



T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü



SOĞUTUCU DOLAPLARINDAKİ ENERJİ TÜKETİMİ VE HATA BİLDİRİMLERİNİN AES TABANLI IOT UYGULAMALARI İLE SAHADA TAKİBİ VE KONTROLÜ

Yüksek Lisans Tezi

Ümit GÜLER

Elektrik-Elektronik Anabilim Dalı

İzmir
2023

T.C.
EGE ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

**Soğutucu Dolaplarındaki Enerji Tüketimi ve Hata Bildirimlerinin
AES Tabanlı IOT Uygulamaları ile Sahada Takibi ve Kontrolü**

Ümit GÜLER

Danışman: Prof. Dr. Radosveta İvanova SOKULLU

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Elektronik Yüksek Lisans Programı

İzmir

2023

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “Soğutucu Dolaplarındaki Enerji Tüketimi ve Hata Bildirimlerinin AES Tabanlı IOT Uygulamaları ile Sahada Takibi ve Kontrolü” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

15/05/2023

Ümit Güler

ÖZET

Soğutucu Dolaplarındaki Enerji Tüketimi ve Hata Bildirimlerinin AES Tabanlı IOT Uygulamaları ile Sahada Takibi ve Kontrolü

GÜLER, Ümit

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof.Dr. Radosveta İvanova Sokullu

Mayıs 2023, 84 sayfa

Günümüzde enerji kaynaklarının hızla tükenmesinden dolayı, enerji kullanım miktarının takibi ve kontrolü büyük önem arz etmektedir. Bu durum, sürekli enerji altında bulunan ve yüksek güç tüketen cihazlarda daha da önemli hale gelmektedir. Hastaneler, alışveriş merkezleri, spor salonları ve günümüzdeki evler göz önüne alındığında, bu cihazlardan birisi de buzdolaplarıdır. Bu tez çalışması kapsamında, buzdolabı üreticileri ve son kullanıcılar için cihazların takibine ve kontrolüne olanak sağlayan bir Nesnelerin İnterneti (Internet of Things (IoT)) sistemi geliştirilmiştir.

Buzdolabı üreticileri tarafından ESP32 (Hibrit Wi-Fi & Bluetooth) serisi kullanılarak geliştirilen dolapların sahada internete bağlanmasıyla gerekli olan tüm veriler MQ Telemetry Aktarımı (Message Queuing Telemetry Transport (MQTT)) protokolü üzerinden, günümüzün en güçlü simetrik şifreleme algoritması olan Gelişmiş Şifreleme Standardı (Advanced Encryption Standard (AES)) ile şifrelenerek MQTT sunucusuna aktarılmaktadır. Bu tez çalışması kapsamında geliştirilen ve MQTT sunucusuna bağlı olan bir Windows Servis uygulaması ile çözülen şifreler, aynı ortamda bulunan veritabanına kaydedilmektedir. Geçmişe yönelik takip için tutulan bu kayıtlar, geliştirilen masaüstü uygulaması ve web sitesi üzerinden kontrol edilebilmektedir.

Bahsedilen masaüstü uygulaması, sadece buzdolabı üreticilerinin kullanımına yönelik tasarlanmıştır. MQTT aracısına direk bağlantı sayesinde, anlık olarak gelen şifreli bilgilere, tam erişim sağlamak ve deşifre ederek analiz yapabilmektedir. Böylece nominal güç tüketimi bilinen bir buzdolabının

arızaya veya yanlış kullanıma dayalı fazladan enerji tüketimi anında tespit edilebilmektedir. Aynı zamanda, istenilen dolaplar şifreli olarak uzaktan kontrol edilebilmektedir. Bunun yanında, arıza durumlarını daha iyi analiz edebilmek için, geçmiş kayıtlar gerek uygulama üzerinden gerekse uygulamanın excel tablosu çıktısı sayesinde rahatlıkla incelenebilmektedir. Böylece olası iyileştirmeler için büyük bir bilgi havuzu oluşmuş olmaktadır.

Web sitesi tasarımı ise, herhangi bir tarayıcı üzerinden hızlı takip yapılabilmesi için buzdolabı üreticileri, yetkili teknik servisler ve son kullanıcılara yönelik tasarlanmıştır. Bu sayede son kullanıcılar için kullanım bilgilerine erişim sağlanırken, teknik ekipler gerek tablo sunumlarından, gerekse grafiklerden dolap geçmişini inceleyerek, arıza tespitine yardımcı bilgiler edinilmesi sağlanmıştır.

Son kullanıcı tarafında Wi-Fi bağlantısı olmayabileceği düşünülerek, cihaz içerisinde bulunan ürünlerinin anlık takibinin yapılabilmesi için bluetooth düşük enerji (Bluetooth Low Energy (BLE)) bağlantısı ile iletişim kurulan bir mobil uygulama geliştirilmiştir.

Bu tez kapsamında geliştirilen tüm uygulamalara, veritabanına önceden kaydedilmiş, kullanıcı adı ve şifresi ile giriş yapılabilmektedir. Ancak uygulamalar tarafından veritabanına doğrudan erişim, güvenlik zaafı olabileceği için ara katman olarak Uygulama Programlama Arayüzü (Application Programming Interface (API)) geliştirilmiştir. Tüm uygulamalar bu API üzerinden Üstün Metin Transfer Protokolü (Hyper Text Transfer Protocol (HTTP)) isteğinde bulunur ve yetki alırlar. Ancak elde edilen bu yetki sınıfına göre gerekli görüntüleme işlemi gerçekleştirilebilir.

Anahtar sözcükler: Enerji İzleme, Uzaktan Kontrol, BLE, Wi-Fi, MQTT Protokolü, AES Şifreleme.

ABSTRACT**Field monitoring and control of energy consumption and error notifications in refrigerators with AES-based IoT applications**

GÜLER, Ümit

MSc in Department of Elektrical-Electronics Engineering

Supervisor: Prof.Dr Radosveta Ivanova Sokullu

May 2023, 84 pages

Monitoring and controlling the amount of energy usage has become significantly important in today's world due to the rapid depletion of energy resources. This becomes even more important in devices that are constantly under energy and consume high power. Considering hospitals, shopping malls, fitness centers and today's homes, one of these devices is refrigerators. Within the scope of this thesis, Internet of Things (IoT) system has been developed for refrigerator manufacturers and end users, which allows devices to be monitored and controlled.

Refrigerators developed using the ESP32 (Hibrit Wi-Fi & Bluetooth) series can connect to the internet in the field. All the necessary data for this connection is encrypted using Advanced Encryption Standard (AES), the most powerful symmetric encryption algorithm of today, and transferred to the MQTT broker via the Message Queuing Telemetry Transport (MQTT) protocol. Within the scope of this thesis study, a Windows Service application, connected to the MQTT server, decrypts the encrypted data and stores it in a local database in the same environment. These records, which are kept for historical follow-up, can be checked through the developed desktop application and website.

The aforementioned desktop application is intended for use by refrigerator manufacturers only. Thanks to the direct connection to the MQTT agent, it provides full access to instantaneous encrypted information and can decipher and analyze. Thus, the excess energy consumption of a refrigerator with a

known nominal power consumption due to malfunction or misuse can be detected immediately. At the same time, the desired cabinets can be controlled remotely with a password when necessary. In addition, in order to analyze fault situations better, historical records can be easily examined both through the application and through the application's excel table output. As a result, a large pool of information is created for possible improvements.

The website is designed for refrigerator manufacturers, authorized technical services and end users so that they can be followed quickly via any browser. In this way, while access to usage information is provided for end users, technical teams can examine the cabinet history both from table presentations and graphics, and to obtain helpful information for fault detection.

Considering that the end user may not have a Wi-Fi connection, a mobile application that communicates with Bluetooth Low Energy (BLE) connection has been developed in order to instantly monitor the products in the device.

All applications developed within the scope of this thesis can be accessed with a user name and password previously saved in the database. However, since direct access to the database by applications may be a security weakness, Application Programming Interface (API) has been developed as a middleware. All applications request and receive authorization for the Superior Text Transfer Protocol (HTTP) over this API. However, the required imaging operation can be performed according to this obtained authorization class.

Keywords: Energy Monitoring, Remote control, BLE, Wi-Fi, MQTT Protocol, AES Encryption.

ÖNSÖZ

Global olarak kaynaklarımızın hızla tükenmesi ve artık daha kısıtlı hale gelen kaynakların mali yükünün artması nedeniyle, tüm gözler verimlilik konusuna çevrilmiştir. Bu bağlamda, belirlenen miktarların üzerinde güç tüketen cihazlar artık kabul görmemeye başlamıştır. Diğer yandan, verimliliği yüksek olarak üretilen yeni nesil cihazlar ise, yıllar içerisinde deformasyona uğrayarak performansını kaybetmektedir. Özellikle sürekli enerji kaynağına bağlı olan ve uzun soluklu çalışan cihazlarda, bu durum çok daha belirgin olmaktadır. Buzdolapları, bu konudaki en güzel örneklerden birisidir. Örneğin, kompresör içerisindeki gaz miktarının azalması veya kapı contalarındaki gevşemeler, dolap performansını etkilemese bile yıllar içerisinde fazladan enerji tüketimine neden olabilmektedir. Daha uzun dönemde ise ani bozulmalar sonucunda içerisindeki kritik malzemeler bozulabilmektedir. Tüm bu senaryolar göz önüne alındığında, bu tez kapsamında geliştirilen IoT uygulamaları (mobil, masaüstü, web sitesi) ile bunun önüne geçilmesi planlanmaktadır. Bu yüzden, harcanması gereken güç miktarının üzerine çıktığında ve bu durum süreklilik gösterdiğinde erken uyarı sistemi geliştirilmiştir. Bunun yanında, uzun vadede yapılabilecek üretimsel iyileştirmeler için güvenlik seviyesi üst düzey olan, önemli bir veri kaynağı elde edileceği de düşünülmektedir.

İZMİR

15/05/2023

Ümit Güler

İÇİNDEKİLER

Sayfa

İÇ KAPAK	ii
KABUL VE ONAY SAYFASI	iii
ETİK KURALLARA UYGUNLUK BEYANI	v
ÖZET	vii
ABSTRACT	x
ÖNSÖZ	xiii
İÇİNDEKİLER	xv
ŞEKİLLER DİZİNİ	xviii
TABLolar DİZİNİ	xxii
SİMGELER VE KISALTMALAR DİZİNİ	xxiv
1 GİRİŞ	1
2 KURAMSAL TEMELLER	5
2.1 Kullanılan Özel Haberleşme Format ve Protokoller	5
2.1.1 Klasik Bluetooth ve BLE (Bluetooth Low Energy)	5
2.1.2 Wi-Fi	7
2.1.3 MQTT (Message Queuing Telemetry Transport)	8
2.1.4 HTTP (Hyper-Text Transfer Protocol)	14
2.1.5 JSON Mesaj Formatı	15
2.1.6 Veritabanı	16
2.2 Güvenlik	18
2.2.1 AES (Advanced Encryption Standard) Şifreleme	19
2.2.2 Taşıyıcı Kimlik Doğrulaması	21
3 SİSTEM TASARIMI	24
3.1 Genel Sistem Mimarisi	24
3.2 Donanım Mimarisi	25

İÇİNDEKİLER (devam)

Sayfa

3.2.1	Güç ve Kontrol Kartı Tasarımı	26
3.2.2	Kullanılan Modül ve Sensörler	29
3.3	Yazılım Mimarisi	33
4	SİSTEMİN ÇALIŞMASI	35
4.1	Gömülü Sistem Kodlama	36
4.1.1	ESP32 Modülü Yazılımı	36
4.1.2	HMI (Human Machine Interface) Yazılımı	37
4.2	Arka Yüz Kodlama	41
4.2.1	Windows Servis Yazılımı	41
4.2.2	API (Application Programming Interface) Yazılımı	44
4.3	Ön Yüz Kodlama	47
4.3.1	Masaüstü Uygulama Yazılımı	47
4.3.2	Web sitesi Uygulama Yazılımı	56
4.3.3	Mobil Uygulama Yazılımı	61
5	TARTIŞMA	63
6	SONUÇ	69
	KAYNAKLAR DİZİNİ	71
	TEŞEKKÜR	76
	ÖZGEÇMİŞ	77
	EKLER	78

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 MQTT sürecinin şematik gösterimi	8
2.2 CloudMQTT giriş sayfası	9
2.3 CloudMQTT bağlı cihazlar	10
2.4 CloudMQTT bağlantı hızları	10
2.5 CloudMQTT gönderilen ve alınan mesaj sayıları	10
2.6 CloudMQTT bağlantı bilgileri	11
2.7 CloudMQTT verilerin yayınlanması	11
2.8 HTTP komutları aracılığı ile iletişim	14
2.9 İlişkisel veritabanı	16
2.10 AES şifreleme mimarisi	19
2.11 Aktarılabacak verilerin şifrenilmesi	21
3.1 Tasarlanan sistemin genel mimarisi	24
3.2 Elektronik kartın regüle kısmı	26
3.3 Akü anahtarlama kısmı	26
3.4 Fan ve lamba röle anahtarlama kısmı	27
3.5 ESP32 modül bağlantı kısmı	27
3.6 PCB - Tasarım	28
3.7 PCB - Final	28
3.8 ESP32-Wroom-32 modülü	29
3.9 PZEM-004T V3.0 modül bağlantısı	31
3.10 DHT22 sıcaklık ve nem sensörü	31
3.11 NX4827P043-011 TFT LCD	32
3.12 NX4827P043-011 bileşenleri	32
4.1 Buzdolabı düğümleri - M170ASI Model	35
4.2 Buzdolabı düğümleri - AC50S Model	35

ŞEKİLLER DİZİNİ (devam)

Sayfa

4.3	Mevcut Wi-Fi ağına bağlantı sağlanması	38
4.4	Buzdolabı set parametrelerinin ayarlanması	39
4.5	Tarih ve saat ayarı	40
4.6	Güç parametrelerine erişim	40
4.7	Sistemde çalışan Windows Servis uygulamaları	41
4.8	Visual Studio Analiz - Şifreleme/Deşifreleme Anahtarı	42
4.9	Visual Studio Analiz - Şifreli Metin	43
4.10	Visual Studio Analiz - Deşifre Metin	43
4.11	Visual Studio Analiz - Veritabanına Kayıt	44
4.12	Postman Platformu - Kullanıcı giriş isteği	45
4.13	Postman Platformu - JWT alınması	46
4.14	Postman Platformu - Ölçümlerin alınması	46
4.15	Masaüstü uygulaması giriş ekranı	48
4.16	Masaüstü uygulaması - JWT token alınması	48
4.17	Menüler - Buzdolabı ekleme sekmesi	49
4.18	Sisteme yeni bir buzdolabı tanımlanması	49
4.19	Buzdolabı kayıt kodlama ekranı	50
4.20	Başarılı bir buzdolabı kayıt işlemi	50
4.21	Menüler - Kullanıcı ekleme sekmesi	51
4.22	Kullanıcı ekleme	51
4.23	Kullanıcı kayıt kodlama ekranı	52
4.24	Başarılı bir kullanıcı ekleme işlemi	52
4.25	Menüler - Yetki atama sekmesi	53
4.26	Kullanıcı yetkilendirme menüsü	53
4.27	Kullanıcı yetkilendirme kodlama ekranı	54
4.28	Başarılı bir yetkilendirme işlemi	54
4.29	Takip ve kontrol paneli	55

4.30	Ölçümlerin excel tablosuna aktarımı	56
4.31	Web Sitesi kullanıcı giriş ekranı	56
4.32	Kullanıcı girişi ile JWT token alınması	57
4.33	Yetkili olunan dolapların öğrenilmesi	58
4.34	Buzdolabı ölçümlerinin listelenmesi	59
4.35	Web sitesi - Güç tüketimi grafiği	60
4.36	Web sitesi - Kabin sıcaklığı grafiği	60
4.37	Mobil uygulama bluetooth bağlantısı	61
4.38	Visual Studio Code - Analiz	62
4.39	Mobil uygulama arayüzü	62
5.1	MQTT Broker - Son İstek ve vasiyetname özelliği	63
5.2	Besleme geriminin kesilmesi testi	64
5.3	Sensör arızalı olması testi	65
5.4	Kapı açık kalması testi	65
5.5	Yüksek sıcaklık limit testi	66
5.6	Test sonuçlarının grafiksel incelenmesi	66
5.7	Hataların veri listesinde incelenmesi	67
5.8	Kullanıcıların yetkilendirilmesi	67
5.9	Id değeri 1 olan bir kullanıcının yetkili olduğu dolaplar	68

TABLOLAR DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Bluetooth BR/EDR ve BLE (Zhang et al., 2019)	6
2.2 IEEE 802.11 Wi-Fi Standartları (Gierłowski et al., 2011)	7
2.3 Veritabanı ölçümler tablosunun listelenmesi	18
3.1 ESP32-Woom-32 özellikleri (Espressif Systems, 2023)	30
3.2 PZEM-004T V3.0 özellikleri (Peacefair Electronics, 2021)	30
3.3 NX4827P043-011 Özellikleri	33

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltmalar</u>	<u>Açıklama</u>
<i>AES</i>	Advanced Encryption Standard
<i>API</i>	Application Programming Interface
<i>BLE</i>	Bluetooth Low Energy
<i>Gbps</i>	Gigabit per second
<i>GHz</i>	Gigahertz
<i>HMI</i>	Human Machine Interface
<i>HTTP</i>	Hyper Text Transfer Protocol
<i>IoT</i>	Internet of Things
<i>JSON</i>	Javascript Object Notation
<i>JWT</i>	JSON Web Tokens
<i>LCD</i>	Liquid Crystal Display
<i>Mbps</i>	Megabit per second
<i>MHz</i>	Megahertz
<i>M2M</i>	Machine-to-Machine

SİMGELER VE KISALTMALAR DİZİNİ (devam)

<u>Kısaltmalar</u>	<u>Açıklama</u>
<i>MQTT</i>	Message Queuing Telemetry Transport
<i>OFDMA</i>	Orthogonal Frequency Division Multiple Access
<i>REST</i>	Representational State Transfer
<i>SQL</i>	Structured Query Language
<i>TFT</i>	Thin Film Transistor
<i>QoS</i>	Quality of Service

1 GİRİŞ

Günümüzün en temel ihtiyaçları arasında yer alan elektrik enerjisine olan talep ve kullanım miktarı her geçen gün artmaktadır. Evlerde kullanılan elektronik cihazların sayısının hızla artması, giderek yaygınlaşan toplu alanlardaki ihtiyaçlar ve fabrikalarda kullanılan makinelerin çok daha güçlü hale gelmesi bunun en önemli etkenlerindendir. Ancak, hayatımızın her alanında olduğu gibi bu alanda da büyük ölçüde israf söz konusu olabilmektedir. Çoğu zaman önemsenmeyen, gözden kaçan veya küçük bir arızaya bağlı fazladan tüketimin maliyeti, global ölçüde çok büyük olabilmektedir (Vikram et al., 2022). Bu yüzden IoT çağını yaşadığımız bugünlerde, kullanılan enerjinin takibi ve kontrolü oldukça önemli hale gelmiştir (Raiker et al., 2020).

Sensör teknolojisindeki önemli gelişmeler ve elde edilen verilerin aktarımında yüksek hızlara çıkılabilmesi, takip ve kontrol uygulamalarının daha verimli hale gelmesini sağlamıştır (Carratú et al., 2022). Özellikle tek bir sensör modülü üzerinden birden fazla verinin ölçülmesi sayesinde, tek bir haberleşme çevre birimi üzerinden tüm veriler okunabilmektedir. Böylece kaynak kısıtlı uygulamalarda bile ayrıntılı bilgilere ulaşılabilir (Devireddy et al., 2021). Bu sensörlerden oluşan düğümlerin, bir broker üzerinden yönlendirilmesi ile geliştirilen uygulamalardan izleme sağlanabilmektedir. Ayrıca yine bu uygulamalar üzerinden gerekli kontrol işlemi de sağlanabilmektedir (Zare & Iqbal, 2020).

Enerji kullanımına yönelik talebin giderek artması ve kaynakların daha kısıtlı hale gelmesi, son kullanıcı açısından kullanım maliyetlerin yükselmesi anlamına da gelmektedir. Buna yönelik olarak da birçok fatura takip sistemi ve uzaktan kontrol uygulamaları geliştirilmiştir. Örneğin, eski sayaç birimleriyle uyumlu, kullanıcı dostu ve insan hatası en az olan bir faturalama sistemi için bir çalışma geliştirilmiştir. LDR sensörü, mevcut mekanizmayı koruyacak şekilde tüketilen miktarı algılamakta ve kullanıcı SMS ile bilgilendirilmektedir (Patel et al., 2019).

Benzer bir başka çalışmada ise, ESP8266 Wi-Fi modülü ve bir Blynk uygulaması kullanarak IoT tabanlı bir enerji ölçer mekanizması kullanılmıştır. Bu çalışmada maksimum talebi belirlemek için enerji ölçerinin, süre ile orantılı olarak 30 dakikada 1000 darbeye ayarlanmış, yanıp sönen bir LED ile güç tüketimini temsil ettiği bir sistem tasarlanmıştır (Ahmad et al., 2021).

Elektrik yüklerinin güç tüketimini izlemeyi ve ilgili kişilere ESP8266 Wi-Fi modülü kullanarak rapor etmeyi amaçlayan bir çalışmada ise, güç ölçümü için yük ile seri bağlantılı akım transformatörü ve gerilim regülatörü ile sınırlı bir bilgi elde edilmektedir. Ayrıca anlık izleme sırasında aşırı güç tüketen cihazları kapatma ve bu durumu bir mobil uygulama üzerinde inceleme de ele alınmıştır (Rao & Sudheer, 2020).

Gerçekleştirilen bir başka çalışmada, ESP8266 modülünün MQTT broker'a bağlanarak sadece anlık izleme sağlayan bir sistem sunulmuştur. Ayrıca, herhangi bir güvenlik ve koruma teknolojisi uygulanmamış ve önerilen sistem güvenlik risklerini azaltmak için lokal bilgisayarda çalıştırılmıştır. Yerel ağda çalışıldığından dolayı herhangi bir güvenlik mekanizmasının uygulanmadığı gözlemlenmiştir (Kodali & Soratkal, 2016).

ESP32 modülü üzerine inşa edilmiş benzer bir çalışmada ise, uzaktan sensör izleme sistemi önerilmektedir. Bu sistem, İleri Şifreleme Standardı (AES) algoritması kullanılarak 256-bit anahtar ile Message Queuing Telemetry Transport (MQTT) protokolünü kullanarak akıllı bir evi hayata geçirmeyi amaçlamaktadır. Bahsedilen çalışmada, MQTT aracısı Raspberry Pi üzerine kurulmuş ve bir mobil uygulama ile iletişim sağlanmaktadır (Setiawan & Magfirawaty, 2021).

Akıllı binaların enerji tüketiminin izlenmesi konusunda tasarlanan bir çalışmada ise, ESP8266 ve PZEM-004T modülü kullanılarak ilgili ölçümler gerçekleştirilmiştir (Tantidham et al., 2018). Gelecek çalışması olarak önerilen, tüm ölçümlerin arayüz tasarımları ile görüntülenmesi işlemi, bu tez çalışmasının gelişiminde değerlendirilmiştir.

Bu tez çalışması kapsamında gerçekleştirilmek istenen ise, takip ve kontrolün sağlanmasının yanında, elde edilen kapsamlı verilere göre geleceğe yönelik gerekli iyileştirmelerin yapılabilmesi de sağlamaktır. Bu bağlamda tasarlanan sistem, günümüzde neredeyse her evde, ticari sektörde, medikal alanda ve bilimsel laboratuvarlarda vazgeçilmez konumda olan buzdolapları üzerinde geliştirilmiştir. Bahsedildiği gibi oldukça geniş bir kullanım alanı bulunan buzdolaplarının, daima bir enerji kaynağına bağlı olması nedeniyle, sürekli enerji tüketimine neden olması, bu çalışma için ayrı bir motivasyon kaynağı olmuştur. Çünkü bu sayede, takip ve kontrol sistemi ile yapılabilecek iyileştirmelerin önemi daha net anlaşılabilmektedir.

Özellikle medikal alanda ve bilimsel laboratuvarlarda, bu cihazlar içerisinde saklanan ürünlerin, bir arızaya dayalı bozulmalar yaşandığında çoğu zaman telafisi olmamaktadır. Bu çalışmada geliştirilen takip ve kontrol sistemi ile bu gibi kötü senaryolara karşı iyi bir önlem alınmış olacaktır. Soğutma sistemindeki gerileme, enerji tüketimindeki artış, kapının açık kalması gibi durumlar hem müşteri tarafında hem de üretici tarafında sahip oldukları akıllı uygulamalar vasıtasıyla anında görüntülenebilecektir. Buzdolabın yanında olan bir kullanıcı ise LCD panel vasıtasıyla sesli ve görsel detaylarla uyarılabilecektir. Tüm bunların yanında enerji tüketimindeki herhangi bir artışın tespit edilebilmesi sayesinde, cihaz henüz arıza durumuna geçmeden servis yönlendirilebilmesi ile kötü bir senaryo önlenilebilecektir.

Diğer yandan, IoT sistemlerinde geniş bir alana hitap edilmesi ve kullanıcı sayısının çok fazla olması nedeniyle güvenlik önemli bir unsur teşkil etmektedir. Bu yüzden, bu tez çalışmasında tasarlanan sistem mimarisinde, güvenlik konusunda çok titiz davranılmıştır. Kullanılan MQTT haberleşme protokolünde verilerin AES şifreleme ile şifrelenerek aktarılması ve akıllı uygulamaların tamamında bilgi aktarımından önce JSON Web Token (JWT) yetkilendirmesi kullanılması (Adam et al., 2020) sistem güvenliğini üst düzeye çıkarmaktadır. Ayrıca tasarlanan sistemde bahsedilen kullanıcı yetkilendirmesi sayesinde admin olan bir tüm buzdolaplarını görüntüleyebilirken, diğer kullanıcılar sadece yetkisi dahilinde olan buzdolaplarını görüntüleyebilmektedir.

Bu alışmanın 2. bölümünde, gerçekleştirilen sistemin daha anlaşılır olmasını sağlamak için kullanılan özel haberleşme protokolleri, sunucu yapısı ve sistem güvenliğinde kullanılan argümanlar hakkında genel bilgilere yer verilmiştir.

3. bölümde ise, genel sistem mimarisinden bahsedildikten sonra, donanım ve yazılım mimarileri ayrı ayrı incelenmiştir. Ayrıca, bu çalışma kapsamında kullanılan sensör ve modüllerden de bu bölümde bahsedilmiştir.

Sistem mimarisinin detaylı anlatımından sonra sistemin çalışma mantığı 4. bölümde anlatılmıştır. Gömülü sistemler tarafında geliştirilen ESP32 modül ve LCD panel arayüz kullanımı, ön yüz (frontend) kodlama kısmında geliştirilen akıllı uygulamaların kullanımı ve arka yüz (backend) kodlamada çalışan uygulamalar, yazılan kod paraçıkları ile bu bölümde anlatılmıştır.

Geliştirilen IoT sistemindeki olası hata durumları, buna yönelik yapılan testler ve uygulanan senaryolar 5. bölümde tartışılmıştır.

Son olarak 6. bölümde, bu tez çalışma kapsamında gerçekleştirilenler ve literatüre yapılan katkılardan bahsedilmiştir. Elde edilen sonuçlara tek tek yer verilmiştir. Ayrıca gelecek çalışması önerileri de bu bölümde yer almaktadır.

2 KURAMSAL TEMELLER

Bu tez çalışması kapsamında, birçok protokol ve formatın ortak kullanımı ile çok daha kararlı ve esnek bir sistem elde edilmiştir. Sistem yapısı, şifreleme ve yetkilendirme ile daha da güçlendirilmiştir. Ayrıca, tüm verilerin bir havuzda toplanması sayesinde geçmişe yönelik görüntüleme de sağlanmıştır.

2.1 Kullanılan Özel Haberleşme Format ve Protokoller

Tasarlanan IoT sisteminde ESP32 (Hybrid Wi-Fi & Bluetooth) serisinin kullanılması sayesinde dolaplara hem Wi-Fi bağlantısı hem de bluetooth bağlantısı yapılabilmektedir. Wi-Fi haberleşmesi ile MQTT ağına bağlantı sağlanırken, bluetooth haberleşmesi ile tasarlanan akıllı uygulamalara bağlantı sağlanmaktadır. Bluetooth bağlantısında ise, klasik bluetooth yerine BLE haberleşmesi tercih edilmiştir.

2.1.1 Klasik Bluetooth ve BLE (Bluetooth Low Energy)

Kısa menzilli kablosuz iletişim teknolojilerinden birisi olan bluetooth, ilk tanıtıldığında 2.4 Ghz frekans bandında ve Temel Hız (Basic Rate(BR)) versiyonu ile 1 Mbps'e kadar veri aktarımını destekliyordu. 10 metreye kadar menzili olan bu sürümde, Frekans Atlamalı Geniş Spektrum (Frequency Hopping Spread Spectrum (FHSS)) teknolojisi kullanılmıştır.

FHSS, kullanılan frekans bandını ayrıştırarak küçük kanallara bölmektedir (Stranne et al., 2002). Veri aktarımı sırasında bu kanallar arasında çok hızlı geçişler sağlayarak, elektromanyetik girişimlerin önüne geçmekte ve parazitleri önlemeye yardımcı olmaktadır.

