

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GELİŞMİŞ MİKRODENETLEYİCİ KONTROLLÜ AKILLI EV
OTOMASYONU

Yunus Emre FEDAR

YÜKSEK LİSANS TEZİ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Programı

Danışman

Prof. Dr. Herman SEDEF

Temmuz, 2019

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**GELİŞMİŞ MİKRODENETLEYİCİ KONTROLLÜ AKILLI EV
OTOMASYONU**

Yunus Emre FEDAR tarafından hazırlanan tez çalışması 16.07.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Elektronik Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof.Dr. Herman SEDEF

Yıldız Teknik Üniversitesi

Danışman

Jüri Üyeleri

Prof.Dr. Herman SEDEF, Danışman

Yıldız Teknik Üniversitesi

Prof.Dr. Mehmet SAĞBAŞ, Üye

İstanbul Yenyüz yıl Üniversitesi

Doç.Dr. Umut Engin AYTEN, Üye

Yıldız Teknik Üniversitesi

Danışmanım Prof. Dr. Herman SEDEF sorumluluğunda tarafımca hazırlanan Gelişmiş Mikrodenetleyici Kontrollü Akıllı Ev Otomasyonu başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim

Yunus Emre FEDAR

İmza



Aileme

ve

değerli milletimize

TEŞEKKÜR

Başta saygıdeğer danışman hocam Prof. Dr. Herman SEDEF olmak üzere, aileme, kardeşlerime, amcama ve bana bu süreçte manevi desteklerini esirgemeyen herkese teşekkürü bir borç biliyorum.

Yunus Emre FEDAR



İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	viii
ŞEKİL LİSTESİ	x
TABLO LİSTESİ	xii
ÖZET	xiii
ABSTRACT	xv
1 Giriş	1
1.1 Literatür Taraması.....	1
1.2 Tezin Amacı	3
1.3 Hipotez.....	3
2 Sistemin Teknolojik Altyapısı, Haberleşme Paket Yapısı ve Şifrelenmesi	4
2.1 Tasarlanan Ev Otomasyonunun Fikri Altyapısı	4
2.2 Kullanılan Gelişmiş Mikrodenetleyici ve Arayüzleri.....	5
2.3 nRF24L01+ GFSK Haberleşme Modülü	9
2.4 LPT-100 Wi-Fi Modülü	12
2.5 Tasarlanan Özgün Haberleşme Paket Yapısı.....	15
2.6 Verilerin AES-128bit Şifreleme ile Korunması.....	19
3 Sistemin Anakartı: Sunucu Cihazı	21
3.1 Donanım Bazında Tasarım Yaklaşımı	21
3.2 Yazılım Yaklaşımı	25
4 Kontrol Cihazları	28
4.1 Dimmer Devresi.....	28
4.2 Doğalgaz Kontrol Devresi	34
4.3 Kapı Kontrol Devresi	37
4.4 Sıcaklık-Nem Kontrol Devresi.....	39

4.5	Ütü Kontrol Devresi	41
4.6	Kombi Kontrol Devresi	44
4.7	Klima Kontrol Devresi.....	46
5	Android Kontrol Yazılımı ve Sistem Davranışı	49
5.1	Android Yazılım Geliştirme Arayüzü QTCreator	49
5.2	Akıllı Ev Otomasyonu Android Yazılımı	49
6	Sonuç ve Öneriler	52
	Ek-1 Cihazların Mikrodenetleyici Kodları	55
	Ek-2 Android Yazılımının Kodları	100
	Ek-3 Akıllı Ev Otomasyonu için Maliyet Analizi	120
	Kaynakça	126
	Tezden Üretilmiş Yayınlar	128

SİMGE LİSTESİ

$^{\circ}C$ Santigrat Derece

$\%H$ Nem

Ω Direnç(Ohm)



KISALTMA LİSTESİ

AC	Alternating Current
ADC	Analog Digital Converter
AES	Advanced Encryption Standard
ARM	Advanced RISC Machine
CPU	Central Process Unit
CRC	Cyclic Redundancy Check
DC	Direct Current
DDOS	Distributed Denial of Service
GFSK	Gaussian Frequency Shift Keying
GHz	Giga Hertz
GPRS	General Packet Radio Service
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IRQ	Interrupt Request
kbps	Kilobit per second
KB	Kilo Byte
LED	Light Emitting Diode
MB	Mega Byte
MCU	Microcontroller Central Unit
MHz	Mega Hertz
MIPS	Million Instruction Per Second
mV	millivolt

ns	Nano saniye
PCB	Printed Circuit Board
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
RS232	Recommended Standard 232(EIA232)
RX	Receive Process
SPI	Serial Peripheral Interface
TX	Transmit Process
UART	Universal Asynchronous Reciever/Transmitter
UPnP	Universal Plug and Play
USB	Universal Serial Bus
V	Volt
W	Watt
Wi-Fi	Wireless Fidelit

ŞEKİL LİSTESİ

Şekil 1.1 “Kitchen Computer” adlı ilk ev otomasyonu.....	2
Şekil 2.1 Sistemin Genel Şeması	5
Şekil 2.2 ARM işlemci görseli.....	6
Şekil 2.3 STM32L052C6 işlemcisi Özellik Tablosu.....	7
Şekil 2.4 STM32L052C6 işlemcisinin Sunucu devresinde kullanılmak üzere STM32CUBEMX yazılımında tasarımı	8
Şekil 2.5 STM32F030F4 işlemcisinin Doğalgaz Kaçak Kontrol devresinde kullanılmak üzere STM32CUBEMX yazılımında tasarımı.....	8
Şekil 2.6 nRF24L01+ modülü	10
Şekil 2.7 nRF24L01’in aynı frekans üzerinde adreslenmesi	11
Şekil 2.8 LPT-100 Wi-Fi modülü anteni ile birlikte.....	12
Şekil 2.9 LPT-100 Wi-Fi modülünün çalışma ağı	13
Şekil 2.11 Tasarlanan Özgün Paket Yapısının Bilgi Dağılımı	15
Şekil 2.12 Paket Yapısı Emir Uygulama Akış Diyagramı.....	19
Şekil 3.1 Sunucu Devresinin Altiumda Çizimi	22
Şekil 3.2 Sunucu Devresinin Altium’da PCB çizimi görüntüsü.....	24
Şekil 3.3 Sunucu Devresinin PCB üreten firmaya bastırılmış hali	24
Şekil 3.4 Sunucu Cihazı.....	25
Şekil 4.1 Dimmer Sıfır Geçişi Devresi.....	28
Şekil 4.2 Opamp Giriş Bacağındaki Sinüzoidal Sinyalden Sıfır Geçişi Elde Edilmesi.....	29
Şekil 4.3 Trafodan Alınan 12VAC Sinyali ile Sıfır Geçişinin Karşılaştırılması.....	30
Şekil 4.5 Dimmer Kontrol Cihazı Akış Diyagramı	31
Şekil 4.6 Dimmer Devresi PCB Çizim Görseli.....	32
Şekil 4.7 Dimmer Devresi Bastırılmış PCB boş hali (alttan görünüm)	32
Şekil 4.8 Dimmer Kontrol Devresi son hali (üstten görünüm)	33
Şekil 4.9 TORK Selenoid Doğalgaz Valfi Normalde Açık ½.....	34
Şekil 4.11 Doğalgaz Kontrol Devresinin Akış Diyagramı.....	35
Şekil 4.12 Doğalgaz Kontrol Devresinin PCB çizim görseli	35

Şekil 4.13 Doğalgaz Kontrol Devresi Bastırılmış PCB boş hali (üstten görünüm) .	36
Şekil 4.14 Doğalgaz Kontrol Devresi son hali (üstten görünüm)	36
Şekil 4.16 Kapı Kontrol Devresi Şeması.....	38
Şekil 4.17 Kapı Kontrol Devresi son hali (üstten görünüm)	38
Şekil 4.18 Sıcaklık-Nem Kontrol Devresi son hali (üstten görünüm)	39
Şekil 4.19 Sıcaklık-Nem Devresi Akış Diyagramı	40
Şekil 4.20 Ütü Kontrol Devresinin resmi.....	41
Şekil 4.21 Ütü Kontrol Devresinin resmi.....	42
Şekil 4.22 Ütü Devresinin Altium'da çizilmiş hali.....	42
Şekil 4.23 Ütü Devresinin PCB'sinin bastırılmış hali.....	43
Şekil 4.25 Ütü Kontrol Devresi Akış Diyagramı	44
Şekil 4.26 Kombi Kontrol Devresi	45
Şekil 4.28 Kombi Kontrol Devresi Akış Diyagramı.....	46
Şekil 4.29 Klima Kontrol Devresi	47
Şekil 4.30 Klima Kontrol Devresinin genel şeması	47
Şekil 4.31 Klima Kontrol Devresi Akış Diyagramı	48
Şekil 5.1 QTCreator Arayüzü.....	49
Şekil 5.2 Ev Otomasyonu Uygulaması Görseli	50
Şekil 5.3 Kontrol Kutuları.....	51
Şekil 6.1 Projedeki bütün cihazların toplu resmi.....	54

