



**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ**

BİTİRME ÇALIŞMASI

AKILLI BİLEKLİK TASARIMI

Bitirme Çalışması Danışmanı: Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN

17014044 Mertcan YILMAZ
17014077 İbrahim ÖZTÜRK
18014011 Murat DELİBALTA

İstanbul, 2022

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

AKILLI BİLEKLİK TASARIMI

Mertcan YILMAZ, İbrahim ÖZTÜRK ve Murat DELİBALTA tarafından hazırlanan bitirme çalışması 16.06.2022 tarihinde aşağıda isimleri belirtilen jüri üyeleri tarafından Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümünde **LİSANS BİTİRME ÇALIŞMASI** olarak kabul edilmiştir.

Jüri Üyeleri

Dr. Öğr Üyesi Zehra Gülrü Çam TAŞKIRAN

Yıldız Teknik Üniversitesi

Doç Dr. Ferkan YILMAZ

Yıldız Teknik Üniversitesi

Bu çalışma, TÜBİTAK BİDEB □2209A / □2209B tarafından desteklenmektedir. (Bu kısım sadece Tübitak tarafından destek almış projeler için doldurulacaktır.)

ÖNSÖZ

Bu proje içerisinde; akıllı bilekliklerin literatür özeti, gelişim süreçleri, akıllı bilekliklerin sahip olması gereken özellikler ve geleceğine değinilmiştir. Akıllı bileklik tasarımı bitirme çalışmamızın amacı ve neden tercih noktası olduğu belirtilmiştir. Günümüzde hızla gelişen ve yaygınlaşan akıllı bileklik konsepti ile ilgili bilgiler, araştırmalar ve yorumlar belirtilmiştir.

Tasarımda kullanılan sensörler, mikrodenetleyici ve diğer tüm modüllerle ilgili detaylı bilgiler bu raporun içerisinde yer verilmiştir. Ayrıca bu sistemi çalıştırmak için gereken kod ekler kısmında belirtilmiştir.

Yapılan bu tasarımda; grup üyelerinin çalışacağı konular ve tasarımda gerçekleştirilen adımlar eklenmiştir. Tüm grup üyeleri, kendi iş paketini tamamlamıştır ve bu sayede tasarım süreci eksiksiz olarak tamamlanmıştır.

Bu tasarım sürecinde, başından beri bizden yardımlarını esirgememiş olan tüm öğretim görevlisi hocalarımıza teşekkür ediyoruz. Proje sürecince bizlere her konuda destek olan danışman hocamız Sayın Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN'a ayrıca minnetlerimizi sunuyoruz.

Yıldız Teknik Üniversitesi ailesinden olmaktan gurur duyuyor ve bu gururumuzu; bu proje ile taçlandırıyoruz. Geçirmekte olduğumuz lisans hayatımızda bizlere çok şey öğreten bu kurumun her bir çalışanına saygılarımızı sunuyoruz.

Haziran, 2022

Mertcan YILMAZ, İbrahim ÖZTÜRK, Murat DELİBALTA

İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER	1
SİMGE LİSTESİ.....	4
KISALTMA LİSTESİ.....	5
ŞEKİL LİSTESİ.....	6
ÇİZELGE LİSTESİ	8
ÖZET	9
ABSTRACT.....	10
BÖLÜM 1	11
1. GİRİŞ	11
1.1 Literatür Özeti	11
1.2 Akıllı Bilekliğin Geleceği	14
1.3 Bitirme Çalışmamızın Amacı	16
BÖLÜM 2	17
2. AKILLI BİLEKLİK TASARIMI	17
2.1 Kullanılan Malzemeler	17
2.1.1 ESP32 Mikrodenetleyici	17
2.1.2 MPU6050 6 Eksenli Gyro ve Eğim Sensörü	18
2.1.3 MAX30100 Kalp Atış Hızı Sensörü ve Nabız Oksimetresi.....	20
2.1.4 SH1106 OLED GRAFİK EKRAN	21
2.1.5 3.7V 450 mAh Lityum Polimer Pil	22
2.1.6 TP4056 1A Lityum Polimer Şarj Devresi.....	23
2.1.7 TC1262 Voltaj Regülatörü	24
2.1.8 DS1042 4 Bacaklı Buton	25
2.1.9 CP2102 USB-UART Çevirici Modül	26
2.2 Tasarımın Şematik Yapısı	26
2.2.1 ESP32 Mikrodenetleyici Bağlantısı.....	27
2.2.1.1 I2C Haberleşme Protokolü	27
2.2.1.2 UART Haberleşme Protokolü	28
2.2.2 Ekran ve Sensörlerin Bağlantıları	29
2.2.3 CP2102 Programlayıcı Bağlantısı.....	29
2.2.4 Batarya ve Regülatör Bağlantıları	30
2.2.5 Buton Bağlantıları	31
2.2.6 Şematiğin Toplu Hali	31

2.3	PCB Tasarımı	32
2.3.1	PCB Tasarımının Ön Yüz 3 Boyutlu Görünümü	33
2.3.2	PCB Tasarımının Arka Yüz 3 Boyutlu Görünümü	34
2.3.3	PCB Tasarımının 2 Boyutlu Görünümü	35
2.4	Donanım İçin Gerekli Yazılımın yazılması	35
2.4.1	Tümleşik Geliştirme Ortamları	35
2.4.1.1	Arduino IDE	36
2.4.1.2	PlatformIO	37
2.4.2	Kullanılan Kütüphaneler	38
2.4.2.1	Wire Kütüphanesi	38
2.4.2.2	WiFi Kütüphanesi	38
2.4.2.3	HttpClient Kütüphanesi	38
2.4.2.4	U8g2 Kütüphanesi	39
2.4.2.5	MAX30100 Kütüphanesi	39
2.5	Kılıf Tasarımı	39
2.5.1	SolidWorks	40
2.5.2	SolidWorks ile Kılıf Tasarımı	41
BÖLÜM 3		42
3. AKILLI BİLEKLİK TASARIMININ GERÇEKLENMESİ		42
3.1	Test için ESP32-WROOM-32D Geliştirme Kartının Kullanılması	42
3.2	PCB'nin Lehimlenmesi	42
3.3	Kılıfın 3D Yazıcı ile Bastırılması	43
3.4	Akıllı Bileklik Tasarımının Son Hali	44
3.5	Ekran Çıktıları	44
3.5.1	Ana Ekran	44
3.5.2	Borsa Ekranları	45
3.5.2.1	BIST100 Endeksi	45
3.5.2.2	Kripto Paralar	46
3.5.2.3	Altın	46
3.5.2.4	Gümüş	46
3.5.2.5	Döviz	47
3.5.3	Kalp Atış Hızı ve Kandaki Oksijen Oranı Göstergesi Ekranı	47
3.5.4	Adımsayar Göstergesi Ekranı	47
3.5.5	Sıcaklık Göstergesi Ekranı	48
3.5.6	Alarm Ekranı	48
3.5.7	Ayarlar Ekranı	49
3.5.8	Hakkında Ekranı	49
BÖLÜM 4		50
4. PROJE TASARIMINDA GÖZ ÖNÜNE ALINAN GERÇEKÇİ KISITLAR VE KOŞULLAR		50
BÖLÜM 5		51
5. SONUÇ VE ÖNERİLER		51

KAYNAKLAR.....	53
EK-A.....	56
A- MALİYET TABLOSU.....	56
EK-B.....	57
B- KATALOG BİLGİLERİ.....	57
EK-C.....	59
KOD	59
TAKIM ÖZGEÇMİŞ BİLGİLERİ	94

SİMGE LİSTESİ

a	ivme
A	Amper
C	Kapasite
c	Deşarj Oranı
F	Kuvvet
m	kütle
mAh	miliAmper / saat
R	Direnç
S	Seri bağlı hücre
t	Zaman
V	Volt
v	Hız
μ	mikro
Ω	ohm
°C	Santigrat Derece

AHA	American Heart Association
BTĐ	Body Rate Dedector
CAD	Computer Aided Design
DC	Direct Current
DHCP	Dynamic Host Configuration Protocol
DMP	Digital Motion Processor
DNS	Domain Name System
EHR	Electronic Health Record
EN	Enable
GND	Ground
GSYİH	Gayri Safi Yurt İçi Hasıla
HR	Heart Rate
HRD	Heart Rate Dedector
Http	Hyper Text Transfer Protocol
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
IPMI	Akıllı Platform Yönetme Ara yüzü
IR	Infrared
IoT	Internet of Things
LiPo	Lityum Polimer
MEMS	Mikro Elektro Mekanik Sistemler
MİB	Merkezi işlem Birimi
NiCd	Nikel Kadmiyum
NiMH	Nikel Metal Hidrit
OLED	Organic Light Emitting Device
PCB	Baskılı Devre Kartı
PEOU	Perceived Ease of Use
PU	Perceived Usefulness
QoD	Quality of Data
QoE	Quality of Experience
QoL	Quality of Life
RISC	Reduced Instruction Set Computer
SCL	Serial Clock Line
SDA	Serial Data
TTL	Transistor-Transistor Logic
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
USB	Universal Serial Bus
VCC	Collector Collector Voltage
WEP	Wireless Equivalent Privacy
WPA	Wi-Fi Protected Access

	Sayfa
Şekil 1.1 Akıllı Bileklik Nabız Ölçümü	12
Şekil 1.2 Akıllı Bileklik Vücut Sıcaklığı Ölçümü	14
Şekil 2.1 ESP32-WROOM-32D Mikrodenetleyici	18
Şekil 2.2 MPU6050 6 Eksenli Gyro ve Eğim Sensörü.....	19
Şekil 2.3 MAX30100 Nabız ve Kalp Atış Hızı Sensörü.....	20
Şekil 2.4 MAX30100 Sistem Blog Diyagramı	21
Şekil 2.5 SH1106 OLED Grafik Ekran	22
Şekil 2.6 3.7 V 450 mAh Lityum Polimer Pil	23
Şekil 2.7 TP4056 1A Lityum Polimer Şarj Devresi	24
Şekil 2.8 TC1262 Voltaj Regülatörü.....	24
Şekil 2.9 DS1042 4 Bacaklı Buton.....	25
Şekil 2.10 CP2102 USB-UART Çevirici Modül.....	26
Şekil 2.11 Altium Programında Oluşturulmuş Şematik Kütüphaneler.....	26
Şekil 2.12 ESP32-WROOM-32D Mikrodenetleyici Bağlantıları	27
Şekil 2.13 I2C Haberleşme Protokolü.....	28
Şekil 2.14 UART Haberleşme Protokolü.....	28
Şekil 2.15 Ekran ve Sensörlerin Bağlantıları.....	29
Şekil 2.16 I2C Pull Up Direnç Bağlantısı	29
Şekil 2.17 CP2102 Programlayıcı Bağlantısı	30
Şekil 2.18 Batarya ve Regülatör Bağlantıları	30
Şekil 2.19 Buton Bağlantıları	31
Şekil 2.20 Şematiğin Bütün Hali	32
Şekil 2.21 Altium Programında Oluşturulmuş PCB Kütüphaneler	32
Şekil 2.22 PCB Tasarımının Ön Yüz 3 Boyutlu Görünümü	33
Şekil 2.23 PCB Tasarımının Arka Yüz 3 Boyutlu Görünümü	34
Şekil 2.24 PCB Tasarımının 2 Boyutlu Görünümü	35
Şekil 2.25 Arduino IDE.....	37
Şekil 2.26 PlatformIO	37
Şekil 2.27 SolidWorks ile Akıllı Bileklik Kılık Tasarımı.....	41
Şekil 3.1 ESP32-WROOM-32D Geliştirme Modülü	42
Şekil 3.2 PCB'nin Ön Yüzünün Lehimlenmiş Hali	43
Şekil 3.3 PCB'nin Arka Yüzünün Lehimlenmiş Hali.....	43
Şekil 3.4 3D Printer ile Bastırılmış Kılıfın Ön Yüzü	43
Şekil 3.5 3D Printer ile Bastırılmış Kılıfın Arka Yüzü	43
Şekil 3.6 Akıllı Bileklik Tasarımının Önden Görünümü.....	44
Şekil 3.7 Akıllı Bileklik Tasarımının Arkadan Görünümü	44
Şekil 3.8 Akıllı Bileklik Tasarımının Yandan Görünümü.....	44
Şekil 3.9 Ana Menüde Bulunan Borsa Ekranı.....	45
Şekil 3.10 BIST100 Endeksi.....	45
Şekil 3.11 Ethereum ve Bitcoin	46
Şekil 3.12 Gram ve ONS Altın	46
Şekil 3.13 Gram ve ONS Gümüş	46
Şekil 3.14 Dolar ve Euro	47

Şekil 3.15 Kalp Atış Hızı ve Kandaki Oksijen Oranı Göstergesi Oranı	47
Şekil 3.16 Adımsayar Göstergesi.....	47
Şekil 3.17 Sıcaklık Göstergesi	48
Şekil 3.18 Ana Menüde Bulunan Alarm Ekranı ve Alarm Ayarı.....	48
Şekil 3.19 Ana Menüde Bulunan Ayarlar Ekranı ve Saat, Tarih, Ekran Ayarı	49
Şekil 3.20 Ana Menüde Bulunan Hakkında Ekranı ve Hakkında Yazısı	49

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 5.1 Makine ve Teçhizat Giderleri	56
Çizelge 5.2 Sarf Giderleri.....	56

AKILLI BİLEKLİK TASARIMI

Mertcan YILMAZ, İbrahim ÖZTÜRK, Murat DELİBALTA

Elektronik ve Haberleşme Mühendisliği Bölümü

Bitirme Çalışması

Bitirme Çalışması Danışmanı: Dr. Öğr. Üyesi Zehra Gülrü ÇAM TAŞKIRAN

Günümüzde akıllı bileklikler sunduğu özellikler ve kolaylıklar ile günlük hayatımızın ayrılmaz bir parçası olma yönünde hızla ilerlemektedir. Birçok özellik ve hizmeti içerisinde bulundururken aynı zamanda insanları rahatsız etmeyecek, aksine günlük bir aksesuar olarak kullanabilecekleri şık ve rahat bir tasarıma sahiplerdir. Akıllı bileklikler temelde günlük aktivitelerimizi, alışkanlıklarımızı ve bazı sağlık verilerimizi takip etmekte kullanılmaktadır.

Genel olarak bu özelliklere baktığımızda birçok modelde birbirine yakın özelliklerler sunulduğu görülmektedir. Bunlar; genellikle saat ve tarih, alarm, adım sayar, nabız ölçer, uyku takibi vb. özelliklerdir. Akıllı bilekliklerden bir diğer beklentimiz ise batarya ömrünün gün içerisinde kullanıcıya yeterli gelmesi, güç tüketiminin minimum düzeyde olması ve uzun pil ömrüdür. Aynı zamanda akıllı bilekliklerin ergonomik bir tasarıma sahip olması beklenmektedir.

Günümüzde borsa gibi yatırım araçlarını, altın gibi değerli maden fiyatlarını ve döviz kurlarını takip etmek için telefondan veya bilgisayardan, kullandığınız banka, aracı kurum veya bu canlı verileri içeren mobil uygulamalara girmeniz ve kontrol etmeniz gerekmektedir. Bu verileri daha kolay ve hızlı bir halde kontrol edebilmenin yolu ise kolunuzdaki akıllı bileklikten bu verileri görüntüleyebilmektir. Bu uygulama sayesinde akıllı bileklik, bu verileri sürekli takip eden insanlar tarafından bir tercih noktası olacak ve onların, sürekli olarak telefonlarını kontrol etme ihtiyacı yerine akıllı bilekliklerini kontrol etmesini sağlayacaktır.

Projede ilk olarak baskılı devre kartı tasarımı yapılmıştır. Sonrasında ise tasarlanan kart gerekli firmaya yollanmış ve baskılı devre kartının fiziksel hali elimize ulaşmıştır. Baskılı devre kartı tasarımı sonrasında donanımı için yazılımın yazılmıştır. Firmadan gelen baskılı devre kartının lehimleme işlemleri yapıldıktan sonra bu yazılım baskılı devre kartının içine gömülmüştür. Tüm bu donanım ve yazılım için gerekli aşamalar sonrası akıllı bilekliğin dış tasarımı yapılmıştır.

Anahtar Kelimeler: Akıllı Bileklik, Baskılı devre kartı, tasarım, yazılım, uygulama, test

YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

SMART BRACELET DESIGN

Mertcan YILMAZ, İbrahim ÖZTÜRK, Murat DELİBALTA

Department of Electronics and Communication Engineering

Graduation Thesis

Supervisor: Dr. Öğr. Üyesi Zehra Gülrü ÇAM TAŞKIRAN

Nowadays, smart bracelets are rapidly moving towards becoming an integral part of our daily life with the features and conveniences they offer. While they include many features and services, they also have a stylish and comfortable design that will not disturb people but can be used as a daily accessory. Smart bracelets are basically used to track our daily activities, habits and some health data.

When we look at these features in general, it is seen that many models offer similar features. These; usually time and date, alarm, pedometer, pulse meter, sleep tracking and similar features. Another expectation from smart bracelets is that the battery life is sufficient for the user during the day, the power consumption is at a minimum level and the battery life is long. At the same time, smart bracelets are expected to have an ergonomic design.

In order to follow investment instruments such as the stock market, precious metal prices such as gold and exchange rates, you need to check your bank, brokerage firm or mobile applications that contain this live data by your phone or computer. The way to control these data more easily and quickly is to view these data from the smart bracelets that you are wearing. Thanks to this application, the smart bracelet will be a point of choice for people who constantly monitor this data, and it will allow them to check their smart bracelets instead of the need to constantly check their phones.

In the project, firstly, the printed circuit board was designed. Afterwards, the designed Printed Circuit Board (PCB) was sent to the related company. After the printed circuit board design, the software for the hardware was written. After soldering the printed circuit board from the company, this software was embedded in the printed circuit board. After these stages, the outer design of the smart bracelet was made.

Keywords: Smart bracelet, PCB, printed circuit board, design, application, coding, test

YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

GİRİŞ

Günümüzde akıllı bileklikler sunduğu özellikler ve kolaylıklar ile günlük hayatımızın ayrılmaz bir parçası olma yönünde hızla ilerlemektedir. Birçok özellik ve hizmeti içerisinde bulundururken aynı zamanda insanları rahatsız etmeyecek, aksine günlük bir aksesuar olarak kullanabilecekleri şık ve rahat bir tasarıma sahiplerdir. Akıllı bileklikler temelde günlük aktivitelerimizi, alışkanlıklarımızı ve bazı sağlık verilerimizi takip etmekte kullanılmaktadır.

Genel olarak bu özelliklere baktığımızda birçok modelde birbirine yakın özellikler sunulduğu görülmektedir. Bunlar; genellikle saat ve tarih, alarm, adım sayar, nabız ölçer vb. özelliklerdir. Akıllı bilekliklerden bir diğer beklentimiz ise batarya ömrünün gün içerisinde kullanıcıya yeterli gelmesi, güç tüketiminin minimum düzeyde olması ve uzun pil ömrüdür. Aynı zamanda akıllı bilekliklerin ergonomik bir tasarıma sahip olması beklenmektedir.

1.1 Literatür Özeti

Akıllı bileklik ile ilgili mevcut araştırmaların çoğu, laboratuvar ortamında veya serbest yaşam koşulları altında cihazların doğruluk özelliklerine odaklanmaktadır. Fakat doğruluğun yanı sıra kalite değerlendirmesi (QoE) de çok önemlidir. Geleneksel olarak kalite değerlendirmesi (QoE), çeşitli uygulamalar için son kullanıcı memnuniyet düzeylerini belirlemenin en yaygın ve popüler yollarından biri olmuştur. Nesnelerin interneti (IoT) ekosistemi bize farklı cihaz türleri, heterojen teknolojiler ve standartlar sunduğundan, kullanıcı memnuniyetini garanti etmek için kalite değerlendirmesi çok önemli hale gelmektedir.

QoE, Veri kalitesi (QoD) ve yaşam kalitesi (QoL) adında iki ana alt dalda incelenmektedir. QoD ile ilgili veriler, akıllı bilekliklerin üç farklı işlevinden elde edilir: Yürünen adım sayısı, kat edilen mesafe ve kalp atış hızı ölçümü. QoL ile ilgili veriler ise, üç farklı perspektiften kapsamlı bir kullanıcı deneyimi analizi yapılarak elde edilir; akıllı giyilebilir uygulamalar tarafından sağlanan algılanan kullanım kolaylığı (PEOU), algılanan kullanılabilirlik (PU) ve

bilgi zenginliđidir (RI). Algılanan kullanım kolaylıđı ve algılanan kullanışlılık tabirleri kullanıcı davranış ve benimseme niyeti anlamına gelmektedir [1]. Bilgi zenginliđi, bilgi ayrıntı düzeyini içeren akıllı giyilebilir uygulamalar tarafından verilerin sunulma şekli ve akıllı giyilebilir cihazların kendi başlarına yapamayacakları diđer ek işlevleri izleme yeteneđi ile ilişkilidir.

Kullanıcılar herhangi bir ürünü kullanmadan önce ürün hakkında algılar oluşturma eğilimindedir ve bu da beklentilere yol açar. Bir ürünü kullandıktan sonra, algılar beklentilerle eşleştiğinde veya beklentilerin üstüne çıktığında benimseme oranı yüksek olacaktır [2].

Akıllı bileklerin tercih edilmesinin genel sebeplerine; çođu insanın aşına olduđu bir cihaz olması, fiziksel aktivitenin ve fizyolojik önlemlerin neredeyse gerçek zamanlı olarak izlenmesini sağlaması, özel mesajlaşma ve hatırlatıcı uygulamaları desteklemesi; hastalar, aile üyeleri ve sağlık hizmeti sağlayıcıları arasındaki iletişimi sağlaması gibi özellikler söylenebilir.

Normal bir kalp atış hızı kişiye, yaşa, vücut büyüklüğüne, kalp rahatsızlıklarına, kişinin oturmasına veya hareket etmesine, ilaç kullanımına ve hatta hava sıcaklığına bađlı olabilmektedir. Duygular da kalp atış hızını etkileyebilmektedir. Örneğin, heyecanlanmak veya korkmak kalp atış hızını arttırabilmektedir. Dinlenme kalp atış hızı, sakince oturduğunuzda veya yattığınızdaki nabzınızdır. Amerika Hastaneler Birliđi (AHA) 'ne göre, sabah yataktan kalkmadan önce dinlenme kalp atış hızınızı ölçmek için daha iyi bir zaman olarak görölmektedir [3].



Şekil 1.1 Akıllı Bileklik Nabız Ölçümü

Stres, vücudumuzun kendisini zor durumlarla yüzleşmeye hazırlamak için verdiği bir tepkidir. Stres, günümüz toplumunun büyüyen sorunlarından biridir ve stres yaşayan insan sayısı her geçen gün artmaktadır. Bir kişi stres altına girdiğinde sinir sistemi stres hormonları salgılayarak tepki verir ve bu hormonlar vücudumuzu acil müdahalelere hazır hale getirir. Bu hormonların sürekli salgılanması insan sağlığı için tehlikeli hale gelir ve bir kişiyi ciddi bir zihinsel bozukluğa sokabilir.

Stresin uzun vadeli etkileri kronik olabilir. Stresin kronik etkisi, hipertansiyon [4], kardiyovasküler hastalıklar [5] ve hafıza sorunları gibi sağlık sorunlarına neden olur. Stresin sebep olduğu yalnızlık ve umutsuzluk duyguları insanları intihara yöneltebilir. İnsanların yüksek stres altında olup olmadıklarını fark etme olasılıkları daha düşük olabilir veya genellikle strese karşı daha az duyarlı olabilirler.

Kalp hızı değişkenliği ve elektrodermal aktivite literatürde stres tespiti için en sık kullanılan fizyolojik sinyallerdir. Stres seviyesini ölçmek için kalp atış hızı değişkenliğini kullanan IoT tabanlı giyilebilir sağlık cihazları ve fitness bantları mevcuttur. Araştırmacılar, fizyolojik sinyallerden bireylerin stresini tespit etmek için ticari akıllı bileklikleri kullanmaya başlamışlardır.

Akıllı bilekliklerde uygulanan stres algılama teknolojisi, aksi takdirde tespit edilemeyecek olan yüksek stres düzeyine ilişkin farkındalıklarını artırarak insanların stresi daha iyi anlamalarına ve rahatlatmalarına yardımcı olabilir. İlk üretilen akıllı bileklikler temel olarak kalp atış hızı değişkenliğini kullanarak kişinin stres altında olup olmadığını tahmin etmektedir. Ancak kalp atış hızı tek başına günlük aktivitelerdeki stres seviyesini doğru bir şekilde tahmin edemez. Stresin neden olduğu fizyolojik tepkiler, koşma, uykusuzluk gibi fiziksel aktiviteler tarafından da tetiklenebilir. Günümüzdeki gelişmiş akıllı bileklikler bu aktiviteleri de göz önüne alarak kişinin stres tahminini yapmaktadır [6], [7].

Sıcaklık sensörü, sıcaklık ölçümünü elektrik sinyali olarak veren bir cihazdır. Bu elektrik sinyali elektrik voltajı şeklindedir ve sıcaklık ölçümü ile orantılıdır. [8]'de belirtilen çalışmada insan sağlığı için giyilebilir izleme bandı ele alınmıştır. Bu giyilebilir izleme bandı bir vücut ısı dedektörü (BTD) ve bir iştirme hızı dedektöründen (HRD) oluşturulmuştur. BTD, mürekkep püskürtmeli baskılı esnek sıcaklık sensörü modülü kullanılarak gerçekleştirilmiştir. HRD için ise ticari bir kalp atış sensörü modülü kullanılmıştır. Üretilen BTD'nin duyarlılığı %99,82 doğruluk ile -31 °C olarak

bulunmuştur. Ticari kalp atış hızı sensörü modülünü kullanan HRD, ticari bir akıllı saatin verileri ile kıyaslandığında 0.85'lik standart sapma ile daha iyi bir performansa sahiptir.

Bir diğer makalede [9] ise insan vücut sıcaklığını ölçen, kullanıcı dostu ve uygun maliyetli bir kablosuz sağlık izleme sistemi önerilmiştir. Vücut sıcaklığını doğru ve hassas bir şekilde ölçmek için maksimum $\pm 0,3$ °C doğruluk seçilmiştir. Sıcaklık ölçüm ve izleme cihazını oluşturmak için yüksek performanslı ve düşük güç tüketimli RISC tabanlı olarak geliştirilmiş işlemci mimarisine (ARM) sahip bir mikrodenetleyici tarafından kontrol edilmiştir.



