

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**TRAFO MERKEZLERİ İÇİN NESNELERİN
İNTERNETİ TABANLI KONTROL SİSTEMİ
TASARIMI VE UYGULAMASI**

YÜKSEK LİSANS TEZİ

Sercan AYGÜN

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM
MÜHENDİSLİĞİ**
Tez Danışmanı : Dr. Öğretim Üyesi Abdullah SEVİN

Ekim 2022

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**TRAFO MERKEZLERİ İÇİN NESNELERİN
İNTERNETİ TABANLI KONTROL SİSTEMİ
TASARIMI VE UYGULAMASI**

YÜKSEK LİSANS TEZİ

Sercan AYGÜN

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM
MÜHENDİSLİĞİ**

Bu tez ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Sercan AYGÜN

05.10.2022

TEŐEKKÜR

Yüksek lisans eğitiminin boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Dr. Öğretim Üyesi Abdullah SEVİN'e teşekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	v
TABLolar LİSTESİ	vii
ÖZET	viii
SUMMARY	ix
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
KAYNAK ARAŞTIRMASI	3
2.1. Nİ (Nesnelerin İnterneti-Internet of Things).....	3
2.2. GZİS (Gerçek Zamanlı İşletim Sistemi)	5
2.3 Tez Uygulaması İle İlgili Çalışmalar	7
BÖLÜM 3.	
MATERYAL VE YÖNTEM	9
3.1. Genel Bilgi	9
3.2. Yöntem	9
3.2.1. Sistem bileşenleri	9
3.2.2. Prototip tasarımı	15
3.2.3. ESP32 yazılım mimarisi	17
3.2.4. Grafiksel arayüz programı yazılım mimarisi	22

BÖLÜM 4.

ARAŞTIRMA BULGULARI	28
4.1. Uygulamanın Test Edilmesi	28

BÖLÜM 5.

TARTIŞMA VE SONUÇ	40
-------------------------	----

KAYNAKLAR	42
-----------------	----

ÖZGEÇMİŞ	46
----------------	----

SİMGELER VE KISALTMALAR LİSTESİ

ADC	: Analog to Digital Convertor
AWS	: Amazon Web Services
CAN	: Controller Area Network
CoAP	: Constraint Application Protocol
DAC	: Digital to Analog Convertor
GPL	: General Public License
GZİS	: Gerçek Zamanlı İşletim Sistemi
I2C	: Inter-Integrated Circuit
I2S	: Integrated Inter-IC Sound
KDE	: K Desktop Environment
LGPL	: Lesser General Public License
MIT	: Massachusetts Institute of Technology
MİB	: Merkezi İşlem Birimi
MQTT	: Message Queuing Telemetry Transport
Nİ	: Nesnelerin İnterneti
NTP	: Network Time Protocol
PWM	: Pulse Width Modulation
RFID	: Radio Frequency Identification
RTOS	: Real-time Operating System
SCADA	: Supervisory Control and Data Acquisition
SoC	: System on Chip
SPI	: Serial Peripheral Interface
UART	: Universal Asynchronous Receiver/Transmitter

ŞEKİLLER LİSTESİ

Şekil 2.1. Beş katmanlı Nİ mimarisi	4
Şekil 2.2. MİB üzerinde görevlerin çizelgelenmesi	6
Şekil 3.1. Röle modülünde bir kanalın elektrik devre şeması	11
Şekil 3.2. Röle modülü	11
Şekil 3.3. Gerilim ölçümü için kullanılan devre	12
Şekil 3.4. Akım ölçümü için kullanılan devre	13
Şekil 3.5. Batarya gerilimini ölçmek için kullanılan devre	13
Şekil 3.6. Sıcaklık sensörü bağlantısı ve besleme devresi	14
Şekil 3.7. ESP32 pin bağlantısı	16
Şekil 3.8. Geliştirilen prototipe ait ekran görüntüsü	16
Şekil 3.9. MQTT protokolü diyagramı	17
Şekil 3.10. ESP32 yazılımına ait ana akış şeması	19
Şekil 3.11. NTP protokolündeki cihazlara ait gösterim	20
Şekil 3.12. NTP ile zaman bilgisinin alınmasına ait kod	20
Şekil 3.13. Kesici ve ayırıcı kontrol yazılımına ait akış şeması	23
Şekil 3.14. Arayüz programındaki paneller	26
Şekil 3.15. Arayüz programının akış şeması	27
Şekil 4.1. Çeşitli konulardaki kayıtlara ait ekran görüntüsü	30
Şekil 4.2. Çeşitli konulardaki JSON formatındaki kayıtlar	31
Şekil 4.3. Arayüz programında ayırıcının güvenli çalışması	31
Şekil 4.4. Arayüz programında kesicinin güvenli çalışması	32
Şekil 4.5. Düşük batarya gerilimi uyarısına ait ekran görüntüsü	33
Şekil 4.6. Yüksek akım hatasının kullanıcıya bildirilmesine ait ekran görüntüsü	33
Şekil 4.7. Kesicinin yüksek akım hatası ile açtırılmasına ait görüntüler	34
Şekil 4.8. Sıcaklığın artması ile birlikte fanın çalışmasına ait görüntü	35
Şekil 4.9. Sıcaklığın artması ile birlikte kesicinin açtırılmasına ait görüntü	35
Şekil 4.10. Arduino IDE seri ekranında görüntülenen bilgiler	36

Şekil 4.11. Alınan hataların bir metin dosyasına kaydedilmesine ait görüntü	37
Şekil 4.12. Akım sensörüne ESP32 kitinin verdiği tepki süresine ait görüntü	38
Şekil 4.13. Gerilim sensörüne ESP32 kitinin verdiği tepki süresine ait görüntü	38
Şekil 4.14. Arayüz programında kesici ve ayırıcı kontrolündeki zaman kaybı	39

TABLolar LİSTESİ

Tablo 4.1. Güneşin doğuş ve batış zamanını için hesap sonuçları	25
Tablo 4.2. Akım sensöründen elde edilen değerler	39

ÖZET

Anahtar kelimeler: Nesnelerin İnterneti, Gerçek Zamanlı İşletim Sistemi, Trafo, İzleme

Elektrik enerjisinin ev ve işyerleri gibi alanlar için uygun gerilim seviyesine dönüştürülmesi trafo merkezleri tarafından sağlanmaktadır. Trafo merkezleri ile önemli bir miktarda enerji iletimi yapılmaktadır. Günümüzde neredeyse tüm kamu ve özel kurum ve kuruluşlara ait kritik altyapılar bu elektrik iletimine bağlı olarak çalışmaktadır. Hastaneler, sanayi kuruluşları, nükleer ve diğer santraller, bankalar, askeri birimler vs. gibi birçok alandaki bu kritik altyapıda oluşabilecek sıkıntılar hayati veya maddi problemlere yol açacaktır. Bu nedenle bu merkezlerde meydana gelen bir arızanın hızlı bir şekilde tespit edilmesi, ilgili kişilere bildirilmesi ve arıza türüne göre doğru müdahalenin yapılması hayati ve maddi kayıpların önüne geçecektir. Trafo merkezi kontrolünde uzun süre kullanılan SCADA sistemleri güvenli bir seçenek olmasının yanı sıra pahalı bir çözüm sunmaktadır. Nesnelerin interneti teknolojisinin gelişmesi ile birlikte bu soruna etkin ve daha az maliyetli çözümler üretilmeye başlanmıştır.

Bu çalışmada trafo merkezlerinin uzaktan izlenmesini ve kontrol edilmesini sağlamak amacıyla nesnelerin interneti ve gerçek zamanlı işletim sistemi teknolojilerini kullanan alternatif bir çözüm önerisi sunulmaktadır. Önerilen sistem, ESP32 geliştirme kiti ile geliştirilen arayüz programı üzerinden izleme ve kontrol yapılmasını sağlamaktadır. Tasarlanan sistem sayesinde yüksek gerilim durumlarına, yüksek sıcaklık ve aşırı yük veya akım gibi problemlere gerçek zamanlı müdahale edilebilmektedir. Daha önce kullanılan sistemlerde bulunan özelliklerin yanı sıra gerçek zamanlı işletim sisteminin kullanılması, öncelik mekanizması, redresör grubu takibi, Bluetooth gibi farklı haberleşme teknolojilerin eklenmesi ve özellikle maliyet etkin bir çözüm önerisi olması sistemin öne çıkan özellikleri olarak sunulmaktadır. Bunun yanı sıra önerilen sistem, nesnelerin interneti teknolojisinin uygulama alanlarından biri olan akıllı şehirlerde kullanılan akıllı lambalar ile entegre bir senaryo ile de kullanılabilir.

DESIGN AND IMPLEMENTATION OF THE INTERNET OF THINGS BASED CONTROL SYSTEM FOR TRANSFORMER CENTERS

SUMMARY

Keywords: Internet of Things, Real-Time Operating System, Substation, Monitoring

The substations convert electrical energy to the appropriate voltage level for homes and workplaces. A significant amount of energy is transmitted through these substations. Today, critical infrastructures of almost all public and private institutions and organizations operate depending on this electricity transmission. Problems in this crucial infrastructure in many areas such as hospitals, industrial establishments, nuclear and other power plants, banks, military, etc., will lead to vital or financial problems. For this reason, rapid detection of a malfunction in these centers, notifying the relevant persons, and making the proper intervention according to the type of malfunction will prevent vital or financial losses. SCADA systems have been used for a long time in substation control, offer an expensive solution, and be a safe option. Effective and less costly solutions have begun to be produced with the development of the internet of things.

This study presents a solution proposal that uses the internet of things and real-time operating system technologies to enable remote monitoring and control of substations. The proposed system monitors and controls the developed GUI application with the ESP32 development kit. The designed system can be intervened in real-time problems such as high voltage situations, high temperature, and overload or current. In addition to the features found in previously used systems, the addition of different communication technologies such as Bluetooth, a real-time operating system, priority mechanism, battery monitoring, and especially a cost-effective solution offer is presented as the outstanding features of the system. In addition, the proposed system can be used with an integrated scenario with smart lamps used in smart cities, which is one of the application areas of the internet of things technology.

BÖLÜM 1. GİRİŞ

Elektrik enerjisi, çeşitli enerji kaynaklarından dönüştürülerek uzun mesafeler katettikten sonra ev ve iş yerlerine gelmektedir. Üretilen enerji yüksek gerilim seviyelerindedir ve bu enerji çeşitli aşamalardan geçerek evlerde kullandığımız gerilim seviyesine dönüştürülmektedir. Trafo merkezleri, bu gerilim dönüştürme işleminin yanı sıra günlük yaşantımızda önemli bir yer tutan elektrik enerjisinin sağlıklı ve verimli dağıtımını ve iletiminden sorumludur. Trafo merkezlerinin sağlıklı işlemesi hayati ve maddi kayıpların önlenmesi için büyük bir önem arz etmektedir.

Trafo merkezlerinin uzun süre kullanılabilmesi için çeşitli etkenlere karşı takibinin yapılması amacıyla SCADA sistemleri kullanılmaktadır [1]–[3]. Bu sistemle veriler internet üzerinden merkezi bir yerde toplanmaktadır. Ancak; bu sistem pahalı olması sebebiyle her trafo merkezinde kullanılamamaktadır [4]. Herhangi bir uzaktan izleme yöntemi kullanılmayan diğer trafo merkezlerinde kontrol, saha ekipleri tarafından yapılmakta ve arızalara geç müdahale edilmektedir. Bu durum ciddi maddi kayıplara ve güvenlik açıklarına yol açmaktadır. Bu sebeple, gelişen teknoloji ile birlikte trafo merkezlerinin uzak kontrolü önemli bir hal almıştır.

Gömülü sistemler, günümüzde yaygın olarak kullanılan ve sadece belirli bir amacı gerçekleştirmesi için üretilen; kısıtlı bir işlemci gücü ve kaynağa (bellek, program hafızası vb.) sahip olan sistemler için kullanılan bir kavramdır. Bu sistemlerde gerçekleştirilmesi gereken işlemler için belirlenen süre hayati öneme sahiptir ve bu zaman kısıtlamasına tamamen uyulması oldukça önemlidir. Örneğin; bir kavşakta bulunan trafik ışıklarında meydana gelebilecek bir zaman sapması, farklı yollar için kısa süre de olsa aynı anda yeşil ışık yanması gibi, trafik kazasına sebep olabilmektedir. Bu sebeple bu tür sistemlerin gerçek zamanlı sistemler ile gerçekleştirilmesi gerekmektedir.

Gerçek zamanlı sistemlerde kullanılan işlemci genellikle tek çekirdekli olduğundan, görevler işlemci üzerinde sıra ile ve belirli süre kısıtlamasına tabi tutularak gerçekleştirilmektedir. Bu planlama, herhangi bir işletim sistemi kullanılmadan yapılabileceği gibi, bu amaç için tasarlanan Gerçek Zamanlı İşletim Sistemleri (GZİS) ile de yapılabilmektedir. GZİS ile görevler arasındaki geçiş belirli zaman aralıklarına bölünmekte ve çizelgeleme için ayrılan zaman azalmaktadır. Kullanılan GZİS'ne bağlı olarak gerekli zaman kısıtlamalarına tam olarak uyulabilmektedir.

Önerilen sistemle trafo merkezlerinin kontrolü ve takibi Nİ teknolojisi yardımıyla uzaktan sağlanabilmektedir. Bu sayede trafo merkezlerinde meydana gelebilecek arızaların tek bir merkezden takibi ile arızaya erken müdahale imkanı sağlanabilmektedir. Böylece olası hayati ve maddi kayıpların boyutu da azalmaktadır. Ayrıca tasarlanan sistem sayesinde arıza sayısı, kesici ve ayırıcı gibi ekipmanlara ait açma ve kapama sayısı gibi istatistiksel verilerin de toplanması arayüz programı sayesinde mümkün olmaktadır. Bu sayede kesici ve ayırıcı gibi ekipmanların bakımları da planlanabilmekte, çok sayıda arıza yapan trafolar için gerekli önlemler alınabilmektedir. Önerilen sistemin ucuz olması ile sistem tüm trafo merkezlerine kurulabilmektedir. Ring sistemi oluşturan trafo merkezlerinde ise besleme aktarımının uzaktan sağlanması bu sayede mümkün olabilecektir. Önerilen sistemde belirlenen görevler arasında gerçek zamanlı çizelgeleme için GZİS kullanılmıştır. Bu sayede görevlerin kesin bir süre içerisinde gerçekleşmesi sağlanmış ve daha sonra İnsan-Makine arayüzü gibi özelliklerin eklenmesine olanak sağlamıştır.

Önerilen sistem ile benzer çalışmaların anlatıldığı 2. bölümde Literatür araştırmasına yer verildikten sonra 3. bölümde önerilen kontrol sistemi ile ilgili bilgi verilmiştir. Bu bölümde; sistemin tasarımı, kullanılan ekipmanlar ve sistem yapısı ile ilgili bilgi verilmiş; oluşturulan prototip ile sistemin uygulanabilirliği gösterilmiştir. Ayrıca, kullanılan teknolojiler (GZİS, yazılım araçları vb.) ile ilgili teorik bilgiye 2 ve 3. bölümlerde yer verilmiştir. 4. bölümde ise sistemin genel bir değerlendirmesi yapılmış; 5. bölümde elde edilen çıktılar ile birlikte sisteme daha sonra eklenebilecek özelliklerin üzerinde durulmuştur.

BÖLÜM 2. KAYNAK ARAŞTIRMASI

2.1. Nesnelerin İnterneti (Nİ)

Nİ, gerçekte birbirleri ile fiziksel bağlantısı olmayan cihazların, iletişim kurmasını sağlaması sayesinde cihazlar arasında bilgi alışverişi yapılabilmesine olanak sağlamaktadır. Bu alışveriş; ZigBee, Bluetooth, Wi-Fi teknolojileri ile MQTT, CoAP gibi haberleşme protokolleri aracılığıyla yapılmaktadır. Nİ günümüzde sağlık, ulaşım, akıllı teknolojiler (akıllı ev ve aydınlatma), enerji ve otomasyon gibi birçok alanda kullanılmaktadır.

Nİ terimi ilk olarak 1999 yılında Kevin Ashton tarafından, RFID etiketlerinin tedarik zincirindeki rolünü anlatan bir sunumda kullanılmıştır [5]. Kavram ortaya atıldıktan sonra Nİ alanına olan ilgi giderek artmış; son yıllarda internet, sensör, işlemci vb. teknolojilerdeki gelişmelerle doruk noktasına ulaşmıştır. Nİ'ne artan ilgi onu önemli bir konuma taşımaktadır. 2019'daki verilere göre dünya çapında birbiri ile bağlantılı olan 25 milyar cihaz olduğu; bu sayının 2025'e kadar üç kat artacağı tahmin edilmektedir [6]. Bununla birlikte; Nİ alanına yapılan yatırım da gün geçtikçe artmaktadır. Yapılan bir araştırmaya göre dünya çapında Nİ alanında yapılan harcamaların 1.1 trilyon doları geçtiği; 2025'e kadar bu alanda yapılan harcamaların 2.7 ile 6.2 trilyon dolar arasında olacağı tahmin edilmektedir [7].

Nİ temelde ağ, uygulama ve algılama olmak üzere üç katmandan oluşmaktadır [8], [9]. Zamanla yeni mimariler önerilmiş olsa da beş katmanlı mimari kabul görmektedir [7]. Bu beş katmanlı mimari Nesnelere, Nesne Soyutlama, Servis Yönetim, Uygulama ve İş katmanlarından oluşmakta ve Şekil 2.1.'de gösterilmektedir.