TABLO LİSTESİ

Tablo 2.1 Modül tipleri ve karşılık gelen numaraları.....	16
Tablo 2.2 Parametreler ve karşılık gelen nümerik değerleri.....	16



Gelişmiş Mikrodenetleyici Kontrollü Akıllı Ev Otomasyonu

Yunus Emre FEDAR

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Herman SEDEF

İnsanlar sürekli olarak hayat standartlarını yükseltme eğilimindedirler. Teknoloji ile beraber bu beklentiler daha da yükselmektedir. İnsanlar kendi yaşam alanlarında konfor ve emniyet ihtiyacını daha çok hissetmektedirler. Günümüz teknolojisi ve Nesnelerin İnterneti (Internet of Things, IoT) bu konforu ve emniyeti sağlamak yolunda hızla ilerlemektedir. Gelişmiş ev otomasyonları bu ilerlemenin bir ürünüdür. Bu çalışmada, gelişmiş ARM çekirdekli mikrodenetleyiciler ile özgün bir ev otomasyonu sisteminin hem donanım hem yazılım olarak tasarımı, testleri ve uygulaması yapılmıştır. Sistem tamamen kablosuz haberleşme üzerine tasarlanmış olup, 2.4 GHz Wi-Fi ve 2.4 GHz RF (Radyo Frekansı) haberleşme modülleri ile gerçekleştirilmiştir. Android tabanlı cihazlarda çalışan ve bu otomasyon sistemini kontrol eden bir arayüz uygulama geliştirilmiştir. Tasarlanan bütün cihazların birbirleri ile anlaşmasını sağlayan, haberleşmenin her safhasında kullanılan yeni bir haberleşme paket yapısı sunulmuştur. Olası güvenlik açıklarına karşı, tüm

haberleşmede AES-128bit şifreleme kullanılmıştır. Sistemin artıları ve eksileri günümüz koşullarına göre tartışılmıştır.

Anahtar Kelimeler: Otomasyon, kontrol, mikrodenetleyici, kablosuz haberleşme



ADVANCED MICROCONTROLLER CONTROLLED SMART HOME AUTOMATION

Yunus Emre FEDAR

Department of Electronics and Communication Engineering

Master Thesis

Advisor: Prof. Dr. Herman SEDEF

People tend to constantly raise their standard of living. With advanced technology, these expectations are increasing. People feel the need for comfort and safety in their living spaces. Today's technology and the Internet of Things (IoT) are advancing rapidly to provide this comfort and safety. Advanced home automation is a product of this progress. In this study, with usage of advanced ARM core microcontrollers, the design, testing as well as hardware and software of an original home automation system were performed. The system is designed for wireless communication with 2.4 GHz Wi-Fi and 2.4 GHz RF (Radio Frequency) communication modules. An interface application that works on Android based devices and controls this automation system has been developed. A new communication package structure is provided, which enables all designed devices to communicate with each other and which is used at every stage of communication. Against possible security vulnerabilities, AES-128bit encryption is used in all

communications. The pros and cons of the system are discussed according to current conditions.

Keywords: Automation, control, microcontroller, wireless communication



1.1 Literatür Taraması

Ev otomasyonları konforu arttırmak, tasarrufu sağlamak ve daha fantastik yaşam hedeflerine ulaşmak amacıyla birçok insanın hayallerini süslemekte, bilim kurgu filmlerine konu olmakta ve araştırmacıların ilgisini çekmektedir.

Bu sebeple günümüzden geçmişe bu konuda birçok araştırma yapılmıştır. İlk ev otomasyonları mekanik ve elektrik aksamlarından oluşmaktaydı ve sınırlı sayıda işlem yapabilmekteydi. İlk ev otomasyonları 1967 yılında geliştirilen “ECHO IV” ve ardından geliştirilen “Kitchen Computer” isimli bir mutfak cihazı idi [1]. Bu cihazlar satın alma listeleri oluşturabiliyor, evin sıcaklığını kontrol edebiliyor ve araç-gereçleri açıp kapatabiliyordu.

Günümüz ev otomasyonları ise ağırlıklı olarak güvenlik, tasarruf ve konfor konuları üzerine yoğunlaşarak bu sektörü geliştirmiş ve geliştirmeye devam etmektedir. Bu süre zarfında elektronik bilim dalı bakış açısıyla literatürde neler yapıldığına bakıldığında; tek bir kontrol ünitesi/devresi üzerinden bütün sistemlerin kontrol edildiği, 8-bit işlemcinin analog ve dijital giriş çıkışlarının istenilen şekilde ayarlanıp kullanılabildiği ev otomasyonları tasarlanmıştır [2]. Evdeki cihazlara ek bir kablolama gerekmeden erişebilmek için PLCC (Power Line Carrier Communication) yani şebeke güç hattı üzerinden verilerin iletildiği, X10 protokolünü içeren sistemler kullanılmıştır [3]. SMS üzerinden bilgisayara komut gönderilen, bilgisayarında web server yazılımı ile analiz edilen verinin, bazı kısımlarında X10 protokolünde kullanılarak, kontrol rölelerine parallel port üzerinden veri akışı sağladığı sistemler geliştirilmiştir [4].

kütüphanelerine ve devrelerine bağımlılığı getirmekle beraber özgün bir yaklaşım olmamaktaydı. Sunulan sistem Ethernet shield, yani Ethernet arayüz kartı üzerinden bu bağlantıyı sağlamakta idi, bu sistemde hazır bir entegrasyonun parçası olmakla beraber, Ar-Ge maliyetlerini ciddi manada düşürmekte idi. Başka bir araştırmada [11] TCP/IP üzerinden Ethernet katmanında Arduino, Raspberry Pi ve Intel Galileo karşılaştırılması yapılmış, kriptografi methodları ve haberleşme algoritmaları bu cihazlarla gerçekleştirilerek performans testleri uygulanmış ve IoT(Internet of Things) ile entegre edilebilirliği araştırılmıştır, bu bir ev otomasyonu araştırması değildir lakin ev otomasyonlarının teknolojik altyapısına katkı sağladığı düşünülebilir. Öte yandan sesli komut ile çalışan Google Ev Asistanı [18] ve Mi Ev Asistanı[19] gelişmiş ev otomasyonlarına örnek verilebilir. Ülkemizde akıllı ev otomasyonunu tasarlayan firmalarda vardır. Bu firmalar dimmer, aç-kapa işlevleri, stor perde kontrolleri, belirli tarih ve zamanda komutlar verebilecek şekilde ayarlanan fonksiyonların girilebilmesi ve senaryolar oluşturulabilmesi bakımından pazarda yerlerini almışlardır.

1.2 Tezin Amacı

Bu çalışmanın amacı, Android işletim sisteminden kontrol edilebilen, gelişmiş ARM mikrodenetleyicili, özgün, güvenli ve ekonomik bir akıllı ev otomasyonu sistemi geliştirmektir.

1.3 Hipotez

Günlük yaşantımızı kolaylaştıran akıllı ev otomasyonu sistemleri gelişmiş mikrodenetleyicilerle (ARM Cortex M0+) tasarlanıp, Android işletim sistemi üzerinden, tamamen kablosuz olacak şekilde (2.4GHz GFSK ve Wi-Fi), AES-128bit şifreli haberleşme ile kontrol edilebilir.

