

**KOCAELİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ**

**PROJE RAPORU**

**IoT TABANLI HAVA DURUMU RAPOR SİSTEMİ**

**HALİL İBRAHİM ÇETİN**

**210208020**

**KOCAELİ 2024**

## İÇİNDEKİLER

ŞEKİLLER DİZİNİ.....	ii
ÖZET.....	iii
1. GİRİŞ.....	1
2. KULLANILAN KOMPONENTLER.....	1
2.1 ESP32.....	1
2.2 DHT22.....	1
2.3 RAIN DROP SENSOR.....	1
2.4 LED DİYOT.....	2
3 SİSTEM KURULUMU.....	2
3.1 PROTOTİPLEME.....	2
3.2 SİSTEM DİZAYNI.....	3
3.3 KART BASIMI.....	4
4. KOD.....	5
4.1 ESP32.....	5
4.2 HTML.....	8
KAYNAKLAR.....	15

## ŞEKİLLER DİZİNİ

Şekil3.1 : Sistem prototip görüntüleri.....	5
Şekil3.2.1 : Sistemin Altium çizimleri.....	6
Şekil3.2.2 : Sistemin Altium çizimlerine dair 3B görüntüleri.....	6
Şekil3.3.1 : Transfer alınmış bakır plaketlerin görünümü.....	7
Şekil3.3.2 : Asit banyosu görüntüsü.....	7
Şekil3.3.3: Sistem görünümü.....	8
Şekil4.1 : Websitesi genel görünümü.....	9

# IoT TABANLI HAVA DURUMU RAPOR SİSTEMİ

**Halil İbrahim ÇETİN**

**Anahtar Kelimeler:** Iot, Nesnelerin interneti, Hava durumu rapor sistemi, C, HTML, ESP32

**Özet:**Sistemin çalışma şeması şu şekildedir:

- ESP32 çalıştığında Wi-Fi bağlantısını gerçekleştirene kadar ağ aradığını belli etmek için GPIO17 pinine bağlı LED'i yakıp söndürür.
- Başarılı bir ağ bağlantısı yaparsa GPIO17 pinindeki LED sürekli yanmaya başlar.
- Sistem GPIO22 ve GPIO21 pinleriyle BMP280 üzerinden, GPIO2 piniyle DHT11 üzerinden , ADC0 piniyle Rain Drop Sensor üzerinden veri alır.
- Bu verileri Wi-Fi aracılığıyla API yardımı ile Google Firebase'e gönderir.
- Eğer tüm veriler veritabanına başarılı bir şekilde gönderilirse , ESP32'nin GPIO16 pinine bağlı LED yanıp söner.Verilerden 1 tanesi dahi başarısız olursa bu işlem gerçekleşmez.
- Veriler Firebase'den yine API sayesinde HTML sitesine çekilir.
- Local'de oluşturduğum site tablo ve grafiklerden oluşmaktadır. Tablo için Bootstrap5,grafikler için chart.js plug-inleri kullanılmıştır.Verileri Firebase'den anlık olarak alıp bu verileri grafikler ile kullanıcıya sunup ortamda bulunan hava durumu değişikliğini gösterir.
- Ayrıca sistemin doğru çalışıp çalışmadığını test etmek için dış ortamda test yapılmıştır ve bu testler sırasında sürekli local site ile uğraşmak istenilmediği için bedava hosting veren sitelerden biri aracılığı ile <http://www.havadurumusite.weebly.com> site web'e taşınmıştır ve akabinde sistem telefon üzerinden tarayıcıdan kontrol edilebilir hale gelmiştir.

# **1.GİRİŞ**

Proje 3 adet sensörden 5 adet veriyi kablosuz bir şekilde veritabanına aktarır, o anki hava koşullarını kullanıcıya internet üzerinde bulunan sisteminden kullanıcıya aktarmayı ve kullanıcıyı bu şekilde bilgilendirmeyi amaçlamaktadır.

## **2.KULLANILAN KOMPONENTLER**

### **2.1 ESP32**

NodeMCU ESP32, Internet of Things (IoT) projeleri için kullanılan bir geliştirme kartıdır. ESP32 çipini temel alan bu kart, Wi-Fi ve Bluetooth özellikleri ile donatılmıştır. ESP32, yüksek performanslı bir mikrodenetleyiciye sahiptir ve düşük güç tüketimi ile dikkat çeker. NodeMCU firmware'i, Lua programlama dilini kullanarak kartın işlevselliğini genişletir ve projelerin hızlı bir şekilde geliştirilmesine olanak tanır. Bu kart, sensörlerle iletişim kurma, veri toplama ve internete bağlanma gibi IoT uygulamaları için ideal bir platform sağlar. NodeMCU ESP32, açık kaynaklı bir yaklaşıma dayanarak geniş bir geliştirici topluluğuna sahiptir, bu da projelerinizi hızlı bir şekilde geliştirmenize ve özelleştirmenize yardımcı olur.