Şekil 2.1. Beş katmanlı Nİ mimarisi.

Beş katmanlı yapı şu katmanları içermektedir:

1. Nesneler: Algılama katmanı olarak da adlandırılmaktadır. Bu katmanda sensörler vasıtasıyla çevresel etmenler ile ilgili bilgi toplanmaktadır. Elde edilen veriler bir üst katman olan Nesne Soyutlama Katmanı'na gönderilmektedir. Ayrıca, büyük veri yine bu katmanda oluşturulmaktadır.
2. Nesne Soyutlama: Nesnelere vasıtasıyla elde edilen veriler güvenli kanallar ile servis yönetim katmanına iletilmektedir. Verilerin iletilmesinde daha önce belirtildiği üzere Bluetooth, Wi-Fi vb. teknolojiler kullanılmaktadır.
3. Servis Yönetim: Bu katman elde edilen verilerin yorumlanıp, gerekli servis hizmetlerinin çalıştırıldığı katmandır.
4. Uygulama: Kullanıcı ihtiyaçlarının karşılandığı katmandır. Örneğin; sisteme ait sensör verileri kullanıcıya bu katmanda görüntülenmektedir.
5. İş: Tüm katmanların izleme ve yönetim işlemleri bu katmanda yapılmaktadır. Sistem bileşenleri, Uygulama Katmanı'ndan alınan verilere göre takip edilmektedir.

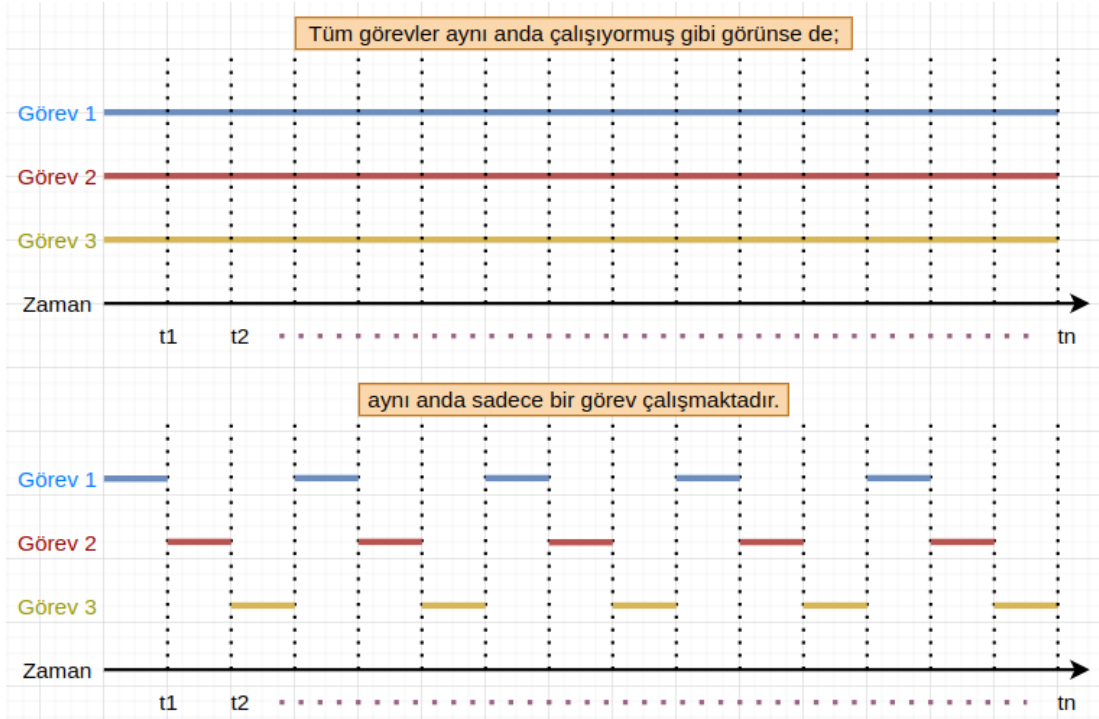
Beş katmanlı mimarinin kolay ve anlaşılır olmasının yanı sıra modüler yapısı onu en çok kullanılan Nİ mimarisi yapmıştır.

2.2. GZİS (Gerçek Zamanlı İşletim Sistemi – Real Time Operating System)

Gömülü sistemler, günümüzde yaygın olarak kullanılan ve sadece belirli bir amacı gerçekleştirmesi için üretilen; kısıtlı bir işlemci gücü ve kaynağa (bellek, program hafızası vb.) sahip olan sistemler için kullanılan bir kavramdır. Bu sistemlerde gerçekleştirilmesi gereken işlemler için belirlenen süre hayati öneme sahiptir ve bu zaman kısıtlamasına tamamen uyulması oldukça önemlidir. Örneğin; bir kavşakta bulunan trafik ışıklarında meydana gelebilecek bir zaman sapması, farklı yollar için kısa süre de olsa aynı anda yeşil ışık yanması gibi, trafik kazasına sebep olabilmektedir. Bu sebeple bu tür sistemlerin gerçek zaman kısıtlarına büyük bir doğrulukla uyması gerekmektedir.

Gerçek zamanlı sistemlerde MİB genellikle tek çekirdekli olduğundan, görevler işlemci üzerinde sıra ile ve belirli süre kısıtlamasına tabi tutularak gerçekleştirilmektedir. Bu planlama, herhangi bir işletim sistemi kullanılmadan yapılabileceği gibi, bu amaç için tasarlanan Gerçek Zamanlı İşletim Sistemleri (GZİS) ile de yapılabilmektedir. Böylece uygulamanın yapabileceği işlem sayısı arttırılabilirken, görevlerin çizelgelenmesi sayesinde her zaman en yüksek önceliğe sahip görevin yürütülmesi garanti altına alınmaktadır. Bu durum ise gerçek zaman kısıtlarına uyulmasını kolaylaştırmaktadır. Bu sayede yazılım geliştirmek için gerekli zaman azaltılmakta; yazılımcı tarafından oluşturulması halinde güvenliği tartışılır bir çizelgeleme yöntemi yerine daha güvenli bir yapı kullanılmış olmaktadır. Ayrıca; takım içindeki kişiler için ortak bir yazılım çerçevesi olması sebebiyle GZİS, birden fazla yazılımcının aynı kod parçası üzerinde çalışmasını kolaylaştırmaktadır.

Gömülü sistemlerde gerçekleştirilen işlemler alt görevlere bölünmektedir. Bu görevler kullanıcıya aynı anda gerçekleşiyormuş hissi vermektedir. Aslında; MİB üzerinde aynı anda sadece bir görev çalıştırılabilmektedir. Bu durum Şekil 2.2.'de görülmektedir. Görevlerin hangi sırada ve zamanda çalıştırılacağı çizelgeleme kavramı ile ifade edilmektedir. Çizelgeleme, birden fazla görevden hangisinin MİB'de çalıştırılacağını belirlemektedir. Çizelgeleme, temelde iki alt gruba ayrılmaktadır: 1) Kesintili (preemptive) ve 2) Kesintisiz (non-preemptive).



Şekil 2.2. Bir MİB'nde çalışan üç görev, aslında ayrı zamanlarda yürütülmektedir.

Kesintili çizelgeleme yönteminde yüksek öncelikli bir görev hazır kuyruğunda olduğunda, çalıştırılmakta olan görev hazır kuyruğuna alınarak yüksek öncelikli görev çalıştırılmaktadır. Dolayısıyla, GZİS'lerinde Kesintili türde çizelgeleme yöntemleri kullanılmaktadır. Böylece her zaman en yüksek önceliğe sahip görevin çalıştırıldığından emin olunmaktadır.

Önerilen sistemin geliştirilmesinde kullanılan GZİS, MİB'nin görevler arasındaki paylaşılma zamanını, kullanıcı tarafından belirlenen FreeRTOSConfig.h dosyasında configTICK_RATE_HZ parametresi ile belirlenmektedir [10]. Bu parametre ile girilen değer FreeRTOS'un çizelgelemesinde kullanılan zaman aralığını Hertz cinsinden vermektedir ve genellikle 1000 Hz olarak seçilmektedir. Böylece MİB'nin görevler arasındaki kullanılma zamanı 1 ms olarak belirlenmektedir.

GZİS kullanmanın birçok avantajı bulunmaktadır. Öncelikle, oluşturulan görevlerin çizelgelemesi kullanıcıdan soyutlanmış olur. Bu durum, kullanılan işlemcinin farklı saat frekanslarında çalışmasının önem arz etmemesinin ve farklı saat frekansına sahip işlemcilerin kullanılabilmesini sağlamasıyla uygulamaya büyük bir esneklik kazandırmaktadır. Çoğunlukla C dili ile yazılması sebebiyle farklı işlemci

mimarilerinde ve derleyicilerde kolaylıkla kullanılabilirken; modüler ve tekrar kullanılabilir kod yazmayı sağlamaktadır. Yoklama (polling) metodunun kullanıldığı ve görevlerin sürekli çalıştırıldığı durumun aksine işlemci sürekli çalıştırılmadan, görevlerin icra edilmediği zamanlarda işlemcinin uyku moduna girmesi sağlanarak güç tasarrufu sağlanabilmektedir. Karşılıklı dışlama ve semafor gibi mekanizmalar sayesinde yarış durumunun meydana gelmesi engellenerek, veri kaybının veya verinin yanlış değerlendirilmesinin önüne geçilmektedir.

2.3. Tez Uygulaması İle İlgili Çalışmalar

Son yıllarda, gelişen teknolojiler ile birlikte Nİ teknolojisi çok fazla alanda kullanılmakta ve ilgi de giderek artmaktadır. Gelişen internet teknolojisi ile birlikte kullanım alanlarının ve ucuz teknolojilerin ortaya çıkması neticesinde yaygınlığının ve bulunabilirliğinin artması sayesinde Nİ'nin trafo merkezlerinde kullanılması da gündeme gelmiştir. Bu amaçla önerilen sistemlerde [11]–[16] trafo merkezlerinin uzaktan kontrolü belli ölçülerde sağlanabilirken, çeşitli sensörlerden (gerilim, akım vb.) gelen veriler takip edilebilmektedir. Bununla birlikte, meydana gelen arızaların e-posta veya SMS olarak iletilebilmesini sağlayan çalışmalar da mevcuttur [17]–[20]. Bunun yanı sıra trafoda meydana gelen hataların hücresel ağ teknolojisi ile iletilebilmesini öneren çalışmalar da yapılmıştır [21]–[23]. Ayrıca, hastane gibi çok önemli yerlere elektrik verilmesinde öncelik tanıyan bir öneri de yapılmıştır [24]. Bu öneriye göre trafo merkezindeki arıza giderildikten sonra elektrik enerjisinin dağıtımı öncelik sırasına göre yapılmaktadır. Böylece; hastane gibi yerlere elektiriğin tekrar verilmesinde öncelik tanınabilmektedir. Bir başka çalışmaya göre [25]; trafo merkezlerinde toplanan veriler, uzak bilgisayarlardan izlenebileceği gibi, bu verilerin Android uygulaması ile telefonda da takip edilebildiği sistemler geliştirilmiştir. Ek olarak, trafo merkezlerinde bulunan kesici gibi bazı ekipmanların uzaktan kontrolünü yapan sistemler de önerilmiştir [26].

Trafoların kontrol ve takibinin yapılmasında, yapay sinir ağları ve bulanık mantık gibi yapay zeka yöntemleri kullanan bazı uygulamalar geliştirilmiştir [27], [28]. Trafo merkezindeki çeşitli bilgilerin FPGA vasıtasıyla toplanıp, ağ zaman protokolü ile saat

bilgisi alınarak uzak bilgisayar ile senkronize eden bir sistem çalışması da yapılmıştır [29]. Sensör değerlerinin kritik değerleri aşması neticesinde kullanıcıya bildirim gönderen bir uygulama önerisi de mevcuttur [30]. Yapılan başka bir çalışmada, sensörlerden alınan verilerin değerlendirilmesinde uç yapay zeka yöntemi kullanılmıştır [31]. Bununla birlikte, Raspberry Pi gibi ucuz ekipmanların kullanıldığı bir öneri de mevcuttur [32]. Bu sayede SCADA ve PLC gibi pahalı yöntemlere bir alternatif önerilmiştir. Trafonun durum kontrolünün termal görüntü ile yapılması bir çalışmada önerilmiştir [33]. Önerilen başka bir sistemde trafo merkezleri ile iletişim uydudan sağlanmıştır [34]. Wang ve arkadaşları ise trafo merkezlerinde bulunan SF6 gibi tehlikeli gazların takibinin yapılmasının yanı sıra kamera ile uzaktan izlemenin yapılabileceği bir sistem önermişlerdir [35]. Aynı sistemde sokak lambalarının kontrolü de sağlanmaktadır. Ayrıca; makine öğrenmesinin Nİ ile birlikte kullanıldığı sistem önerisi de bulunmaktadır [36].

Gerçekleştirilen çalışmada ise ESP32 geliştirme kiti ve geliştirilen arayüz programı üzerinden izleme ve kontrol yapılabilmektedir. Önerilen sistem ile yüksek gerilim durumlarına, yüksek sıcaklık ve aşırı yük veya akım gibi problemlere gerçek zamanlı müdahale imkanı sağlanmıştır. Daha önce kullanılan sistemlerde bulunan özelliklere ek olarak gerçek zamanlı işletim sisteminin kullanılması, öncelik mekanizması, Bluetooth gibi farklı haberleşme teknolojilerinin eklenmesi, mobil haberleşme ve özellikle maliyeti emsallerine göre düşük bir çözüm önerisi olması sistemin öne çıkan özellikleridir. Önerilen sistemin nesnelere internetin uygulama alanlarından biri olan akıllı şehirlerde kullanılan akıllı lambalar ile entegre bir senaryo ile birlikte kullanılması da mümkündür.

BÖLÜM 3. MATERYAL VE YÖNTEM

3.1. Genel Bilgi

Önerilen sistem için ESP DEVKIT V1 modülü ile breadboard üzerinde bir uygulama gerçekleştirilmiştir. Bu uygulamada, oluşturulan görevlerin çizelgelemesi amacıyla açık kaynaklı GZİS'lerden biri olan FreeRTOS kullanılmıştır. Hızlı prototipleme amacıyla Arduino IDE üzerinde geliştirilme yapılmıştır. Modül ile WiFi bağlantısı üzerinden MQTT protokolü ile kurulan uzak bağlantı neticesinde sistemin durumu Qt kütüphanesi ile gerçekleştirilen bir masaüstü uygulamasında görüntülenmiştir. MQTT protokolü ile uzak bağlantının kurulabilmesi için Adafruit MQTT Broker kullanılmıştır.

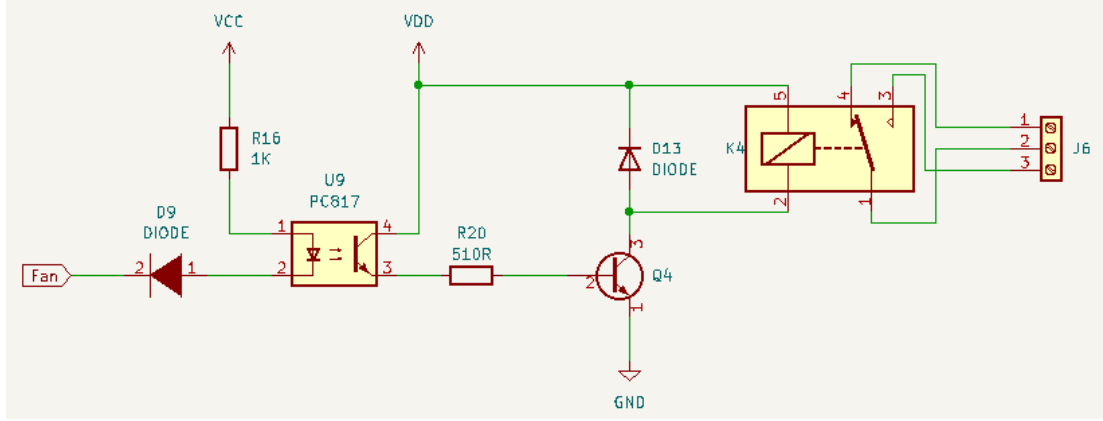
Trafo merkezi için geliştirilen prototip düşük güçlü (300 VA) kuru tip trafo, kesici ve ayırıcı yerine kullanılan röle ve bu röleleri sürmek için kullanılan röle modülü, trafo sargılarının sıcaklığını ölçmek amacıyla DS18B20 sıcaklık sensörü, sıcaklık sensöründen elde edilen veriye göre kontrol edilen fan, trafonun primer tarafındaki gerilimi takip etmek için ikincil bir trafo ve trafo sekonderinden akan akımın takip edilmesi amacıyla ACS712 akım sensöründen oluşmaktadır. Trafo merkezinde bulunan redresör grubu yerine sistemin yedek beslemesinde kullanılmak amacıyla 12V akü; önerilen sistemin beslemesi amacıyla güç kaynağı ve sistem çıktılarını gözlemlemek amacıyla multimetre ve lojik analizör kullanılmıştır.