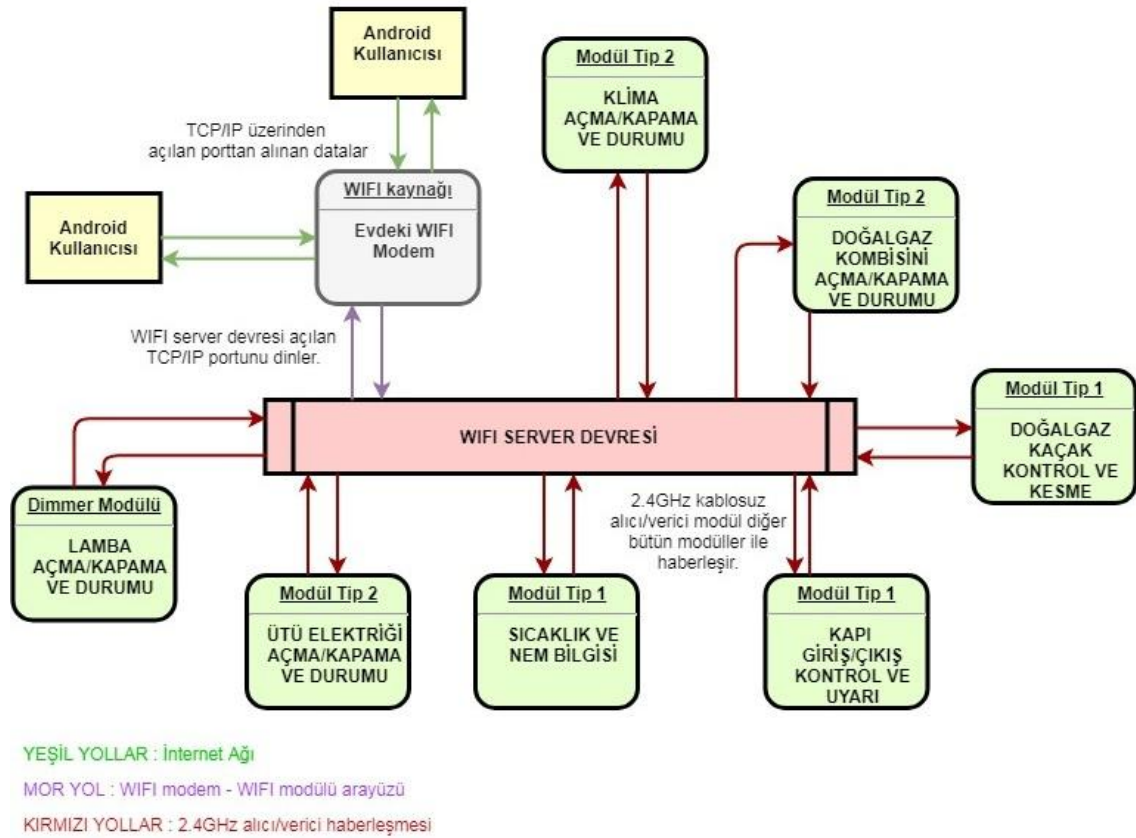
Sistemin Teknolojik Altyapısı, Haberleşme Paket Yapısı ve Şifrelenmesi

2.1 Tasarlanan Ev Otomasyonunun Fikri Altyapısı

Sunulan bu sistem tümüyle kablosuz haberleşme üzerine olup, en güncel 32-bit ARM işlemci teknolojisini içermektedir. Tüm donanım ve yazılım Ar-Ge'si tamamen özgündür ve asla taklit eseri değildir. Tasarım aşamasında geliştirme kitleri üzerinde gerekli testler yapılmış olup, ihtiyaç olan PCB'ler Altium'da tasarlanarak bastırılmıştır. Özgün olan bu sisteme ayrıca özgün siber anahtarlı bir kriptolu haberleşme paket yapısı tasarlanmıştır, bölümünde ayrıntılı olarak anlatılacaktır.

İnternet dünyasında güncel olarak kullanılan AES-128bit, haberleşme verisini şifrelemede kullanılmıştır. Kontrol cihazları arası 2.4GHz haberleşme bandına sahip modül tercih edilmiştir (maliyet düşürme amaçlı seçilmiştir). Donanımsal CRC hata düzeltme protokolü mevcuttur ve veri stabilitesini kontrol eden GFSK haberleşme modüllerinin kullanıldığı, anakarttan internete açılma işlevinde kablolu Ethernet katmanından değil, entegresi kablosuz ve kolay Wi-Fi modül yapısı ve Wi-Fi TCP/IP soket haberleşmesi kullanılmıştır. Wi-Fi modül, Access Point veya Client olarak kullanılabilir, Server olarak kullanımında bağlanılıp Wi-Fi modülün tüm ayarlamaları yapılabilir, client modunda çalışırken sistemin bir parçası olarak haberleşme görevini yerine getirmektedir. Wi-Fi tasarlanan sistemin internetle olan arayüzünde karmaşıklığını azaltmakla beraber anakart cihazının kablolarla olan bağlılığını bitirmektedir. Sistem dışarıdan sızmaya karşı güvenli olup 2.4 GHz seviyesinde herhangi bir Wi-Fi bandı seçildiği için dinlenilmesinin neredeyse imkansız olduğu, Wi-Fi kısmında soket enjeksiyonunun ise engellendiği bir sistem yapısı oluşturulmuştur, enjeksiyon başarılı olsa bile evin içinde zararlı değişiklikler yapılamayacaktır. DDoS koruması kısmına, evde bulunan ticari modemlerin sahip olduğu DDoS protection yapısından dolayı girilmemiştir. Wi-Fi üzerinden olmasının bir iyi özelliği de şudur ki, evin içinde ticari modem açık olmaksızın, daha önceden anakarta SSID'si ve şifresi girilmiş olan bir telefon

hotspotuna da ev otomasyonu sistemi bağlanabilmekte ve haberleşme aynen devam edebilmektedir. Evin içindeki ticari modem üzerinden port yönlendirmesi yapılarak ev otomasyonunun kontrolü ev dışına, mobil veriye sahip bir android cihaza da taşınabilecektir. Bu sistemin her noktadan kontrol edilebilirliği anlamına gelmektedir. Burada dikkatleri çeken bir diğer husus Android cihazına yüklenen yazılımda bulunan güvenlidir. Android yazılımı bağlı bulunduğu cihazın MAC adresini anakarta göndermekte, anakart üzerinde bulunan tanımlı MAC adresleri dışında erişim sağlanamamaktadır. Bu cihazın güvenliğini kat kat arttırmaktadır. Ayrıca gelen bu sızma amaçlı sorgunun sahibi olan MAC adresi depolanmakta ve log kayıtları tutulmaktadır.

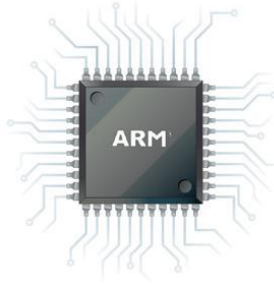


Şekil 2.1 Sistemin Genel Şeması

2.2 Kullanılan Gelişmiş Mikrodenetleyici ve Arayüzleri

ARM, Advanced RISC (azaltılmış komut seti bilgisayarı) makinesi anlamına gelmektedir. ARM, Acorn BCC bilgisayar üreticilerinin bir parçası olarak hayata

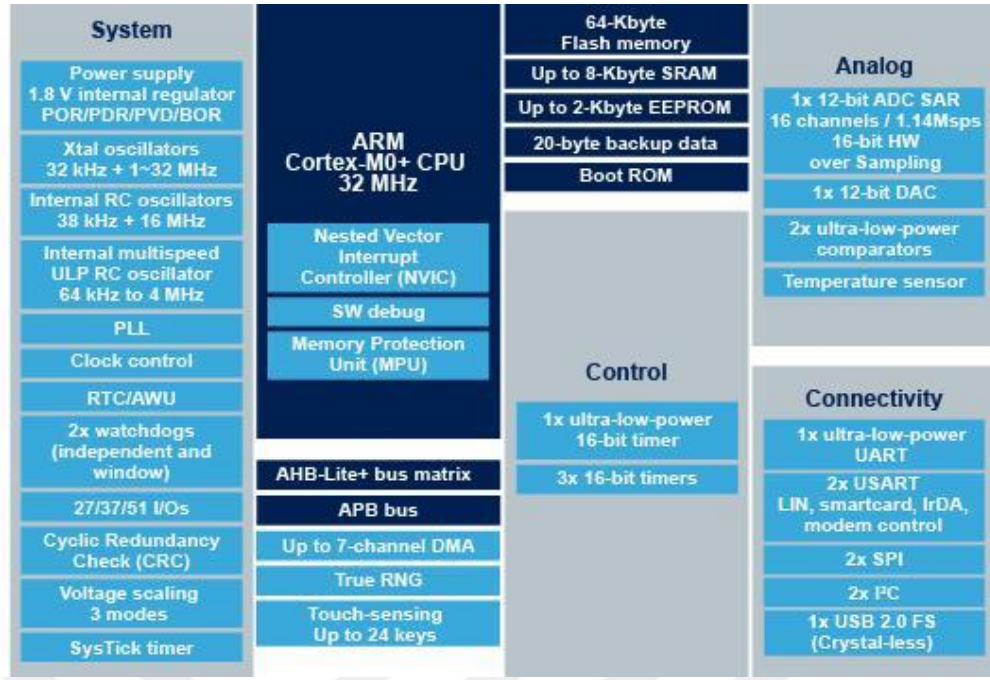
başlamıştır ve şuan apple iPad için çip tasarlamaktadır. İlk olarak ARM 1978'de Cambridge Üniversitesi'nde kurulmuştur. Acorn grup bilgisayarları 1985'te ilk ARM ticari RISC işlemcisini geliştirmiştir. ARM, kurulduktan sonra 1990'da çok popüler olmuş, 2008 yılında 10 milyar işlemci postalanmıştır. ARM, mikrodenetleyici ve mikroişlemcilerin yerine geçmekte olan son teknolojidir. Temelde ARM, 16 bit / 32 bit işlemciler veya kontrolörlerdir. ARM, cep telefonları otomotiv sistemleri dijital kameraları ve ev ağı ve kablosuz teknolojiler gibi gelişmiş dijital ürünlerin vazgeçilmezlerden olmuştur [12].



