiboot bileşenini kullanır. Bu bileşen sayesinde Arduino kart üzerinde kolaylıkla programlanabilir. Arduino'nun çok tercih edilmesinin sebeplerinden biri ise mikrodenetleyici konusunda çok detaylı bilgi gerektirmeden kolaylıkla programlama yapabilmemizi sağlamasıdır. Basit programlar da kütüphaneler aracılığıyla gerçekleştirilebilmektedir. Kütüphaneler, yazılım yazdığımız editör ile birlikte gelmektedir. Son olarak da Arduino da derlenen kodları programlamak için AVR Dude bileşeni kullanılır.

#### 2.1.1.2. *Arduino Elektronik Kartı İle Yapılabilecekler*



Bu geliştirme cihazı için hazırlanmış birçok kütüphaneyi kullanarak kolaylıkla kodlar yazılabilmektedir. Dijital ve analog sinyalleri rahatlıkla işlenebilmektedir. Sensörlerden gelen bilgileri kullanarak çevresiyle etkileşim içinde olan sistemler tasarlanabilmektedir. Oluşturulan projeye kullanıcıya bilgi verecek şekilde ses, ışık, hareket gibi tepki verici sistemler eklenebilmektedir. Arduino kartı için birçok versiyon mevcuttur. Bu versiyonları kullanarak çok fazla sayıda proje gerçekleştirilebilir.

#### 2.1.1.3. *Arduino Elektronik Kartının Kullanımı*

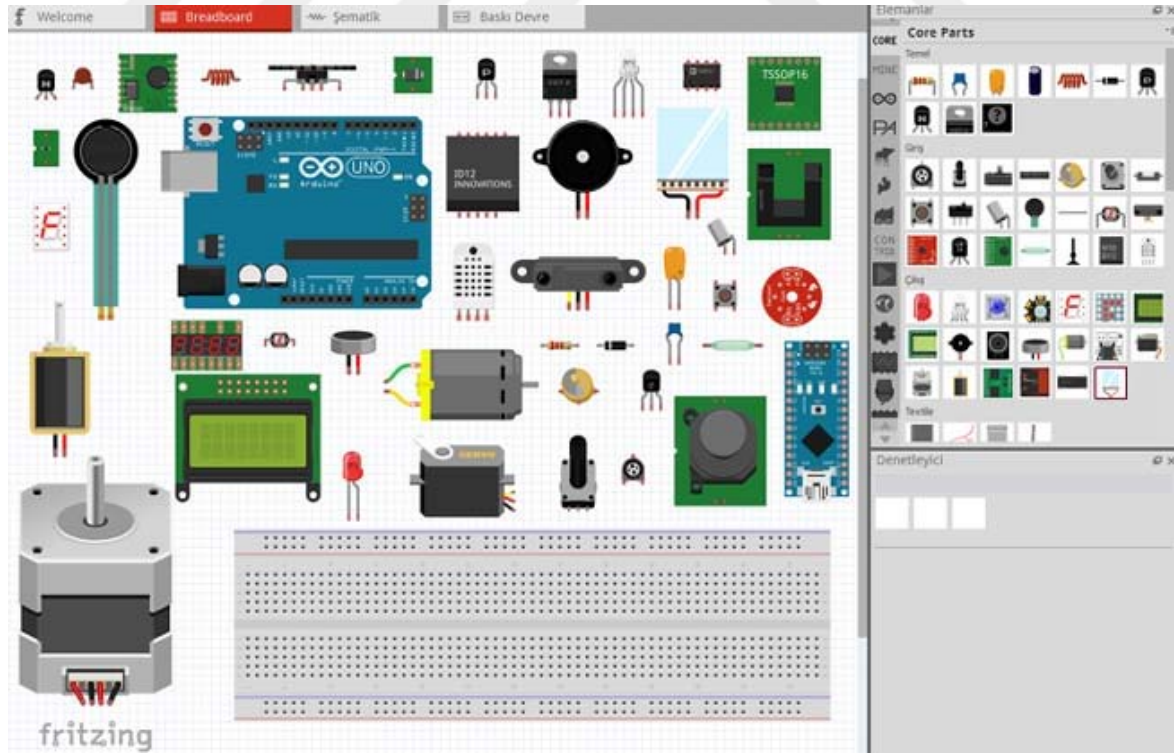
Arduino ön prototiplemeyi kolaylaştıran bir cihazdır. Daha gelişmiş bir sistem yapabilmek için elektronik ve yazılım bilgisine ihtiyaç duyulmaktadır. Ayrıca Arduino kartını kendi yazılım platformu ile %100 performansla kullanamayacağımızı da bilmemiz gerekmektedir. Arduino kartı kullanmadan önce bu kartın değişik modellerinden olan Uno, Nano, Mikro... vb gibi Arduino kartları hakkında bilgi sahibi olup projemize göre uygun kartı seçmemiz gerekmektedir. Arduino kartlarının hepsi aynı prensiple kodlanır ve çalıştırılır, fakat her kartın kendine ait özellikleri vardır. Çizelge 2.1' de en çok kullanılan Arduino Uno kartı ve Arduino Mega kartı görülmektedir.



Çizelge 2.1. Mega ve Uno kartı özellikleri.

	<p><i>Uno Kartı</i></p> <ul style="list-style-type: none"><li>- Bu geliştirme ortamının en çok kullanılan elektronik kartıdır.</li><li>- ATmega328 entegre içerir.</li><li>- 14 Dijital Giriş Çıkış bağlantısı, 6 PWM bağlantısı ve 6 ADC bağlantısı bulunur.</li><li>- Flash olarak 32 KB hafıza içerir.</li></ul>
	<p><i>Mega Kartı</i></p> <ul style="list-style-type: none"><li>- ATmega2560 entegresi içerir.</li><li>- Uno'dan daha fazla sayıda giriş ve çıkış bağlantısı içerir.</li><li>- Flash olarak 256 KB hafıza içerir.</li></ul>

Bu elektronik kartlarla yapılabilecek projelerin bağlantılarını elektronik ortamda yapabilmek için Firtzing programı kullanılmaktadır. Şekil 2.2' de Fritzing programı ve bazı bileşenleri görülmektedir. Bu program ile yalnızca projelerin çizimleri gerçekleştirilebilmektedir, simülasyon özelliği bulunmamaktadır.



Şekil 2.2. Fritzing programı.

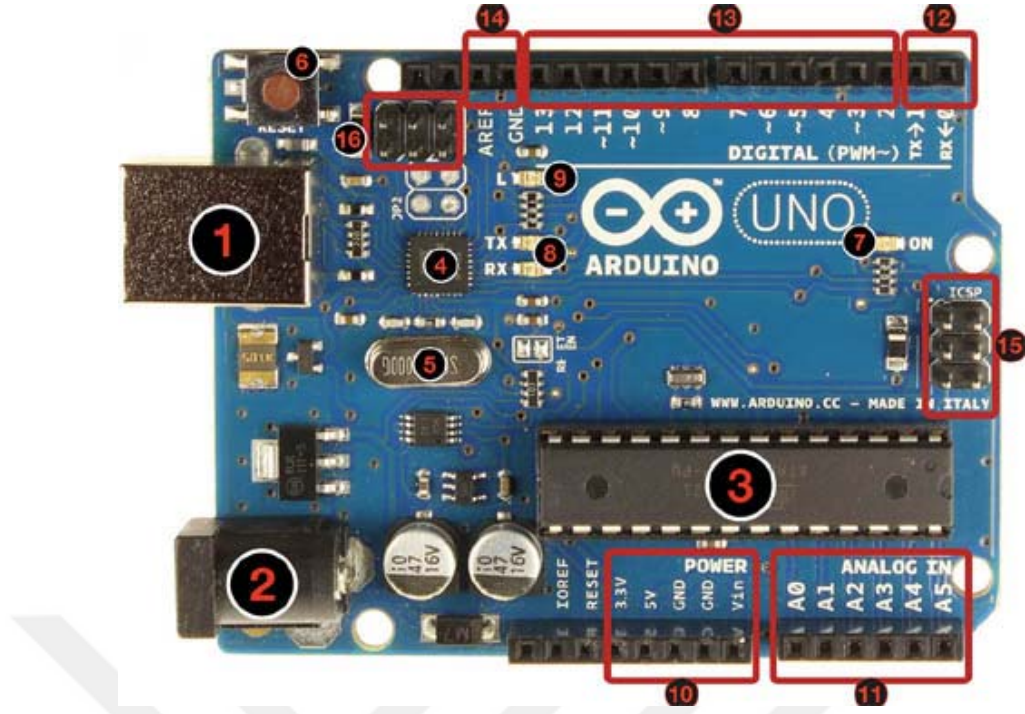
### 2.1.2. Uno Elektronik Kartı

Uno içerisinde ATmega328 bulunur. Bu markanın çok popüler bir kartıdır. R2, R3 ve SMD gibi versiyonları vardır. Genuino Uno kartı ile tüm özellikleri birebir aynıdır.



Şekil 2.3. Arduino Uno / Genuino Uno görünümü.

Arduino kartı üzerinde 14 adet dijital giriş / çıkış pin bağlantıları mevcuttur. Bu pinlerden 6 tanesini PWM çıkışı olarak da kullanabilir. Bunların haricinde 6 adet analog giriş, 16MHZ'lik bir adet kristal, 2.1mm güç bağlantısı, USB bağlantısı, ICSP bağlantısı ve reset butonu da bulunmaktadır. Bir mikrodenetleyiciyi destekleyen tüm bileşenleri Arduino Uno içermektedir. Arduino Unoyu harici güç kaynağından besleyerek (pil, adaptör gibi) çalıştırabilmemiz mümkündür. Şekil 2.4' te Uno R3'ün bölümleri görülmektedir.



Şekil 2.4. Uno görünümü.

1. USB bağlantı noktası
2. Güç bağlantı noktası (7V-12V Doğru Akım)
3. ATMEGA328
4. İletişim entegresi
5. Kristal (16MHz)
6. Resetleme butonu
7. Güç LED'i
8. Seri iletişim ledleri
9. LED
10. Power pinleri
11. Giriş pinleri (Analog)
12. Seri iletişim pini
13. Giriş-Çıkış pinleri (Dijital) (~ işaretliler PWM pini)
14. Topraklama ve AREF pini
15. ATMEGA328 ICSP
16. USB ICSP

### 2.1.2.1. *Arduino Uno Elektronik Kartının Teknik Yapısı*

- ATMEGA328 işlemcisi
- DC 5V besleme
- 7V-12V DC gerilim regüle özelliği
- 14 Giriş ve Çıkış bağlantısı, bunların 6 tanesi PWM olarak kullanılabilir.
- 6 tane Analog giriş
- Pinler başına 40mA akım kapasitesi
- Regüledi 3,3V için 50mA akım kapasitesi
- 32K Flash, bunun 0.5K' sını bootloader'e kullanılmaktadır.
- 2K SRAM
- 1K EEPROM
- 16 MHZ kristal

### 2.1.2.2. *Güç*

Arduino Uno dışarıdan adaptör, pil vb güç kaynakları ile beslenebileceği gibi dahili olarak USB kablo üzerinden de beslenebilir. Arduino AC/DC bir adaptör ile veya pil ile beslenebilir. Bu bağlantı 2.1mm power girişinden yapılmalıdır. Bağlantı yaparken mutlaka GND ve Vin bağlantılarının doğru bağlandığına dikkat edilmelidir.