3.2. Yöntem

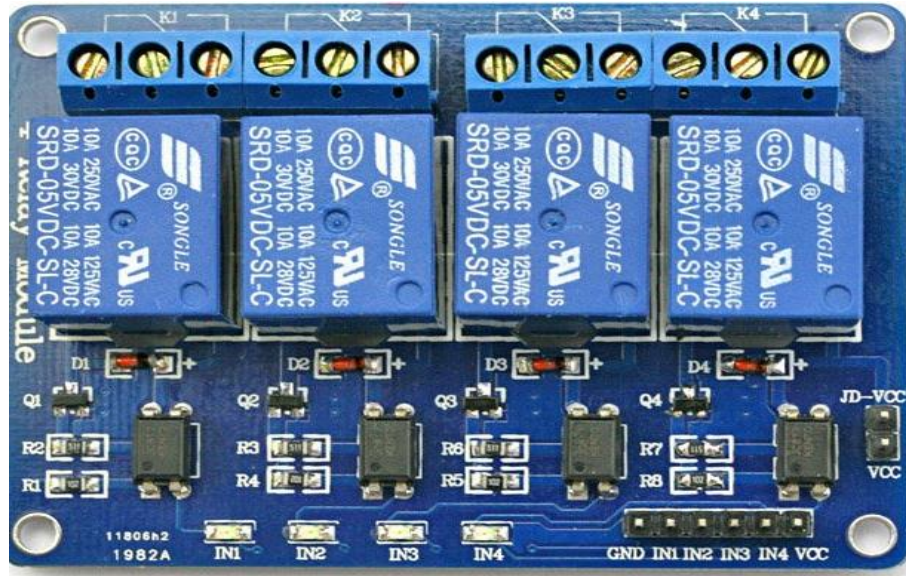
3.2.1. Sistem bileşenleri

Tasarımı yapılan uygulamanın gerçekleştirilmesi amacıyla aşağıdaki bileşenler kullanılmıştır:

1. ESP Devkit V1: Büyük bir hızla gelişen Nİ teknolojisi tümleşik devrelere olan ihtiyacı da beraberinde getirmiştir. Artan işlem gücü, çevre birimleri ve dahili program belleği gibi ihtiyaçların karşılanması talebi beraberinde yarıiletken üreticileri tarafından hızla bu talebin karşılanmasına da sebep olmuştur. Günümüzde Nİ tabanlı uygulama geliştirme amacıyla tek çip üzerinde sistem üretimi yapan çok sayıda üretici bulunmaktadır [37]. Bu üreticilerden biri olan Espressif firması Nİ alanında kullanılmak üzere farklı çözümler önermektedir. Bu çözümlerden biri olan Devkit V1 geliştirme kiti, önerilen sistemde kullanılmıştır. Kitin bazı özellikleri şu şekildedir:
 - a. CPU: 2 çekirdekli işlemci birimi, 32 bit mimari
 - b. WiFi: 2.4 GHz bandında 150 Mbits/s'a kadar
 - c. BLE (Bluetooth Low Energy)
 - d. İşlemci saat frekansı: 240 MHz'e kadar
 - e. RAM: 512 KB
 - f. Çevresel birimler: ADC, DAC, I2C, UART, CAN 2.0, SPI, I2S, PWM.
2. Trafo: Prototip için kullanılan trafo 230VAC/12-15VAC dönüşüm oranına sahip 300VA gücündedir. Trafonun girişi şebekeye bağlanarak, geliştirilen arayüz ile kontrol edilen kesici ve ayırıcı ile trafo kontrol edilmektedir.
3. Röle Modülü: Prototipte kesici ve ayırıcı yerine röle kullanılmıştır. Ayrıca; sokak lambası ve fanın kontrolü de röle ile sağlanmaktadır. ESP32 geliştirme kitini harmoniklerden korumak amacıyla röle kontrolü elektriksel izolasyon ile yapılmaktadır. Röle kontrolüne ait elektriksel şema Şekil 3.1.'de, röle modülü ise Şekil 3.2.'de görülmektedir. Bu şemaya göre, ilgili pin lojik-0 olduğunda bu pine bağlı optokuplör iletime geçerek röle çektilmektedir. Böylece, rölenin normalde açık kontağına bağlı çıkış aktif edilerek röle çıkışı sürülmektedir. Röle modülü kesici, ayırıcı olarak kullanılmasının yanı sıra, sokak lambası ve fan kontrolü için de kullanılmaktadır.

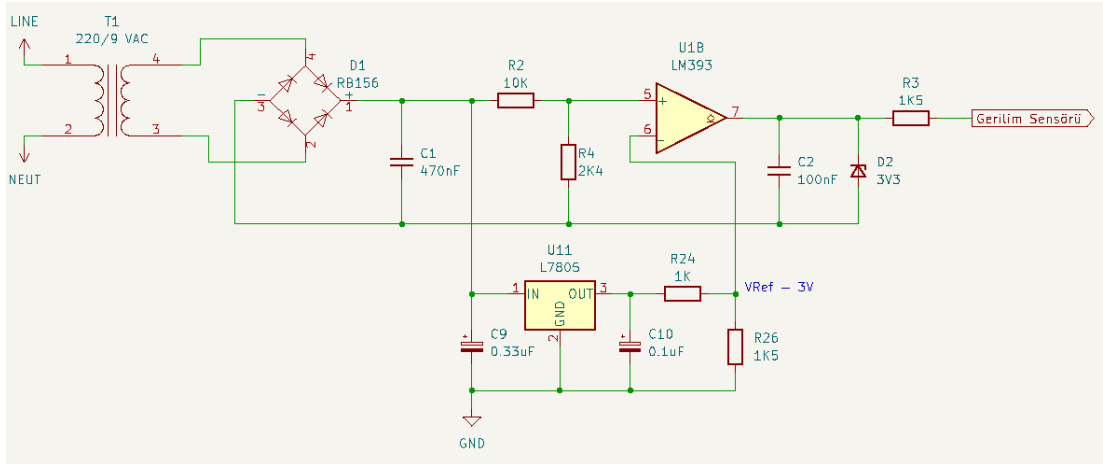


Şekil 3.1. Röle modülünde bir kanalın elektrik devre şeması.



Şekil 3.2. Röle modülü.

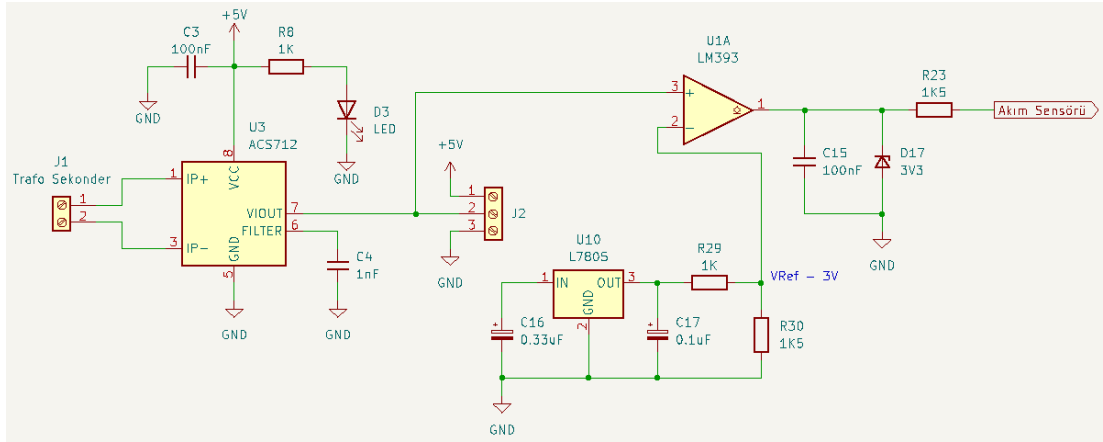
4. Gerilim Sensörü: Giriş geriliminin trafonun sargılarına zarar vermemesi için izlenmesi gerekmektedir. Bu nedenle Şekil 3.3.'teki devre kullanılmıştır. Giriş gerilimi 230/9VAC dönüşüm oranına sahip bir trafo ile düşürülmektedir. Trafo ile elektriksel izolasyon da sağlanmaktadır. Ölçü trafosunun çıkışındaki gerilimin DC gerilime dönüştürülmesi için köprü diyot kullanılmaktadır. Filtrenin ardından gerilim bölücü yardımıyla, karşılaştırıcı op-amp devresinin evirmeyen girişinde 230VAC şebeke gerilimi için 2,5VDC civarında bir gerilim bulunmaktadır. Karşılaştırıcı op-amp devresinin eviren girişine ise, gerilim regülatörü kullanılarak elde edilen 3VDC referans gerilimi bağlanmıştır. Böylece, giriş geriliminden elde edilen değer bu değer üstüne çıkmasıyla ESP32'ye lojik-1 bilgisi gönderilmektedir. Sonuçta, giriş gerilimi



Şekil 3.3. Gerilim ölçümü için kullanılan devre.

270VAC değerini aştığında kesici açtırılarak trafo korunabilmektedir.

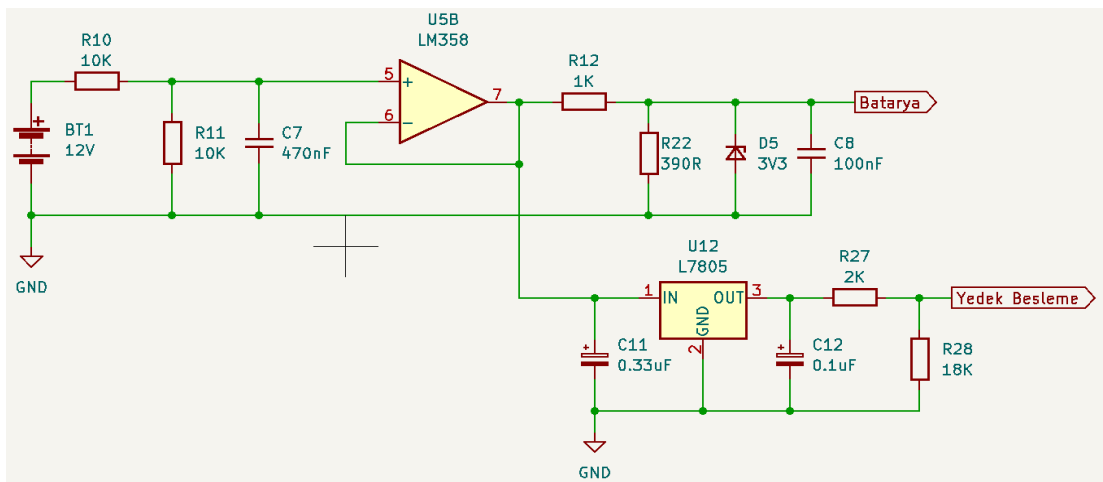
5. Akım Sensörü: Trafonun sekonder kısmına bağlanan yükün aşırı akım çekmesi durumunda trafonun zarar görmemesi için çıkış akımının takip edilmesi gerekmektedir. Bu amaçla, ACS712 akım sensörü kullanılmıştır. Sensörü içeren kit ile çıkıştaki akım izlenmektedir. ACS712 sensörü, Hall etkisi ile ölçüm yapmaktadır. Sensör sayesinde elektriksel izolasyon sağlanabilmektedir. Giriş pinlerinde akıtılan akım sensörün algılayabileceği bir manyetik alan oluşturmaktadır. Manyetik alanın akım değerine uygun gerilime dönüştürülmesiyle elde edilen değer karşılaştırıcı op-amp devresinin evirmeyen girişine bağlanmıştır. Bu değer, gerilim regülatörü ile elde edilen 3VDC gerilim değerini aşması ile karşılaştırıcı çıkışındaki lojik-1 bilgisi ESP32'ye gönderilmektedir. Sonuçta, trafo çıkış akımının 2.5A seviyesini aşması durumunda kesici açtırılarak trafo korunmaktadır. Trafo çıkış akımını ölçmek için kullanılan devre Şekil 3.4.'te görülmektedir.
6. Redresör: Trafo merkezlerinde, enerjinin olmadığı durumlarda kesici ve ayırıcıyı kontrol edebilmek için redresör grubu kullanılmaktadır. Geliştirilen prototipte redresör grubu yerine 12V akü kullanılmıştır. Akü gerilimi kritik değer altına düştüğünde kullanıcının bilgilendirilmesi amacıyla, akü geriliminin takip edilmesi gerekmektedir. Gerilimi takip etmek için kullanılan Şekil 3.5.'teki devrede, gerilim bölücülerle yarıya düşürülen akü gerilimi gerilim izleyici op-amp devresinin ile takip edilmektedir. Gerilim izleyici dev-



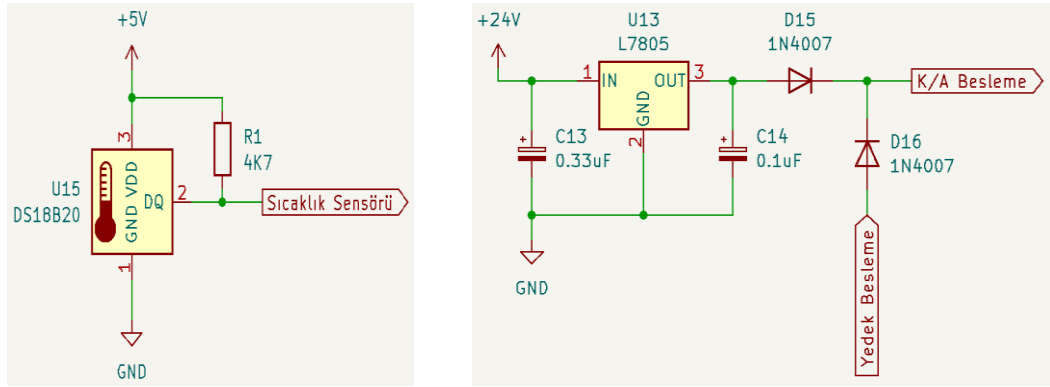
Şekil 3.4. Akım ölçümü için kullanılan devre.

resinin çıkışında kullanılan gerilim bölücü ile 12V için işlemcide 1.7V elde edilmiştir. Akü gerilimi için kritik değer 7V olarak belirlendiğinden, geliştirilen yazılımda akü gerilimi bu değer altına düştüğünde kullanıcı bilgilendirilmiştir. Gerilim izleyici devresinin çıkışındaki gerilimin 3.3V seviyesini aşmaması için 3.3V zener diyot kullanılmıştır.

7. Sıcaklık Sensörü: Trafo sargılarının sıcaklığını takip etmek amacıyla DS18B20 sıcaklık sensörü kullanılmıştır. Sensör, tek hat haberleşme ile ESP32'ye bilgi göndermektedir. Tek hat protokolünde kaydedicilerde tutulan veriler belirli sekans ile okunmaktadır. DS18B20'de sıcaklık bilgisi düşük ve yüksek anlamlı iki byte şeklinde tutulmaktadır. Ayrıca, kullanıcının belirleyebildiği yüksek ve düşük sıcaklık eşik değerlerine göre alarm çıkışı da verilebilmektedir. DS18B20'nin ESP32 ile olan bağlantı Şekil 3.6.'da görülmektedir.



Şekil 3.5. Batarya gerilimini ölçmek için kullanılan devre.



Şekil 3.6. Sıcaklık sensörü bağlantısı (sol) ve besleme devresi (sağ).

8. Fan: Sıcaklık değerinin belirli bir değerin üzerine çıkması trafonun güvenliği için engellenmelidir. Bu sebeple, sargı sıcaklığı 45 °C'yi aştığında röle çıkışına bağlı fan çalıştırılmaktadır.
9. Besleme Devresi: Sistem beslemesinin yedekli çalışabilmesi amacıyla Şekil 3.6.'daki devre kurulmuştur. Elektrik kesintisi durumlarında kesici ve ayırıcıyı kontrol etmek bu sayede mümkün olmuştur.
10. Qt Çalışma Alanı (Framework): Qt, platformlar arası grafiksel arayüz tasarımına imkan veren bir C++ kütüphanesidir. Bununla birlikte; Python, Ruby, Java, Go gibi birçok programlama dili ile bağlantıları mevcuttur [38]. Bu sebeple; geliştiriciler için programlama dilinden bağımsız geliştirmeye imkan tanımaktadır. Qt; Windows, MacOS, Linux işletim sistemlerinin yanı sıra mobil işletim sistemlerini de desteklemektedir [39]. Ayrıca, gömülü sistemler için de arayüz tasarımı yapılabilmektedir [40]. Qt; özellikle, Linux işletim sistemlerinde kullanılan K masaüstü ortamının (KDE) geliştirilmesinde kullanılmaktadır ve GPL, LGPL gibi açık kaynak lisanslarına sahiptir.

Geliştirilen uygulamada, Qt kütüphanesinin Python bağlantılarından PySide6 kullanılmıştır. PySide, LGPL lisansına sahip olması nedeniyle ticari uygulamalarda kullanılabilir ve Qt geliştiricilerinin resmi desteğine sahiptir [41]. Bir başka popüler 3. parti Qt bağlantısı olan PyQt'ye göre daha yavaş güncellenmesine rağmen; PyQt'nin aksine ticari uygulamaların geliştirilmesinde kullanılabilir.