### **2.2 DHT22**

DHT22, sıcaklık ve nem ölçümü yapabilen bir sensördür. Projelerinizde bu sensörü kullanarak çevresel koşulları izleyebilir ve verileri toplayabilirsiniz. DHT22, dijital bir sıcaklık ve nem sensörüdür.

### **2.3 RAIN DROP SENSOR**

Rain drop sensör,yağmur damlalarını varlığını algılayabilen bir sensördür.Bu sensör genellikle yağış durumunu izlemek veya su seviyelerini ölçmek için kullanılır.Analog ve dijital sinyaller gönderebilir.

## 2.4 BMP280

BMP280 ; sıcaklık ve basınç ölçümü yapabilen bir sensördür.Bu sensör,genellikle hava basıncı , yükseklik ve sıcaklık gibi atmosferik koşulları ölçmek için kullanılır.

## 2.5 LED DİYOT

LED (Light Emitting Diode), elektrolüminesan özellik gösteren yarı iletken bir aydınlatma cihazıdır. Elektrik akımı ile uyarıldığında ışık yayarak çalışır.

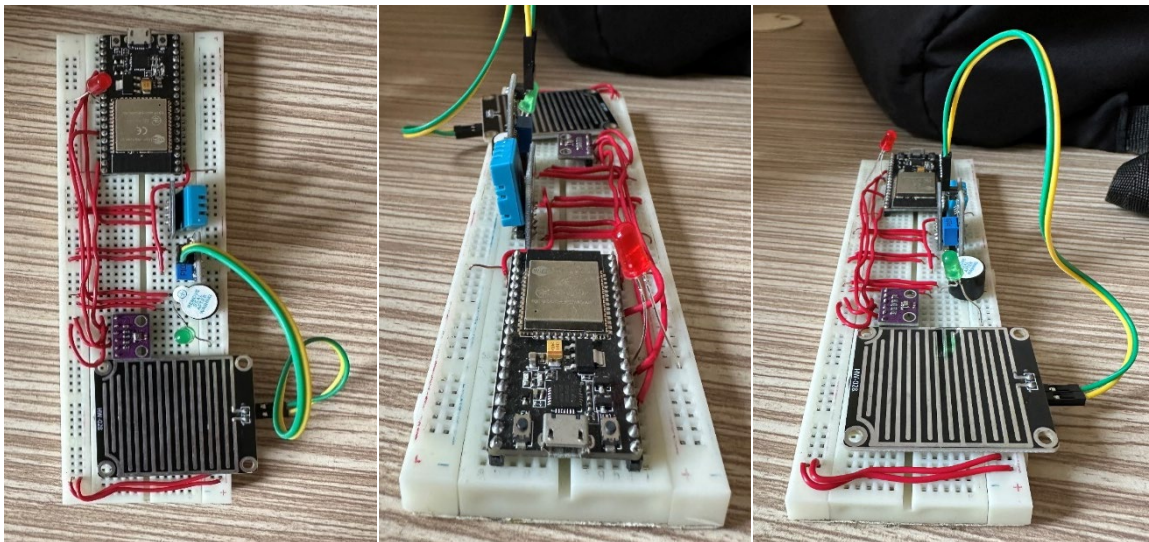
## 3. SİSTEM KURULUMU

### 3.1 PROTOTİPLEME

Sistemi PCB'ye basmadan önce her şeyin doğru çalıştığını anlamak ve emin olmak için önce Breadboard üzerinde prototipledim.Bu aşamada kodlamayı ve sensör bağlantılarını nasıl yapmam gerektiğini belirledim ve sistemi bu süreç içinde oldukça geliştirdim.