Şekil 1.2 Akıllı Bileklik Vücut Sıcaklığı Ölçümü

[10]'da belirtilen çalışmada ise bileğe takılabilir cihazlarda kullanılan sensörlerin ve ölçülen parametrelerin görselleştirilmesi için kullanılan yöntemlerin yanı sıra, çeşitli bileğe takılabilir cihazlardan elde edilen verilerin analizi için kullanılan yöntemlerin incelenmesini içermektedir. Bu derlemenin sonucunda, invaziv olmayan ve bileğe takılabilir cihazlarda kullanılan sensörlerin işlevleri ve yöntemlerinin sınıflandırılması bir araya getirilmiştir.

1.2 Akıllı Bilekliğin Geleceği

Akıllı bilekliklerden elde edilen bilgiler, bağlamlar arasında kişisel sağlık yörüngelerinin bütünsel görüşlerini sağlamak için Lab of Things [11],[12] veya Piloteur [13] gibi yeni ortaya çıkan nesnelerin interneti platformları ve elektronik sağlık kaydı (EHR) verileriyle entegre olma yeteneğine sahiptir.

Günümüzde sağlık bakım uzmanları, hastaların; aritmiler, aktivite seviyeleri veya kan basıncı değişiklikleri gibi kontrollerini yapmak için yüz yüze görüşme yapmaları gerekmektedir. Bu kontroller, kişilerin günlük yaşantısının aksamasına yol açmaktadır.

Akıllı bileklik teknolojisinin gelişmesi ve sağlık sistemine entegre olmasıyla beraber bu gibi kişisel periyodik kontroller, günlük yaşantıya engel olmadan yapılabilecek hale gelecektir. Bu öngörülerle birlikte akıllı bileklikler, gelişmekte olan bir teknolojidir ve bu cihazlarda sağlıkla ilgili amaçlara yönelik araştırmalar ilerleyen yıllarda artarak gelişecek ve hayatımızın bir parçası olacaktır [14].

Refah seviyesinin en önemli belirleyicilerinden biri olan iktisadi büyüme, tüm ülkeler açısından güncelliğini yitirmeyecek derecede önemli bir konudur. Günümüzün gelişmiş ekonomilerinde iktisadi büyümenin en önemli faktörleri arasında teknolojik gelişme ve inovasyon başta gelmektedir.

Londra merkezli PricewaterhouseCoopers şirketi tarafından yapılan bir ankete göre yenilikçi bir IoT ürününün başlıca örneği olan akıllı bilekliklerin insanların hayatını olumlu yönde değiştirmesi gözlenmiştir ve ilerleyen dönemlerde bu değişimin artarak devam etmesi beklenmektedir [15].

Marka değeri ülkemize ait olan gelişmiş ve ucuza mal edilmiş bir akıllı bileklik tasarımının ülkemize sosyoekonomik açıdan yararı yadsınamaz bir gerçektir. Son dönemlerde yapılan çalışmalar neticesinde teknolojik gelişme ve inovasyonu ortaya çıkartan Ar-Ge faaliyetlerinin, firma ve ülkelerin öncelikle rekabet güçlerini korumasının ardından rekabet avantajı sağlayabilmek için maliyetleri düşürmenin yanı sıra üretimde kalite artışıyla birlikte ekonomik ve toplumsal faydaya neden olduğu vurgulanmaktadır [16].

Giyilebilir teknolojinin her geçen yıl önemli ölçüde büyüyeceği tahmin edilmektedir. Genel olarak giyilebilir teknoloji pazarı 2025 yılında yaklaşık 78 milyar dolara ve tahmin edilen satış rakamlarının ise yaklaşık 2,5 milyara ulaşması beklenmektedir. İnsanlardaki memnuniyet ve sürekli kullanım niyeti, tüketicilerin psikolojik özellikleri, yenilikçi olmaları, kullanım motivasyonu, moda liderliği ve sağlık kaygısı, ürünlerdeki kullanışlılık, kullanım kolaylığı ve estetik, insanlarda akıllı bilekliğe olan rağbeti arttırmaktadır. Akıllı bilekliklerin de bu pazarda olduğunu düşünürsek akıllı bilekliklerin önümüzdeki yıllarda artarak kullanımının devam edeceği öngörülmektedir.

Çoğu akıllı bileklik şirketi, Çin ve ABD’de konumlanmaktadır. Çin, imalat merkezidir ve yaratıcı fikirler burada hayata geçmektedir. Akıllı bileklik üretiminin gelişmiş ülkeler ve sayılı firmalar tarafından tekelleştirildiği bu dönemde ülkemizden çıkacak özgün ve

gelişmiş bir akıllı bileklik, ülkemizin ihracatını artıracığı gibi Gayri Safi Yurt İçi Hasıla (GSYİH) miktarını da artırarak ülke ekonomisine katkı sağlaması muhakkaktır.

1.3 Bitirme Çalışmamızın Amacı

Projemizin amacı günlük hayatımızın bir parçası haline gelmiş akıllı bileklikleri, insan hayatını daha fazla kolaylaştıracak ve günlük yaşantılarında insanlara yardımcı olacak şekilde geliştirmek ve insanların onlar için önemli olan başlıca sağlık ve günlük yaşam bilgilerine kolay ve rahat bir şekilde ulaşabilmesini sağlamaktır.

- Akıllı bilekliğimizin saat, tarih ve alarm gibi temel bilgileri insanlara sunması dışında adım sayar, nabız ölçümü, sıcaklık ölçümü gibi insanların düzenli olarak takip etmesi gereken günlük sağlık verilerini kişilere sunabilmesi,
- Bu özelliklere ek olarak günlük hayatta sık sık kontrol edilen kur, değerli maden gibi verilerin takibinde kolaylık sağlaması ve kişiye anlık olarak bilgilendirme yapabilmesi,
- Ergonomik ve kullanımı rahat olması, kişilerin kullanım şekillerine göre ayar yapılabilmesi,
- Günlük hayatta kullanıcıların beğeneceği şık bir tasarıma sahip olması,
- Güç tüketiminin minimum düzeyde tutulması ve pil ömrünün uzun olması,
- Bu özelliklere ek olarak kullanıcı dostu olması ve her gruptan insanın rahatça kullanabilmesi öncelikli amaçlarımızdandır.

AKILLI BİLEKLİK TASARIMI

2.1 Kullanılan Malzemeler

2.1.1 ESP32 Mikrodenetleyici

Bir mikrodenetleyici, bir mikroişlemcinin, merkezi işlem birimi (MİB), hafıza ve giriş-çıkışlar, kristal osilatör, zamanlayıcılar, seri ve analog giriş çıkışlar, programlanabilir hafıza gibi bileşenlerle tek bir tümleşik devre üzerinde üretilmiş halidir.

Kısıtlı miktarda olmakla birlikte, yeterince hafıza birimlerine ve giriş – çıkış uçlarına sahip olmaları sayesinde tek başlarına çalışabildikleri gibi, donanımı oluşturan diğer elektronik devrelerle irtibat kurabilir, uygulamanın gerektirdiği fonksiyonları gerçekleştirebilirler. Üzerlerinde analog-dijital çevirici gibi tümleşik devreler barındırmaları sayesinde algılayıcılardan her türlü verinin toplanması ve işlenmesinde kullanılabilirler. Ufak ve düşük maliyetli olmaları gömülü uygulamalarda tercih edilmelerini sağlamaktadır.

Ayrıca mikrodenetleyiciler sıradan mikroişlemcilere nazaran aşağıda listelenen 4 temel avantajları sayesinde elektronik sanayinde günümüzde oldukça büyük bir uygulama alanına sahiptirler:

- Oldukça küçük boyutludurlar,
- Çok düşük güç tüketimine sahiptirler,
- Düşük maliyetlidirler,
- Yüksek performansa sahiptirler.

Sisteme enerji verildiğinde, bağlantı halinde olduğu sensörlerden veri alan, sensörlere veri gönderen, gerektiğinde onları enerjilendirilebilip gerektiğinde ise enerjisiz kalmaları için sinyal üreten mikrodenetleyiciler, programlanabilir mantık denetleyicileri sınıfında çok önemli yer tutup günümüzde çamaşır makinesinden bulaşık makinesine, cep telefonundan robotlara kadar pek çok alanda kullanılmaktadır [17].



Şekil 2.1 ESP32-WROOM-32D Mikrodenetleyici

Projemizde Espressif firmasına ait ESP-32 WROOM-32D mikrodenetleyici kullanılmıştır. Bu mikrodenetleyici seçimindeki en büyük etken üzerinde dahili olarak Wi-Fi ve Bluetooth özelliklerini de bulundurmasıdır. Bu sebeple farklı bir işlemci tercih edip ekstra Wi-Fi veya Bluetooth modülü almak yerine üzerinde her ikisini de dahili olarak bulunduran bir mikrodenetleyici tercihi yönelinmiştir. Sahip olduğu zengin çevre birimi ve diğer teknik özellikleriyle fiyat – performans anlamında bu proje için fazlasıyla yeterli görülmüştür.

2.1.2 MPU6050 6 Eksenli Gyro ve Eğim Sensörü

Akıllı bilekliğin adım ölçme verisine sahip olacağı amaçlarımızda belirtilmişti. Akım sayabilmek için ivmeölçer ve jiroskop verilerini elde etmemiz gerekir. İvmeölçer ve jiroskop terimlerini açıklayacak olursak:

İvmeölçer, hızlanmayı yani zaman birimi başına hızdaki değişimi ölçer.

(1) de verilen formül fizikte hızın zamanla değişiminin ivmenin tanımı olduğunu göstermektedir.

$$a = \frac{dv}{dt} \quad (1)$$

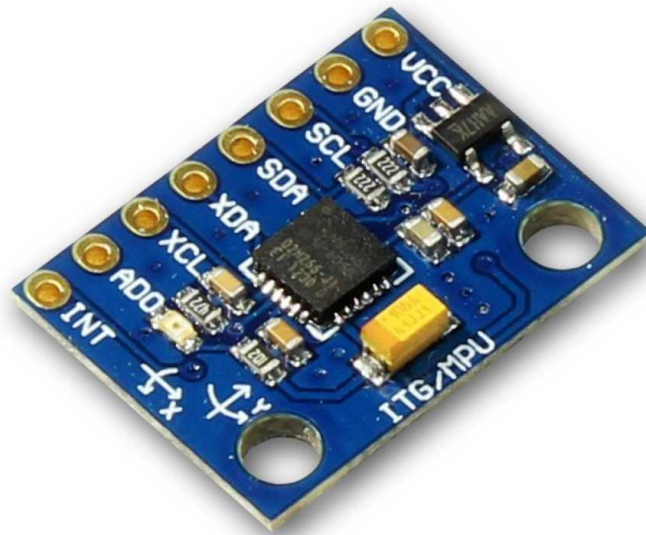
Newton'un İkinci Yasasında belirtilen (2) formülü ivmeölçerlerin çalışmak için kullandıkları formüldür, yani nesnenin kuvvet ve kütle parametrelerini kullanırlar.

$$a = \frac{F}{m} \quad (2)$$

Bunun elektronikte uygulanabilmesi için, mekanik parçalar bir Mikro Elektro Mekanik Sistemler (MEMS)'de oluşturulduğundan, geleneksel elektronik çip üretim tekniklerinden farklı olan MEMS teknikleri kullanılır. Bu durumda, ivmeleri ölçebilen izler veya elemanlar oluşturulur. Eğer ivme zamana göre entegre edilmişse, hız ve yer değiştirme gibi birçok başka verinin elde edilebileceği anlamına gelir. Yani, bir nesnenin konumunu bilmek veya hareketini tespit etmek için bu parametreler kullanılabilir.

Jiroskop, bir nesnenin açısal hızını, yani zaman birimi başına açısal yer değiştirmeyi veya bir cismin kendi eksenini etrafında ne kadar hızlı döndüğünü ölçen bir cihazdır. MEMS teknikleri de coriolis olarak bilinen bir etkiyi kullanarak bu hızı ölçmek için kullanılır. Bu sayede açısal hız ölçülebilir veya açısal hız zamana göre entegre edilerek açısal yer değiştirme elde edilebilir.

Jiroskop ve ivmeölçerler tek başlarına yeterince güvenli bilgi vermezler. Bu yüzden genellikle bu iki sensör birleştirilerek yönelim, hız, pozisyon gibi bilgiler tek bir üniteden alınmaktadır. Bu üniteye Atalet ölçü birimi (IMU) denilir.



Şekil 2.2 MPU6050 6 Eksenli Gyro ve Eğim Sensörü

Bu bilgiler doğrultusunda projemizi gerçekleştirmek için piyasada kullanılan ivme sensörleri incelenip kıyaslanmıştır. Bu karşılaştırmalar sonucunda MPU6050 sıcaklık da ölçebildiğinden ötürü sıcaklık için ekstra sensör gerekmeyp, akıllı bileklik için uygun ivmeölçer sensörü olarak seçilmiştir.

2.1.3 MAX30100 Kalp Atış Hızı Sensörü ve Nabız Oksimetresi

MAX30100, entegre bir nabız oksimetresi ve kalp atış hızı izleme sensörüdür. Nabız oksimetresi ve kalp atış hızı sinyallerini algılamak için iki LED'e, bir fotodedektöre, optimize edilmiş optiklere ve düşük gürültülü analog sinyal işlemeye sahiptir.

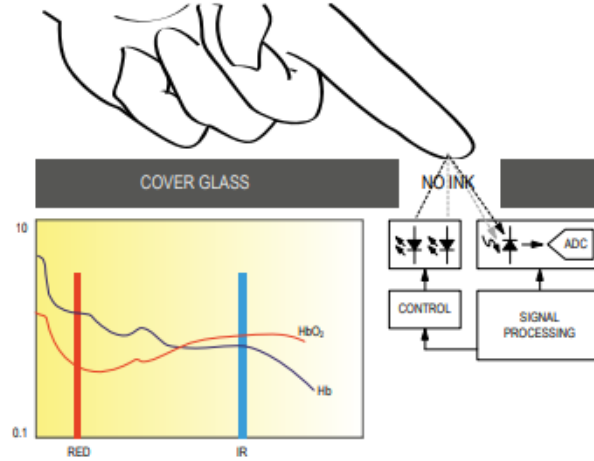
MAX30100 entegresine baktığımızda tasarımı giyilebilir sağlık sektörü için basit ve kolay bir çözüm sunmaktadır. Ultra düşük güçlerde çalışabildiğinden dolayı giyilebilir cihazların pil ömrünü arttırmaktadır. Gelişmiş işlevsel yapısı, ölçüm performansını iyileştirmektedir. En önemli özelliklerinden birisi de hızlı veri çıkış yeteneğinin olmasıdır.



Şekil 2.3 MAX30100 Nabız ve Kalp Atış Hızı Sensörü

Oksijen olarak zengin kan, daha az oksijen miktarına sahip kandan daha fazla kırmızı ışığı absorbe etmektedir ve içinden kırmızı ışığı geçirmektedir. Bu prensibi kullanan Pulse Oksimetreler, kırmızı ışık ve kızıl ışıkların kandan ne kadar geçtiğini bilmek için bir kırmızı ışık ve kızıl ışık LED'leri ile donatılmıştır. Kalp kan pompalandığında vücuttaki oksijenli kan artmakta ve kalp gevşediğinde oksijenli kan azalmaktadır. Sensörler yardımıyla kandaki oksijen hacmindeki artış ve azalma arasındaki zaman farkı hesaplanabilmektedir, buradan da nabız ölçümü yapılmaktadır.

MAX30100, SpO2 alt sisteminin sıcaklığa karşı kalibre edebilmek için çip üzerinde bir sıcaklık sensörüne sahiptir. SpO2 algoritması, IR LED'in dalga boyuna nispeten duyarsızdır, ancak kırmızı LED'in dalga boyu, verilerin doğru yorumlanması için kritik öneme sahiptir. Sıcaklık sensörü verileri, ortam sıcaklığı değişiklikleriyle SpO2 hatasını telafi etmek için kullanılmaktadır.



Şekil 2.4 MAX30100 Sistem Blok Diyagramı

Şekil 2.4'te sensör aktif olduğunda hem IR hem de Kırmızı LED'ler yanmaktadır. Örneğin, parmağımızı sensöre yerleştirdiğinizde, vücuttaki oksijen seviyelerine göre ışınlar parmaktan geçip başka bir fotosensör tarafından emildikten sonra kalan ışınları toplamakta ve daha sonra dijital sinyal üretmektedir. Devamında ise sinyaller ve I2C veri yolu arayüzü aracılığıyla mikrodenetleyiciye iletilmektedir [18].

2.1.4 SH1106 OLED GRAFİK EKLAN

SH1106 1,3 inç genişliğinde 128x64 çözünürlüğe sahip OLED ekran modülüdür. I2C seri arayüz hattı ile kontrol edilmektedir. Tercih noktası olmasının en büyük sebeplerinden biri ekranın akıllı bilekliğimize uygun boyutlarda olmasıdır. 0.96 inch bir ekran bu tasarım için küçük kalmaktayken, 1.3 inçten de daha büyük bir ekran akıllı bileklik tasarımı boyutları çerçevesinde büyük kalmaktadır. Her ne kadar LCD, dokunmatik vb. diğer farklı özelliklerdeki ekranlar incelense de maliyeti ciddi derecede arttıracığından ötürü SH1106 OLED grafik ekran en uygun çözüm olarak karşımıza çıkmıştır.



Şekil 2.5 SH1106 OLED Grafik Ekran

2.1.5 3.7V 450 mAh Lityum Polimer Pil

Yapısında Lityum ve Polimer kimyasallarını barındıran pillere kısaca LiPo pilleri denir. LiPo piller NiCd ve NiMH pillere göre çok daha fazla akım üretebilirler. Kıyaslama yapıldığında diğer pillere oranla daha hafiftirler ve kullanım süreleri daha uzundur. Ayrıca istenilen şekilde ve ölçüde üretilebilirler. Dezavantaj olarak ise kullanılmaları, şarj edilmeleri ekstra dikkat gerektirir. Ayrıca özel şarj cihazı gerektirirler.

LiPo piller hücrelerden oluşur. Her bir hücrenin nominal voltajı 3.7 voltur. Seri bağlı hücreler S harfi ile gösterilir. Bu durumda S harfinden önceki rakam hücre sayısını gösterir. Yani 2S denildiğinde $3.7 \times 2 = 7.4$ Volt anlaşılır. Buna ilave olarak LiPo piller için 3S2P şeklinde de tanımlamalar yapılmıştır. Bunun anlamı ise 3 adet seri bağlı pilden 2 adeti paralel bağlanmıştır. Bu durumda voltaj değişmez ancak pilin akım değeri ikiye katlanır.

Pilin bir hücresinin boş haldeki voltajı 3V, tam şarjlı halde voltajı 4.2V olmalıdır. Kullandığınız pilin herhangi bir hücresinin 3V altına düşmesi durumunda pilin ömrünü tamamlama riski vardır. Aşırı şarjın yani 4.20V dan fazla şarj etmenin de güvenlik açısından riskleri vardır. Pil deforme olabilir yani şişebilir veya alev almaya başlayabilir.

Kapasite, bataryanın ne kadar enerji depolayabileceğinin göstergesidir. MiliAmper/ saat cinsinden hesaplanır ve mAh olarak ifade edilir. Yani mAh, batarya boşalınca kadar bir saat boyunca pilden ne kadar akım çekebileceğimizin göstergesidir.



Şekil 2.6 3.7 V 450 mAh Lityum Polimer Pil

Bir diğer husus ise bataryanın C değeri yani deşarj oranıdır. Bazı bataryaların üzerinde iki farklı deşarj değeri bulunabilmektedir. Bunlardan ilki sürekli deşarj oranını, diğeri ise patlama (peak) deşarj oranını ifade etmektedir. Deşarj hızı bataryanın ne kadar hızlı boşalabildiğini gösterir. Ne kadar hızlı boşalırsa o kadar fazla akım aktarır.

Bu projede kullanılan Lityum Polimer pil Şekil 2.6'da verilmiştir. 3.7V 450 mAh olan bu pil, akıllı bileklik tasarımımız için kullanabileceğimiz maksimum boyutta olup, muadillerine göre dikdörtgen yerine kare biçiminde olmasından, yani kompakt yapısından ötürü tercih noktası olmuştur. Ayrıca 450 mAh kapasite değeri akıllı bileklik projemizi gerçeklemek için fazlasıyla yeterlidir.

2.1.6 TP4056 1A Lityum Polimer Şarj Devresi

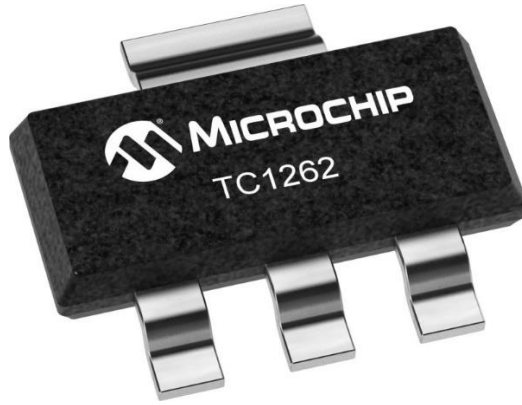
1A LiPo pil şarj devresi, üzerinde bulunan TP4056 lineer şarj entegresi ile 3.6V – 4.2V arası çalışan 1S pilleri şarj etmek için tasarlanmıştır. Kart hem lityum polimer hem de lityum iyon pilleri şarj etmek için kullanılabilir. Çıkış hassasiyeti %1,5 olan bu şarj devresi, Micro USB girişi aracılığıyla veya üzerinde bulunan IN+, IN- giriş pinleri ile 4.5V-5.5V aralığındaki gerilimlerle beslenebilmektedir [19].



Şekil 2.7 TP4056 1A Lityum Polimer Şarj Devresi

2.1.7 TC1262 Voltaj Regülatörü

Voltaj regülatörü; bir güç kaynağına bağlı olarak çalışan ya da şarj olan makine ve cihazların bağlı oldukları süre boyunca doğru voltaj seviyesine maruz kalmalarını sağlayan bir sistemdir. Lineer yani ileri beslemeli ya da negatif geri beslemeli olarak çalışan regülatör, olası bir voltaj değişikliği durumunda tampon görevi görerek güç kaynağına bağlı makine ya da cihaz donanımlarının zarar görmesini önler.



Şekil 2.8 TC1262 Voltaj Regülatörü

Standart bir regülatör üç temel bileşene sahiptir; geri besleme devresi, kararlı referans voltajı ve geçiş elemanı kontrol devresi. Geri besleme devresi, DC voltaj çıkışındaki değişiklikleri algılar, Kararlı referans voltajı bir sinyal üretir ve geçiş elemanı kontrol devresi sinyalle öğrendiği değişiklikleri düzenleyerek doğru voltajı ayarlar. Yani regülatör bileşenleri anlık olarak voltajı kontrol eder, değişiklikleri gözlemler, değişikliği fark eder ve sonunda ortaya çıkan bu değişikliğin cihaza ya da makineye zarar vermesini önlemek için fazlalıkları emer ve doğru seviyeye getirir.

Akıllı bileklik tasarlarken kullandığımız bazı komponentlerin çalışma gerilimi 3.3V'tur. Bataryamız 3.7V olduğundan ötürü bataryadan gelen gücü 3.3 voltajında sabit tutacak bir regülatör ihtiyacı duyulmuştur. Piyasa araştırıldığında bu ihtiyacı karşılayacak regülatörün TC1262 olduğuna karar verilmiştir.

2.1.8 DS1042 4 Bacaklı Buton

Buton, iterek üzerine basıldığında, makine veya yazılımlardaki bir sürecin başlamasını ve kontrolünü sağlayan basit bir geçiş mekanizmasıdır. Butonlar tipik olarak genellikle sert plastik veya metal malzemeden imal edilir. Yüzeyi insan eline uygun şekilde tasarlanmış olup, genellikle basılacak bölümü düz bir yapıya sahiptir. Butonların pek çok çeşidi olsa da basitçe itme ve uygulanan bu kuvvet karşısında tepki veren yay sisteminden oluşur. Butona uygulanan her kuvvet önceden belirlenmiş bir sürecin çalışmasını sağlar. Butonlar, hesap makinesi, telefon, mutfak aletleri, diğer çeşitli mekanik ve elektronik aletlerde ticari olarak kullanılmaktadır [20].



Şekil 2.9 DS1042 4 Bacaklı Buton

Bu projede DS1042 4 bacaklı buton kullanılmıştır. 4 bacaklı butonlar 2 bacaklı butonlara göre baskılı devre kartına (PCB) daha sağlam bağlandığından ötürü tercih edilmiştir. Bu tipte olan butonların karşılıklı pinleri birbirleriyle kısa devre olmaktadır.

Projede buton kullanılmasının sebebi ise ekranda arayüz işlemlerini sağlamaktır. Ekranımız dokunmatik olmadığından ötürü işlemleri gerçekleştirmek için 4 adet buton kullanılmıştır. Bu butonlardan 2 tanesi sağa ve sola geçişler için, diğer 2 buton ise bulunduğu menünün içerisine giriş ve çıkışları kontrol etmek için tasarlanmıştır.

2.1.9 CP2102 USB-UART Çevirici Modül

Bu modül, üzerinde bulunan CP2102 entegresi ile USB üzerinden UART bağlantısı kurmayı sağlamaktadır. Sistemde kullandığımız mikrodenetleyici, modül halinde olduğundan ötürü geliştirme kartlarında olduğu gibi direkt olarak programlanamamaktadır.



Şekil 2.10 CP2102 USB-UART Çevirici Modül

Akıllı Bilekliğin şematik devresi, PCB basımı öncesinde kontrol etmek amacıyla ESP32-WROOM-32D geliştirme modülü ile BÖLÜM 3'te gerçekleştirilmiştir. PCB kart bastırıldıktan ve elimize ulaştıktan sonra ise sistemi test etmemiz gerekmektedir. Bunun için lehimleme sonrasında mikrodenetleyici'yi kodlayabilmek için CP2102 USB-UART çevirici modül gerekmektedir. Daha detaylı bilgi BÖLÜM 3'te aktarılacaktır.