Şekil 2.2 ARM işlemci görseli

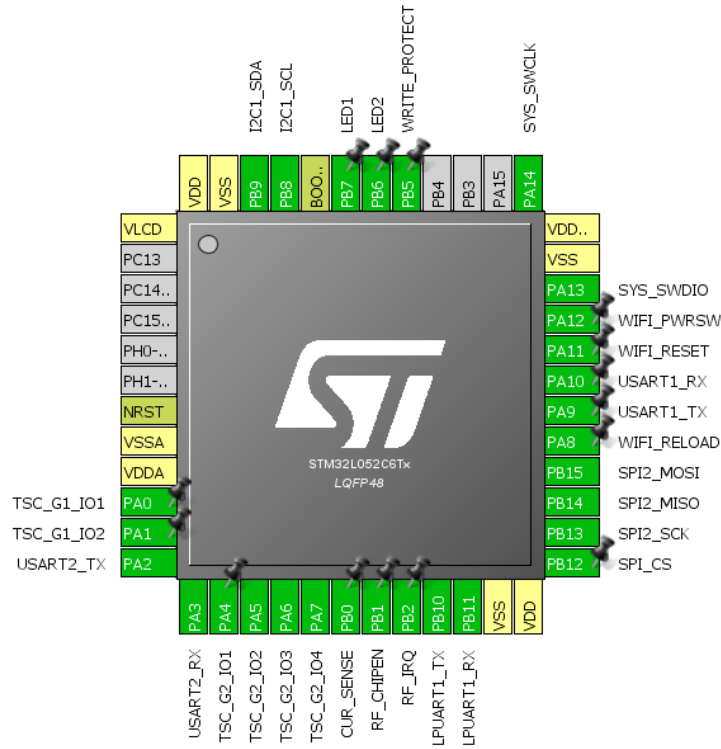
ARM işlemcilerin yaygın olmasının sebeplerinden biri; düşük güç tüketimi ve makul performansı nedeniyle özellikle taşınabilir cihazlarda kullanılan en popüler işlemciler olma özelliğidir. İkinci olarak, ARM diğer işlemcilerle karşılaştırıldığında daha iyi bir performansa sahiptir. ARM işlemci temel olarak düşük güç tüketimi ve düşük maliyet felsefesinden oluşmuştur. ARM'ı hızlı ve verimli uygulama geliştirmeleri için kullanmak çok kolaydır, bu yüzden ARM'ın şuanda en popüler olmasının ana nedeni budur [12].

Sistemimizde kullandığımız STM32L052C6 ve STM32F030F4 işlemcileri 32-bit ARM işlemcilerinin ST Microelectronics firması tarafından özelleştirilmiş halidir.



Şekil 2.3 STM32L052C6 işlemcisi Özellik Tablosu

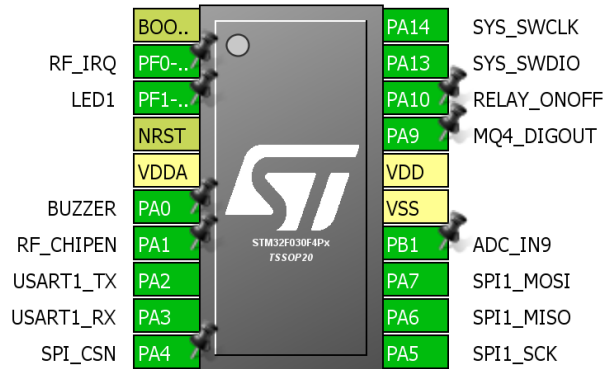
STM32L052C6 işlemcisi max. 32MHz'de çalışan ARM Cortex M0+ çekirdeğe sahiptir. Sunucu devresinde ve Dimmer devresinde kullanılmıştır. Sunucu devresinde kullanılmasının başlıca sebepleri; 64KB Flash hafızaya sahip olması, 37 G/Ç biriminin olması, üç tane UART arayüzüne sahip olması ki sunucu devresinde biri konsol arayüzü için, ikincisi Wi-Fi modül UART üzerinden haberleştiği için, üçüncüsü de Sunucu devresine Nextion LCD ekran bağlanabildiği ve LCD ekran UART üzerinden haberleştiği için. İki tane SPI çıkışı var lakin bir tanesini GFSK modül için kullanılıyor. I2C çıkışına Sıcaklık-Nem modülü eklenebilecek şekilde ayarlandı. İşlemcinin fonksiyonlarından faydalanma noktasında, STMicroelectronics firmasının sunduğu STM32CUBEMX yazılımı ile işlemci ayarlarının kolayca yapılandırılması ve C koduna yansıtılması kolayca yapılabilmektedir.



Şekil 2.4 STM32L052C6 işlemcisinin Sunucu devresinde kullanılmak üzere STM32CUBEMX yazılımında tasarımı

STM32F030F4 işlemciside max. 32MHz'de çalışan ARM Cortex M0+ çekirdeğe sahiptir. Diğer bütün kontrol cihazlarında kullanılmıştır. Konsol için 1 adet UART arayüzü, nRF24L01+ GFSK modül için bir SPI arayüzü ve diğer fonksiyonları görecektir kapasitesi bulunduğu ve ucuz olduğu için tercih edilmiştir.

İşlemcinin STM32CUBEMX'teki görüntüsü şekildeki gibidir.



Şekil 2.5 STM32F030F4 işlemcisinin Doğalgaz Kaçak Kontrol devresinde kullanılmak üzere STM32CUBEMX yazılımında tasarımı

İşlemcilerimiz maksimum saat frekansı olan 32MHz'e değil, 24MHz'e ayarlanmıştır. Bunun sebebi işlemciyi maksimum kapasitede kullanmadan, yormayıp ömrünü uzatmak ve gereksiz yere işlemcinin yüksek güç harcamasına engel olmaktır.

32-bit ve ARM işlemciler gelişmiş işlemcilerdir, düşük güç tüketimleri, yüksek hızlarda çalışabilmeleri ve aynı zamanda çok fazla fonksiyon ve arayüz içermeleri onları ciddi anlamda ön plana çıkarmaktadır ve bugün güncel olarak cep telefonlarından, biyomedikal ve özel cihazlara elektronik ile ilgilenen tüm kurumların en çok rağbet ettiği işlemcilerdir. Çok fonksiyonlu oluşu işi karmaşıktırmakla beraber STMicroelectronics gibi firmalar sunduğu geliştirme yazılımları ile mühendislere büyük kolaylıklar sunmaktadır.

İşlemci kodlama arayüzü olarak System Workbench for STM32 (SW4STM32) arayüzü kullanılmıştır. Bu arayüz Eclipse IDE'nin modlanmış halidir ve Eclipse'in C/C++ tasarım arayüzünde tasarım yapılmaktadır.

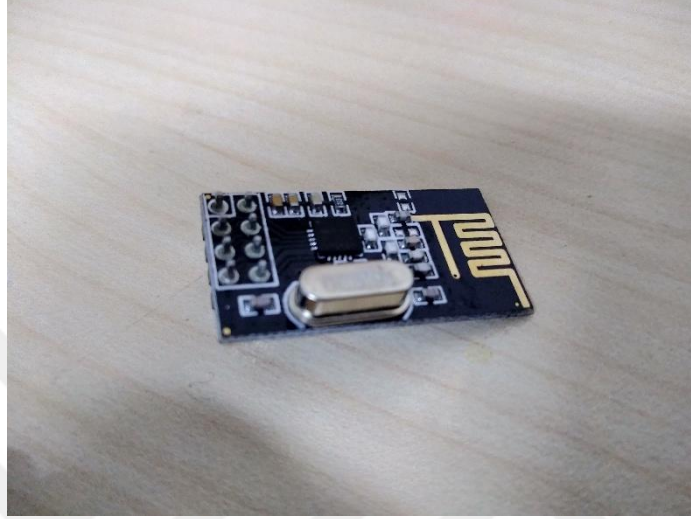
STM32CUBEMX işlemci çevresel birimleri tasarım yazılımı, SW4STM32 ile entegredir ve görsel olarak tasarlanan işlemci çevresel birimlerini kodlara dönüştürür. Kodlar ARM kütüphanelerini içermekte ve HAL (Hardware Abstraction Layer) kodları kullanmaktadır. Böylece kütüphanelerin entegrasi için gereken zamandan ciddi tasarruf sağlanmakta, mühendise veya tasarımcıya ciddi kolaylıklar sağlamaktadır.