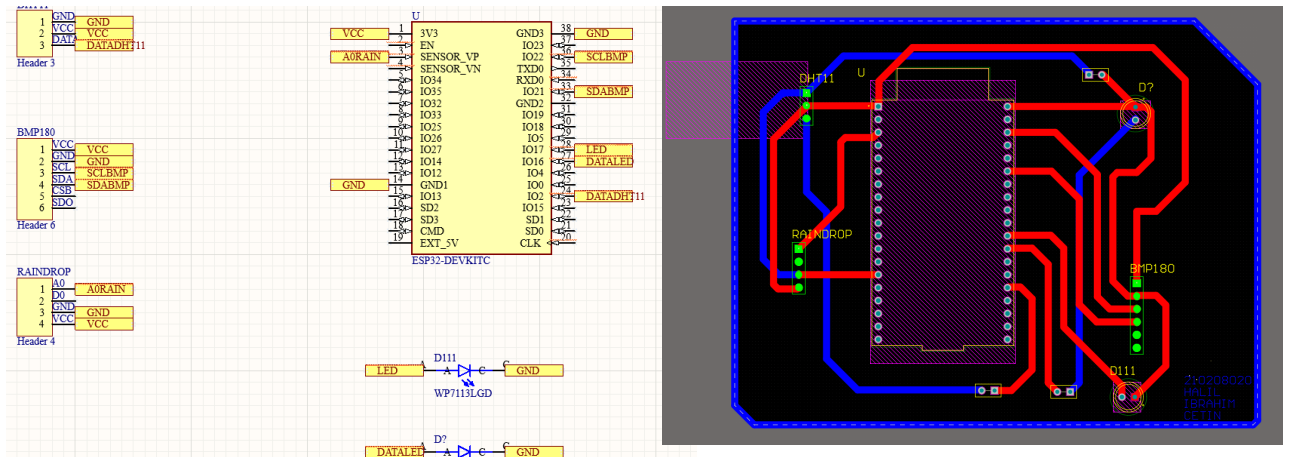
Projede NodeMCU kodlanırken IDE olarak danışıldığı üzere Arduino IDE kullanılmıştır ve kullanılan tüm kütüphaneler de bu IDE'nin kütüphaneleridir.

Projede veritabanı olarak Google Firebase kullanılmıştır ve amaca dahil olan kullanıcı bilgilendirme ekranı Local'de HTML ile oluşturulmuştur.