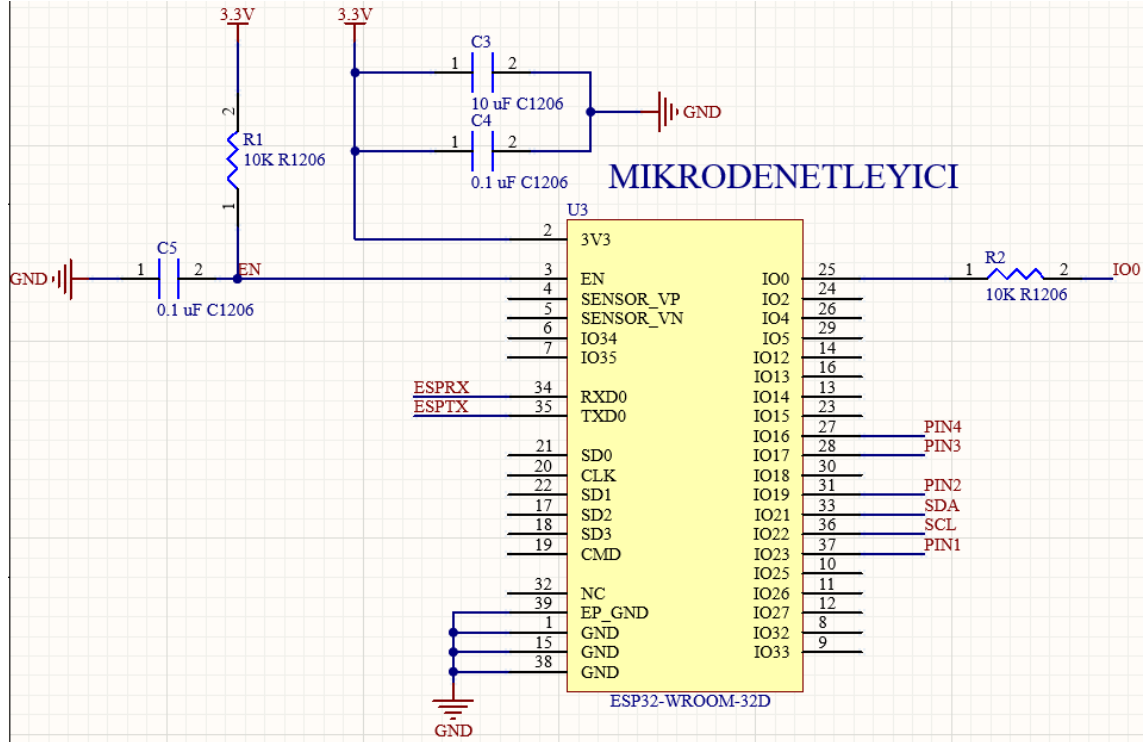
2.2 Tasarımın Şematik Yapısı

Projede gereken şematik tasarım için Altium programı kullanılmıştır. Öncelikle devrede ihtiyaç duyulan tüm elemanlar için Şekil 2.11'de görüldüğü gibi ayrı ayrı şematik kütüphaneleri oluşturulmuştur.

Design Item ID	Description
TP4056	BATTERY CHARGER
TC1262-3.3V Regül	TC1262-3.3V
MPU6050	MPU6050 gyro
MMBT2222A	Bipolar_BJT_Transistor N
MAX30100	KALP VE NABIZ SENSORU
Header 2.54 1x7	PROGRAMLAYICI
Header 2.54 1x4	EKRAN
ESP32	ESP32
Buton4	4 buton
Buton3	4 buton
Buton2	4 buton
Buton1	4 buton
10 uF C1206	10 uF C1206
10K R1206	10K R1206
2.2K R1206	2.2K R1206
1 uF C1206	1 uF C1206
0.1 uF C1206	0.1 uF C1206

Şekil 2.11 Altium Programında Oluşturulmuş Şematik Kütüphaneler

2.2.1 ESP32 Mikrodnetleyici Baęlantısı



Şekil 2.12 ESP32-WROOM-32D Mikrodnetleyici Baęlantıları

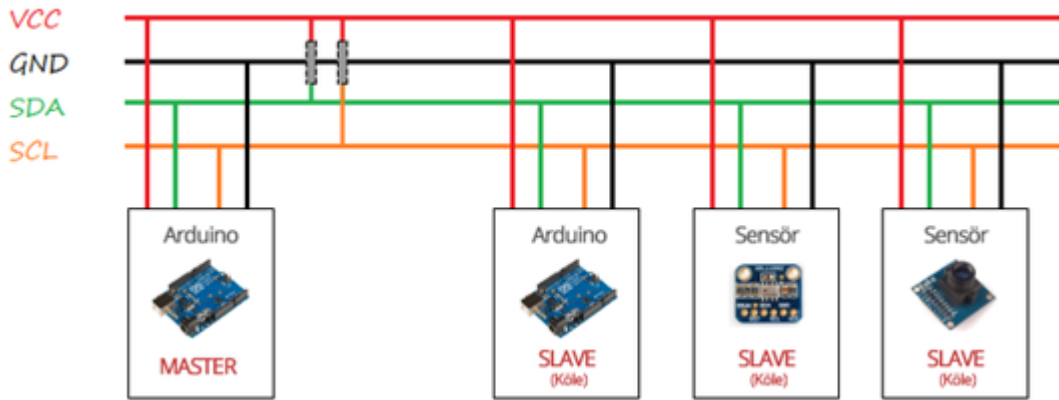
Şekil 2.12’de ESP32-WROOM-32D mikrodnetleyici baęlantıları verilmiştir. [21]’de bulunan ESP32-WROOM-32D datasheet incelenerek EN ve 3.3V GND ve IO0 baęlantıları şekildeki gibi yapılmıştır. Bunlara ek olarak PIN1, PIN2, PIN3 ve PIN4’ler buton baęlantılarını temsil etmektedir. Devrede kullanılan sensörler ve ekran I2C protokolü ile haberleştikleri için SDA ve SCL pin baęlantıları gerçekleştirilmiştir. Mikrodnetleyiciye yapılan son baęlantı ise UART protokolü ile ESPTX ve ESPRX baęlantılarıdır. Bu baęlantılar ise CP2102 tarafından mikrodnetleyiciyi programlamak için gereklidir.

2.2.1.1 I2C Haberleşme Protokolü

I2C (Inter-Integrated Circuit) haberleşme protokolü, 50’den fazla şirket tarafından üretilen 1000’in üzerinde tümleşik devre için uygulanan, bir dünya standardıdır [22]. Buna ek olarak, Sistem Yönetim Veri yolu (SMBus), Güç Yönetim Veri yolu (PMBus), Akıllı Platform Yönetme Ara yüzü (IPMI) gibi değişik kontrol mimarileri için kullanılmaktadır.

I2C protokolü, biri clock diğeri de veri olmak üzere 2 adet iletişim kanalına sahiptir. Bu kanallar, SCL (Serial Clock) ve SDA (Serial Data) olarak isimlendirilmektedir. SDA veri

iletişimi için kullanılmakta, SCL ile ise gönderen ve alan taraflar veri senkronizasyonunu sağlamaktadır.

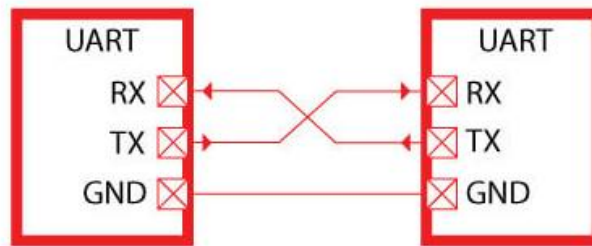


Şekil 2.13 I2C Haberleşme Protokolü

Şekil 2.13'te görülebileceği gibi I2C haberleşmesinde, haberleşmeyi kontrol eden master cihazı bulunur. Her haberleşmede bir tane master bulunmalıdır. Haberleşmenin sağlanabilmesi için haberleşme hattına en az bir adet slave (köle) cihaz bağlanmalıdır. Hatta bağlanan birden fazla slave cihazlardan hangisinin veri aktaracağına, master cihaz karar verir. Böylece hat sayısında bir değişiklik olmadan birden fazla cihazla haberleşme sağlanır. Master ve slave cihazların aynı besleme hattına bağlanmasına gerek yoktur. Fakat iletişimin sağlanması için toprak hatlarının aynı olması gerekir.

2.2.1.2 UART Haberleşme Protokolü

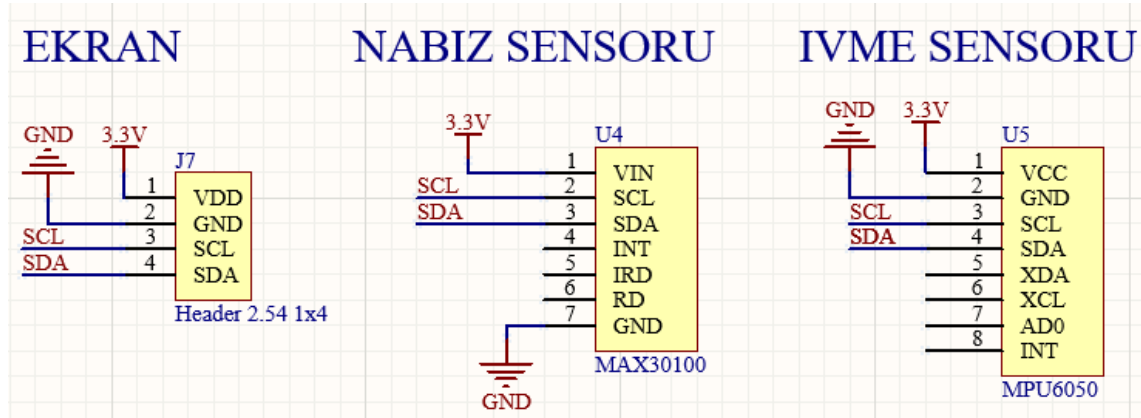
UART (Universal Asynchronous Receiver Transmitter), bilgisayar ve mikrodenetleyici veya mikrodenetleyici ve çevre birimler arasında haberleşmeyi sağlayan haberleşme protokolüdür. Asenkron olarak çalıştığı için herhangi bir "clock" ihtiyacı duymaz.



Şekil 2.14 UART Haberleşme Protokolü

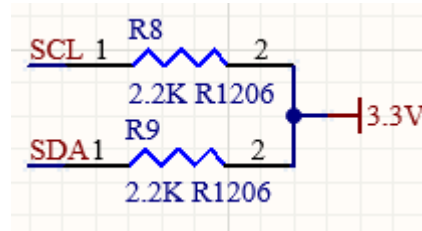
UART haberleşmesi için mikrodenetleyicide daha önceden tanımlanmış olan pinleri kullanılabilir. Bunun için USB-TTL dönüştürücü veya RS232 modülüne ihtiyaç vardır. Her iki modülden gelen RX-TX pinleri, mikrodenetleyicinin RX-TX pinleri ile çapraz olarak bağlantısı yapılır ve haberleşme sağlanır.

2.2.2 Ekran ve Sensörlerin Bağlantıları



Şekil 2.15 Ekran ve Sensörlerin Bağlantıları

Projede kullanılan Ekran, nabız sensörü ve ivme sensörünün bağlantıları birbirleri ile aynıdır. Şekil 2.15'te görüldüğü gibi 3.3V olacak şekilde güç bağlantısı gerçekleştirilmiştir. Ayrıca her birine ortak SCL ve SDA bağlanmış olup, bu pinler 2.2k Ω pull up direnciyle Şekil 2.16'daki gibi bağlanmıştır.

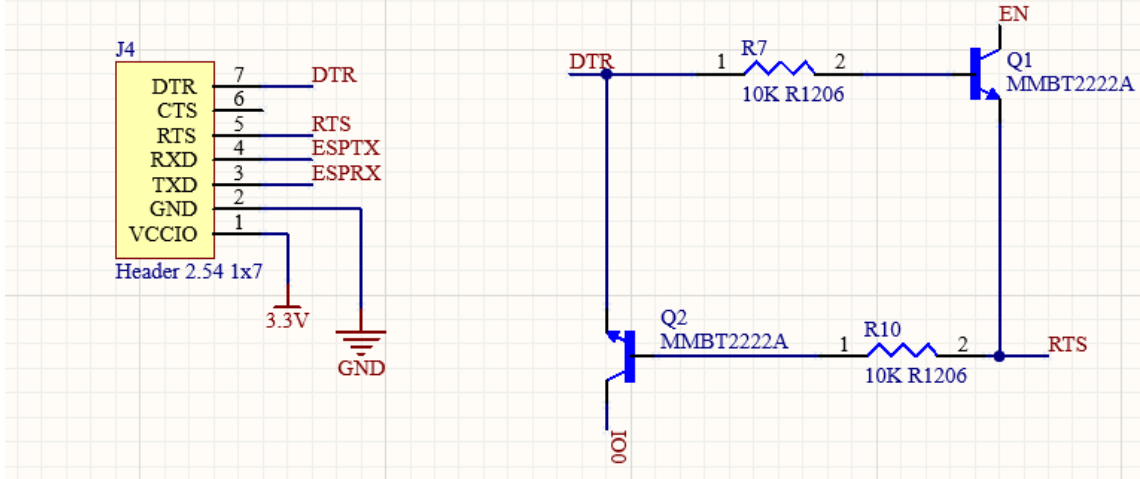


Şekil 2.16 I2C Pull Up Direnç Bağlantısı

2.2.3 CP2102 Programlayıcı Bağlantısı

Baskılı devre kartımızda bulunan CP2102 entegresi ile USB üzerinden UART bağlantısı kurulduğunu belirtmiştik. Burada bulunan ESPTX ve ESPRX bağlantıları mikrodenetleyicinin çapraz bacaklarına Şekil 2.12'deki gibi bağlanmıştır. Bu modülün diğer bağlantıları [23]'teki CP2102 datasheetinden faydalanılarak Şekil 2.17'de verilmiştir. NPN transistör olarak MMBT2222A modeli tercih edilmiştir.

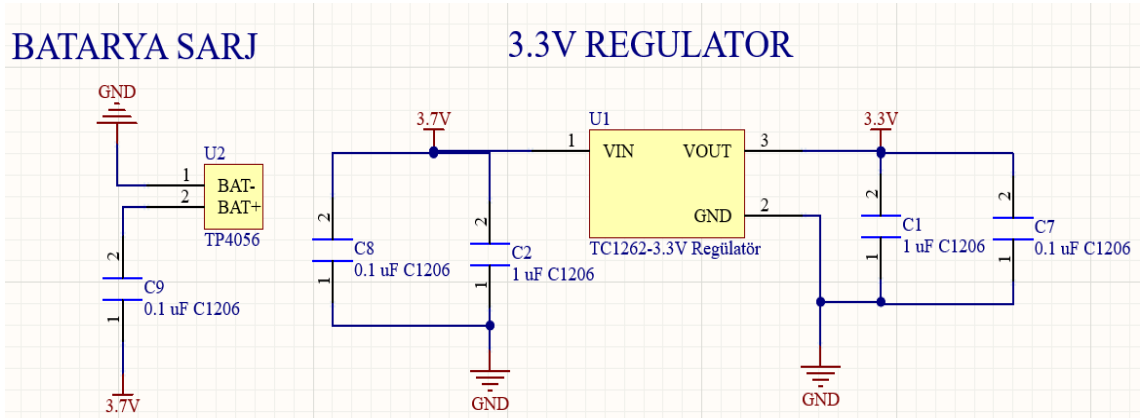
CP2102 PROGRAMLAYICI



Şekil 2.17 CP2102 Programlayıcı Bağlantısı

2.2.4 Batarya ve Regülatör Bağlantıları

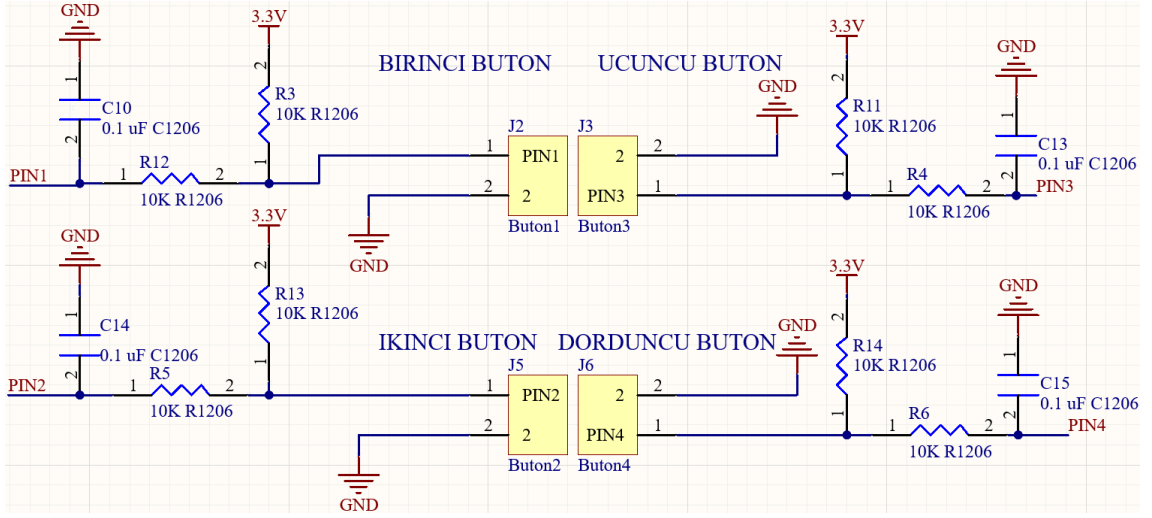
Batarya ve regülatör için gereken bağlantılar Şekil 2.18’de verilmiştir. Pilin toprağını BAT- , güç kısmını ise arada 0.1 μ F kapasite olacak şekilde bağlanmıştır. TC1262 regülatör ise [24]’te bulunan datasheetine uygun olarak bağlantıları gerçekleştirilmiştir. TC1262’nin giriş ve çıkış voltajlarına 1 μ F’lık kapasiteler bağlanmıştır. Böylelikle girişe 3.7V verildiğinde çıkışta sabit 3.3V alınması hedeflenmiştir.



Şekil 2.18 Batarya ve Regülatör Bağlantıları

Batarya şarj aletinde bulunan 0.1 μ F kapasite bağlantısının aynısının TC1262 3.3V regülatörde de olduğu görülmektedir. Bu ekstra kapasite bağlantısının sebebi gürültüyü engellemek ve sistemin daha stabil çalışmasını sağlamaktır.

2.2.5 Buton Bağlantıları



Şekil 2.19 Buton Bağlantıları

Projede kullanılan butonlar için gerekli bağlantı Şekil 2.19’da verilmiştir. PIN1, PIN2, PIN3 ve PIN4’ler ise Şekil 2.12’de gösterildiği gibi mikrodeneleyicinin ilgili bacaklarına bağlanmıştır. Fonksiyonları yerine getirir bir buton bağlantısı yaparken R3, R4, R5 ve R6 dirençleri gibi sadece pull-up direnci bağlamak yeterlidir. Fakat bu durumda bounce oluşabilmektedir.

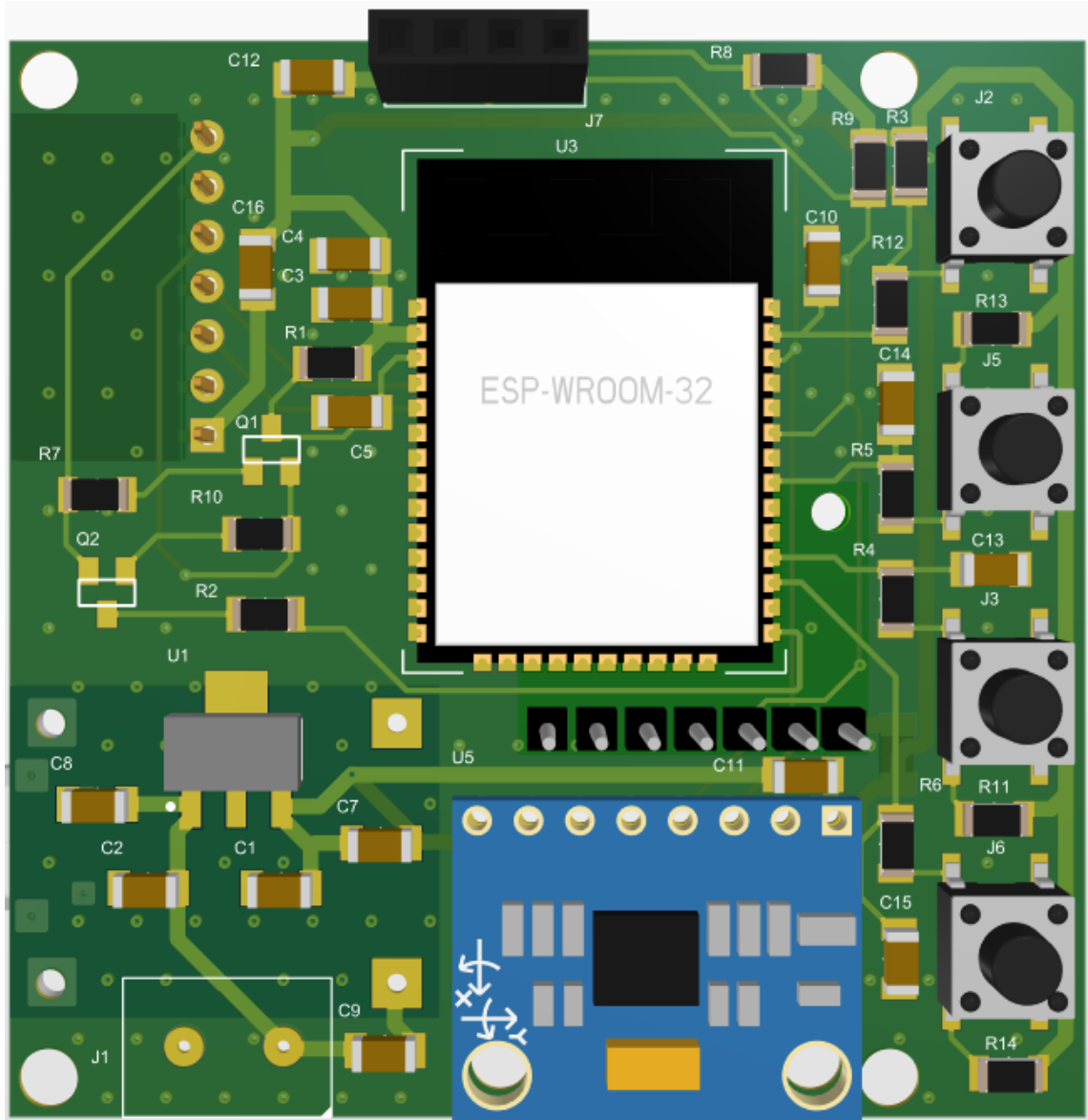
Bir switchin durumunu değiştirdiğiniz zaman meydana gelen mekanik titreme sonucunda anahtar birkaç defa açık ve kapalı pozisyonlar arasında gidip gelir. Bu olay mikrosaniyeler düzeyinde gerçekleşir ve Switch Bounce olarak adlandırılır.

Switch bounce olayı devrelerimizde kısa süreli noise etkisi oluşturduğu için Mux, mikrodeneleyici vb. kullandığımız devrelerde problem oluşturması yüksek bir olasılıktır. Bu bounce etkisini önlemeye debounce denir. Bounce önlemenin yaygın yollarından biri de bu devrede olduğu gibi basit bir RC filtresiyle bunu sağlamaktır.

2.2.6 Şematiğin Toplu Hali

Şematiğin bütün hali Şekil 2.20’de verilmiştir. Bu aşamaya kadar belirtilen tüm modüller ve elemanları kapsayan bu şematik tasarımından sonraki aşama ise PCB tasarımından oluşmaktadır.

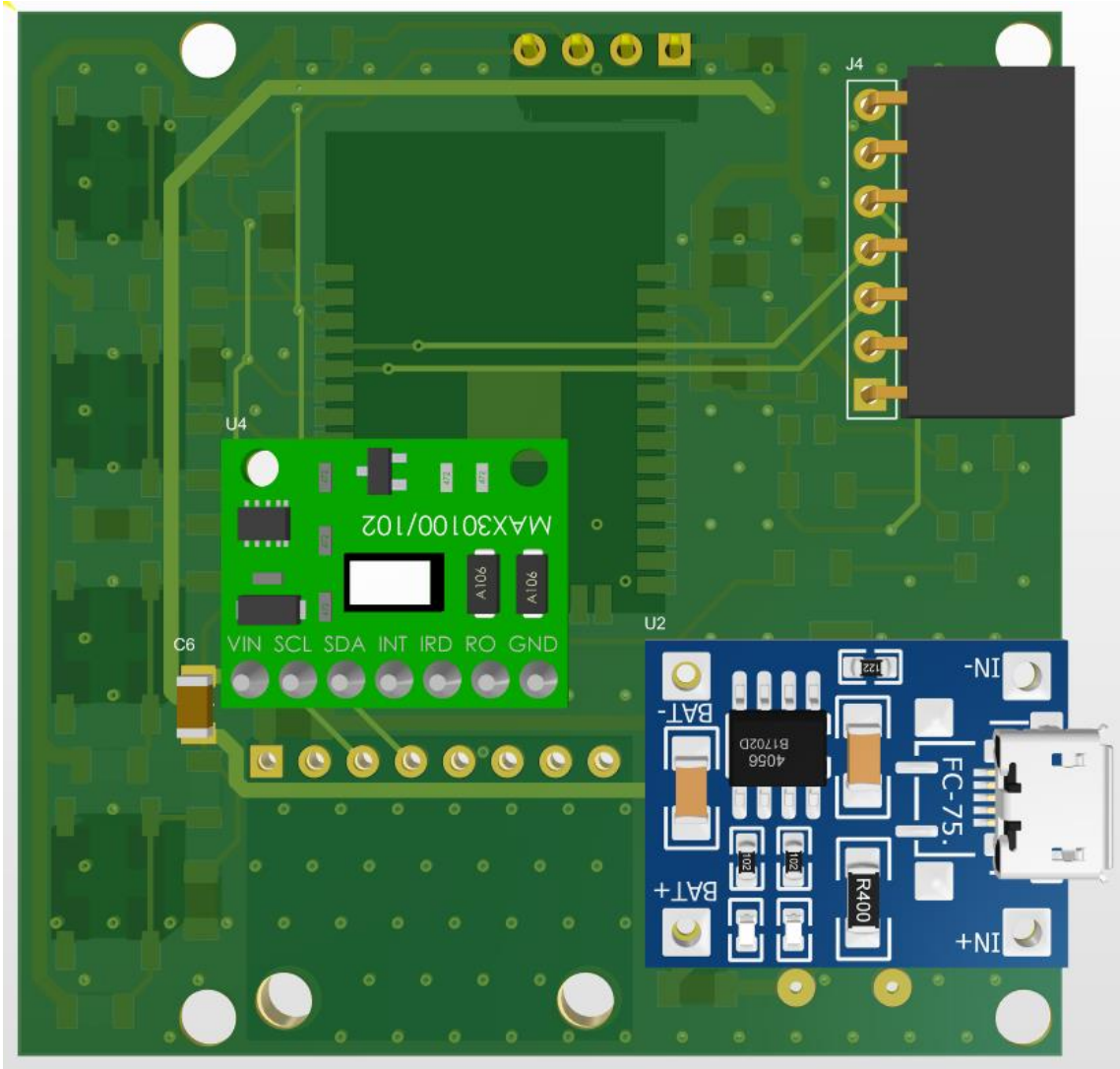
2.3.1 PCB Tasarımının Ön Yüz 3 Boyutlu Görünümü



Şekil 2.22 PCB Tasarımının Ön Yüz 3 Boyutlu Görünümü

PCB Tasarımının ön yüz 3 boyutlu görünümü Şekil 2.22’de verilmiştir. PCB kartımızın boyutları 55mm x 55mm olarak belirlenmiştir. ESP32-WROOM-32D mikrodenetleyici, SH1106 OLED ekran için gerekli headerlar, MPU6050 6 eksen ivme ve gyro sensörü, TC1262 regülatör ve butonlar ön yüzde konumlandırılmıştır.