Tasarlanan bu kodlar ST-link_V2 ile mikrodnetleyicilere yüklenmektedir. ST-Link_V2 ise kendi bir cihaz olarak mevcut olduğu gibi bu tasarımda aktif olarak kullanılan NUCLEO-L053R8 geliştirme kitinin içinde de mevcuttur. Hem geliştirme kiti hemde programlayıcı olarak kullanılabilen NUCLEO-L053R8 bu konuda büyük kolaylık sağlamaktadır. Mikrodnetleyicileri kodlamak için ekstra bir cihaz almak gerekmemektedir. Üzerinde işlemciside olan bu geliştirme kitiyle hızlı testler yapılabilmektedir. Örneğin; özellikle belirtmek gerekir ki, SPI arayüzünü kullanan nRF24L01+ GFSK haberleşme modülünün testlerinde ciddi olarak bu geliştirme kitinden faydalanılmıştır.

2.3 nRF24L01+ GFSK Haberleşme Modülü

Nordic Semiconductor üreticisinin bir ürünü olan nRF24L01+, ultra düşük güçlü kablosuz uygulamalar için tasarlanmış gömülü bir temel bant protokol motoruna

(Enhanced ShockBurst™) sahip tek bir yonga 2.4GHz alıcı-vericidir. nRF24L01+, dünya çapında ISM frekans bandında 2.400 - 2.4835GHz'de çalışabilecek şekilde tasarlanmıştır. NRF24L01 ile bir radyo sistemi tasarlamak için bir MCU (mikrodenetleyici) ve çok az harici pasif bileşene ihtiyaç vardır [13]. Oldukça ucuz olan bu modül günümüzde bir tanesinin fiyatı \$1 altında satılmaktadır.



Şekil 2.6 nRF24L01+ modülü

Yapılan testler sonucunda nRF24L01+ modülünün sistemde kullanılabileceği öngörülmüştür. nRF24L01+ modülünü kullanmak için gerekli kütüphane araştırmaları yapılmış ve github [14]'dan faydalanılarak gerekli kütüphaneler elde edilmiş ve bu genel kütüphanelerin çalıştığımız mikrodenetleyiciye entegresine ciddi çaba sarfedilmiş yüzlerce test yapılmıştır.

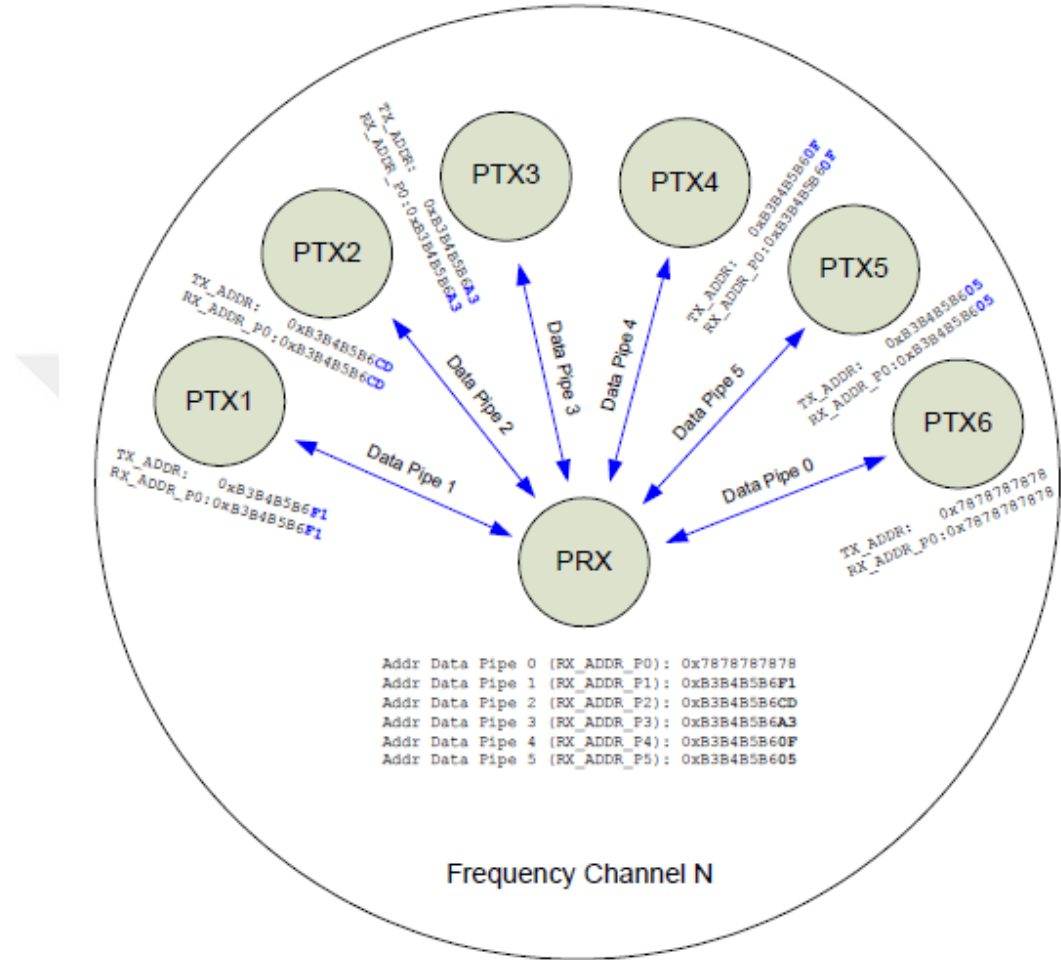
nRF24L01+ kendi içinde donanımsal olarak CRC özelliğine sahiptir ve bu özellik 2 byte doğrulama şeklinde kullanılmaktadır.

Modül 2400MHz ile 2550MHz arasında 127 tane farklı frekans kanalına sahiptir ve istenilen kanal kullanılabilir. Tasarımımızda 2520MHz frekans bandı kullanılmıştır. Bunun sebebi 2.4GHz Wi-Fi bandının gürültüsünden bir miktar uzaklaşmaktır. Burada dikkat edilmesi gereken husus modülün Wi-Fi protokolüyle ortak bantlarının bulunmasıdır. 2520MHz gibi daha yüksek bir bant seçilerek genel Wi-Fi frekanslarından uzaklaşmak amaçlanmıştır.

Modül 250kbps, 1Mbps, 2Mbps gibi haberleşme hızlarına sahiptir. Haberleşme hızı olarak projemizde 250kbps seçilmiştir. Datasheet'i incelediğimizde düşük haberleşme hızının güvenilirliği arttırdığı ve kayıp paket sayısını azalttığından

bahsedilmiştir. 32byte'lık bir haberleşme paket yapısına sahip sistemimiz için hız önemli değildir ve paket güvenilirliği esas alınmıştır.

nRF24L01+ paketleri adresleyebilme özelliğine sahiptir. 6 farklı adresten gelen paketleri aynı anda dinleyebilmektedir.