Bluetooth 2.0 geliştirildiğinde ise, Gelişmiş Veri Hızı (Enhanced Data Rate(EDR)) versiyonu ile veri hızı yaklaşık 3 Mbps'ye kadar yükseltilmiştir. Bu sürümde, frekans bandında herhangi bir değişiklik yapılmadan, parazitlere karşı saniyede 1600 kez atlama yapabilen Uyarlanabilir Frekans Atlamalı (Adaptive Frequency Hopping (AFH)) teknolojisi kullanılmıştır (Chang, 2014).

Bahsedilen bu iki protokol sürümünün birleştirilmesi ile ortaya çıkan Bluetooth BR/EDR teknolojisi, Klasik Bluetooth olarak da adlandırılmaktadır. Bu yeni sürümde 100 metre menzilde 3 Mbps'e kadar veri aktarım hızı mümkün olabilmektedir (Bayılmış vd., 2004).

Bluetooth 4.0 geliştirildiğinde ise klasik bluetooth yanında Bluetooth Smart olarak da bilinen Bluetooth Düşük Enerji (Bluetooth Low Energy(BLE)) de tanıtılmıştır (Varol, 2019). BLE tasarımıdaki temel amaç ise, gelişen IoT teknolojisine uyum sağlayarak, sensörler ve giyilebilir cihazlar için düşük güçlü kablosuz bağlantı sağlamaktır.

Tablo 2.1: Bluetooth BR/EDR ve BLE (Zhang et al., 2019)

Parametre	Bluetooth BR/EDR (V4.2)	BLE (V4.2)
Menzil	30 m'ye kadar	60 m'ye kadar
Frekans	2,402 – 2,481 GHz	2,402 – 2,481 GHz
Veri Hızı	1- 3 Mbit/s	1 Mbit/s
Güç Tüketimi	Yaklaşık 1W	Yaklaşık 0,01-0,5W
Uygulamalar	Ses, Video vb.	Sensörler, IoT vb.
Ağ Standardı	IEEE 802.15.1	IEEE 802.15.1

Klasik Bluetooth(BR/EDR) ve BLE arasındaki en temel fark ise veri aktarımında izlenen yol ve buna bağlı olarak güç tüketiminde gerçekleşen değişimdir. Klasik Bluetooth'da noktadan noktaya bağlantı kurulmakta ve bu bağlantı devamlılık arz etmektedir. Bu bağlantı odaklı yaklaşım, ses ve video aktarımı için oldukça elverişli olmaktadır (Tablo 2.1). Ancak bu durum, düşük güç tüketimi gerektiren IoT cihazları uygun değildir.

Geliştirilen BLE teknolojisinde ise, sürekli bağlantıya gerek duyulmamaktadır. Veri aktarımı ya da alımı olmadığı zaman diliminde düşük güç tüketimi modunda beklemede kalınmaktadır. Veri aktarımında ise aralıklı ve düşük bant genişliği tercih edilerek, haberleşmenin tamamında düşük güç tüketimi sağlanmaktadır.

2.1.2 Wi-Fi

Wireless Fidelity'nin kısaltması olan ve Kablosuz Bağlantı Alanı olarak tanımlanan Wi-Fi teknolojisinde, elektronik cihazlar radyo dalgaları ile bir lokal ağa bağlanmaktadır. Bu sayede kısa mesafede herhangi bir kablolu sistemine ihtiyaç duyulmaksızın internet erişimi sağlanabilmektedir (Fan et al., 2022).

Wi-Fi teknolojisi, yüksek hızda veri aktarımı, düşük güç tüketimi ve oldukça çeşitli cihazlarla uyumluluk sağlaması sayesinde IoT sistemlerinin gelişiminde önemli bir rol oynamıştır. Aynı zamanda, cihazların merkezi bir erişim noktasına ihtiyaç duymadan bir ağda birbirleriyle iletişim kurmasını sağlayarak, IoT ağlarının güvenliğini ve esnekliğini de arttırmıştır.

Kablosuz internet bağlantısını sağlayan protokoller, Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers (IEEE)) tarafından belirlenen standartlara göre şekillenir. Bu standartların genel adı IEEE 802.11 olarak belirtilir (Khasanova, 2021).

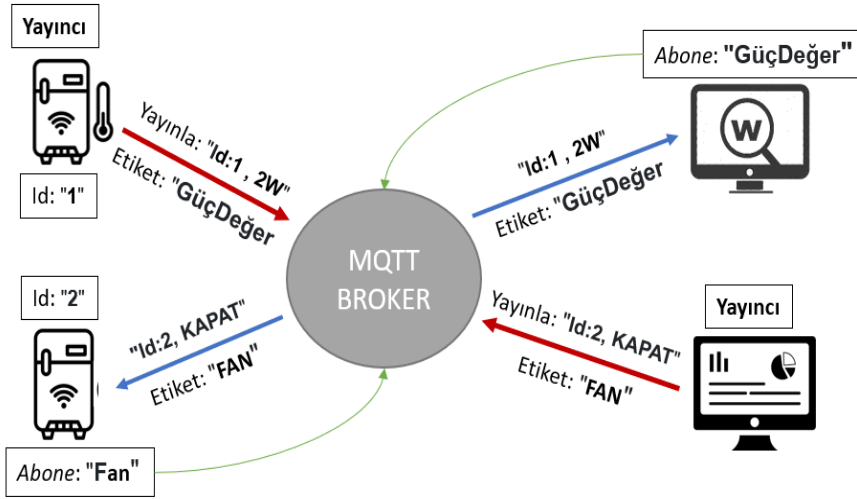
Tablo 2.2: IEEE 802.11 Wi-Fi Standartları (Gierłowski et al., 2011)

Standartlar	802.11a	802.11b	802.11g	802.11n
Frekans	5 GHz	2,4 GHz	2,4 GHz	2,4/5 GHz
Maksimum Hız	54 Mbps	11 Mbps	54 Mbps	600 Mbps
Kapsadığı Mesafe	100 metre	150 metre	150 metre	250 metre

Günümüze kadar birçok Wi-Fi standardı geliştirilmiştir. Bu tez çalışması kapsamında kullanılan ESP32-Wroom-32 modülünün desteklediği tüm standartlar Tablo 2.2'de ayrıntıları ile verilmiştir. Bu tabloda olmayan, daha yeni versiyon 802.11ac ve 802.11ax Wi-Fi standartları, bu çalışmada kullanılan ESP32 tarafından desteklenmemektedir.

2.1.3 MQTT (Message Queuing Telemetry Transport)

MQTT, IoT ve Makineden Makineye (Machine to Machine (M2M)) iletişimde kullanılmak üzere tasarlanmış hafif, açık kaynaklı ve yayınla-abone ol mesajlaşma modeli ile çalışan bir protokoldür (Hmissi & Ouni, 2022).



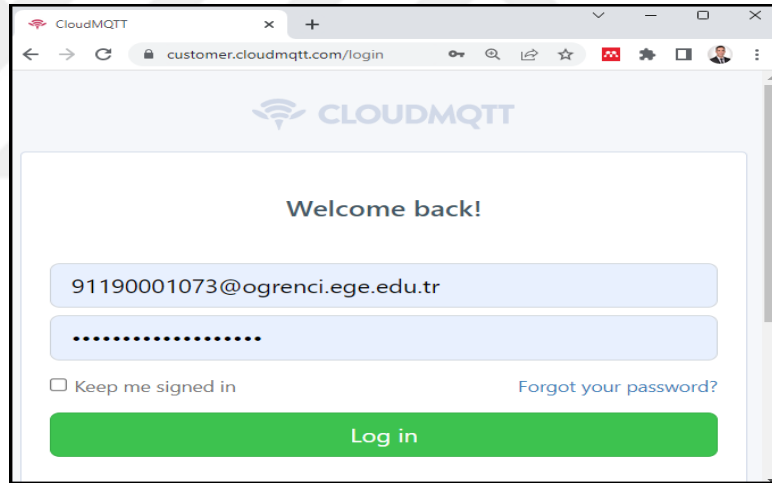
Şekil 2.1: MQTT sürecinin şematik gösterimi

Hafif olarak adlandırılmasının sebebi, düşük ağ bant genişliği ve sistem kaynaklarının verimli kullanılmasıdır (Mishra & Kertesz, 2020). Yayınla-abone ol mesajlaşma modelinde, istemciler belirli konulara yönelik iletileri belli bir etiket ile yayınlatabilir ve diğer istemciler iletileri almak için bu konu etiketlerine abone olabilir. Şekil 2.1’de gösterildiği gibi 1 numaralı ID değerine sahip buzdolabı, ”GüçDeğer” etiketine abone olan tüm kullanıcılara ”2 Watt” güç tüketim değerini göndermektedir. Diğer yandan, 2 numaralı ID değerine sahip buzdolabı abone olduğu ”Fan” etiketi vasıtası ile dolap içerisindeki fanın çalışıp çalışmayacağını öğrenebilmektedir.

Bahsedilen tüm bu haberleşme ağının ideal bir şekilde çalışabilmesi için bir MQTT aracısına (MQTT Broker) ihtiyacı vardır. Bu MQTT aracısının rolü ise, istemcilerden gelen mesajları almak ve onları uygun hedeflere yönlendirmektir (Longo et al., 2022).

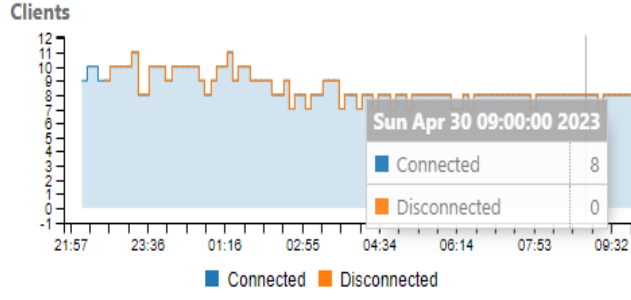
MQTT araçları sayesinde büyük çaplı projelerde kolaylıkla ölçeklendirme yapılabilmektedir. (Jutadhamakorn et al., 2017). Aynı anda birçok istemciden veri alabilmekte ve iletebilmektedirler. Bu görevini yerine getirirken güvenilirliği de arttırmak için mesajları birden çok sunucuya dağıtabilirler. Güvenilir mesaj teslimi sağlamak için ise genellikle kimlik doğrulama, şifreleme gibi özellikler de kullanılmaktadır.

MQTT araçları projenin yer aldığı sunucu içinde veya herhangi bir bulutta çalışabilmektedir. Ayrıca birçok bulut sağlayıcısı, yönetilebilen bir MQTT aracı hizmetleri sunmaktadır. Günümüzümüzde popüler olan bazı MQTT araçları Mosquitto, HiveMQ ve EMQ X'i olarak sıralabilir (Mehmood et al., 2019). Bu tez çalışması kapsamında, Mosquitto alt yapısını kullanan CloudMQTT aracı tercih edilmiştir.



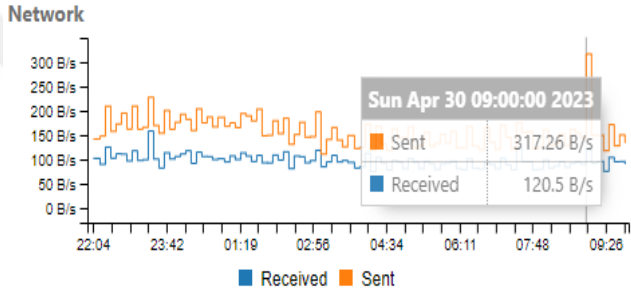
Şekil 2.2: CloudMQTT giriş sayfası

Şekil 2.2’de gösterildiği gibi, CloudMQTT kullanıcı arayüzü, kullanıcı adı ve şifresi korunmaktadır. Böylece sadece yetkili kullanıcılar bu arayüze erişebilmektedir. MQTT aracısının arayüzü üzerinden geçmişe yönelik bağlı olan cihazların sayısını, veri aktarım hızını ve mesaj sayılarını gözlemlemek mümkündür. Ayrıca web socket özelliği ile anlık olarak gelen verilerin görüntülenmesi de yapılabilmektedir. Ancak, anlık verilerin geçmişe yönelik kayıt mümkün değildir.



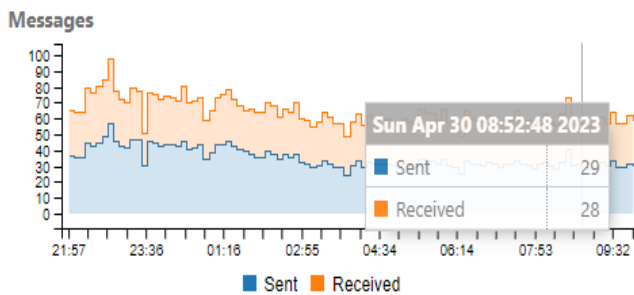
Şekil 2.3: CloudMQTT bağlı cihazlar

30 Nisan 2023 saat 09:00'da anlık bağlı olan cihaz sayısının 8 olduğu Şekil 2.3'de görülmektedir. Bağlantısı kopan cihaz ise bulunmamaktadır.

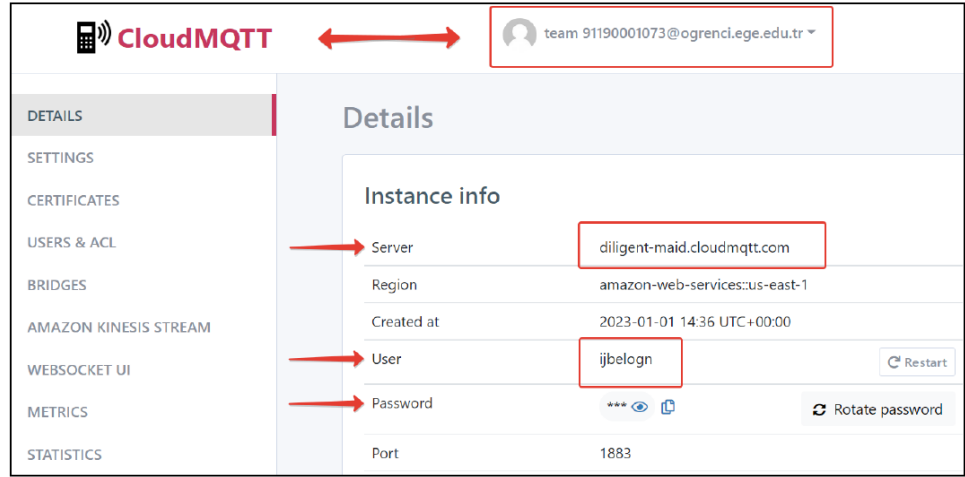


Şekil 2.4: CloudMQTT bağlantı hızları

Aynı zaman diliminde ağdaki veri aktarım hızının, gönderirken 317.26 B/s ve alırken 120.5 B/s olduğu Şekil 2.4'de incelenmektedir. Bu sırada, gönderilen mesaj sayısının 29 ve alınan mesaj sayısının 28 olduğu Şekil 2.5'de verilmiştir.



Şekil 2.5: CloudMQTT gönderilen ve alınan mesaj sayıları



Şekil 2.6: CloudMQTT bağlantı bilgileri

Yazılımlarda ise, MQTT aracısı üzerinden haberleşebilmek için Şekil 2.6’da gösterildiği gibi bir server adresi, port numarası, bağlantı için kullanıcı adı ve şifresi gereklidir. Yetkili kullanıcıların haricindeki kullanımı engelleyen tüm bu önlemlerin yanında MQTT aracısı üzerinden gerçekleşen haberleşme de AES şifreleme ile yapılmaktadır (Şekil 2.7).

Received messages			
Topic	Message	Flags	
SifreliOlcumBilgileri	y0kkkpqIt2YRdxq2FvPZRFezy7aHI/NW+tkqgkqMC6nODO0aib0OffefczbkRmUmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhKIHTiScTe5P9Lz7TFLPeQv	Q0	
SifreliOlcumBilgileri	soB6aOKbHvh96Bw4dUEJVN4CEAIAsWRwSInCR4+zTactJOUfRHHRWSDGFxOaWx9X6aKo5dWT+Oyadm6dl+tyo4wWkyh5TUYXUQI7NNG+P3h/yliwXieV/fpelcTKbjm	Q0	
SifreliOlcumBilgileri	RKFVXqLFIAWwUmHR8yiNLT/w/NtfxeD0KdX8rtVap4RaYmaWUwbE2q8hYb3k4jmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhJ8P1cB/DUuZP+gNihTpkXm	Q0	
SifreliOlcumBilgileri	y0kkkpqIt2YRdxq2FvPZRFezy7aHI/NW+tkqgkqMC6nODO0aib0OffefczbkRmUmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhKIHTiScTe5P9Lz7TFLPeQv	Q0	
SifreliOlcumBilgileri	soB6aOKbHvh96Bw4dUEJVN4CEAIAsWRwSInCR4+zTactJOUfRHHRWSDGFxOaWx9X6aKo5dWT+Oyadm6dl+tyo4wWkyh5TUYXUQI7NNG+P3h/yliwXieV/fpelcTKbjm	Q0	
SifreliOlcumBilgileri	RKFVXqLFIAWwUmHR8yiNLT/w/NtfxeD0KdX8rtVap4RaYmaWUwbE2q8hYb3k4jmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhJ8P1cB/DUuZP+gNihTpkXm	Q0	
SifreliOlcumBilgileri	y0kkkpqIt2YRdxq2FvPZRFezy7aHI/NW+tkqgkqMC6nODO0aib0OffefczbkRmUmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhKIHTiScTe5P9Lz7TFLPeQv	Q0	
CevrimdisiDolap	[{"AC100":{"SogutmaDurdu":"","CevrimdisiESPid":3,"status":0}]	Q0	
SifreliOlcumBilgileri	RKFVXqLFIAWwUmHR8yiNLT/w/NtfxeD0KdX8rtVap4RaYmaWUwbE2q8hYb3k4jmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhJ8P1cB/DUuZP+gNihTpkXm	Q0	
SifreliOlcumBilgileri	y0kkkpqIt2YRdxq2FvPZRFezy7aHI/NW+tkqgkqMC6nODO0aib0OffefczbkRmUmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhKIHTiScTe5P9Lz7TFLPeQv	Q0	
SifreliOlcumBilgileri	soB6aOKbHvh96Bw4dUEJVN4CEAIAsWRwSInCR4+zTactJOUfRHHRWSDGFxOaWx9X6aKo5dWT+Oyadm6dl+tyo4wWkyh5TUYXUQI7NNG+P3h/yliwXieV/fpelcTKbjm	Q0	
SifreliOlcumBilgileri	RKFVXqLFIAWwUmHR8yiNLT/w/NtfxeD0KdX8rtVap4RaYmaWUwbE2q8hYb3k4jmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhJ8P1cB/DUuZP+gNihTpkXm	Q0	
SifreliOlcumBilgileri	y0kkkpqIt2YRdxq2FvPZRFezy7aHI/NW+tkqgkqMC6nODO0aib0OffefczbkRmUmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhKIHTiScTe5P9Lz7TFLPeQv	Q0	
CevrimdisiDolap	[{"M400PLZ":{"SogutmaDurdu":"","CevrimdisiESPid":2,"status":0}]	Q0	
SifreliOlcumBilgileri	soB6aOKbHvh96Bw4dUEJVN4CEAIAsWRwSInCR4+zTactJOUfRHHRWSDGFxOaWx9X6aKo5dWT+Oyadm6dl+tyo4wWkyh5TUYXUQI7NNG+P3h/yliwXieV/fpelcTKbjm	Q0	
SifreliOlcumBilgileri	y0kkkpqIt2YRdxq2FvPZRFezy7aHI/NW+tkqgkqMC6nODO0aib0OffefczbkRmUmFHRjnGsfm898m6RMpu97RXHFUIOMcRgo+G3rGijuhKIHTiScTe5P9Lz7TFLPeQv	Q0	
SifreliOlcumBilgileri	soB6aOKbHvh96Bw4dUEJVN4CEAIAsWRwSInCR4+zTactJOUfRHHRWSDGFxOaWx9X6aKo5dWT+Oyadm6dl+tyo4wWkyh5TUYXUQI7NNG+P3h/yliwXieV/fpelcTKbjm	Q0	

Şekil 2.7: CloudMQTT verilerin yayınlanması

MQTT protokolünde gönderilen mesajların aracıya teslim edildiğinden emin olmak için Hizmet Kalitesi (Quality of Service (QoS)) özelliğinden yararlanılabilmektedir. MQTT protokollünde kullanılan 3 ayrı QoS seviyesi vardır (Sadeq, 2019).

- **QoS 0:** Bu QoS seviyede yalnızca bir kez teslimat yapılır. Mesajlar aracıya gönderilir ve abonelere ulaşıp ulaşmadığına dair bir onay alınmaz. Bu durumda abonelerden bir ya da birkaçı bağlı değilse mesajlar kaybolabilir.
- **QoS 1:** Bu yöntemde ise, mesajlar aboneye en az bir kez teslim edildiğinden emin olunur. Yöntem olarak, mesajlar aracıya gönderilir ve aracı tarafından onaylanır. Daha sonra aracı kendisine ulaşan mesajları bağlı olan tüm abonelere gönderir. Herhangi bir aboneden onay alınamazsa tekrar gönderim sağlanır ancak bu yinelenen mesajlara sebebiyet verebilir.
- **QoS 2:** Son hizmet kalitesi seviyesinde ise tam olarak bir kez teslimat gerçekleştiğinden emin olunur. Bu yöntemde bir öncekinden farklı olarak aracı aboneden onay geleşiye kadar gönderime devam eder.

QoS seviyesinin seçimi, uygulama gereksinimlerine ve ağ koşullarına bağlıdır (Gaikwad, 2021). QoS 0, arada meydana gelebilecek veri kayıplarının önemsiz olduğu ve ağ meşguliyetinin az olması gereken ve sık tekrar bilgisi olan yerlerde tercih edilmektedir. Bu yüzden Şekil 2.7'deki bayrak (Flag) kısmında gösterildiği gibi, bu çalışma kapsamındaki haberleşmede QoS 0 tercih edilmiştir. QoS 1, Mesaj kaybının yaşanmaması gereken ama yenilemenin kabul edilebilir olduğu uygulamalarda kullanılabilir. Diğer yandan QoS 2 ise, mesaj kaybının ve mesaj tekrarının asla kabul edilmediği sistemlerde tercih edilir. Ancak, bu uzun doğrulama sürecinin sonunda ağ trafiği oldukça artmaktadır. Bu yüzden hassas konumlama ve askeri işler gibi hatayayer verilmeyen uygulamalarda tercih edilmektedir.

Ayrıca, MQTT araçlarının desteklediği bazı ek özellikler vardır. Bunlardan birisi olan muhafaza özelliği sayesinde, yeni bağlanan bir aboneye daha önce yayınlanmış bir mesajı almalarına izin veren tutma biti (retain bit) aktif edilebilir (Wu & Li, 2018). Yayıncı tarafından tutma biti 1'e ayarlandığında MQTT aracısı mesajı depolar ve konunun yeni abonelerine iletir.

Diğer bir özellik olan mesaj kalıcılığı (persistence) ise, bir aracının o anda bağlı olmayan istemciler için iletileri depolamasına izin veren MQTT protokolünün bir özelliğidir. Kalıcılık özelliği, mesajların zorlu ağ ortamlarında bile istemcilere güvenilir bir şekilde teslim edilmesini sağlamak için Hizmet Kalitesi (QoS) seviyeleri ve saklama bayrağı gibi diğer MQTT özellikleriyle birlikte kullanılabilir (Luna et al., 2018).

MQTT protokünde "Son İstek ve Vasiyetname" (last will and testament) özelliğinin aktif edilmesi ise, istemcinin aracıyla bağlantısının beklenmedik bir şekilde kesilmesi durumunda, bir etiket vasıtası ile yayınlanacak bir mesajı belirtmesine izin verir. Örneğin, bu çalışma kapsamında bir dolabın verilerinin yayınlaması sırasında, ağda meydana gelebilecek herhangi bir bağlantı kesintisi durumunda, tüm sistemin uyarılması için çok önemli özelliktir. Şekil 2.7'de, MQTT aracısının arayüzünün web socket bağlantısı sayesinde görüntülenen iki ayrı çevrim dışı bilgisi kırmızı çerçeve ile belirtilmiştir. Şifreli haberleşmede yer alan bu iki açık metine örneğin daha anlaşılır olabilmesi için yer verilmiştir.

Son olarak, "Canlı Tut" (Keep Alive) özelliği, istemci ve MQTT aracısı arasındaki bağlantının hala aktif ve düzgün çalışmasını sağlamak için düzenli aralıklarla birbirleriyle iletişim kurmasını sağlayan bir özelliktir (Ajitomi et al., 2015). Canlı Tut özelliği, MQTT aracısının, artık aktif olmayan veya yanıt vermeyen bağlantıları kapatarak kaynakları etkili bir şekilde yönetmesine olanak tanır. Bu durum, gereksiz ağ trafiğini önlemeye yardımcı olur ve istemcilerin aracıyla verimli ve güvenilir bir şekilde bağlantı kurmasını ve iletişim kurmasını sağlar.

2.1.4 HTTP (Hyper-Text Transfer Protocol)

Hiper Metin Transfer Protokolü (Hypertext Transfer Protocol(HTTP)) internet üzerinden veri aktarımı için kullanılmaktadır. Temel anlamda, internet erişiminde istemciler ve sunucular arasında iletişim için köprü görevindedir (Fidaeldin & Abdulrazzaq, 2019).



Şekil 2.8: HTTP komutları aracılığı ile iletişim

Veri gönderimi ve alımı için, kaynağı gösteren bir tekdüzen kaynak bulucu (Uniform Resource Locator (URL)) ile HTTP isteği yapılır. GET, POST, PUT, DELETE ve da birçok farklı HTTP istek komutu vardır (Roig et al., 2019). Her HTTP isteği belli bir amaca yönelik kullanılır. Örneğin, sunucudan bilgi almak istendiğinde GET komutu kullanılırken, veri göndermek veya kaydettirmek için POST komutu komutu kullanılır (Şekil 2.8). PUT komutu ise daha çok düzenleme işlemlerine hizmet eder.

Sunucular ise, bu gelen isteklere gerekli olan yanıtları üretir ve gönderirler. Genelde bir HTTP yanıtı, isteğin başarılı ya da başarısız olduğunu belirten bir durum kodu, yanıt hakkında ek bilgi sağlayan içerikler ve sunucu tarafından cevaplanan verileri içerir.

Bu tez çalışması kapsamında, HTTP istekleri özel bir sunucuda bulunan bir API ile iletişim için kullanılır. Sadece veritabanına kayıtlı yetkili kişilerin API erişimi bulunmaktadır. Böylece veritabanında kayıtlı olan tüm bilgiler, yetkili kişiler tarafından API aracılığı ile elde edilir. Veritabanında kayıt işlemleri ve kayıt güncelleme terar düzenleme işlemleri de yine aynı yöntemle sağlanır.

2.1.5 JSON Mesaj Formatı

JavaScript Nesne Gösterimi (JavaScript Object Notation (JSON)) hafif bir veri değişim formatı anlamına gelir. Bu veri değişimi, insanların daha kolay okuması/yazması ve makineler için verileri daha kolay ayrıştırması bakımından tercih edilir. Farklı uygulamalar arasında veri alış verişi sırasında basit ve standart bir metin formatı uygulanmış olmaktadır.

Bir JSON mesajının temel yapısı, ad/değer çiftleri veya anahtar/değer çiftleri koleksiyonundan oluşur. Bahsedilen ad veya anahtar bir dizedir ve değer değişkeni ise, bir sayı, boolean, dizi veya başka bir JSON nesnesi olabilir. İsim/değer çiftleri süslü parantez içine alınır ve her çift bir virgülle ayrılır.

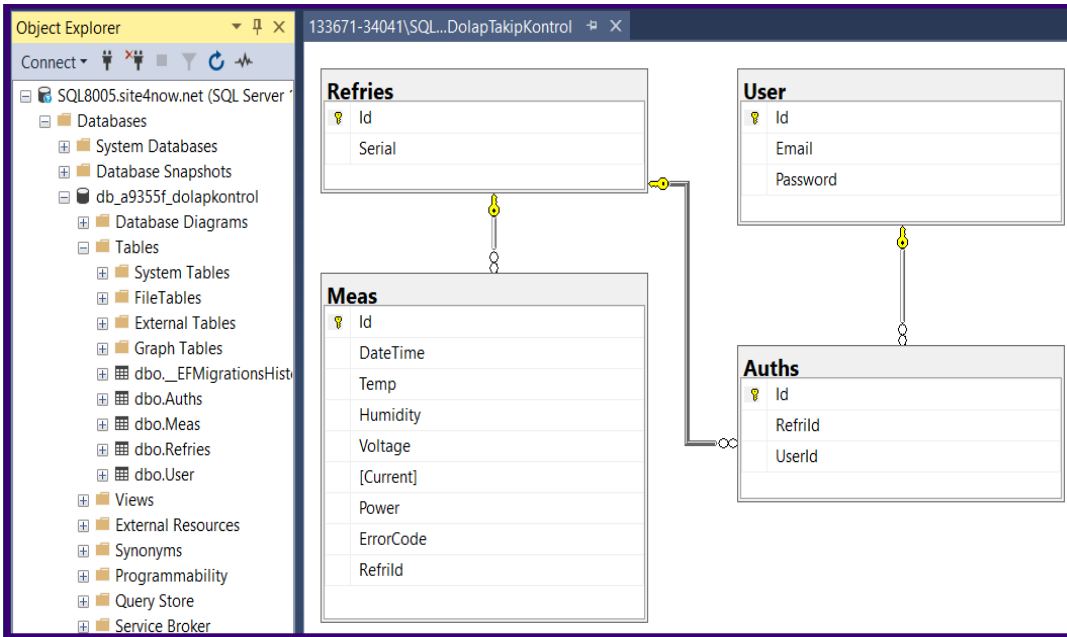
```
[
{
  "id": 1, "dolapId": 2, "kullaniciId": 3,
  "dolap": { "id": 2, "SeriNo": "45430001210520180002" }
  "kullanici": { "id": 3, "e-posta": "deneme3@den.com", "sifre": "123" }
}
]
```

Yukarıda yer alan JSON mesaj formatının içeriğinde, 2 numaralı Id değerine sahip buzdolabının tüm bilgilerine ulaşılabilmektedir. Bu mesajda, 5 adet ad/değer çifti bulunmaktadır. "id", "dolapId" ve "kullaniciId" adları basit değerleri temsil ederken, "dolap" ve "kullanici" bir diziyi temsil eder. Ayrıca "dolap" anahtarı, bir dolabı temsil eden iki ek ad/değer çifti içeren başka bir JSON nesnesini temsil eder. "kullanici" anahtarı ise, bir kullanıcıyı temsil eden üç ek ad/değer içerir.