VIN : Uno'ya dışarıdan besleme yapılmak istendiğinde Vin pininden voltaj girişi yapılır.

5V : Bu karta güç girişinden besleme verildiğinde 5V regülatör entegresi sayesinde uygulanan DC gerilim 5V'a dönüştürülür ve gerekli yerlere dağıtılır. Bu pin ucu harici olarak 5V'a ihtiyaç duyduğumuzda kullanılacak bağlantı noktasıdır.

3.3V : Bu kartın regüle edilmiş 3.3V pin bağlantı noktasıdır. Bu pin ucu harici olarak 3.3V'a ihtiyaç duyduğumuzda kullanılacak bağlantı noktasıdır.

GND : Ground bağlantı noktasıdır.

IOREF : Bu pin sayesinde kartın çalıştığı voltaj referansı bulunabilir. Bu bağlantıya uygun olarak tasarlanmış bir katman kartı, bu pinden kartın voltajını algılayabilir ve kendine uygun olan güç bağlantısını kullanabilir.

### 2.1.2.3. *Giriş ve Çıkışlar*

Bu kartta giriş veya çıkışı kullanabilmek için çeşitli fonksiyonlar kullanılabilir. Bu fonksiyonlar digitalRead(), digitalWrite() ve pinMode() fonksiyonlarıdır. Giriş ve çıkış olarak yönlendirilen bu pinler 5V DC ile çalıştırılır. Pin başına en fazla 40mA akım ile

kullanılabilir. Bu pinlerden daha fazla akım çekilmesi isteniyorsa pull up veya pull down dirençleri kullanılabilir.

*Seri bağlantı pinleri (0-RX, 1-TX) :* Seri bağlantı pinler sayesinde TTL seviyede data alınabilir veya gönderilebilir.

*Dış kesmeler (2pin ve 3pin) :* Bu pinler, dışarıdan yapılan bir kesmeyi algılamak için kullanılır.

*PWM: 3pin, 5pin, 6pin, 9pin, 10pin, ve 11pin :* Bu pinler sayesinde 8 bitlik analog PWM sinyali üretilebilir.

*SPI: 10pin, 11pin, 12pin, 13pin :* Seri iletişimin dışında SPI haberleşme yapılmak isteniyorsa bu pinler kullanılır.

*LED 13 :* 13 numaralı pine bağlıdır. Genellikle kartı test amaçlı kullanılır. Lojik 1 uygulandığında yanar, Lojik 0 uygulandığında söner.

Bu kartta A0, A1,A2, A3, A4 ve A5 pinleri analog sinyalleri okumak için kullanılır. Bu pinler sayesinde analog olarak 10 bitlik okuma yapabiliriz. 0 ile 5V gerilim seviyesi arasında ölçüm yapabilir.

*TWI: A4pini veya SDA ve A5pini veya SCL üzerinden Wire kütüphanesi kullanılarak TWI haberleşmesi yapılabilir.*

*AREF :* Analog okuma yapacaksak ve belirli bir referans voltajına göre okuma yapacaksak bu pin kullanılabilir.

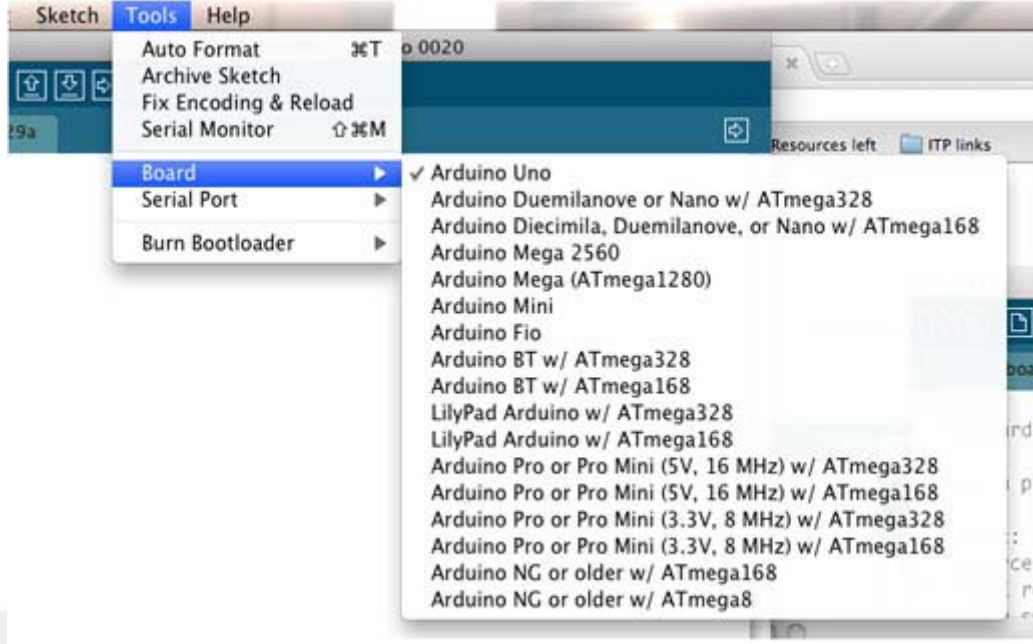
*RESET :* Kartımızı resetlemek istiyorsak bu pin kullanılır. Karta harici bir kart bağlıyorsak bu pin sayesinde kartı resetleyebiliriz.

#### 2.1.2.4. İletişim

Bu kart çeşitli iletişim yöntemleri sunmaktadır. Bunlardan biri UART TTL (5V) seri haberleşmedir. Yazılımsal ve donanımsal olarak bu seri port kullanılabilir. Kart üzerinde TX ve RX uçlarında birer adet led bağlıdır, veri transferi sırasında bu ledler yanıp sönererek veri transferi hakkında kullanıcıya bilgi verir. Yazılımsal seri portu kullanırken SOFTWARESERIAL kütüphanesini mutlaka yazılıma eklenmelidir. Bu haberleşme yönteminin yanında Arduino I2C ve SPI haberleşmeyi de destekler.

#### 2.1.2.5. Programlama

Arduino kendine ait arayüz programı ile programlanır. Bu programı Arduino'nun web sayfasından indirebilir. Programı indirdikten sonra Tools > Board menüsünden hangi Arduino'nun programlanacağı seçilmelidir.



Şekil 2.5. Arduino programı board seçimi.

Arduino üzerinde bulunan mikrodenetleyiciye önceden bir bootloader dosyası yüklüdür. Bootloader dosyası sayesinde harici bir programlayıcıya ihtiyaç duymadan Arduinoyu kolaylıkla programlanabilir. Arduino üzerinde bulunan bootloader dosyasını ISP bölümünü kullanarak devre dışı bırakılabilir. Bootloader devre dışı kaldıktan sonra ICSP pini üzerinden program yazılabilir.

#### 2.1.2.6. USB Aşırı Akım Koruması

Arduino'lar üzerinde aşırı akımı önlemek için birçok koruma sigortası mevcuttur. Bilgisayarların hemen hemen hepsinde aşırı akım koruması olmasına rağmen Arduino'larda bulunan sigortalar ekstra bir koruma sağlamaktadır. Arduino 500mA den fazla akım çekerse bu sigortalar devreye girerek aşırı akım durumunu ortadan kaldırır ve programı resetler. Bu durum aşırı akım durumu ortadan kalkıncaya dek bu şekilde devam etmektedir.

#### 2.1.3. Android Tasarımı

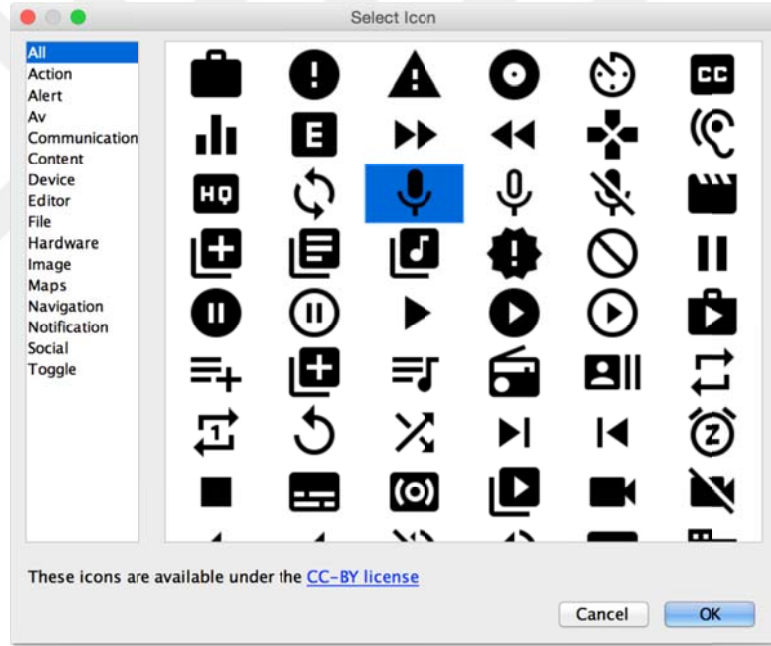
Sistemin Android platformlarda çalışabilmesi için bir android arayüz programı tasarlanmıştır. Bu sistem Android Studio 1.4 kullanılarak gerçekleştirilmiştir. 2015 yılında çıkarılan Android Studio 1.4 'ün getirdiği yenilikler şunlardır.

- Geliştirici Hizmetleri
- Performans İzleyicileri
- Proje Sihirbazı
- Tema Editörü
- Vektörel Varlıklar

1.4 sürümü kullanıcılarına geliştirici hizmetleri, performans izleyicileri, yeni tasarım araçları gibi birçok yeni hizmet sunmaktadır. Oluşturulmuş tasarım araçlarıyla kullanabileceğimiz vektörel grafikler, tema editörü ve proje şablonlarının yer aldığı proje sihirbazı gibi birçok özellik gelmektedir.