2.3.2 PCB Tasarımının Arka Yüz 3 Boyutlu Görünümü

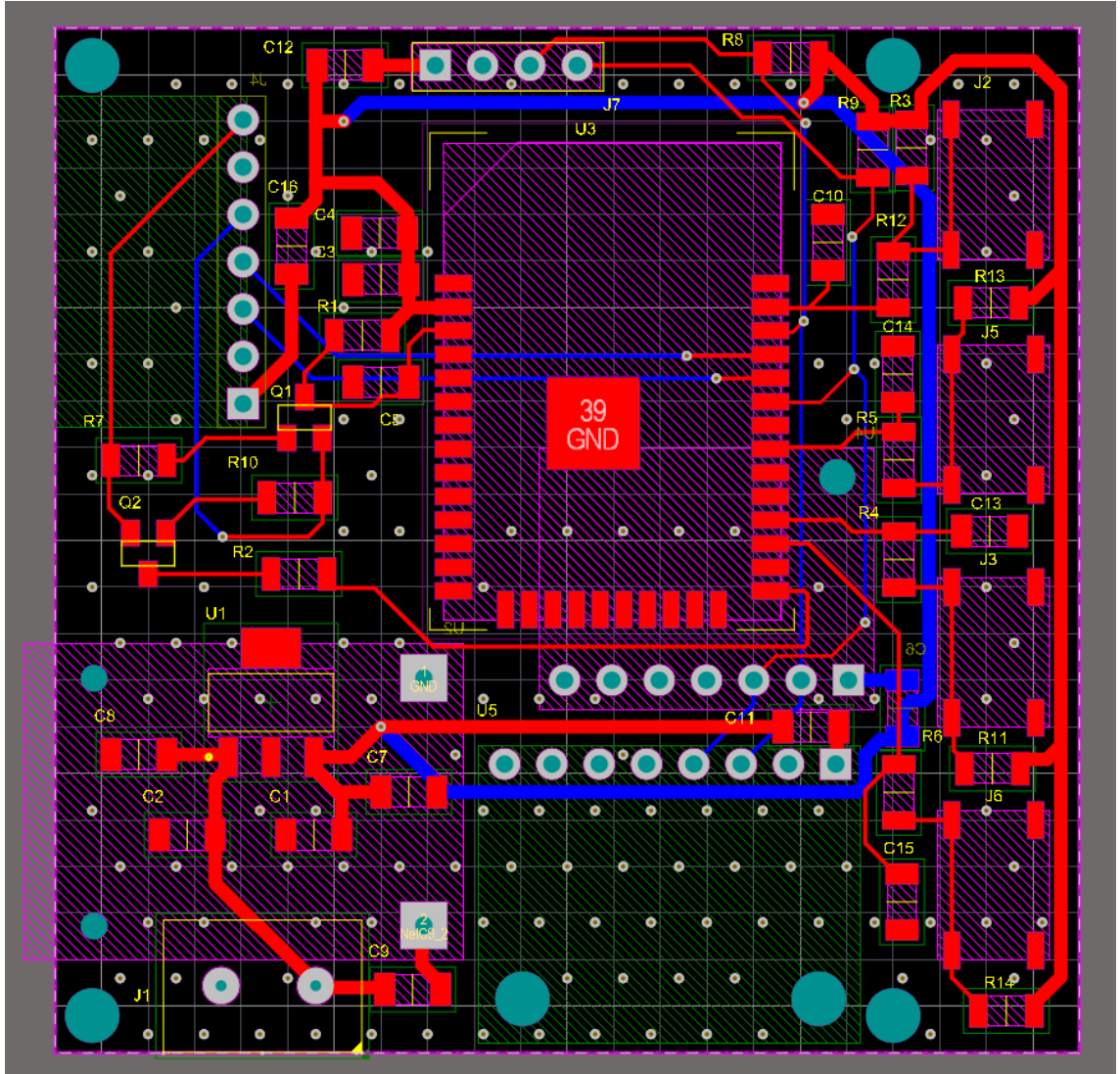


Şekil 2.23 PCB Tasarımının Arka Yüz 3 Boyutlu Görünümü

PCB tasarımının arka yüz 3 boyutlu görünümü Şekil 2.23'te verilmiştir. Bu tarafta ise MAX30100 kalp atış hızı sensörü ve nabız oksimetresi, TP4056 1A lityum polimer şarj devresi ve mikrodenetleyiciyi CP2102 USB-UART çevirici modül ile programlayabilmek için dışarıya doğru uzanan 1 x 7'lik header bulunmaktadır.

MAX30100 Kalp atış hızı sensörü ve nabız oksimetresinin arka yüzde olmasının sebebi ölçüm yapabilmek için vücuda temas etmesi gerekmesindendir. Bunun dışındaki tasarımda modüller, minimum yol mesafesi ile bağlantı sağlayabilecek şekilde dizayn edilmiştir.

2.3.3 PCB Tasarımının 2 Boyutlu Görünümü



Şekil 2.24 PCB Tasarımının 2 Boyutlu Görünümü

PCB tasarımının 2 boyutlu görünümü Şekil 2.24'te verilmiştir. Bu aşamadan sonra PCB tasarımı evresi tamamlanmıştır. Kart bastırılmak üzere Çin'den sipariş verilmiştir.

2.4 Donanım İçin Gerekli Yazılımın yazılması

2.4.1 Tümleşik Geliştirme Ortamları

Tümleşik geliştirme ortamı (IDE) bilgisayar programcılarının hızlı ve rahat bir şekilde yazılım geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür [25].

IDE'nin amacı, birden fazla geliştirme yardımcı programını bir araya getirmek için gereken yapılandırmayı azaltmaktır. Bunun yerine, tek bir uyumlu birim ile aynı yetenek kümesini sağlar. Bunlar; kurulum süresini kısaltmak, özellikle IDE'yi kullanmayı öğrenmenin tüm bireysel araçları manuel olarak entegre etmekten ve öğrenmekten daha hızlı olduğu durumlarda geliştirici verimliliğini arttırabilmektedir [26]. Tüm geliştirme görevlerinin daha sıkı entegrasyonu, yalnızca kurulum görevlerine yardımcı olmanın ötesinde genel üretkenliği artırma potansiyeline sahiptir. Örneğin, kod düzenlenirken sürekli olarak ayrıştırılabilir, sözdizimi hataları ortaya çıktığında anında geri bildirim sağlamaktadır. Böylece geliştiricilerin bir IDE ile kodu çok daha hızlı ve daha kolay bir şekilde hata ayıklamasına izin vermektedir [27].

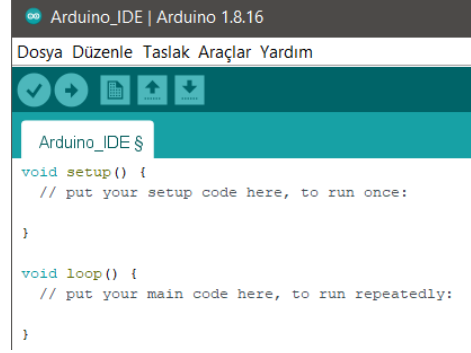
2.4.1.1 Arduino IDE

Arduino, temelde, çevrelerindeki fiziksel dünyayı algılayabilen ve kontrol edebilen dijital cihazlar ve etkileşimli nesneler oluşturmak için kullanılan açık kaynaklı bir bilgisayar donanım - yazılım platformudur. Etkileşimli nesneler veya ortamlar yaratmakla ilgilenen herkes için tasarlanmıştır. Arduino'nun yazılımı, Linux, Windows ve Macintosh gibi her türlü işletim sistemine çok uygundur. Programlama dili kitaplıkları ve genişletilebilir ve değiştirilebilir. Yeni başlayanlar için kullanımı çok basit ve aynı zamanda ucuzdur. Sensörleri ve aktüatörleri kullanarak çevre ile etkileşime girebilen cihazlar oluşturmak için kullanılabilir. Bazı yaygın örnekler arasında robot, termostatlar ve hareket dedektörleri bulunur.

Arduino için Entegre Geliştirme Ortamı (IDE), C ve C++ dilleri ile yazılmış bir platformlar arası uygulamadır. Arduino uyumlu kartlara program yazmak ve yüklemek için kullanılır, aynı zamanda 3. taraf çekirdekler ve satıcıların geliştirme kartları içinde kullanılabilir [28].

Arduino IDE'nin kaynak kodu GNU Genel Kamu Lisansı sürüm 2 ile yayınlanmıştır. Arduino IDE'si, özel kod yapılandırması kuralları kullanarak C ve C ++ dillerini destekler. Arduino IDE, Wiring projesinde bulunan birçok yaygın giriş ve çıkış prosedürünü bir yazılım kütüphanesi ile sağlar. Kullanıcı tarafından yazılan kod iki temel fonksiyon gerektirmektedir, başlama noktası olarak adlandırabileceğimiz bölüm ve ana döngünün

gerçekleşeceği kısım. Bunlar GNU araç zinciri sayesinde bağlanır ve derlenir. Arduino IDE temelde, çalıştırılabilir kodu hexadecimal format ile metin dosyasına işler. Ardından kullandığımız Arduino IDE'si bu metin dosyasını bağlı olan Arduino kartının firmware'ına yükleyici program ile aktarmayı gerçekleştirir [29].

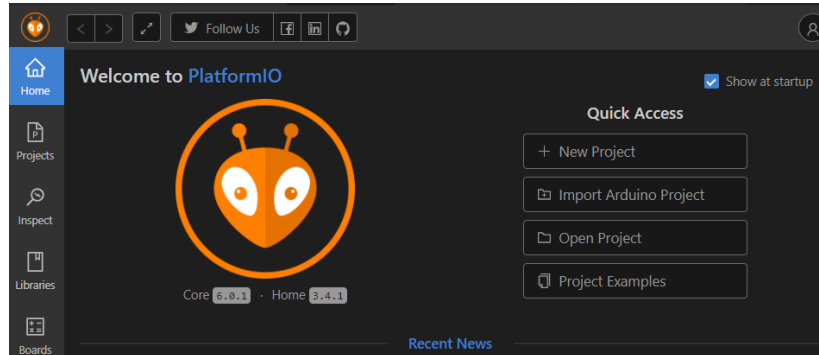


Şekil 2.25 Arduino IDE

Bu projede mikrodenetleyici olarak ESP32 kullanılmıştır. Arduino IDE programının ESP32 kartını desteklemesi ve destekleyici kütüphaneler bulundurması sebebiyle Arduino IDE projede kullanılan tümleşik geliştirme ortamlarındandır.

2.4.1.2 PlatformIO

PlatformIO, profesyonel kod düzenleyicisi ve çok sayıda platform için kaynak kodunu derleyebilmeniz için derleyici, bir hata ayıklayıcısı ve ayrıca programlama için bir dizi çok ilginç işlev ve araç ile entegre bir geliştirme ortamıdır. Seri test monitörü, kod analizörü, kod otomatik tamamlama, kütüphane yöneticisi vb özellikler sunmaktadır, açık kaynaklıdır ve yetenekleri eklentiler veya uzantılar aracılığıyla genişletilebilir. Projede kullanılan ESP32 mikrodenetleyicisini desteklediği ve C++ dilinde yazılıma olanak sağladığı için kullandığımız tümleşik geliştirme ortamlarından biri olmuştur.



Şekil 2.26 PlatformIO

2.4.2 Kullanılan Kütüphaneler

Kütüphaneler çoğu programlama platformunda kullanılan, yazılım genişletilmesine ve kolaylaştırılmasını sağlayan hazır kod dizileridir.

2.4.2.1 Wire Kütüphanesi

Arduino Wire kütüphanesi, I2C haberleşme yöntemiyle donanımsal olarak senkron seri haberleşme sağlayan kütüphanedir. Bu kütüphane ile bir clock hattı ve bir veri hattı kullanarak 2 ya da daha fazla cihaz ile donanımsal senkron seri haberleşme sağlanabilir. Wire kütüphanesi donanımsal bir kütüphane olduğu için hızlı haberleşme sağlamaktadır. Kullanılan ESP32 kartı için wire kütüphanesi ile haberleşmeye olanak sağlayan pinler IO21(SDA) ve IO22(SCL) pinleridir.

I2C haberleşme yöntemi ile birden fazla cihaz haberleşme sağlayabilir ancak, bir cihaz veri gönderirken diğer cihazlar veri yolunun boşalmasını beklemek zorundadır. Aynı anda 2 cihaz veri gönderirse, veri hattında çakışma meydana gelecektir.

2.4.2.2 WiFi Kütüphanesi

WiFi kütüphanesi kullanılarak ağ bağlantısı (yerel ve İnternet) etkinleştirilmektedir. Bu kitaplık ile; sunucuları, istemcileri somutlaştırabilmekte ve WiFi üzerinden UDP paketleri gönderip alınabilmektedir. WiFi Kütüphanesi ile mikrodenetleyiciler koruması kapalı veya şifreli ağlara (WEP, WPA) bağlanılabilmektedir. IP adresi statik olarak veya bir DHCP aracılığıyla atanabilir. Kütüphane ayrıca DNS'yi de yönetebilmektedir [30].

WiFi Kütüphanesi Wifi destekleyen işlemcilerde ve WiFi kartlarında İnternet ortamına bağlanılmasına, veri alışverişine olanak sağlar.

2.4.2.3 HttpClient Kütüphanesi

Web sunucuları ile etkileşimi kolaylaştıran, bir web sunucusuna HTTP GET, POST ve PUT istekleri yapmak için kullanılan kitaplıktır. İstemciden türetilen herhangi bir sınıfla çalışır, bu nedenle minimum kod değişikliği ile Ethernet, WiFi ve GSMClient arasında geçiş yapmak mümkündür [31].

Projemizde gerekli verilen web sunucularından alınması için kullanılmaktadır.

2.4.2.4 U8g2 Kütüphanesi

U8g2, gömülü cihazlar için tek renkli bir grafik kitaplığıdır. U8g2, birçok mikrodenetleyici destekleyen ve projede kullanılan ekran dahil olmak üzere çoğu tek renkli OLED (SH1106, SSD1306, MAX7219) ve LCD ekranları desteklemektedir [32].

U8g2 kütüphanesi birçok yazı tipini destekler ve neredeyse yazı tipi yüksekliğinde herhangi bir kısıtlama yoktur. Ayrıca tüm çizgi, kutu, daire çizimi gibi grafik türlerini desteklemekle beraber drawXBM ve bitmap komutları ile ekranda resim gösterilmesini sağlar. Ekranı oluşturmak için mikrodenetleyicide biraz bellek gerektirir.

2.4.2.5 MAX30100 Kütüphanesi

Bu kütüphane, MAXIM MAX30100 Kalp Atış Hızı ve SpO2 sensör çipi ile arayüz oluşturmak üzere tasarlanmıştır. Kütüphanenin mevcut durumu, Kalp Atış Hızı belirleme için IR değerlerinin okunmasına ve herhangi bir kaydın değiştirilmesine izin vermektedir. Wire kütüphanesini destekleyen Arduino, ESP32 ve birçok mikrodenetleyici tarafından desteklenmektedir. Sensöre gerekli olan tek bağlantı I2C veriyoludur, I2C hattı sayesinde kalp atış hızı ve kandaki oksijen oranını belirlenmesine yardımcı olmaktadır.

2.5 Kılıf Tasarımı

Günümüzde ürünlerin işlevselliğinin yanında dış tasarımın da önemli bir yeri vardır. Ürünün ergonomik, konforlu ve aynı zamanda şık, beğenilen bir tasarıma sahip olması gerekir. Akıllı bir bileklikte tasarımda en önemli unsur genel yapıyı çevreleyen bir kılıf ve kordon tasarımıdır. Bu çalışmada da son ürün olarak bu gerekliliklere uygun bir tasarım yapılmıştır. Tasarım yapılırken kullanılan devreyi çevreleyip dış etkenlerden koruyabilen ve aynı zamanda vücudun anatomisine uygun, vücuda rahatsızlık vermeyen, kullanıcıya şık bir görüntü sunan bir kılıf tasarlanmaya çalışılmıştır.

3D tasarım yapmak için kullanılan çok sayıda gelişmiş tasarım programları bulunmaktadır. Bu programlardan en yaygınları Fusion 360, Solidworks, Tinkercad, Sketchup programlarıdır. Program seçimi yapılırken kaynak miktarı, kullanım kolaylığı ve uygulamaların çıktı kalitesi göz önünde bulundurulmuştur.

Koşullara bakıldığında bu proje için kullanılabilecek en uygun programın Solidworks olduğuna karar verilmiş ve tasarım bu program üzerinden gerçekleştirilmiştir.

2.5.1 SolidWorks

SolidWorks, bilgisayar destekli 3 boyutlu katı modelleme ve tasarım 3D CAD yazılımıdır. SolidWorks Corporation, Aralık 1993'te Massachusetts Teknoloji Enstitüsü mezunu Jon Hirschtick tarafından kurulmuştur. Başlangıçta Hirschtick, kullanımı kolay, uygun maliyetli ve Windows masaüstünde kullanılabilen 3D CAD yazılımı oluşturma hedefiyle bir mühendis ekibini işe almıştır. Daha sonra Concord, Massachusetts'ten faaliyete geçen SolidWorks, ilk ürünü SolidWorks 95'i Kasım 1995'te piyasaya sürdü. SolidWorks, 100 milyon dolarlık bir gelir şirketi haline geldi.

SolidWorks bir katı modelleyicidir ve başlangıçta PTC tarafından geliştirilen parametrik özellik tabanlı bir yaklaşım kullanır. Parametreler, çizgi uzunlukları veya daire çapları gibi sayısal parametreler veya teğet, paralel, eş merkezli, yatay veya dikey gibi geometrik parametreler olabilir. Sayısal parametreler, tasarım amacını yakalamalarına izin veren ilişkilerin kullanımı yoluyla birbirleriyle ilişkilendirilebilir. Tasarım amacı, parçanın yaratıcısının değişikliklere ve güncellemelere nasıl yanıt vermesini istediğidir.

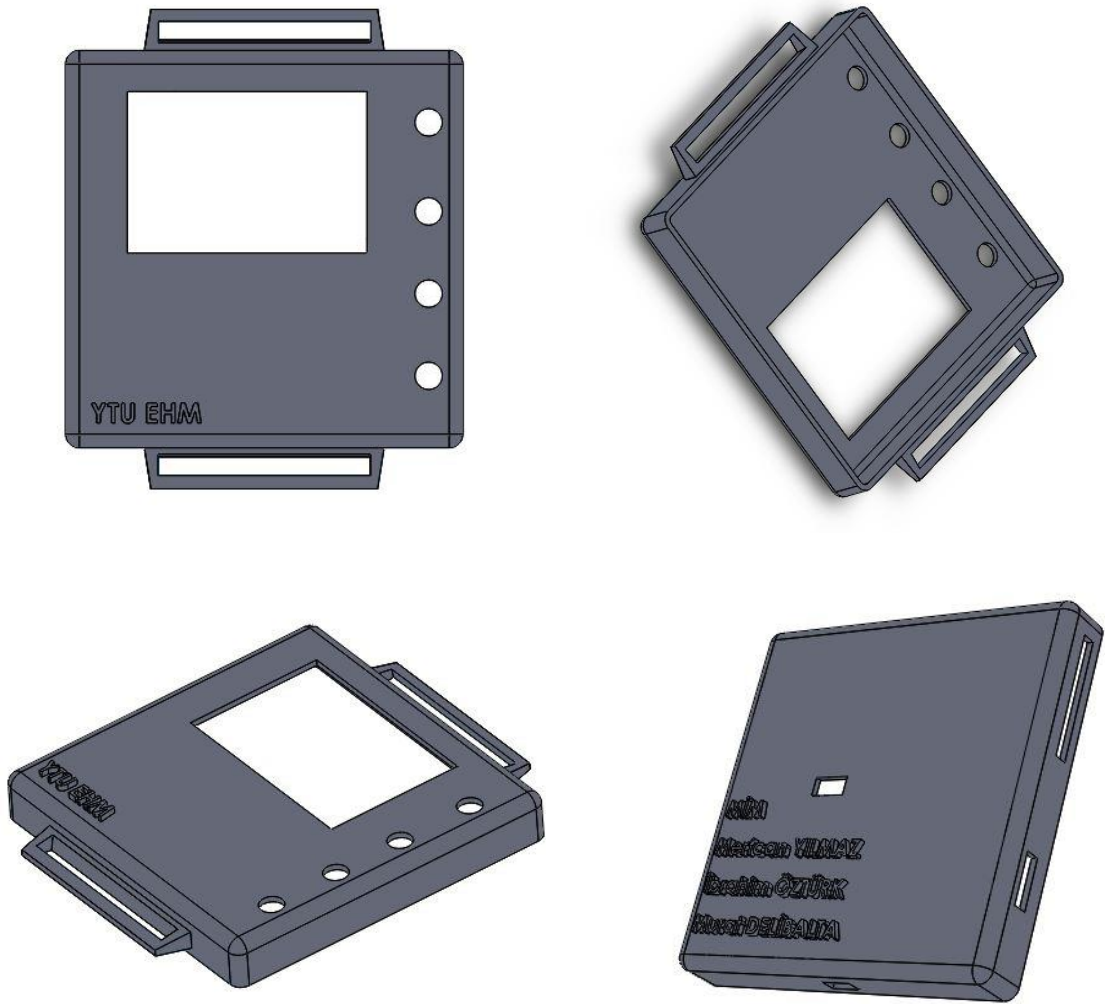
SolidWorks'te bir model oluşturmak genellikle bir 2B çizimle başlar. Çizim, noktalar, çizgiler, yaylar, koniler ve eğri çizgiler gibi geometriden oluşur. Geometrinin boyutunu ve konumunu tanımlamak için çizime boyutlar eklenir. SolidWorks'ün parametrik doğası, ölçümlendirmelerin ve ilişkilerin geometriyi yönlendirdiği anlamına gelir. Çizimdeki ölçümlendirmeler bağımsız olarak veya çizimin içindeki veya dışındaki diğer parametrelerle olan ilişkilerle kontrol edilebilir.

Çizim ilişkilerinin çizim geometrisine göre teğetlik, paralellik ve eşmerkezlilik gibi koşulları tanımlaması gibi, montaj ve montaj ilişkilerini de tek tek parçalar veya bileşenlerle ilgili olarak eşdeğer ilişkileri tanımlayarak montajların kolay oluşturulmasını sağlar.

Teknik resimler, parçalardan veya montajlardan oluşturulabilir. Görünümler katı modelden otomatik olarak oluşturulur ve notlar, boyutlar ve toleranslar daha sonra gerektiğinde teknik resme kolayca eklenebilir [33].

2.5.2 SolidWorks ile Kılıf Tasarımı

Kılıf tasarlanırken ilk olarak lehimleme işlemi yapılmış PCB devresi üzerinde gerekli ölçümler alınmıştır. Ekran, buton, şarj giriş soketi, programlayıcı headerları, batarya pin çıkışları ve nabız sensörünün kızıl ötesi led kısmı için açılacak çıkışların konumları belirlenmiştir. Daha sonra kordon yapısının tasarımı yapılmıştır. Sonrasında kılıfın taslak bir çizimi yapılarak ölçüler ve konumlar belirlenmiş olup bilgisayar destekli tasarım için ön hazırlık yapılmıştır. Son aşamada ise tasarım SolidWorks programında çizilmiştir. Şekil 2.27 'de kılıf tasarımının son hali görülmektedir.



Şekil 2.27 SolidWorks ile Akıllı Bileklik Kılıf Tasarımı

AKILLI BİLEKLİK TASARIMININ GERÇEKLENMESİ**3.1 Test için ESP32-WROOM-32D Geliştirme Kartının Kullanılması**

BÖLÜM 2’de Akıllı bileklik tasarımının şematik ve PCB tasarımları verilmişti. Bu PCB tasarımının bastırılmadan önce gerçek hayatta da gerçekleşmesi gerekmektedir. Bunun sebebi devre bağlantılarında bir yanlışlık olup olmadığını kontrol etmektir. Bu sebepten ötürü Şekil 3.1’de görülen ESP32-WROOM-32D geliştirme modülüyle beraber Bölüm 2.2.6’da bulunan şematiğin tüm hali, PCB devresi bastırılmaya göndermeden önce denenmiştir. Bu devrenin fiziksel ortamda doğru çalıştığı görüldükten sonra PCB bastırılmıştır.



Şekil 3.1 ESP32-WROOM-32D Geliştirme Modülü

3.2 PCB’nin Lehimlenmesi

Bundan sonraki aşamada PCB’nin Çin’den gelmesi beklenmiştir. 6 iş günü süresi sonucunda elimize ulaşmıştır. Elimize ulaşan PCB’yi test etmek için lehimleme işlemine

başlanmıştır. Şekil 3.2’de ön yüzünün lehimlenmiş hali görülmektedir. Şekil 3.3’te ise arka yüzünün lehimlendikten sonraki görünümü verilmiştir.



Şekil 3.2 PCB'nin Ön Yüzünün
Lehimlenmiş Hali



Şekil 3.3 PCB'nin Arka Yüzünün
Lehimlenmiş Hali

3.3 Kılıfın 3D Yazıcı ile Bastırılması

Solidworks tasarımı yapıldıktan sonra tasarım üzerinde gerekli kontroller yapılarak son tasarım çıktısı elde edilmiştir. Tasarımın çıktısı alınabilmesi için .STL dosya uzantılı belge alınması gerekmektedir. Bu belge alındıktan sonra bu belgeyle 3D tasarımın çıktısı alınmıştır. Şekil 3.4 ve Şekil 3.5'te kılıfın gerçekleştirilmiş halleri gösterilmiştir.