JSON mesaj formatı bu tez çalışması kapsamında veritabanına veri yüklerken veya veri alırken bir önceki konuda bahsedilen HTTP istekleri sırasındaki mesaj trafiğinde kullanılmaktadır.

2.1.6 Veritabanı

Veritabanları, elde edilen bilgilerin uygun şekilde depolanmasına ve yönetilmesini sağlayan veriler topluluğudur. Basit tablolardan oluşabileceği gibi çok büyük sistemlerdeki karmaşık yapıların biçimlerine de uyarlanabilir. Veritabanları lokal bir bilgisayarda olabileceği gibi, özel sunucularda da tutulabilmektedir. Bu çalışmada özel bir sunucuda bulunmaktadır. Günümüzde, ilişkisel veritabanları, NoSQL veritabanları, nesne yönelimli veritabanları ve grafik veritabanları gibi birçok farklı veritabanı türü vardır.



Şekil 2.9: İlişkisel veritabanı

En yaygın veritabanı olan ve bu çalışmada da tercih edilen ilişkisel veritabanlarında, veriler satır ve sütunlardan oluşan tablolar halinde düzenlenir. Verileri işlemek ve yönetmek için, Yapılandırılmış Sorgu Dili (Structured Query Language (SQL)) kullanılır. Veri tabanı tablolarının ilişkilendirilmeleri de yine SQL ile sağlanır (Şekil 2.9). Örneğin "Ölçümler" tablosu ve "Buzdolapları" tablosunun olduğu bir veritabanında belli bir dolabın ölçümlerine bu sayede ulaşılabilmektedir.

İlişkisel veritabanları, büyük miktardaki verinin verimli bir şekilde depolanması, erişilebilmesi ve yönetilebilmesi için kullanılmaktadır. Kullanıcıların CREATE, INSERT, UPDATE ve DELETE gibi komutları kullanarak veritabanları oluşturmaya ve düzenlemesine olanak tanır. Ayrıca SELECT, WHERE ve JOIN gibi komutlarla verilerin sorgulanması, filtrelenmesi, gruplanması ve alınması da sağlanabilmektedir.

Veritabanları, depolanan verileri yönetmek için araçlar sağlayan ve bir yazılım uygulaması olan veritabanı yönetim sistemleri (Database Management System (DBMS)) kullanılarak yönetilebilir. Bu çalışma boyunca veritabanı kontrol ve denemeleri bu arayüz üzerinden test edilmiştir. Örneğin, aşağıda yer alan SQL sorgusu SQL Server DBMS üzerinden "ölçümler" tablosunun ilgili değişkenlerine göre ilk 15 satırı çağrılmaktadır.

```
SELECT TOP (15) [Id]
, [TarihSaat]
, [Sıcaklık]
, [Nem]
, [Gerilim]
, [Akım]
, [Güç]
, [HataKodu]
, [DolapId]
FROM [DolapTakip].[dbo].[ölçümler]
```

Tablo 2.3'te, yukarıda yer alan sorgunun cevabı olarak, ilişkisel veritabanının bir örneği yer almaktadır. Görüldüğü gibi tüm ölçüm değerleri zaman ve dolabın Id değerine göre listelenmektedir. Tabloda, Id değerlerinden anlaşılacağı gibi 3 ayrı dolap bulunmaktadır. Bu biçimlendirme özelliği uygulamaların yapabileceği sorgulama yeteneği arttırmaktadır.

Tablo 2.3: Veritabanı ölçümler tablosunun listelenmesi

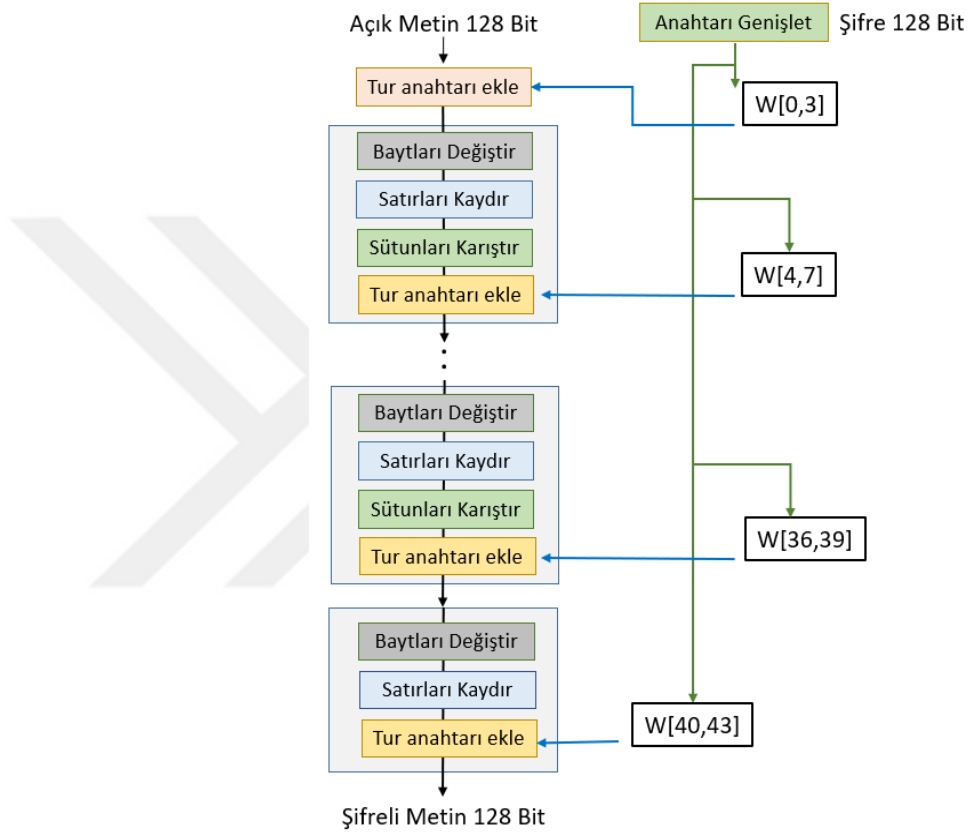
dbo.ölçümler							
<i>TarihSaat</i>	<i>Sıcaklık</i>	<i>Nem</i>	<i>Gerilim</i>	<i>Akım</i>	<i>güç</i>	<i>Hata</i>	<i>D.Id</i>
22.04.20..	4,3	48,1	233,1	0,5	116,5	0	1
22.04.20..	5,1	49,3	232,4	1,3	312,1	1	2
22.04.20..	-8,3	50,2	233,5	3,1	723,8	0	3
22.04.20..	4,1	49,5	234,2	0,7	163,9	0	1
22.04.20..	6,5	48,4	232,3	1,5	348,4	3	2
22.04.20..	-8,6	48,6	233,3	3,2	746,6	0	3
22.04.20..	3,2	49,1	231,6	0,6	138,9	0	1
22.04.20..	4,4	50,3	235,5	1,7	400,3	4	2
22.04.20..	-9,1	51,4	230,4	3,3	760,3	0	3
22.04.20..	4,3	52,1	232,7	0,5	116,3	0	1
22.04.20..	4,3	52,1	232,7	0,5	116,3	0	1
22.04.20..	4,3	52,1	232,7	0,5	116,3	0	1
22.04.20..	4,3	52,1	232,7	0,5	116,3	0	1
22.04.20..	4,3	52,1	232,7	0,5	116,3	0	1
22.04.20..	4,3	52,1	232,7	0,5	116,3	0	1

2.2 Güvenlik

Gelişen IoT sistemlerinde en önemli konu başlıklarından birisi güvenlidir. Kişisel veri gizliliği ya da kötü niyetli kişilerin sisteme erişimini engelleme gibi konular üzerinde birçok çalışma bulunmaktadır. Bu tez çalışması kapsamında, MQTT haberleşmesindeki verilerin gizliliği için AES-128 şifreleme algoritmaları kullanılmıştır. Veritabanının güvenliği için ise, Taşıyıcı Kimlik Doğrulaması tercih edilmiştir. Böylece sadece yetkili kişiler, veritabanına bir API vasıtasıyla erişebilmektedir.

2.2.1 AES (Advanced Encryption Standard) Şifreleme

Gelişmiş Şifreleme Standardı (Advanced Encryption Standard (AES)) hassas verileri şifrelemek için verimliliği ve esnekliği nedeniyle, günümüzde en çok tercih edilen simetrik şifreleme algoritmasıdır. 128 bitlik bir düz metin bloğu ve 128, 192 veya 256 bitlik değişken uzunluklu bir anahtar üzerinde çalışır.



Şekil 2.10: AES şifreleme mimarisi

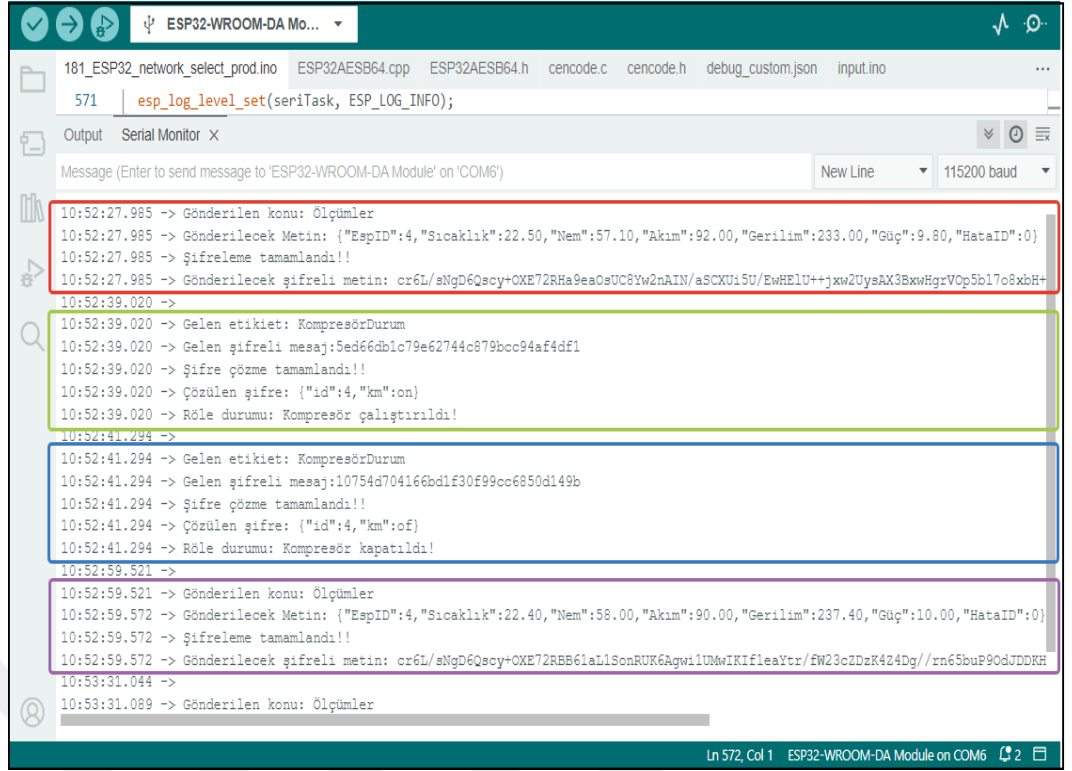
AES-128 şifreleme algoritmasında, açık metin birçok değiştirme ve permütasyon işlemine tabi tutularak şifreli metne dönüştürülür. Aşağıda yer alan algotirma basamaklarında işleyiş hakkında ayrıntılı bilgilere yer verilmiştir.

- **Anahtar Genişletme:** AES-128 algoritması, orijinal 128 bitlik anahtarı, bir anahtar programı vasıtasıyla daha büyük bir döngüsel anahtar grubuna genişleterek başlar. Bu işlem sonucunda her biri 128 bit uzunluğunda olan toplam 11 döngüsel anahtar üretilir.

- **İlk Tur:** Açık metin, Şekil 2.10'de gösterildiği gibi 4x4 matrisinde düzenlenen 16 baytlık bloklara bölünür. İlk şifreleme turu, durum dizisinin her baytının ilk döngüsel anahtarla "Özel VEYA" (Exclusive OR (XOR)) işleminin uygulandığı bir "Tur anahtarı ekle" adımıyla başlar.
- **1-9. Turlar:** Geriye kalan 9 turda, üç adımın kombinasyonları kullanılır. Bunlar sırasıyla baytları değiştir, satırları kaydır, sütunları karıştır olarak uygulanır. Baytları değiştir adımında, dizisinin her baytı, doğrusal olmayan bir dönüşüm kullanılarak bir ikame kutusundan (S-kutusu) karşılık gelen bir değerle değiştirilir. Satırları karıştır adımında, dizisinin satırları döngüsel olarak farklı miktarlarda kaydırılır. Kolonları karıştır adımında, dizisinin her bir sütunu, yeni bir değer üretmek için sabit bir polinomla çarpılır.
- **Son Tur:** Aslında son turda ilk tura benzer. Sadece kolonları karıştır adımı uygulanmaz. Durum dizisi önce "baytları karıştır" ve "satırları karıştır" adımlarına tabi tutulur ve ardından son döngüsel anahtarla mantıksal XOR mantıksal işlemi uygulanır.
- **Şifreli Metin:** Son turda uygulandıktan sonra, dizi tekrar bir bayt dizisine dönüştürülür ve şifreli metin olarak çıktı verir.

Genel olarak, AES-128 algoritması, şifreli metnin rastgele görünmesini ve kriptanaliz saldırılarına dirençli olmasını sağlamak için doğrusal ve doğrusal olmayan dönüşümlerin bir kombinasyonunu kullanır.

Şekil 2.11'de gösterilen seri port çıktısında, bir açık metnin şifreli metine dönüştürülmesi ve şifreli metnin açık metine dönüştürülmesine yer verilmektedir. Şifreleme işlemlerinde, önceki konularda ayrıntılarına değinilen JSON veri formatı kullanılarak, tek seferde tüm ölçümlerin şifrlenmesi sağlanmıştır. Bu sayede şifreleme ve deşifreleme işlemlerinde herhangi bir zaman kaybı olmadığı ve teoride belirtildiği gibi mikrosaniyeler içinde işlemin tamamlandığı görülebilmektedir.



Şekil 2.11: Aktarılabilecek verilerin şifrelenmesi

2.2.2 Taşıyıcı Kimlik Doğrulaması

Yetkisiz kullanıcıların sisteme erişimi, geliştirilen IoT uygulamaları için büyük bir tehdit unsurudur. Bu durumu önlemek için, akıllı uygulamalarda kimlik doğrulaması yapılması sistemi oldukça güçlendirmektedir. Bir çok kimlik doğrulaması yöntemi olmakla birlikte, bu çalışma kapsamında JWT (JSON Web Token) tercih edilmiştir.

JWT, taraflar arasında JSON formatında güvenli bilgi aktarımı için kullanılabilen bağımsız bir belirteç olarak tanımlanır. Yöntemin işleyişi, oldukça basittir. Sistemden JWT token üretilebilmesi için kullanıcı kendisine ait bilgilerle bir oturum açmalıdır. Eğer kullanıcı bilgileri doğrulanırsa, sistem tarafından bir JWT kodu üretilir. Bu adımdan sonra kullanıcı tarafından sunucuya yapılacak tüm istekler, bu JWT kodu ile sağlanır. Eğer kullanıcı tarafından gönderilen JWT yanlış ise, sunucu "Yetkisiz Kullanıcı" cevabı döndürür. Bu durum, ya oturum süresi dolduğu ya da kullanıcı bilgilerinin doğru olmadığı anlamına gelir.

Bir JWT belirteci üç bölümden oluşur. Bunlar sırasıyla, başlık (Header), yük (Payload) ve imzadır (Signature). Her bir bölümün birbirinden ayrılması için nokta (.) karakteri kullanılır.

- **Başlık:** Başlık kısmında , JWT türü ve imzalama algoritması tanımlanır. JSON formatında kodlanır ve aşağıdaki yer aldığı gibi görünür.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

JSON formatında görüldüğü gibi "alg" imzalama algoritmasını (HMAC-SHA256) gösterir, "typ" ise, JWT'nin türünü ("JWT") belirtir.

- **Yük:** JWT'nin barındırdığı verileri bulundurur. Yük kısmında da başlık olduğu gibi JSON formatı kullanılır ve claims adı verilen verileri içerir. Claims, JWT'yi kullanan uygulamaların birbirleriyle iletişim kurmasına yardımcı olur. Örnek bir payload şu şekildedir:

```
{
  http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress:
  "91190001073@ogrenci.ege.edu.tr",
  exp: 1714490712,
  iss: "https://localhost:7094/",
  aud: "https://localhost:7094/"
}
```

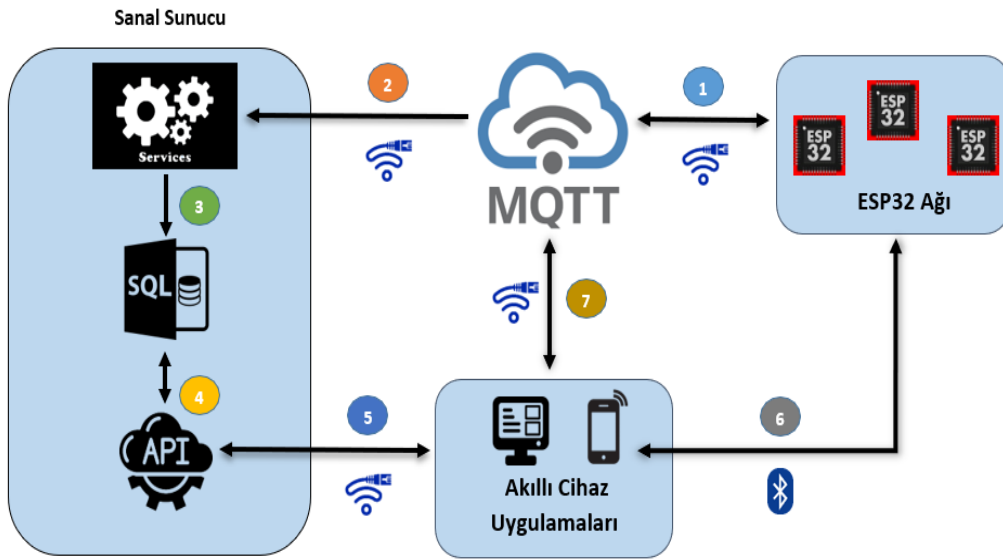
Yukarıda yer alan JSON formatındaki zaman aşımı tarihi (Expiration Time (Exp.)), yetkilendirmenin sonlanacağı süreyi gösterir. Yayınlayıcı (Issuer (Iss.)) ise, kimin tarafından oluşturulduğunu ve kimin tarafından imzalandığını belirtir. Dinleyici (Audience(Aud.)) kimin için oluşturulduğunu gösterir.

3 SİSTEM TASARIMI

Bu tez çalışması kapsamında tasarlanan IoT sistem mimarisinde, tüm buzdolaplar Wi-Fi ya da bluetooth (BLE) bağlantısı üzerinden bilgi aktarabilmektedir (Şekil 3.1). İsteğe bağlı olarak, iki haberleşme protokolünün aynı anda kullanılabilmesi de mümkündür. Wi-Fi bağlantısı üzerinden aktarılan bilgiler, masaüstü ve web sitesi vasıtasıyla kullanılabilirken, bluetooth bağlantısı üzerinden gelen veriler, mobil uygulama aracılığı ile görüntülenmektedir.

3.1 Genel Sistem Mimarisi

IoT sistemi içerisindeki dolapların, Wi-Fi ağına bağlantısı gerçekleştirildiğinde, ESP32 modülü üzerinden, bir MQTT aracısına bağlantı sağlanmaktadır. Diğer yandan, bu çalışmada geliştirilen ve özel bir sunucuda bulunan Windows Servis uygulamasının da MQTT aracısına bağlantısı ve ilgili konulara abone olması ile elde edilen tüm veriler, veritabanına kayıtlanmaktadır. Veritabanı güvenliğini sağlamak adına, tasarlanan akıllı uygulamaların veritabanına direk erişimi tercih edilmemiştir. Bu yüzden bir API yazılımı geliştirilmiş ve kullanıcı yetkilendirilmesine dayalı veri erişimine izin verilmiştir.



Şekil 3.1: Tasarlanan sistemin genel mimarisi

Şekil 3.1’de gösterilen ve 1 numaralı işaretçinin olduğu bölgede, buzdolapları Wi-Fi bağlantısıyla MQTT aracısına bağlanmaktadır (Şekil 3.1). Buzdolabı içerisindeki sensör modüllerinden alınan dolap iç kabin sıcaklığı, nemi, çalışma gerilimi, kullandığı akımı ve tüketilen güç değerleri, JSON formatına çevrildikten sonra, bir bütün halinde AES-128 şifreleme algoritmasıyla şifrelenir. Daha sonra istenilen süre aralığında, MQTT aracısına gönderimi sağlanır. Herhangi bir hata durumunda ise, süre beklenmeksizin anlık gönderim sağlanır.

Windows servis uygulamasının MQTT aracısına bağlantısı ise, 2 numaralı işaretçi ile gösterilmiştir. Windows servis uygulamasının veritabanına kesintisiz bağlantı sağlayabilmesi için özel bir sunucuda bulunmaktadır ve sürekli dinleme modundadır.

Numara 3 ile işaretlenmiş alanda ise, Windows Servis uygulamasıyla deşifre edilmiş hassas bilgiler, veritabanına kayıt edilmektedir. 4 ve 5 numaralı bölgede ise akıllı uygulamalar, geliştirilen API uygulamasına kullanıcı adı ve şifresi ile HTTP isteğinde bulunurlar. Verilen bilgiler API tarafından veritabanında sorgulanır ve eğer doğru ise sisteme girişe izin verilir.

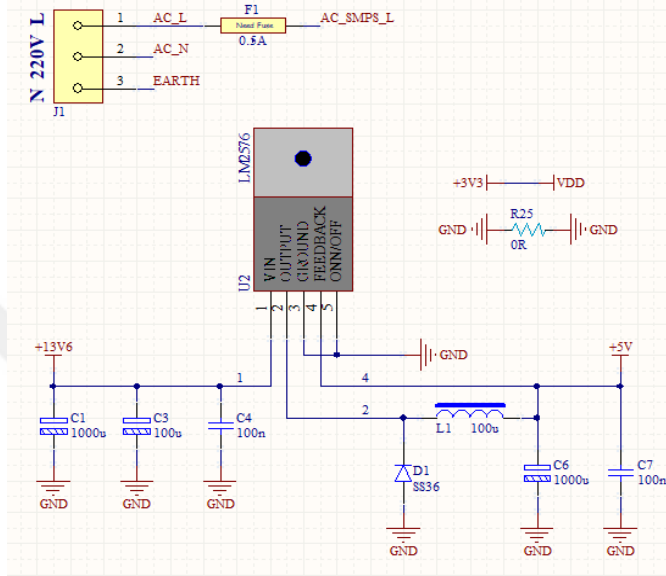
Bluetooth Low Enerji (BLE) teknolojisi ile Wi-Fi bağlantısı olamayan yerler için 6 numarada ile gösterilen bir mobil uygulama geliştirilmiştir. Bu sayede, istenilen tüm veriler yakın mesafeden son kullanıcıya aktarılabilir.

3.2 Donanım Mimarisi

Bu tez çalışması kapsamında, mevcut ve yeni üretilen buzdolaplarında kullanılabilecek bir dizi donanım geliştirilmiş ve entegre edilmiştir. Güç ve kontrol kartı, özel tasarım olarak geliştirilirken, dokunmatik LCD olarak günümüzde oldukça popüler olan Nextion markasının NX4827P043-011C modeli tercih edilmiştir. Ayrıca, sistemde Elektromanyetik Girişim (Elektromagnetic Interference (EMI)) önlemek için EMI filtreler ve ferrit nüveler kullanılmıştır.

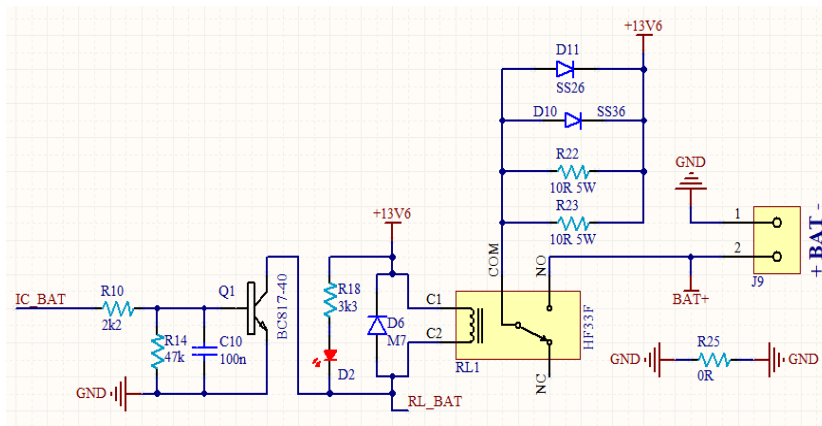
3.2.1 Güç ve Kontrol Kartı Tasarımı

Elektronik kart, voltaj düşürücü ve dönüştürücü kısmı, ESP32-Wroom-32 modül kısmı, ölçüm modüllerinin bağlantı soketleri ve son olarak röle anahtarlama kısmından oluşmaktadır.



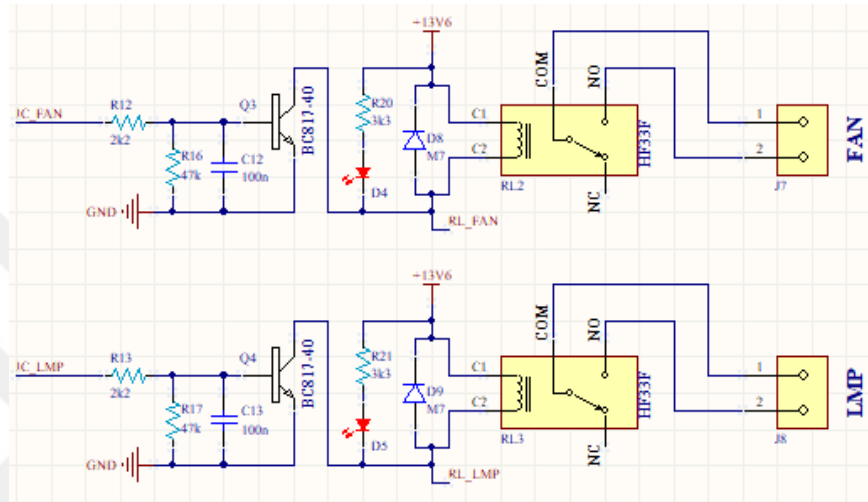
Şekil 3.2: Elektronik kartın regüle kısmı

Elektronik kartın girişine uygulanan 220/240VAC gerilimi, Anahtarlama Güç Kaynağı (Switch Mode Power Supply (SMPS)) vasıtasıyla önce 15VDC seviyesine dönüştürülmektedir. Ardından LM2576 regüle devresi ile çevre birimlerinin gerilimi olan 5VDC seviyesine düşürülmektedir (Şekil 3.2).



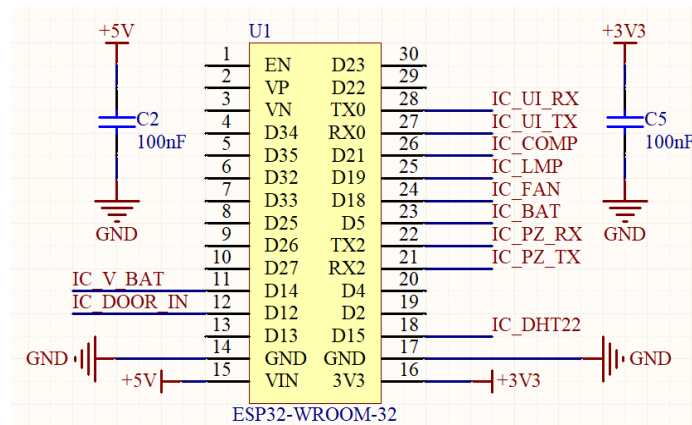
Şekil 3.3: Akü anahtarlama kısmı

220/240VAC besleme kesintilerine karşı, sistemde yedek bir 12VDC batarya bulunmaktadır. Bu bataryanın sistem soğutmasına bir etkisi bulunmamasıyla birlikte, ölçümlerin devam etmesine, LCD panel üzerinden görüntülenmesine ve eğer internet ağı hala aktif ise bilgi gönderilmesini sağlamaktadır. Dolap enerjisi tamamen kesilmek istendiğinde, akü rölesi vasıtasıyla, LCD panel üzerinden kontrol edilebilmektedir (Şekil 3.3).