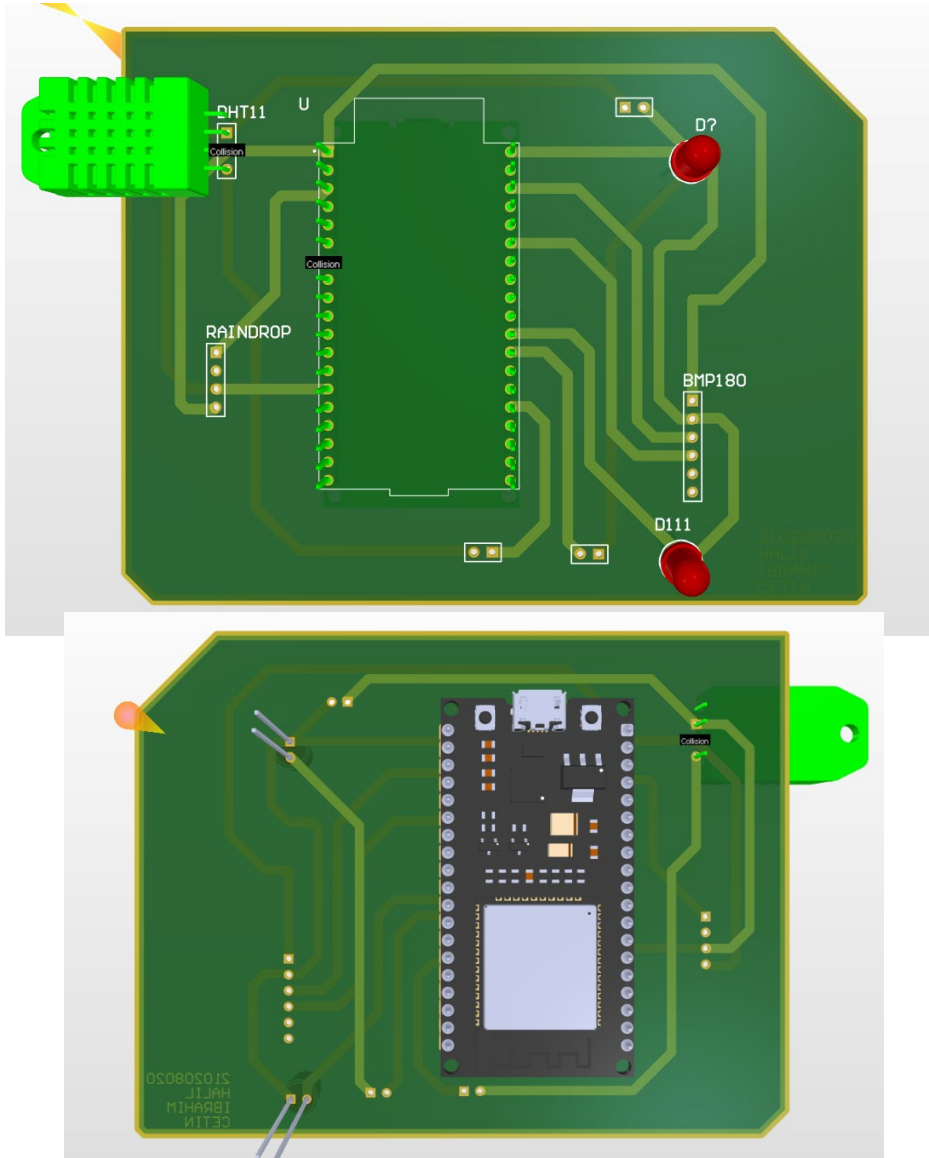


Şekil 3.1 Sistem prototip görüntüleri

### 3.2 SİSTEM DİZAYNI



Şekil3.2.1 : Sistemin Altium çizimleri

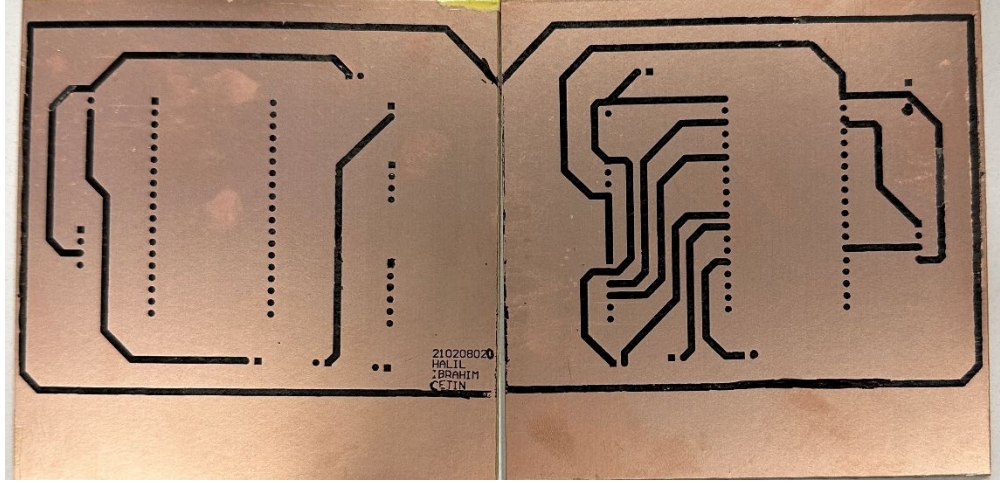


Şekil3.2.2: Sistemin Altium çizimine dair 3B görünümleri



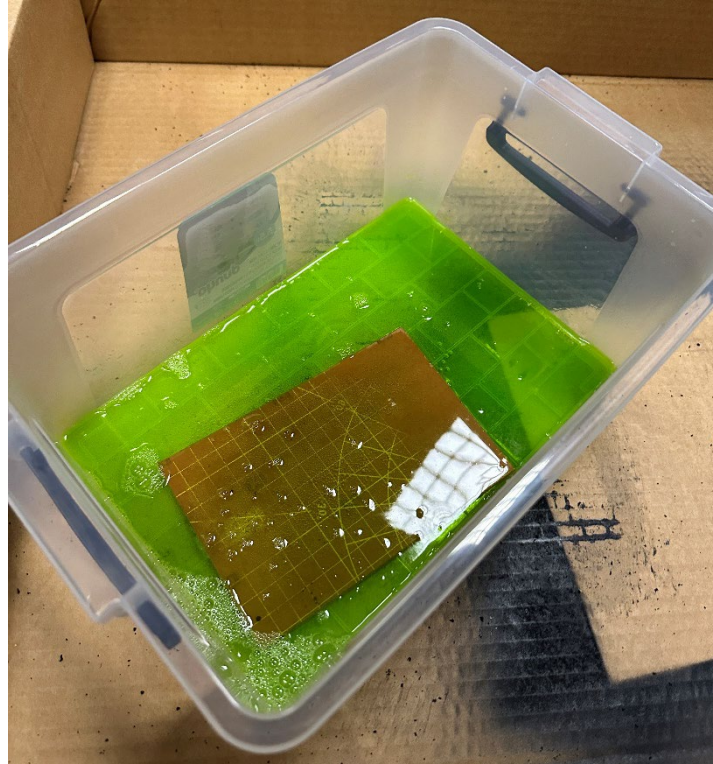
### 3.3 KART BASIMI

Öncelikle Altiumda çizilen kart tasarımı Baskı devre kağıdına yazdırılıp daha sonra 10x10cm bakır plakelere arkalı önlü transfer edilmiştir.