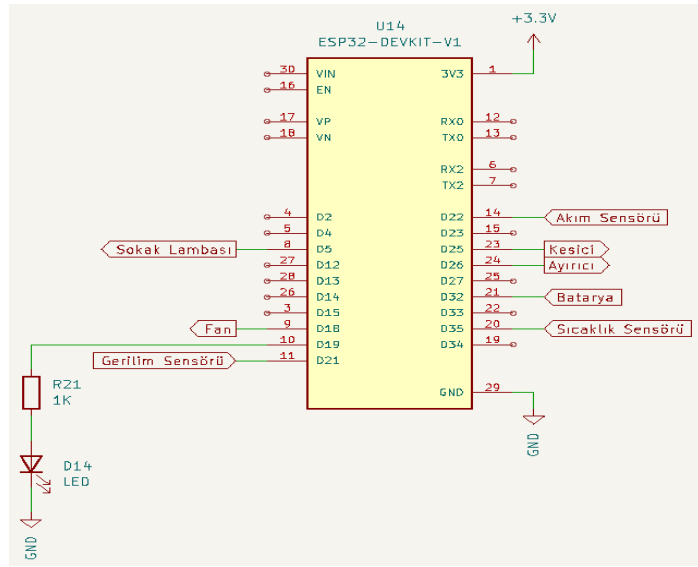
11. FreeRTOS: Richard Barry tarafından 2003 yılında geliştirilmeye başlanmasının ardından geliştiren şirketin 2017 yılında Amazon firmasına geçmesiyle birlikte AWS bünyesine katılmıştır. Çoğunlukla C dili ile yazılmasının yanında gerekli yerlerde Assembly dili kullanılmıştır. Günümüzde her üç dakikada bir kez indirilmesi sebebiyle en çok kullanılan açık kaynak kodlu GZİS'tir [42]. FreeRTOS, 35 farklı mimari ve çok sayıda derleyici ile kullanılabilmesi sayesinde geliştirilen uygulamaların işlemcilerden bağımsız geliştirilmesine imkan tanımaktadır [43]. Engelleyen ve engellemeyen türde çizelgeleme kullanması sebebiyle kullanım durumuna göre çizelgeleme türü değiştirilebilmektedir. Ayrıca, beş farklı bellek yönetim metodu kullanması sebebiyle bellek yönetiminde esneklik sağlamaktadır.

Geliştirilen uygulamada FreeRTOS, Arduino IDE'de ESP32 ile birlikte kullanılmıştır. Yapılan işlemler modüllere bölünüp her modüle öncelik atanmasıyla engelleyen türde çizelgeleme kullanılmıştır. Her bir görevin çalışma zamanı görevlerin önceliğine göre belirlenmiştir. MQTT haberleşmesinin sağlanması Adafruit MQTT kütüphanesi kullanılmıştır. Arduino kodlarının kullanılması uygulamanın farklı işlemci mimarilerinde kullanılabilmesine de olanak tanımaktadır.

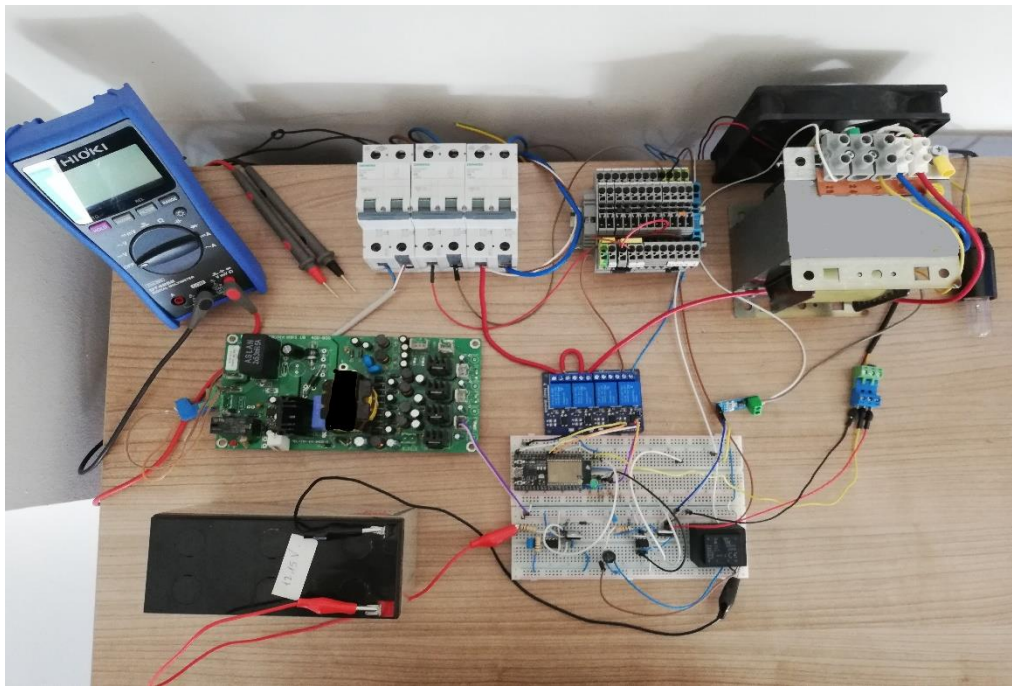
3.2.2. Prototip tasarımı

Tez çalışması için gerçekleştirilen prototipte 1 adet ESP32 geliştirme kiti breadboard üzerine yerleştirilerek WiFi üzerinden bağlantı kurulmuştur. Prototipte, DS18B20 sıcaklık sensörü ve ACS712 akım sensörü kullanılmıştır. Giriş geriliminin ölçülmesi için karşılaştırmalı op-amp devresi kurulmuştur. Batarya gerilimini ölçümü, gerilim izleyici op-amp devresi ile mümkün olmuştur. Sistemde aşırı gerilim ve aşırı akım için dijital girişler ilgili karşılaştırmalı çıkışına bağlanarak, arızalar için tahsis edilen pinlerin lojik-1 olup olmadığı kontrol edilmektedir. Sıcaklık için ayrılan pinden okunan değere göre fan, röle ile çalıştırılmaktadır. Röle sürme devresinde optokuplör kullanıldığından, fanın çalışması işlemciden izole edilmiştir. Bu nedenle ESP32 daha güvenli çalıştırılmıştır.

Kesici ve ayırıcı olarak kullanılan röleler, geliştirilen arayüz programı ile uzaktan kontrol edilmektedir. Bu rölelerin, geliştirilen grafiksel arayüz programı ile aktif ve deaktif edilebilmesi için MQTT üzerinden ESP32 kiti ile bağlantı kurulmuştur. Bu bağlantının kurulabilmesi amacıyla kit üzerindeki WiFi modülü kullanılmış, Adafruit MQTT Broker ile konular yönlendirilmiştir. Konulardaki veriler hem bulutta hem de yerel bilgisayarda, sonradan değerlendirilmek üzere, kayıt altına alınmıştır. ESP32 kitindeki giriş ve çıkış pinleri Şekil 3.7.'de, uygulamaya ait görüntü ise Şekil 3.8.'de görülmektedir.



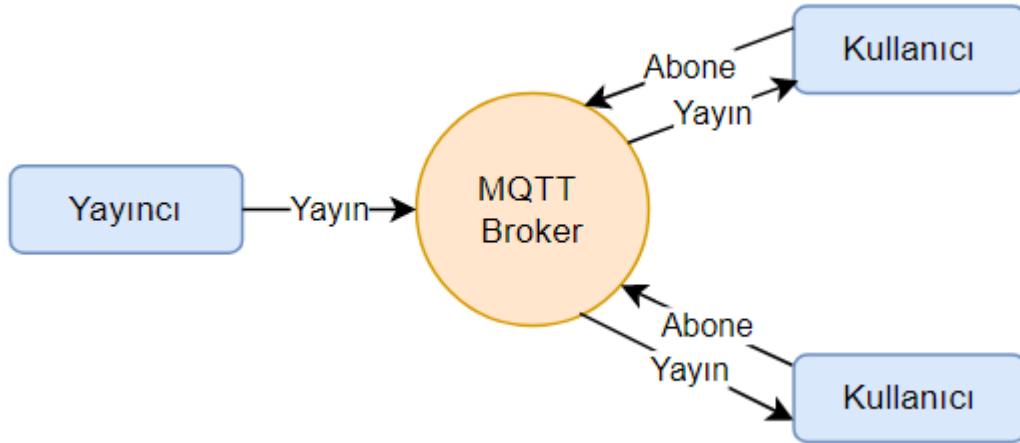
Şekil 3.7. ESP32 pin bağlantısı.



Şekil 3.8. Geliştirilen prototipe ait ekran görüntüsü.

Geliştirilen prototipin uzaktan kontrol ve takibinin yapılabilmesi amacıyla platformlararası çalışabilen C++ kütüphanesi Qt'nin Python bağlantılarından olan PySide6 kullanılmıştır. Uygulama ile kesici ve ayırıcı kontrolü uzaktan sağlanabildiği gibi, sokak lambası ve sensörlerin durumları görüntülenebilmektedir. Prototip ile haberleşme MQTT v3.1.1 protokolü üzerinden gerçekleşmektedir. MQTT protokolünde konular; abone ve yayınlama metodu kullanılarak, kullanıcı ve uç cihazlar arasında veri akışı sağlanmaktadır.

MQTT protokolünü en iyi anlatan benzetme TV yayınları ve izleyicilerdir. Bu örnekte, TV yayını belirli bir kanal üzerinden kullanıcıya yayın yapmaktadır. Kullanıcı, istediği yayını kanala abone olarak izleyebilmektedir. Bu durumda kullanıcı ile yayıncı arasında doğrudan bir bağlantı bulunmamaktadır. Protokolde ise, yayıncı bir konu üzerinde mesaj yayınlamakta; abone ise bu konuya üye olarak mesaj içeriğine ulaşabilmektedir. Bununla birlikte, protokol bir Broker gerektirmektedir. Broker, abone ve yayıncı arasındaki mesaj alışverişinin ve yönlendirmesinin yapılmasını sağlamaktadır. MQTT protokolünü anlatan diyagram Şekil 3.9.'da görülmektedir.



Şekil 3.9. MQTT protokolü diyagramı.

3.2.3. ESP32 yazılım mimarisi

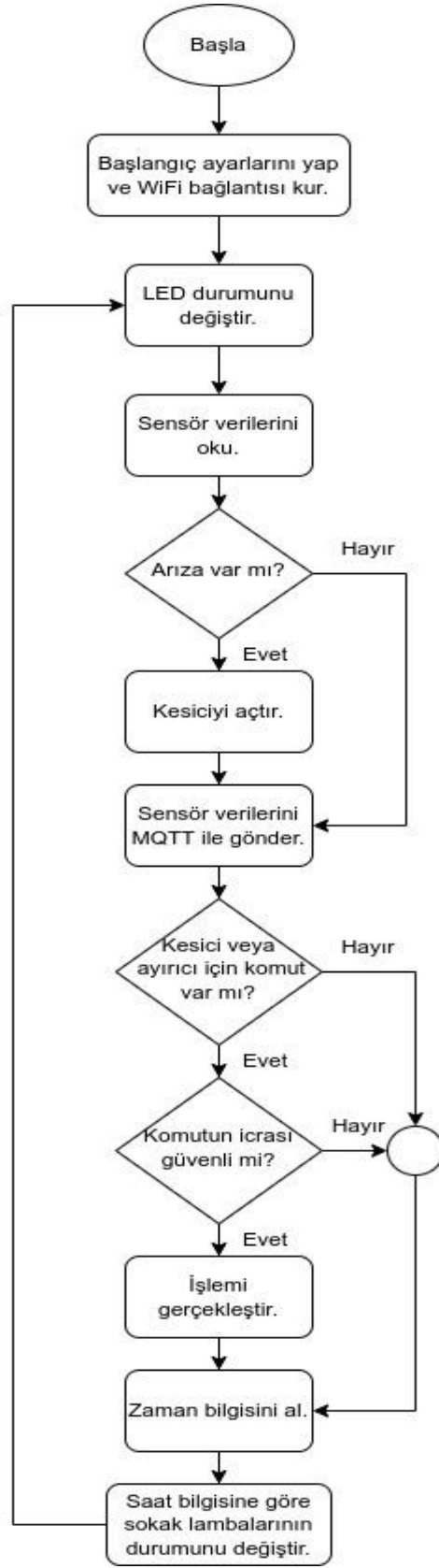
ESP32 geliştirme kiti ile uzak bağlantının sağlanması amacıyla Arduino IDE üzerinde geliştirme yapılmıştır. Bir trafo merkezi ile uzak bağlantının sağlandığı uygulamada görevler adımlara bölünerek, her bir görevin çalışma zamanı FreeRTOS ile çizelgelen-

miştir. Her bir görev öncelik durumu atanarak, yüksek öncelikli görevlerin tam zamanında işlenmesi sağlanmıştır. FreeRTOS kullanılması ile birlikte sisteme ileride eklenebilecek yeni görevlerin çizelgelemesinin daha kolay olması ve sistemin geliştirilmesinde esneklik kazanılması sağlanmıştır.

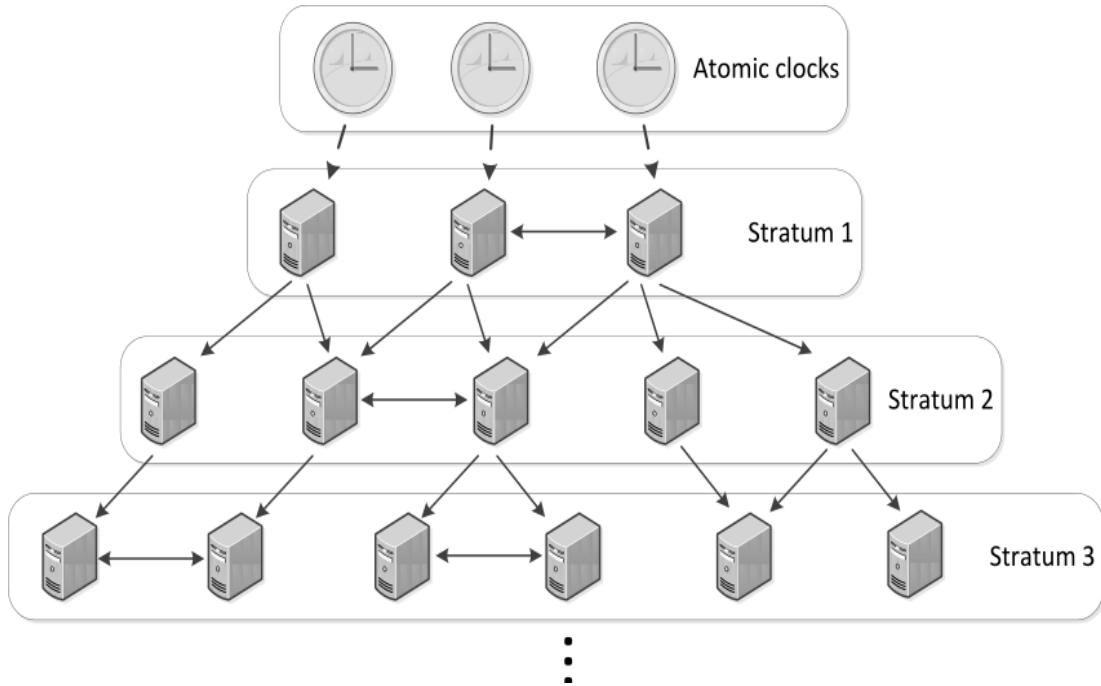
Yazılıma ait bileşenler şu şekildedir: *getTime.ino* modülünde, anlık zaman bilgisi bir saniyelik periyotlarla alınırken; bu bilginin *lighting.ino* modülünde değerlendirilerek sokak lambalarının kontrolünün sağlanması amacıyla lojik çıkış üretilmektedir. *toggleLED.ino* modülünde, WiFi bağlantısının ve sistemin sağlıklı olarak çalıştığını kullanıcıya göstermek amacıyla, tahsis edilen pine bağlı ledin 500 ms’de bir yanıp sönmesi sağlanmıştır. *mqttControl.ino* modülünde, kesici ve ayırıcının kontrolü yapılırken; *status.ino* modülünde, sensör bilgileri değerlendirilerek MQTT protokolü üzerinden yayın yapılmakta ve kesici kontrol edilmektedir. *main.ino* modülünde, sistemin ilk ayarları ve yukarıda bahsi geçen görevlerin çizelgelemesi yapılmaktadır. *config.h* dosyasında, modüllerde kullanılan tanımlamalar ve sınır değerler belirlenmektedir. Geliştirilen yazılımın ana akış şeması Şekil 3.10.’da verilmiştir.

Sokak lambasının kontrol edilmesi için yazılımda iki görev bulunmaktadır: *getTime* ve *lighting*. *getTime* modülünde saat bilgisi NTP (Network Time Protocol) ile alınmaktadır. NTP, aynı ağda bulunan bilgisayarların zamanlarının senkronize olmasını sağlamak için 1985’te geliştirilmiş bir protokoldür ve uygulama katmanında çalışmaktadır. Protokol, hiyerarşik yarı katmanlı zaman kaynaklarından oluşan bir yapı kullanmaktadır. Buna göre, bir ağ yapısındaki cihazlar strata denilen katmanlara ayrılmaktadır.

NTP katmanlarından stratum-0’da yüksek doğruluğa sahip atomik veya radyo saatleri yer alırken; stratum-1’de kendi sistem zamanları bağlı oldukları stratum-0’daki cihazlar ile birkaç milisaniye senkronize olan sunucular bulunmaktadır. stratum-2 katmanında ise stratum-1’deki bir veya birden fazla cihaz ile bir ağ üzerinden bağlantı kuran bilgisayarlar yer almaktadır. Bu katmandaki bilgisayarlar daha doğru bir zaman için aynı katmanlardaki cihazlar ile de bağlantı kurabilmektedir. Bu katmandan sonra aynı yapı diğer katmanlarda da kurulmaktadır. Bu yapı Şekil 3.11.’de görülmektedir.



Şekil 3.10. ESP32 yazılımına ait ana akış şeması.