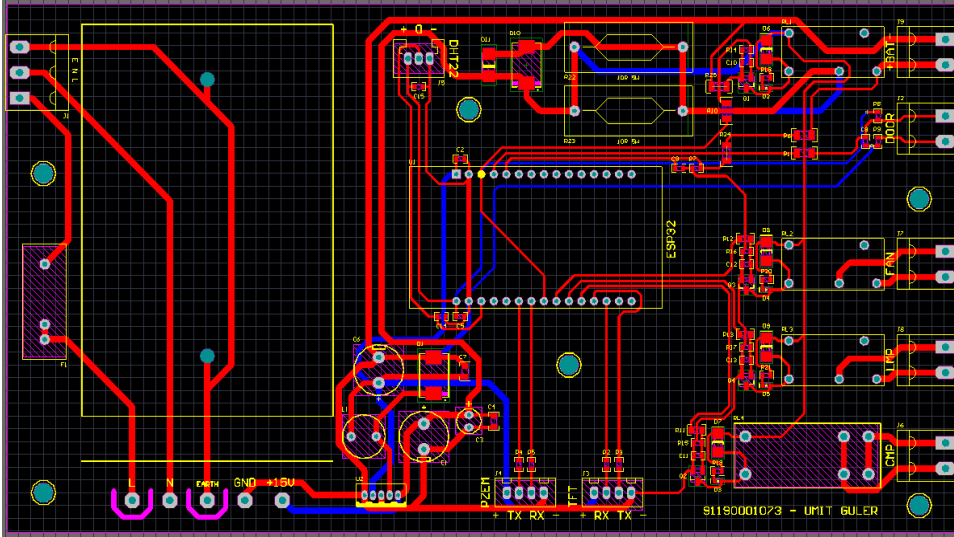
Şekil 3.4: Fan ve lamba röle anahtarlama kısmı

Şekil 3.4’de görüldüğü gibi Kompresör, fan ve lamba gibi komponentler için de röle anahtarlama kullanılmaktadır. Kompresör ve fan soğutma algoritmasına göre anahtarlanırken, lamba LCD panel üzerinden kontrol edilebilmektedir.



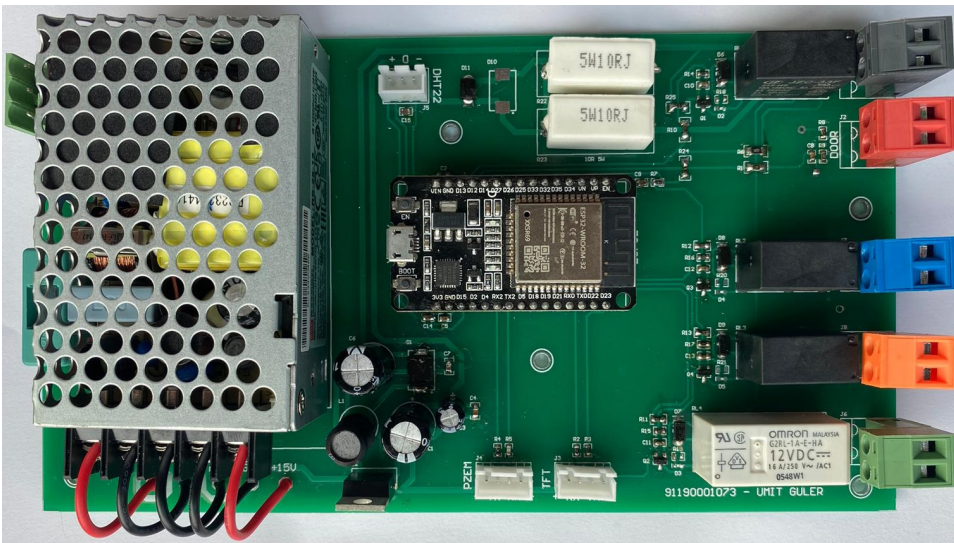
Şekil 3.5: ESP32 modül bağlantı kısmı

Devre yazılımının yürütüldüğü ESP32-Wroom-32 modülünün herhangi bir arıza durumuna bağlı olarak, kolay değiştirilebilmesi ve progranlanabilmesi için Şekil 3.5’de görüldüğü gibi, modül olarak kullanılması ve ilgili sokete takılması tercih edilmiştir.



Şekil 3.6: PCB - Tasarım

Şekil 3.6’da görüldüğü gibi, baskı devre tasarımında çift katlı tasarım kullanılmıştır. Böylece çok daha uygun bir boyut elde edilmiştir(Şekil 3.7).

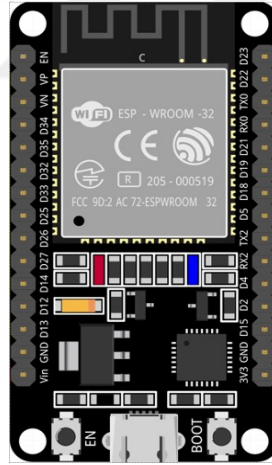


Şekil 3.7: PCB - Final

3.2.2 Kullanılan Modül ve Sensörler

Sistem tasarımımda, mümkün olduğunca stratejik bir yaklaşım yürütülmüştür. Testler, arıza durumları ve gerekli değişimler düşünülerek bazı malzemeler modül olarak kullanılmıştır. Örneğin ESP32, PZEM-004T ve DHT sensör kolay entegre edilebilir, taşınabilir ve değiştirilebilir olması çalışmanın ileriye yönelik uygulamaların çok daha esnek ve ölçeklendirilebilir bir yapıya sahip olmasını sağlamıştır.

Tasarlanan sistemin en önemli bileşeni ve tüm gömülü yazılımının yürütüldüğü modül olan ve Şekil 3.8’de gösterilen ESP32-Wroom-32, çift çekirdekli bir işlemciye sahiptir. Ayrıca, önceki bölümlerde ayrıntıları ile anlatılan Wi-Fi ve Bluetooth (BLE) gibi kablosuz iletişimler hibrit olarak tek bir modül üzerinden kullanılabilir. ESP32-Wroom-32 modülü üzerinden kullanılacak bu kablosuz protokollerin ayrıntılarına Tablo 3.1’de yer verilmiştir.



Şekil 3.8: ESP32-Wroom-32 modülü

Çevre birimlerinin arasında bulunan 3 adet Evrensel Asenkron Alıcı Verici (Universal Asynchronous Receiver Transmitter (UART)) arayüzlerinin birincisi LCD haberleşmesinde, ikincisi PZEM-004T enerji ölçüm modülünde kullanılmaktadır.

Tablo 3.1: ESP32-Woom-32 özellikleri (Espressif Systems, 2023)

Özellikler	Açıklamalar
Mikroişlemci	Xtensa Dual-Core-bit LX6
Frekans	240 MHz'e kadar
Wi-Fi	802.11 b/g/n
Bluetooth	Bluetooth v4.2 BR/EDR ve BLE
3 x UART	UART arayüzü
18 x Kanal ADC	12 bit çözünürlük
2 x DAC	8 bit çözünürlük
2 x I2C	I2C arayüzü
4 x SPI	SPI arayüzü

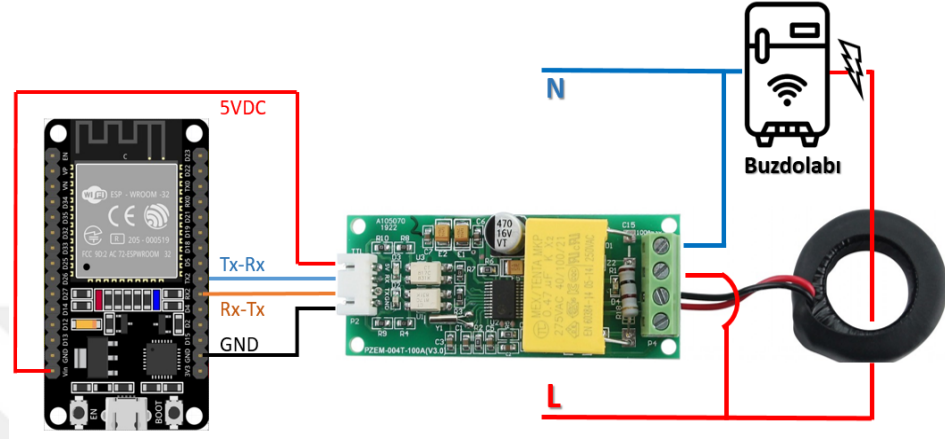
Bu çalışma kapsamında kullanılan modüllerden birisi olan PZEM-004T V3.0 ise tek fazlı AC devrelerde gerilim, akım, güç, enerji, frekans ve güç faktörü ölçmek için kullanılan bir modüldür.

Genellikle güç izleme ve enerji yönetimi gibi uygulamalarda tercih edilen PZEM-004T V3.0 ölçüm parametre özellikleri Tablo 3.2'de verilmiştir. Görüldüğü gibi 80-260VAC gerilim aralığında, 100 Amper ve 22 KW'a kadar güç tüketimi ölçülebilmektedir. Ayrıca hassasiyetin %1 olması çok daha kararlı bir ölçüm imkanı sunmaktadır.

Tablo 3.2: PZEM-004T V3.0 özellikleri (Peacefair Electronics, 2021)

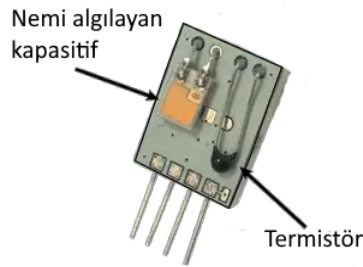
Parametre	Ölçüm Aralığı	Çözünürlük	Hassasiyet
Gerilim	80-260 VAC	0,1 V	$\pm$ %0.5
Akım	0-100 A	0.01 A	$\pm$ %0.5
Güç	0-23 kW	0.1 W	$\pm$ %0.5
Enerji	0-9999.99 kWh	1 kWh	$\pm$ %0.5
Frekans	45-65Hz	0.1 Hz	$\pm$ %0.5
Güç faktörü	0.00-1.00	0.01	$\pm$ %0.1

Bu çalışmada, buzdolaplarının tüketim parametrelerini ölçmek için kullanılan PZEM-004T V3.0 bağlantısı Şekil 3.9’da verilmiştir. Görüldüğü gibi, enerji kablolarından birisine takılan bir bobin vasıtasıyla tüm ölçüm parametreleri alınabilmektedir.

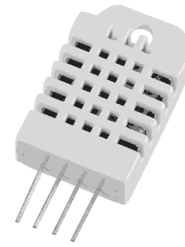


Şekil 3.9: PZEM-004T V3.0 modül bağlantısı

PZEM-004T V3.0 ve ESP32-Wroom-32 modülünün ara bağlantısı UART çevre birimi üzerinden gerçekleştirilmektedir. Bunun için bir ara bağlantı kabosu kullanılmak ve bu bağlantı ferit nüveler ile elektriksel gürültülere karşı korunmaktadır.



(a) DHT22 iç yapısı



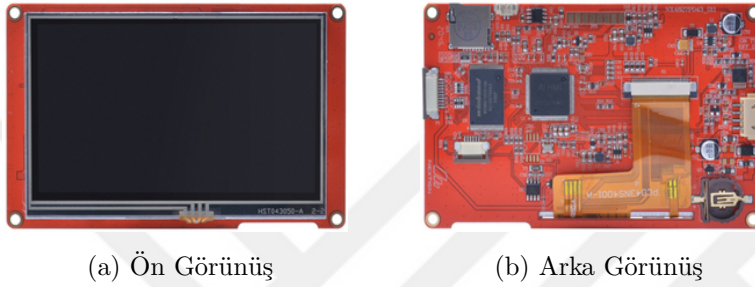
(b) DHT22

Şekil 3.10: DHT22 sıcaklık ve nem sensörü

Sıcaklık ve bağıl nemi ölçmek için ise, tek bir kılıf halinde olan AM2302 adıyla da tanınan DHT22 sensörü tercih edilmiştir.

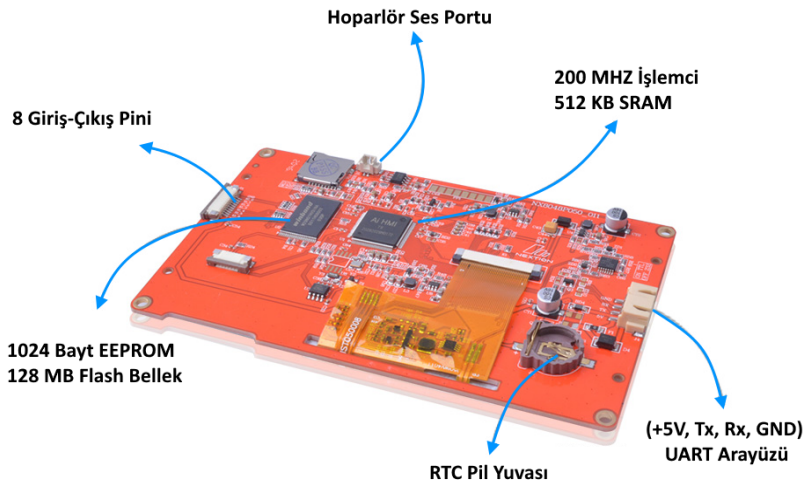
Sıcaklığı ölçmek için kullanılan termistör ve nemi ölçmek için kullanılan kapasitif nem sensörü Şekil 3.10.a'da verilmiştir. -40°C ile 80°C arasındaki sıcaklıkları $\pm 0,5^{\circ}\text{C}$ hassasiyetle ve %0 ile %100 arasındaki nem seviyelerini ± 2 RH hassasiyetle ölçerek dijital bir çıkış olarak ESP32'ye göndermektedir.

Son kullanıcı tarafından görüntüleme ve kontrol yapılabilmesi için İnsan-Makine Arayüzü (Human-Machine Interface (HMI)) olarak Nextion marka NX4827P043-011 TFT LCD kullanılmıştır.



Şekil 3.11: NX4827P043-011 TFT LCD

Şekil 3.12'de gösterilen NX4827P043-011 modeli, 4.3inch ekran boyutuna ve 480x272 çözünürlüğe sahiptir. Ekran dokunmatığı ise çoklu dokunma desteğine sahip ve kapasitif olduğu için, rahat bir kullanım sağlamaktadır.



Şekil 3.12: NX4827P043-011 bileşenleri

UART arayüzünden, ESP32 ile 115200 iletişim hızında haberleşmektedir. NX4827P043-011 modeli hakkındaki ayrıntılı bilgiler, Tablo 3.3'da yer almaktadır.

Tablo 3.3: NX4827P043-011 Özellikleri

Özellikler	Açıklamalar
Ekran boyutu	4.3 inç
Çözünürlük	480x272 piksel
Ekran tipi	TFT LCD
Renk derinliği	16 bit
Dokunmatik	Kapasitif
İletişim arayüzü	UART
Gerçek zamanlı saat	Pil destekli dahili RTC
Parlaklık kontrolü	Ayarlanabilir

3.3 Yazılım Mimarisi

Bu tez çalışması kapsamında, gömülü sistem (Embedded System) alanında, ön yüz geliştirme (Frontend) alanında ve arka yüz geliştirme (Backend) alanında olmak üzere toplam 3 katagoride yazılım geliştirilmiştir.

Gömülü sistem alanındaki yazılımlar ESP32-Wroom-32 modülü ve Nextion NX4827P043-011C modeli ile gerçekleştirilmiştir. Buzdolapları üzerinde monte edilen bu iki elektronik kart içerisinde çalışan gömülü sistem yazılımları, hem buzdolaplarının soğutma algoritmalarını yönetmekte, hem de kablosuz uzak bağlantı için kullanılmaktadır.

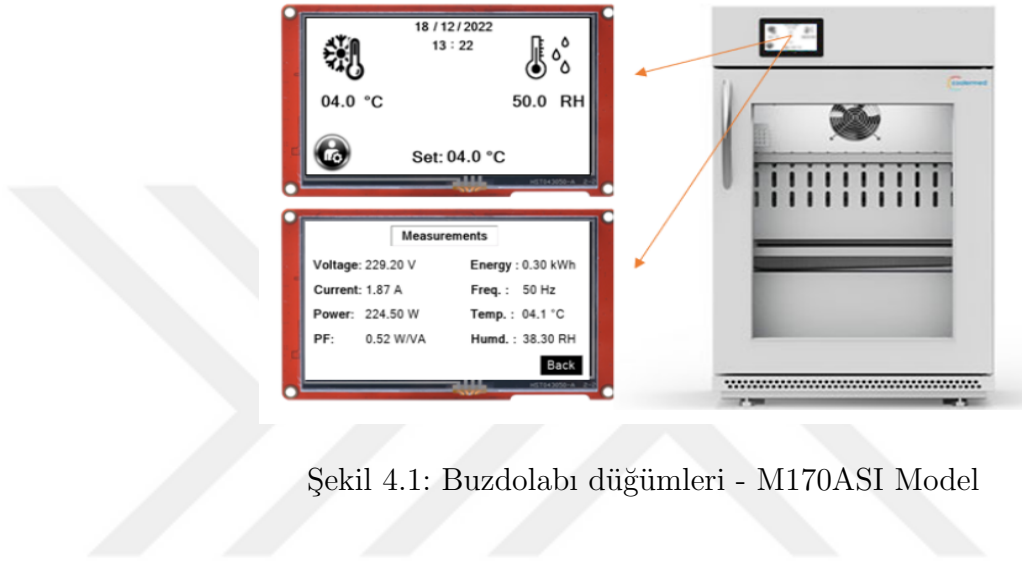
Ön yüz geliştirme alanındaki yazılımlar masaüstü, web sitesi ve mobil arayüz için geliştirilmiştir. Kullanıcı arayüzü oluşturmak için kullanılan bu uygulamalar gerekli verilerin listelemesi, grafiğe dökülmesi ve uzaktan kontrol gibi amaçlarla kullanılmaktadır.

Son olarak, arka yüz geliştirme alanındaki yazılımlar ise, RestFul API ve windows servis uygulamaları için gerçekleştirilmiştir. Bahsedilen bu iki uygulama bir server içerisine aktarılarak kesintisiz çalışması sağlanmıştır. Sonuç olarak, 3 ayrı kategoride, toplam 7 farklı yazılım kodlaması gerçekleştirilmiştir. Geliştirilen tüm ögeler, ekler bölümde ayrıntıları ile incelenmiş ve kodlama parçacıklarına yer verilmiştir.



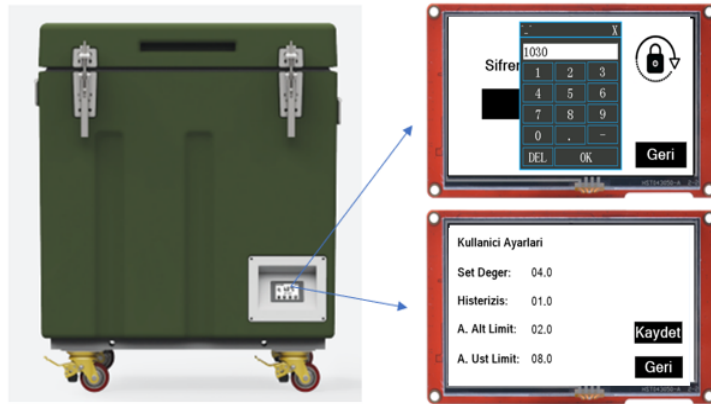
4 SİSTEMİN ÇALIŞMASI

Bu tez çalışması kapsamında geliştirilen IoT sistemi, Öztaş Global Soğutma tarafından üretilen dolaplar üzerinde test edilmiştir. Kompresör, kondanser, evaporatör ve fanlardan oluşan soğutma sistemi, bu tez çalışması kapsamında sabit kabul edilmiş ve soğutma sistemi aktif olacak şekilde kullanılmıştır.



Şekil 4.1: Buzdolabı düğümleri - M170ASI Model

Buzdolaplarının soğutma anahtarlamasının yapılabilmesi için güç ve kontrol kartının montajı gerçekleştirildikten sonra, LCD panelin montajı yapılmıştır (Şekil 4.1). İlgili sensör ve ölçüm modülleri de yerleştirildikten sonra, dolap kablosuz ağa bağlanmaya hazır hale gelmiştir (Şekil 4.2).



Şekil 4.2: Buzdolabı düğümleri - AC50S Model

Birkaç buzdolabının yan yana test edildiği bu çalışmada, tüm dolapların internete bağlanması ile MQTT ağına bilgi gönderimi başlamıştır. Uzaktan kontrol sayesinde ise, cihazlar servis moduna geçmekte ve tüm soğutma komponentleri tek tek kontrol edilebilmektedir.

Sistem çalışmasının daha net anlaşılabilmesi için 3. bölümde bahsedilen gömülü sistem, ön yüz ve arka yüz alanındaki kodlamalar bu bölümde 3 ayrı ana başlık altında incelenecektir.

4.1 Gömülü Sistem Kodlama

Gömülü sistem yazılımlarında, ESP32 modülü ve Nextion LCD arayüzü için kodlama yapılmıştır. ESP32 modülü için Arduino Geliştirme Ortamı tercih edilirken, Nextion arayüzü tasarımı için ise Nexion Editör kullanılmıştır.

4.1.1 ESP32 Modülü Yazılımı

Donanım bölümünde ayrıntıları ile incelenen ESP32-Wroom-32'nin çift çekirdek yapısı, tek bir modül üzerinden Wi-Fi & BLE desteği sağlaması ve Gerçek Zamanlı İşletim Sistemi (Real Time Operating System (RTOS)) desteklemesi sayesinde, çok daha modüler ve esnek bir yazılım tasarlanmasını sağlamıştır.

RTOS, gerçek zamanlı uygulamalarda çoklu görev dağılımlarına izin vermektedir. Bir bütün halinde olan algoritma, küçük alt görevlere (Task) bölünür ve her görev birbirinden bağımsız çalışır. Yani, her görevin kendisine ait ve sonsuz bir döngüsü vardır. Bunun sonucu olarak çok daha hafif, taşınabilir ve ölçeklenebilir bir kodlama yapısı oluşmaktadır.

ESP32 kodlanmasında, günümüzde en popüler gerçek zamanlı işletim sistemi olan FreeRTOS tercih edilmiştir. Wi-Fi bağlantısı, BLE bağlantısı, UART seri haberleşmesi, ölçümler ve röle anahtarlama biribirinden bağımsız görevler içerisinde yürütülürler. Böylece herhangi bir görev bloğunda bir gecikme yaşanırsa, diğer görev blokları bunlardan etkilenmemektedir.

Wi-Fi görev bloğunda, önce Wi-Fi bağlantısı sağlanır ve ardından MQTT aracısına bağlanıp, istenilen etiketlere abone olunur. Wi-Fi bağlantısının kopması durumuna karşı, algılama ve tekrar bağlanma algoritmaları bu görev bloğunda yer almaktadır. Ölçülen değerler tek bir dizi içerisine yerleştirildikten sonra şifrelenip, MQTT aracısına gönderilmektedir.

ESP32 modülü, yerleşik BLE desteği içerir ve görev bloğunda, herhangi bir akıllı ile cihaz ile eşleşebilmek için gerekli algoritmalar yürütülür. Burada, modülün sağlayacağı hizmetleri ve özellikleri tanıtmak üzere GATT protokolü kullanılmaktadır. Verilerin nasıl tutulduğu, düzenlendiği ve gönderileceği bu protokol üzerinden belirlenmektedir. BLE üzerinden etkileşimde, mevcut hizmetleri ve özellikleri keşfetmek için Evrensel Benzersiz Tanımlayıcı (Universally Unique Identifier (UUID)) kullanılmaktadır. UUID, 128 bitliktir ve kısa çizgilerle ayrılan beş gruptan oluşmaktadır.

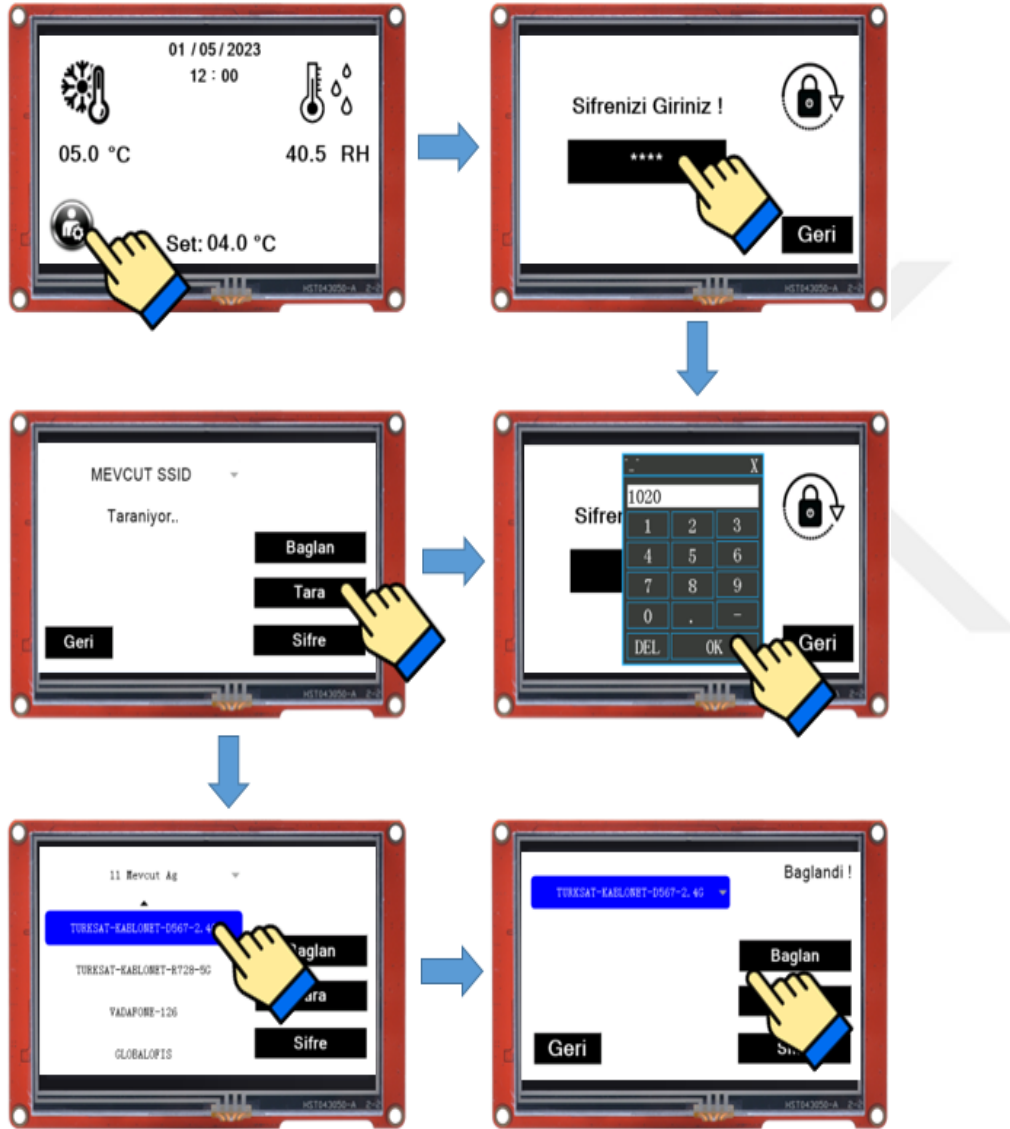
Ölçümler görev bloğunda, D15 dijital pinine bağlı DHT22 Sıcaklık & Nem ölçümleri ve UART çevre birimi üzerinden 9600 baud değerinde veri hızı ile gelen gerilim, akım, güç, enerji, frekans ve güç faktörü PZEM-004T parametre değerleri alınır.

Kullanılan bir diğer görev bloğu ise LCD ve ESP32 haberleşme görev bloğudur. Burada gerekli görülen değişkenler UART çevre birimi üzerinden LCD panele gönderilmektedir. Aynı zamanda LCD panelden ESP32 modülüne ulaşan bilgiler ESP32 içerisinde yer alan EEPROM içerisine kaydedilmektedir.