Şekil 3.3.1 : Transfer alınmış bakır plakelerin görünümü

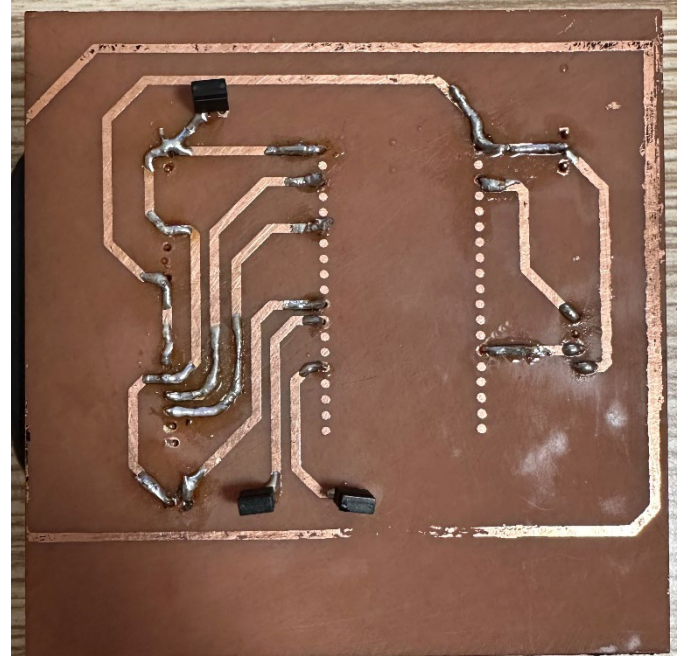
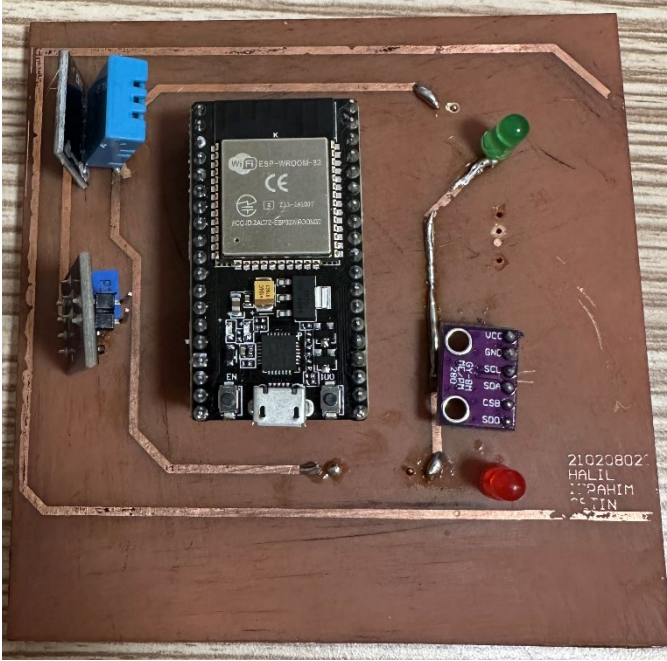
Daha sonra bakır plakeler 3 ölçek HCl 1 ölçek  $H_2O_2$  çözeltisine atılıp bakırların çözünmesi beklenmiştir.



Şekil 3.3.2 : Asit banyosu görünümü



Asit banyosu işleminden sonra elimizde PCB'lerimiz hazır şekilde bulunmaktadır ve daha sonra lehim işlemlerini de hallettikten sonra sisteminiz tamamen hazır şekilde bulunur.