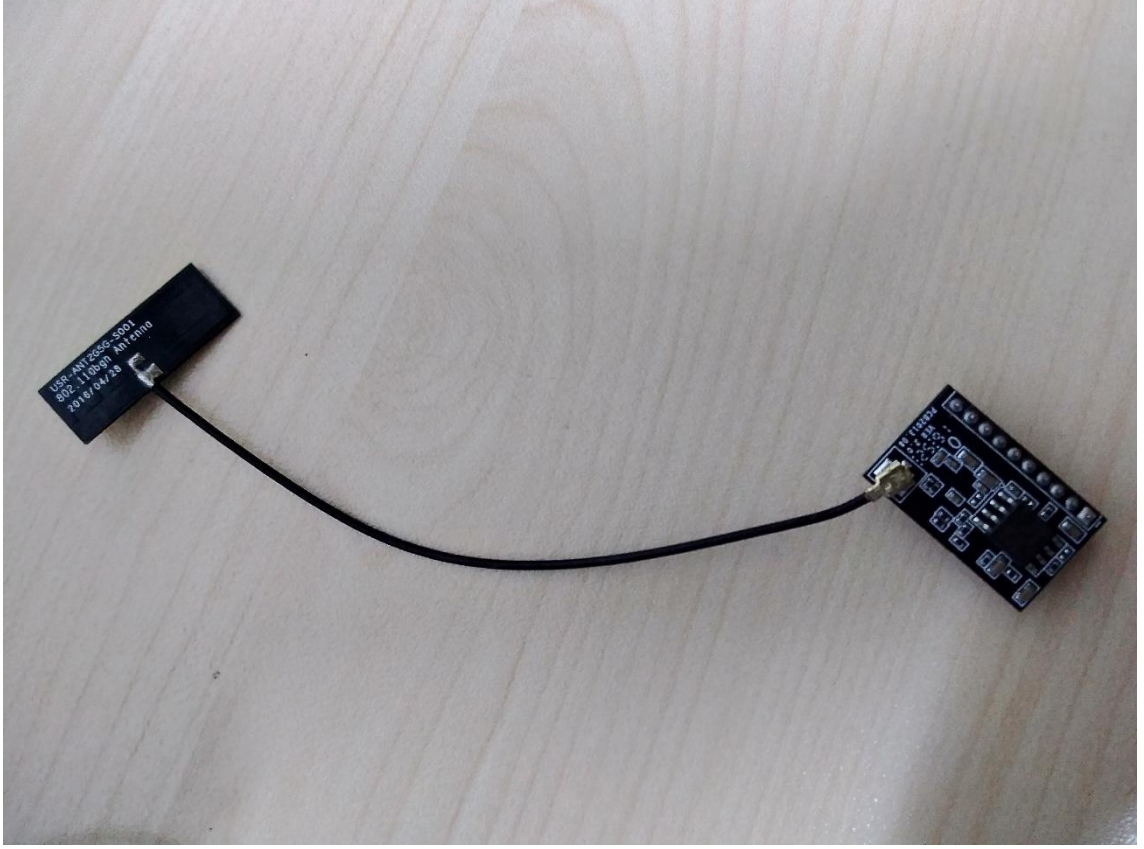


Şekil 2.7 nRF24L01'in aynı frekans üzerinde adreslenmesi

Şekil 1.8'de görüldüğü gibi PRX veri toplayıcı konumunda olduğu düşünülebilir ve sunucu anakart devresi bu pozisyonadadır. Diğer PTX ile tanımlanan diğer cihazlar ile half-duplex sorunsuz haberleşebilmektedir. nRF24L01+ half-duplex bir modüldür ve veri gönderirken aynı anda dinleyemez. Aynı zamanda birden fazla farklı adresten veri gelirse bu durumda da veri kaybolabilir. Lakin programatik şekilde gönder-al işlemi yapılırsa haberleşme güvenilir olarak gerçekleşmektedir.

2.4 LPT-100 Wi-Fi Modülü

WIFI-LPT100 / WIFI-LPB100, veri aktarımı için Seri arayüze sahip herhangi bir cihaza kablosuz arayüz sağlayan, tamamen bağımsız küçük form faktörü, tek akış, 802.11b / g / n Wi-Fi modülüdür.



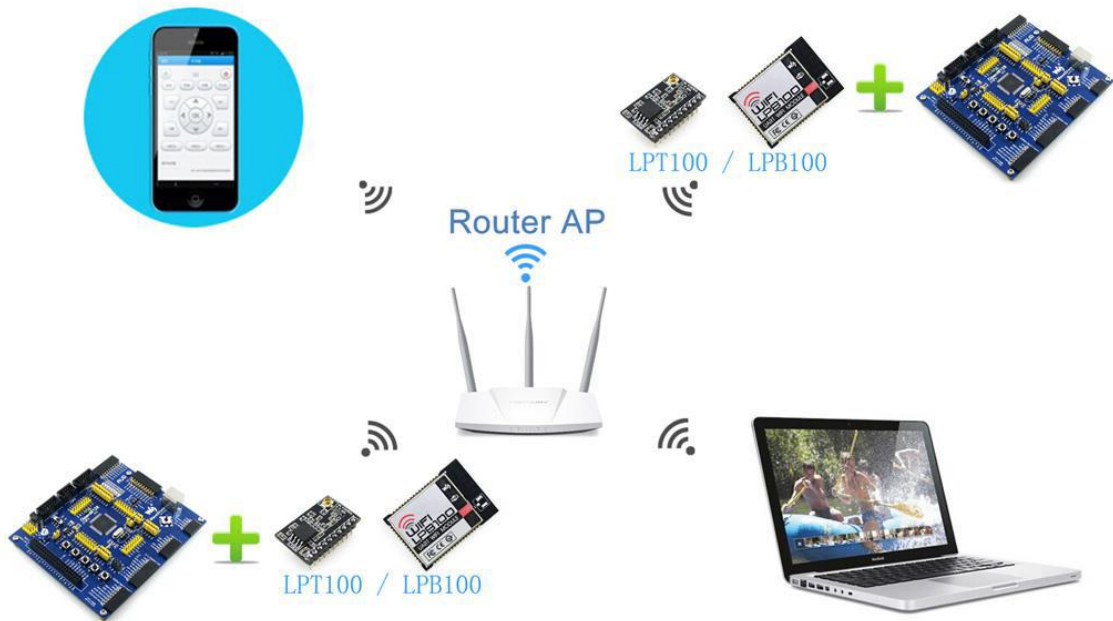
Şekil 2.8 LPT-100 Wi-Fi modülü anteni ile birlikte

WIFI-LPT100, yerleşik donanım yazılımında MAC, temel bant işlemcisi, "donanımda güç amplifikatörlü RF alıcı verici", bütün Wi-Fi protokolleri, yapılandırma işlevselliği ve ağ yığını çeşitli uygulamalarda tamamen kendi kendine yeten bir "802.11b / g / n" Wi-Fi çözümü yapmak için bir araya getirir.

WIFI-LPT100, dünyanın en düşük güç tüketimi yerleşik mimarisini benimser. Ev otomasyonunda, akıllı şebekede, el cihazında, kişisel tıbbi uygulamada ve daha düşük veri oranına sahip endüstriyel kontrolde her türlü istemci uygulaması için düşük hızda veri aktarımı veya alımı için optimize edilmiştir.

WIFI-LPT100, tüm Wi-Fi işlevselliğini, uygulamaya özel devrelerle ana PCB'ye kolayca monte edilebilen düşük profilli, küçük modül paketine entegre eder. Ayrıca, modül dahili anten, harici anten seçeneği sunar.

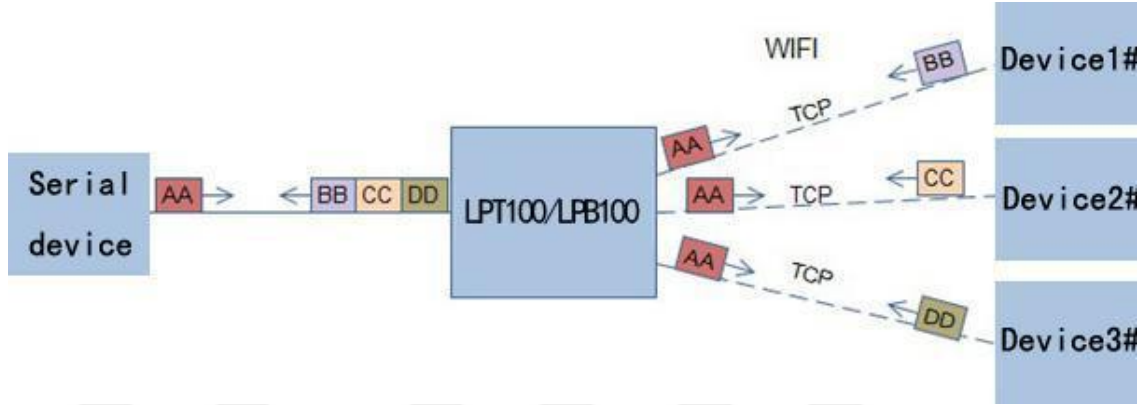
- EE IEEE802.11b / g / n Kablosuz Standartlarını destekler
- IFI WIFI@2.4 GHz'i destekler, WEP, WPA / WPA2 güvenlik modunu destekler.
- Kendi için Geliştirilen Yüksek Performanslı MCU
- Mükemmel Güç Tasarruf Şemasıyla Pil Uygulamaları için Ultra Düşük Güç
- UART / PWM / GPIO Veri İletişim Arayüzünü destekler
- STA / AP / AP + STA Modu Olarak Çalışmayı Destekler
- SmartLink İşlevi'ni destekler (APP programı sağlar)
- Kablosuz Yükseltme İşlevini Destekler
- WPS İşlevini Destekler
- Multi Çoklu TCP Bağlantısı (5 Kanal) Uygulamasını Destekler
- Dahili / Harici (I-PEX / SMA) Anten Seçeneğini destekler
- Tek + 3.3V Güç Kaynağı
- En Küçük Boyutlarda
- FCC / CE Belgeli



Şekil 2.9 LPT-100 Wi-Fi modülünün çalışma ağı

Kullandığımız bu modül yukarıdaki gibi bir ağ topolojisini desteklemektedir. Bu sayede tasarımı için uygun olduğu düşünüldü, alındı. Daha sonra terminalden ve modülün kendi yazılımından gerekli testler yapılarak çalışıp çalışmadığı kontrol

edildi. Çok yüksek hızlara ihtiyaç olunmadığı için ve veri güvenliği ön planda olduğu için istenilen performansı vermiştir.



Şekil 2.10 Wi-Fi modülünün sunduğu çok TCP bağlantılı veri iletim yapısı

Veri iletim yapısı kısmını incelediğimizde cihazlardan gelen verinin birbirlerine iletmeden sadece anakarta (serial device olarakda resimde görülebilir) veri göndermeleri istenilen bir husustu. Aynı zamanda anakartın gönderdiği verilen ise bütün Android cihazlara (Device1#,2#,3# vs.) aynı anda gönderilmesi de istenilen bir husustu. Kullanılan transparan modda, çift yönlü gelen datayı direk hedefe UART ile göndererek büyük bir zorluk aşılmış oldu.