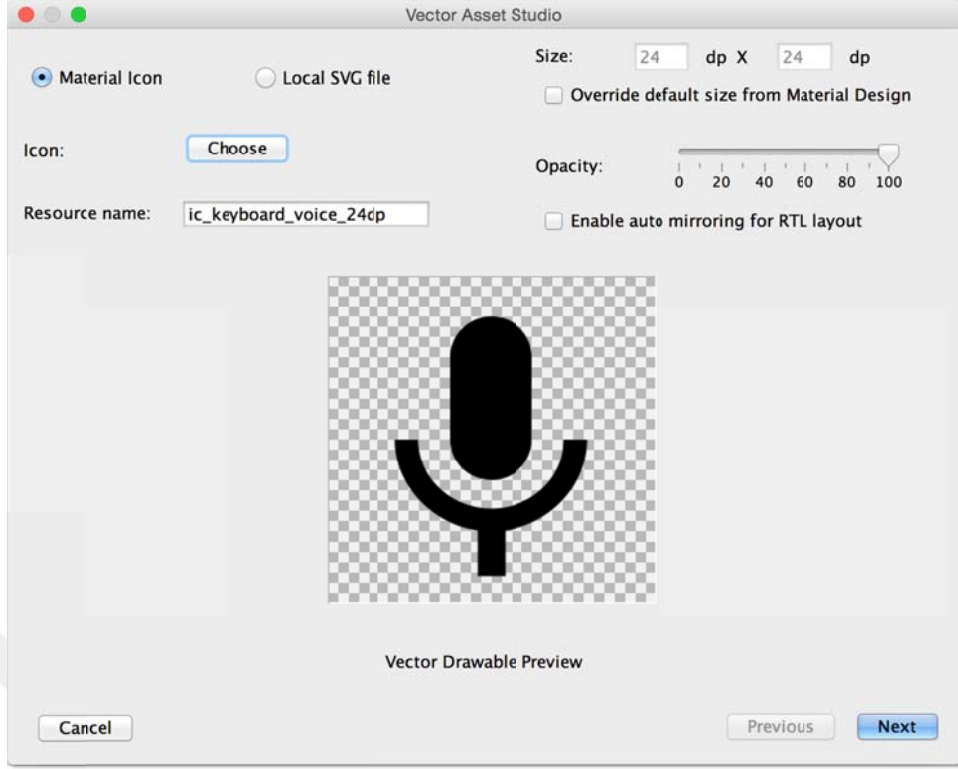
### 2.1.3.1. Vektörel Varlıklar

Android Studio'nun yeni versiyonunda Vektörel Varlıklarda bulunmaktadır.



Şekil 2.6. Android Studio vektörel varlıklar penceresi.

Projelerimizde kullanabileceğimiz birçok vektörel bileşen bu paketin içerisinde. Uygulamanız üzerinde res/drawable sağ tıklayıp Yeni>Vektör yöntemini takip ederek vektörel grafikleri uygulamanıza ekleyebilmemiz mümkündür. API 21 ve üzeri sürümlerde vektörel varlıkları kullanabilmeniz mümkündür.



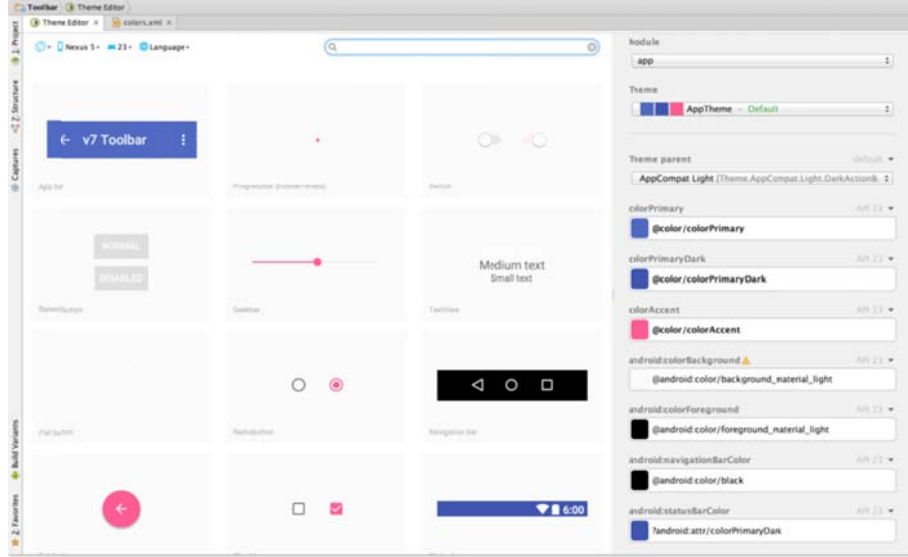
Şekil 2.7. Android Studio vektörel varlık ekleme.

Vektörel varlık çizimlerinin yanında kendi vektörel çizimlerimizi de ekleyip değişiklikler yapabiliriz.

#### 2.1.3.2. Tema Editörü

Var olan bileşenlerinizi kendinize özgü olarak tasarlayabilmek için Tema Editörü kullanılabilir. **Araçlar** → **Android** → **Tema Editorü** bağlantısı takip edilerek Tema editörüne ulaşılabilir.

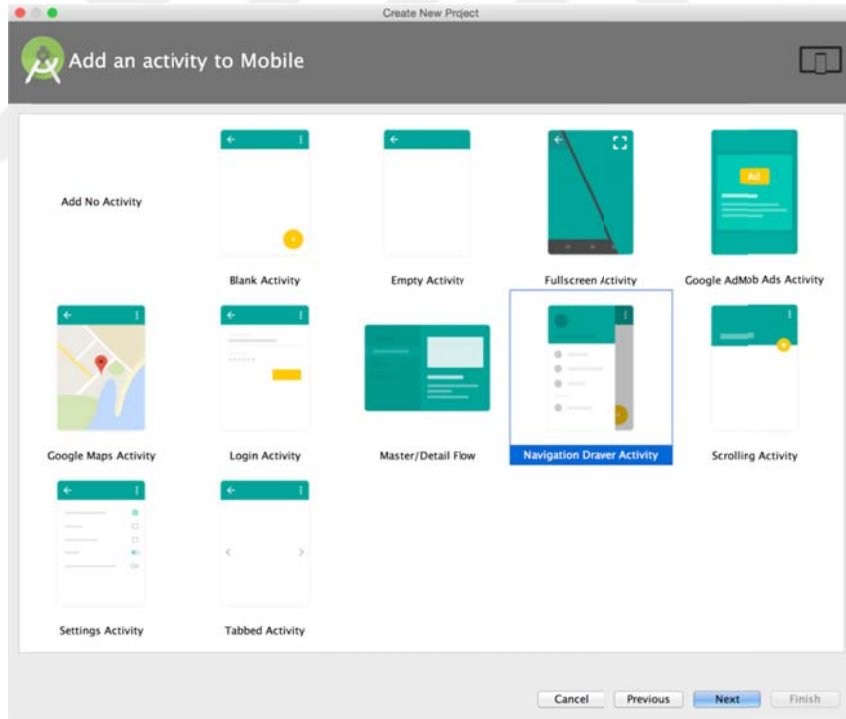




Şekil 2.8. Android Studio tema editörü.

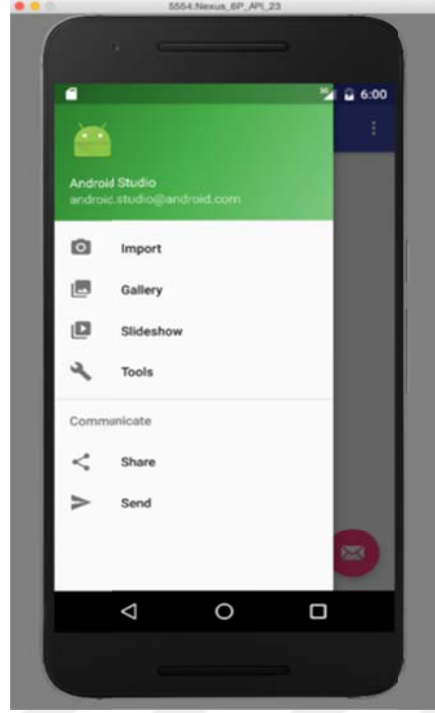
### 2.1.3.3. Proje Şablonu

Projelerinizde kullanabileceğiniz birçok Activity'ler mevcuttur.



Şekil 2.9. Android Studio proje şablonu.

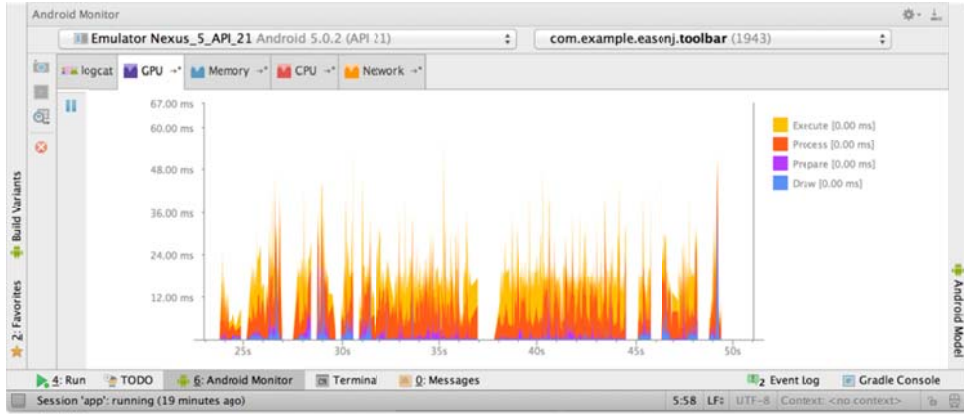
Projelere kayan bir menü eklemek için Navigation Drawer Activity'i kullanabiliriz.



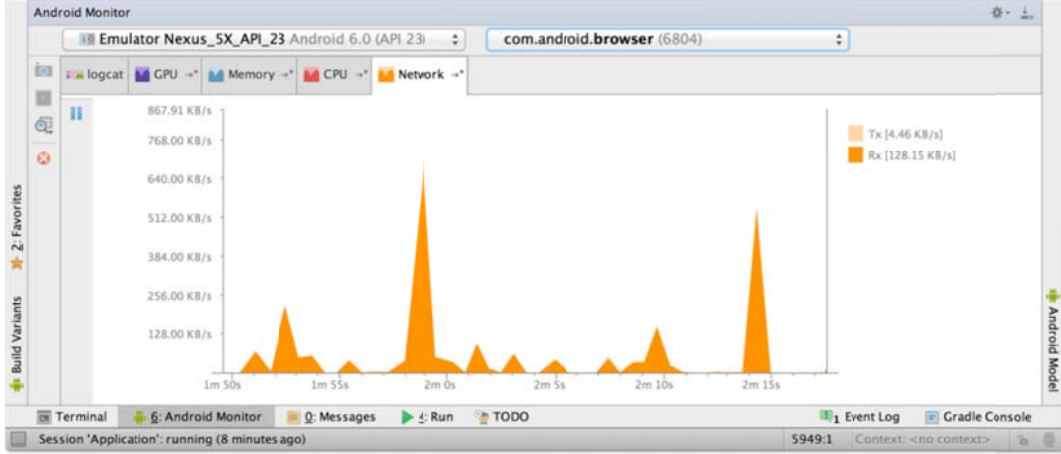
Şekil 2.10. Android Studio kayan menü örneği.

#### 2.1.3.4. Performans İzleyiciler

Ağ kullanımını ölçülemek amacıyla Ağ İzleyici ve işlemcinin performans bilgilerini görüntülemek amacıyla Grafik İşlemci İzleyicisi de gelen yenilikler arasındadır.