Şekil 3.3.3 : Sistemin görünümü

## 4. KOD

### 4.1 ESP32

```
include <DHT.h>
#include <WiFi.h>
#include <IOXhop_FirebaseESP32.h>
#include <Adafruit_BMP280.h>
#include <Wire.h>
#include <SPI.h>

DHT dht(2,DHT11);

//DEFINE KISIMLARI BASLANGICI
#define WIFI_SSID "ASUS"
#define WIFI_SIFRE "55201056622Sc@"
#define FIREBASE_AUTH "https://iot-weather-report-for-muhtas-default-rtdb.europe-west1.firebaseio.com/"
//#define FIREBASE_HOST "Ux1DIxGD7dZN9oft22Lk15VX6pMx0vy3ErQnZT50"
#define FIREBASE_HOST "AIzaSyDft3CF8JriwsZB8RX--h1NWaH3B0W98jM"
```

```

Adafruit_BMP280 bmp;
//DEFINE SONU

//SETTING UP BASLANGICI
void setup() {
    pinMode(16,OUTPUT);
    pinMode(17,OUTPUT);
    dht.begin();
    delay(2000);
    Serial.begin(9600);

    /*unsigned durum;
    durum = bmp.begin(0x76);*/

    WiFi.begin(WIFI_SSID,WIFI_SIFRE);
    Serial.println("Bağlanılıyor...");
    while(WiFi.status() != WL_CONNECTED)
    {
        Serial.println(".");
        delay(500);
        digitalWrite(17,HIGH);
        delay(500);
        digitalWrite(17,LOW);
    }
    Serial.println();
    Serial.println("Bağlandı");
    delay(500);
    digitalWrite(17,HIGH);
    Serial.println(WiFi.localIP());
    Firebase.begin(FIREBASE_AUTH, FIREBASE_HOST);
    bmp.begin(0x76);
}

//SETTING UP SONU

//FLAG ATAMALARI
int n = 0;
//FLAG ATAMALARI SONU

//SERİ MONİTÖR KISMI
void loop() {
    double sicaklik=dht.readTemperature();
    double nem=dht.readHumidity();
    float yagmursensoru=analogRead(A0);
    yagmursensoru =map(yagmursensoru, 0, 1023 ,0.0 ,100.0);
    yagmursensoru =(yagmursensoru/float(4.0));
    double basinc = bmp.readPressure();
    double irtifa = bmp.readAltitude();

```

```

basinc=basinc/100;
basinc*=1.0377551;
irtifa*=1.44540001;
Serial.print("\nSicaklik: ");
Serial.print(sicaklik);
Serial.print(" Nem:");
Serial.print(nem);
Serial.print(" Sensor kuruluđu:");
Serial.print(yagmursensoru);
Serial.print(" Basınc:");
Serial.print(basinc);
Serial.print("Pa");
Serial.print(" Rakım:");
Serial.print(irtifa);
Serial.print("m");
delay(2000);

//SERİ MONİTÖR SONU

//FIREBASE

Firebase.setFloat("Nem:",nem);
if(Firebase.failed()){
    Serial.println("!! (NEM) VERİLER SUNUCUYA İLETİLEMEDİ!!");
    Serial.println(Firebase.error());
    return;
}
Firebase.setFloat("Sicaklik:",sicaklik);
if(Firebase.failed()){
    Serial.println("!! (SICAKLIK) VERİLER SUNUCUYA İLETİLEMEDİ!!");
    Serial.println(Firebase.error());
    return;
}
Firebase.setFloat("Sensor Kurulugu:", yagmursensoru);
if(Firebase.failed()){
    Serial.println("!! (YAĞMUR SENSÖRÜ) VERİLER SUNUCUYA İLETİLEMEDİ!!");
    Serial.println(Firebase.error());
    return;
}
delay(200);

Firebase.setFloat("Basinc", basinc);
if(Firebase.failed()){
    Serial.println("!! (BASINÇ) VERİLER SUNUCUYA İLETİLEMEDİ!!");

```

```

    Serial.println(Firebase.error());
    return;
}
Firebase.setFloat("Rakim:", irtifa);
if(Firebase.failed()){
    Serial.println("!! (RAKIM) VERİLER SUNUCUYA İLETİLEMEDİ!!");
    Serial.println(Firebase.error());
    return;
}

Serial.println(" Tüm veriler başarıyla gönderildi.");

digitalWrite(16,HIGH);//tüm veriler başarıyla gönderildiğinde LED ile bildir komutu
delay(100);
digitalWrite(16,LOW);
delay(1000);
}

```

## 4.2 HMTL



Şekil 3.3.4 Websitesi genel görünümü

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">

```

```

<meta name="viewport" content= ,width=device-width, initial-scale=1">
<title>Sensör verileri</title>
<!-- Bootstrap tablo düzenlemeleri-->
<style type="text/css">
  body {
    background-color: #a2c9a5;
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
  }

  .data-container {
    display: flex;
    justify-content: space-between;
    width: 85%;
    margin: 75px auto;
    padding: 30px;
    background-color: #FFF;
    box-shadow: 0 2px 20px rgba(0, 0, 0, 0.1);
    border-radius: 25px;
  }