Şekil 3.4 3D Printer ile Bastırılmış Kılıfın
Ön Yüzü



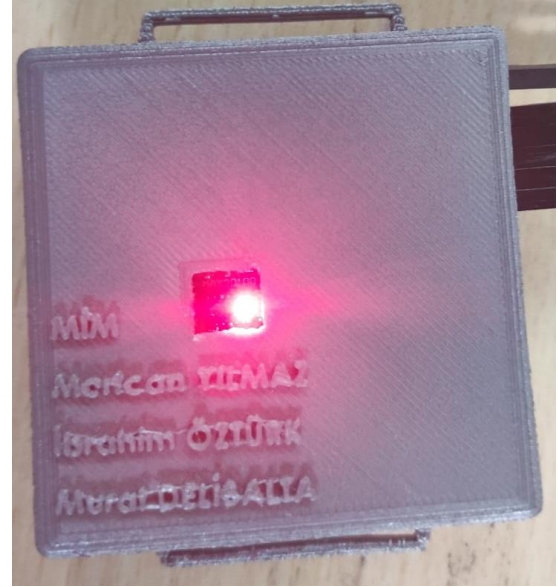
Şekil 3.5 3D Printer ile Bastırılmış Kılıfın
Arka Yüzü

3.4 Akıllı Bileklik Tasarımının Son Hali

PCB lehimlendikten ve kılıfın 3D yazıcı ile baskısı tamamlandıktan sonra tasarım son halini almıştır. Akıllı bileklik tasarımımızın son hali Şekil 3.6, 3.7 ve 3.8’de verilmiştir.



Şekil 3.6 Akıllı Bileklik Tasarımının Önden Görünümü



Şekil 3.7 Akıllı Bileklik Tasarımının Arkadan Görünümü



Şekil 3.8 Akıllı Bileklik Tasarımının Yandan Görünümü

3.5 Ekran Çıktıları

3.5.1 Ana Ekran

Ana ekranda Şekil 3.6’da görüldüğü gibi saat, dakika, saniye, gün, ay ve tarih verileri görülmektedir. Bunun yanında günlük hayatta merak edilen kalp atış hızı, adım sayısı ekranda mevcuttur. Alarmin aktif olup olmadığını gösteren alarm göstergesi ve WiFi bağlantısı durumunu gösteren WiFi simgesi ekranda görülmektedir.

3.5.2 Borsa Ekranları

Ana Menüde bulunan borsa ekranında bir borsa simgesi bulunmaktadır. Borsa sekmesinin içine girildiğinde günümüz Türkiye'sinde popüler, sürekli kontrol edilme ihtiyacı olan endeks, döviz, kripto para ve değerli madenlerin anlık verileri ekrana verilmektedir. Ekranda gösterilen veriler BIST 100, Dolar, Euro, Bitcoin, Ethereum, Altın ve Gümüş'ün anlık verileridir.



Şekil 3.9 Ana Menüde Bulunan Borsa Ekranı

3.5.2.1 BIST100 Endeksi



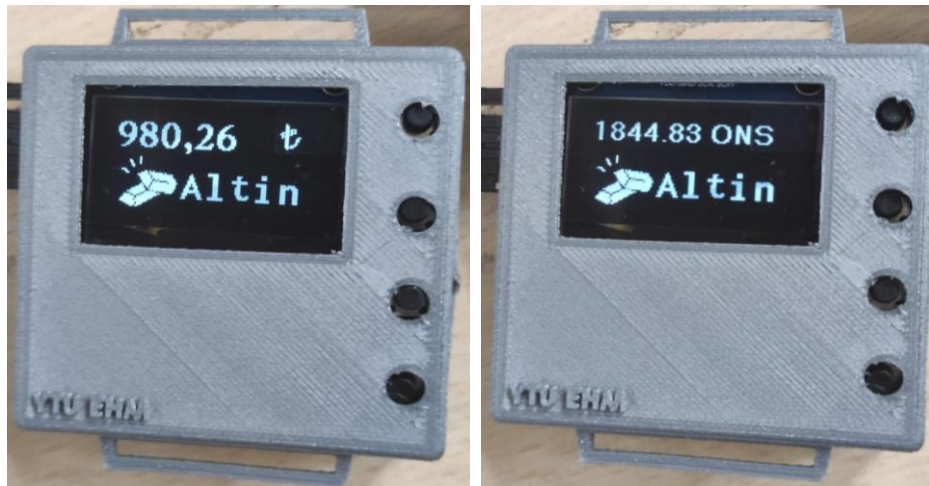
Şekil 3.10 BIST100 Endeksi

3.5.2.2 Kripto Paralar



Şekil 3.11 Ethereum ve Bitcoin

3.5.2.3 Altın



Şekil 3.12 Gram ve ONS Altın

3.5.2.4 Gümüş



Şekil 3.13 Gram ve ONS Gümüş

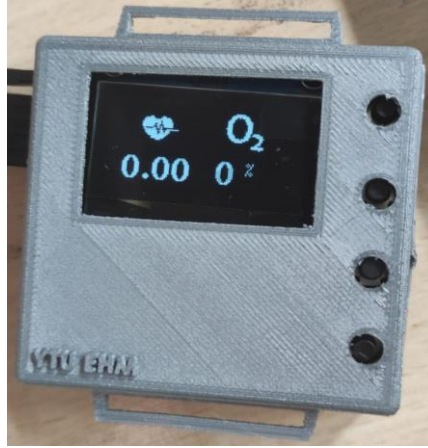
3.5.2.5 Döviz



Şekil 3.14 Dolar ve Euro

3.5.3

Kalp Atış Hızı ve Kandaki Oksijen Oranı Göstergesi Ekranı



Şekil 3.15 Kalp Atış Hızı ve Kandaki Oksijen Oranı Göstergesi Oranı

3.5.4 Adımsayar Göstergesi Ekranı



Şekil 3.16 Adımsayar Göstergesi

3.5.5 Sıcaklık Göstergesi Ekranı



Şekil 3.17 Sıcaklık Göstergesi

3.5.6 Alarm Ekranı

Alarm ekranında alarmin saatini, dakikasını belirlemek için 2 satır ve aktif veya pasif konuma getirmek için 1 satır bulunmaktadır.



Şekil 3.18 Ana Menüde Bulunan Alarm Ekranı ve Alarm Ayarı

3.5.7 Ayarlar Ekranı

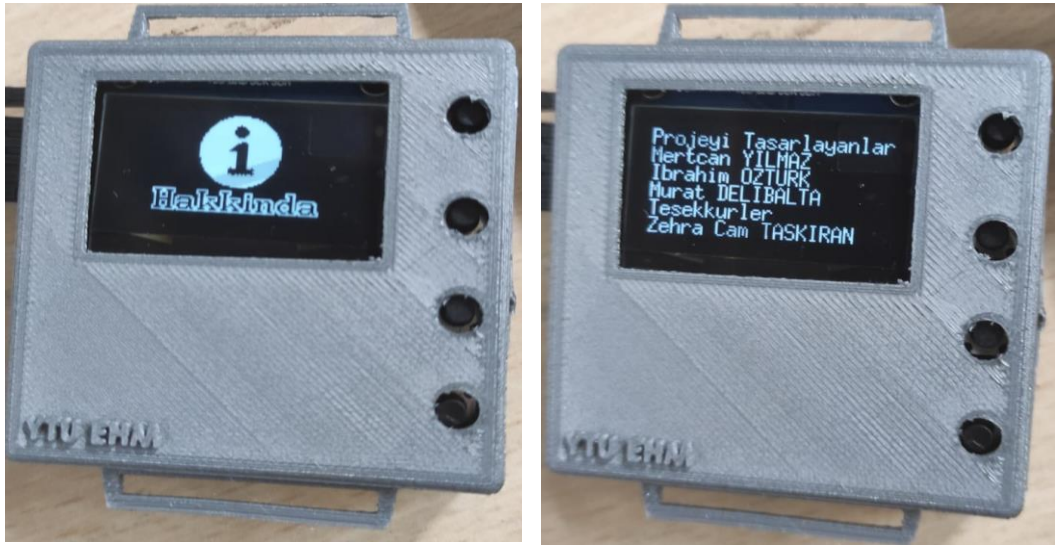
Ayarlar kısmında saat ayarlarını, tarih ayarlarını ve ekran ayarlarının düzenleneceği bölümler mevcuttur.



Şekil 3.19 Ana Menüde Bulunan Ayarlar Ekranı ve Saat, Tarih, Ekran Ayarı

3.5.8 Hakkında Ekranı

Hakkında bölümünde projeyi tasarlayan öğrenciler belirtilmiş, destekleyen ve yardımcı olan öğretim görevlilerine teşekkürler sunulmuştur.



Şekil 3.20 Ana Menüde Bulunan Hakkında Ekranı ve Hakkında Yazısı

PROJE TASARIMINDA GÖZ ÖNÜNE ALINAN GERÇEKÇİ KISITLAR VE KOŞULLAR

Tasarımı gerçekleştirilen sistem;

- Bu projenin temel amacı insanların hayatını kolaylaştırmaktır. Bu bağlamda özgün değer olan borsa, kripto para, kur ve değerli maden gibi verilerin hızlı tepkime süreleriyle ve sık frekanslarla alınması projenin can alıcı noktasıdır.
- Bu projenin bir diğer amacı insanların sağlık verilerini düzenli olarak takip edebilmesidir. Herhangi bir sağlık problemine yol açmayıp, ortaya çıkan problemlerin en kısa zamanda fark edilmesi ve aksiyon alınmasına imkan sağlar.
- Sistemde kullanılan sensörlerin %85 veya daha yüksek oranda doğrulukta veri toplaması hedeflenmiştir.
- Sistemin tasarımında öncelik, insan vücuduna herhangi bir zarar vermemesidir. Bunun için kullanılan malzemelerin insan vücuduna zarar verebilecek olanları kullanılmayarak bu tarzda oluşabilecek olayların önüne geçmek amaçlanmıştır. Aynı zamanda kullanılan donanımların çevre dostu yapısı sebebiyle, çalışırken çevreye zarar vermemektedir.
- Projede kullanılan tüm entegreler uluslararası piyasada kabul görmüş sertifikalara sahiptir.
- Bu projede kullanılan malzemelerin maliyeti düşük tutulmuştur. Bununla birlikte ürünün geniş bir kitleye ulaşabilmesine imkan sağlanmıştır.

SONUÇ VE ÖNERİLER

Bu çalışmada, akıllı bileklik ve akıllı bilekliklerde bulunması gereken özellikler incelendi. Farklı sensörler ile yapılan akıllı bileklik tasarımları incelendi. Elde edilen bu veriler doğrultusunda akıllı bileklik tasarımı konusunda araştırmalara başlandı.

Birinci bölümde, akıllı bileklik tasarımının literatür taraması ile başlanılmıştır. Literatür taraması yapılırken piyasada erişilebilecek çoğu kaynak incelenmiş ve iyice kavranmıştır. Sonrasında ise akıllı bilekliğin geleceğinden bahsedilmiştir. Bu bölümün sonunda yaptığımız bitirme çalışmasının amaçları belirtilmiş ve ikinci bölüme geçilmiştir.

İkinci bölümde, ilk bölümde yapılan araştırmalar sonucunda elde edilen bilgilere göre kullanılacak malzeme seçimlerine başlanmıştır. Projede ESP32 mikrodenetleyici, ivme sensörü, kalp atış hızı sensörü ve nabız oksimetresi, OLED grafik ekran, 3.7V LiPo pil, LiPo şarj devresi, voltaj regülatörü, gerekli sayıda buton ve programlayabilmek için USB-UART çevirici modül kullanılmasına karar kılınmıştır.

Belirtilen bu malzemelerin tanıtımı sonrasında tasarımın şematik yapısına geçilmiştir. Belirtilen her bir malzeme için bağlantılar gösterilmiştir. Projede ihtiyaç duyulan haberleşme protokolleri anlatılmıştır. Bu aşamadan sonra ise PCB tasarımı yer almaktadır. Oluşturulan şematik yapıya göre PCB tasarımı tamamlanmıştır.

PCB tasarımı sonrasında bu kart için gerekli yazılımın yazılması gerekmektedir. Bu aşamada ise tümleşik geliştirme ortamlarından ve kullanılan kütüphanelerden bahsedilmiştir. Akıllı bileklik tasarımı bölümü adı altında son olarak kılıf tasarımı yapılmıştır.

Üçüncü Bölümde, akıllı bileklik tasarımının gerçekleşmesine yer verilmiştir. İlk olarak test için geliştirme kartının kullanılmasına ve bunun önemine değinilmiştir. Daha sonrasında elimize ulaşan PCB kartın lehimleme işlemleri gerçekleştirilmiştir. Bu aşamayla senkron olarak tasarlanan kılıf 3D yazıcı ile bastırılmıştır.

Bu aşamalardan sonra akıllı bilekliğimiz son halini almıştır. Son halini alan akıllı bilekliğin ekran çıktıları bu bölümün sonunda verilmiştir. Bu ekran çıktıları, ana ekran, borsa

ekranları, kalp atış hızı ve kandaki oksijen oranı göstergesi, adımsayar göstergesi, sıcaklık göstergesi, alarm ekranı, ayarlar ve hakkında ekranını kapsamaktadır.

Bundan sonraki bölümde ise proje tasarımında göz önüne alınan gerçekçi kısıtlar ve koşullar belirtilmiş olup projenin bölümleri tamamlanmıştır. Bu bölümlerin dışında EK kısmında; maliyet tablosu, katalog bilgileri ve akıllı bileklik tasarımı projesi için gereken kod verilmiştir.

Mühendislik etiği çerçevesinde; teknik anlamda kalitesi ve fiyatları göz önünde bulundurularak en iyi parçalar seçilmeye çalışılmış, tasarımlarda bu parçaların verimli çalışmaları göz önünde bulundurularak efektif tasarımlar gerçekleştirilmeye çalışılmıştır. Kullanılan parçalar hakkında piyasada kapsamlı araştırmalar yapılmış ve parçaların özellikleri hakkında doğru bilgiler rapora eklenmiştir.

Bu tasarımın gerçekleşmesinde birden fazla alanda elektronik elemanlar kullanılmıştır. Bu elektronik elemanları bir sistem içerisinde çalıştırmak, 4 yıllık lisans eğitimimizde öğrendiğimiz teorik bilgileri pratikte kullanmamızı sağlamıştır. Ayrıca iş hayatında oldukça önemli olan grup çalışması, iş bölümü, düzenli çalışma, zamanın verimli kullanılması gibi değerleri bize katmıştır.

- [1] C. W. K. Gritton, «What's wrong with smart TV?: To improve user experience,» *IEEE Consum. Electron. Mag.*, cilt 2, no. 4, pp. 40-43, 2013.
- [2] D. Pal, A. Tassanaviboon, C. Arpikanondt ve B. Papasratorn, «Quality of Experience of Smart-Wearables: From Fitness-Bands to Smartwatches,» *Smart Home Healthcare*, p. 49, 2020.
- [3] M. R. Srikrishnan, M. R. Srikrishnan ve J. Niresh, «The Evolution of Smart Wrist Band by Using Sensors,» *Journal of Siberian Federal University. Engineering & Technologies*, cilt 5, no. 13, pp. 529-530, 2020.
- [4] A. S. Ptoe, «Invited review: The link between stress and illness,» *J Psychosom Res*, cilt 35, no. 6, pp. 633-644, 1991.
- [5] C. Schubert, M. Lambertz, R. Nelesen, W. Bardwell, J.-B. Choi ve J. Dimsdale, «Effects of stress on heart rate convexity- a comparison between short-term and chronic stress,» *Biol Psychol*, cilt 80, no. 3, pp. 325-332, 2009.
- [6] M. Zubair ve C. Yoon, «Smart Wearable Band for Stress Detection,» *Gwangju*.
- [7] Y. S. Can, N. Chalabianloo, D. Ekiz, J. F.-. ' Alvarez, C. Repetto ve ' Giuseppe Riva, «Real-Life Stress Level Monitoring using Smart Bands in the Light of Contextual Information».
- [8] D. C. Han, H. J. Shin, S. H. Yeom ve Wanghoon Lee, «Wearable Human Health monitoring Band using Inkjet-printed Flexible Temperature Sensor,» *Journal of Sensor Science and Technology*, cilt 26, no. 5, pp. 301-305, 2017.
- [9] Q. Wei, H.-J. Park ve J. H. Lee, «Development of a Wireless Health Monitoring System for Measuring Core Body Temperature of Human from the Back of the Body,» *Hindawi Journal of Healthcare Engineering*, p. 8, 2019.
- [10] A. Kamišalić, I. Fister, M. T. Jr. ve S. Karakatič, «Sensors and Functionalities of Non-Invasive Wrist Wearable Devices,» cilt 18, no. 6, Mayıs 2018.
- [11] A. Brush, R. Mahajan ve A. Samuel, «Lab of things: simplifying and scaling deployments of experimental technology in homes,» *Handbook of Smart Homes*, pp. 1-9, 2014.
- [12] A. Samuel, A. Brush ve R. Mahajan, «Lab of things: building a research platform for connected devices in the home and beyond,» *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, cilt 18, no. 4, pp. 37-40, 2015.

- [13] J. Feminella, D. Pisharoty ve K. Whitehouse, «Piloteur: a lightweight platform for pilot studies of smart homes,» *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ACM, Memphis, Tennessee,, pp. 110-119, 2014.
- [14] B. Reeder ve A. David, «Health at hand: A systematic review of smart watch uses for health and wellness,» *Journal of Biomedical Informatics*, p. 270, 2016.
- [15] P. Corcoran, «The Internet of things: Why now, and what's next?,» *IEEE Consum. Electron. Mag.*, cilt 5, no. 1, pp. 63-68, Ocak 2016.
- [16] E. S ve C. Ş., «İktisadi Büyüme ve Araştırma & Geliştirme (Ar-Ge) Harcamaları İlişkisi Üzerine Teorik Bir İnceleme,» *Muş Alparslan Üniversitesi Sosyal Bilimler Dergisi*, no. 30.
- [17] «Vikipedi,» [Çevrimiçi]. Available: <https://tr.wikipedia.org/wiki/Mikrodenetleyici>. [Erişildi: Mayıs 2022].
- [18] «MAX30100 Datasheet,» Maxim Integrated, 2014. [Çevrimiçi]. Available: <https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>. [Erişildi: Mayıs 2022].
- [19] «TP4056 Datasheet,» NanJing Top Power ASIC Corp., [Çevrimiçi]. Available: <https://pdf.direnc.net/upload/tp4056.pdf>. [Erişildi: Mayıs 2022].
- [20] «Vikipedi,» [Çevrimiçi]. Available: <https://tr.wikipedia.org/wiki/Buton>. [Erişildi: Mayıs 2022].
- [21] «ESP32-WROOM-32D Datasheet,» Espressif, [Çevrimiçi]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf. [Erişildi: Mayıs 2022].
- [22] K. GÜNSAY, «MİKRODENETLEYİCİLİ ENDÜSTRİYEL SERİ PROTOKOL ÇÖZÜMLEYİCİ SİSTEMİNİN PROGRAMI,» *İstanbul*, 2009.
- [23] «CP2102 Datasheet,» Silicon Laboratories, [Çevrimiçi]. Available: <https://www.waveshare.com/w/upload/8/81/CP2102.pdf>. [Erişildi: Mayıs 2022].
- [24] «TC1262 Datasheet,» MICROCHIP, [Çevrimiçi]. Available: <file:///C:/Users/MURAT/Desktop/entegreler/datasheetler/tc1262.pdf>. [Erişildi: Mayıs 2022].
- [25] M. M. M. M. Kimmig Markus, «Querving source code with natural language,» %1 içinde *International Conference on Automated Software Engineering*, USA, 2011.

- [26] "Interaktives Programmieren als Systems-Schlager". www.computerwoche.de (Almanca). [Erişildi: Mayıs 2022].
- [27] "TOP IDE Top Integrated Development Environment index" pypl.github.io (İngilizce), 28 Ocak 2015 [Erişildi: 26 Mayıs 2022].
- [28] "Updated: Arduino announces FPGA board, ATmega4809 in Uno Wi-Fi mk2, cloud-based IDE and IoT hardware". Electronics Weekly (İngilizce). 18 Mayıs 2018. [Erişildi: Mayıs 2022].
- [29] Banzi Shiloh, Massimo Michael. Getting started with Arduino. Sebastopol, CA. ISBN 9781449363314.
- [30] «WiFi Kütüphanesi,» Arduino, [Çevrimiçi]. Available: <https://www.arduino.cc/reference/en/libraries/wifi/>. [Erişildi: Mayıs 2022].
- [31] «Http Client Kütüphanesi,» github, [Çevrimiçi]. Available: <https://github.com/amcewen/HttpClient>. [Erişildi: Mayıs 2022].
- [32] «u8g2 Kütüphanesi,» github, [Çevrimiçi]. Available: <https://github.com/olikraus/u8g2>. [Erişildi: Mayıs 2022].
- [33] «SolidWorks» [Çevrimiçi]. Available: <https://en.wikipedia.org/wiki/SolidWorks>.

MALİYET TABLOSU

Çizelge 5.1 Makine ve Teçhizat Giderleri

Makine ve Teçhizat Giderleri			
Adı / Modeli	Alım Türü	Kullanım Gerekçesi	Bedeli (TL)
ESP32-WROOM-32U Geliştirme Modülü	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Geliştirme kartı	118
ESP32-WROOM-32U	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Mikrodenetleyici	87
1.3 inch SH1106 OLED Grafik Ekran	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Led ekran modülü	111
MAX30100 Kalp Nabız Sensörü	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Nabız sensörü	46
MPU6050 6 Eksen Gyro ve İvme Sensörü	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Hareket sensörü	39
3.7 V 1S Lipo Batarya-Pil 450 mAh	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Batarya	58
CP2102 Programlayıcı	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Programlayıcı Kart	102
DS1042 Buton	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Buton	9
TC1262 3.3V Regülatör	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Regülatör	45
TP4056 Lipo Pil Şarj Devresi	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Batarya şarj cihazı	12
SMD Direnç	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Devre Gerçeklemesi	22
SMD Kondansatör	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Devre Gerçeklemesi	32
SMD Transistör	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Devre Gerçeklemesi	3.5
Header	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Pin Girişleri	18
Kordon	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Dış Tasarım	62
TOPLAM MALİYET			764.5

Çizelge 5.2 Sarf Giderleri

Sarf Giderleri			
Adı	Alım Türü	Kullanım Gerekçesi	Bedeli (TL)
Devre kartını bastırmak	<input type="checkbox"/> Yurt İçi <input checked="" type="checkbox"/> Yurt Dışı	Tasarlanan devre kartını gerçeklemek	240
Lehim makinesi, lehim	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Baskılı devre kartını entegrelerle birleştirmek	50
3D Baskı	<input checked="" type="checkbox"/> Yurt İçi <input type="checkbox"/> Yurt Dışı	Akıllı bilekliğin dış tasarımını yapmak	70
TOPLAM =			360

KATALOG BİLGİLERİ

ESP32-WROOM-32D Wifi Modül

- Çip: ESPG2-DOWD
- Desteklenen Protokol: 802.11 b / g / n
- WiFi Frekansı:2.4GHz- 2.5GHz
- Çalışma Gerilimi / Besleme Gerilimi:2.7V- 3.6V
- Kristal: 40Mhz
- Yerleşik anten
- Boyutlar: 18 mm x 25.5 mm x 3.1 mm
- Veri Hızı: 150 Mb / s
- Çıkış Gücü: 20 dBm
- Ortalama Çalışma Akımı: 80mA
- Minimum Tedarik Akımı: 500mA
- Minimum Çalışma Sıcaklığı: -40 C
- Maksimum Çalışma Sıcaklığı: +85 C
- Entegre SPI Flash: 4 MB

SH1106 OLED Grafik Ekran

- Çözünürlük: 128x64
- Ekran Boyutu: 3.3cm (1.3 inç)
- Pin Sayısı: 4
- I2C Arayüzü Desteği
- Çalışma Gerilimi: 3V – 5V
- Minimum Çalışma Sıcaklığı: -40 C
- Maksimum Çalışma Sıcaklığı: +70 C
- Boyutlar: 35.4 mm x 33.5 mm
- Ağırlık: 5.7g

MAX30100 Kalp Nabız Sensörü

- Boyutlar: 5.6 mm x 2.8 mm x 1.2 mm
- Çalışma Gerilimi: 1.8V- 3.3V
- Minimum Çalışma Sıcaklığı: -40 C
- Maksimum Çalışma Sıcaklığı: +85 C
- Optik sensör: IR ve kırmızı LED, Fotodedektör
- I2C Arayüzü Desteği

MPU6050 6 Eksen İvme ve Gyro Sensörü

- Boyutlar: 20.3 mm X 15.6 mm
- Çalışma gerilimi: 3V- 5V
- Gyro ölçüm aralığı: + 250 500 1000 2000 ° / s

- Açısal ivme ölçer ölçüm aralığı: $\pm 2 \pm 4 \pm 8 \pm 16$ g
- I2C Arayüzü Desteği
- Dahili 16 bit AD dönüştürücü, 16 bit veri çıkışı

3.7V 450 mAh Li-Polymer Pil

- Kimya: Lityum Polimer
- Voltaj: 3.7V
- Kapasite: 450 mAh
- Boyutlar: 30 mm x 30 mm x 5 mm

TP4056 Lipo Pil Şarj Devresi

- Micro USB girişi
- Kırmızı ve Mavi LED göstergeler
- Giriş Voltajı: 4.5V- 5.5V
- Değiştirilebilir Akım Değeri: 1.2A- 0.13A
- Çıkış Gerilimi: 2.9V- 4.2V
- Çıkış Gerilimi Hassasiyeti: %1,5
- Şarj Akımı: 1A
- Boyutlar: 25 mm x 19 mm x 10 mm

TC1262 3.3V Regülatör

- Giriş Voltajı: 2.7V- 6V
- Çıkış Gerilimi: 3.3V
- Çıkış Akımı: 500mA
- Paket Tipi: 3-Pin SOT-223
- Minimum Çalışma Sıcaklığı: -65 C
- Maksimum Çalışma Sıcaklığı: +150 C
- Boyutlar: 7.3 mm x 6.7 mm x 1.8 mm

DS1042 Buton

- Boyutlar: 6 mm x 6 mm x 7.3 mm
- Minimum Çalışma Sıcaklığı: -25 C
- Maksimum Çalışma Sıcaklığı: +75 C
- Maksimum Çalışma Gerilimi: 12V
- Ortalama Çalışma Akımı: 50mA