Şekil 2.11. Android Studio grafik işlemci izleyicisi.



Şekil 2.12. Android Studio ağ izleyicisi.

### 2.1.3.5. Geliştirici Hizmetleri

Cloud



**Firebase**  
 Firebase is a complete platform for building mobile and web applications.  
[Learn More](#)  
 Continue configuration by following the instructions [here](#)

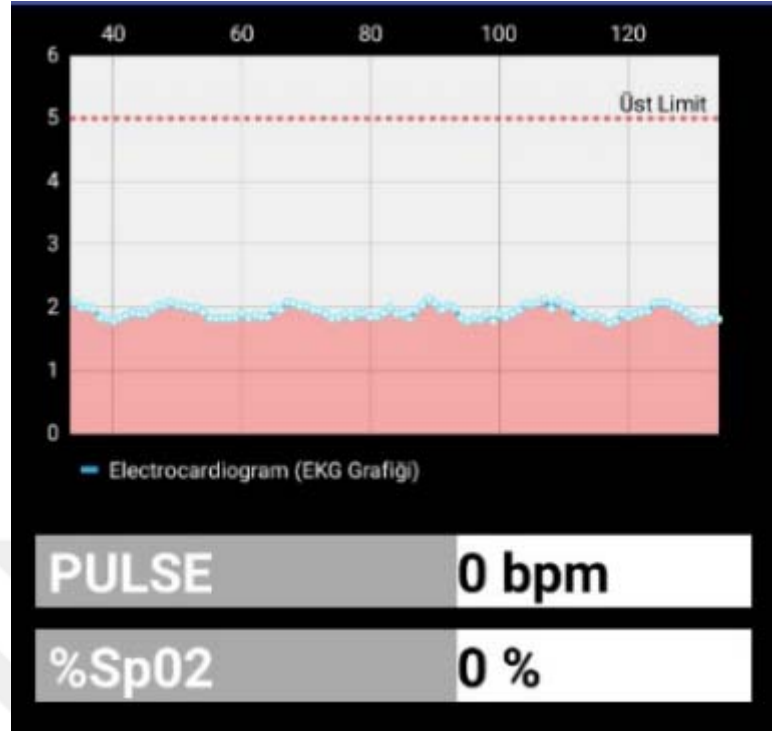
Enabling this service will...

- Add dependencies: `com.firebase:firebase-client-android:2.3.1`
- Add permissions: `INTERNET`
- Create/modify files: `AndroidManifest.xml`

Şekil 2.13. Android Studio firebase veri depolama.

Firebase veri depolama, statik hosting, kimlik doğrulaması ve birçok özellik barındırmaktadır. Bu bileşenlere ulaşmak için **Dosya** → **Proje Yapısı** → **Bulut** bağlantısını takip edebilirsiniz. 1.5.x' in daha önceki versiyonlarını güncelleyebilmek için **Yardım** > **Güncellemeleri Kontrol Et** bağlantısını takip edebilirsiniz.

#### 2.1.4. Vital Bulguların Ölçülmesi



Şekil 2.14. Vital bulguların görüntülediği ekran.

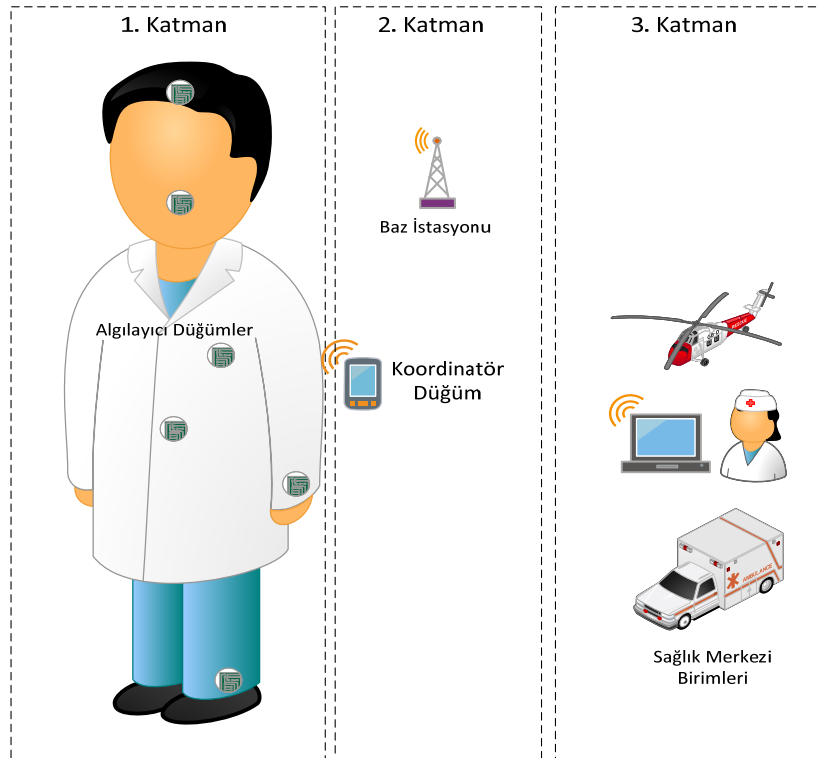
Şekil 2.14'den anlaşılacağı üzere tasarlanan sistemimiz insan vital bulgularından olan Elektrokardiyogram, Nabız ve Kandaki Oksijen Miktarını ölçümleyebilmektedir. Hastaya bağlanan bir elektronik devre tarafından ölçümlenen bu değerler kablosuz iletişim sayesinde bluetooth üzerinden içinde android sistem olan ve programın kurulu olduğu cep telefonuna gönderilmektedir. Yani elektronik sistemin iletişimde olduğu android arayüz programına veriler gönderilmektedir. Bu veriler real time yani anlık olarak gönderilmektedir ve hastadan alınan sonuçlar arayüz programında canlı olarak görülebilmektedir. Şekil 2.14'te grafik olarak görülen bölümde hastanın EKG sonuçları grafiksel olarak görülmektedir. Bu şekilde hastaya cihaz bağlanmadığı için dikkat edersek sonuçlar sabit bir şekilde devam etmektedir. Şekil 2.14'te Pulse yazan bölümde hastanın dakikadaki nabız atım hızı görülmektedir. %SpO2 yazan bölümde ise hastanın kanındaki oksijen miktarı gözlemlenebilmektedir. Grafik bölümünde de bahsettiğim gibi cihaz herhangi bir hastaya bağlanmadığı için nabız ve %SpO2 değerlerde sıfır olarak gözlemlenmektedir.

### 2.1.5. Kablosuz Veri Transferi

Oluşturulan sistemde kablosuz veri transferi Bluetooth 2.0 teknolojisi ile yapılmaktadır. Bu teknoloji, birinci nesil teknolojinin üzerinde yapılan değişimlerden sonra ortaya çıkmıştır. Bluetooth 1.x teknolojileri saniyede 1 Mbit hızında veri transferi yapabilmektedir. Günümüzde çok fazla sayıda veri iletimine ihtiyaç duyulmasından dolayı Bluetooth 2.x teknolojisine geçilmiştir. Bu teknoloji sayesinde saniyede 3 Mbit veri iletimi sağlanmıştır. Çalışma frekans bandı ise 2.4 GHZ aktarım bandıdır. Bluetooth 2.0 teknolojisi günümüzde en çok kullanılan bluetooth teknolojisidir.

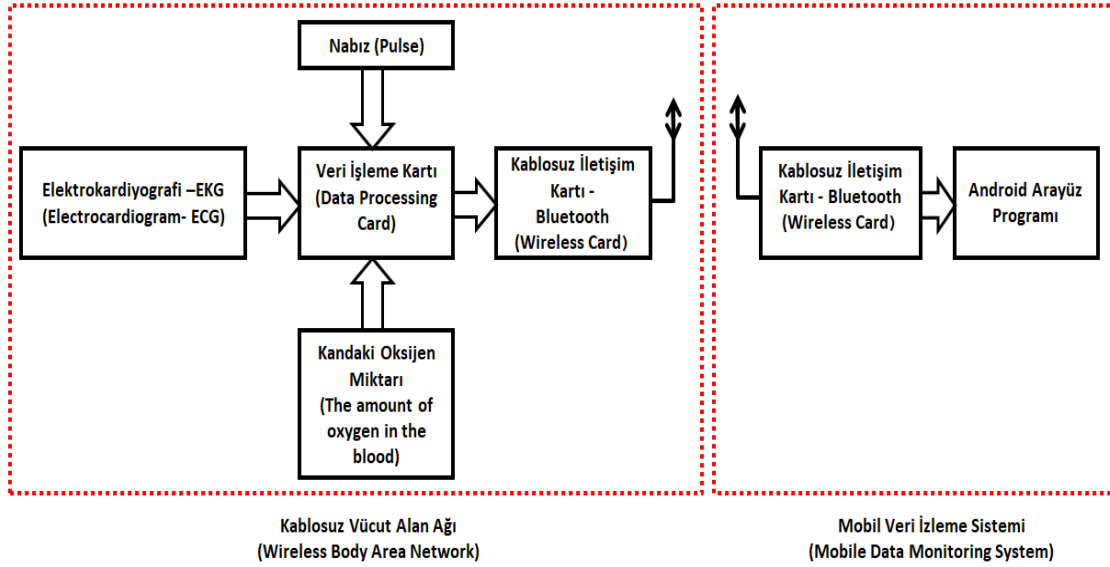
### 2.1.6. Vücut Alan Ağları

Vücut alan ağları, insan vücudunun etrafında ya da üzerinde bulunan algılayıcı veya eyleyici özellikleri üzerinde barındıran algılayıcı düğümlerin oluşturduğu radyo frekans tabanlı kablosuz ağ teknolojisidir. Vücut Alan Ağları, ağın oluşmasını sağlayan eyleyiciler ve algılayıcılar vasıtasıyla insan vücudunun fonksiyonlarını ve çevresinde olup bitenleri izlemeye yarar. Ayrıca ağ içinde bulunan koordinatör algılayıcı, düğümden gelen verileri istenilen sağlık merkezine güvenilir bir şekilde iletmektedir. Sistem bu şekilde çalışarak hastanın sürekli gözetim altında alınmasını sağlamaktadır. Şekil 2.15'te genel bir Vücut Alan Ağı katmanlı yapısının şekli görülebilmektedir.