  .data-item {
    text-align: center;
  }

  .data-item h2 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 10px;
  }

  .data-item h3 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 20px;
    font-weight: bold;
    margin-bottom: 10px;
  }

  .data-item p {
    font-size: 20px;
    font-weight: bold;
    color: #6EB7FF;
  }

  .data-head {
    margin: auto;
    width: 50%;

```

```

        text-align: center;
        font-size: 45px;
        font-weight: bolder;
        margin: 50px auto;
        padding: 20px;
        background-color: #FFF;
        box-shadow: 0 5px 20px rgba(0, 0, 0, 0.1);
        border-radius: 20px;
    }
</style>
</head>

<body>
    <!--Tablo Başlıklar-->

        <div class="data-head"><font color=#FF0004>Canlı Sensör Verileri"</font></div>

    <div class="data-container">
        <div class="data-item">
            <h2><font color=#FF0004> Sıcaklık</font></h2>
            <p class="value" id="sicaklik">22 &#8451;</p>
        </div>
        <div class="data-item">
            <h2><font color=#FF0004>Nem</font></h2>
            <p class="value" id="nem">10%</p>
        </div>
        <div class="data-item">
            <h2><font color=#FF0004>Rakım</font></h2>
            <p class="value" id="rakim">100</p>
        </div>
        <div class="data-item">
            <h2><font color=#FF0004>Basınç</font></h2>
            <p class="value" id="basinc">1016</p>
        </div>
        <div class="data-item">
            <h2><font color=#FF0004>Sensör Kuruluşu</font></h2>
            <p class="value" id="sensor_kurulugu">50</p>
        </div>
    </div></font>

    <!-- Canvas ayarlamaları -->
    <canvas id="NemGrafik" width="200" height="100"></canvas>
    <canvas id="SicaklikGrafik" width="200" height="100"></canvas>
    <canvas id="BasincGrafik" width="200" height="100"></canvas>
    <canvas id="KurulukGrafik" width="200" height="100"></canvas>
    <canvas id="RakimGrafik" width="200" height="100"></canvas>

```

```

<!--Chart.js entegrasyonu -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<!--Firebase entegrasyonu -->
<script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>

<script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-database.js"></script>

<script type="module">

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.20.0/firebase-app.js";
const firebaseConfig = {
  apiKey: "AIzaSyDft3CF8JriwsZB8RX--h1NWaH3B0W98jM",
  authDomain: "iot-weather-report-for-muhtas.firebaseio.com",
  databaseURL: "https://iot-weather-report-for-muhtas-default-rtdb.europe-
west1.firebaseio.com",
  projectId: "iot-weather-report-for-muhtas",
  storageBucket: "iot-weather-report-for-muhtas.appspot.com",
  messagingSenderId: "1022443910148",
  appId: "1:1022443910148:web:dc42f434eea9d89838b151",
  measurementId: "G-TQ9BLRDKL3"
};

firebase.initializeApp(firebaseConfig);

// Database'den Veri klasörlerinin verilerini alma
var database = firebase.database();

// getting reference to the data we want
var Veri1 = database.ref('Nem:');
var Veri2 = database.ref('Sicaklik:');
var Veri3 = database.ref('Rakim:');
var Veri4 = database.ref('Basinc:');
var Veri5 = database.ref('Sensor Kurulugu:');

//Değişkenlere göre grafik oluşturma
var grafik1 = document.getElementById('NemGrafik').getContext('2d');
var grafik2 = document.getElementById('SicaklikGrafik').getContext('2d');
var grafik3 = document.getElementById('RakimGrafik').getContext('2d');
var grafik4 = document.getElementById('BasincGrafik').getContext('2d');
var grafik5 = document.getElementById('KurulukGrafik').getContext('2d');

//Grafikler
var NemGrafik = new Chart(grafik1, {

```



```

type: 'line',
data: {
  labels: [],
  datasets: [{
    label: 'Nem',
    borderColor: 'blue',
    data: [],
  }]
},
options: {
  responsive: true,
  maintainAspectRatio: true,
  width: 400,
  height: 200,
},
});

var SicaklikGrafik = new Chart(grafik2, {

  type: 'line',
  data: {
    labels: [],
    datasets: [{
      label: 'Sıcaklık',
      borderColor: 'red',
      data: [],
    }]
  },
  options: {
    responsive: true,
    maintainAspectRatio: true,
    width: 800,
    height: 400,
  },
});

var RakimGrafik = new Chart(grafik3, {

  type: 'line',
  data: {
    labels: [],
    datasets: [{
      label: 'Rakım',
      borderColor: 'green',
      data: [],
    }]
  },
},

```