CP2102 Programlayıcı

- Çalışma Gerilimi: 3.3V- 5V
- USB 2.0 Full-Speed Destekli (12 Mbps)
- TX, RX ve güç LED'leri
- Minimum Çalışma Sıcaklığı: -40 C
- Maksimum Çalışma Sıcaklığı: +80 C
- Mac, Linux, Android, WinCE, Windows 7/8/8.1/10 uyumlu
- USB-A Bağlantı

```

#include <Arduino.h>
#include <U8g2lib.h>
#include <HTTPClient.h>
#include <Wire.h>
#include <WiFi.h>
#include "MAX30100_PulseOximeter.h"
#define MPU6050_ACCEL_XOUT_H 0x3B
#define MPU6050_PWR_MGMT_1 0x6B
#define MPU6050_PWR_MGMT_2 0x6C
#define MPU6050_WHO_AM_I 0x75
#define MPU6050_I2C_ADDRESS 0x68
#define SWAP(x,y) swap = x; x = y; y = swap

//U8G2_SH1106_128X32_VISIONOX_F_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8X8_PIN_NONE);
U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8X8_PIN_NONE);
//U8G2_SH1106_128X64_NONAME_HW_I2C u8g2(U8G2_R0, /* reset=*/
U8X8_PIN_NONE);

const char *ssid = "DESKTOP-14JK4CN";
const char *pass = "ibrahimibrahim";
char ssidtel[] = "Geme4";
const char* passtel = "ibrahim123";

#define REPORTING_PERIOD_MS 1000
#define WIFI_TIMEOUT_MS 20000

typedef union accel_t_gyro_union
{
    struct
    {
        uint8_t x_accel_h;
        uint8_t x_accel_l;
        uint8_t y_accel_h;
        uint8_t y_accel_l;
        uint8_t z_accel_h;
        uint8_t z_accel_l;
        uint8_t t_h;
        uint8_t t_l;
        uint8_t x_gyro_h;
        uint8_t x_gyro_l;
        uint8_t y_gyro_h;
        uint8_t y_gyro_l;
        uint8_t z_gyro_h;
        uint8_t z_gyro_l;
    } reg;
    struct
    {
        int16_t x_accel;
        int16_t y_accel;
        int16_t z_accel;
        int16_t temperature;
        int16_t x_gyro;
    }

```

```

        int16_t y_gyro;
        int16_t z_gyro;
    } value;
};

unsigned long last_read_time;
float        last_x_angle; // These are the filtered angles
float        last_y_angle;
float        last_z_angle;
float        last_gyro_x_angle; // Store the gyro angles to compare
drift
float        last_gyro_y_angle;
float        last_gyro_z_angle;

void set_last_read_angle_data(unsigned long time, float x, float y,
float z, float x_gyro, float y_gyro, float z_gyro) {
    last_read_time = time;
    last_x_angle = x;
    last_y_angle = y;
    last_z_angle = z;
    last_gyro_x_angle = x_gyro;
    last_gyro_y_angle = y_gyro;
    last_gyro_z_angle = z_gyro;
}

inline unsigned long get_last_time() {return last_read_time;}
inline float get_last_x_angle() {return last_x_angle;}
inline float get_last_y_angle() {return last_y_angle;}
inline float get_last_z_angle() {return last_z_angle;}
inline float get_last_gyro_x_angle() {return last_gyro_x_angle;}
inline float get_last_gyro_y_angle() {return last_gyro_y_angle;}
inline float get_last_gyro_z_angle() {return last_gyro_z_angle;}

float        base_x_accel;
float        base_y_accel;
float        base_z_accel;
float        base_x_gyro;
float        base_y_gyro;
float        base_z_gyro;

int read_gyro_accel_vals(uint8_t* accel_t_gyro_ptr) {

    accel_t_gyro_union* accel_t_gyro = (accel_t_gyro_union *)
    accel_t_gyro_ptr;

    int error = MPU6050_read (MPU6050_ACCEL_XOUT_H, (uint8_t *)
    accel_t_gyro, sizeof(*accel_t_gyro));

    uint8_t swap;

    SWAP ((*accel_t_gyro).reg.x_accel_h, (*accel_t_gyro).reg.x_accel_l);
    SWAP ((*accel_t_gyro).reg.y_accel_h, (*accel_t_gyro).reg.y_accel_l);
    SWAP ((*accel_t_gyro).reg.z_accel_h, (*accel_t_gyro).reg.z_accel_l);
    SWAP ((*accel_t_gyro).reg.t_h, (*accel_t_gyro).reg.t_l);
    SWAP ((*accel_t_gyro).reg.x_gyro_h, (*accel_t_gyro).reg.x_gyro_l);
    SWAP ((*accel_t_gyro).reg.y_gyro_h, (*accel_t_gyro).reg.y_gyro_l);
    SWAP ((*accel_t_gyro).reg.z_gyro_h, (*accel_t_gyro).reg.z_gyro_l);

    return error;
}

```

```

void calibrate_sensors() {
    int                num_readings = 10;
    float              x_accel = 0;
    float              y_accel = 0;
    float              z_accel = 0;
    float              x_gyro = 0;
    float              y_gyro = 0;
    float              z_gyro = 0;
    accel_t_gyro_union accel_t_gyro;

    read_gyro_accel_vals((uint8_t *) &accel_t_gyro);

    for (int i = 0; i < num_readings; i++) {
        read_gyro_accel_vals((uint8_t *) &accel_t_gyro);
        x_accel += accel_t_gyro.value.x_accel;
        y_accel += accel_t_gyro.value.y_accel;
        z_accel += accel_t_gyro.value.z_accel;
        x_gyro += accel_t_gyro.value.x_gyro;
        y_gyro += accel_t_gyro.value.y_gyro;
        z_gyro += accel_t_gyro.value.z_gyro;
        delay(100);
    }
    x_accel /= num_readings;
    y_accel /= num_readings;
    z_accel /= num_readings;
    x_gyro /= num_readings;
    y_gyro /= num_readings;
    z_gyro /= num_readings;

    base_x_accel = x_accel;
    base_y_accel = y_accel;
    base_z_accel = z_accel;
    base_x_gyro = x_gyro;
    base_y_gyro = y_gyro;
    base_z_gyro = z_gyro;
}

const unsigned char borsalogo [] PROGMEM = {
    0xFF, 0xFF, 0xBF, 0xD9, 0xB6, 0x03, 0xFF, 0xFF, 0xF7, 0xBF, 0xDF,
    0x01,
    0xFF, 0xFF, 0xFF, 0xFB, 0xD6, 0x01, 0xFF, 0x3F, 0xE0, 0xDF, 0x7F,
    0x03,
    0xFF, 0x3F, 0xA0, 0xFE, 0x5F, 0x01, 0xFF, 0x3F, 0xE0, 0x97, 0xF9,
    0x03,
    0xFF, 0x7F, 0xF0, 0xFF, 0xFB, 0x03, 0xFF, 0x3F, 0xE0, 0xCD, 0x9F,
    0x02,
    0xFF, 0x7F, 0x70, 0xC0, 0xF6, 0x03, 0xFF, 0x3F, 0x70, 0xE0, 0x7F,
    0x02,
    0xFF, 0x3F, 0xF0, 0xC0, 0x2F, 0x02, 0xFF, 0x3F, 0x70, 0x60, 0x1D,
    0x02,
    0xFF, 0x3F, 0x60, 0xC0, 0x07, 0x03, 0xBF, 0x3F, 0xF0, 0xC0, 0x00,
    0x02,
    0xBF, 0x78, 0x70, 0xE0, 0x00, 0x02, 0x3F, 0x3A, 0x70, 0xC0, 0x01,
    0x03,
    0x1F, 0x38, 0x70, 0xC0, 0x03, 0x03, 0x3F, 0x38, 0xF0, 0xC0, 0x01,
    0x03,
    0x3F, 0x38, 0x70, 0xC0, 0x41, 0x02, 0x7F, 0x7C, 0x70, 0xE0, 0xC8,
    0x03,
    0x3F, 0x78, 0xE0, 0x60, 0xB2, 0x03, 0x7F, 0x38, 0xF0, 0x70, 0xFF,
    0x03,

```



```

    0x1F, 0x38, 0x70, 0x30, 0xFC, 0x03, 0x1F, 0x38, 0xF0, 0x88, 0xFD,
0x01,
    0x1F, 0x38, 0x70, 0xC0, 0x7E, 0x03, 0x3F, 0x30, 0xF0, 0x02, 0xFF,
0x03,
    0x1F, 0x78, 0xF0, 0xC1, 0xEF, 0x03, 0x1F, 0x30, 0x70, 0x90, 0xFF,
0x03,
    0x1F, 0x38, 0x10, 0xC0, 0xFB, 0x03, 0x3F, 0x38, 0x0E, 0xF0, 0xFF,
0x03,
    0xBF, 0xFC, 0x01, 0xF0, 0xFF, 0x03, 0xFF, 0x3F, 0x00, 0xFC, 0x7F,
0x03,
    0x0F, 0x00, 0x10, 0xE7, 0xFF, 0x03, 0x27, 0x04, 0xC4, 0xF1, 0xFF,
0x03,
    0x07, 0x45, 0xFA, 0xFE, 0xFF, 0x03, 0x07, 0x35, 0xFE, 0xF4, 0xFF,
0x03,
    0xDF, 0xFF, 0xFF, 0xF6, 0xFF, 0x03, 0xFF, 0xFF, 0xFF, 0xEE, 0xFF,
0x03,
    0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x03,
    0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x03,
};

```

```

const unsigned char ayarlarlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00,
0x00,
    0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x00,
0x00,
    0x00, 0x00, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x01, 0x00,
0x00,
    0x00, 0x07, 0xFE, 0x81, 0x03, 0x00, 0x80, 0x1F, 0xFF, 0xE1, 0x07,
0x00,
    0xC0, 0xFF, 0xFF, 0xFF, 0x0F, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0x0F,
0x00,
    0xC0, 0xFF, 0xFF, 0xFF, 0x0F, 0x00, 0x80, 0xFF, 0xFF, 0xFF, 0x07,
0x00,
    0x80, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x03,
0x00,
    0x00, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x00, 0xFF, 0x87, 0xFF, 0x03,
0x00,
    0x80, 0xFF, 0x01, 0xFE, 0x07, 0x00, 0xE0, 0xFF, 0x00, 0xFC, 0x1F,
0x00,
    0xFC, 0xFF, 0x00, 0xFC, 0xFF, 0x00, 0xFE, 0x7F, 0x00, 0xF8, 0xFF,
0x01,
    0xFC, 0x7F, 0x00, 0xF8, 0xFF, 0x00, 0xFC, 0x7F, 0x00, 0xF8, 0xFF,
0x00,
    0xFC, 0x7F, 0x00, 0xF8, 0xFF, 0x01, 0xFC, 0xFF, 0x00, 0xFC, 0xFF,
0x00,
    0xF0, 0xFF, 0x00, 0xFC, 0x1F, 0x00, 0x00, 0xFF, 0x01, 0xFE, 0x03,
0x00,
    0x00, 0xFF, 0x87, 0xFF, 0x03, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x03,
0x00,
    0x00, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x80, 0xFF, 0xFF, 0xFF, 0x07,
0x00,
    0x80, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0x0F,
0x00,
    0xC0, 0xFF, 0xFF, 0xFF, 0x1F, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0x0F,
0x00,
    0x80, 0x1F, 0xFE, 0xE1, 0x07, 0x00, 0x00, 0x07, 0xFE, 0x81, 0x03,
0x00,
    0x00, 0x00, 0xFE, 0x01, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x00, 0x00,
0x00,
    0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x00,

```

```

0x00,
    0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,    };

```

```

const unsigned char alarmlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x58, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x48, 0x00, 0x00, 0x00, 0x00, 0x70, 0x78, 0x38, 0x00,
0x00,
    0x00, 0xEC, 0x31, 0xEE, 0x00, 0x00, 0x00, 0x86, 0x79, 0x86, 0x01,
0x00,
    0x00, 0xE3, 0xEF, 0x1B, 0x03, 0x00, 0x00, 0xF1, 0x11, 0x3E, 0x03,
0x00,
    0x00, 0x7B, 0xFE, 0x7B, 0x02, 0x00, 0x00, 0xA9, 0x13, 0x77, 0x02,
0x00,
    0x00, 0xD7, 0xB2, 0xED, 0x03, 0x00, 0x00, 0x5E, 0x06, 0xD8, 0x01,
0x00,
    0x00, 0xE8, 0x10, 0x5C, 0x00, 0x00, 0x00, 0xB8, 0x30, 0xD4, 0x00,
0x00,
    0x00, 0x14, 0x30, 0xA0, 0x00, 0x00, 0x00, 0x54, 0x70, 0xB0, 0x00,
0x00,
    0x00, 0xFC, 0xD8, 0xFC, 0x00, 0x00, 0x00, 0x14, 0x30, 0xA0, 0x00,
0x00,
    0x00, 0x2C, 0x00, 0xB4, 0x00, 0x00, 0x00, 0xE8, 0x01, 0x5E, 0x00,
0x00,
    0x00, 0x68, 0x00, 0x78, 0x00, 0x00, 0x00, 0xD0, 0x86, 0x29, 0x00,
0x00,
    0x00, 0xB0, 0x31, 0x37, 0x00, 0x00, 0x00, 0x60, 0xA7, 0x1B, 0x00,
0x00,
    0x00, 0xC0, 0xFC, 0x0C, 0x00, 0x00, 0x00, 0x80, 0x87, 0x03, 0x00,
0x00,
    0x00, 0x00, 0xFD, 0x06, 0x00, 0x00, 0x00, 0x80, 0x00, 0x04, 0x00,
0x00,
    0x00, 0xE0, 0x01, 0x1C, 0x00, 0x00, 0x00, 0xE0, 0x01, 0x1F, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,    };

```

```

const unsigned char nabizlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x81, 0x01,
0x00,
    0x00, 0xF0, 0xC3, 0x0F, 0x00, 0x00, 0xF8, 0xFF, 0x1F, 0x00, 0x00,

```

```

0xF8,
    0xBF, 0x1F, 0x00, 0x00, 0xFC, 0xBF, 0x3F, 0x00, 0x00, 0xFC, 0xB3,
0x3F,
    0x00, 0x00, 0xFC, 0x1B, 0x3F, 0x00, 0x00, 0xFC, 0x51, 0x3F, 0x00,
0x00,
    0x08, 0x56, 0x00, 0x00, 0x00, 0xF8, 0x47, 0xFC, 0x03, 0x00, 0xF0,
0xEF,
    0x0E, 0x00, 0x00, 0xF0, 0xCF, 0x0E, 0x00, 0x00, 0xE0, 0xEF, 0x07,
0x00,
    0x00, 0xC0, 0xFF, 0x03, 0x00, 0x00, 0x00, 0xFF, 0x01, 0x00, 0x00,
0x00,
    0xFF, 0x00, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,    };

```

```

const unsigned char oksijenlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x01, 0x00, 0x00,
0x00,
    0xFE, 0x07, 0x00, 0x00, 0x00, 0xAF, 0x0F, 0x00, 0x00, 0x00, 0x07,
0x0E,
    0x00, 0x00, 0x80, 0x03, 0x1E, 0x00, 0x00, 0xC0, 0x03, 0x3C, 0x00,
0x00,
    0xC0, 0x01, 0x3C, 0x00, 0x00, 0xC0, 0x01, 0x38, 0x00, 0x00, 0xC0,
0x01,
    0x38, 0x00, 0x00, 0xC0, 0x01, 0x38, 0x00, 0x00, 0xE0, 0x01, 0x38,
0x00,
    0x00, 0xC0, 0x01, 0x38, 0x00, 0x00, 0xC0, 0x01, 0x3C, 0x00, 0x00,
0xC0,
    0x01, 0x9C, 0x07, 0x00, 0xC0, 0x03, 0xDC, 0x1F, 0x00, 0x80, 0x07,
0x1E,
    0x1E, 0x00, 0x80, 0x07, 0x0E, 0x38, 0x00, 0x00, 0xBF, 0x0F, 0x18,
0x00,
    0x00, 0xFE, 0x03, 0x18, 0x00, 0x00, 0xF8, 0x00, 0x0C, 0x00, 0x00,
0x00,
    0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0xC0,
0x3F,
    0x00, 0x00, 0x00, 0xC0, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00,    };

```

```

const unsigned char hakkindalogo [] PROGMEM = {
    0x00, 0x00, 0xFF, 0x01, 0x00, 0x00, 0x00, 0xE0, 0xFF, 0x1F, 0x00,
0x00,
    0x00, 0xF8, 0xFF, 0x7F, 0x00, 0x00, 0x00, 0xFE, 0xFF, 0xFF, 0x01,
0x00,
    0x00, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x80, 0xFF, 0xEF, 0xFF, 0x07,

```

```

0x00,
    0xC0, 0xFF, 0x83, 0xFF, 0x0F, 0x00, 0xE0, 0xFF, 0x03, 0xFF, 0x1F,
0x00,
    0xF0, 0xFF, 0x01, 0xFE, 0x3F, 0x00, 0xF0, 0xFF, 0x01, 0xFE, 0x7F,
0x00,
    0xF8, 0xFF, 0x01, 0xFE, 0x7F, 0x00, 0xFC, 0xFF, 0x03, 0xFF, 0xFF,
0x00,
    0xFC, 0xFF, 0x87, 0xFF, 0xFF, 0x00, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF,
0x01,
    0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0x01, 0xFE, 0xFF, 0x1F, 0xFF, 0xFF,
0x01,
    0xFF, 0xFF, 0x07, 0xFE, 0xFF, 0x01, 0xFF, 0xFF, 0x03, 0xFE, 0xFF,
0x03,
    0xFF, 0xFF, 0x01, 0xFE, 0xFF, 0x03, 0xFF, 0xFF, 0x00, 0xFE, 0xFF,
0x03,
    0xFF, 0x7F, 0x00, 0xFE, 0xFF, 0x03, 0xFF, 0xFF, 0x07, 0xFE, 0xFF,
0x03,
    0xFF, 0xFF, 0x07, 0xFE, 0xFF, 0x03, 0xFF, 0xFF, 0x07, 0xFE, 0xFF,
0x03,
    0xFF, 0xFF, 0x07, 0xFE, 0xFF, 0x03, 0xFE, 0xFF, 0x07, 0xFE, 0xFF,
0x01,
    0xFE, 0xFF, 0x07, 0xFE, 0xFF, 0x01, 0xFE, 0xFF, 0x07, 0xFE, 0xFF,
0x01,
    0xFE, 0xFF, 0x07, 0xFE, 0xFF, 0x01, 0xFC, 0xFF, 0x07, 0xFE, 0xFF,
0x00,
    0xFC, 0xFF, 0x06, 0xF6, 0xFF, 0x00, 0xF8, 0x7F, 0x00, 0xE0, 0x7F,
0x00,
    0xF8, 0x7F, 0x00, 0xF0, 0x7F, 0x00, 0xF0, 0x7F, 0x00, 0xE0, 0x3F,
0x00,
    0xE0, 0xFF, 0xA9, 0xF6, 0x1F, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0x0F,
0x00,
    0x80, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0x00, 0xFF, 0xFF, 0xFF, 0x03,
0x00,
    0x00, 0xFE, 0xFF, 0xFF, 0x01, 0x00, 0x00, 0xF8, 0xFF, 0x7F, 0x00,
0x00,
    0x00, 0xE0, 0xFF, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFE, 0x03, 0x00,
0x00,
};

```

```

const unsigned char alarmlogoanaekran [] PROGMEM = {
    0x48, 0x00, 0xFC, 0x00, 0x58, 0x00, 0x40, 0x00, 0x04, 0x00, 0xA4,
0x00,
    0x80, 0x00, 0x08, 0x00, 0x48, 0x00, 0x30, 0x00, };

```

```

const unsigned char wifilogo [] PROGMEM = {
    0x00, 0x00, 0x38, 0x00, 0xE8, 0x00, 0xB4, 0x00, 0x58, 0x00, 0x78,
0x00,
    0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, };

```

```

const unsigned char adimlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0xE0, 0x01, 0x00, 0x00, 0xF0, 0x01, 0x00, 0x00, 0xF0,
0x03,
    0x00, 0x80, 0xFD, 0x01, 0x00, 0xC0, 0xDF, 0x06, 0x00, 0xE0, 0x3F,
0x06,

```

```

    0x00, 0x70, 0x7F, 0x07, 0x00, 0xB8, 0xFF, 0x03, 0x00, 0x98, 0xCF,
0x01,
    0x00, 0xC8, 0xC7, 0x00, 0x00, 0xE0, 0x03, 0x00, 0x00, 0xF0, 0x07,
0x00,
    0x00, 0xF0, 0x07, 0x00, 0x00, 0xF8, 0x0F, 0x00, 0xE0, 0x3F, 0x1F,
0x00,
    0xE0, 0x3F, 0x1E, 0x00, 0xE0, 0x1F, 0x1C, 0x00, 0x00, 0x00, 0x0C,
0x00,
    0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x0E,
0x00,
    0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x06,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, };
```

```

const unsigned char sicakliklogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF8, 0x00, 0x00,
0x00,
    0x00, 0x00, 0xFC, 0x03, 0x00, 0x00, 0x00, 0x00, 0xDC, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x8E, 0x03, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x0E, 0x07, 0x00, 0x00, 0x00, 0x00, 0x0E, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x6C, 0x03, 0x00, 0x00, 0x00, 0x00, 0x6E, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x6E, 0x07, 0x00, 0x00, 0x00, 0x00, 0x2E, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x6E, 0x07, 0x00, 0x00, 0x00, 0x00, 0x66, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x6E, 0x03, 0x00, 0x00, 0x00, 0x00, 0x2E, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x6E, 0x03, 0x00, 0x00, 0x00, 0x00, 0x6C, 0x03, 0x00,
0x00,
    0x00, 0x00, 0x6E, 0x03, 0x00, 0x00, 0x00, 0x00, 0x2E, 0x07, 0x00,
0x00,
    0x00, 0x00, 0x6F, 0x0F, 0x00, 0x00, 0x00, 0x80, 0x63, 0x0E, 0x00,
0x00,
    0x00, 0x80, 0xFB, 0x1C, 0x00, 0x00, 0x00, 0xC0, 0xF9, 0x39, 0x00,
0x00,
    0x00, 0xC0, 0xFD, 0x3B, 0x00, 0x00, 0x00, 0xC0, 0xFC, 0x33, 0x00,
0x00,
    0x00, 0xC0, 0xFC, 0x3B, 0x00, 0x00, 0x00, 0xC0, 0xFD, 0x3B, 0x00,
0x00,
    0x00, 0xC0, 0xF9, 0x19, 0x00, 0x00, 0x00, 0x80, 0x73, 0x1C, 0x00,
0x00,
    0x00, 0x80, 0x07, 0x0E, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x0F, 0x00,
0x00,
    0x00, 0x00, 0xFE, 0x03, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x01, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```



```

    0x00, 0x00, 0xF0, 0x81, 0x1F, 0x00, 0xFE, 0xFF, 0xFF, 0x7F, 0x00,
0xF8,
    0x81, 0x0F, 0x00, 0x00, 0x00, 0x00, 0xF8, 0xC0, 0x0F, 0x00, 0xFF,
0xFF,
    0xFF, 0xFF, 0x00, 0xF0, 0x03, 0x1F, 0x00, 0x00, 0x00, 0x00, 0xFC,
0xE0,
    0x07, 0x80, 0xFF, 0xFF, 0xFF, 0xFF, 0x01, 0xC0, 0x07, 0x3F, 0x00,
0x00,
    0x00, 0x00, 0x7E, 0xF0, 0x03, 0x80, 0xFF, 0xFF, 0xFF, 0xFF, 0x01,
0x80,
    0x0F, 0x3E, 0x00, 0x00, 0x00, 0x00, 0x3E, 0xF8, 0x00, 0xC0, 0xFF,
0xFF,
    0xFF, 0xFF, 0x03, 0x80, 0x1F, 0x7C, 0x00, 0x00, 0x00, 0x00, 0x1F,
0xFC,
    0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x1F, 0xF8, 0x00,
0x00,
    0x00, 0x00, 0x1F, 0x7C, 0x00, 0x80, 0xFF, 0xFF, 0xFF, 0xFF, 0x01,
0x00,
    0x3E, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x1F, 0x7C, 0x00, 0x80, 0xFF,
0xFF,
    0xFF, 0xFF, 0x01, 0x00, 0x7E, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x1F,
0x3E,
    0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x7C, 0xF0, 0x01,
0x00,
    0x00, 0x80, 0x0F, 0x3E, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0xFF, 0x03,
0x00,
    0x7C, 0xF0, 0x01, 0x00, 0x00, 0x80, 0x0F, 0x3E, 0x00, 0x80, 0xFF,
0xFF,
    0xFF, 0xFF, 0x01, 0x00, 0x7C, 0xF0, 0x01, 0x00, 0x00, 0x80, 0x0F,
0x3E,
    0x00, 0x80, 0xFF, 0xFF, 0xFF, 0xFF, 0x03, 0x00, 0x7C, 0xF0, 0x01,
0x00,
    0x00, 0x80, 0x0F, 0x3E, 0x00, 0xC0, 0xFF, 0xFF, 0xFF, 0xFF, 0x01,
0x00,
    0x7C, 0xF0, 0x01, 0x00, 0x00, 0x00, 0x1F, 0x3E, 0x00, 0x80, 0xFF,
0xFF,
    0xFF, 0xFF, 0x03, 0x00, 0x7C, 0xF0, 0x01, 0x00, 0x00, 0x80, 0x0F,
0x3E,
    0x00, 0xE0, 0xFF, 0xFF, 0xFF, 0xFF, 0x07, 0x00, 0x7C, 0xF0, 0x00,
0x00,
    0x00, 0x00, 0x1F, 0x7C, 0x00, 0xE0, 0xFF, 0xFF, 0xFF, 0xFF, 0x07,
0x00,
    0x3E, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x1E, 0x7C, 0x00, 0xF0, 0xFF,
0xFF,
    0xFF, 0xFF, 0x0F, 0x00, 0x3E, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x3E,
0xF8,
    0x00, 0xF8, 0xFF, 0xFF, 0xFF, 0xFF, 0x1F, 0x00, 0x1F, 0x7C, 0x00,
0x00,
    0x00, 0x00, 0x7E, 0xF8, 0x01, 0xFC, 0xFF, 0xFF, 0xFF, 0xFF, 0x3F,
0x80,
    0x1F, 0x3E, 0x00, 0x00, 0x00, 0x00, 0xFC, 0xE0, 0x03, 0xFE, 0xFF,
0xFF,
    0xFF, 0xFF, 0x7F, 0xC0, 0x0F, 0x3E, 0x00, 0x00, 0x00, 0x00, 0xF8,
0xE0,
    0x07, 0x66, 0x59, 0xFA, 0x9F, 0x59, 0x66, 0xE0, 0x87, 0x1F, 0x00,
0x00,
    0x00, 0x00, 0xF0, 0xC3, 0x1F, 0x00, 0x00, 0xF8, 0x0F, 0x00, 0x00,
0xF0,
    0xC3, 0x0F, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x07, 0x3F, 0x00, 0x00,
0xC0,
    0x03, 0x00, 0x00, 0xFC, 0xC0, 0x07, 0x00, 0x00, 0x00, 0x00, 0xC0,

```