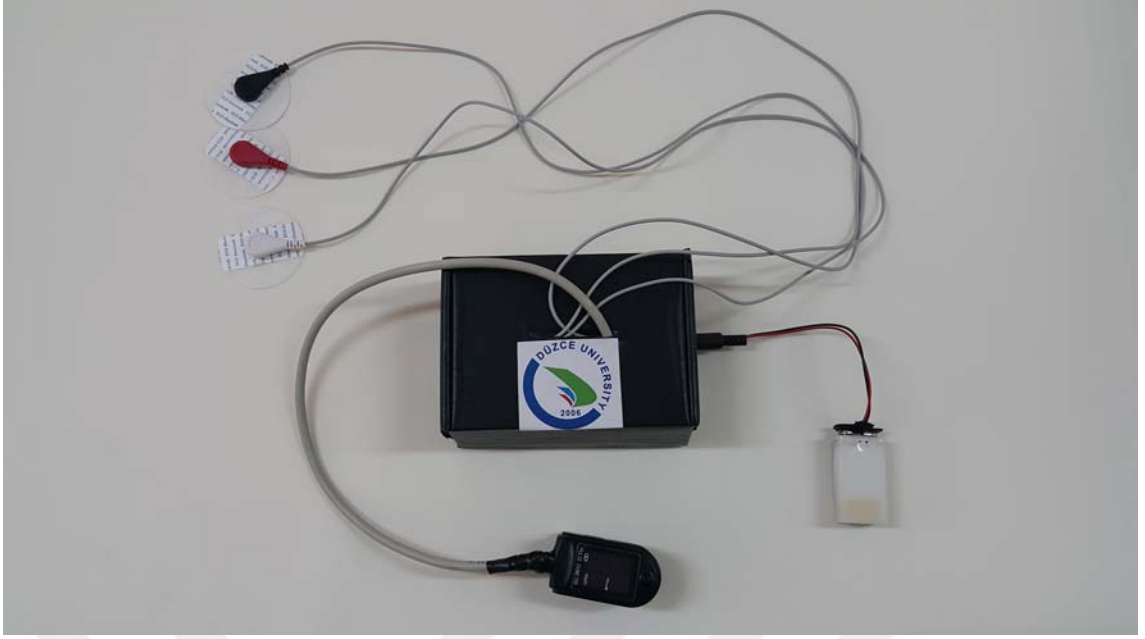
Şekil 2.15. Vücut alan ağı katmanlı yapısı.

## 2.2. TASARLANAN SİSTEMİN ÇALIŞMASI



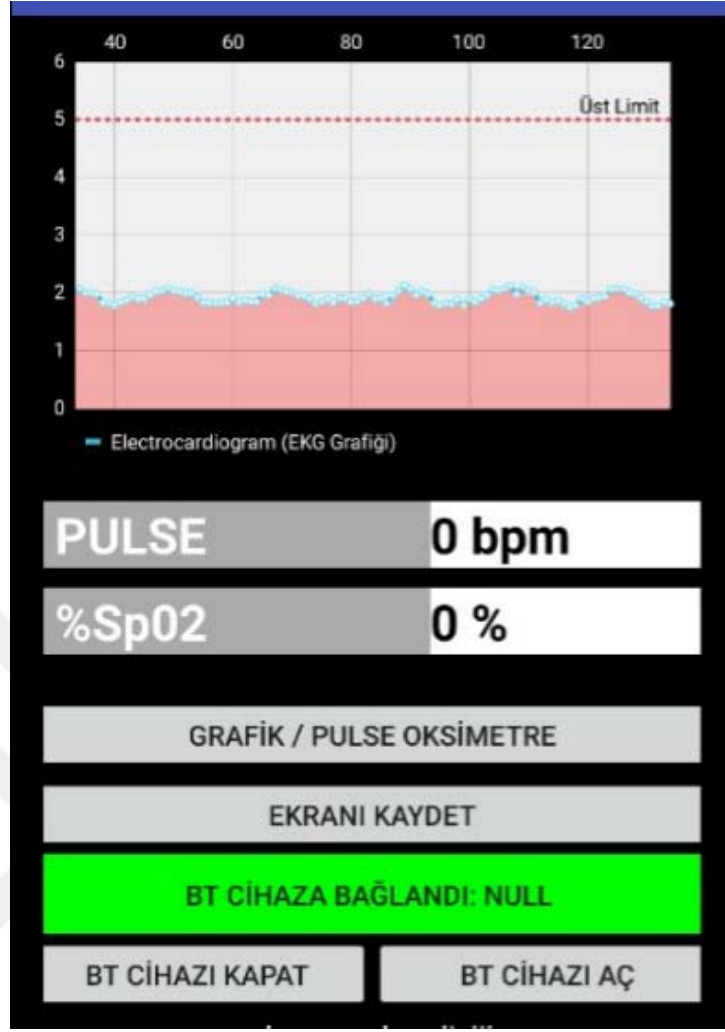
Şekil 2.16. Sistemin blok diyagramı.

Tasarlanan sistem Şekil 2.16’da görüleceği üzere iki ayrı kablosuz bölüm olarak oluşturulmuştur. Birinci bölüm KVAA bölümünü oluştururken, ikinci bölüm ise verilerin izlenebildiği mobil veri izleme sistemi olarak tasarlanmıştır. KVAA bölümüne dikkat edecek olursak düğümler ve bunlara bağlı olduğu koordinatör düğüm görülmektedir. Şekil 2.16’da KVAA bölümünde görüleceği üzere düğümlerden hastanın vital bulgularından olan nabız, kandaki oksijen miktarı ve EKG bilgileri gelmektedir. Bu bilgiler bir koordinatör düğüm olan veri işleme kartında toplanmaktadır. Veri işleme kartı bu verileri değerlendirerek, gerektiği şekilde kablosuz iletişim kartına yönlendirmektedir. Kablosuz iletişim kartı Bluetooth 2.0 teknolojisini kullanarak verileri bağlı olduğu bluetooth cihaza göndermektedir. Bu iki cihaz arasında çift yönlü haberleşme işlemi gerçekleştirebilmektedir. Sistemin bitmiş fiziksel hali Şekil 2.17’de görülmektedir.



Şekil 2.17. Tasarlanan sistem.

Oluşturulan sistemin ikinci bölümünde kablosuz olarak bluetooth aygıtı üzerinde gelen veriler mobil cihaz tarafından alınır. Android olarak mobil cihazlar için tasarlanmış olduğumuz arayüz programında KVAA'dan gönderilen nabız değeri sayısal olarak dakikadaki kalp ritmini gösterir. Kandaki oksijen miktarı Spo2 değeri % olarak kanın içindeki oksijen miktarı değerini belirtir. EKG değeri ise arayüz programının üst bölümünde grafiksel olarak gerçek zamanlı değerlere bağlı olarak şekilsel olarak gösterilmektedir. Android Studio programı kullanılarak tasarlanan arayüz programı Şekil 2.18'de gösterilmektedir. Şekil 2.18'de görüleceği gibi Android tabanlı mobil sağlık izleme programı yeşil renkte görüleceği üzere "NULL" isimli KVAA bluetooth cihaza bağlanmıştır. Fakat KVAA cihazda ölçüm yapılmadı için değerler bu şekilde gözlemlenmektedir.

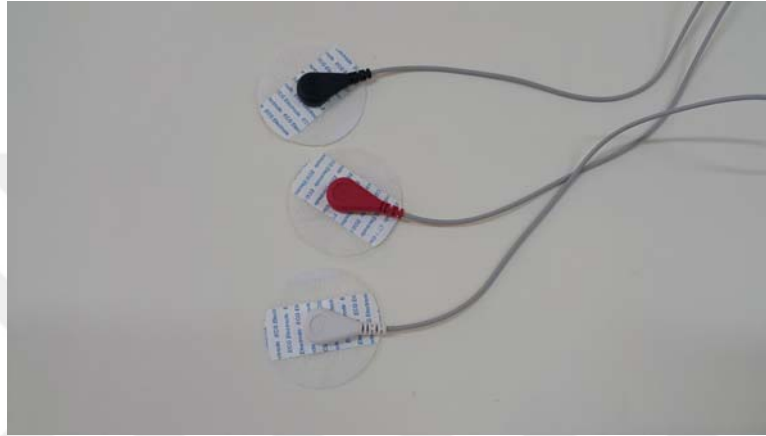


Şekil 2.18. Tasarlanan mobil program.



### 3. BULGULAR VE TARTIŞMA

Sistem fiziksel olarak Şekil 2.17’de görüleceği üzere 4 bölümden oluşmaktadır. Birinci bölüm EKG sinyallerinin ölçülmesini sağlayan ve Şekil 3.1’de gösterilen ölçüm sensörlerinden oluşmaktadır.



Şekil 3.1. EKG ölçüm sensörü.

İkinci bölüm, nabız ve SpO2 değerlerini ölçülmesini sağlayan ve Şekil 3.2’de gösterilen sensörden oluşur.



Şekil 3.2. Nabız ve SpO2 ölçüm sensörü.

Üçüncü bölüm, devrenin beslenmesi için kullanılan 9V şarjlı pil kullanılarak oluşturulan ve Şekil 3.3’te gösterilen besleme bölümüdür.



Şekil 3.3. Sistemin besleme ünitesi.

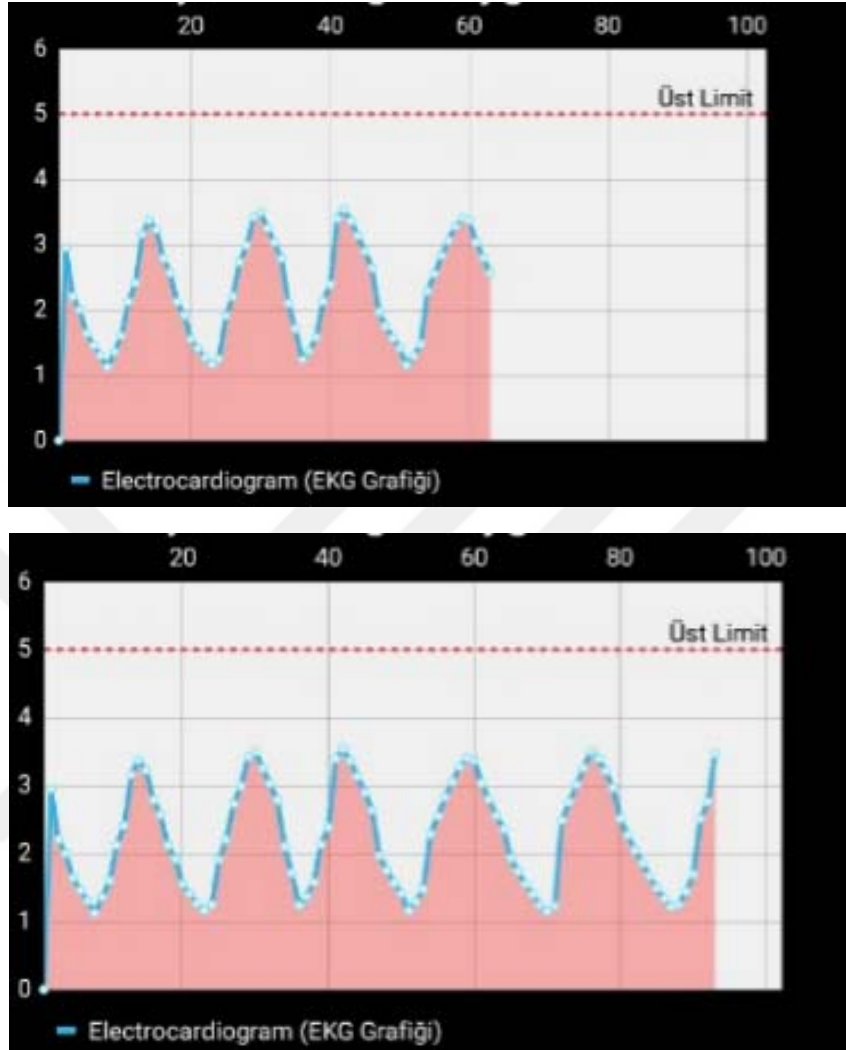
Dördüncü ve son bölüm olan yer ise tüm verilerin toplandığı koordinatör düğümün ve bluetooth cihazın bulunduğu Şekil 3.4'te gösterilen bölümdür.