```

        options: {
            responsive: true,
            maintainAspectRatio: true,
            width: 800,
            height: 400,
        },
    });

var BasincGrafik = new Chart(grafik4, {

    type: 'line',
    data: {
        labels: [],
        datasets: [{
            label: 'Basınc',
            borderColor: 'yellow',
            data: [],
        }]
    },
    options: {
        responsive: true,
        maintainAspectRatio: true,
        width: 800,
        height: 400,
    },
});

var KurulukGrafik = new Chart(grafik5, {

    type: 'line',
    data: {
        labels: [],
        datasets: [{
            label: 'Sensör Kuruluğu',
            borderColor: 'black',
            data: [],
        }]
    },
    options: {
        responsive: true,
        maintainAspectRatio: true,
        width: 800,
        height: 400,
    },
});

```

```

// Grafikleri firebase'e göre güncelleme
function updateCharts(chart, newData) {
    var time = new Date().toLocaleTimeString();
    chart.data.labels.push(time);
    chart.data.datasets[0].data.push(newData);
    chart.update();
}

// Verilerin grafiğine entegrasyonu
Veri1.on('value', function(getdata1) {
    var humi = getdata1.val();
    document.getElementById('nem').innerHTML = "%" + humi;
    updateCharts(NemGrafik, humi);
});

Veri2.on('value', function(getdata2) {
    var temp = getdata2.val();
    document.getElementById('sicaklik').innerHTML = temp + "&#8451;";
    updateCharts(SicaklikGrafik, temp);
});

Veri3.on('value', function(getdata3) {
    var irtifa = getdata3.val();
    document.getElementById('rakim').innerHTML = irtifa + "m";
    updateCharts(RakimGrafik, irtifa);
});

Veri4.on('value', function(getdata4) {
    var basinc = getdata4.val();
    document.getElementById('basinc').innerHTML = basinc + "hPa";
    updateCharts(BasincGrafik, basinc);
});

Veri5.on('value', function(getdata5) {
    var kuruluk = getdata5.val();
    document.getElementById('sensor_kurulugu').innerHTML = "%" + kuruluk;
    updateCharts(KurulukGrafik, kuruluk);
});
</script>
</body>
</html>

```

## KAYNAKLAR

Adafruit , 2021 , DHT.h , [www.github.com](https://www.github.com) , [github.com/adafruit/DHT-sensor-library/blob/master/DHT.h](https://github.com/adafruit/DHT-sensor-library/blob/master/DHT.h) (Ziyaret tarihi:1 Kasım 2023)

ArtonShop , 2017 , IOXhop\_FirebaseESP32.h , [www.github.com](https://www.github.com) , [github.com/ArtronShop/IOXhop\\_FirebaseESP32/blob/master/IOXhop\\_FirebaseESP32.h](https://github.com/ArtronShop/IOXhop_FirebaseESP32/blob/master/IOXhop_FirebaseESP32.h) (Ziyaret tarihi 1 Kasım 2023)

Adafruit , 2022 , Adafruit\_BMP280.h , [www.github.com](https://www.github.com) , [github.com/adafruit/Adafruit\\_BMP280\\_Library/blob/master/Adafruit\\_BMP280.h](https://github.com/adafruit/Adafruit_BMP280_Library/blob/master/Adafruit_BMP280.h) (Ziyaret tarihi:1 Kasım 2023)

Nicholas Zambetti , 2006 , [www.github.com](https://www.github.com) , [github.com/esp8266/Arduino/blob/master/libraries/Wire/Wire.h](https://github.com/esp8266/Arduino/blob/master/libraries/Wire/Wire.h) (Ziyaret tarihi:1 Kasım 2023)

Hristo Gochkov , 2015 , [www.github.com](https://www.github.com) , [github.com/esp8266/Arduino/blob/master/libraries/SPI/SPI.h](https://github.com/esp8266/Arduino/blob/master/libraries/SPI/SPI.h) (Ziyaret tarihi:1 Kasım 2023)

MIT , 2014 , [www.chartjs.org](https://www.chartjs.org) , [chartjs.org](https://chartjs.org) (Ziyaret tarihi:15 Kasım 2023)

MIT, 2011 , [www.getbootstrap.com](https://www.getbootstrap.com) , [www.getbootstrap.com](https://www.getbootstrap.com) (Ziyaret tarihi:7 Kasım 2023)