```

0x00,
    0xC0, 0x0C, 0x00, 0x00, 0x00, 0xC0, 0x0C, 0x00, 0x00, 0x00, 0xFE,
0x3F,
    0x00, 0x00, 0x00, 0xFE, 0xFF, 0x00, 0x00, 0x00, 0xF8, 0xFF, 0x01,
0x00,
    0x00, 0xF0, 0xE0, 0x01, 0x00, 0x00, 0xF0, 0xE0, 0x03, 0x00, 0x00,
0xF0,
    0xE0, 0x03, 0x00, 0x00, 0xF0, 0xF0, 0x01, 0x00, 0x00, 0xF0, 0xFF,
0x00,
    0x00, 0x00, 0xF0, 0xFF, 0x01, 0x00, 0x00, 0xF0, 0xFA, 0x03, 0x00,
0x00,
    0xF0, 0xE0, 0x07, 0x00, 0x00, 0xF0, 0xC0, 0x07, 0x00, 0x00, 0xF0,
0xC0,
    0x07, 0x00, 0x00, 0xF0, 0xC0, 0x07, 0x00, 0x00, 0xF0, 0xF9, 0x03,
0x00,
    0x00, 0xFE, 0xFF, 0x03, 0x00, 0x00, 0xFE, 0xFF, 0x00, 0x00, 0x00,
0xE0,
    0x0E, 0x00, 0x00, 0x00, 0xC0, 0x0C, 0x00, 0x00, 0x00, 0xE0, 0x0C,
0x00,
    0x00, 0x00, 0xC0, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    };

```

```

const unsigned char dolarlogo [] PROGMEM = {
    0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00,
0x00,
    0x0F, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0xE0, 0x7F,
0x00,
    0x00, 0x00, 0xF8, 0xFF, 0x00, 0x00, 0x00, 0xFC, 0xFF, 0x01, 0x00,
0x00,
    0xFC, 0xFF, 0x03, 0x00, 0x00, 0xFE, 0xFF, 0x07, 0x00, 0x00, 0x3F,
0xEF,
    0x07, 0x00, 0x00, 0x1F, 0x8F, 0x07, 0x00, 0x00, 0x1F, 0x8F, 0x0F,
0x00,
    0x00, 0x1F, 0x0F, 0x00, 0x00, 0x00, 0x1F, 0x0F, 0x00, 0x00, 0x00,
0xFF,
    0x0F, 0x00, 0x00, 0x00, 0xFE, 0x3F, 0x00, 0x00, 0x00, 0xFE, 0xFF,
0x00,
    0x00, 0x00, 0xFC, 0xFF, 0x03, 0x00, 0x00, 0xF8, 0xFF, 0x07, 0x00,
0x00,
    0xE0, 0xFF, 0x07, 0x00, 0x00, 0x00, 0xFF, 0x0F, 0x00, 0x00, 0x00,
0x8F,
    0x0F, 0x00, 0x00, 0x00, 0x8F, 0x0F, 0x00, 0x00, 0x0F, 0x0F, 0x0F,
0x00,
    0x00, 0x1F, 0x0F, 0x0F, 0x00, 0x00, 0x1F, 0x8F, 0x0F, 0x00, 0x00,
0xFF,
    0xFF, 0x07, 0x00, 0x00, 0xFE, 0xFF, 0x07, 0x00, 0x00, 0xFC, 0xFF,
0x03,
    0x00, 0x00, 0xFC, 0xFF, 0x01, 0x00, 0x00, 0xF0, 0xFF, 0x00, 0x00,
0x00,
    0xE0, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00,
0x0F,
    0x00, 0x00, 0x00, 0x00, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x0F, 0x00,
0x00,
    };

```

```

const unsigned char ethlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00,

```

```

    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00,
0x00,
    0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x00, 0x00, 0x00, 0x00,
0x0F,
    0x00, 0x00, 0x00, 0x00, 0x1F, 0x00, 0x00, 0x00, 0x80, 0x1D, 0x00,
0x00,
    0x00, 0x40, 0x3E, 0x00, 0x00, 0x00, 0x40, 0x7D, 0x00, 0x00, 0x00,
0xA0,
    0x7E, 0x00, 0x00, 0x00, 0x60, 0xFF, 0x00, 0x00, 0x00, 0xD0, 0xFF,
0x01,
    0x00, 0x00, 0xF8, 0xFF, 0x01, 0x00, 0x00, 0xF8, 0xFF, 0x01, 0x00,
0x00,
    0xE0, 0xFF, 0x00, 0x00, 0x00, 0xC0, 0xBF, 0x01, 0x00, 0x00, 0x90,
0xDF,
    0x00, 0x00, 0x00, 0x60, 0x66, 0x00, 0x00, 0x00, 0x80, 0x79, 0x00,
0x00,
    0x00, 0x80, 0x3D, 0x00, 0x00, 0x00, 0x80, 0x1E, 0x00, 0x00, 0x00,
0x00,
    0x1F, 0x00, 0x00, 0x00, 0x00, 0x0D, 0x00, 0x00, 0x00, 0x00, 0x06,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00,
    };

```

```

const unsigned char bist100logo [] PROGMEM = {
    0xFF, 0xFF, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F, 0xFF,
0xFF,
    0xFE, 0xFF, 0x0F, 0xFF, 0x0F, 0x80, 0xFF, 0x0F, 0xFF, 0x03, 0x00,
0xFE,
    0x0F, 0xFF, 0x00, 0x30, 0xF8, 0x0F, 0x7F, 0x00, 0x38, 0xF0, 0x0F,
0x3F,
    0x40, 0x3C, 0xE0, 0x0F, 0x1F, 0x40, 0x38, 0xC0, 0x0F, 0x0F, 0xE0,
0x18,
    0x80, 0x0F, 0x0F, 0xE0, 0x00, 0x83, 0x0F, 0x07, 0xF0, 0x81, 0x03,
0x0F,
    0x07, 0xF0, 0xC1, 0x01, 0x0F, 0x03, 0xF8, 0xE3, 0x01, 0x0E, 0x03,
0xFC,
    0xE3, 0x01, 0x0E, 0x03, 0xFC, 0xE3, 0x01, 0x0E, 0x03, 0xFE, 0xE1,
0x01,
    0x0E, 0x01, 0xFF, 0xF1, 0x03, 0x0C, 0x83, 0xFF, 0xF8, 0x01, 0x0E,
0x81,
    0x3F, 0xFE, 0x03, 0x0E, 0x83, 0x1F, 0xFF, 0x03, 0x0E, 0xC3, 0x8F,
0xFF,
    0x03, 0x0E, 0xC3, 0x0F, 0xFF, 0x03, 0x0E, 0xC3, 0x1F, 0xFE, 0x03,
0x0F,
    0xC7, 0x3F, 0xFC, 0x03, 0x0F, 0xC7, 0xFF, 0xF8, 0x03, 0x0F, 0xCF,
0xFF,
    0xF0, 0x81, 0x0F, 0x9F, 0xFF, 0xF0, 0xC1, 0x0F, 0x9F, 0xFF, 0xF1,
0xE1,
    0x0F, 0xBF, 0xFF, 0xF1, 0xE0, 0x0F, 0xFF, 0xFF, 0x70, 0xF8, 0x0F,
0xFF,
    0xFF, 0x38, 0xFC, 0x0F, 0xFF, 0x7F, 0x08, 0xFF, 0x0F, 0xFF, 0x7F,
0xEE,
    0xFF, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0xFF, 0xFF,
0x0F,
    };

```

```

const unsigned char eurologo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x01, 0x00, 0x00, 0x00, 0xFF,
    0x03,
    0x00, 0x00, 0xC0, 0xFF, 0x03, 0x00, 0x00, 0xE0, 0xFF, 0x03, 0x00,
    0x00,
    0xE0, 0xFF, 0x03, 0x00, 0x00, 0xF0, 0x8F, 0x01, 0x00, 0x00, 0xF8,
    0x03,
    0x01, 0x00, 0x00, 0xF8, 0x01, 0x00, 0x00, 0x00, 0xF8, 0x00, 0x00,
    0x00,
    0x00, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x00, 0x00,
    0xFF,
    0xFF, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0x00, 0x80, 0xFF, 0xFF,
    0x00,
    0x00, 0x00, 0x7E, 0xA4, 0x00, 0x00, 0x00, 0x7C, 0x00, 0x00, 0x00,
    0x00,
    0x7E, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x7F, 0x00, 0x00, 0x00, 0xFF,
    0x7F,
    0x00, 0x00, 0x80, 0xFF, 0x7F, 0x00, 0x00, 0x00, 0xFC, 0x08, 0x00,
    0x00,
    0x00, 0x7C, 0x00, 0x00, 0x00, 0x00, 0xFC, 0x00, 0x00, 0x00, 0x00,
    0xF8,
    0x00, 0x00, 0x00, 0x00, 0xF8, 0x01, 0x02, 0x00, 0x00, 0xF0, 0x03,
    0x03,
    0x00, 0x00, 0xF0, 0xFF, 0x03, 0x00, 0x00, 0xE0, 0xFF, 0x03, 0x00,
    0x00,
    0xC0, 0xFF, 0x03, 0x00, 0x00, 0x80, 0xFF, 0x03, 0x00, 0x00, 0x00,
    0xFF,
    0x01, 0x00, 0x00, 0x00, 0x50, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    };

const unsigned char altinlogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x10,
    0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x40, 0x10, 0x00,
    0x00,
    0x00, 0xC0, 0x10, 0x00, 0x00, 0x00, 0x80, 0x10, 0x00, 0x00, 0x00,
    0x80,
    0x01, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    0x00, 0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x18, 0x80, 0x15, 0x00,
    0x00,
    0x30, 0xE0, 0xFD, 0x0B, 0x00, 0x00, 0xF8, 0xFB, 0xFF, 0x00, 0x00,
    0xF0,
    0xF7, 0xFF, 0x00, 0x00, 0xE0, 0xF7, 0xFF, 0x01, 0x00, 0xE0, 0x4F,
    0xFF,
    0x03, 0x00, 0xC0, 0xAF, 0xE0, 0x03, 0x00, 0xC0, 0xF7, 0xDF, 0x07,
    0x00,
    0xA0, 0xF7, 0xFF, 0x03, 0x00, 0x70, 0xFB, 0xFF, 0x03, 0x00, 0x7C,
    0xFA,
    0xFF, 0x03, 0x00, 0xFF, 0xFC, 0xFF, 0x01, 0x80, 0xFF, 0xE8, 0xFF,
    0x01,
    0xE0, 0x7F, 0x3F, 0xF0, 0x00, 0x80, 0x9F, 0xFF, 0x01, 0x00, 0x78,
    0xE6,
    0x7F, 0x00, 0x00, 0xF8, 0xF9, 0x1F, 0x00, 0x00, 0xFC, 0xFB, 0x0F,
    0x00,
    0x00, 0xFE, 0xFD, 0x03, 0x00, 0x00, 0xFE, 0xFF, 0x01, 0x00, 0x00,
    0xF8,
    0x7B, 0x00, 0x00, 0x00, 0xE0, 0x3B, 0x00, 0x00, 0x00, 0x80, 0x0D,
    0x00,

```

```

    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,    };

const unsigned char gumuslogo [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x04, 0x00, 0x00,
    0x00,
    0xE0, 0x7F, 0x00, 0x00, 0x00, 0xE0, 0x7F, 0x00, 0x00, 0x00, 0x70,
    0xE0,
    0x00, 0x00, 0x00, 0x70, 0xE0, 0x00, 0x00, 0x00, 0x30, 0xC0, 0x00,
    0x00,
    0x00, 0x30, 0xC0, 0x00, 0x00, 0x00, 0xBD, 0xE9, 0x0B, 0x00, 0x00,
    0xFF,
    0xFF, 0x0F, 0x00, 0x80, 0xFF, 0xFF, 0x1F, 0x00, 0x80, 0x81, 0x1F,
    0x18,
    0x00, 0xC0, 0x01, 0x0F, 0x38, 0x00, 0xC0, 0x01, 0x0F, 0x38, 0x00,
    0xC0,
    0x00, 0x0E, 0x30, 0x00, 0xFC, 0xFF, 0xFF, 0xFF, 0x01, 0xFC, 0xFF,
    0xFF,
    0xFF, 0x03, 0x6C, 0xFF, 0xF5, 0x6F, 0x07, 0x0E, 0x7C, 0xE0, 0x03,
    0x07,
    0x06, 0x7C, 0xE0, 0x03, 0x06, 0x07, 0x38, 0xC0, 0x01, 0x0E, 0x07,
    0x38,
    0xC0, 0x01, 0x0E, 0xFF, 0xFF, 0xFF, 0xFF, 0x0F, 0xFF, 0xFF, 0xFF,
    0xFF,
    0x0F, 0x62, 0x28, 0x86, 0x62, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00,    };

const unsigned char nabizlogoanaekran [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x00, 0x40,
    0x00,
    0xE0, 0x00, 0xF8, 0x26, 0xBC, 0x3F, 0xA0, 0x05, 0x80, 0x01, 0x00,
    0x01,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,    };

const unsigned char adimlogoanaekran [] PROGMEM = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x02,
    0x00, 0x0F, 0xF0, 0x1E, 0x18, 0x10, 0x78, 0x3B, 0xE0, 0x0B, 0x00,
    0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,    };

const unsigned char tllogo [] PROGMEM = {
    0x10, 0x00, 0x30, 0x00, 0x30, 0x00, 0xB0, 0x03, 0xF0, 0x02, 0xBC,
    0x03,
    0xF6, 0x00, 0x3C, 0x00, 0x34, 0x18, 0x30, 0x0C, 0x30, 0x0C, 0x30,
    0x04,
    0x30, 0x06, 0xB0, 0x03, 0xF0, 0x01, 0x20, 0x00,    };

const String urlbist100 =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=LQ15TA
J4DNB76Y68";
const String urlbtc =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=13VEXP

```

```

DDZUW4XG84";
const String urleth =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=2JEYM4
ZX6U6R8WK6";
const String urlaltinons =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=QSJHW5
WAM9DGTD1N";
const String urlaltingram =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=K2GY2P
OTQE2QAFV3";
const String urlgumusons =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=OXKD75
KBMEH0UI9S";
const String urlgumusgram =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=IZU82F
82LG3JW4D9";
const String urleuro =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=32VUN6
5F4BTE839M";
const String urldolar =
"https://api.thingspeak.com/apps/thinghttp/send_request?api_key=2LWZF1
WRXHKS4V9T";
float bist1001;
float btc1;
float eth1;
float altinons1;
float altingram1;
float gumusons1;
float gumusgram1;
float euro1;
float dolar1;

PulseOximeter pox;
uint32_t tsLastReport = 0;
int alarmaktif =1;
int alarmSaat = 1;
int alarmDakika = 25;
int Saat = 13;
int Dakika = 49;
int Saniye = 0;
int Gun = 1;
int Ay = 5;
int Yil = 2022;
int Aygunu = 23;
int i = 0;
int k;
String gunler[]={"x","Pazartesi", "Sali", "Carsamba",
"Persembe","Cuma","Cumartesi","Pazar"};
String aylar[]={
"x","Ocak","Subat","Mart","Nisan","Mayis","Hazrn","Temmmz","Agsts",
"Eylul","Ekim","Kasım","Arlık"};
float x,y,z;
int count=0,prev=0;
int threshold=3;
int tempe;

// Değişkenler...
byte Menu = 0;
bool Flip = false;
byte Contrast = 1;
// Butonlar...

```

```

#define Left_In 23
#define Ok_In 17
#define Right_In 19
#define Geri_In 16
byte Left, Ok, Right, Geri;

void setup(void) {
    Serial.begin(115200);
    Serial.println("lasdasdas");
    //WiFi.begin(ssid, pass);
    // Butonlar Input Olarak Ayarlandı...
    Serial.println("ibo");
    xTaskCreate(saathesaplama, "saathesaplama", 1000, NULL, 1, NULL);
    Serial.println("2");
    //xTaskCreate(tempe, "sicaklik", 1000, NULL, 2, NULL);
    xTaskCreate(keepWiFiAlive, "keep wifi alive", 2000, NULL, 2, NULL);
    //xTaskCreatePinnedToCore(Task1code, "Task1", 10000, NULL, 1, &Task1, 0);
    //xTaskCreatePinnedToCore(Task2code, "Task2", 50000, NULL, 1, &Task2, 1);
    //xTaskCreate(Mpu6050, "mpu6050", 50000, NULL, 1, NULL);
    //xTaskCreate(alert, "alert", 5000, NULL, 1, NULL);
    //xTaskCreate(butonoku, "alert", 500, NULL, 1, NULL);
    pinMode(Left_In, INPUT);
    pinMode(Ok_In, INPUT);
    pinMode(Right_In, INPUT);
    pinMode(Geri_In, INPUT);
    Serial.println("mertcan");
    // Max30100 Başlatılıyor
    Serial.print("Initializing pulse oximeter..");
    if (!pox.begin()) {
        Serial.println("FAILED");
        for(;;);
    } else {
        Serial.println("SUCCESS");
    }
    pox.setOnBeatDetectedCallback(onBeatDetected);

    //MPU6050
    int error;
    uint8_t c;
    Serial.begin(115200);
    Wire.begin();
    error = MPU6050_read (MPU6050_WHO_AM_I, &c, 1);
    error = MPU6050_read (MPU6050_PWR_MGMT_2, &c, 1);
    MPU6050_write_reg (MPU6050_PWR_MGMT_1, 0);

    calibrate_sensors();
    set_last_read_angle_data(millis(), 0, 0, 0, 0, 0, 0);
    // u8g2 Başlatılıyor
    u8g2.begin();
    u8g2.setFlipMode(Flip);
    u8g2.setContrast(Contrast);
    //u8g2.setFont(u8g2_font_crox2tb_tf);
    u8g2.setFont(u8g2_font_unifont_t_symbols);
    u8g2.setFontRefHeightExtendedText();
    u8g2.setDrawColor(1);
    u8g2.setFontPosTop();
    u8g2.setFontDirection(0);
    Serial.println("3");

    // Intro
    u8g2.clearBuffer(); // Display Clear

```

```

u8g2.drawStr( 0, 0, "Bitirme Projesi");
u8g2.drawLine(0, 16, 127, 15);
u8g2.drawStr( 0, 17, "Ibrahim Ozturk");
u8g2.drawStr( 0, 32, "Mertcan Yilmaz");
u8g2.drawStr( 0, 47, "Murat Delibalta");
u8g2.sendBuffer(); // Display Update
Serial.println("4");
delay(800);
}

void loop(void) {
  delay(250);
  while(true) {
    Serial.println("scdsf");
    butonOku();
    saatgoster();
    if ( !Right ) {
      MenuekranBorsa();
    }
    if ( !Left ) {
      MenuekranHakkinda();
    }
    u8g2.sendBuffer(); // u8g2 Update
  }
}

void saatgoster() {
  u8g2.clearBuffer(); // Display Clear
  Mpu6050();
  //u8g2.drawXBMP( 15, 2, 10, 10, alarmlogoanaekran);
  //u8g2.drawXBMP( 0, 2, 10, 10, wifilogo);
  u8g2.drawXBMP( 0, 2, 16, 16, nabizlogoanaekran);
  u8g2.drawXBMP( 0, 16, 16, 16, adimlogoanaekran);
  u8g2.setFont( u8g2_font_10x20_tf);
  u8g2.setCursor( 18, 3);u8g2.print("82");
  u8g2.setCursor( 18, 19);u8g2.print(count);
  u8g2.setFont(u8g2_font_unifont_t_symbols);
  if(alarmaktif == 1){
    u8g2.drawGlyph(110, 1, 0x23f2);}
  if(WiFi.status() == WL_CONNECTED){
    u8g2.drawGlyph(1, 1, 0x2604); }
  u8g2.setFont(u8g2_font_courB24_tf);
  if ( Saat < 10 ) {
    u8g2.setCursor( 62, 10);u8g2.print("0");
    u8g2.setCursor( 82, 10);u8g2.print(Saat);}
  else {
    u8g2.setCursor( 62, 10);u8g2.print(Saat);}
  if ( Dakika < 10 ) {
    u8g2.setCursor( 62, 38);u8g2.print("0");
    u8g2.setCursor( 82, 38);u8g2.print(Dakika);}
  else {
    u8g2.setCursor( 62, 38);u8g2.print(Dakika);}
  u8g2.setFont(u8g2_font_ImpactBits_tr);
    if ( Saniye < 10 ) {
    u8g2.setCursor( 105, 47);u8g2.print("0");
    u8g2.setCursor( 114, 47);u8g2.print(Saniye);}
  else {
    u8g2.setCursor( 105, 47);u8g2.print(Saniye);}
  u8g2.setCursor( 2, 36);
  u8g2.print(Aygunu);
  u8g2.setCursor( 20, 36);

```

```

        u8g2.print(aylar[Ay]);
        u8g2.setFont(u8g2_font_helvR08_tf);
        u8g2.setCursor( 2, 51);
        u8g2.print(gunler[Gun]);
        u8g2.sendBuffer(); // u8g2 Update
    //*****
}

void saathesaplama( void * parameter )
{
    while(true) {
        Saniye++;
        if ( Saniye == 60 ) {
            Saniye=0;
            Dakika++;}
        if ( Dakika == 60 ){
            Dakika=0;
            Saat++;}
        if ( Saat == 24 ){
            Saat=0;
            Gun++; Aygunu++;}
        if ( Gun==8 ){
            Gun=1;}
        if ( Ay == 1 || Ay == 3 || Ay == 5 || Ay == 7 || Ay == 9 ||
Ay == 11 || Ay == 13 )
        { if (Aygunu == 32 ){ Aygunu = 1, Ay++;}}
        if ( Ay == 4 || Ay == 6 || Ay == 8 || Ay == 10 || Ay == 12 )
        { if (Aygunu == 31 ){ Aygunu = 1, Ay++;}}
        if ( Ay == 2 ) { if (Aygunu == 30 ) {Aygunu = 1, Ay++;}}
        if ( Ay==13 ) {Yil++, Ay=1;}
        delay(1000);
    //*****
    }}
void MenuekranBorsa(){
    delay(300);
    while(true) {
        u8g2.clearBuffer(); // Display Clear
        u8g2.drawXBMP( 43, 2, 42, 42, borsalogo);
        u8g2.setFont(u8g2_font_lucasarts_scumm_subtitle_o_tf);
        u8g2.setCursor( 37, 45); u8g2.print("Borsa");
        u8g2.sendBuffer(); // u8g2 Update
        butonOku();
        if ( !Right ) {
            nabiz();
        }
        if ( !Left ) {
            loop();
        }
        if ( !Ok ) {
            bist100();
        }
        if ( !Geri ) {
            loop();
        }
    }
}
void nabiz (){
    delay(300);
    u8g2.clearBuffer();
    while ( true ) {
        u8g2.drawXBMP( 12, 2, 40, 40, nabizlogo);
        u8g2.drawXBMP( 72, 2, 36, 36, oksijenlogo);
    }
}

```



```

u8g2.setFont(u8g2_font_crox5tb_tr);
pox.update();
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
    //u8g2.setCursor( 5, 18); u8g2.print("Nabziniz:");
    u8g2.setCursor( 10, 40); u8g2.print(pox.getHeartRate());
    //u8g2.setCursor( 5, 34); u8g2.print("bpm / SpO2:");
    u8g2.setCursor( 72, 40); u8g2.print(pox.getSpO2());
    Serial.println(pox.getHeartRate());
    Serial.println(pox.getSpO2());
    u8g2.setFont(u8g2_font_6x13_tf);
    u8g2.setCursor( 92, 40); u8g2.print("%");
    Serial.println("Beat!");
    tsLastReport = millis();
    butonOku();
    if ( !Right ) {
        adimsayar();}
    if ( !Left ) {
        MenuekranBorsa();
    }
    if ( !Geri ) {
        loop();}
    u8g2.sendBuffer();}
}

void onBeatDetected(){}

void adimsayar(){
    delay(300);
    while(true){
        Mpu6050();
        u8g2.clearBuffer();
        u8g2.drawXBMP( 16, 8, 32, 48, adimlogo);
        u8g2.setFont(u8g2_font_lucasarts_scumm_subtitle_o_tf);
        u8g2.setCursor( 65, 26); u8g2.print(count);
        u8g2.sendBuffer();
        butonOku();
        if ( !Right ) {
            sicaklik();}
        if ( !Left ) {
            nabiz();
        }
        if ( !Geri ) {
            loop();}
    }
}

void sicaklik(){
    delay(300);
    while(true){
        Mpu6050();
        u8g2.clearBuffer();
        u8g2.drawXBMP( 16, 8, 42, 42, sicakliklogo);
        u8g2.setFont(u8g2_font_lucasarts_scumm_subtitle_o_tf);
        u8g2.setCursor( 65, 26); u8g2.print(tempe);
        u8g2.sendBuffer();
        butonOku();
        if ( !Right ) {
            MenuekranAlarm();}
        if ( !Left ) {
            adimsayar();
        }
        if ( !Geri ) {
            loop();}
    }
}

```

```

    }}

void MenuekranAlarm(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        u8g2.drawXBMP( 43, 2, 42, 42, alarmlogo);
        u8g2.setFont(u8g2_font_lucasarts_scumm_subtitle_o_tf);
        u8g2.setCursor( 37, 45); u8g2.print("Alarm");
        u8g2.sendBuffer();
        butonOku();
        if ( !Ok ) {
            alarmAyari();}
        if ( !Right ) {
            MenuekranAyarlar();}
        if ( !Left ) {
            sicaklik();
        }
        if ( !Geri ) {
            loop();}
    }}

void MenuekranAyarlar(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        u8g2.drawXBMP( 43, 2, 42, 42, ayarlarlogo);
        u8g2.setFont(u8g2_font_lucasarts_scumm_subtitle_o_tf);
        u8g2.setCursor( 27, 45); u8g2.print("Ayarlar");
        u8g2.sendBuffer();
        butonOku();
        if ( !Ok ) {
            Ayarlar();}
        if ( !Right ) {
            MenuekranHakkinda();}
        if ( !Left ) {
            MenuekranAlarm();
        }
        if ( !Geri ) {
            loop();}
    }}

void MenuekranHakkinda () {
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.drawXBMP( 43, 2, 42, 42, hakkindalogo);
        u8g2.setFont(u8g2_font_lucasarts_scumm_subtitle_o_tf);
        u8g2.setCursor( 20, 45); u8g2.print("Hakkinda");
        u8g2.sendBuffer();
        butonOku();
        if ( !Ok ) {
            Hakkinda ();}
        if ( !Right ) {
            loop();}
        if ( !Left ) {
            MenuekranAyarlar();
        }
        if ( !Geri ) {
            loop();}
    }
}

```