Şekil 3.11. NTP protokolündeki cihazlara ait gösterim [44].

```

for(;;)
{
    uint8_t month = 0;
    uint8_t day = 0;
    uint16_t year = 0;

    timeClient.update();
    unsigned long epochTime = timeClient.getEpochTime();
    struct tm *ptm = gmtime ((time_t *)&epochTime);
    month = ptm->tm_mon+1;
    day = ptm->tm_mday;
    hour = timeClient.getHours() * 60 + timeClient.getMinutes() + 180;
    year = ptm->tm_year + 1900;

    uint16_t n1, n2, n3 = 0;
    n1 = floor(275 * month / 9);
    n2 = floor((month + 9) / 12);
    n3 = (1 + floor((year - 4 * floor(year / 4) + 2) / 3));
    n1 = n1 - (n2 * n3) + day - 30;

    setDayOfTheYear(n1);
    vTaskDelay(pdMS_TO_TICKS(1000));
}

vTaskDelete(NULL);

```

Şekil 3.12. NTP ile zaman bilgisinin alınmasına ait kod.

Arduino IDE’de zaman bilgisinin alınması için kullanılan kod Şekil 3.12.’de verilmiştir. Koda göre, öncelikle NTP ile UNIX zamanı (epoch time) bilgisi alınmaktadır. UNIX zamanı, 1 Ocak 1970’ten itibaren geçen zamanı saniye cinsinden tutmaktadır. Hesaplanan bu değere (UTC+3) 180 dakika eklenerek saat bilgisi dakika

cinsinden elde edilmektedir. UNIX zamanından elde edilen yıl, gün ve ay bilgileri kullanılarak yılın hangi gününde olduğu hesaplanmaktadır. Bu değer, güneşin doğuş ve batış zamanının hesaplanmasında kullanılmaktadır.

Güneşin doğuş ve batış zamanını hesaplamak için kullanılan algoritmaya göre, öncelikle bulunulan günün yılın hangi gününe denk geldiği belirlenmelidir [45]. Bu amaçla, güncel saat bilgisinin yanı sıra ay ve yıl bilgisi de alınmaktadır. Daha sonra güneşin ortalama anomali hesabını yapabilmek için bulunulan boylam değeri saat cinsine dönüştürülmektedir. Ortalama anomali değeri ile, güneşin açısal uzaklığını hesaplamak için kullanılan sağ açıklık değerini hesaplayabilmek için güneşin gerçek boylam değeri hesaplanır. Böylece güneşin belirli bir zamanda belirtilen koordinat değerlerine göre konumu belirlenebilecektir. Bununla birlikte, güneşin gerçek boylam değeri, güneşin ekvatora göre olan konumunu ifade eden dik açıklık değerini hesaplamak için de kullanılmaktadır. Hesaplanan dik açıklık değeri ve güneş ışınları ile dikey yön arasındaki açıyı belirten zenit açısı kullanılarak yerel saat bilgisi açı cinsinden hesaplanmaktadır. Son olarak hesaplanan bu değer dakika cinsine dönüştürülmekte, yerel ofset değeri de eklenerek güneşin doğuş ve batış zamanı belirlenebilmektedir. Sokak lambalarının aktif veya deaktif edilmesi güneşin doğuş ve batış zamanından 20 dakika önce veya sonra yapılmaktadır. Bu değer *config.h* dosyasında değiştirilebilmektedir. Böylece bölgesel ve mevsimsel ihtiyaçlara göre bu değer değiştirilebilecektir. Gözlemlenen sonuçlara göre güneşin gerçek doğuş batış zamanı ile hesaplanan değerler arasında oldukça küçük bir fark bulunduğu sonucuna varılmıştır.

Enerji aktarımının yapılmasını sağlamak için kesici ve ayırıcının anahtarlama işlemine manevra denilmektedir. Manevra işleminin sağlıklı yapılabilmesi amacıyla belirli şartlar sağlanmalıdır. Öncelikle, ayırıcının yük altında anahtarlama yapmasının önüne geçilmelidir. Yük altında ya da enerjili iken ayırıcının açılması, ayırıcının kontak uçlarında meydana gelen ark sebebiyle, ayırıcıya ciddi zarar verebilmektedir. Ayırıcının hatta enerji var ve kesici kapalı iken kapatılması da yine aynı sebepten oldukça zararlıdır. Bu nedenle kesici ve ayırıcının manevra işlemlerinin yapılmasında

bu şartların yanı sıra sistemde bulunan arızalar da kesicinin kontrol edilmesini etkilemektedir.

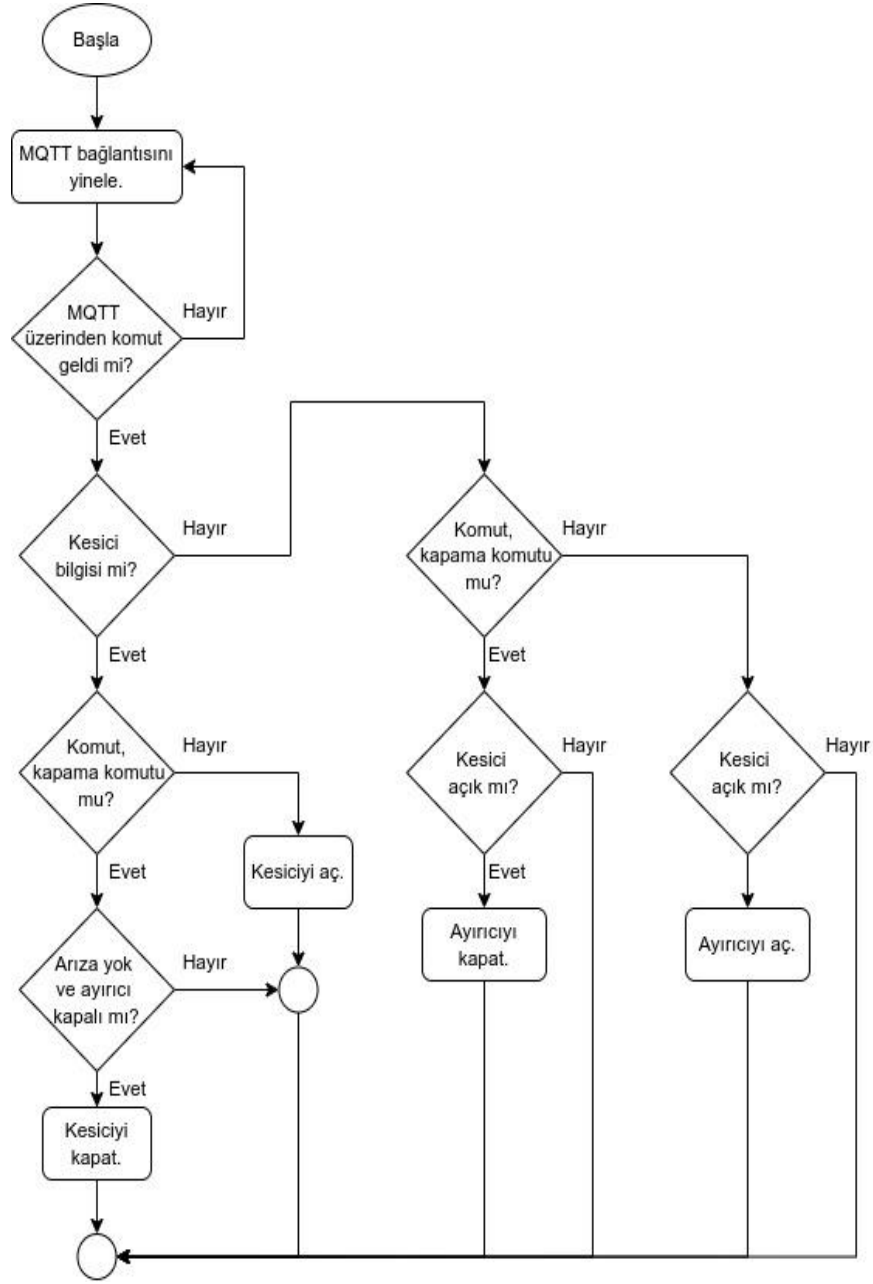
Kesici ve ayırıcının manevra komutu, geliştirilen grafiksel arayüz programı aracılığıyla MQTT protokolü üzerinden sağlanmaktadır. Kesici ve ayırıcının durumu hem ESP32 hem de grafiksel arayüz programında kontrol edilmektedir. Kesici ve ayırıcının kontrolü için geliştirilen algoritmaya ait akış şeması Şekil 3.13.'te verilmiştir.

Görevlerin çizelgelemesi FreeRTOS ile *main.ino* modülünde yapılmıştır. Her bir göreve farklı öncelik atanması ile öncelikli görevler belirlenmiştir. Görevlerdeki öncelik sıralaması değiştirilebileceğinden, eklenecek görevlerin önceliği de bu duruma adapte edilebilecektir. Her bir görevin çalışma zamanı, yüksek öncelikli göreve daha düşük çalışma zamanı verilerek belirlenmiştir.

Şekil 3.7.'den görüleceği üzere; kesicinin açtırılması güvenlik gerekçesiyle belirli bir şarta bağlanmamıştır. Kesici, açtırılma komutu geldiğinde, açtırılmaktadır. Böylece trafo merkezinin enerjisiz kalması istendiğinde bu durum doğrudan yapılabilmektedir. Bu durumun aksine, sistemde bir arıza olması durumunda kesicinin aktif edilmesi engellenerek sistem güven altında tutulmaya devam edilmiştir. Kurulan algoritma ile birlikte kesici ve ayırıcının uzaktan kontrolü güvenle yapılmaktadır.

3.2.4. Grafiksel arayüz programı yazılım mimarisi

Arduino IDE'de ESP32 geliştirme kiti ile geliştirilen sistemin uzak kontrolünün sağlanması için bir MQTT Broker üzerinden sistemle bağlantı kurulması gerekmektedir. MQTT Broker, abone-yayıncı mantığı üzerinden haberleşme bağlantısı kurulan iki sistem arasında yayının abonelere yönlendirilmesini sağlamak amacıyla kullanılır. Günümüzde birçok MQTT Broker bulunmaktadır: EMQX, HiveMQ, Mosquitto vb. Bununla birlikte, MQTT bağlantısının bir cep telefonu uygulaması üzerinden yönlendirilebildiği MQTT Dashboard, Blynk ve Easy MQTT gibi uygulamalar da mevcuttur.



Şekil 3.13. Kesici ve ayırıcı kontrol yazılımına ait akış şeması.

ESP32 ile MQTT haberleşmesinin yönlendirilebilmesi için Adafruit MQTT Broker kullanılmıştır. Adafruit Broker seçilmesinin sebebi; ücretsiz versiyonunun bulunması, Arduino ve Python'a entegresinin kolay olması, kanallardaki verilerin kaydedilmesi sebebiyle sonradan kullanılabilmesi ve internet sitesinde buton, metin ve grafik gibi bileşenlerden oluşan bir grafiksel arayüz tasarımına imkan tanınmasıdır. Adafruit MQTT Broker'ın Arduino IDE ve grafiksel arayüz programına entegre edilebilmesi amacıyla Adafruit tarafından geliştirilen kütüphaneler kullanılmıştır.

MQTT Broker seçildikten sonra arayüz programının tasarlanması için platform bağımsız çalışabilen Qt kütüphanesi kullanılmıştır. Qt, C++ için geliştirilmesine rağmen diğer programlama dilleri için de bağlantıları mevcuttur. Adafruit Broker'ın Python ile kullanılması, Qt'nin Python bağlantısının seçilmesine sebep olmuştur. Birçok Qt Python bağlantısı arasından PySide6 bağlantısı seçilmiştir. Bunun sebebi, PySide kütüphanesinin Qt'nin resmi desteğine sahip olmasıdır. Dolayısıyla, uygulama geliştirilirken resmi destek sayfalarından oldukça yararlanılmıştır.

Arayüz programı, Linux işletim sistemine sahip bilgisayarda Qt tarafından geliştiricilerin kullanımına sunulan Qt 6 Designer programında tasarlandıktan sonra, VSCodium metin editöründe kodlanmıştır. Arayüz tasarımı, Qt Designer programında Python koduna dönüştürülerek uygulama kodunda bu sınıftan oluşturulan nesne kullanılmıştır.

Qt'nin en önemli özelliği sinyal ve yuva (signal-slot) mekanizmasıdır. Bu yapı, aydınlatma anahtarının durum değiştirmesinin (sinyal), lambanın açılmasına veya kapanmasına (yuva) sebep olmasına benzetilebilir. Bu, bazı programlama araçlarında bulunan geri çağırma fonksiyonlarına benzetilebilir. Bu fonksiyonlar bir bildirim fonksiyonuna bağlıdır ve beklenen bir olay meydana geldiğinde geri çağırma fonksiyonuna işaretçi aracılığıyla bildirimde bulunur. Ancak; Qt'de kullanılan sinyal ve yuva yapısı önemli farklar arz etmektedir. Bunlardan biri, geri çağırma fonksiyonlarının argümanlar üzerinde tip doğrulaması yapmamasıdır.

Sinyal ve yuva yapısı kullanılarak bir sınıftan oluşan nesne ile kullanıcının etkileşime geçmesi sonucunda ne gibi işlemlerin yapılacağı belirlenir. Örneğin; bir butona basıldığında üzerinde çalışılan işlemin bir dosyaya kaydedilmesi sinyal ve yuva yapısının kullanılmasına örnek teşkil etmektedir. Bu örnekte, butona kullanıcı tarafından basılması sinyali, dosyaya yazılma işlemini gerçekleştiren metod ise yuvayı temsil etmektedir.

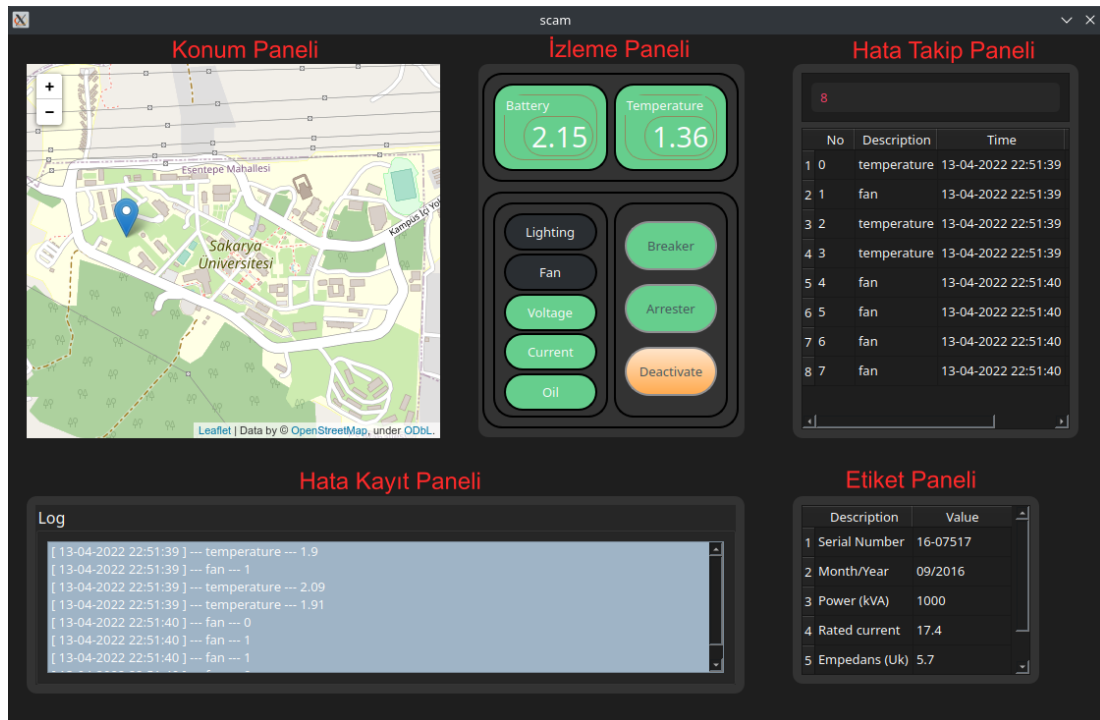
Qt kütüphanesi ile kullanılan nesnelere durum değiştirdiklerinde sinyal yaymaktadırlar. Yuva, yayılan bu sinyalleri alarak üzerinde çalışır. Bir sınıfın doğal üyesi olmasıyla,

birden fazla sinyalin aynı yuvaya bağlanmasının yanı sıra, bir sinyal başka bir sinyale de bağlanabilmektedir. Ayrıca, bir sinyal birden fazla yuvaya da bağlanabilmektedir. Bununla birlikte, yuva kendisine herhangi bir sinyalin bağlanıp bağlanmadığı bilgisini göremez. Qt nesnelere birçok ön tanımlı sinyale sahiptir.

Arayüz programında, trafo merkezinin konumunu gösteren bir bölüm bulunmaktadır. Bu bölümün oluşturulmasında Python kütüphanelerinden Folium kullanılmıştır. Bu kütüphane açık kaynak kodlu OpenStreetMap haritasının kullanılabilmesini sağlamaktadır. OpenStreetMap, dünya haritasının uygulamalarda ücretsiz şekilde kullanılmasını sağlamaktadır. Bunun için tek gereklilik, kullanılan haritaya “© OpenStreetMap contributors” ibaresinin eklenmesidir. OpenStreetMap ile birçok katman kullanılabilir. Arayüz programı başladığında ön tanımlı koordinata sahip bir trafo merkezinin konumu görüntülenmektedir. Böylece, sisteme ileride eklenecek trafo merkezinin konumu ile ayırt edilmesi sağlanmıştır.

Arayüz programında, sensör bilgilerini takip etmek ve kesici-ayırıcı birimlerinin kontrolünü sağlamak amacıyla bir panel yer almaktadır. Tüm paneller isimleri ile birlikte Şekil 3.14.’te görülmektedir. Sıcaklık ve batarya gerilimi uzaktan takip edilmekte, gerilim seviyelerine göre kullanıcıya görsel uyarı verilmektedir. Sensörlerden alınan analog bilgi 0-1023 arasında bir değerdir. Bu değer analog gerilim seviyesine dönüştürülerek kullanıcıya bu bilgi gösterilmektedir. Diğer lojik sensörlerden bir bilgi alınması durumunda bu bilgi anlık olarak panelde görüntülenmektedir. Kesici ve ayırıcının durumları uygun renklerle kullanıcıya bildirilirken; programda kesici ve ayırıcıyı tek seferde deaktif etmek için de bir buton bulunmaktadır.