4.1.2 HMI (Human Machine Interface) Yazılımı

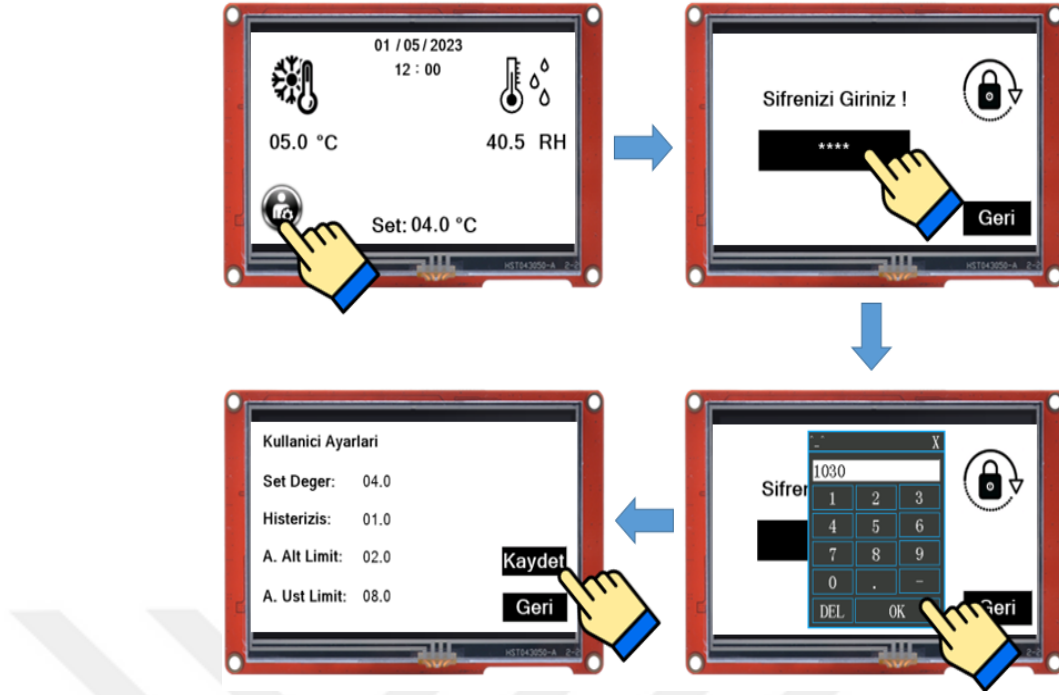
Buzdolabı kullanıcıları için bilgilendirme ve kontrol paneli olan, Nextion marka NX4827P043-011 model LCD ekran yazılımı Nextion Editör üzerinde düzenlenmiş ve kodlanmıştır. Lcd panelin, ana ekranında ilgili parametreler ve olası hata durumları gözlemlenebilmektedir. Ayrıca, şifre korumalı kullanıcı ekranları vasıtasıyla, dolap Wi-Fi bağlantısı, dolap soğutma set parametreleri, sensör kalibrasyonları ve tarih saat ayarları yapılabilir.

Şekil 4.3’de, dolabın Wi-Fi ağına bağlantısını gösteren akış bulunmaktadır. Kullanıcı menüsünden, 1020 şifresiyle ulaşılan Wi-Fi ekranından ağ taraması sağlanabilmektedir. Taramanın tamamlanmasının ardından, mevcut ağlardan birisi seçilmektedir. Seçilen ağın şifresinin girilmesi ile birlikte, istenilen ağa bağlantı sağlanmış olunmaktadır.



Şekil 4.3: Mevcut Wi-Fi ağına bağlantı sağlanması

Şekil 4.4’de, buzdolabının set değerlerinin değiştirilme adımları gösterilmektedir. 1030 şifresine sahip yetkili kullanıcı ile buzdolabının sıcaklık set değeri, soğutmanın duracağı histerezis değeri ve alarmların devreye gireceği alt ve üst sıcaklık değerleri ayarlayabilmek mümkündür.

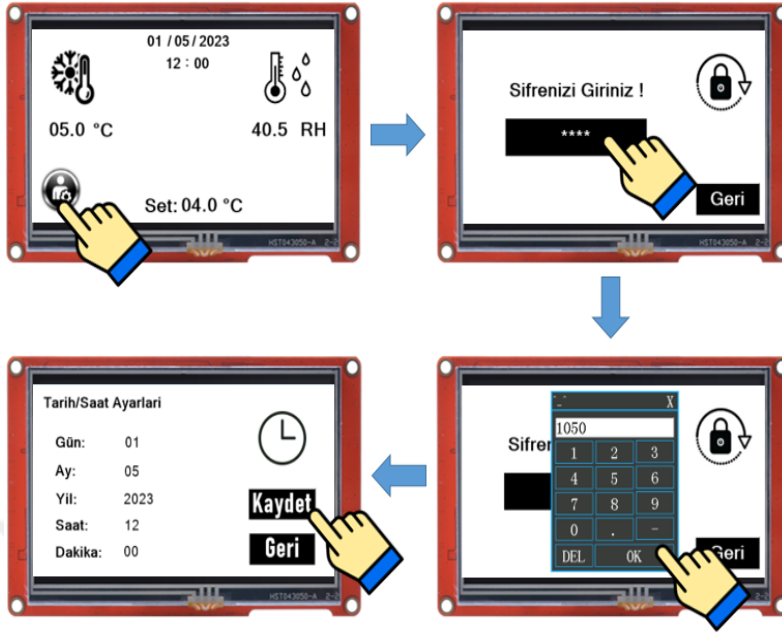


Şekil 4.4: Buzdolabı set parametrelerinin ayarlanması

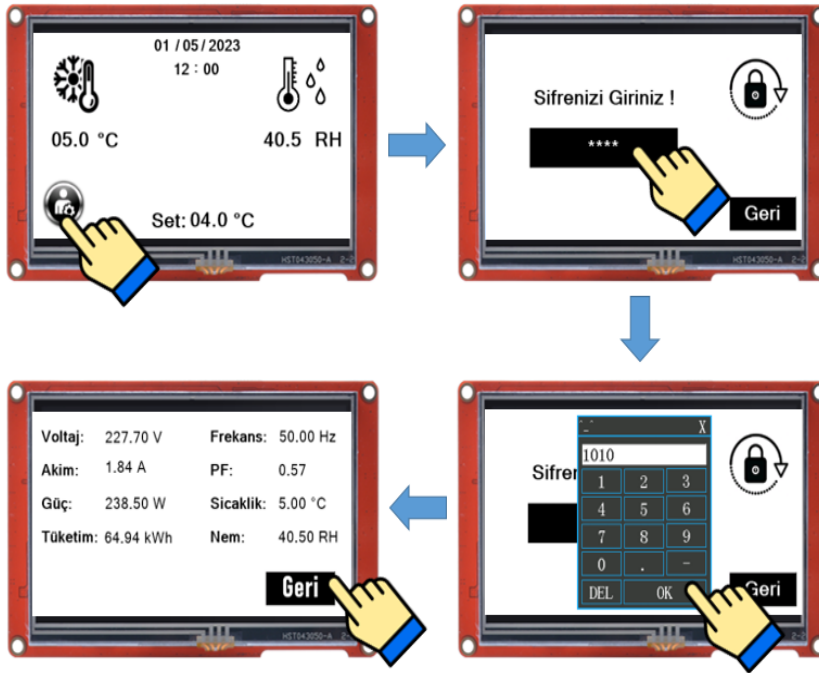
Ekler bölümünde yer alan kod parçacığında set parametrelerinin ayarlandığı sayfadaki zamanlayıcı algoritması yer almaktadır. Şekil 4.4’de gösterilen kaydet butonuna basılması ve 50 ms aralıklarla çalışan zamanlayıcı vasıtasıyla, girilen tüm set parametreleri ESP32 modülüne gönderilmektedir. Gönderilen tüm değerler ESP32 modülüne ulaştıktan sonra EEPROM içerisine kaydedilmektedir.

Tarih/Saat ayarların yapılabilirdiği akış ise, Şekil 4.5’de verilmiştir. Gerçek Zamanlı Saat (Real Time Clock (RTC)) kütüphanesinin kullanıldığı bu bölümde, LCD panel üzerinde bulunan bir pil vasıtasıyla saat ayarlandıktan sonra güç kesilse bile, Tarih/Saat değerleri arka planda çalışmaya devam etmektedir.

Şekil 4.6’da ise, daha çok servis yetkililerinin ihtiyacı olabilecek güç parametreleri menünün görüntülenme adımları gösterilmektedir. Buna göre, bir servis yetkilisi 1010 şifresi ile ilgili menüye ulaşabilmektedir. Elde edilecek bu verilere göre müdahale yapmak çok daha sağlıklı olacaktır.



Şekil 4.5: Tarih ve saat ayarı



Şekil 4.6: Güç parametrelerine erişim

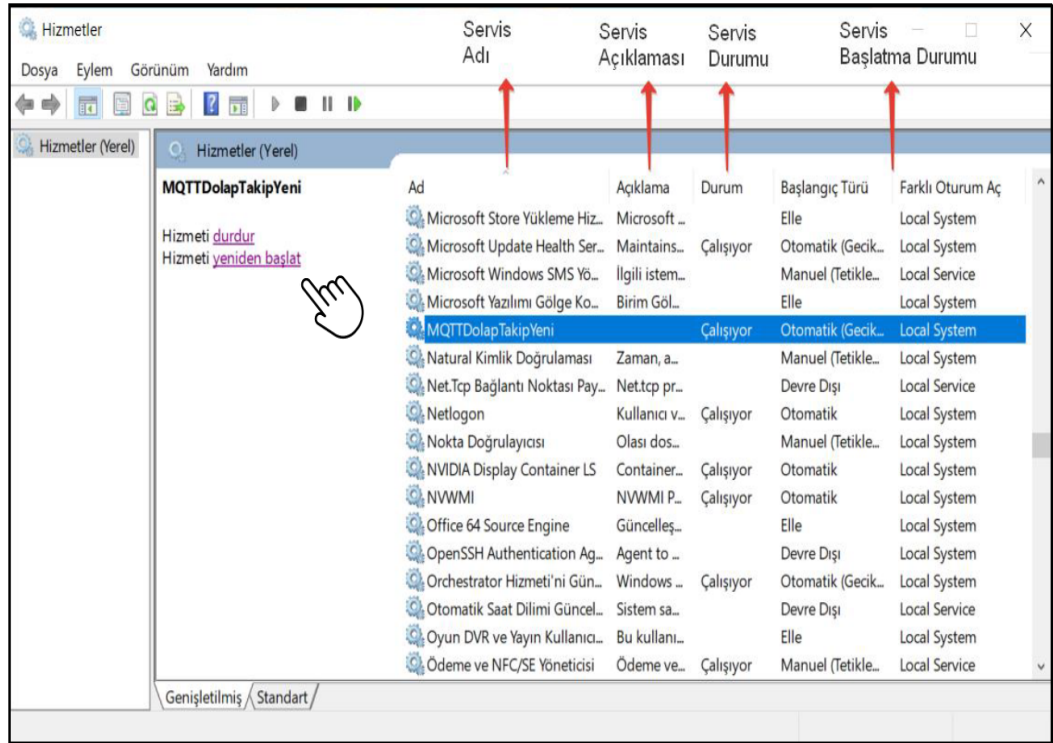
4.2 Arka Yüz Kodlama

Arka yüz Kodlama (Back-End Coding) tasarlanan bir sistemin arka planda sunucu ile ilişkili olan kısımları yürütmekle sorumlu olan yazılımlardır. Genellikle veritabanı için gerekli olan bilgileri işleme ve gerektiğinde verileri alma işlemleri yapılır.

Bu tez çalışması kapsamında arka yüz kodlamada kısmında, MQTT aracısına bağlı ve bir sunucu üzerinde çalışan Windows Servis uygulaması ve yine aynı sunucu üzerinde bulunan Uygulama Programlama Arayüzü (Application Programming Interface (API)) geliştirilmiştir.

4.2.1 Windows Servis Yazılımı

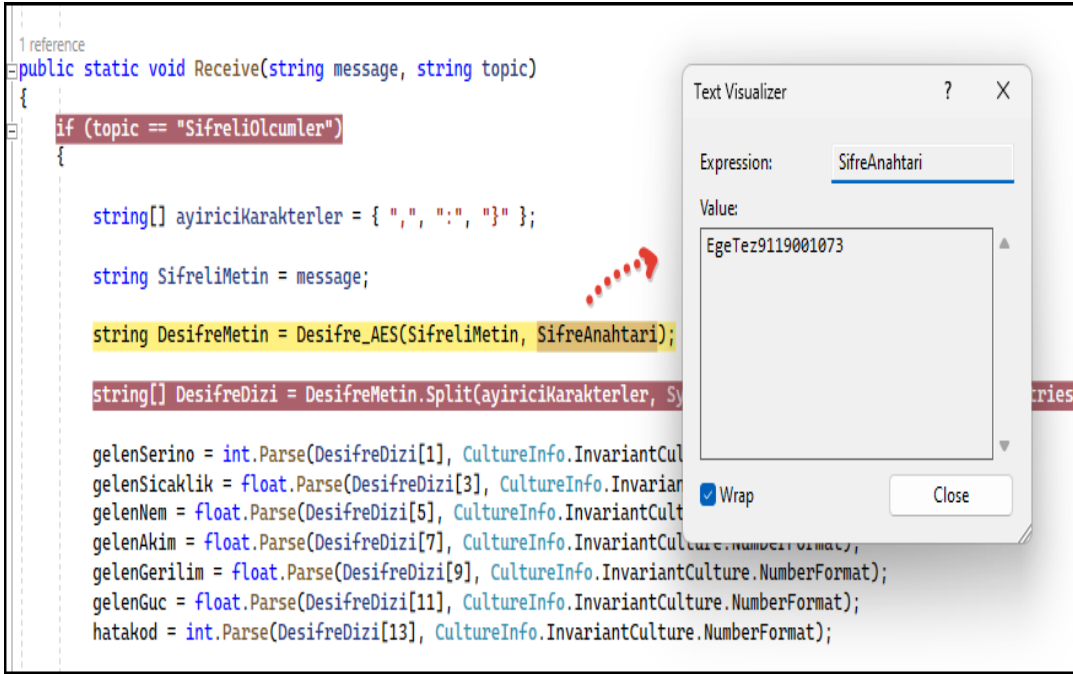
Windows Servis uygulamaları, Windows işletim sistemlerinde herhangi bir kullanıcı müdahalesi olmadan ve arka planda ilgili kod parçacıklarını çalıştırmak için kullanılmaktadır (Şekil 4.7).



Şekil 4.7: Sistemde çalışan Windows Servis uygulamaları

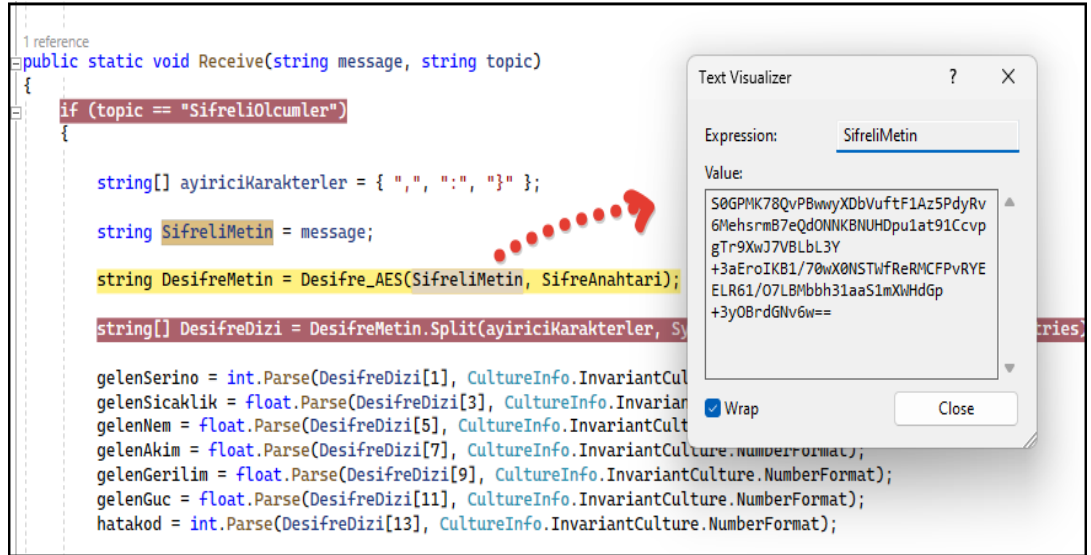
Bu çalışmadaki kullanım amacı, veritabanına direk erişimi olmayan MQTT aracısına bağlanarak, abone olunan tüm bilgilere erişebilmek ve bu değerleri veritabanına işlemektir. Bu sayede ilereye yönelik sistem gelişiminde çok daha esnek ve ölçeklenebilir bir yapı sağlanmış olmaktadır.

Windows servis uygulaması, Visual Studio Entegre Geliştirme Ortamında (Integrated Development Environment (IDE)) ve C# dili ile kodlanmıştır. Visual Studio IDE, geniş bir Hata Ayıklama (Debugging) özelliği sağlamaktadır ve bütün analiz ve geliştirmeler bu vasıta ile yapılmıştır.



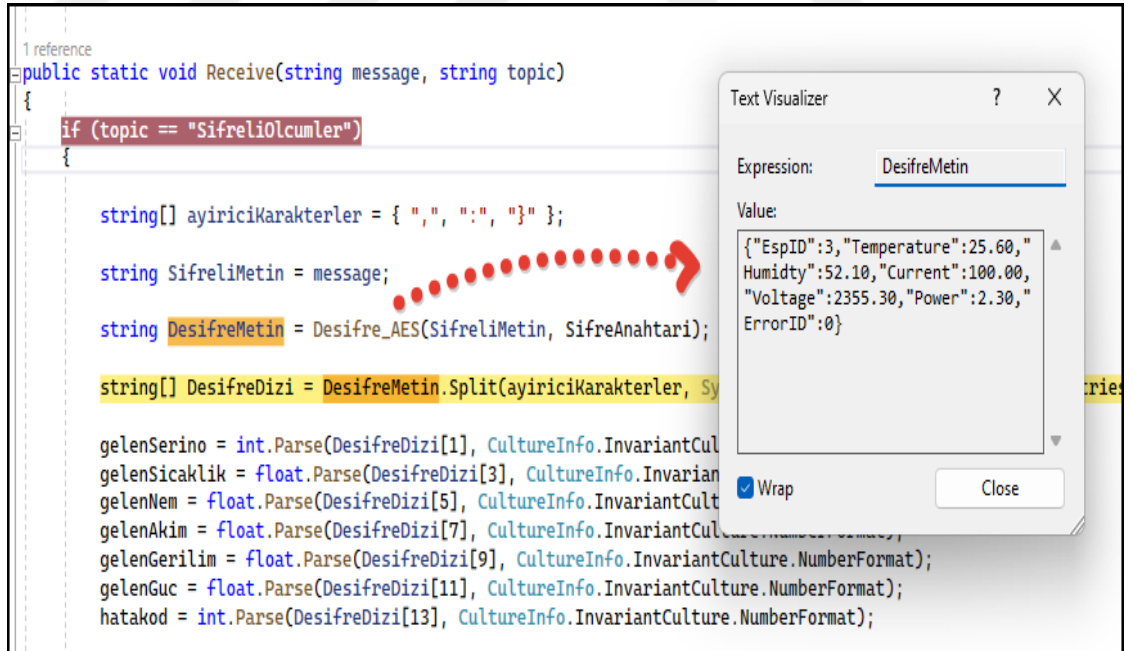
Şekil 4.8: Visual Studio Analiz - Şifreleme/Deşifreleme Anahtarı

Şekil 4.8’de Visual Studio IDE üzerinde bir analiz işlemi görüntülenmektedir. MQTT aracısından "SifreliMetinler" etiketiyle gelen şifreli metinler, aynı şekilde gösterildiği gibi 16 bayt (128 bit) olan "EgeTez9119001073" 16 bayt (128 bit) şifre anahtarı ile deşifre edilmektedir. Şekil 4.9’da ise MQTT aracısından gelen şifreli metin içeriği gösterilmektedir.

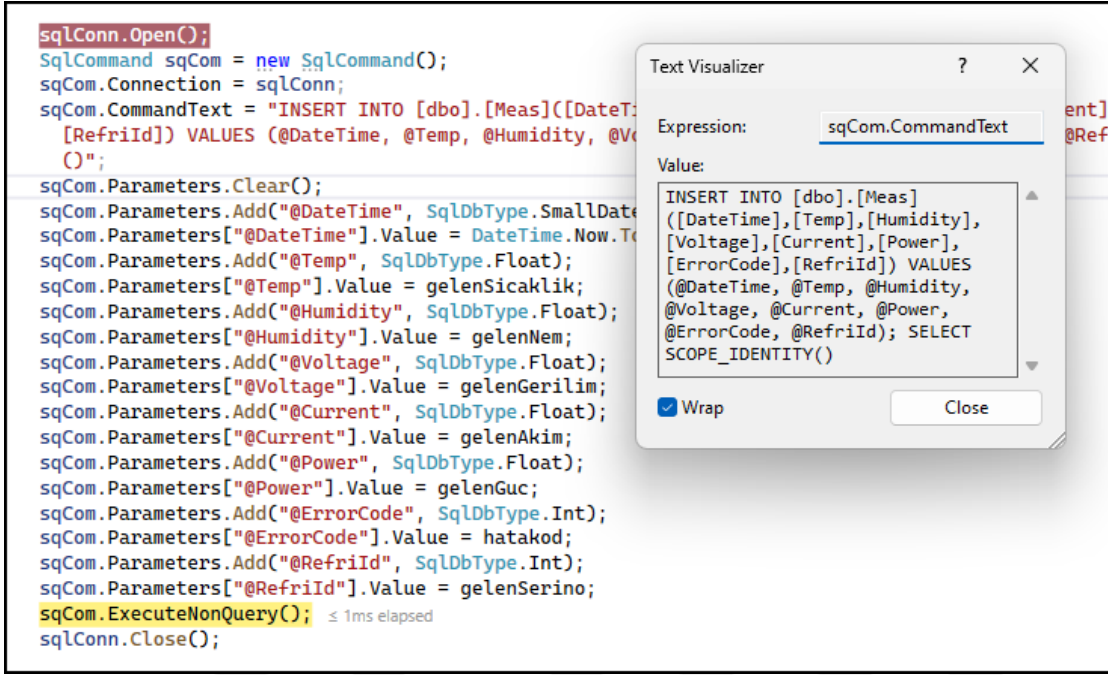


Şekil 4.9: Visual Studio Analiz - Şifreli Metin

Şekil 4.10'da deşifre edilmiş metin içeriği görülmektedir. JSON formatındaki bu verinin ayrıştırılması ile tüm veriler Şekil 4.11'de gösterildiği gibi veritabanına kaydedilmektedir.



Şekil 4.10: Visual Studio Analiz - Deşifre Metin



Şekil 4.11: Visual Studio Analiz - Veritabanına Kayıt

4.2.2 API (Application Programming Interface) Yazılımı

Uygulama Programlama Arayüzü (Application Programming Interface (API)) geliştirilen farklı yazılım uygulamalarının aralarındaki ortak iletişim için veya bir veritabanı ile uygulamalar arasındaki köprü olabilmesi için kullanılan bir dizi kurallar ve protokollerdir.

Bu çalışma kapsamında ise, geliştirilen akıllı uygulamaların, veritabanındaki verilere erişebilme ve veritabanına veri işleme amaçlı Temsili Durum Transferi (Representational State Transfer (REST)) mimarisi kullanan RESTful API kullanılmaktadır. RestFul API, önceki başlıklar altında ayrıntılı değinilen HTTP komutları ile JSON formatında bir veri alış-verişi yapılmaktadır.

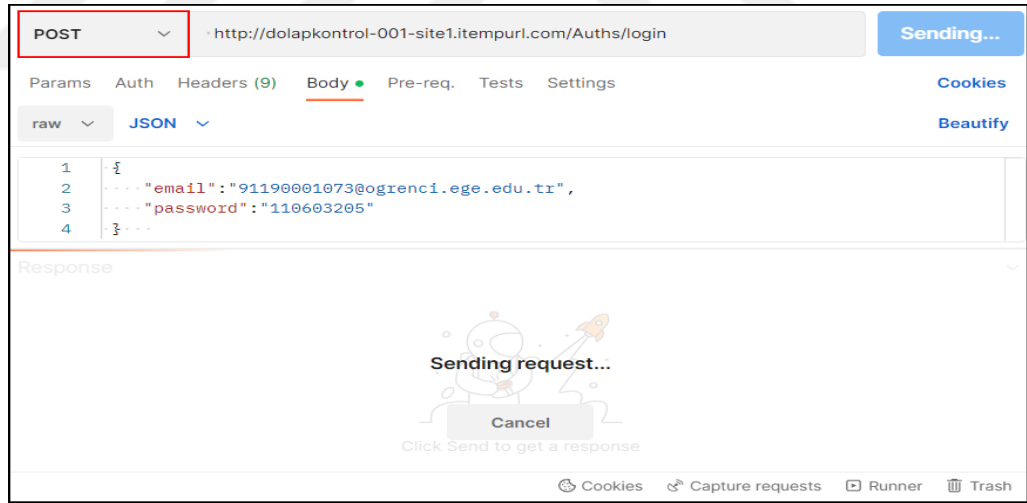
Geliştirilen RestFul API, Visual Studio IDE vasıtasıyla ve C# dili ile tasarlanmıştır. Ayrıca ASP .NET Framework alt yapısı kullanılarak kodlanmıştır. Tasarım deseni olarak Model, Görünüm ve Denetleyici (Model-View-Controller (MVC)) kısmından oluşan bir yapı tercih edilmiştir.

API ve web sitesi tasarımlarında sıkça kullanılan bu desen, temel anlamda tasarlanan uygulamayı sunumun yapılacağı, isteklerin yönetileceği ve depo-

lama işlerinin yürütüleceği 3 ayrı bileşene ayırmaktadır.

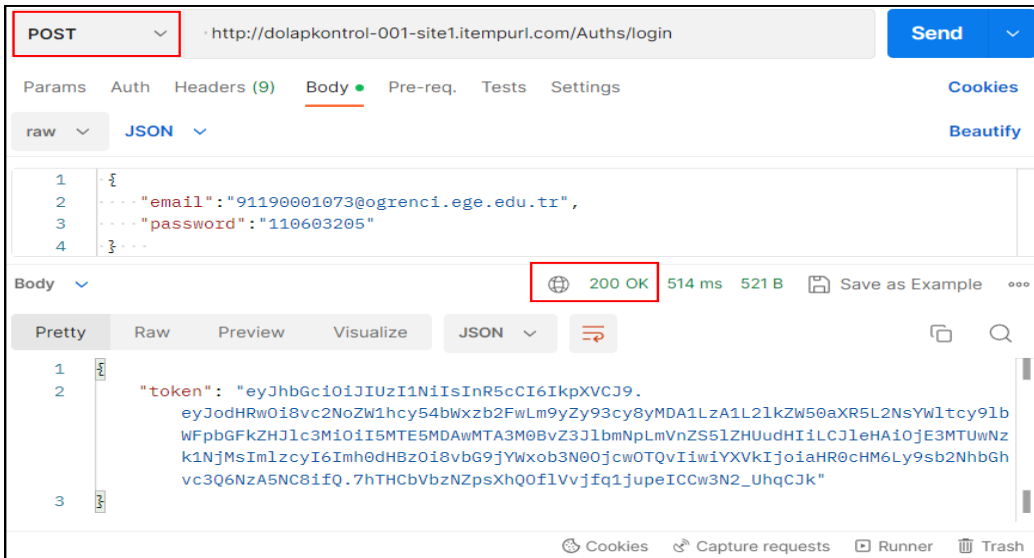
- **Model:** Veritabanına kayıt için kullanılacak değişklerle bir model kullanılmaktadır. Bu sayede kurallar ve prosedürler tanımlanmış olmaktadır.
- **Görünüm:** Sunum katmanı olarak da bilinen bu katmanda, verilerin istemciye anlamlı gelmesi sağlanmaktadır. Veriler modelden alınır ve istemciye görüntüleme yapabilmesi için biçimlendirme yapmaktadır.
- **Denetleyici:** Model ve Görünüm arasında bir köprü görevinde olan bu bileşen, istemciden gelen veya istemciye gönderilen verileri modele göre işlem yapılmasını sağlamaktadır.

Kullanıcı modeline bağlı olarak kullanıcı girişi ve buna bağlı olarak bir önceki bölümlerde ayrıntıları ile incelenen JSON formatında JWT yetkilendirme kodu alınmaktadır.



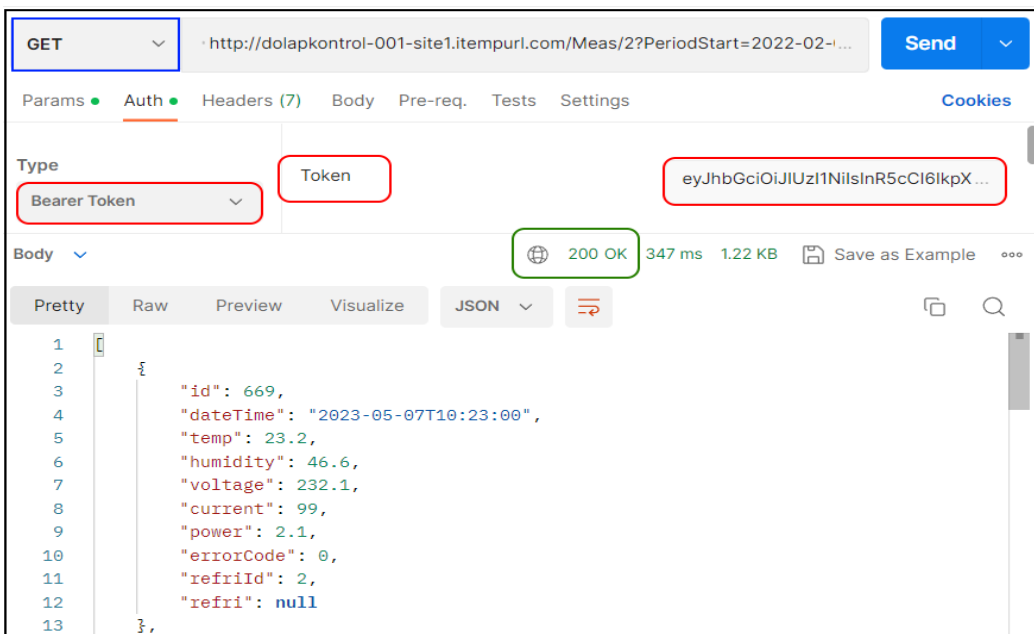
Şekil 4.12: Postman Platformu - Kullanıcı giriş isteği

Ekler bölümünde yer alan kullanıcı girişi kod parçasığının testi ise, geliştirilen API'nin bir sunucu üzerinde yayınlanmasının ardından, Postman platformu aracılığı ile sağlanmıştır. Şekil 4.12'de "http://dolapkontrol-001-site1.itempurl.com/Auths/login" adresine, sisteme giriş için bir istekte bulunmaktadır.