```

}

void Hakkinda () {
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_6x13_tf);
        u8g2.setCursor( 2, 2); u8g2.print("Projeyi Tasarlayanlar:");
        u8g2.setCursor( 2, 12); u8g2.print("Mertcan YILMAZ");
        u8g2.setCursor( 2, 22); u8g2.print("Ibrahim OZTURK");
        u8g2.setCursor( 2, 32); u8g2.print("Murat DELIBALTA");
        u8g2.setCursor( 2, 42); u8g2.print("Tesekkurler");
        u8g2.setCursor( 2, 52); u8g2.print("Zehra Cam TASKIRAN");
        u8g2.sendBuffer();
        butonOku();

        if ( !Geri ) {
            MenuekranHakkinda ();}
    }
}

void alarmAyari() {
    Menu = 0;
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_crox2tb_tf);
        u8g2.setCursor( 25, 3);
        u8g2.print("Alarm Ayari");
        u8g2.drawLine(0, 16, 127, 15);
        u8g2.setCursor( 15, 17); u8g2.print("Saat:");
        u8g2.setCursor( 50, 17); u8g2.print(alarmSaat);
        u8g2.setCursor( 15, 32); u8g2.print("Dakika:");
        u8g2.setCursor( 75, 32); u8g2.print(alarmDakika);
        u8g2.setCursor( 15, 47); u8g2.print("Alarm:");
        if (alarmaktif == 1){
            u8g2.setFont(u8g2_font_unifont_t_symbols);
            u8g2.drawGlyph(75, 47, 0x2714);}
        if (alarmaktif == 0){
            u8g2.setFont(u8g2_font_unifont_t_symbols);
            u8g2.drawGlyph(75, 47, 0x2715); }
        butonOku();
        if ( !Right ) {
            Menu++;
        }
        if ( Menu >= 2 ) Menu = 2;
    }
    if ( !Left ) {
        Menu--;
    }
    if ( Menu <= 0 ) Menu = 0;}
    if ( Menu == 0 ) {
        u8g2.setCursor( 0, 17); u8g2.print(">");
    } else if ( Menu == 1 ) {
        u8g2.setCursor( 0, 32); u8g2.print(">");
    }
    }
    else if ( Menu == 2 ) {
        u8g2.setCursor( 0, 47); u8g2.print(">");
    }
    }
    u8g2.sendBuffer(); // u8g2 Update
    if ( !Ok ) {
        alarmdegis();
    }
}

```

```

    if ( !Geri ) {
        MenuekranAlarm();
    }
    u8g2.sendBuffer(); // u8g2 Update
}
}
void alarmdegis(){
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_crox2tb_tf);
        u8g2.setCursor( 25, 3);
        u8g2.print("Alarm Ayari");
        u8g2.drawLine(0, 16, 127, 15);
        u8g2.setCursor( 15, 17); u8g2.print("Saat:");
        u8g2.setCursor( 50, 17); u8g2.print(alarmSaat);
        u8g2.setCursor( 15, 32); u8g2.print("Dakika:");
        u8g2.setCursor( 75, 32); u8g2.print(alarmDakika);
        u8g2.setCursor( 15, 47); u8g2.print("Alarm:");
        if (alarmaktif == 1){
            u8g2.setFont(u8g2_font_unifont_t_symbols);
            u8g2.drawGlyph(75, 47, 0x2714);
        }
        if (alarmaktif == 0){
            u8g2.setFont(u8g2_font_unifont_t_symbols);
            u8g2.drawGlyph(75, 47, 0x2715);
        }
        if ( Menu == 0 ) {
            u8g2.setCursor( 0, 17); u8g2.print("***");
        } else if ( Menu == 1 ) {
            u8g2.setCursor( 0, 32); u8g2.print("***");
        }
        else if ( Menu == 2 ) {
            u8g2.setCursor( 0, 47); u8g2.print("***");
        }
        butonOk();
    }
    if ( !Right ) {
        if ( Menu == 0 )
            alarmSaat++;
        if ( Menu == 1 )
            alarmDakika++;
        if ( Menu == 2 )
            alarmaktif =1;
        if ( alarmSaat > 23 ) alarmSaat = 0;
        if ( alarmDakika > 59 ) alarmDakika = 0;
        delay(100);
    }
    if ( !Left ) {
        if ( Menu == 0 )
            alarmSaat--;
        if ( Menu == 1 )
            alarmDakika--;
        if ( Menu == 2 )
            alarmaktif =0;
        if ( alarmDakika < 0 ) alarmDakika = 23;
        if ( alarmSaat < 0 ) alarmSaat = 59;
        delay(100);
    }
    if ( !Geri ) {
        alarmAyari();
    }
    u8g2.sendBuffer(); // u8g2 Update
}

```

```

}
}

void Ayarlar () {
    Menu = 0;
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_crox2tb_tf);
        u8g2.setCursor( 40, 3);
        u8g2.print("Ayarlar");
        u8g2.drawLine(0, 15, 127, 15);
        u8g2.setCursor( 15, 16); u8g2.print("Saat Ayari");
        u8g2.setCursor( 15, 32); u8g2.print("Tarih Ayari");
        u8g2.setCursor( 15, 47); u8g2.print("Ekran Ayari");
        butonOk();
        if ( !Right ) {
            Menu++;
            if ( Menu > 2 ) Menu = 2;
            delay(100);
        }
        if ( !Left ) {
            Menu--;
            if ( Menu < 0 ) Menu = 0;
            delay(100);
        }
        if ( Menu == 0 ) {
            u8g2.setCursor( 0, 22); u8g2.print(">");
        } else if ( Menu == 1 ) {
            u8g2.setCursor( 0, 34); u8g2.print(">");
        } else if ( Menu == 2 ) {
            u8g2.setCursor( 0, 46); u8g2.print(">");
        }
        if ( !Ok ) {
            if ( Menu == 0 ) {
                saatAyari();
            } else if ( Menu == 1 ) {
                tarihAyari();
            } else if ( Menu == 2 ) {
                //ekranAyari();
            }
        }
        if ( !Geri ) {
            MenuekranAyarlar();
        }
        u8g2.sendBuffer(); // u8g2 Update
    }
}

void saatAyari() {
    Menu = 0;
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_crox2tb_tf);
        u8g2.setCursor( 25, 3);
        u8g2.print("Saat Ayarlari");
        u8g2.drawLine(0, 16, 127, 15);
        u8g2.setCursor( 15, 17); u8g2.print("Saat:");
        u8g2.setCursor( 50, 17); u8g2.print(Saat);
        u8g2.setCursor( 15, 32); u8g2.print("Dakika:");
    }
}

```

```

u8g2.setCursor( 75, 32); u8g2.print(Dakika);
u8g2.setCursor( 15, 47); u8g2.print("Saniye:");
u8g2.setCursor( 75, 47); u8g2.print(Saniye);
butonOku();
if ( !Right ) {
    Menu++;
    if ( Menu >= 2 ) Menu = 2;
}
if ( !Left ) {
    Menu--;
    if ( Menu <= 0 ) Menu = 0;
}
if ( Menu == 0 ) {
    u8g2.setCursor( 0, 17); u8g2.print(">");
} else if ( Menu == 1 ) {
    u8g2.setCursor( 0, 32); u8g2.print(">");
} else if ( Menu == 2 ) {
    u8g2.setCursor( 0, 47); u8g2.print(">");
}
u8g2.sendBuffer(); // u8g2 Update
if ( !Ok ) {
    saatdegis();
}
if ( !Geri ) {
    Ayarlar();
}
u8g2.sendBuffer(); // u8g2 Update
}
}

void saatdegis() {
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_crox2tb_tf);
        u8g2.setCursor( 25, 3);
        u8g2.print("Saat Ayarlari");
        u8g2.drawLine(0, 16, 127, 15);
        u8g2.setCursor( 15, 17); u8g2.print("Saat:");
        u8g2.setCursor( 50, 17); u8g2.print(Saat);
        u8g2.setCursor( 15, 32); u8g2.print("Dakika:");
        u8g2.setCursor( 75, 32); u8g2.print(Dakika);
        u8g2.setCursor( 15, 47); u8g2.print("Saniye:");
        u8g2.setCursor( 75, 47); u8g2.print(Saniye);
        if ( Menu == 0 ) {
            u8g2.setCursor( 0, 17); u8g2.print("***");
        } else if ( Menu == 1 ) {
            u8g2.setCursor( 0, 32); u8g2.print("***");
        } else if ( Menu == 2 ) {
            u8g2.setCursor( 0, 47); u8g2.print("***");
        }
    }
    butonOku();
    if ( !Right ) {

        if ( Menu == 0 )
            Saat++;
        if ( Menu == 1 )
            Dakika++;
        if ( Menu == 2 )
            Saniye++;
        if ( Saat > 23 ) Saat = 0;
    }
}

```

```

        if ( Dakika > 59 ) Dakika = 0;
        if ( Saniye > 59 ) Saniye = 0;
        delay(100);
    }
    if ( !Left ) {
        if ( Menu == 0 )
            Saat--;
        if ( Menu == 1 )
            Dakika--;
        if ( Menu == 2 )
            Saniye--;
        if ( Saat < 0 ) Saat = 23;
        if ( Dakika < 0 ) Dakika = 59;
        if ( Saniye < 0 ) Saniye = 59;
        delay(100);
    }
        if ( !Geri ) {
            saatAyari();
        }
        u8g2.sendBuffer(); // u8g2 Update
    }
}

void tarihAyari() {
    Menu = 0;
    delay(300);
    while ( true ) {
        u8g2.clearBuffer();
        u8g2.setFont(u8g2_font_crox2tb_tf);
        u8g2.setCursor( 22, 3);
        u8g2.print("Tarih Ayarlari");
        u8g2.drawLine(0, 15, 127, 15);
        u8g2.setCursor( 15, 19); u8g2.print("Gun:");
        u8g2.setCursor( 50, 19); u8g2.print(Gun);
        u8g2.setCursor( 15, 34); u8g2.print("Ay:");
        u8g2.setCursor( 50, 34); u8g2.print(Ay);
        u8g2.setCursor( 15, 49); u8g2.print("Yil:");
        u8g2.setCursor( 50, 49); u8g2.print(Yil);
        butonOku();
        if ( !Right ) {
            Menu++;
            if ( Menu >= 2 ) Menu = 2;
        }
        if ( !Left ) {
            Menu--;
            if ( Menu <= 0 ) Menu = 0;
        }
        if ( Menu == 0 ) {
            u8g2.setCursor( 0, 19); u8g2.print(">");
        } else if ( Menu == 1 ) {
            u8g2.setCursor( 0, 34); u8g2.print(">");
        } else if ( Menu == 2 ) {
            u8g2.setCursor( 0, 49); u8g2.print(">");
        }
        u8g2.sendBuffer(); // u8g2 Update
        if ( !Ok ) {
            tarihdegis();
        }
        if ( !Geri ) {
            Ayarlar();
        }
    }
}

```

```

    }
    u8g2.sendBuffer(); // u8g2 Update
  }
}

void tarihdegis() {
  delay(300);
  while ( true ) {
    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_crox2tb_tf);
    u8g2.setCursor( 22, 3);
    u8g2.print("Tarih Ayarlari");
    u8g2.drawLine(0, 15, 127, 15);
    u8g2.setCursor( 15, 19); u8g2.print("Gun:");
    u8g2.setCursor( 50, 19); u8g2.print(Gun);
    u8g2.setCursor( 15, 34); u8g2.print("Ay:");
    u8g2.setCursor( 50, 34); u8g2.print(Ay);
    u8g2.setCursor( 15, 49); u8g2.print("Yil:");
    u8g2.setCursor( 50, 49); u8g2.print(Yil);

    butonOku();

    if ( Menu == 0 ) {
      u8g2.setCursor( 0, 19); u8g2.print("***");
    } else if ( Menu == 1 ) {
      u8g2.setCursor( 0, 34); u8g2.print("***");
    } else if ( Menu == 2 ) {
      u8g2.setCursor( 0, 49); u8g2.print("***");
    }
    u8g2.sendBuffer(); // u8g2 Update

    if ( !Right ) {
      if ( Menu == 0 )
        Gun++;
      if ( Menu == 1 )
        Ay++;
      if ( Menu == 2 )
        Yil++;
      if ( Gun > 31 ) Gun = 0;
      if ( Ay > 12 ) Ay = 0;
      if ( Yil > 2099 ) Yil = 2000;
      delay(100);
    }
    if ( !Left ) {
      if ( Menu == 0 )
        Gun--;
      if ( Menu == 1 )
        Ay--;
      if ( Menu == 2 )
        Yil--;
      if ( Gun < 0 ) Gun = 31;
      if ( Ay < 0 ) Ay = 12;
      if ( Yil < 2000 ) Yil = 2099;
      delay(100);
    }
    if ( !Geri ) {
      tarihAyari();
    }
    u8g2.sendBuffer(); // u8g2 Update
  }
}

```



```

    }
}

void butonOku () {
    Left  = digitalRead(Left_In);
    Ok    = digitalRead(Ok_In);
    Right = digitalRead(Right_In);
    Geri  = digitalRead(Geri_In);
    delay(100);
}

void Mpu6050 ()
{
    int error;
    double dT;
    accel_t_gyro_union accel_t_gyro;
    error = read_gyro_accel_vals((uint8_t*) &accel_t_gyro);
    unsigned long t_now = millis();

    Serial.print(F("temperature: "));
    dT = ( (double) accel_t_gyro.value.temperature + 12412.0) / 340.0;
    tempe = dT;
    Serial.print(dT, 3);
    Serial.print(F(" degrees Celsius"));
    Serial.println(F(""));

    float FS_SEL = 131;
    float gyro_x = (accel_t_gyro.value.x_gyro - base_x_gyro)/FS_SEL;
    float gyro_y = (accel_t_gyro.value.y_gyro - base_y_gyro)/FS_SEL;
    float gyro_z = (accel_t_gyro.value.z_gyro - base_z_gyro)/FS_SEL;
    float accel_x = accel_t_gyro.value.x_accel;
    float accel_y = accel_t_gyro.value.y_accel;
    float accel_z = accel_t_gyro.value.z_accel;
    float RADIANS_TO_DEGREES = 180/3.14159;
    float accel_angle_y = atan(-1*accel_x/sqrt(pow(accel_y,2) +
pow(accel_z,2)))*RADIANS_TO_DEGREES;
    float accel_angle_x = atan(accel_y/sqrt(pow(accel_x,2) +
pow(accel_z,2)))*RADIANS_TO_DEGREES;
    float accel_angle_z = atan(sqrt(pow(accel_x,2) +
pow(accel_y,2))/accel_z)*RADIANS_TO_DEGREES;;
    //float accel_angle_z = 0;

    float dt =(t_now - get_last_time())/1000.0;
    float gyro_angle_x = gyro_x*dt + get_last_x_angle();
    float gyro_angle_y = gyro_y*dt + get_last_y_angle();
    float gyro_angle_z = gyro_z*dt + get_last_z_angle();
    float unfiltered_gyro_angle_x = gyro_x*dt + get_last_gyro_x_angle();
    float unfiltered_gyro_angle_y = gyro_y*dt + get_last_gyro_y_angle();
    float unfiltered_gyro_angle_z = gyro_z*dt + get_last_gyro_z_angle();
    float alpha = 0.96;
    float angle_x = alpha*gyro_angle_x + (1.0 - alpha)*accel_angle_x;
    float angle_y = alpha*gyro_angle_y + (1.0 - alpha)*accel_angle_y;
    float angle_z = gyro_angle_z; //Accelerometer doesn't give z-angle
    set_last_read_angle_data(t_now, angle_x, angle_y, angle_z,
unfiltered_gyro_angle_x, unfiltered_gyro_angle_y,
unfiltered_gyro_angle_z);
    int mag=sqrt(pow(x-angle_x,2)+pow(y-angle_y,2)+pow(z-angle_z,2));

```

```

    if(mag>=threshold && prev<threshold)
    {
        count+=1;
        Serial.print("steps= ");
        Serial.println(count);
    }
    prev = mag;
    x=angle_x;
    y=angle_y;
    z=angle_z;

    delay(100);
    //Serial.write(10);
}
int MPU6050_read(int start, uint8_t *buffer, int size)
{
    int i, n, error;
    Wire.beginTransaction(MPU6050_I2C_ADDRESS);
    n = Wire.write(start);
    if (n != 1)
        return (-10);
    n = Wire.endTransmission(false);    // hold the I2C-bus
    if (n != 0)
        return (n);
    Wire.requestFrom(MPU6050_I2C_ADDRESS, size, true);
    i = 0;
    while(Wire.available() && i<size)
    {
        buffer[i++]=Wire.read();
    }
    if ( i != size)
        return (-11);
    return (0);    // return : no error
}

int MPU6050_write(int start, const uint8_t *pData, int size)
{
    int n, error;
    Wire.beginTransaction(MPU6050_I2C_ADDRESS);
    n = Wire.write(start);    // write the start address
    if (n != 1)
        return (-20);
    n = Wire.write(pData, size);    // write data bytes
    if (n != size)
        return (-21);
    error = Wire.endTransmission(true);    // release the I2C-bus
    if (error != 0)
        return (error);
    return (0);    // return : no error
}

int MPU6050_write_reg(int reg, uint8_t data)
{
    int error;
    error = MPU6050_write(reg, &data, 1);
    return (error);
}

void bist100(){
    delay(300);
    while(true){

```

```

        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
    {
        HTTPClient http;
        http.begin(urlbist100);
        int httpCode = http.GET();
        if (httpCode > 0) {
            String bist100 = http.getString();
/* get the return value in string*/
            bist1001 = bist100.toFloat();
        }
        http.end();
    }
    u8g2.drawXBMP( 8, 27, 36, 36, bist100logo);
    u8g2.drawXBMP( 110, 5, 13, 16, tllogo);
    u8g2.setFont(u8g2_font_crox5tb_tf);
    u8g2.setCursor( 5, 5); u8g2.print(bist1001);
    u8g2.setFont( u8g2_font_10x20_tf);
    u8g2.setCursor( 45, 38); u8g2.print("BIST100");
    u8g2.sendBuffer();
    butonOku();
    if ( !Right ) {
        bitcoin();}
    if ( !Left ) {
        dolar();
    }
    if ( !Geri ) {
        MenuekranBorsa();
    }
}}

void dolar(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
    {
        HTTPClient http;
        http.begin(urldolar);
        int httpCode = http.GET();
        if (httpCode > 0) {
            String dolar = http.getString();
/* get the return value in string*/
            dolar1 = dolar.toFloat();
        }
        http.end();
    }
    u8g2.drawXBMP( 8, 27, 36, 36, dolarlogo);
    u8g2.setFont(u8g2_font_crox5tb_tf);
    u8g2.setCursor( 5, 5); u8g2.print(dolar1);
    u8g2.setFont(u8g2_font_inbl6_mr);
    u8g2.setCursor( 45, 37); u8g2.print("USD");
    u8g2.sendBuffer();
    butonOku();
    if ( !Right ) {
        bist100();}
    if ( !Left ) {
        euro();
    }
}

```

```

        if ( !Geri ) {
            MenuekranBorsa();}
    }}

void euro(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urleuro);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String euro = http.getString();
/* get the return value in string*/
                euro1 = euro.toFloat();
            }
            http.end();
        }
        u8g2.drawXBMP( 8, 27, 36, 36, eurologo);
        u8g2.setFont(u8g2_font_crox5tb_tf);
        u8g2.setCursor( 5, 5); u8g2.print(euro1);
        u8g2.setFont(u8g2_font_inb16_mr);
        u8g2.setCursor( 45, 37); u8g2.print("Euro");
        u8g2.sendBuffer();
        butonOku();
        if ( !Right ) {
            dolar();}
        if ( !Left ) {
            gumusgram();
        }
        if ( !Geri ) {
            MenuekranBorsa();}
    }}

void bitcoin(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urlbtc);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String btc = http.getString();
/* get the return value in string*/
                btc1 = btc.toFloat();
            }
            http.end();
        }
        u8g2.drawXBMP( 8, 27, 36, 36, bitcoinlogo);
        u8g2.setFont(u8g2_font_crox5tb_tf);
        u8g2.setCursor(5, 5); u8g2.print(btc1);
        u8g2.setFont(u8g2_font_inb16_mr);
        u8g2.setCursor( 45, 37); u8g2.print("BTC");
        u8g2.sendBuffer();
    }
}

```

```

    butonOku();
    if ( !Right ) {
        eth();}
    if ( !Left ) {
        bist100();
    }
    if ( !Geri ) {
        MenuekranBorsa();}
    }}

void eth(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urleth);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String eth = http.getString();
/* get the return value in string*/
                eth1 = eth.toFloat();
            }
            http.end();
        }
        u8g2.drawXBMP( 8, 27, 36, 36, ethlogo);
        u8g2.setFont(u8g2_font_crox5tb_tf);
        u8g2.setCursor( 5, 5); u8g2.print(eth1);
        u8g2.setFont(u8g2_font_inb16_mr);
        u8g2.setCursor( 45, 37); u8g2.print("ETH");
        u8g2.sendBuffer();
        butonOku();
        if ( !Right ) {
            altin();}
        if ( !Left ) {
            bitcoin();
        }
        if ( !Geri ) {
            MenuekranBorsa();}
        }}
void altin(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urlaltinons);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String altinons = http.getString();
/* get the return value in string*/
                altinons1 = altinons.toFloat();
            }
            http.end();
        }
        u8g2.drawXBMP( 8, 27, 36, 36, altinlogo);
        u8g2.setFont(u8g2_font_crox3hb_tf );

```

```

u8g2.setCursor( 5, 5); u8g2.print(alтинons1);
u8g2.setFont(u8g2_font_inbl6_mr);
u8g2.setCursor( 45, 37); u8g2.print("Altin");
u8g2.sendBuffer();
butonOku();
    if ( !Right ) {
        altingram();}
    if ( !Left ) {
        eth();
    }
    if ( !Geri ) {
        MenuekranBorsa();}
}}

void altingram(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urlaltingram);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String altingram = http.getString();
/* get the return value in string*/
                altingram1 = altingram.toFloat();
            }
            http.end();
        }
        u8g2.drawXBMP( 8, 27, 36, 36, alтинlogo);
        u8g2.drawXBMP( 100, 5, 13, 16, tllogo);
        u8g2.setFont(u8g2_font_crox5tb_tf);
        u8g2.setCursor( 5, 5); u8g2.print(altingram1);
        u8g2.setFont(u8g2_font_inbl6_mr);
        u8g2.setCursor( 45, 37); u8g2.print("Altin");
        u8g2.sendBuffer();
        butonOku();
        if ( !Right ) {
            gumus();}
        if ( !Left ) {
            alтин();
        }
        if ( !Geri ) {
            MenuekranBorsa();}
    }}

void gumus(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urlgumusons);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String gumusons = http.getString();
/* get the return value in string*/

```

```

        gumusons1 = gumusons.toFloat();
    }
    http.end();
}
u8g2.drawXBMP( 8, 27, 36, 36, gumuslogo);
u8g2.setFont(u8g2_font_crox3hb_tf);
u8g2.setCursor( 5, 5); u8g2.print(gumusons1);
u8g2.setFont(u8g2_font_inb16_mr);
u8g2.setCursor( 45, 37); u8g2.print("Gumus");
u8g2.sendBuffer();
butonOku();
    if ( !Right ) {
        euro();}
    if ( !Left ) {
        gumusgram();
    }
    if ( !Geri ) {
        MenuekranBorsa();}
}}

void gumusgram(){
    delay(300);
    while(true){
        u8g2.clearBuffer();
        if ((WiFi.status() == WL_CONNECTED)) //Check the current
connection status
        {
            HTTPClient http;
            http.begin(urlgumusgram);
            int httpCode = http.GET();
            if (httpCode > 0) {
                String gumusgram = http.getString();
                /* get the return value in string*/
                gumusgram1 = gumusgram.toFloat();
            }
            http.end();
        }
        u8g2.drawXBMP( 8, 27, 36, 36, gumuslogo);
        u8g2.setFont(u8g2_font_crox5tb_tf);
        u8g2.setCursor( 5, 5); u8g2.print(gumusgram1);
        u8g2.setFont(u8g2_font_inb16_mr);
        u8g2.setCursor( 45, 37); u8g2.print("Gumus");
        u8g2.sendBuffer();
        butonOku();
        if ( !Right ) {
            euro();}
        if ( !Left ) {
            gumus();
        }
        if ( !Geri ) {
            MenuekranBorsa();}
    }}

void alert(void * parameter){
    while ( true ) {
if ( alarmSaat == Saat and alarmDakika == Dakika and Saniye == 1 and
alarmaktif == 1) {
        for ( i = 1 ; i<11; i++){
            if (i % 2 == 0){
                loop();
            }
        }
    }
}

```

```

        else {
            u8g2.clearBuffer();
            u8g2.setFont(u8g2_font_courB24_tf);
            u8g2.setCursor( 15, 15);u8g2.print("Borsa");
            u8g2.sendBuffer();
        }
        butonOku();
        if ( i == 10 or !Ok) {
            i=11;
            loop();
        }
        delay(1000);
    }}}}

void keepWiFiAlive(void * parameter){
    for(;;){
        if ((WiFi.status() != WL_CONNECTED)){
            WiFi.begin(ssid,pass);
        }
        if ((WiFi.status() != WL_CONNECTED)){
            WiFi.begin(ssidtel,passtel);
        }
        delay(2000);
    }
}

```


TAKIM ÖZGEÇMİŞ BİLGİLERİ

1. ÖĞRENCİ KİŞİSEL BİLGİLER

Adı SOYADI : Mertcan YILMAZ
Doğum Tarihi ve Yeri : 02.08.1999 Kartal/ İstanbul
Yabancı Dili : İngilizce (B2)
E-posta : mertcanyilmazn@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektronik ve Haberleşme Mühendisliği	Yıldız Teknik Üniversitesi	Devam Ediyor
Lise	Sayısal	ECA Elginkan Anadolu Lisesi	2017

TAKIM ÖZGEÇMİŞ BİLGİLERİ

1. ÖĞRENCİ KİŞİSEL BİLGİLER

Adı SOYADI : İbrahim ÖZTÜRK
Doğum Tarihi ve Yeri : 10.06.1999 Nevşehir
Yabancı Dili : İngilizce (B2)
E-posta : ibrahimr.ozturk@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektronik ve Haberleşme Mühendisliği	Yıldız Teknik Üniversitesi	Devam Ediyor
Lise	Sayısal	Nevşehir Hüseyin Avni İncekara Fen Lisesi	2017

TAKIM ÖZGEÇMİŞ BİLGİLERİ

1. ÖĞRENCİ KİŞİSEL BİLGİLER

Adı SOYADI : Murat DELİBALTA
Doğum Tarihi ve Yeri : 08.06.2000 Kadıköy/İstanbul
Yabancı Dili : İngilizce (B2)
E-posta : murat_delibalta@outlook.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Elektronik ve Haberleşme Mühendisliği	Yıldız Teknik Üniversitesi	Devam Ediyor
Lise	Sayısal	ECA Elginkan Anadolu Lisesi	2018