Kullanıcı sistemdeki uyarıcıları takip ederek, sistemin sağlığı hakkında bilgi sahibi olabilmektedir. Arayüz programına eklenebilecek bildirim özelliği sayesinde, kullanıcı sistemdeki arıza durumlarında hızla bilgi sahibi olabilecektir. Bu sayede sistemin izlenebilirliği artırılabilir. Arızaların hızla değerlendirilmesi; bakım zamanı hesaplaması, arıza kaynağının tespiti ile aynı arızanın tekrar alınmasını önleyerek maddi kayıpların önüne geçilmesi verimliliği önemli ölçüde arttıracaktır.



Şekil 3.14. Arayüz programındaki paneller.

Arayüz programı çalışırken elde edilen hatalar, izleme panelinde anlık görüldüğünden elde edilen son hataların sıralandığı bir panel de arayüze eklenmiştir. Buna göre; alınan son hata, MQTT konu adı ve alınan değer ile birlikte bu panelde, hatanın alındığı zaman bilgisi ile birlikte görüntülenmektedir. Hata takip panelinde program çalıştıktan sonra alınan toplam hata sayısı da izlenebilmektedir. Alınan hataların kayıt altında tutulması için, hatalar belirli bir formatta bir metin dosyasına yazılmaktadır. Yazılan hatalar arayüz programında hata kayıt panelinde görüntülenmektedir. Trafo merkezinde bulunan trafoya ait etiket bilgileri için de bir panel, programa eklenmiştir. Arayüz programına ait akış şeması Şekil 3.15.'te görülmektedir. Akış şemasında görüleceği üzere; program başlayıp Adafruit ile MQTT bağlantısı kurulmasının ardından yayımlanan son sensör bilgileri alınmaktadır. Bu bilgiler değerlendirilerek uygun görsel uyarıcılar ile kullanıcı bilgilendirilmektedir. Program çalışırken sensörlerden okunan bilgi değiştiğinde ilgili alandaki değer yinelenmektedir. Böylece izleme panelinde sistem canlı bir şekilde izlenebilmektedir. Bununla birlikte, elde edilen değerlerin daha sonra analizi yapılarak, elde edilen sonuçlara göre sistemdeki bileşenlerin bakım zamanları önceden belirlenerek, bu cihazların arıza yapmasının önüne geçilmesi mümkündür. Sonuçta, sistemin kullanım ömrü artacak, sistemin işletme maliyeti azalacaktır.



Şekil 3.15. Arayüz programının akış şeması.

Programda kesici ve ayırıcı butonlarına basıldığındaki işlem adımı sırası Şekil 3.10.'daki akış şemasına benzer şekilde yürütülmektedir. Böylece trafo merkezindeki kesici ve ayırıcının güvenliği iki adımda yapılmaktadır. Kesici ve ayırıcının aktif edilemediği durumlarda, kullanıcı hata kayıt panelinde bilgilendirilmektedir. Kesici ve ayırıcı ünitelerinin aynı anda deaktif edilmesi istendiği durumlarda, deaktifasyon butonuna basılması ile birlikte önce kesici, ardından ayırıcı deaktif edilmektedir. Sistemin tekrar devreye alınması için kesici ve ayırıcı ayrı ayrı aktif edilmektedir.

BÖLÜM 4. ARAŞTIRMA BULGULARI

4.1. Uygulamanın Test Edilmesi

Önerilen sistem, iki bileşenden oluşmaktadır: Sistemin kontrol edildiği ESP32 ile uygulanan prototip ve bu prototip ile uzak izleme ve kontrolün sağlandığı arayüz programı. Arayüz programı ile prototip arasındaki haberleşme bağlantısı MQTT protokolü üzerinden sağlanmıştır. Bağlantı için gerekli Adafruit Broker kullanılmıştır. Adafruit firması tarafından sağlanan Arduino IDE ve Python kütüphaneleri kullanılmıştır.

Uygulama ile trafo merkezlerinin uzaktan izleme ve kontrol işlemlerinin yapılabileceği; sistemdeki arızaların kullanıcıya bildirilmesi ve batarya gerilimindeki düşüş gibi kullanıcının bakım için önceden bildirilmesi gibi faydalı sonuçlara ulaşılmıştır. Prototipin uzak kontrolü ve sistemde bulunan sensörlerden elde edilen bilgiler doğru şekilde, arayüz programında görüntülenmiştir. Trafo merkezlerinde meydana gelebilecek arızaların kaydı arayüz programında tutulmuştur. Bu amaçla meydana gelen arızalar bir metin dosyasına kaydedilirken; kullanıcı da bu işlem ve meydana gelen arıza konusunda, arayüz programında bulunan ilgili paneller üzerine arıza bilgilerinin yazılması ile bilgilendirilmiştir.

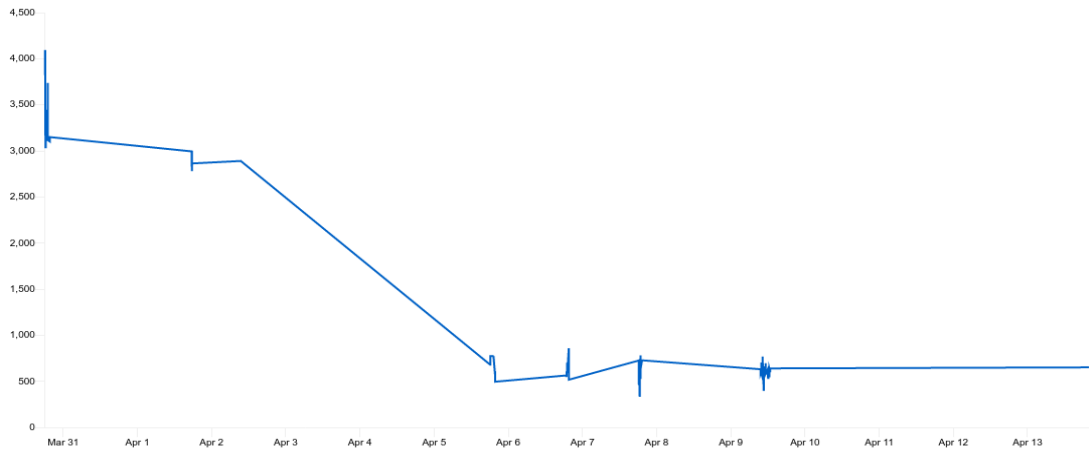
Sokak lambalarının, güneşin doğuş ve batış zamanlarına göre otomatik kontrolü çok az bir sapma ile sağlanabilmiştir. Sokak lambalarının kontrolü için kullanılan kod parçacığı ile hesaplanan güneşin doğuş ve batış zamanları ile gerçek değerler arasındaki farklar Tablo 4.1.'de yılın farklı günleri için 30 günlük bir zaman aralığında hesaplanmıştır. Tablodaki verilerden görüleceği üzere çok az fark bulunmaktadır. Hesaplanan değerler uygulamada tolere edilmiştir. Böylece sonuçlarla birlikte sokak lambası doğru bir şekilde kontrol edilebilmiştir.

Tablo 4.1. Güneşin doğuş ve batış zamanı için hesap sonuçları.

No	Yılın Günü	Hesaplanan doğuş zamanı	Doğuş zamanı [46]	Doğuş zamanı farkı	Hesaplanan batış zamanı	Batış zamanı [46]	Batış zamanı farkı
1	1	8:22	8:20	+2	17:41	17:42	-1
2	21	8:17	8:15	+2	18:02	18:03	-1
3	36	8:05	8:03	+2	18:20	18:21	-1
4	47	7:51	7:49	+2	18:33	18:35	-2
5	67	7:22	7:20	+2	18:57	18:58	-1
6	80	7:01	6:58	+3	19:11	17:12	-1
7	89	6:46	6:43	+3	19:20	19:22	-2
8	100	6:28	6:26	+2	19:32	19:33	-1
9	115	6:05	6:03	+2	19:47	19:49	-2
10	125	5:52	5:50	+2	19:58	20:00	-2
11	137	5:39	5:37	+2	20:10	20:12	-2
12	148	5:31	5:29	+2	20:20	20:21	-1
13	157	5:27	5:26	+1	20:26	20:28	-2
14	172	5:27	5:25	+2	20:33	20:35	-2
15	181	5:30	5:28	+2	20:33	20:35	-2
16	192	5:36	5:35	+1	20:30	20:32	-2
17	207	5:48	5:48	0	20:20	20:22	-2
18	224	6:05	6:04	+1	20:01	20:02	-1
19	234	6:14	6:14	0	19:47	19:48	-1
20	247	6:27	6:27	0	19:26	19:28	-2
21	256	6:36	6:35	+1	19:11	19:13	-2
22	266	6:46	6:45	+1	18:54	18:56	-2
23	274	6:54	6:53	+1	18:41	18:42	-1
24	284	7:04	7:04	0	18:25	18:26	-1
25	297	7:19	7:18	+1	18:05	18:07	-2
26	305	7:28	7:27	+1	17:55	17:56	-1
27	318	7:43	7:42	+1	17:41	17:43	-2
28	333	8:00	7:59	+1	17:32	17:34	-2
29	342	8:09	8:08	+1	17:30	17:32	-2
30	355	8:18	8:17	+1	17:33	17:35	-2

Adafruit internet sitesinde konulara gönderilen bilgiler ücretsiz versiyonda 30 gün boyunca kaydedilmektedir. Bu sebeple; sensörlerden alınan değerler ve kesici ile ayırıcı manevraları 30 günlük periyotlar ile görüntülenebilmektedir. Bu değerler gerektiğinde kullanıcı cihazına JSON veya CSV formatında kaydedilebilmektedir. Böylece kayıtlı veri üzerinde gerekli değerlendirmeler yapılabilmektedir. Bu durum kullanıcıya kesici ve ayırıcının manevra sayısı üzerinden bu ünitelerin kullanılma miktarı bilgisini vereceğinden bu ünitelere yapılacak bakım zamanının yaklaşp yaklaşmadığının anlaşılmasına yol açacaktır. Ayrıca; arızaların kaydı tutulduğundan trafo merkezinde en fazla alınan arıza türü ve miktarı da belirlenebilmesi sebebiyle, ilgili trafo merkezindeki iyileştirmeler yapılarak, trafo merkezinin faaliyet süresi arttırılabilecektir. Adafruit internet sitesinde çeşitli konularda alınan kayıtlara ait ekran görüntüsü Şekil 4.1.'de verilmiştir.

Battery voltage value feed.



Temperature value feed.



Şekil 4.1. Çeşitli konulardaki kayıtlara ait ekran görüntüsü.

Şekil 4.1.'de ekran görüntüsü paylaşılan konulara ait değerlere ait JSON formatındaki verilerin bir kısmı Şekil 4.2.'de görülmektedir. Veriler ilk verinin alındığı tarihten başlayarak en son verinin alındığı tarihe doğru artmaktadır. Kaydedilen veriler, programlama dilleri ve çeşitli araçlar ile kolay bir şekilde değerlendirilebilmektedir.

```
[{"id": "0EZTC2HMTD4RJS6MD1JHG93J84", "value": "4095", "feed_id": "1844054", "created_at": "2022-03-30T18:08:49.441Z"},
{"id": "0EZTC2JMAVR4T6XYJSGE96CEDQ", "value": "4095", "feed_id": "1844054", "created_at": "2022-03-30T18:08:52.668Z"},
{"id": "0EZTC2K6718CGKW1MNMWCXR3B", "value": "4095", "feed_id": "1844054", "created_at": "2022-03-30T18:08:55.523Z"},
{"id": "0EZTC2MFFD8X5FF7XRVGW9QGKY", "value": "4095", "feed_id": "1844054", "created_at": "2022-03-30T18:08:58.724Z"},
{"id": "0EZTC2NC9XCQASBXAQP6X4MEE1", "value": "3520", "feed_id": "1844054", "created_at": "2022-03-30T18:09:01.676Z"},
{"id": "0EZTC2P86187R00RRKY3ZP99DK", "value": "3517", "feed_id": "1844054", "created_at": "2022-03-30T18:09:04.531Z"},
{"id": "0EZTC2Q6CZ80QMF9E1BATDWE4", "value": "3510", "feed_id": "1844054", "created_at": "2022-03-30T18:09:07.625Z"},
{"id": "0EZTC2R472X3MPWPBP33VZVJN9Z", "value": "3509", "feed_id": "1844054", "created_at": "2022-03-30T18:09:10.678Z"},
{"id": "0EZTC2T0ESJVJHMBD08MZ7D7X", "value": "3521", "feed_id": "1844054", "created_at": "2022-03-30T18:09:16.847Z"},
{"id": "0EZTC2T0F75HDXRW65EB48RNY5", "value": "3523", "feed_id": "1844054", "created_at": "2022-03-30T18:09:16.848Z"},
{"id": "0EZTC2HMFHT6N9BK2VA142JXZ", "value": "1360", "feed_id": "1844055", "created_at": "2022-03-30T18:08:49.406Z"},
{"id": "0EZTC2J393GRPS3GKNNEQPZNXA", "value": "1361", "feed_id": "1844055", "created_at": "2022-03-30T18:08:52.457Z"},
{"id": "0EZTC2KFKX436SD97XYX4S3CQT", "value": "1361", "feed_id": "1844055", "created_at": "2022-03-30T18:08:55.462Z"},
{"id": "0EZTC2MCS5NVAWBTMDM151GYHG", "value": "1360", "feed_id": "1844055", "created_at": "2022-03-30T18:08:58.448Z"},
{"id": "0EZTC2NA7CEZ6H55NNQPAN26TT", "value": "1360", "feed_id": "1844055", "created_at": "2022-03-30T18:09:01.463Z"},
{"id": "0EZTC2P7M91DFBQMQV5MJEW7Z", "value": "1359", "feed_id": "1844055", "created_at": "2022-03-30T18:09:04.474Z"},
{"id": "0EZTC2Q4VDY5PZ4BHXRMV835", "value": "1360", "feed_id": "1844055", "created_at": "2022-03-30T18:09:07.466Z"},
{"id": "0EZTC2R28EQWVTT7347XV0RRQJ", "value": "1385", "feed_id": "1844055", "created_at": "2022-03-30T18:09:10.478Z"},
{"id": "0EZTC2T0EY7JXTGNSPK9CFAQTR", "value": "1363", "feed_id": "1844055", "created_at": "2022-03-30T18:09:16.847Z"},
{"id": "0EZTC2T0F7Z1SR759YS53NNG08", "value": "1360", "feed_id": "1844055", "created_at": "2022-03-30T18:09:16.848Z"}]
```

Şekil 4.2. Çeşitli konulardaki JSON formatındaki kayıtlar.

Prototipte, kesici ve ayırıcı için ayrılan ledlerin kontrolü grafiksel arayüz programında yapılmıştır. Bu programda bahsedilen ünitelerin güvenli kontrolü Şekil 4.3. ve 4.4.'te görülmektedir. Şekil 4.3.'te enerji altında iken ayırıcının açılması önlenmişken; Şekil 4.4.'te ayırıcı açık iken kesicinin kapatılması engellenmiştir.



Şekil 4.3. Arayüz programında ayırıcının güvenli çalışması.



Şekil 4.4. Arayüz programında kesicinin güvenli çalışması.

Batarya geriliminin takibi MQTT protokolü ile sağlanmıştır. Gerilimin belli bir seviyenin altına inmesi sonucunda kullanıcı görsel olarak uyarılmıştır. Gerilimin tekrar uygun seviyeye çıkması ile birlikte gerilim normal şekilde görüntülenmiştir. Gerilim belli bir seviyenin altına düştüğünde arayüz programında görüntülenen uyarıya ait ekran görüntüsü Şekil 4.5.'te görülmektedir. Gerilimin düşük olduğu bilgisi hata takip ve hata kayıt panellerinde görüntülenmiştir. Alınan hata ise metin dosyasına kaydedilmiştir.