Şekil 4.13: Postman Platformu - JWT alınması

Gönderilen kullanıcı isteğinin sonucunda, "HTTP 200" başarılı durum kodu ile birlikte JWT kodu elde edilmiştir (Şekil 4.13). Şekil 4.14'de "http://dolapkontrol-001-site1.itempurl.com/Meas/2?PeriodStart=2022-02-02&PeriodEnd=2023-06-02" sunucu adresinden, 2 numaralı Id değerine sahip bir dolabın, "2022-02-02" tarihinden "2023-06-02" tarihine kadar olan ölçümleri "GET" HTTP isteği ile alınmaktadır.



Şekil 4.14: Postman Platformu - Ölçümlerin alınması

4.3 Ön Yüz Kodlama

Geliştirilen uygulamaların istemci tarafında, kullanıcı arayüzünü (User Interface (UI)) ve kullanıcı deneyimini (User Experience (UX)) tanımlamak için kullanılan Ön Yüz Kodlama (Frontend Coding) sayesinde kullanıcının konfor alanını genişleten bir çok uygulama geliştirilebilmektedir.

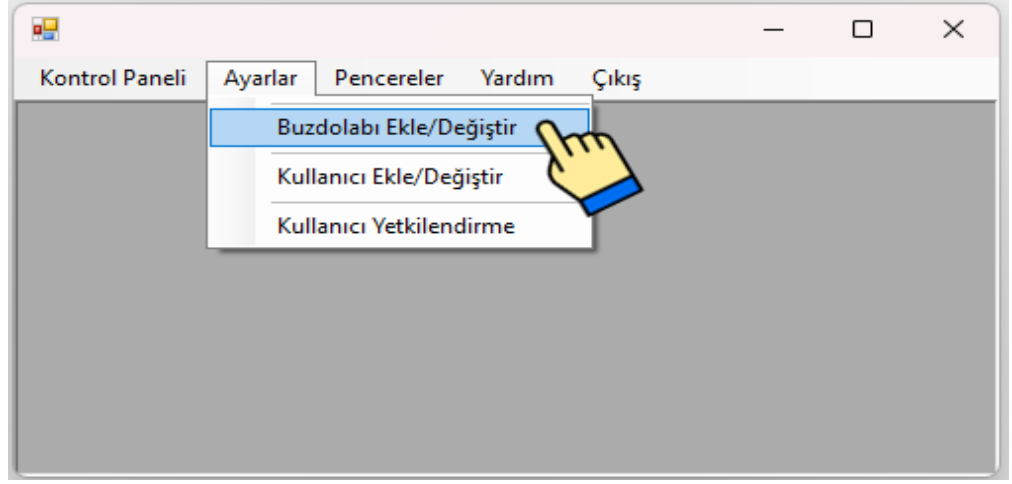
Ön yüz geliştirme uygulamaları genellikle veri tabanından ilgili verilere ulaşabilmek için arka yüz uygulamaları ile haberleşmektedir. Bu haberleşme sırasındaki önemli nokta kullanıcı bilgilerinin güvenliği sağlamaktır. Bu yüzden tüm bağlantılar için sadece yetkili kullanıcılara izin verilmektedir.

Bu tez çalışması kapsamındaki ön yüz kodlama kısmında, buzdolabı üreticilerinin kontrolüne ve takibine sunulan bir masaüstü uygulama, müşteri tarafında görüntüleme yapılabilmesi için bir web sitesi ve Wi-Fi bağlantısı olmayan alanlar için bluetooth uygulaması geliştirilmiştir. Böylece gelişen IoT sistemlerindeki tüm teknolojilere yer vermeye ve katkıda bulunabilmeye çalışılmıştır.

4.3.1 Masaüstü Uygulama Yazılımı

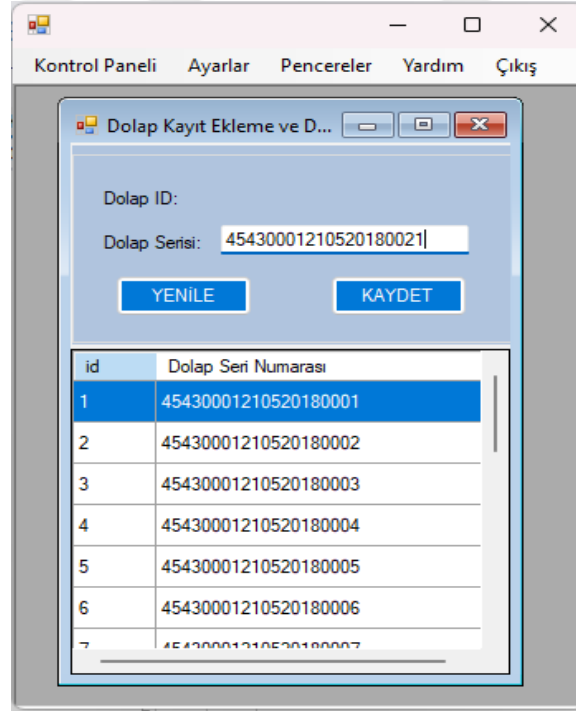
Buzdolabı üreticileri için geliştirilmiş masaüstü uygulaması, kapsamlı bir kontrol olanağı sunulmaktadır. Bu bağlamda geliştirilen özelliklerinden birincisi anlık takip ve kontroldür. Sistem üzerinden seçilen bir buzdolabına ait tüm ölçümler ve hata bildirimleri anlık olarak gözlemlenebilmektedir. Diğer bir özellik ise, seçili olan buzdolabına ait geçmiş kayıtları, hem uygulama üzerinden hem de uygulamanın Excel formatındaki aktarımı sayesinde detaylı inceleme olanağı bulunmaktadır. Ayrıca uygulama üzerinden, sisteme kullanıcı ve buzdolabı ekleme veya yeniden düzenleme özellikleri yapılabilmektedir.

Masaüstü uygulamasına giriş yapılabilmesi için veritabanına kayıtlı kullanıcı adı ve şifresi gereklidir (Şekil 4.15). Kullanıcı yetkilendirmesi sağlanmasının ardından sisteme giriş gerçekleşmektedir.

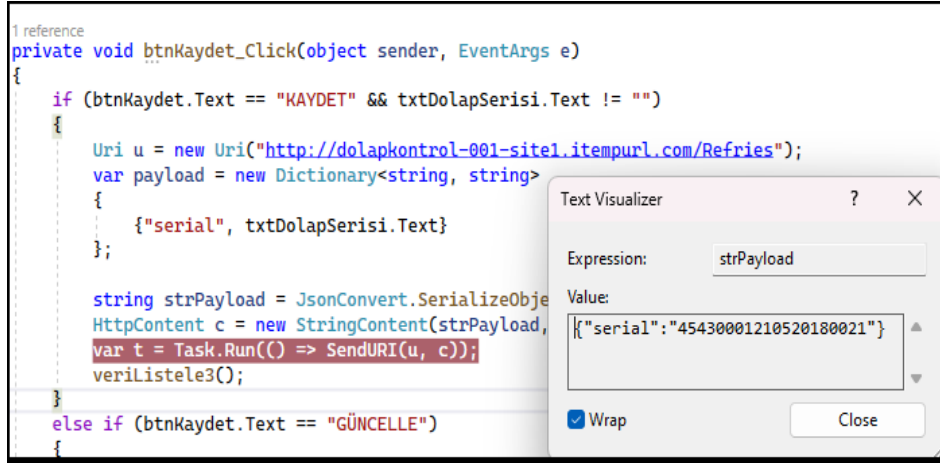


Şekil 4.17: Menüler - Buzdolabı ekleme sekmesi

Buzdolabı üreticileri, üretilen her cihazın takibini kolaylaştırmak için benzersiz bir seri numarası tanımlamaktadır. Bu özellikten yararlanılarak, "45430001210520180021" seri numaralı dolabın, sisteme kaydı Şekil 4.18'da gösterilmektedir. "http://dolapkontrol-001-site1.itemurl.com/Refries" adresine bir istek yapılarak buzdolabı veritabanına kayıt edilmektedir (Şekil 4.19).

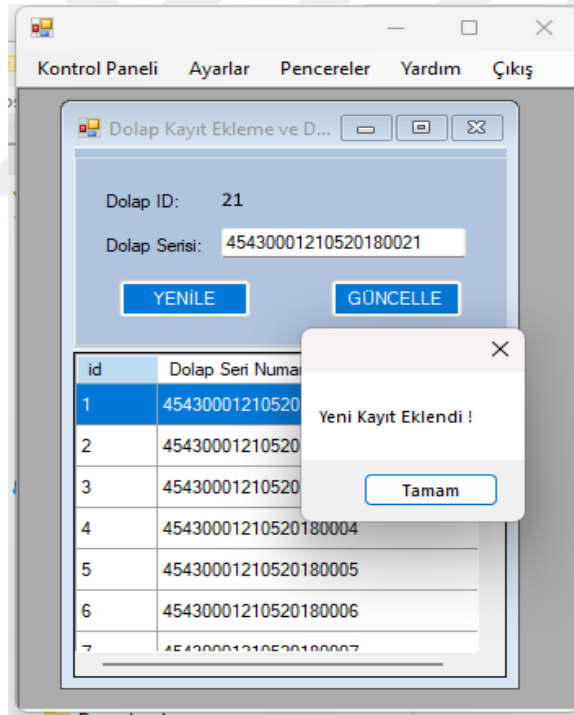


Şekil 4.18: Sisteme yeni bir buzdolabı tanımlanması



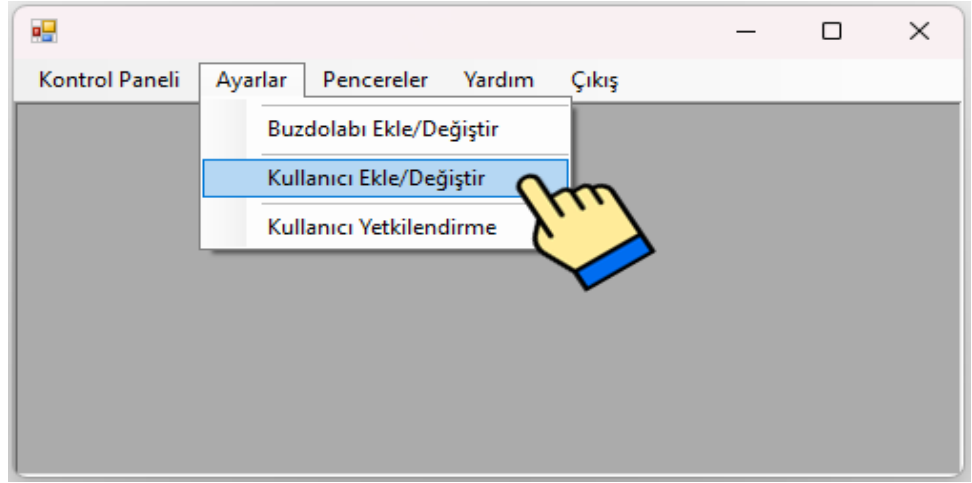
Şekil 4.19: Buzdolabı kayıt kodlama ekranı

Şekil 4.20’de buzdolabı kaydının başarılı bir şekilde tamamlandığı gösterilmektedir. Kaydedilen tüm veriler, daha sonra güncelleme butonu vasıtasıyla yeniden düzenlebilmektedir.



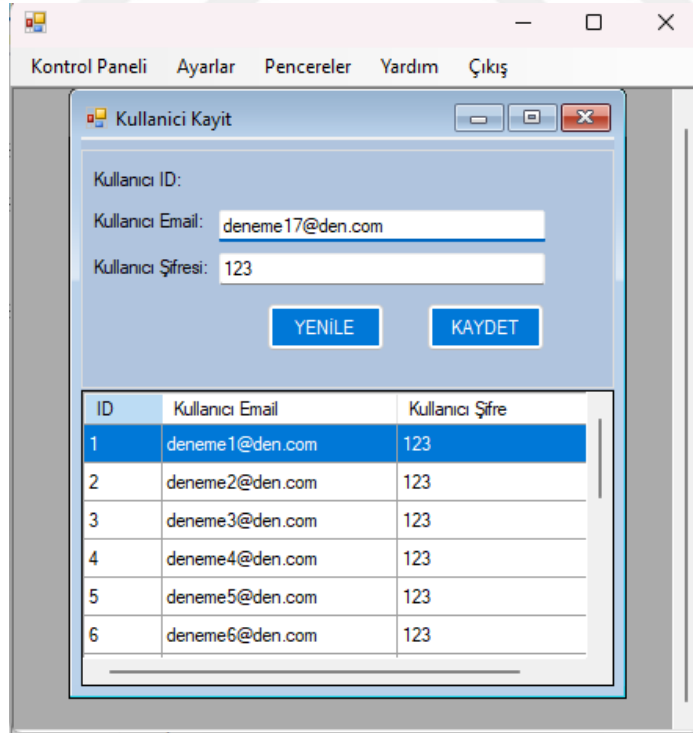
Şekil 4.20: Başarılı bir buzdolabı kayıt işlemi

Şekil 4.21’de görüldüğü gibi, istenildiği takdirde ana menü sekmesinden yeni kullanıcı kaydı da yapılabilmektedir. Bu tanımlama, üretici tarafında yapıldıktan sonra son kullanıcıya bildirilmektedir.

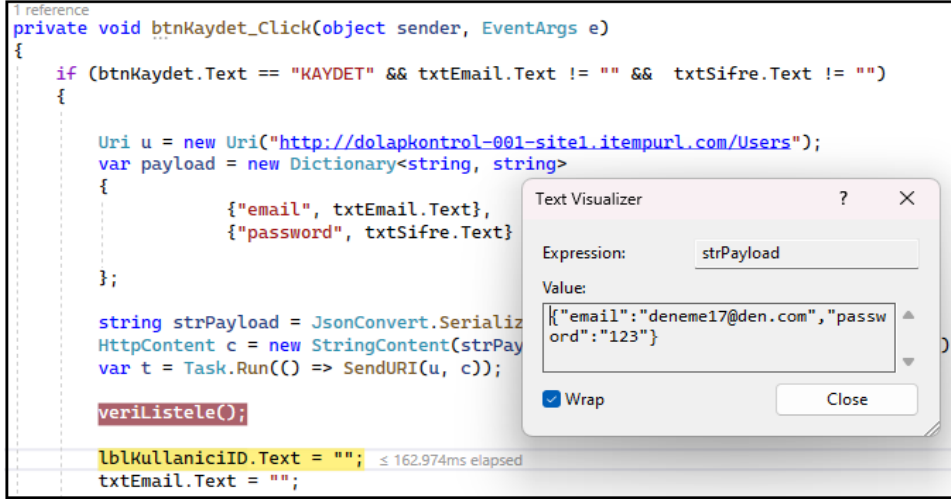


Şekil 4.21: Menüler - Kullanıcı ekleme sekmesi

Şekil 4.22’de görüldüğü gibi, yeni bir kullanıcı adı ve şifresi tanımlanmaktadır. ”http://dolapkontrol-001-site1.itempurl.com/Users” adresine JWT token kodu ile yapılan istek Şekil 4.23’de gösterilmektedir. Kullanıcı girişindeki verilen bilgilerin, analiz ekranında ve JSON mesaj formatında gönderimi incelenebilmektedir.

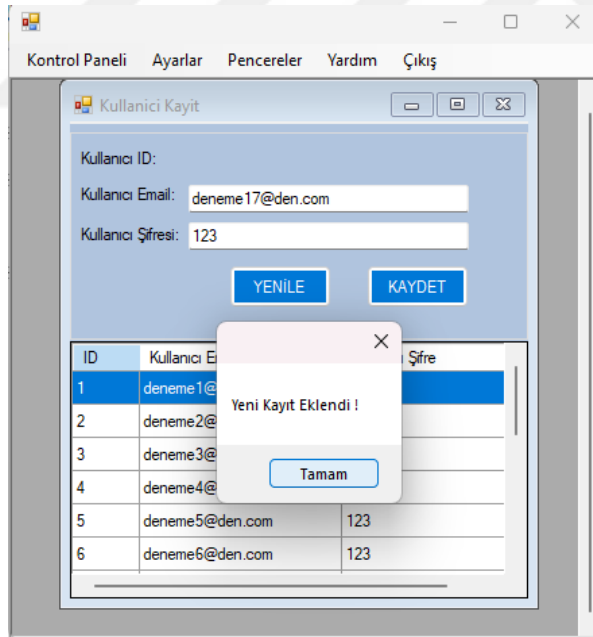


Şekil 4.22: Kullanıcı ekleme



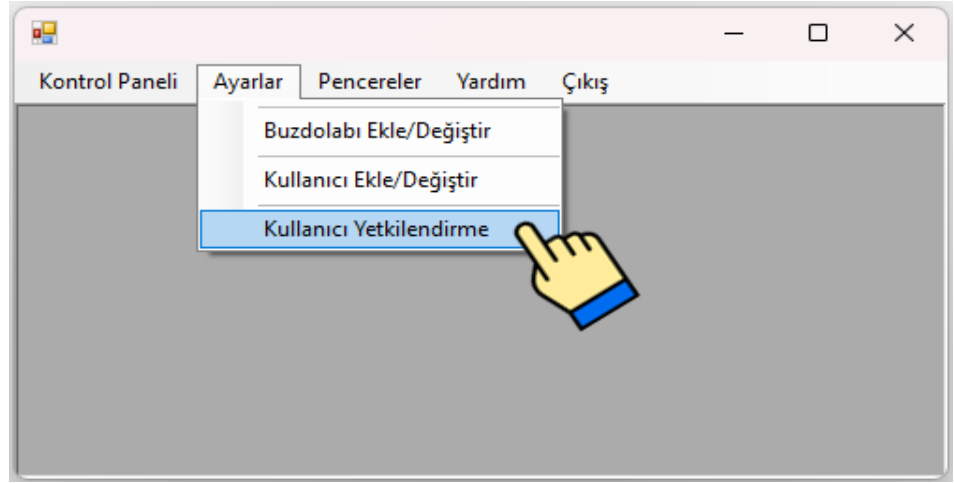
Şekil 4.23: Kullanıcı kayıt kodlama ekranı

Yeni kullanıcı eklemenin başarılı olmasıyla birlikte, ekranda bilgilendirme kutusu belirmektedir (Şekil 4.24). Bu aşamadan sonra, istenildiği takdirde güncelleme butonu vasıtasıyla bilgilerin yeniden düzenlenmesi de mümkündür.



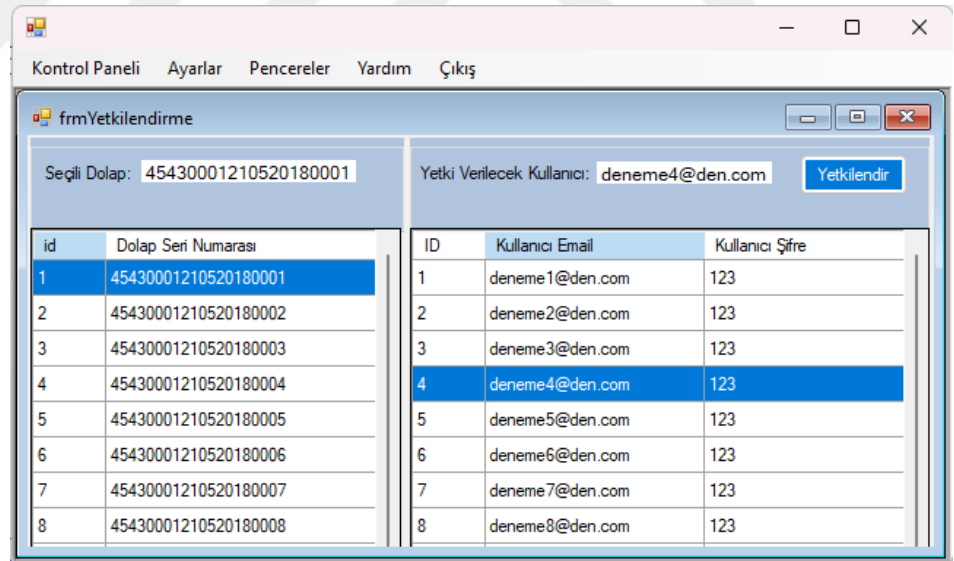
Şekil 4.24: Başarılı bir kullanıcı ekleme işlemi

Şekil 4.25’de ana menü üzerinden kullanıcıların yetkilendirme sekmesi görülmektedir. Bu sekmedeki açılan sayfada sistemde kayıtlı buzdolapları ve kullanıcılar listelenmektedir.



Şekil 4.25: Menüler - Yetki atama sekmesi

İstenilen kullanıcılara, belirtilen buzdolapların yetkilendirmesi ile web sitesi uygulamasında kullanıcılar sadece kendilerine ait dolapları görüntüleyebilmektedir (Şekil 26). Böylece sadece yetki sınırları içerisindeki dolapların verilerine erişim sağlanabilecektir.



Şekil 4.26: Kullanıcı yetkilendirme menüsü

Şekil 4.27 ise hata ayıklama ekranında kodlamanın işleyişi gösterilmektedir. ”http://dolapkontrol-001-site1.itempurl.com/Auths” adresine yapılan istek ile yetkilendirme sağlanmaktadır.

```

1 reference
static async Task SendURI(Uri u, HttpContent c)
{
    var response = string.Empty;
    using (var client = new HttpClient())
    {
        HttpRequestMessage request = new HttpRequestMessage
        {
            Method = HttpMethod.Post,
            RequestUri = u,
            Content = c
        };

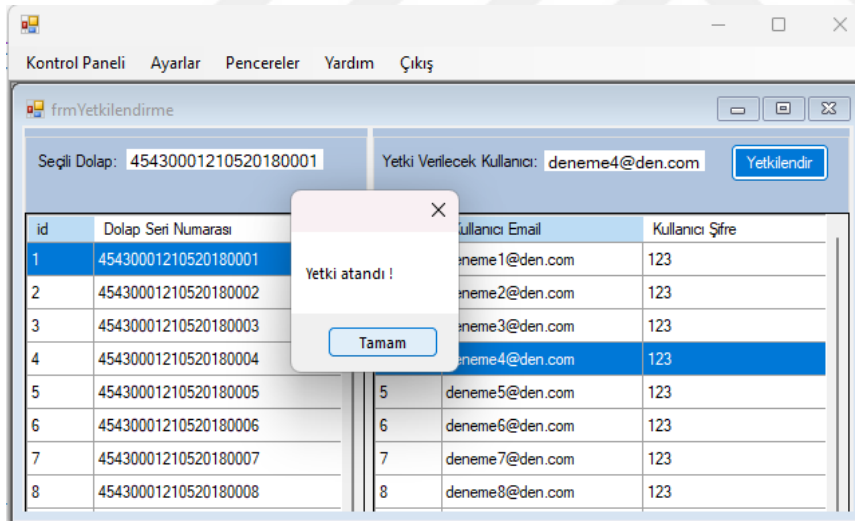
        client.DefaultRequestHeaders.Authorization =
            new AuthenticationHeaderValue("Bearer", frmGiris.token2);

        HttpResponseMessage result = await client.SendAsync(request);
        if (result.IsSuccessStatusCode)
        {
            response = result.StatusCode.ToString();
            MessageBox.Show("Yetki atandı !");
        }
    }
}

```

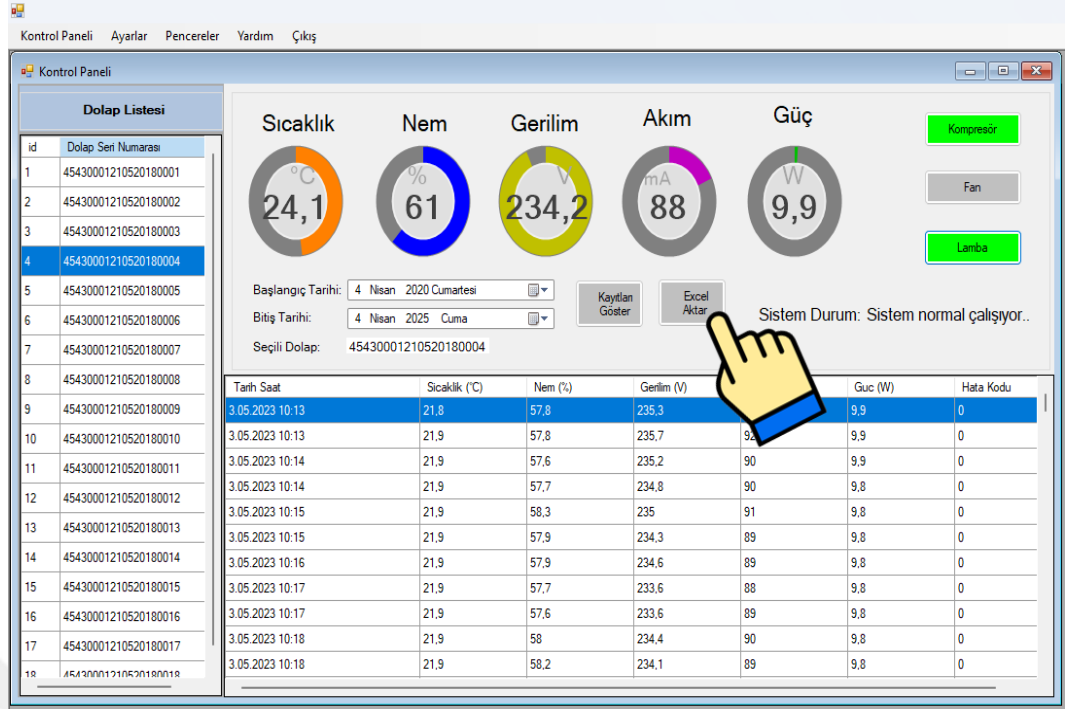
Şekil 4.27: Kullanıcı yetkilendirme kodlama ekranı

Yetki atama işleminin başarılı sonuçlandığı Şekil 4.28'deki bilgilendirme kutucuğunda görülebilmektedir.



Şekil 4.28: Başarılı bir yetkilendirme işlemi

Takip ve kontrol paneli üzerinden MQTT aracısına direk bağlantı sayesinde anlık olarak şifreli gelen veriler, deşifre edilerek kullanıcı gözlemine sunulmaktadır.



Şekil 4.29: Takip ve kontrol paneli

Şekil 4.29’da görüldüğü gibi, ”45430001210520180004” seri numaralı buzdolabı seçilmiş ve sadece bu dolabın anlık verileri görüntülenmektedir. Gelen verilerin animasyon tarzındaki grafiklerle kullanıcı dostu bir tasarım amaçlanmıştır. Bunun yanında, görüldüğü gibi anlık hata bildirimi de bu arayüz üzerinden takip edilebilmektedir.

Herhangi bir hata durumunda veya rutin kontrollerde ise, seçili olan dolabın soğutma komponentleri ilgili butonlar vasıtası ile uzaktan ve şifreli olarak kontrol edilebilmektedir.

Buzdolaplarının geçmişe yönelik kayıtları incelenmek istendiğinde ise, ilgili buton vasıtasıyla arayüz ekranına getirilebilmektedir. ”Excel Aktar” seçeneği sayesinde mevcut dolaba ait tüm veriler, Excel tablosuna aktarılıp, bilgisayar içerisindeki istenilen diske aktarılabilir (Şekil 4.30). Bu özellik sayesinde hem veri inceleme kolaylığı sağlanmış hem de ayrıntılı ve matematiksel grafik incelemeleri için alt yapı oluşturulmuştur.

	A	B	C	D	E	F	G
1	Tarih Saat	Sicaklik (°C)	Nem (%)	Gerilim (V)	Akim (mA)	Guc (W)	Hata Kodu
2	3.05.2023 10:13	21,8	57,8	235,3	91	9,9	0
3	3.05.2023 10:13	21,9	57,8	235,7	92	9,9	0
4	3.05.2023 10:14	21,9	57,6	235,2	90	9,9	0
5	3.05.2023 10:14	21,9	57,7	234,8	90	9,8	0
6	3.05.2023 10:15	21,9	58,3	235	91	9,8	0
7	3.05.2023 10:15	21,9	57,9	234,3	89	9,8	0
8	3.05.2023 10:16	21,9	57,9	234,6	89	9,8	0
9	3.05.2023 10:17	21,9	57,7	233,6	88	9,8	0
10	3.05.2023 10:17	21,9	57,6	233,6	89	9,8	0
11	3.05.2023 10:18	21,9	58	234,4	90	9,8	0
12	3.05.2023 10:18	21,9	58,2	234,1	89	9,8	0

Şekil 4.30: Ölçümlerin excel tablosuna aktarımı

4.3.2 Web sitesi Uygulama Yazılımı

Buzdolabı üreticileri ve son kullanıcılar için tasarlanan web sitesi, internet erişimi olan her ortamdan, bir tarayıcı vasıtasıyla tüm verilere erişim ve analiz imkanı sunmaktadır.