Normalde Ethernet PHY ile yani Ethernet fiziksel katmanında tasarım planlanıyordu. 32-bit mikrodenetleyici bunu desteklemesine rağmen hem kablosuz bir çözüm tercih edilerek doğru bir yaklaşım tutuldu, hemde Ethernet PHY devresinin büyüklüğü ve elektronik malzemeler toplam maliyeti ciddi bir biçimde arttıracaktı.

Wi-Fi modülünün kendi yaydığı LPT-100 alan adı ile bağlanıldığında ve datasheette verilen kullanıcı adı ve parola ile modül ayarlarına ulaşıldığında burada arzu edilen tüm ayarlar uygulanabilmektedir. UART ayarları, soket ayarları, Wi-Fi modülü çalışma modları vs. ilgili tüm ayarlar Wi-Fi modülünün kendi alan adına bağlanılarak yapılabilmektedir.

2.5 Tasarlanan Özgün Haberleşme Paket Yapısı

Özgün bir sistem, özgün bir haberleşme paket yapısı gerektirmekte idi. Gönderdiğimiz haberleşme verilerinin karşı tarafta anlamlı bir sonuç üretmesi ve bu sonucun güvenli olması hedeflenen başlıca amaçlardandı. Bu sebeple önce haberleşme modüllerinin tek seferde gönderebildiği byte kapasiteleri incelendi ki, eğer kapasite aşılsa, ard arda gönderilecek olan paketlerin birbirlerine zincirlenmeleri gerekecekti. 32 byte'a kadar tek seferde gönderme kolaylığı sağlayan 2.4GHz GFSK modülü olan nRF24L01+, haberleşme paket yapısının byte uzunluğunu belirledi, Wi-Fi kısmında TCP/IP zaten byte gönderme işlevini kendi otomatik yaptığı ve paketler kaybolsa bile tekrar gönderme özelliğinden dolayı, gelen paketlerden emin olunduğu için GFSK haberleşmesi kısmı özgün paket yapısının uzunluğunu belirledi.

HEDEF MODUL ID				KAYNAK MODUL ID				RTC/ANROID MAC						MODUL TYPE	WRITE ENABLED
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

												AÇMA/KESME SÜRESİ			
PARAM	AYAR	PARAM	AYAR	PARAM	AYAR	PARAM	AYAR	PARAM	AYAR	PARAM	AYAR	TEMPH	TEMPL	HUMIDITYH	HUMIDITYL
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Şekil 2.11 Tasarlanan Özgün Paket Yapısının Bilgi Dağılımı

Byte uzunluğu belirlendikten sonra hangi byte'ın ne amaçla kullanacağına karar verildi. Haberleşmede ihtiyacımız olan eşsiz kimlikler için hedef ve kaynak olmak üzere 32-bit iki tane alan yani 8 byte'lık bir alan ayrıldı. Burada 0. Byte MSB ve 3. Byte LSB olacak işte kodlanmıştır. ID yani kimlik bilgisinin 32 bit olması, $2^{32} = 4294967296$ tane kimlik adreslenebildiğini bize gösterir. Yani Eşsiz kimlik verme konusunda hiçbir problem yaşanmaz. Bu da cihazları seri üretime elverişli hale getirmek amaçlıdır.

Devamında bulunan 6 byte'lık RTC/Android MAC alanı, sunucu devresi üzerinde tanımlı olan izin verilmiş Android cihazlar için 6 byte'lık Android MAC adresi alanıdır. Bu alan ihtiyaç halinde RTC (Gerçek Zamanlı Saat) içinde kullanıma ayrıldı.

Modül Tipi alanı haberleşmede kontrol cihazlarına giden komutların ayrışmasını ve sadece o modül tipine tanımlanmış fonksiyonların çalışması için oluşturulmuş $2^8 = 255$ tane modül tipi tanımlanabilecek bir byte'lık alandır. Örneğin; dimmer verisi Android cihazdan gönderildiği zaman eğer yanlışlıkla bile olsa doğalgaz kaçak

kontrol devresine gelirse, dimmer verisi devrede hiçbir deęişiklik yapmaz ve yok hükmünde sayılır.

Tablo 2.1 Modül tipleri ve karşılık gelen numaraları

Nümerik Deęer	Modül Tipi
1	Dimmer
2	Doęalgaz Kaçak Sensörü
3	Kapı Kontrol
4	Sıcaklık-Nem
5	Ütü Kontrol
6	Kombi Kontrol
7	Klima Kontrol

Yazma Aktifleştirme kısmından önce Parametre ve Ayar kısımlarına deęinirsek, önceden tanımlanmış bu nümerik parametreler hangi emir bilgisinin gönderildiğini içerir, ayar kısmı ise gönderilen emirde ayar bilgisi gerekiyorsa onu ifade eder. Örneğin; Dimmer verisi göndermek istersek; parametre kısmına 1, ayar kısmına 158 yazarsak, bu Dimmer verisinin gönderildiğini ve ayarının % üzerinden hesaplanırsa $\frac{158}{255} = \%62$ 'lik bir parlaklık ayarına tekabül ettiğini ifade eder. Parametre tablosu aşağıda ifade edilmiştir.

Tablo 2.2 Parametreler ve karşılık gelen nümerik deęerleri

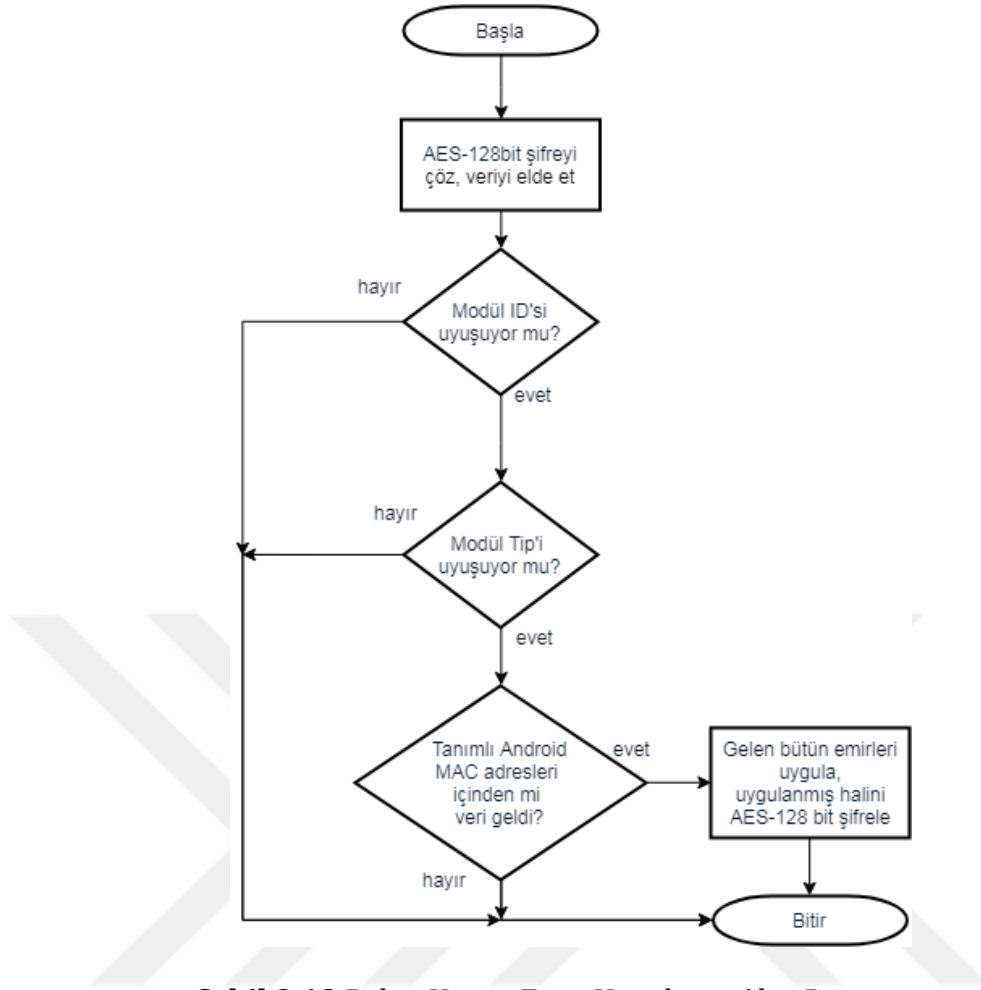
Nümerik Deęer	Parametreler ve Anlamına Geldiğı Komutlar
1	Dimmer ayar bilgisi