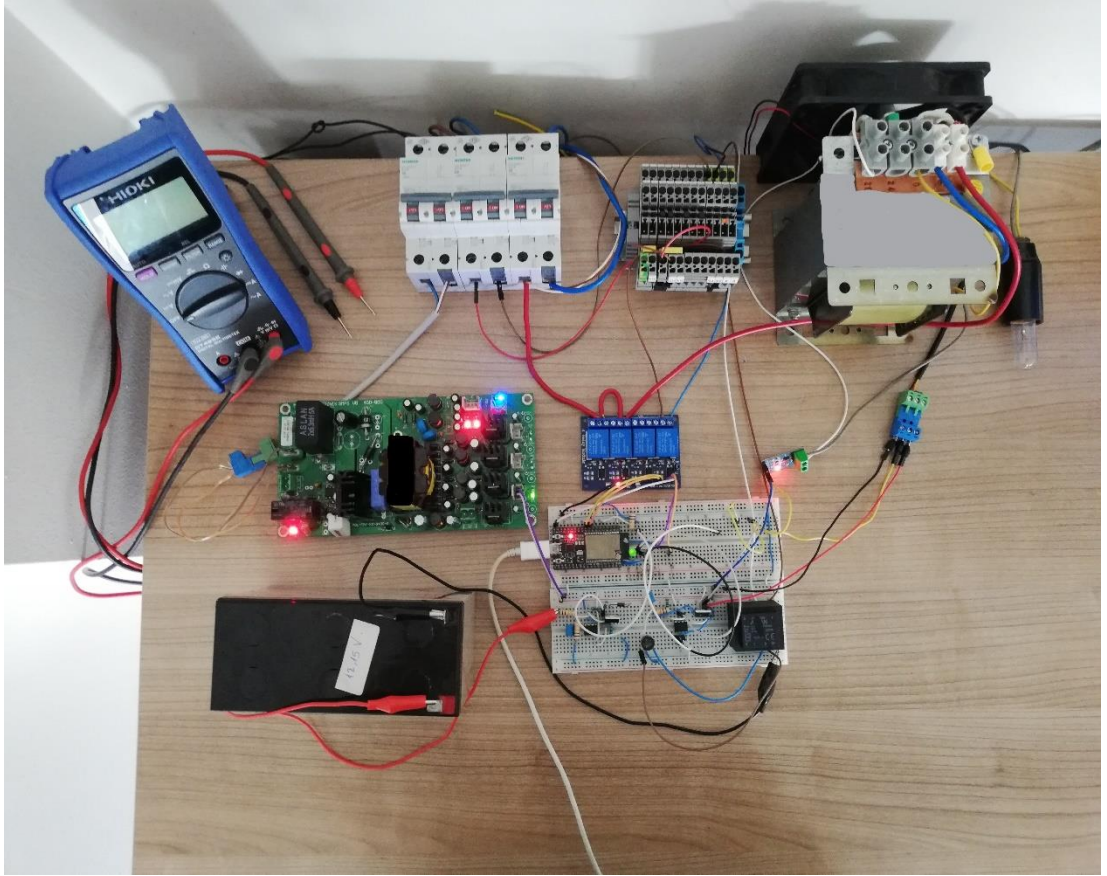
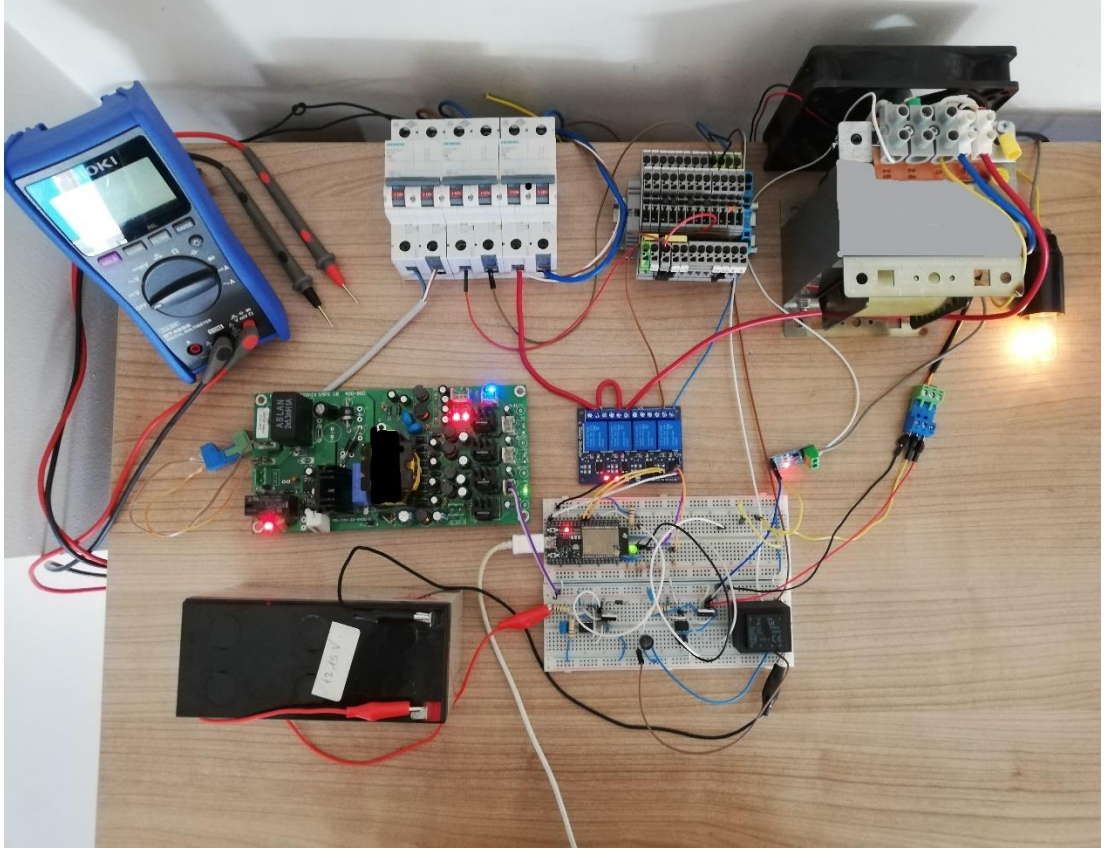
Sistemde, trafo merkezinde olduğu düşünülen sensörlerden gelen lojik bilgilere göre kesici açtırılmaktadır. Örneğin yüksek akım hatası alındığında kesici açtırılmış; hata bilgisi arayüz programında görüntülenmiş ve prototipte ilgili ledler deaktif edilmiştir. Lojik bilgiye sahip hataların, anlık olmaları sebebiyle arayüz programında görüntülenmesi mümkün değildir. Hata kayıtlarının tutulması ile arıza takibi yapılabilmektedir. Bununla birlikte; hatanın izleme panelinde görüntülenebildiğini göstermek amacıyla yüksek akım hata sinyali sürekli verilmiştir. Şekil 4.6.'da izleme ve hata panellerinde arızanın kullanıcıya bildirilmesi; Şekil 4.7.'de ise kesici ve ayrıncı çıkışlarının hata alınmadan önce ve hata alındıktan sonraki durumları görülmektedir.



Şekil 4.5. Düşük batarya gerilimi uyarısına ait ekran görüntüsü.



Şekil 4.6. Yüksek akım hatasının kullanıcıya bildirilmesine ait ekran görüntüsü.



Şekil 4.7. Kesicinin yüksek akım hatası ile açtırılmasına ait görüntüler.

Trafoaların sıcaklıklarının belli bir değerin üzerine çıkması engellenmelidir. Sıcaklık kontrol edilmezse trafonun yanması gibi sonuçlar ile karşılaşılabilir. Önerilen sistemde bu durumu önlemek amacıyla sıcaklık sensöründen okunan değere göre fana bağlı röle çektilirerek fanın çalışması sağlanmıştır. Böylece trafonun yüksek sıcaklığa bağlı arızalanmasının önüne geçilmiştir. Okunan sıcaklık değerinin artarak 75 °C'nin üzerine çıkması durumunda kesici açtırılmıştır. Arayüz programında sıcaklığa bağlı olarak fanın çalışması, kullanıcının görsel olarak uyarılması ve kesicinin belli bir sıcaklık üzerinde açtırılmasına ait ekran görüntüleri Şekil 4.8. ve 4.9.'da verilmiştir.



Şekil 4.8. Sıcaklığın artması ile birlikte fanın çalışmasına ait görüntü.



Şekil 4.9. Sıcaklığın artması ile birlikte kesicinin açtırılmasına ait görüntü.

Prototip kontrol edilirken, arayüz programı ile gönderilen komutlara ve sensörlerden gelen bilgilere verilen tepkiler Arduino IDE seri ekranına yazılmaktadır. Böylece arayüz programına erişimin olmadığı durumlarda prototip ile iletişim kurulabilmektedir. Arduino IDE seri ekranına ait ekran görüntüsü Şekil 4.11.'de verilmektedir. Ayrıca arayüz programında karşılaşılan hatalar bir metin dosyasına kaydedilmektedir. Bu nedenle, program çalışırken karşılaşılan sorunlar daha sonra değerlendirilebilmektedir. Elde edilen hata kayıtlarına ait ekran görüntüsü Şekil 4.12.'de görülmektedir.

Kayıt altına alınan hatalar daha sonra değerlendirilmek üzere kullanılabilir. Böylece alınan hata miktarları ve hangi hatanın daha fazla alındığı ile istatistiksel bilgiler değerlendirilebilmektedir. Sistemde karşılaşılan durumların Arduino IDE'de görüntülenmesi ile sistemin lokal takibi yapılabilir. Bunun sonucunda internet bağlantısının olmadığı durumlarda saha ekiplerinin sistemi gözetlemesi kolaylaşacaktır.

```

/dev/ttyUSB0
| Send
KRa0L{jrB{C{}}=D{?M=BSrR{Connecting to WiFi .....WiFi connected!
Connecting to MQTT... MQTT Connected!
Arrester request: ON
ARRESTER: ON
Breaker request: ON
BREAKER: ON
Sıcaklık yüksek! Fan çalıştırılıyor.
Sıcaklık aşırı yüksek! Kesici açtırılıyor.
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Batarya gerilimi düşük!
Breaker request: OFF
BREAKER: OFF
Breaker request: ON
BREAKER: ON
Yağ seviyesi düşük!
Breaker request: OFF
BREAKER: OFF
Arrester request: OFF
ARRESTER: OFF
Autoscroll Show timestamp Newline 9600 baud Clear output

```

Şekil 4.10. Arduino IDE seri ekranında görüntülenen bilgiler.

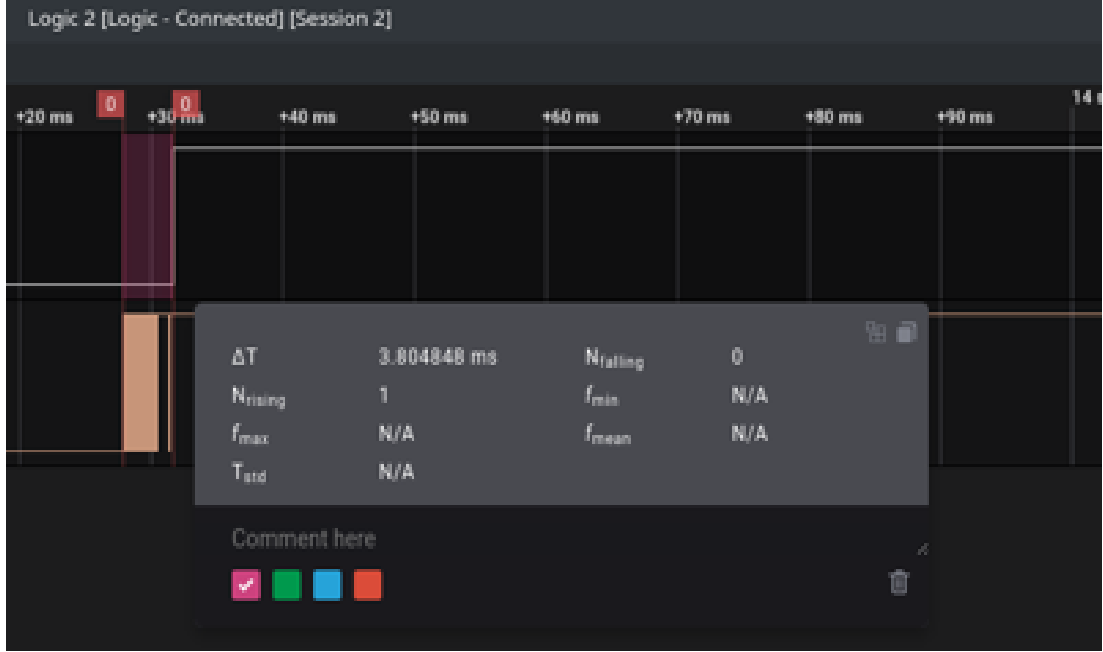

```

1 [ 23-04-2022 14:58:16 ] --- fan --- 1
2 [ 23-04-2022 14:58:16 ] --- temperature --- 1.92
3 [ 23-04-2022 14:58:22 ] --- fan --- 0
4 [ 23-04-2022 14:58:22 ] --- fan --- 1
5 [ 23-04-2022 14:58:23 ] --- temperature --- 2.21
6 [ 23-04-2022 14:58:25 ] --- temperature --- 2.0
7 [ 23-04-2022 14:58:25 ] --- temperature --- 2.19
8 [ 23-04-2022 14:58:25 ] --- temperature --- 2.0
9 [ 23-04-2022 14:58:25 ] --- temperature --- 2.37
10 [ 23-04-2022 14:58:26 ] --- temperature --- 2.18
11 [ 23-04-2022 14:58:26 ] --- temperature --- 2.36
12 [ 23-04-2022 14:58:26 ] --- temperature --- 2.17
13 [ 23-04-2022 14:58:27 ] --- temperature --- 2.4
14 [ 23-04-2022 14:58:27 ] --- temperature --- 2.19
15 [ 23-04-2022 14:58:27 ] --- temperature --- 2.37
16 [ 23-04-2022 14:58:27 ] --- temperature_fault --- 2.68
17 [ 23-04-2022 14:58:27 ] --- temperature_fault --- 2.51
18 [ 23-04-2022 14:58:30 ] --- temperature_fault --- 2.7
19 [ 23-04-2022 14:58:31 ] --- temperature --- 2.5
20 [ 23-04-2022 14:58:34 ] --- temperature --- 2.32
21 [ 23-04-2022 14:58:34 ] --- temperature --- 1.86
22 [ 23-04-2022 14:58:34 ] --- temperature --- 2.19
23 [ 23-04-2022 14:58:35 ] --- temperature --- 2.01
24 [ 23-04-2022 14:58:35 ] --- temperature --- 1.84
25 [ 23-04-2022 14:58:36 ] --- fan --- 0
26 [ 23-04-2022 14:58:36 ] --- fan --- 1
27 [ 23-04-2022 14:58:36 ] --- fan --- 0
28 [ 23-04-2022 14:59:39 ] --- battery --- 1.84
29 [ 23-04-2022 15:00:04 ] --- oil --- 1
30 [ 23-04-2022 15:00:05 ] --- oil --- 0
31 [ 23-04-2022 15:00:41 ] --- lighting --- 0
32 [ 23-04-2022 15:01:17 ] --- fan --- 1
33 [ 23-04-2022 15:01:17 ] --- temperature --- 1.91
34 [ 23-04-2022 15:01:20 ] --- temperature --- 2.07

```

Şekil 4.11. Alınan hataların bir metin dosyasına kaydedilmesine ait görüntü.

Akım ve gerilim ölçmek için kullanılan devrelerden gelen sinyale işlemcinin hızlı tepki vermesi güvenlik sebebiyle oldukça önemlidir. Bu sinyallerin işlemci tarafından algılanması ile kesici açtırılarak trafonun korunması sağlanmalıdır. Lojik analizör ile yapılan ölçümler neticesinde akım ve gerilim sensörü için tepki süresi sırasıyla 3.8 ve 5.5 ms olarak gözlemlenmiştir. En geç tepki süresi ise, bu pinleri kontrol eden görevin 20 ms'de bir çalışması sebebiyle 20 ± 1.5 ms olarak elde edilmiştir. İlgili pinlerin kesme ile takip edilmesi, tepki süresini önemli ölçüde düşürecek olsa da; geliştirilen uygulama için elde edilen değerlerin yeterli olduğu sonucuna varılmıştır. Akım ve gerilim sensörü için ölçülen tepki süresine ait ekran görüntüleri Şekil 4.12. ve Şekil 4.13.'te verilmiştir.



Şekil 4.12. Akım sensörüne ESP32 kitinin verdiği tepki süresine ait görüntü.



Şekil 4.13. Gerilim sensörüne ESP32 kitinin verdiği tepki süresine ait görüntü.

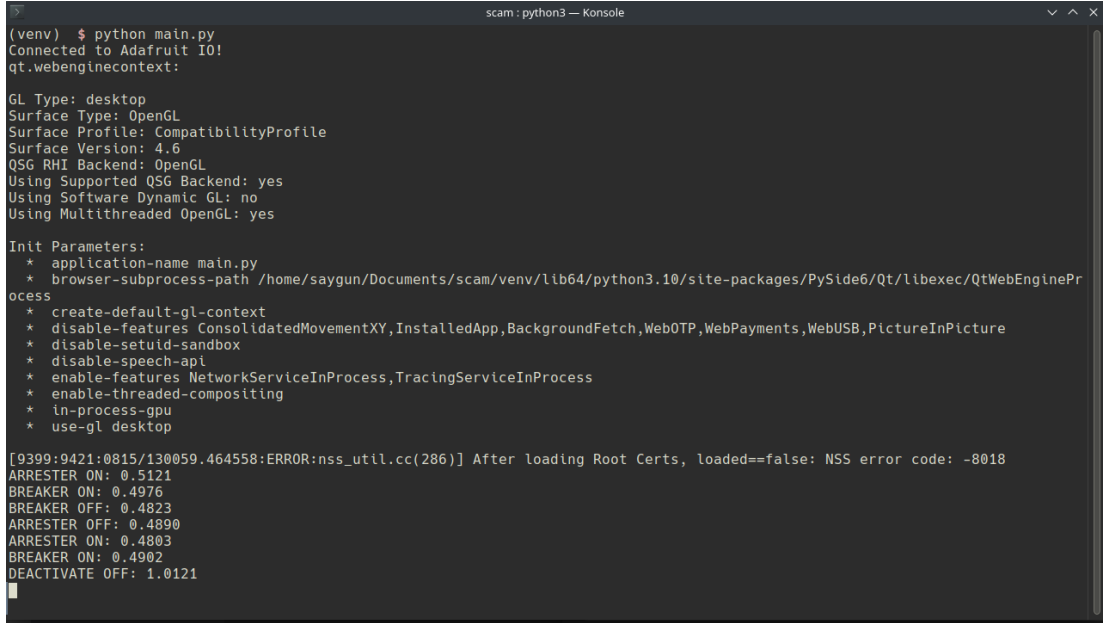
Akım sensörünün doğruluğunun tespiti amacıyla, sensörün ölçüm uçları seri olarak yüke bağlanmıştır. Sensöre bağlı yük, yük üzerinden akım ve sensörün akım değerine göre ürettiği gerilim değerleri Tablo 4.2.'de görülmektedir. Tablodaki verilerden görüleceği üzere; elde edilen değerlerin sensörün veri sayfalarında yer alan değerler ile örtüştüğü sonucuna varılmıştır. Trafonun yüksek akım koruma değeri, sensörden okunan değere göre değiştirilebilmektedir. Bu sayede, sistemin daha esnek olması sağlanmaktadır. Geliştirilen prototipte bu akım değeri 2.5A olarak belirlenmiştir. Böylece, trafonun sekonder kısmından 2.5A'in üzerinde bir akım akması durumunda

kesici açtırılarak, trafo korunmuştur. Trafonun çalışması bu sayede daha güvenli hale getirilmiştir.