Şekil 3.4. Koordinatör düğüm ve iletişim ünitesi.

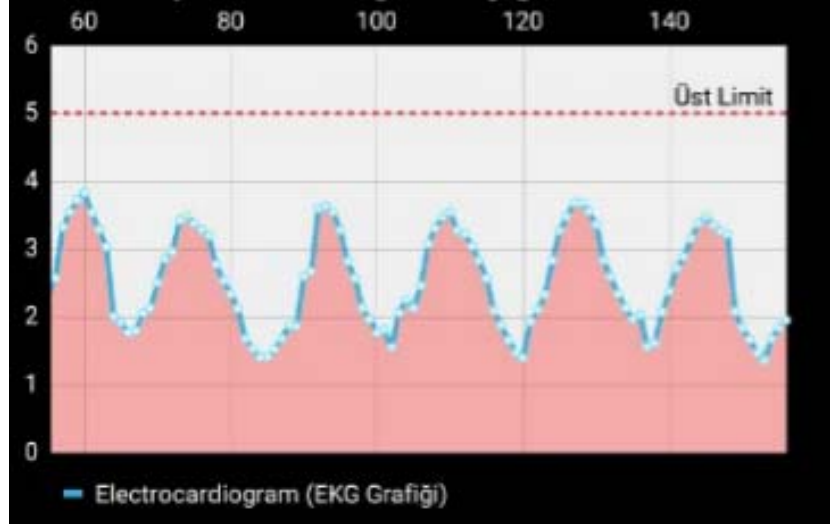
Tasarlanan mobil sağlık izleme sistemini analiz ederken 2 farklı yöntem uygulanmıştır. Bu yöntemlerden birinde sadece EKG değerleri gönderilerek sistemin grafiksel performansı incelenmiştir. İkinci yöntem ise EKG, nabız ve SpO2 değerleri gönderilerek sistemin performansı incelenmiştir. EKG değerlerinin ise kendi içinde 3 yöntem denenmiştir. Öncelikle 1 sn aralıklarla veriler gönderilmiştir ve verilerin ekran

grafığının yavaş bir şekilde Şekil 3.5'te görüldüğü gibi çizim şeklinin gözle görülebilecek şekilde ekranda oluştuğu gözlemlenmiştir.



Şekil 3.5. EKG ilk test ölçüm sonuçları.

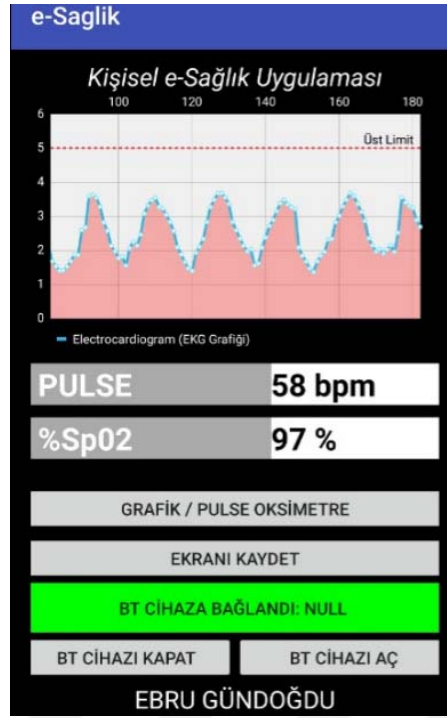
EKG değerlerinin görüntülenmesinde ikinci yöntem ise, EKG değerlerinin 50 ms aralıklarla gönderilmesidir. Bu değerleri gönderdiğimizde Şekil 3.6'da görülen grafikler elde edilmektedir.



Şekil 3.6. EKG son test ölçüm sonuçları.

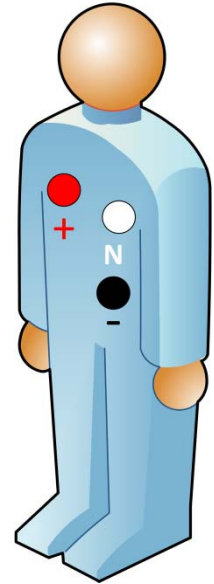
Son yöntemde ise 50 ms altında değerlerle veri göndermek işlemi gerçekleştirilmiştir. 50ms altında değerlerde veri gönderdiğimizde sistem belirli bir süre çalıştıktan sonra Android İşletim Sistemi hata mesajı vermeye başladığı ve sistemi zararlı yazılım olarak algılamaya başladığı gözlemlenmiştir. Yapılan denemeler ve araştırmalar sonucunda, sisteme tanımladığımız grafik çizim kütüphanesinin çizimleri düzgün bir şekilde gösterebilmesi ve verileri işleyebilmesi için belirli bir sürenin geçmesi gerektiği gözlemlenmiştir. Bu nedenden dolayı 50 ms altında değerlerde veriler işlenmeden tekrar veri gönderdiğimizde sistem hata algılamakta ve yaptığımız yazılım otomatik olarak kapanmaktadır.

Android programda çok fazla işlem gerektiren EKG veri testleri bittikten sonra sistem çalıştırıldığında, gözlemlerimize göre 65ms değeri ile verimli bir ölçüm ve gözlem gerçekleştirilmiş olmaktadır. Sistemin tamamen çalışmasına ait olan android arayüz Şekil 3.7'de görülmektedir.



Şekil 3.7. Sistem için tasarlanan android programı.

Tasarlanan sistemde nabız, SpO2 değerlerinin ölçülmesi ve EKG problemlerinin vücuda nasıl takılacağına ilişkin görsel Şekil 3.8’de görülmektedir.



Şekil 3.8. Nabız, SpO2 ölçülmesi ve EKG problemlerinin bağlantıları.

#### 4. SONUÇLAR VE ÖNERİLER

Sonuç olarak Şekil 2.16'daki blok diyagram elektronik ve yazılımsal olarak oluşturulmuştur. Oluşturulan sistem içindeki KVAA ile insanın vital bulgularından olan nabız, kandaki oksijen miktarı ve EKG değerleri algılayıcılar aracılığıyla ölçümlenmiştir. Ölçümlenen bu değerler koordinatör düğüm olarak görev yapan elektronik devreye gönderilmiştir. Gönderilen bu değerler koordinatör düğüm aracılığıyla değerlendirilerek Bluetooth 2.0 kablosuz veri iletişimi teknolojisi ile mobil android cihaza gönderilmiştir. Android cihaza bluetooth aracılığı ile gelen veriler Android Studio ile tasarlanan arayüz programında istenilen yerlerde kullanıcıya gösterilmiştir. Bu şekilde hastanın gözlenmesi gereken vital bulguları büyük ve pahalı cihazlar yerine mobil ve daha ucuz bir yöntem olarak gerçekleştirdiğimiz sistem ile çok kolay bir şekilde kullanılabilir. Hastalar çok rahat bir şekilde izlenebilir. Günümüzde bu ölçümleri yapabilmek için hastaya bağladığımızda hasta hareket etmekte zorlanmaktadır. Bu sistem hastanın hareketini de kolaylaştırmaktadır.

Bir sonraki aşamada KVAA aracılığı ile ölçümlenen veriler bu sisteme aktarılmanın yanında internet üzerinden anlık olarak da aktarılması planlanmaktadır. Bu şekilde sürekli gözlem altında olması gereken hastaları hastaneye yatırmak yerine kendilerine bu tarz bir sistem vererek hastanın günlük yaşantısına devam etmesi ya da bu süreci evinde geçirmesi sağlanabilecektir. Sonuçta, hastaneden bağımsız olarak hasta sürekli gözlem altında kalabilmesi sağlanacaktır. Bu şekilde para, zaman ve mekândan tasarruf edilebilecektir.

## 5. KAYNAKLAR

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] A. Karahan, İ. Ertürk, S. Atmaca, ve S. Çakıcı, “TDMA tabanlı kablosuz algılayıcı ağ ortam erişim kontrol protokolleri için genel analitik ve benzetim modelleri,” *Uluslararası Teknolojik Bilimler Dergisi*, c. 5, s. 1, ss. 128-141, 2013.
- [3] M. R. Yuce, P. C. Ng and J. Y. Khan, “Monitoring of physiological parameters from multiple patients using wireless sensor network,” *Journal Medical Systems*, vol. 32, pp. 433–441, 2008.
- [4] L. Zhang and X. Wu, “Recent progress in challenges of wireless biomedical sensor network,” in *Proceedings of the IEEE 3rd International Conference on Bioinformatics and Biomedical Engineering (ICBBE 2009)*, Beijing, China, 2009.
- [5] K. Gündoğdu, “Algılayıcı ağ kontrollü engelli aracı tasarımı,” Yüksek lisans tezi, Elektrik-Elektronik ve Bilgisayar Mühendisliği Bölümü, Düzce Üniversitesi, Düzce, Türkiye, 2015.
- [6] A. Calhan and S. Atmaca, “A new network coordinator node design selecting the optimum wireless technology for wireless body area networks,” *KSII Transactions on Internet and Information Systems*, vol. 7, no. 5, pp. 1077-1093, 2013.
- [7] A. Çalhan, “Trafik duyarlı kablosuz vücut alan ağlarının başarımlarının analizi,” *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 2015.
- [8] H. Cao, V. Leung, C. Chow and H. Chan, “Enabling technologies for wireless body area networks: Survey and outlook,” *IEEE Communications Magazine*, vol. 47, no. 12, pp. 84–93, 2009.
- [9] M. A. Hanson, et al., “Body area sensor networks: Challenges and opportunities,” *IEEE Computer*, vol. 42, no. 1, pp. 58–65, 2009.
- [10] F. Jingling, L. Wei and L. Yang, “Performance enhancement of wireless body area network system combined with cognitive radio,” in *Proceedings of Communications and Mobile Computing (CMC)*, Shenzhen, China, 2010, pp. 313-317.
- [11] R. Chavez-Santiago and I. Balasingham, “Cognitive radio for medical wireless body area networks,” in *Proceedings of Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2011 IEEE 16th International Workshop on , Kyoto, Japan, 2011, pp. 148-152.
- [12] K. A. Ali, J. H. Sarker and H. T. Mouftah, “A MAC protocol for cognitive wireless sensor body area networking,” *Wireless Communications & Mobile Computing*, vol. 10, pp. 1656–1671, 2010.
- [13] S. Sharma, A. L. Vyas, B. Thakker, D. Mulvaney and S. Datta, “Wireless body area network for health monitoring,” in *Proceedings of Biomedical Engineering and Informatics (BMEI) 4th International Conference on*, Shanghai, China, 2011,

pp. 2183-2186.