AngularAuth

localhost:4200

Giriş Sayfası

91190001073@ogrenci.ege.edu.tr

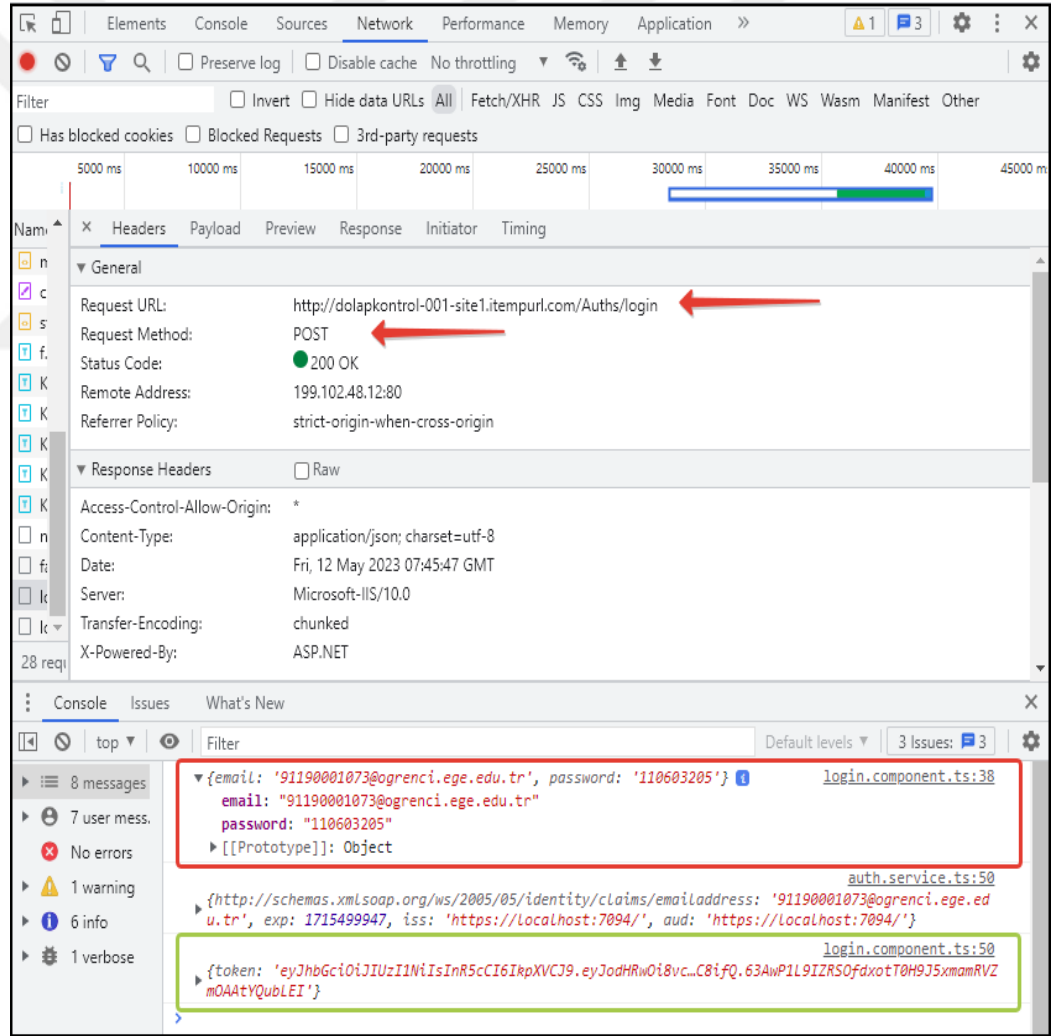
110603205

Giriş Yap

Şekil 4.31: Web Sitesi kullanıcı giriş ekranı

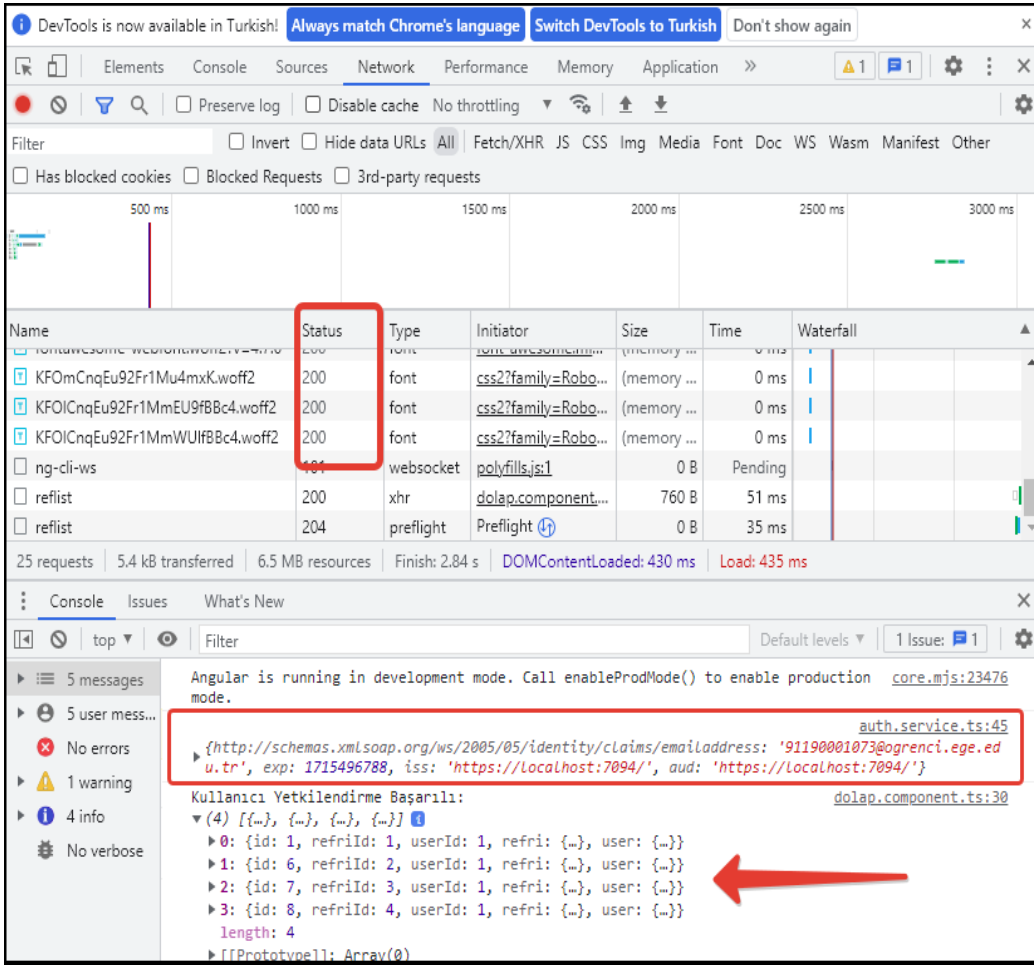
Web sitesi tasarımı, JavaScript tabanlı bir web uygulama geliştirme platformu olan Angular framework'den yararlanılmıştır. Angular modüler yapısı sayesinde uygulamayı bileşenlerine bölerek daha esnek bir tasarımı sağlamaktadır. Kodlama, Visual Studio Kod Geliştirme Ortamında, TypeScript dili ile geliştirilmiştir.

Masaüstü uygulamasında olduğu gibi, web sitesinde de kullanıcı adı ve giriş sayfası bulunmakta ve sadece yetkili kullanıcılar giriş yapabilmektedir (Şekil 4.31). Veritabanında kayıtlı tüm kullanıcılar, sadece yetkili olduğu dolapların verilerine erişebilmektedir.



Şekil 4.32: Kullanıcı girişi ile JWT token alınması

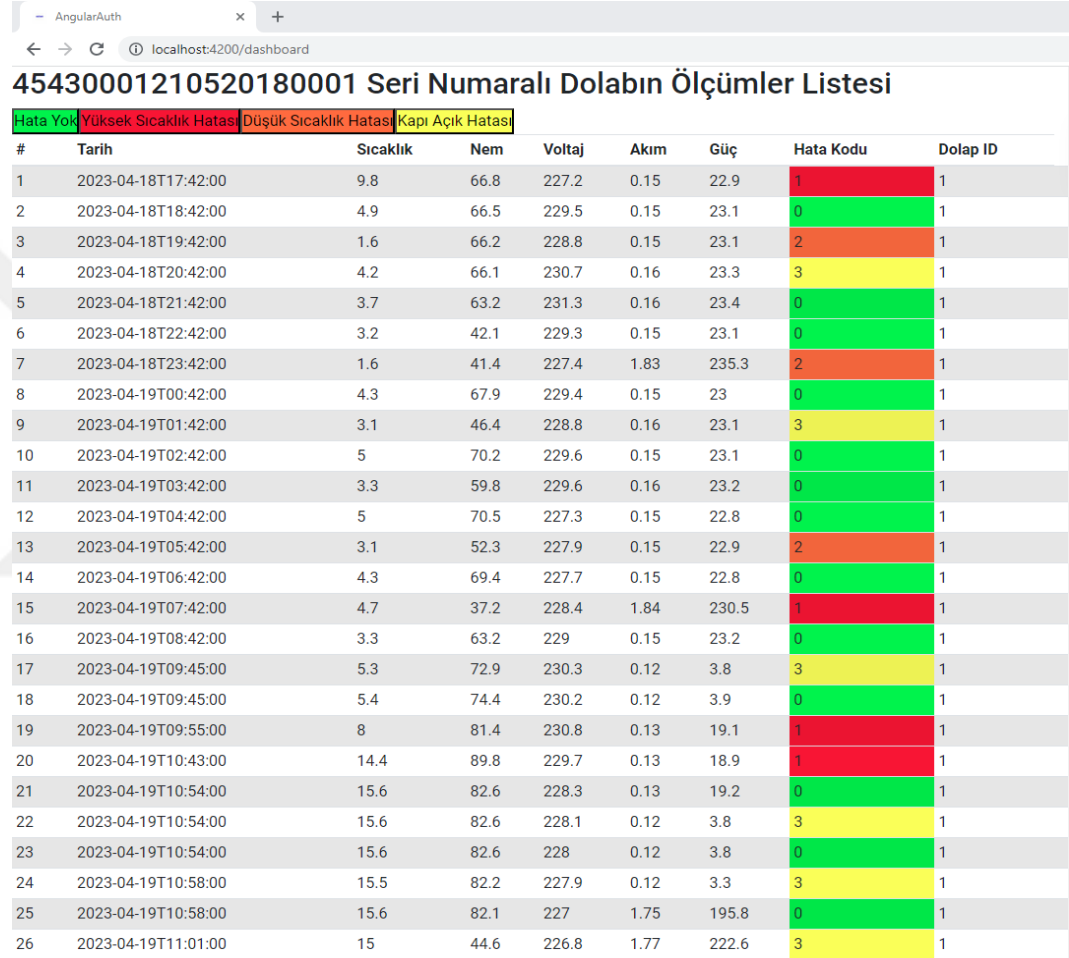
Şekil 4.32’de Visual Studio Code analiz ekranında, ”http://dolapkontrol-001-site1.itemurl.com/Auths/login” API adresine kullanıcı girişi isteği yapılması gözlemlenmektedir. Görüldüğü gibi istek sonunda durum olarak ”HTTP 200” başarılı kodu alınmıştır. Hemen aşağısında yer alan konsol ekranında ise elde edilen JWT token değeri görülmektedir.



Şekil 4.33: Yetkili olunan dolapların öğrenilmesi

Şekil 4.33’de ise, yine analiz ekranının konsol kısmında, kullanıcıya ait dolaplar görüntülenmektedir. Görüldüğü gibi istek yapılan API tarafından kullanıcı Id değeri ”1” ve yetkili olunan dolap Id değerleri ”1”, ”2”, ”3” ve ”4” olarak gönderilmiştir. JSON formatı içerisindeki tüm etiketler Id değeri olarak verilmiştir. Bunun sebebi bir önceki bölümlerde incelenen tablolardaki ilişkilendirme özelliğindendir.

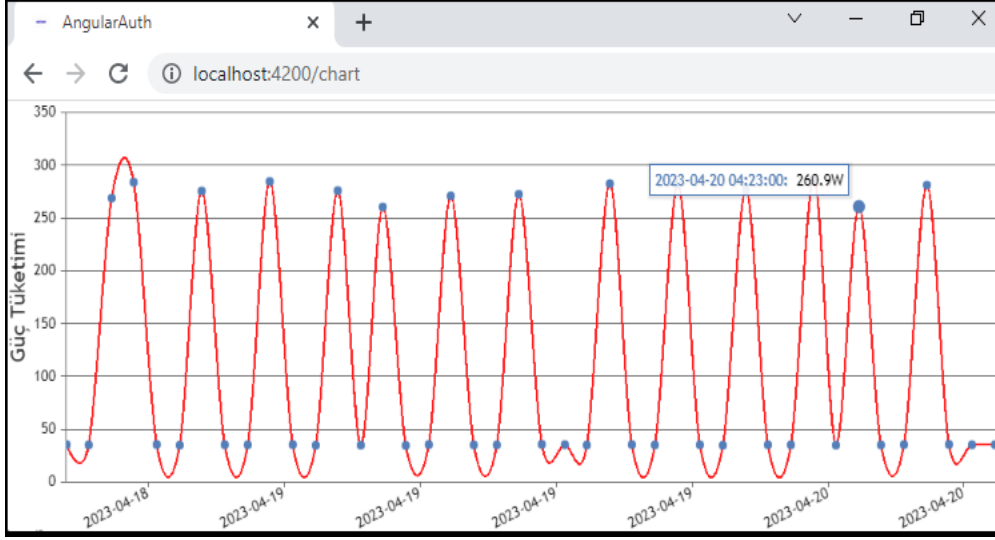
Son kullanıcıya ait olan 1 numaralı Id değerine sahip ve seri numarası ”45430001210520180001” olan dolabın verileri görülmektedir (Şekil 4.34). Alınan hata kodları analiz sırasında kolaylık sağlaması için hata cinsine göre renklendirilmiştir. Hatanın olmadığı durumlarda yeşil, yüksek sıcaklık durumunda kırmızı, düşük sıcaklık durumunda turuncu ve kapının belli bir süreden fazla açık kalması durumunda sarı renkle kodlanmıştır.



#	Tarih	Sıcaklık	Nem	Voltaj	Akım	Güç	Hata Kodu	Dolap ID
1	2023-04-18T17:42:00	9.8	66.8	227.2	0.15	22.9	1	1
2	2023-04-18T18:42:00	4.9	66.5	229.5	0.15	23.1	0	1
3	2023-04-18T19:42:00	1.6	66.2	228.8	0.15	23.1	2	1
4	2023-04-18T20:42:00	4.2	66.1	230.7	0.16	23.3	3	1
5	2023-04-18T21:42:00	3.7	63.2	231.3	0.16	23.4	0	1
6	2023-04-18T22:42:00	3.2	42.1	229.3	0.15	23.1	0	1
7	2023-04-18T23:42:00	1.6	41.4	227.4	1.83	235.3	2	1
8	2023-04-19T00:42:00	4.3	67.9	229.4	0.15	23	0	1
9	2023-04-19T01:42:00	3.1	46.4	228.8	0.16	23.1	3	1
10	2023-04-19T02:42:00	5	70.2	229.6	0.15	23.1	0	1
11	2023-04-19T03:42:00	3.3	59.8	229.6	0.16	23.2	0	1
12	2023-04-19T04:42:00	5	70.5	227.3	0.15	22.8	0	1
13	2023-04-19T05:42:00	3.1	52.3	227.9	0.15	22.9	2	1
14	2023-04-19T06:42:00	4.3	69.4	227.7	0.15	22.8	0	1
15	2023-04-19T07:42:00	4.7	37.2	228.4	1.84	230.5	1	1
16	2023-04-19T08:42:00	3.3	63.2	229	0.15	23.2	0	1
17	2023-04-19T09:45:00	5.3	72.9	230.3	0.12	3.8	3	1
18	2023-04-19T09:45:00	5.4	74.4	230.2	0.12	3.9	0	1
19	2023-04-19T09:55:00	8	81.4	230.8	0.13	19.1	1	1
20	2023-04-19T10:43:00	14.4	89.8	229.7	0.13	18.9	1	1
21	2023-04-19T10:54:00	15.6	82.6	228.3	0.13	19.2	0	1
22	2023-04-19T10:54:00	15.6	82.6	228.1	0.12	3.8	3	1
23	2023-04-19T10:54:00	15.6	82.6	228	0.12	3.8	0	1
24	2023-04-19T10:58:00	15.5	82.2	227.9	0.12	3.3	3	1
25	2023-04-19T10:58:00	15.6	82.1	227	1.75	195.8	0	1
26	2023-04-19T11:01:00	15	44.6	226.8	1.77	222.6	3	1

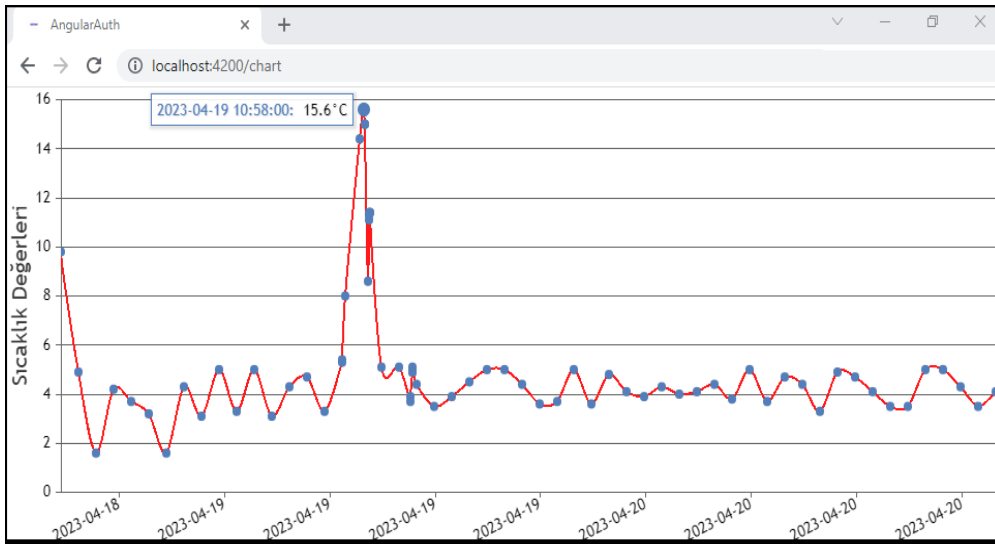
Şekil 4.34: Buzdolabı ölçümlerinin listelenmesi

Şekil 4.34’de görüldüğü gibi, ”45430001210520180001” seri numaralı dolaba ait geniş bir bilgi havuzu oluşmuş olmaktadır. Örneğin, soğutma için kompresör devrede iken yaklaşık 235 Watt güç tüketimi söz konusu iken, istenilen set değerine gelinmesi ve kompresörün durması ile birlikte bu değer yaklaşık 23 Watt değerine düşmektedir.



Şekil 4.35: Web sitesi - Güç tüketimi grafiği

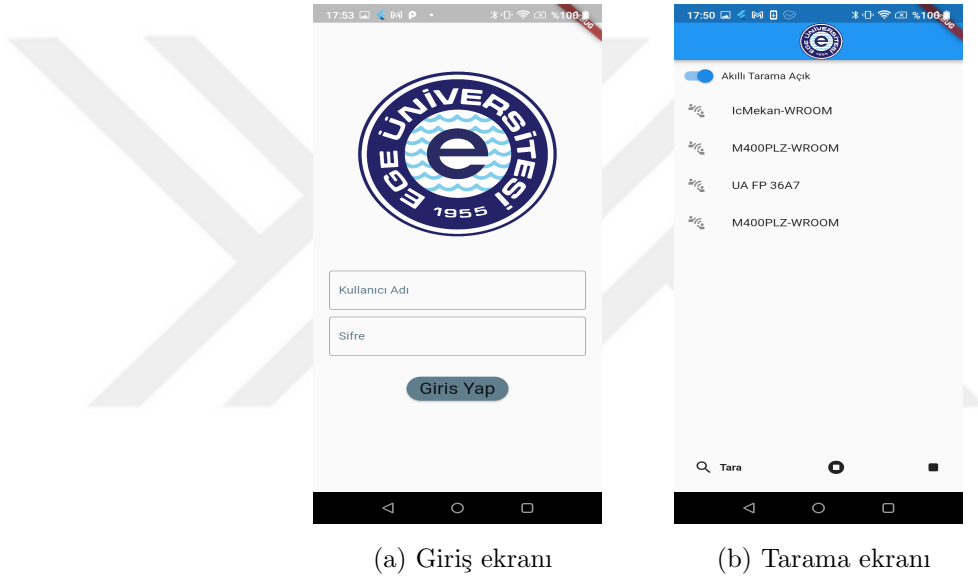
Tüm ölçümler, istenildiğinde bir grafik olarak da gözlemlenebilmektedir. Bu bağlamda, Şekil 4.35’de güç tüketimi grafiği verilmiştir. Liste görünümünde olduğu gibi, burada da termostatik hareket sırasındaki iniş çıkışlar göze çarpmaktadır. Görüldüğü gibi şeklin üzerinde tıklandığında ayrıntılı veri bilgisine de ulaşılabilir. Şekil 4.36’da ise, bir hata durumuna bağlı, ani sıcaklık yükselmesi gözlemlenmektedir.



Şekil 4.36: Web sitesi - Kabin sıcaklığı grafiği

4.3.3 Mobil Uygulama Yazılımı

Dolap verilerinin bir mobil cihaz üzerinde görüntülenmesi, bluetooth (BLE) bağlantısı ile sağlanmıştır. Yazılım geliştirme sürecinde, öncelikle Flutter Yazılım Geliştirme Kitinden (Software Development Kit (SDK)) yararlanılarak, Visual Studio Code IDE üzerinde dart dili ile kodlanmıştır. Ayrıca bluetooth kodlarının aktif kullanılabilmesi için "flutter_blue" kütüphane paketi yüklenmiştir. Daha sonra bu kütüphanenin sağlamış olduğu unsurları kullanabilmek için mobil cihaz üzerinden gerekli izinler elde edilmiştir.



Şekil 4.37: Mobil uygulama bluetooth bağlantısı

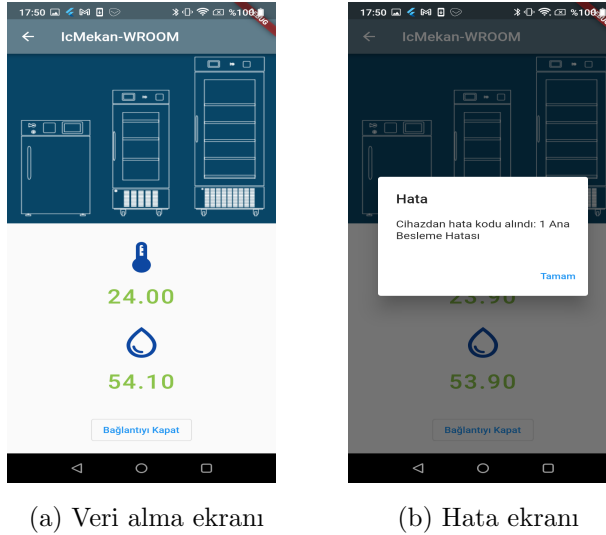
Mobil uygulamaya giriş için, Şekil 4.37(a)'da görüldüğü gibi, kullanıcı adı ve şifresi gereklidir. Geliştirilen yazılımda bu bilgiler yazılım içerisine kodlanmıştır. Kullanıcı adı ve şifresinin doğru olması durumunda Şekil 4.37(b)'de görüldüğü gibi, tarama ekranına geçilmesine izin verilmektedir. Tarama ekranında, mobil uygulamanın bağlanabileceği aktif olan tüm dolap isimleri listelenmektedir. Gerekli eşleşmenin sağlanmasının ardından, bu bağlantı üzerinden dolap sensör verileri alınmaya başlanmaktadır. Ayrıca herhangi bir hata durumunda da hata bildirimi alınabilmektedir.

Önceki bölümlerde bahsedilen, ESP32 üzerinden gelen bluetooth verilerinin mobil uygulamada bir modele aktarım işlemi Visual Studio Code üzerindeki analiz bölümünde teyit edilmektedir (Şekil 4.38).



Şekil 4.38: Visual Studio Code - Analiz

Şekil 5.37(a)'da ise, bu işlemin canlı analiz şeklindeki gösterimi görülmektedir. Aynı zamanda, Şekil 4.39(b)'de ise anlık olarak gelen "Besleme Yok Hatası" mesaj şeklinde görüntülenmektedir.



(a) Veri alma ekranı

(b) Hata ekranı

Şekil 4.39: Mobil uygulama arayüzü

5 TARTIŞMA

Geliştirilen IoT sistemi oldukça zorlu koşullara ve stres testlerine maruz bırakılmıştır. İnternet bağlantısındaki aksaklıklar, LCD panel sayfaları arasında hızlı geçişler, dolap set parametrelerinin değiştirilip yüksek veya düşük sıcaklık hatalarına sebebiyet vermek, buzdolabının besleme gerilimin kesilmesi ve kapısının çok sık açılıp kapatılarak kapı açık hatalarını görüntülemek gibi birçok senaryo denemesi yapılmıştır.

İnternet bağlantısındaki aksaklıklara yönelik ESP32 modül içerisindeki gömülü sistem yazılımında, tekrar bağlantı kurulması için birçok algoritma çalıştırılmaktadır.

Ayrıca kuramsal temeller bölümünde bahsedilen "Son İstek ve Vasiyetname" (Last Will and Testament) özelliği sayesinde MQTT aracısı tüm abonelere ilgili dolabın artık sisteme bağlı olmadığı mesajını göndermektedir. Şekil 5.1'de görüldüğü gibi, AC50S dolabının internet bağlantısı koparılarak bildirim gönderilmesi test edilmiştir. Son İstek ve Vasiyetname özelliği test edilirken konunun daha net anlaşılabilmesi için şifrelenmemiştir. Diğer bilgilerin şifreli olduğu aynı şekil üzerinde gözlemlenebilir.

aesMQTT	team 91190001073@ogrenci.ege.edu.tr
SifreliOlcumBilgileri	BRap/rV4wNufgjk4tse97LT/w/NtfXeD0KdX8rtVap4vHmbI/yaVIV2FSsEg4Mi3KNKqSpMWkGTyJW8VDLe
SifreliOlcumBilgileri	yA5sBZhTE7oYUeI/L9TQkZBQ96ApQkq4H3ZEn5m6YzN+Ut2LZLf4tORGUWNpzMDmFHRjnGsfrn89Bm
SifreliOlcumBilgileri	RKFVXqILFiAWwUmHR8yiNGGUINO6lt4Es719R/VMDJolftF98rbcnIKEi6JLU/Jo2t6ATxqe731otSyFI526x0
CevrimdisiDolap	{"Dolap":"AC50S","CevrimdisiESPid":3,"status":0}
SifreliOlcumBilgileri	BRap/rV4wNufgjk4tse97B66xYahx3UkE5sZohUUbjVdjfZ4pQPuy1mcP7uoKybK7K1DjumkmKREs+WNU
SifreliOlcumBilgileri	yA5sBZhTE7oYUeI/L9TQkd7R3Uo3zlySHL6GV46BPUY2PyJCDac6B/0EFAsUU+gDmFHRjnGsfrn89Bm6Rn
SifreliOlcumBilgileri	soB6aOKbHvh96Bw4dUEJVJBQ96ApQkq4H3ZEn5m6YzNEuqgTVLX4OtuAxlrgunPbt2LH6Am3f+wrNLx
SifreliOlcumBilgileri	RKFVXqILFiAWwUmHR8yiNGGUINO6lt4Es719R/VMDJolftF98rbcnIKEi6JLU/Jo2t6ATxqe731otSyFI5260
SifreliOlcumBilgileri	BRap/rV4wNufgjk4tse97DTOMecokVEGgWXIwVDr0deiXadhC/6e9cWG58tsUdFm7K1DjumkmKREs+W

Şekil 5.1: MQTT Broker - Son İstek ve vasiyetname özelliği

Buzdolaplarının olası hata durumları da ayrıntılı incelenmiş ve test edilmiştir. Örneğin, besleme geriliminin kesilmesi durumunda 12V 7A değerinde bir akü ilgili alarm için devreye girmektedir (Şekil 5.2). Akünün soğutma performansına bir katkısı bulunmamaktadır. Sadece son kullanıcının uyarılması için geliştirilmiş bir mekanizmadır.



Şekil 5.2: Besleme geriliminin kesilmesi testi

Akü vasıtasıyla, anakart üzerindeki işlemci ve LCD besleme gerilimi sağlanmış olmaktadır. Bu sayede LCD üzerinden görsel bir uyarı yapılırken Wi-Fi bağlantısı veya Bluetooth bağlantısı üzerinden kullanıcının uzak erişim bağlantısı devam etmektedir. Elektrik kesintilerinde Wi-Fi ağının çökmesi de ayrıca değerlendirilmiş ve bu durumda Bluetooth bağlantısının aktif olarak kullanılmaya devam edip, hata bildirimlerinin mobil uygulama üzerinden devam ettiği gözlemlenmiştir.

Dolap ölçümlerinin alındığı sensör modüllerinde bir problem olması durumunda, son kullanıcı LCD ekran üzerinden görsel olarak uyarılmaktadır (Şekil 5.3). Bunun yanında, kayıt mekanizması halen aktiftir ve problem kaynağı testpitinde kullanılabilmektedir.



Şekil 5.3: Sensör arızalı olması testi

Bir diğer önemli hata senaryosu, kapının açık unutulması veya tam kapanmamasıdır (Şekil 5.4). Bu durum, dolap kapısında takılı olan bir anahtar tarafından algılanıp kullanıcı LCD ekran üzerinden uyarılmaktadır. Bu hata bilgileri aynı zamanda veri tabanına kayıt edilmekte ve sonrasında analizlere imkan sunmaktadır. Tüm analizler bir önceki bölümlerde bahsedildiği gibi masaüstü ve websitesi üzerinden yapılabilmektedir.