Tablo 4.2. Akım sensöründen elde edilen değerler.

No	Akım Değeri (A)	Sensörden Elde Edilen Gerilim Değeri (VDC)
1	1	2.62
2	1.5	2.73
3	2	2.85
4	2.5	3.00

Geliştirilen arayüz programında, kesici ve ayırıcı ünitelerinin kontrol edilmesinde belirli bir zaman geçmesi beklenmiştir. Kodda ilgili kısma zamanlayıcı eklenmesi ile birlikte, her bir kesici ve ayırıcı aktif ve deaktif etme işleminin yaklaşık 0.5 sn sürdüğü gözlemlenmiştir. Arayüz programı çalışırken yapılan ölçüme ait ekran görüntüsü Şekil 4.14.'te verilmiştir.



```

scam: python3 — Konsole
(venv) $ python main.py
Connected to Adafruit IO!
qt.webenginecontext:

GL Type: desktop
Surface Type: OpenGL
Surface Profile: CompatibilityProfile
Surface Version: 4.6
QSG RHI Backend: OpenGL
Using Supported QSG Backend: yes
Using Software Dynamic GL: no
Using Multithreaded OpenGL: yes

Init Parameters:
* application-name main.py
* browser-subprocess-path /home/saygun/Documents/scam/venv/lib64/python3.10/site-packages/PySide6/Qt/libexec/QtWebEnginePr
ocess
* create-default-gl-context
* disable-features ConsolidatedMovementXY,InstalledApp,BackgroundFetch,WebOTP,WebPayments,WebUSB,PictureInPicture
* disable-setuid-sandbox
* disable-speech-api
* enable-features NetworkServiceInProcess,TracingServiceInProcess
* enable-threaded-compositing
* in-process-gpu
* use-gl desktop

[9399:9421:0815/130059.464558:ERROR:nss_util.cc(286)] After loading Root Certs, loaded==false: NSS error code: -8018
ARRESTER ON: 0.5121
BREAKER ON: 0.4976
BREAKER OFF: 0.4823
ARRESTER OFF: 0.4890
ARRESTER ON: 0.4803
BREAKER ON: 0.4902
DEACTIVATE OFF: 1.0121

```

Şekil 4.14. Arayüz programında kesici ve ayırıcı kontrolündeki zaman kaybı.

BÖLÜM 5. TARTIŞMA VE SONUÇ

Trafo merkezleri, günlük yaşantımızı her yönde etkileyen elektriğe ulaşmamız için önemli bir konumda yer almaktadır. Bu merkezlerde yaşanabilecek enerji kesintisi maddi ve güvenlik açısından birçok olumsuz sonuca yol açmaktadır. İşte bu sebeple; trafo merkezlerinin kontrol ve takibi oldukça önemlidir. Meydana gelen bir arızanın tespiti, bu arızaya yapılacak müdahalenin çeşidini de belirleyecek; kesici ve ayırıcı gibi ekipmanların kullanılma miktarı ile bakım zamanlarının belirlenmesini kolaylaştıracak ve redresör grubu gibi ekipmanların takibi ile meydana gelebilecek arızaların önüne geçilebilecektir.

Önerilen sistemde; trafo merkezinde bulunan sensör verileri değerlendirilerek kesicinin açtırılması ile birlikte sistemin enerjisiz kalma sebebi belirlenmiştir. Prototip ile birlikte bir arayüz programı geliştirilmiş ve sistemin uzak takibi ve kontrolü bu program ile sağlanmıştır. Elde edilen hatalar kayıt altına alınmış ve kullanılan MQTT Broker ile birlikte bu veriler bulutta da saklanmıştır. Saklanan verilerin daha sonra değerlendirilmesi bu şekilde sağlanmıştır. Bu verilerin değerlendirilmesi ile birlikte sistemde hangi hatanın daha çok geldiği belirlenebilecek ve buna göre iyileştirme yapılabilmektedir.

Trafo merkezlerinde bulunan redresör grubu, trafo merkezinin enerjisiz kalması ile birlikte kesici ve ayırıcı ünitelerinin kontrol edilmesini sağlamaktadır. Bu grubun gerilimi takip edilerek, redresör grubunda meydana gelecek bir arızanın önceden tespit edilmesi ile birlikte bu arızaya müdahale edilmesi arızanın önüne geçilmesini sağlamaktadır. Kesici ve ayırıcı ünitelerinin uzak kontrolünün bu sayede sürekli yapılabilmesi, ihtiyaç durumunda kesici ve ayırıcı manevralarının yapılabilmesine olanak vermektedir. Bu ünitelerin kontrolü ile birlikte birden fazla trafo merkezinden oluşan ring yapısında enerji aktarımı yapılabilmektedir.

Prototip yazılımının geliştirilmesinde Arduino IDE ve GZİS kullanılmıştır. Prototipin geliştirilmesinde kullanılan gerçek zamanlı işletim sistemi ARM, Intel, Xilinx mikroişlemcileri başta olmak üzere birçok işlemci mimarisini desteklemektedir. Bu nedenle prototipin diğer mimarilere aktarılmasına olanak sağlamaktadır. Böylece, yüksek kapasiteli mikroişlemcilerin kullanılabilmesi ile birlikte sistemin geliştirilmesi kolaylaşmıştır. Bununla birlikte; GZİS, sistemde yapılacak geliştirmelerin ve bu geliştirmelere ait çizelgelemenin kolaylıkla yapılmasına olanak tanımıştır.

Prototipin geliştirilmesinde kullanılan araçlar, önerilen sistemin gelişime açık olmasını ve farklı sistemlere adapte edilebilmesini sağlamıştır. Böylece, önerilen sisteme İnsan-Makine arayüzü gibi eklenebilecek özelliklerin sayısı artmıştır. Bu nedenle, sistemin güvenliğini ve izlenmesini arttıracak eklentiler yapılması oldukça kolaylaşmıştır. Akıllı lambalar gibi birçok nesnelerin interneti uygulamasıyla önerilen sistemin birlikte kullanılması da mümkündür.

Tasarlanan prototip, bir trafo merkezinde bulunan neredeyse her analog ve dijital girişler ile merkezin kontrolünde kullanılan dijital çıkışlar için uygundur. Trafo merkezindeki neredeyse tüm ihtimaller, prototipte ele alınmış; prototipte kullanılan kuru tip trafo ile kullanılmayan yağ seviye sensörü gibi girişler için de senaryoların uygulanabilir olduğu kanısına varılmıştır. Bunun yanı sıra, sisteme eklenebilecek özelliklerin prototiple gerçekleştirilmesi de mümkündür.

KAYNAKLAR

- [1] Wester C., Engelman N., Smith T., Odetunde K., Anderson B., ve Reilly J., The role of the SCADA RTU in today's substation, 2015 68th Annual Conference for Protective Relay Engineers, College Station, TX, USA, ss. 622-628, Mar. 2015.
- [2] Peharda D., Ivanković I., ve Jaman N., Using Data from SCADA for Centralized Transformer Monitoring Applications, Procedia Eng., c. 202, ss. 65-75, 2017.
- [3] Tom R. J. ve Sankaranarayanan S., IoT based SCADA integrated with Fog for power distribution automation, 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), ss. 1-4, Haz. 2017.
- [4] SCADA. <https://www.sedas.com/tr-tr/DagitimHizmetleri/Pages/Scada.aspx>
Erişim Tarihi: 14.05.2021.
- [5] How Kevin Ashton named The Internet of Things. <https://blog.avast.com/kevin-ashton-named-the-internet-of-things>.,
Erişim Tarihi: 28.12.2021.
- [6] Introduction to IoT. <https://www.cambridge.org/core/books/introduction-to-iot/7EDBB6CD17DA3E1F6FB5EBA54E0025A2>., Erişim Tarihi:27.12.2021.
- [7] Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., ve Ayyash M., Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Commun. Surv. Tutor., c. 17, sy 4, ss. 2347-2376, 2015.
- [8] Yang Z., Yue Y., Yang Y., Peng Y., Wang X., ve Liu W., Study and application on the architecture and key technologies for IOT, 2011 International Conference on Multimedia Technology, ss. 747-751, Tem. 2011.
- [9] Wu M., Lu T. J., Ling F. Y., Sun J., ve Du H., Research on the architecture of Internet of Things, c. 5, ss. V5-484, Eyl. 2010.
- [10] FreeRTOS Customisation. Developer Docs. https://www.freertos.org/a00110.html#configTICK_RATE_HZ,
Erişim Tarihi: 09.08.2022.

- [11] Kumar T. A. ve Ajitha A., Development of IOT based solution for monitoring and controlling of distribution transformers, 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), ss. 1457-1461, Tem. 2017.
- [12] Pathania S. ve Aggarwal S. K., IoT Based Distribution Transformer Health Monitoring System, Yüksek Lisans Tezi, 2019.
- [13] Mohamad A. A. H., Mezaal Y. S., ve Abdulkareem S. F., Computerized power transformer monitoring based on internet of things, Int. J. Eng. Technol., c. 7, sy 4, s. 2773, Eki. 2018.
- [14] Jamal H., Nadeem Khan M. F., Anjum A., ve Janjua M. K., Thermal Monitoring and Protection for Distribution Transformer Under Residential Loading Using Internet of Things, içinde 2018 IEEE Global Conference on Internet of Things (GCIoT), ss. 1-6, Ara. 2018.
- [15] Khandait A. P., Kadaskar S., ve Thakare G., Priyadarshini college of Engineering, Real Time Monitoring of Transformer using IOT, Int. J. Eng. Res., c. V6, sy 03, Mar. 2017.
- [16] Tarase K. B. ve Panchade V. M., Controlling of Substation Using IoT in Distribution Power Grid, 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), ss. 66-70, Mar. 2020.
- [17] Dimpal M. P., Mayank P., Pratik G., Pramod K., ve Kshitija D., Monitoring and Controlling of Transformer Using IOT, c. 6, sy 5, s. 7, 2020.
- [18] Kore P., Ambare V., Dalne A., Amane G., Kapse S., ve Bhavarkar S., IOT Based Distribution Transformer Monitoring and Controlling System, 2019.
- [19] Pawar R. R., Wagh P. A., ve Deosarkar S. B., Distribution transformer monitoring system using Internet of Things (IoT), 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, ss. 1-4, Haz. 2017.
- [20] Srivastava D. ve Tripathi M. M., Transformer Health Monitoring System Using Internet of Things, 2018 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, ss. 903-908, Eki. 2018.
- [21] Cheng X. ve Wang Y., The remote monitoring system of transformer fault based on The internet of Things, Proceedings of 2011 International Conference on Computer Science and Network Technology, Ara. 2011.
- [22] Emi P., Sivasankari R., Kumar P., Prabha R., Jayageetha J., ve Karhikeyan A., Fault Detection in Transformer Using GSM Technology, ss. 868-873, Mar. 2019.

- [23] Rahman S., Dey S. K., Bhawmick B. K., ve Das N. K., Design and implementation of real time transformer health monitoring system using GSM technology, 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), ss. 258-261, Şub. 2017.
- [24] Hasan W. K. A., Alraddad A., Ashour A., Ran Y., Alkelsh M. A., ve Ajele R. A. M., Design and Implementation Smart Transformer based on IoT, 2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE), London, United Kingdom, ss. 16-21, Ağu. 2019.
- [25] Duttachowdhury D., Patil V., Parab A., Patel R., ve Nikum K., Transformer Monitoring and Control Using Iot.
- [26] Hossain S., Rahman M., Sarker T., E. Haque E., ve Jahid A., A smart IoT based system for monitoring and controlling the sub-station equipment, Internet Things, c. 7, s. 100085, Eyl. 2019.
- [27] Khairnar V., Kolhe L., Bhagat S., Sahu R., Kumar A., ve Shaikh S., Industrial Automation of Process for Transformer Monitoring System Using IoT Analytics, Inventive Communication and Computational Technologies, Singapore, ss. 1191-1200, 2020.
- [28] Žarković M. ve Stojković Z., Analysis of artificial intelligence expert systems for power transformer condition monitoring and diagnostics, Electr. Power Syst. Res., c. 149, ss. 125-136, Ağu. 2017.
- [29] Zhao L., Matsuo I., Zhou Y., ve Lee W. J., Design of an Industrial IoT-Based Monitoring System for Power Substations, IEEE Trans. Ind. Appl., c. 55, sy 6, ss. 5666-5674, Kas. 2019.
- [30] Yaman O. ve Biçen Y., An Internet of Things (IoT) Based Monitoring System for Oil-immersed Transformers, Balk. J. Electr. Comput. Eng., c. 7, sy 3, Art. sy 3, Tem. 2019.
- [31] Thangiah L., Ramanathan C., ve Chodisetty L. S., Distribution Transformer Condition Monitoring Based on Edge Intelligence for Industrial IoT, içinde 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), ss. 733-736, Nis. 2019.
- [32] John A., Varghese R., Krishnan S. S., Thomas S., Swayambu T. A., ve Thasneem P., Automation of 11 KV Substation Using Raspberry Pi, 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), ss. 1-5, Nis. 2017.
- [33] Mariprasath T. ve Kirubakaran V., A real time study on condition monitoring of distribution transformer using thermal imager, Infrared Phys. Technol., c. 90, ss. 78-86, May. 2018.

- [34] Sohraby K., Minoli D., Occhiogrosso B., ve Wang W., A Review of Wireless and Satellite-Based M2M/IoT Services in Support of Smart Grids, *Mob. Netw. Appl.*, c. 23, sy 4, ss. 881-895, Ağu. 2018.
- [35] Wang J. H., Li X. Z., Lu R. T., ve Sun F. J., Internet of Things Based Auxiliary Control System for Smart Substation, *Adv. Mater. Res.*, c. 614-615, ss. 1790-1795, 2013.
- [36] Zhang C., He Y., Du B., Yuan L., Li B., ve Jiang S., Transformer fault diagnosis method using IoT based monitoring system and ensemble machine learning, *Future Gener. Comput. Syst.*, c. 108, ss. 533-545, Tem. 2020.
- [37] WiFi SoC Manufacturers for IoT | WiFi SoC Vendors. <https://www.rfwireless-world.com/Vendors/WiFi-SoC-manufacturers.html>., Erişim Tarihi: 11.04.2022.
- [38] Language Bindings - Qt Wiki. https://wiki.qt.io/Language_Bindings., Erişim Tarihi: 12.04.2022.
- [39] Supported Platforms | Qt 5.15. <https://doc.qt.io/qt-5/supported-platforms.html> Erişim Tarihi: 12.04.2022.
- [40] Supported Platforms | Qt 6.2. <https://doc.qt.io/qt-6.2/supported-platforms.html> Erişim Tarihi: 12.04.2022.
- [41] Qt for Python - Qt Wiki. https://wiki.qt.io/Qt_for_Python., Erişim Tarihi: 12.04.2022.
- [42] <http://www.openrtos.net/RTOS.html>., Erişim Tarihi: 13.04.2022.
- [43] Microcontrollers and compiler tool chains supported by FreeRTOS, https://www.freertos.org/RTOS_ports.html., Erişim Tarihi: 13.04.2022.
- [44] Synchronization of current and voltage measurements in a modular motor diagnostic system, https://www.researchgate.net/figure/Topology-of-NTP-server-structure_fig4_259267101, Erişim Tarihi: 16.05.2022
- [45] Ed Williams, Sunrise/Sunset Algorithm, *Almanac for Computers*, 1990, https://edwilliams.org/sunrise_sunset_algorithm.htm., Erişim Tarihi: 16.04.2022.
- [46] <https://sunrise-sunset.org/>., Erişim Tarihi: 21.04.2022.

ÖZGEÇMİŞ

Adı Soyadı : Sercan AYGÜN

ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar ve Bilişim Mühendisliği	2018-Halen
Lisans	Atatürk Üniversitesi / Mühendislik Fakültesi / Elektrik ve Elektronik Mühendisliği	2014
Lise	Van Milli Piyango Anadolu Lisesi	2010

İŞ DENEYİMİ

Yıl	Yer	Görev
2020-Halen	TÜRASAŞ Sakarya Bölge Md.	Saha Mühendisi
2017-2020	Türkiye Vagon Sanayi A.Ş.	Saha Mühendisi
2016-2017	Tekin Elektrik	Şantiye Şefi

YABANCI DİL

İngilizce

ESERLER (makale, bildiri, proje vb.)

1. S. Aygün, A. Sevin, Trafo Merkezleri İçin Nesnelerin İnterneti Tabanlı Kontrol Sistemi Tasarımı ve Uygulaması, International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Eki. 2021. DOI: 10.31590/ejosat.1010748.