- [14] S. N. Ramli and R. Ahmad, "Surveying the wireless body area network in the realm of wireless communication," in *Proceedings of information Assurance and Security (IAS) 7th International Conference on*, Melaka, Malaysia, 2011, pp. 58-61.
- [15] E. Gündoğdu, K. Gündoğdu ve A. Çalhan, "Kablosuz vücut alan ağları ile mobil sağlık izleme uygulaması," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 4, ss. 640-645, 2016.
- [16] H. Gürkan, S. Yarman, ve A. N. Gönülenen, "Elektrokardiyogram ( EKG ) işaretlerinin temel tanım ve zarf fonksiyonları ile modellenmesi," *İstanbul Teknik Üniversitesi Dergisi*, c. 5, ss. 49-57, 2006.
- [17] B. R. Nandkishor and A. B. A. N. Architecture, "Android smartphone based body area network for monitoring and evaluation of medical parameters," *Networks & Soft Computing (ICNSC), 2014 First International Conference on*, Guntur, India, 2014, pp. 284-288.
- [18] K. Gündoğdu and A. Çalhan, "An implementation of wireless body area networks for improving priority data transmission delay," *Journal of Medical Systems*, vol. 40, no. 3, pp. 1-7, 2016.
- [19] A. Ahmad, A. Riedl, W. J. Naramore, N. Y. Chou and M. S. Alley "Scenario-Based traffic modeling for data emanating from medical instruments in clinical environment," *Computer Science and Information Engineering, 2009 WRI World Congress on*, Los Angeles, CA, USA, 2009, pp.529-533.
- [20] Z. He and X. Bai, "A wearable wireless body area network for human activity recognition," *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conference on*, Shanghai, China, 2014, pp. 115-119.
- [21] Ö. Eriş, H. Korkmaz, K. Toker, ve A. Buldu, "İnternet üzerinden hasta takibi amaçlı pic mikrodenetleyici tabanlı kablosuz pals-oksometre ölçme sistemi tasarımı ve LabVIEW uygulaması," *VII. Ulusal Tıp Bilişimi Kongresi*, Magusa, KKTC, 2010, pp. 16-25.
- [22] F. Aktaş, C. Çeken and Y. E. Erdemli, "IoT-Based healthcare framework for biomedical applications," *Journal Of Medical And Biological Engineering*, pp. 1-14, 2017.
- [23] S. P. Kumar, V. Richard, R. Samson, U. B. Sai, P. L. S. D. M. Rao, and K. K. Eswar, "Smart health monitoring system patient through IoT," *I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2017, pp. 551-556.
- [24] D. De, A. Mukherjee, A. Sau, and I. Bhakta, "Design of smart neonatal health monitoring system using SMCC," *Healthcare Technology Letter*, vol. 4, no. 1, pp. 13-19, 2017.
- [25] R. B. Caldo et al., "Development of Wi-Fi- based switch control system for home appliances using android phone," *Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) 2015 International Conference on*, Cebu City, Philippines, 2015.
- [26] M. S. Öcal, "Android kontrollü wifi keşif robotu," Yüksek lisans tezi, Elektrik-Elektronik Mühendisliği Bölümü, Haliç Üniversitesi, İstanbul, Türkiye, 2017.



- [27] C. Doukas, T. Pliakas and I. Maglogiannis, "Mobile healthcare information management utilizing cloud computing and android OS," *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, Buenos Aires, Argentina, 2010, pp. 1037–1040.
- [28] M. Kh'tour, "Android tabanlı hastane uygulaması," Yüksek lisans tezi, Bilgisayar Mühendisliği Bölümü, Kırıkkale Üniversitesi, Kırıkkale, Türkiye, 2015.
- [29] B. Takgil, "Android mobil uygulama testi," Yüksek lisans tezi, Bilgisayar Mühendisliği Bölümü, Düzce Üniversitesi, Düzce; Türkiye, 2015.
- [30] K. Avcı, "Android tabanlı uzaktan arıza tespit platformu," Yüksek lisans tezi, Elektrik-Bilgisayar Anabilim Dalı, Marmara Üniversitesi, İstanbul, Türkiye, 2017.
- [31] S. Spat, K. Theuermann, B. Holl, P. Beck, and T. R. Pieber, "Development, integration and operation of mobile, Android-based medical devices in hospitals: Experiences from the GlucoTab system," *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, Athens, Greece, 2015, pp. 128–131.
- [32] F. Nasri, N. Moussa, and A. Mtibaa, "Smart mobile system for health parameters follow ship based on WSN and android," *Computer and Information Technology (WCCIT), 2013 World Congress on*, Sousse, Tunisia, 2013.
- [33] Robotiksistem, (2018, 10 Mart), *Arduino Nedir*  
[Online]. Erişim: <http://www.robotiksistem.com>
- [34] K. Gündoğdu, *StemKodLab ile Kolay Kodlama*, 1. Baskı, Bursa, Türkiye: Bilgin Matbaa, 2018, böl. 5, ss. 14-29.
- [35] Robotiksistem, (2018, 10 Mart), *Arduino Uno*  
[Online]. Erişim: <http://www.robotiksistem.com>
- [36] Geleceği Yazarlar/Turkcell, (2018, 15 Ocak), *Android Studio'yu Tanıyalım*  
[Online]. Erişim: <https://gelecegiyazarlar.turkcell.com.tr>

## 6. EKLER

### 6.1. EK 1: ANDROID STUDIO KODLARI

```
package com.ebru.e_saglik;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.media.MediaScannerConnection;
import android.net.Uri;
import android.os.Environment;
import android.os.SystemClock;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.Legend;
import com.github.mikephil.charting.components.LimitLine;
import com.github.mikephil.charting.components.XAxis;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.utils.ColorTemplate;
import com.google.android.gms.appindexing.AppIndex;
import com.google.android.gms.common.api.GoogleApiClient;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.UnsupportedEncodingException;
```

```

import java.lang.reflect.Method;
import java.util.ArrayList;
import java.util.Date;
import java.util.Set;
import java.util.UUID;

public class MainActivity extends AppCompatActivity {

    Button ac;
    Button kapat;
    Button resimButon;
    private Button grafikPulseButon;
    LineChart mChart;
    private GoogleApiClient client;
    ImageView imageView;
    Button aramaButonu;

    private TextView mBluetoothStatus;
    private BluetoothAdapter myBluetooth = null;
    private Set<BluetoothDevice> pairedDevices;
    String address = null;
    String name = null;
    String nabiz=null;
    String oksijen=null;
    String stringEKG=null;
    String readMessage = null;

    float EKG = 0;
    int startIndexNabiz=0;
    int startIndexOksijen=0;
    int startIndexEKG=0;
    int startIndexPulse=0;
    TextView textViewPulseDeger;
    TextView textViewSpo2Deger;
    int sayac=1;
    boolean grafikPulse=true;

    private BluetoothSocket mBTSocket = null;
    private final static int REQUEST_ENABLE_BT = 1;
    private final static int MESSAGE_READ = 2;
    private final static int CONNECTING_STATUS = 3;
    private android.os.Handler mHandler;
    private ConnectedThread mConnectedThread;
    private final String TAG =
MainActivity.class.getSimpleName();

    ArrayList<Entry> values = new ArrayList<Entry>();

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    ac = (Button) findViewById(R.id.button6);
    aramaButonu = (Button) findViewById(R.id.button7);
    kapat = (Button) findViewById(R.id.button3);
    resimButon = (Button) findViewById(R.id.button5);
    grafikPulseButon = (Button)
findViewById(R.id.button2);
    grafikPulseButon.setBackgroundColor(Color.LTGRAY);
    myBluetooth = BluetoothAdapter.getDefaultAdapter();

    textViewPulseDeger = (TextView)
findViewById(R.id.textViewPulseDeger);
    textViewSpo2Deger = (TextView)
findViewById(R.id.textViewSpo2Deger);

    mChart = (LineChart) findViewById(R.id.linechart);
    mChart.getDescription().setEnabled(false);
    mChart.setTouchEnabled(true);
    mChart.setDragDecelerationFrictionCoef(0.9f);
    mChart.setDragEnabled(true);
    mChart.setScaleEnabled(true);
    mChart.setDrawGridBackground(true);
    mChart.setHighlightPerDragEnabled(true);
    mChart.setBackgroundColor(Color.BLACK);
    mChart.setMaxVisibleValueCount(10);

    drawLineChart(); // grafiği çizme fonksiyonunu
çağırdık

    mHandler = new android.os.Handler() {
        public void handleMessage(android.os.Message
msg) {
            if (msg.what == MESSAGE_READ)
            { // mesajı okuyor.
                readMessage = null;

                try
                {
                    readMessage = new String((byte[])
msg.obj, "UTF-8");
                }
                catch (UnsupportedEncodingException e)
                {

```

```

        e.printStackTrace();
    }

    startIndexPulse
=readMessage.indexOf("P");
    startIndexNabiz =
=readMessage.indexOf("A");
    startIndexOksijen=
=readMessage.indexOf("B");
    startIndexEKG
=readMessage.indexOf("C");

    if(grafikPulse==true &&
startIndexEKG>0)
    {
        stringEKG =
=readMessage.substring(startIndexEKG-4, startIndexEKG);
EKG=Float.valueOf(stringEKG.trim()).floatValue();

        ekleGrafik(EKG);
    }

    else if(grafikPulse==false &&
startIndexNabiz>0 && startIndexOksijen>0 &&
startIndexPulse>0) // pulse çalıştır
    {

        nabiz =
=readMessage.substring(startIndexPulse+1, startIndexNabiz);
        textViewPulseDeger.setText(nabiz +
" bpm");

        oksijen =
=readMessage.substring(startIndexNabiz + 1,
startIndexOksijen);
        textViewSpo2Deger.setText(oksijen +
" %");
    }

    readMessage = null;
}

```

```

        if (msg.what == CONNECTING_STATUS) {
            if (msg.arg1 == 1) {

                aramaButonu.setText("Bt Cihaza
Bağlandı: "+ name);

                aramaButonu.setBackgroundColor(Color.GREEN);

            }

            else {

                aramaButonu.setBackgroundColor(Color.rgb(214, 215, 215));
                aramaButonu.setText("Bağlantı
Sağlanamadı");
            }
        }
    };

    if (myBluetooth == null)
    {

        Toast.makeText(getApplicationContext(),
"Bluetooth Aygıt Kullanılamaz veya Yok",
        Toast.LENGTH_LONG).show();

        finish();
    } else
    {

        if (myBluetooth.isEnabled()) // bluetooth
bağlantısı varsa
        {

            veriGonder();

        }
        else
        {

            Intent turnBTon = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(turnBTon, 1);
        }
    }
}