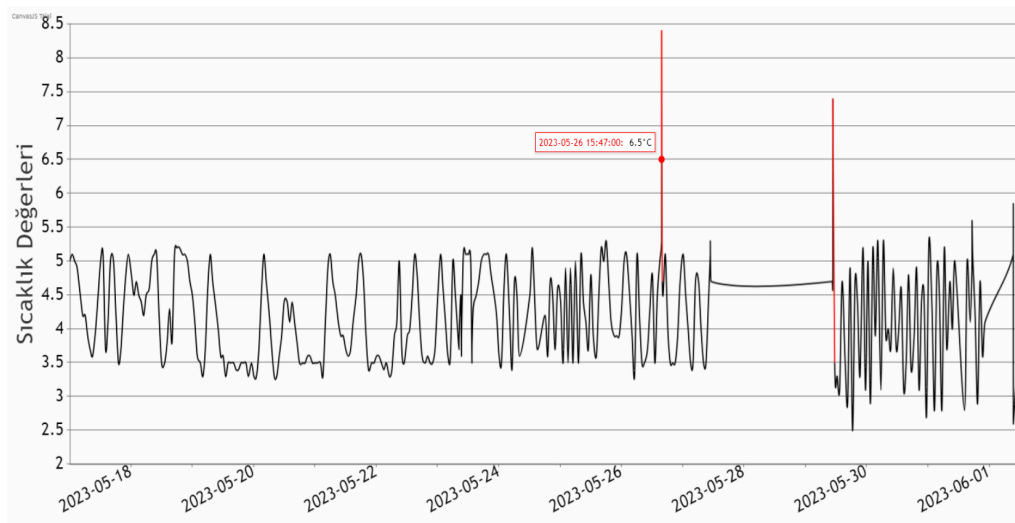


Şekil 5.4: Kapı açık kalması testi



Şekil 5.5: Yüksek sıcaklık limit testi

Kapı açık bırakılması veya sık açama/kapama stres testi ile birlikte, yüksek sıcaklık limitine ulaşıldığı gözlemlenmiştir (Şekil 5.5). Bu durum, aynı zamanda kompresör veya fan arızası durumlarında da gözlemlenebilecek bir hatadır. Yüksek sıcaklık limitleri kullanıcı tarafından ayarlar menüsünden ayarlanabilmektedir.



Şekil 5.6: Test sonuçlarının grafiksel incelenmesi

Buzdolapları ekranlarında gözlemlenen hata durumları aynı zamanda veri tabanına da kaydedilmektedir. Bu sayede web sitesi üzerinden yapılan izleme işleminde grafik üzerinde yüksek sıcaklık limitine ulaşıldığı gözlemlenebilmektedir.

Tüm Veri Listesi									
#	Tarih/Saat	Hata	Sıcaklık	Nem	Gerilim	Akım	Güç	Tüketimi	Dolap ID
234	2023-05-26T15:47:00	Kapı Açık	5.3°C	%83.8	229.8V	0.14A	7.9W		1
235	2023-05-26T15:47:00		5.9°C	%85.9	230.7V	1.89A	260W		1
236	2023-05-26T15:47:00	Kapı Açık	6.5°C	%87	229.4V	1.84A	243.7W		1
237	2023-05-26T15:47:00	Yüksek Sıcaklık	6.9°C	%87.5	230.4V	1.91A	259.6W		1
238	2023-05-26T15:48:00	Kapı Açık	7.6°C	%87.8	230.4V	1.86A	248.6W		1
239	2023-05-26T15:48:00	Yüksek Sıcaklık	7.8°C	%88	230.4V	1.86A	246.9W		1
240	2023-05-26T15:48:00	Kapı Açık	8.2°C	%88.7	230.3V	1.85A	244.8W		1
241	2023-05-26T15:48:00	Yüksek Sıcaklık	8.4°C	%88.8	230.2V	1.92A	259.6W		1

Şekil 5.7: Hataların veri listesinde incelenmesi

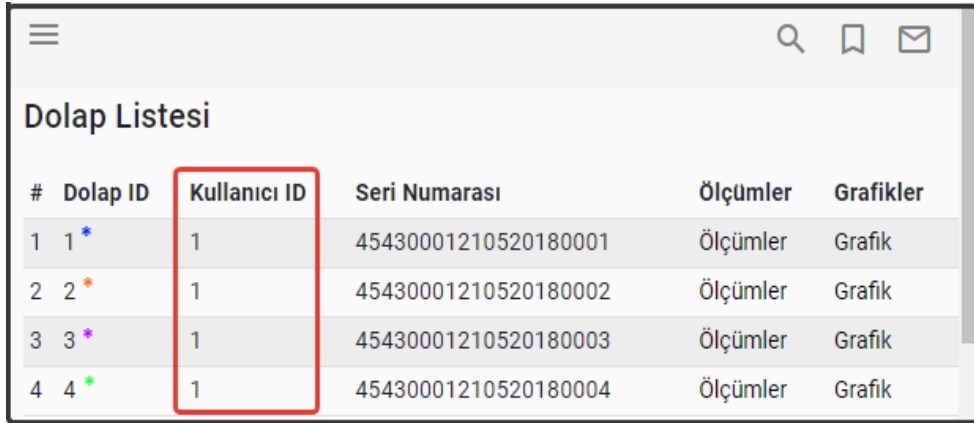
Bu tez çalışması kapsamında gerçekleştirilen önemli noktalardan birisi kullanıcı yetkilendirmesidir. Kullanılan dolabın birden fazla kullanıcısının olduğu durumda, her bir kullanıcıya ayrı yetki atanabilmektedir.

id	Dolap Seri Numarası
1	45430001210520180001
2	45430001210520180002
3	45430001210520180003
4	45430001210520180004
5	45430001210520180005
6	45430001210520180006
7	45430001210520180007
8	45430001210520180008

ID	Kullanıcı Email	Kullanıcı Şifre
1	deneme1@den.com	123
2	deneme2@den.com	123
3	deneme3@den.com	123
4	deneme4@den.com	123
5	deneme5@den.com	123
6	deneme6@den.com	123
7	deneme7@den.com	123
8	deneme8@den.com	123

Şekil 5.8: Kullanıcıların yetkilendirilmesi

Ayrıca birden fazla dolap ve kullanıcının olduğu kurumlarda, kullanıcıların sadece yetkili olduğu dolaba erişmesi isteniyorsa, üretici firma masaüstü uygulamasının aracılığı ile her bir kullanıcıya farklı dolapların tanımlaması da yapılabilmektedir (Şekil 5.8). Gerçekleştirilen yetkilendirmenin ardından, web sitesi üzerinden sisteme giriş yapan son kullanıcı, sadece kendisine ait olan dolapları görüntülemektedir (Şekil 5.9).



#	Dolap ID	Kullanıcı ID	Seri Numarası	Ölçümler	Grafikler
1	1 *	1	45430001210520180001	Ölçümler	Grafik
2	2 *	1	45430001210520180002	Ölçümler	Grafik
3	3 *	1	45430001210520180003	Ölçümler	Grafik
4	4 *	1	45430001210520180004	Ölçümler	Grafik

Şekil 5.9: Id değeri 1 olan bir kullanıcının yetkili olduğu dolaplar

6 SONUÇ

Bu tez çalışması kapsamında MQTT protokol yapısından yararlanılarak oldukça esnek ve şifre korumalı bir IoT ağı oluşturulmuştur. Şifreleme için günümüzün en güçlü algoritması olan AES simetrik şifrelemenin seçilmesi, IoT sistemlerinin güvenliği konusunda farklı bir bakış açısı sağlamıştır. Tasarlanan IoT sisteminin ana amacı olan takip ve kontrol için geliştirilen, 3 ayrı akıllı uygulama bu tez çalışması kapsamında gerçekleştirilen en önemli katkılar arasındadır.

Bahsedilen akıllı uygulamalardan birisi olan masaüstü uygulamasında, MQTT protokol üzerinden anlık kontrol ve takip sağlanırken, belirli bir dolabın geçmişine ait bilgiler, bir API üzerinden veritabanına ulaşılarak alınabilmektedir. Diğer bir uygulama olan web sitesi aracılığıyla, herhangi bir dolabın tüm ölçüm parametrelerine liste veya grafik olarak ulaşılabilmesi sağlanmıştır. Web sitesine girişte kullanıcı doğrulaması yapılması, sistem güvenliğini bir üst düzeye taşımıştır. Kullanım ortamında Wi-Fi ağı olmayabileceği düşünülerek, son kullanıcı için bluetooth (BLE) bağlantısı ile çalışan bir mobil uygulama geliştirilmiş ve kullanıcı takibi kolaylaştırılmıştır.

Bu çalışmanın ulaşmak istediği hedeflerden birisi olan verimlilik konusundaki iyileştirmede ise, enerji tüketiminin ve hata bildirimlerinin bir veritabanında kayıt altına alınması ile önemli bir veri kaynağı oluşturulmuştur. Bu bilgi havuzu sayesinde, gerekli olabilecek tüm sistemsel iyileştirmelerin önü açılmıştır.

Sık sık enerji kesintilerinin olduğu bölgelerde, donanım kısmındaki akü girişi sayesinde uzak ve yakın alarm sistemi kurtarıcı çözüm niteliğindedir. Akünün soğutma sistemine katkısı olmamakla birlikte, ürünlerin başka bir ortama taşınabilmesi için uyarıcı haber mekanizması olarak kullanılmaktadır.

Ayrıca sistemdeki tüm yazılım bileşenlerinin, bu çalışma içerisinde geliştirilmesi ve uygun bir elektronik kart tasarımı ile maliyetin oldukça düşük olması sağlanmıştır. Bu sayede, hali hazırda kullanılan bir buzdolabına bile entegre edilebilir olması sağlanmıştır.

Gelecek çalışması olarak, geliştirilen bu IoT sistemine, istenilen herhangi bir cihazın veya teknolojinin entegre edilebilmesinin mümkün olması sağlanmıştır. Günümüzde yapılan herhangi bir iyileştirme çalışmasına da entegre edilebilir olarak tasarlanmıştır.

Bu çalışma, güçlü ve güvenli IoT sistemleri oluşturulması için iyi bir alt yapı sağlayabileceği gibi, aynı zamanda diğer şifreleme algoritmaları ile güvenlik seviyesi daha da arttırılabilir.

3 ayrı katagoride, 7 farklı yazılım olarak geliştirilen bu IoT sisteminin kodlama yapısı, yeni başlanan geniş ölçekli projeler için farklı bir bakış açısı kazandırabilir.

Kaynaklar

- Adam, S. I., Moedjahedy, J. H., Maramis, J.,** (2020). 2nd International Conference on Cybernetics and Intelligent System (ICORIS), doi: 10.1109/ICORIS50180.2020.9320801
- Ahmad, N., Yusof, I. S. M., Abdullah, N. E., Jamaludin, N. F., Kamaruzzaman, D., Anuar, N.** (2021) The Development of Smart Home Energy Monitoring System”, IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), (pp. 293-298) doi: 10.1109/ICSGRC53186.2021.9515294,
- Ajitomi, D., Kawazoe, H., Minami, K., Esaka, N.,** (2015) A Cost-Effective Method to Keep Availability of Many Cloud-Connected Devices, IEEE 8th International Conference on Cloud Computing, (pp. 1-6) doi: 10.1109/CLOUD.2015.11
- Bayılmış, C., Ertürk, İ., Çeken, C.,** (2004). Kablosuz Bilgisayar Ağlarının Karşılaştırmalı İncelemesi, Politeknik Dergisi, Cilt 7, Sayı 3, (ss. 201-210)
- Carratú, M., Gallo, V., Paciello, V.,** (2022). Communication among smart sensors using MQTT protocolT, 2022 IEEE International Symposium on Measurements & Networking (M & N)),(pp. 1-6) doi: 10.1109/MN55117.2022.9887708
- Chang, K. H.,** (2014) , ”Bluetooth: a viable solution for IoT?,” IEEE Wireless Communications, vol.21, no.6, (pp. 6-7) doi: 10.1109/MWC.2014.7000963
- Devireddy, Y., Sanke, M., Maganti, R. H., Thorlikonda, A.,** (2021) Real Time Energy Monitoring And Controlling System using IoT, 2nd International Conference on Smart Electronics and Communication (ICOSEC), (pp. 157-161) doi: 10.1109/ICOSEC51865.2021.9591867
- Espressif Systems** (2023) ”ESP32WROOM32 Datasheet”, https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (Erişim tarihi: 10 Mayıs 2023)

- Fan, S., G, Y., Yu, Xiang.,** (2022) Comparison Analysis and Prediction of Modern Wi-Fi Standards, 2022 International Conference on Big Data, Information and Computer Network (BDICN), (pp. 581-585) doi: 10.1109/BDICN55575.2022.00112
- Gierłowski, K., Kostuch, A., Woźniak, J., Nowicki, K.,** (2011) Testbed analysis of video and VoIP transmission performance in IEEE 802.11 b/g/n networks, *Telecommunication Systems*, 48:247–260 pp, doi: 10.1007 s11235-010-9341-z
- Hmissi, F. & Ouni, S.,** (2022). TD-MQTT: Transparent Distributed MQTT Brokers for Horizontal IoT Applications, 5th International Conference on Trends in Electronics and Informatics (ICOEI), (pp. 868-872) doi: 10.1109/SETIT54465.2022.9875881
- Fidaeldin, A. Y., & Abdulrazzaq, A. H.,** (2019) Reducing HTTP Request times through Utilizing BGP: An Experimental Study, International Conference on Fourth Industrial Revolution (ICFIR), (pp. 1-5) doi: 10.1109/ICFIR.2019.8894793
- Gaikwad, K.,** (2021) IoT based Water Management System using MQTT protocol, IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)), (pp. 408-414) doi: 10.1109/ICOEI51242.2021.9452900
- Jutadhamakorn, P., Pillavas, T., Visoottiviseth, V., Takano, R., Haga, J., Kobayashi, D.,** (2017) 2nd International Conference on Information Technology (INCIT), (pp. 1-5). doi: 10.1109/INCIT.2017.8257870
- Khasanova, A. M.,** (2021) Detection of Attacks on Wi-Fi Access Points, IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), (pp. 28-32). doi: 10.1109/ElConRus51938.2021.9396420
- Kodali, R. K. & Soratkal, S.,** (2016). MQTT based home automation system using ESP8266, 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), (pp. 1-5). doi: 10.1109/R10-HTC.2016.7906845

Longo, E., Enrico, A., Redondi, C, Cesana, M., Manzoni, P., (2022) A Low Cost Internet of Things (LCIoT) Based System for Monitoring and Control of UPS System using Node-Red, CloudMQTT and IBM Bluemix, IEEE Internet of Things Journal, vol.9, no.18,(pp. 17728 - 17740) doi: 10.1109/ICECCE47252.2019.8940686

Luna, M., Tona, G., Piazza, M. C., Pucci, M. Accetta, A., Taibi, D., Vetro, C., Grassa, R., (2018) A Prototype of Wireless Sensor for Data Acquisition in Energy Management Systems, IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I & CPS Europe), (pp. 1-6) doi: 10.1109/EEEIC.2018.8494413

Mehmood, M. U., Ali, W., Ulasyar, A., Zad, H. S., Khattak, A., Imran, K., (2019) International Conference on Electrical, Communication, and Computer Engineering (ICECCE), (pp. 1-5). doi: 10.1109/ICECCE47252.2019.8940686

Mishra, B., & Kertesz, A., (2020) The Use of MQTT in M2M and IoT Systems: A Survey, in IEEE Access, (Vol. 8 pp. 201071-201085). doi: 10.1109/ACCESS.2020.3035849,

Patel, H. K. Mody, T. Goyal, A. (2019) Bluetooth Low Energy (BLE) Security and Privacy, CloudMQTT and IBM Bluemix, 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU).(pp. 1-6) doi: 10.1109/IoTSIU.2019.8777490,

Espressif Systems (2021) "PZEM-004T V3.0 User Manual", <http://en.peacefair.cn/products/608.html> (Eriřim tarihi: 1 Mayıs 2023)

Rao,B. N., & Sudheer, R., (2020).Energy Monitoring using IoT, International Conference on Inventive Computation Technologies (ICICT) (pp. 868-872) doi: 10.1109/ICICT48043.2020.9112426

Raiker, G. A., Reddy, Subba R. B., Umanand, Umanand, L., Agrawal, S., Thakur, A. S., Ashwin K., Barton, J. P., Thom-

- son, M.**, (2020). IEEE International Conference on Power Electronics, Smart Grid and Renewable Energy (PESGRE2020), (pp. 1-5). doi: 10.1109/PESGRE45664.2020.9070739
- Roig, P. J., Alcaraz, S., Gilly, K., Juiz, C.**, (2019) Algebraic Formal Modelling for HTTP Main Methods using ACP, 23rd International Conference Electronics, (pp. 1-8) doi: 10.1109/ELECTRONICS.2019.8765572
- Sadeq, A. S., Hassan, R., Al-rawi, S. S., Jubair, A. M., Aman, A. H. M.**, (2019) A Qos Approach For Internet Of Things (Iot) Environment Using Mqtt Protocol, International Conference on Cybersecurity (ICoCSec), (pp. 59-64). doi: 10.1109/ICoCSec47621.2019.8971097
- Setiawan, F. B. & Magfirawaty** (2021). Securing Data Communication Through MQTT Protocol with AES-256 Encryption Algorithm CBC Mode on ESP32-Based Smart Homes , 2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE), (pp. 166-170). doi: 10.1109/COSITE52651.2021.9649577
- Stranne,A., Floren,F., Edfors, O., Molin, B. A.**, (2002) THROUGHPUT OF IEEE 802.11 FHSS NETWORKS IN THE PRESENCE OF STRONGLY INTERFERING BLUETOOTH NETWORKS, The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,(pp. 161-166) doi: 10.1109/PIMRC.2002.1046681
- Tantidham, T., Ngamsuriyaros, S., Tungamnuayrith, N., Nildam, T. ,Banthao, K., Intakot, P.** (2018) Energy Consumption Collection Design for Smart Building, International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES),(pp. 1-6) doi: 10.1109/ICESIT-ICICTES.2018.8442052
- Varol, A. B.**, (2019). Compilation of Data Link Protocols: Bluetooth Low Energy (BLE), ZigBee & Z-Wave, 4th International Conference on Computer Science and Engineering (UBMK), (pp. 85-91). doi: 10.1109/UBMK.2019.8907068

- Vikram, S. M., Aswin, R., Rahul, S., Harish, A., Deepa, K., Manitha, P. V., Sailaja, V.,** (2022). Internet of Things based Smart Energy Meter with Fault Detection Feature using MQTT Protocol, 8th International Conference on Advanced Computing and Communication Systems (ICACCS), (pp. 1592-1596). doi: 10.1109/ICACCS54159.2022.9785026
- Wu , X. & Li, N.,** (2018). Improvements of MQTT Retain Message Storage Mechanism, 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC),(pp. 957-961) 10.1109/IMCEC.2018.8469192
- Zare, A., & Iqbal, M. T.,** (2020).Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System, 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), (pp. 1-5) doi: 10.1109/IEMTRONICS51293.2020.9216412
- Zhang, Y., Weng, J., Dey, R., Fu, X.,** (2019) Arduino Based Smart Energy Meter using GSM, Encyclopedia of Wireless Networks.(pp. 1-12) doi: 10.1007/978-3-319-32903-1_298-1

TEŞEKKÜR

Yüksek Lisans eğitimim boyunca engin deneyimleri ile yol gösteren değerli danışman hocam Prof. Dr. Radosveta Sokullu'ya sonsuz teşekkür ederim.

Ayrıca tez çalışmam boyunca her zaman desteğini hissettiğim ve Öztaş Akademi olarak adlandırdığım değerli şirketime teşekkürü bir borç bilirim.

Son olarak, bu çalışmamın tamamlanmasını sabırla bekleyen ve hatta uykusuz gecelerime ortak olan tüm sevdiklerime kucak dolusu teşekkür ederim.

15/05/2023

Ümit Güler

ÖZGEÇMİŞ

Ümit GÜLER, 2004 yılında İTO Vakfı Süleyman Taştekin Anadolu Meslek Lisesi'nde Endüstriyel Elektronik bölümü ile başladığı mesleki eğitimine, 2008 yılında Ege Meslek Yüksek Okulu'nda Elektronik Haberleşme bölümü ile devam etti. Buradaki eğitimi sırasında "ODTÜ Robot Günleri" ve "İTÜ Robot Olimpiyatları"nda okuduğu üniversiteyi temsil etti. "Ege Mucidini Arıyor" yarışmasında 3.lük ödülüne layık görüldü. Bu eğitiminden mezun olduktan sonra 2010-2011 yılları arasında Ermaş Elektronik'te elektronik teknikeri olarak çalıştı. 2011 yılında Niğde Üniversitesi Elektrik-Elektronik Mühendisliği bölümünü kazanmasıyla 4 yıl boyunca bu kentte yaşamına devam etti. Bu süre zarfında IEEE RAS kulübünde aktif rol aldı. 3.25 ortalama ile mezun olarak Onur Belgesi almaya hak kazandı. 2015 yılında mezuniyetinin ardından 5 yıl boyunca Ermaş Elektronik'te Ar-Ge mühendisi olarak çalıştı. 2020 yılında Ege Üniversitesinde Elektrik-Elektronik Mühendisliği Anabilim Dalında, Elektronik Yüksek Lisans programı eğitimine başladı. Aynı yıl, Manisa Organize Bölgesinde yer alan Öztaş Global Soğutma firmasından aldığı Yazılım ve Donanım Geliştirme Yöneticiliği teklifi üzerine, görevine burada devam etme kararı almıştır.

EKLER

EK-1 Kodlama Parçacıkları

```

1 # ESP32 Wi-Fi MQTT kod parcacigi
2
3 void wifiTask(void *pvParameters){
4     static const char *wifiTask = "wifiTask";
5     esp_log_level_set(wifiTask, ESP_LOG_INFO);
6
7     while (1)
8     {
9         unsigned long currentMillis = millis();
10        if (currentMillis - previousMillis >= interval){
11            String encrypted_str = aesB64.ency_arr2str((char*)
12            payload.c_str(), encryption_key);
13            MQTTclient.publish(pubTopic1, (char*) encrypted_str.c_str()
14            );
15            previousMillis = currentMillis;
16        }
17        vTaskDelay(2000 / portTICK_PERIOD_MS);
18    }
19 }

```

```

1 # ESP32 BLE kurulum kod parcacigi
2
3 BLEDevice::init("ESP32wroom32");
4 pServer = BLEDevice::createServer();
5 pServer->setCallbacks(new MyServerCallbacks());
6 BLEService *pService = pServer->createService(SERVICE_UUID);
7 pCharacteristic = pService->createCharacteristic(
8     CHARACTERISTIC_UUID,
9     BLECharacteristic::PROPERTY_READ |
10    BLECharacteristic::PROPERTY_WRITE |
11    BLECharacteristic::PROPERTY_NOTIFY |
12    BLECharacteristic::PROPERTY_INDICATE
13    );
14 pCharacteristic->addDescriptor(new BLE2902());
15 pService->start();
16 BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();

```

```

17 pAdvertising->addServiceUUID(SERVICE_UUID);
18 pAdvertising->setScanResponse(false);
19 pAdvertising->setMinPreferred(0x0);
20 BLEDevice::startAdvertising();

```

```

1 # ESP32 Olcumler kod parcacigi
2
3 void olcumlerTask(void *pvParameters)
4 {
5     static const char *olcumlerTask = "olcumlerTask";
6     esp_log_level_set(olcumlerTask, ESP_LOG_INFO);
7
8     while (1)
9     {
10         updateTemp(dht.readTemperature());
11         updateHumidity(dht.readHumidity());
12
13         # ESP32 Olcumler kod parcacigi (devam)
14
15         voltage = pzem.voltage();
16         current = pzem.current();
17         power = pzem.power();
18         energy = pzem.energy();
19         frequency = pzem.frequency();
20         pf = pzem.pf();
21         vTaskDelay(2000 / portTICK_PERIOD_MS);
22     }
23 }

```

```

1 # ESP32 UART kod parcacigi
2
3 void LCDseriTask(void *pvParameters)
4 {
5     static const char * LCDuartTask = " LCDuartTask ";
6     esp_log_level_set(LCDuartTask, ESP_LOG_INFO);
7
8     while (1){
9         unsigned long currentMillis = millis();
10         if (currentMillis - previousMillis3sn >= 3000){
11             Serial.print("temp.txt=\""+String(temp)+"\""+endChar);

```

```

12     Serial.print("nem.txt=\""+String(nem)+"\""+endChar);
13     Serial.print("set.txt=\""+String(set)+"\""+endChar);
14     previousMillis3sn = currentMillis;
15 }
16 vTaskDelay(2000 / portTICK_PERIOD_MS);
17 }
18 }

```

```

1 # Nextion parametre ayarlari kod parcacigi
2
3 counter.val=counter.val+1
4
5 if(counter.val<=1){
6     prints "LCD:SET",0
7     prints ":",0
8     prints setDeger.txt,0
9     prints "?",0
10 }
11 else if(counter.val<=2){
12     prints "LCD:HIS",0
13     prints ":",0
14     prints hisDeger.txt,0
15     prints "?",0
16 }
17 else if(counter.val<=3){
18     prints "LCD:ALT",0
19     prints ":",0
20     prints altDeger.txt,0
21     prints "?",0
22 }
23 else if(counter.val<=4){
24     prints "LCD:UST",0
25     prints ":",0
26     prints ustDeger.txt,0
27     prints "?",0
28 }
29 else if(counter.val>5{
30     counter.val=0
31 }

```

```

1 # Nextion Tarih/Saat ayarlari kod parcacigi
2
3 counter.val=counter.val+1
4 if(counter.val<=1){
5     covx t5.txt,va0.val,2,0
6     rtc2=va0.val
7 }else if(counter.val<=2){
8     covx t6.txt,va1.val,2,0
9     rtc1=va1.val
10 }else if(counter.val<=3){
11     covx t7.txt,va2.val,4,0
12     rtc0=va2.val
13 }else if(counter.val<=4){
14     covx t8.txt,va3.val,2,0
15     rtc3=va3.val
16 }else if(counter.val<=5){
17     covx t9.txt,va4.val,2,0
18     rtc4=va4.val
19 }else if(counter.val>5){
20     counter.val=0
21 }

```

```

1 # API kullanici modeli kod parcacigi
2
3 using System.Diagnostics.CodeAnalysis;
4
5 namespace MeasRefri.API.Models.Domain
6 {
7     public class User {
8         public int Id { get; set; }
9         public string Email { get; set; }
10        public string Password { get; set; }
11    }
12 }

```

```

1 # API kullanici giris kod parcacigi
2
3 [HttpPost]
4 [Route("login")]
5 public async Task<IActionResult> LoginAsync(Models.DTO.

```

```

        LoginRequest loginRequest){
6
    var user = await userRepository.AuthenticateAsync (
    loginRequest.Email, loginRequest.Password);
8
    if (user != null)
9
    {
10
        var token = await tokenHandler.CreateTokenAsync(user);
11
        return Ok(new { token= token});
12
    }
13
    return BadRequest(new { MessageProcessingHandler =
14
        "Kullanici adi veya sifresi yanlis."});
15
16
17 }

```

```

1
2 # API olcum istek kod parcacigi
3
4 [HttpGet]
5 [Authorize]
6 [Route("{id:int}")]
7 [ActionName("GetByDateAsync")]
8 public async Task<IActionResult> GetByDateAsync(int id,
    DateTime PeriodStart, DateTime PeriodEnd)
9 {
10
    var measDomain = await measRepository.GetByDateTimeAsync(id
    , PeriodStart, PeriodEnd);
11
    if (measDomain == null) return NotFound();
12
    var measDTO = mapper.Map<List<Models.DTO.Meas>>(measDomain)
13
    ;
14
    return Ok(measDTO);
15
16
17 }

```

```

1
2 # API GetByDateTimeAsync kod parcacigi
3
4 public async Task<IEnumerable<Meas>> GetByDateTimeAsync(int id,

```

```

        DateTime PeriodStart, DateTime PeriodEnd)
5 {
6     return await measRefriDbContext.Meas.FromSqlInterpolated($"
        SELECT * FROM Meas WHERE RefriId = {id} AND DateTime >= {
        PeriodStart} AND DateTime <= {PeriodEnd} ")
7     .ToListAsync();
8 }

```

```

1 # Akilli tarama kod parcacigi
2
3 ValueListenableBuilder<bool>(
4 valueListenable: _isScanMode,
5 builder: (context, isScanMode, _) {
6     return Row(
7         children: [
8             Switch(
9                 value: isScanMode,
10                onChanged: (val) async {
11                    _isScanMode.value = val;
12                    if (!val) {
13                        await FlutterBlue.instance.stopScan();
14                    } else {
15                        await FlutterBlue.instance.startScan();
16                    }
17                },
18            ),
19            Text('Akilli Tarama ${isScanMode ? 'Acik' : 'Kapali'}'),
20        ],
21    );
22 })),

```

```

1 # Bluetooth Tarama kod parcacigi
2
3 Future<void> startScan() async {
4     await FlutterBlue.instance.startScan();
5 }
6
7 Future<void> stopScan() async {
8     await FlutterBlue.instance.stopScan();
9 }

```

```
1 # WiFi tarama kod parcacigi
2
3 if ((wifiClient.connected()) && (WiFi.status()==WL_CONNECTED))
4 MQTTclient.loop();
5 else{
6     if (!(WiFi.status() == WL_CONNECTED) )
7     {
8         connectToWiFi();
9     }
10    vTaskDelay(250);
11    connectToMQTT();
12 }
```