```

```

myBluetooth = BluetoothAdapter.getDefaultAdapter();

registerForContextMenu(aramaButonu);

ac.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if (myBluetooth == null) {

            Toast.makeText(MainActivity.this,
"Bluetooth Ayrıtı Bulunamadı", Toast.LENGTH_LONG).show();
        } else {
            if (!myBluetooth.isEnabled()) {
                Intent bluetoothBaslat = new
Intent(myBluetooth.ACTION_REQUEST_ENABLE);

startActivityForResult(bluetoothBaslat, 1);
                // durum.setText("Bluetooth
Açıldı.");

                Toast.makeText(MainActivity.this,
"Bluetooth Açıldı", Toast.LENGTH_LONG).show();
            }
        }
    }
});

aramaButonu.setOnClickListener(new
View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        openContextMenu(aramaButonu);
    }
});

kapat.setOnClickListener(new View.OnClickListener()
{

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

```

```

        if (myBluetooth.isEnabled()) {
            myBluetooth.disable();
            // durum.setText("Bluetooth
Kapatıldı.");
            Toast.makeText(MainActivity.this,
"Bluetooth Kapatıldı.", Toast.LENGTH_LONG).show();

aramaButonu.setBackgroundColor(Color.rgb(214,215,215));
            aramaButonu.setText("Bt Cihaza
Bağlan");
        }
    });

    resimButon.setOnClickListener(new
View.OnClickListener()
    {
        public void onClick(View v) {

            takeScreenshot();

        }

    });

    grafikPulseButon.setOnClickListener(new
View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            if(grafikPulse==true)
            {
                grafikPulse=false;

grafikPulseButon.setBackgroundColor(Color.rgb(255, 64,
64));

                grafikPulseButon.setText("Grafik /
Pulse Oksimetre = 'PULSE'");
                msg("Pulse Aktif");
            }
            else if(grafikPulse==false)
            {
                grafikPulse=true;

```



```

grafikPulseButon.setBackgroundColor(Color.LTGRAY);
        grafikPulseButon.setText("Grafik /
Pulse Oksimetre = 'GRAFİK'");
        msg("Grafik Aktif");
    }

    });

    client = new
GoogleApiClient.Builder(this).addApi(AppIndex.API).build();

}

private void drawLineChart()
{

    values.add(new Entry(sayac, 0));

    LineDataSet set1 = new LineDataSet(values,
"Electrocardiogram (EKG Grafiği)"); // grafikte çizgi
oluşturuldu

    set1.setAxisDependency(YAxis.AxisDependency.LEFT);
    set1.setColor(ColorTemplate.getHoloBlue());

set1.setValueTextColor(ColorTemplate.getHoloBlue());
    set1.setLineWidth(2f);
    set1.setDrawCircles(true);
    set1.setDrawValues(true);
    set1.setFillAlpha(100);
    set1.setFillColor(Color.rgb(255, 64, 64));
    set1.setHighLightColor(Color.rgb(244, 117, 117));
    set1.setDrawCircleHole(true);
    set1.setDrawFilled(true);

    LineData data = new LineData(set1);
    data.setValueTextColor(Color.BLACK);
    data.setValueTextSize(10f);

    mChart.setData(data);
    mChart.invalidate();

    LimitLine upper_limit = new LimitLine(5f,
"Üst Limit");
    upper_limit.setLineWidth(2f);
    upper_limit.enableDashedLine(10f,10f,0f);

```

```

upper_limit.setLabelPosition(LimitLine.LimitLabelPosition.R
IGHT_TOP);
    upper_limit.setTextSize(10f);

    Legend lg = mChart.getLegend();
    lg.setForm(Legend.LegendForm.LINE);
    lg.setTextColor(Color.WHITE);
    XAxis x1 = mChart.getXAxis();
    x1.setTextColor(Color.WHITE);
    x1.setDrawGridLines(true);
    x1.setGranularity(1f);

    YAxis y1 = mChart.getAxisLeft();
    y1.setTextColor(Color.WHITE);
    y1.setAxisMaximum(6f);
    y1.setAxisMinimum(0f);
    y1.setDrawGridLines(true);
    y1.addLimitLine(upper_limit);

    YAxis y12= mChart.getAxisRight();
    y12.setEnabled(false);

}

@Override
public void onCreateContextMenu(ContextMenu menu, View
v, ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    menu.setHeaderTitle("Bluetooth Aygıtı Seçiniz");

    pairedDevices = myBluetooth.getBondedDevices();

    int i = 0;

    if (pairedDevices.size() > 0)
    {
        for (BluetoothDevice bt : pairedDevices)
        {
            menu.add(0, i, 0, bt.getName() + " - " +
bt.getAddress());
            i++;
        }
    } else
    {
        Toast.makeText(getApplicationContext(), "No
Paired Bluetooth Devices Found.",

```

```

Toast.LENGTH_LONG).show();
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {

    boolean donus = true;

    String info = item.getTitle().toString();
    address = info.substring(info.length() - 17);
    name = info.substring(0, info.length() - 19);

    aramaButonu.setText("Bağlanıyor...");
    new Thread() {
        public void run()
        {

            boolean fail = false;

            BluetoothDevice device =
myBluetooth.getRemoteDevice(address);

            try {
                mBTSocket =
createBluetoothSocket(device);
            } catch (IOException e) {
                fail = true;
                Toast.makeText(getBaseContext(),
"Socket creation failed", Toast.LENGTH_SHORT).show();
            }

            try {
                mBTSocket.connect();
            } catch (IOException e) {
                try {
                    fail = true;
                    mBTSocket.close();
                }

                mHandler.obtainMessage(CONNECTING_STATUS, -1, -1)
                    .sendToTarget();
                } catch (IOException e2) {

                    Toast.makeText(getBaseContext(),
"Socket creation failed", Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
    if (fail == false) {

```

```

        mConnectedThread = new
ConnectedThread(mBTSocket);
        mConnectedThread.start();

mHandler.obtainMessage(CONNECTING_STATUS, 1, -1, name)
        .sendToTarget();
    }
}
}.start();

return donus;

}

private void msg(String s)
{
    Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_LONG).show();
}

private BluetoothSocket
createBluetoothSocket(BluetoothDevice device) throws
IOException {
    try {
        final Method m =
device.getClass().getMethod("createInsecureRfcommSocketToSe
rviceRecord", UUID.class);
        return (BluetoothSocket) m.invoke(device,
myUUID);
    } catch (Exception e) {
        Log.e(TAG, "Could not create Insecure RFComm
Connection", e);
    }
    return
device.createRfcommSocketToServiceRecord(myUUID);
}

private class ConnectedThread extends Thread {

    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;

    public ConnectedThread(BluetoothSocket socket) {
        mmSocket = socket;
        InputStream tmpIn = null;

```

```

OutputStream tmpOut = null;

try {
    tmpIn = socket.getInputStream();
    tmpOut = socket.getOutputStream();
} catch (IOException e) {
}

mmInStream = tmpIn;
mmOutputStream = tmpOut;
}

public void run() {
    byte[] buffer = new byte[1024];
    int bytes;
    while (true) {
        try {

            bytes = mmInStream.available();
            if (bytes != 0) {
                buffer = new byte[1024];
                SystemClock.sleep(100);
                bytes = mmInStream.available();
                bytes = mmInStream.read(buffer, 0,
bytes);

mHandler.obtainMessage(MESSAGE_READ, bytes, -1, buffer)
                    .sendToTarget();
            }
        } catch (IOException e) {
            e.printStackTrace();

            break;
        }
    }
}

public void write(String input) {
    byte[] bytes = input.getBytes();
    try {
        mmOutputStream.write(bytes);
    } catch (IOException e) {
    }
}

public void cancel() {
    try {
        mmSocket.close();
    }
}

```

```

        } catch (IOException e) {
        }
    }

}

public void veriGonder()
{
    if (mConnectedThread != null)
        mConnectedThread.write("V");
}

void ekleGrafik(float kalp)
{
    mChart.moveViewToX(sayac);
    sayac++;
    values.add(new Entry(sayac, kalp));

    mChart.notifyDataSetChanged();
    mChart.setVisibleXRange(0,100);

    LineDataSet set1 = new LineDataSet(values,
"Electrocardiogram (EKG Grafiği)");

    set1.setAxisDependency(YAxis.AxisDependency.LEFT);
    set1.setColor(ColorTemplate.getHoloBlue());

set1.setValueTextColor(ColorTemplate.getHoloBlue());
    set1.setLineWidth(2f);
    set1.setDrawCircles(true);
    set1.setDrawValues(true);
    set1.setFillAlpha(100);
    set1.setFillColor(Color.rgb(255, 64, 64));
    set1.setHighLightColor(Color.rgb(244, 117, 117));
    set1.setDrawCircleHole(true);
    set1.setDrawFilled(true);

    LineData data = new LineData(set1);
    data.setValueTextColor(Color.BLACK);
    data.setValueTextSize(10f);

    mChart.setData(data);
    mChart.invalidate();
}

```

```

    private void takeScreenshot() {
        Date now = new Date();
        android.text.format.DateFormat.format("yyyy-MM-
dd_hh:mm:ss", now);

        try {
            String mPath =
Environment.getExternalStorageDirectory().toString() +
"/PICTURES/Screenshots/" + now + ".jpg";

            View v1 =
getWindow().getDecorView().getRootView();
            v1.setDrawingCacheEnabled(true);
            Bitmap bitmap =
Bitmap.createBitmap(v1.getDrawingCache());
            v1.setDrawingCacheEnabled(false);
            File imageFile = new File(mPath);
            FileOutputStream outputStream = new
FileOutputStream(imageFile);
            int quality = 100;
            bitmap.compress(Bitmap.CompressFormat.JPEG,
quality, outputStream);
            outputStream.flush();
            outputStream.close();
            MediaScannerConnection.scanFile(this,
                new String[]{imageFile.toString()},
null,
                new
MediaScannerConnection.OnScanCompletedListener() {
                    public void onScanCompleted(String
path, Uri uri) {
                        Log.i("ExternalStorage",
"Scanned " + path + ":");
                        Log.i("ExternalStorage", "->
uri=" + uri);
                    }
                });
            msg("Kayıt Oluşturuldu");
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
} // MainActivity sonu

```

# ÖZGEÇMİŞ

## KİŞİSEL BİLGİLER

Adı Soyadı : Ebru Gündoğdu  
Doğum Tarihi ve Yeri : 1986 - Bursa  
Yabancı Dili : İngilizce  
E-posta : ebru\_alic@hotmail.com

## ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektronik Öğretmenliği	Kocaeli Üniversitesi	2009
Lise	Uçak Bakım Teknisyenliği ve Elektroniği	Bursa Hürriyet Anadolu Teknik Lisesi	2004

## YAYINLAR

1. E. Gündoğdu, K. Gündoğdu and A. Çalhan, “Kablosuz vücut alan ağları ile mobil sağlık izleme uygulaması”, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 4, pp. 640-645, 2016.
2. E. Gündoğdu, K. Gündoğdu and İ. Yücedağ, “Bulanık mantık ile akıllı fırının modellenmesi,” *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 4, no. 2, 2016.