2	Kontrol Cihazları üzerindeki küçük LED'in ayar bilgisi
3	LM35 sıcaklık sensörünün verisi
4	MQ4 Doğalgaz Sensörü verisi
5	Gaz Rölesi Aç/Kapa verisi
6	Kapı durumu bilgisi
7	Buzzer ayar bilgisi
8	Sıcaklık-Nem bilgisi
9	Yüksek Wattlı Röle Aç/Kapa verisi
10	Modül Tip 2 Güç Açma/Kesme işlevi zamanlayıcısı başlat/durdur bilgisi(0:Durdur, 1: Açma başlat, 2:Kesme başlat)
12	Alarm Bilgisi
15	MQ4 Dijital Verisi
16	Sıcaklık-Nem cihazı aktif mi? bilgisi
17	Sıcaklık-Nem cihazı devamlı veri göndersin mi? bilgisi
18	MQ4 aktif mi? bilgisi
19	Kapı Alarm süresi çarpanı bilgisi
20	Kapı Uyarı süresi çarpanı bilgisi

21	Dimmer Zero-Crossing(Sıfır Geçişi) kaynak seçimi bilgisi(0: MOC3021 devresi, 1:Stabil kompatör devresi)
22	Dimmer Ampül Tipi seçimi bilgisi(0: Halojen,Dim Edilebilir, 1:Florasana, 2:LED)
28,29,30,31	Modül 2 Açma-Kesme Süre verisi//Sıcaklık ve Nem verisi

Bu parametreler kontrol cihazları içinde tanımlanmış paket yapısı kütüphanesinde hangi parametre geldiği zaman hangi emirler gerçekleştirilmeli hepsi tanımlanmıştır. 6 tane parametre ayar bölümünde bu parametre ve ayarları uygulamak için kontrol cihazlarında 6 defa emir uygulama fonksiyonu çağırılır, bu fonksiyon parametrelerin nümerik değerine bağlı olarak, ayar bilgisi de gerekiyorsa gerekli if else bloğu içinde kendine ait kod bloğuna gelir ve verilen emir gerçekleştirir. 6 tane emir tek haberleşme bloğunda aynı anda verilebilir. Kontrol cihazlarında 6 tane emir şimdilik yeterli olmaktadır ve bütün modül bilgileri tek seferde iletilebilmektedir.

Yazma Aktifleştirme (Write Enabled) alanı Android cihaz üzerinden gönderilen komutun, kontrol cihazlarında değişiklik yapmasını sağlar. 1 gönderilirse yazma işlevi yani gönderilen emirleri gerçekleştirmeyi uygular ve geri gönderir. 0 gönderilirse gönderilen emirler okunur ve geri gönderilir.



Şekil 2.12 Paket Yapısı Emir Uygulama Akış Diyagramı

2.6 Verilerin AES-128bit Şifreleme ile Korunması

Tasarımda öne çıkan argümanlardan biri de ev otomasyonunun haberleşmesinin şifreli bir biçimde korunmasıdır. AES (Geliştirilmiş Şifreleme Sistemi) ile korunan sistemimiz dışarıdan erişildiğinde ele geçirilen verilerin anlaşılmasını imkansız kılmaktadır. Oluşturulan bir Cypher Key(Siber anahtar) ile 32 byte'lık haberleşme bloğu şifrelenmekte ve hem Wi-Fi TCP/IP socket kısmında, hemde nRF24L01+ GFSK modül haberleşmesinde şifreli halde gelen veriler analiz edilmektedir. Şifreli data gelince verilen aynı Siber anahtar ile çözülür. Şifreli verinin çözülmesi işlemi işlemcide belirli bir zaman harcamakla beraber kullanımda belirgin değildir. Çözme veya şifreleme işlemi yaklaşık 100ms civarında zaman harcamaktadır. Çözülen data işleme alınır ve eğer kontrol cihazları tarafından okunuyorsa önce verinin kendisine gelip gelmediğini kontrol eder, yani hedef modül olup olunmadığına bakılır. Ardından emir fonksiyonları çağırılarak parametre ve ayar emirleri gerçekleştirilir. Android kısmında gelen veri çözüldükten sonra, Android yazılımı üstünde

tanımlanan modül kutucuklarına geri bildirim göndererek ilgili kutuyu günceller. Şifreleme algoritması olarak “github” da “kokke” adlı yazarın “tiny-AES-c” kütüphanesi kullanılmıştır. Bilgisayarda çalışan uygulamaların hazırlandığı diller için C dili ile kodlanmış, lakin alındığı gibi hali kullanılmaya elverişli olmayan bu kütüphane mikrodenetleyicide çalıştırabilmek için düzenlenmiştir.



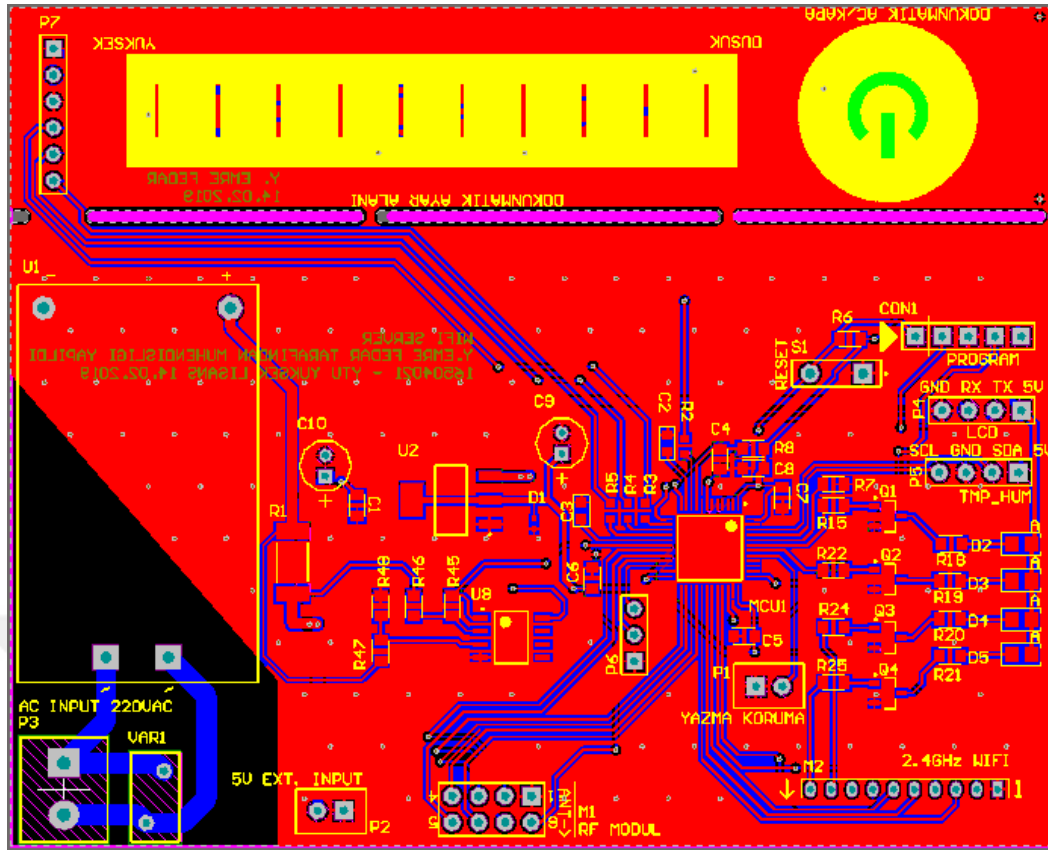
3.1 Donanım Bazında Tasarım Yaklaşımı

Sunucu Devresinin tasarımı, düzenlenmesi ve sağlıklı çalışması için birçok parametre göz önünde bulundurulmuştur. Devre Altium'da çizilmeye başlanmadan önce işlemci seçilmiş, beklentileri karşılıp karşılamayacağı tartışılmıştır. Türkiye'de elektronik komponent satan firmalardaki işlemciler ve stokları incelenmiş, stoklarda çokça olan, birçok arayüz içeren ve STM32 ailesinden güçlü bir işlemci olan STM32L052C6 işlemcisi seçilmiştir. Sunucu cihazı için bu işlemcide karar verilince, diğer kontrol cihazlarıyla uyumlu çalışması için yine STM32 ailesinden daha uygun fiyatlı 32-bit işlemciler seçilmiştir. Sunucu cihazı için seçilen işlemci barındırdığı birçok özellekle beraber dokunmatik algılama kütüphaneleri ve sensörleri de içermektedir. LCD arayüzü için gereken fazladan UART arayüzünün de olması işlemcinin seçilme sebeplerindendir. Wi-Fi arayüzü için LPT-100 modülünün UART arayüzü kullanılmakta ve işlemci ile bağlantılı harici 6 ayar bacağı bulunmaktadır. Kontrol cihazlarıyla haberleşmeyi sağlayan nRF24L01+ GFSK 2.4GHz modülü ile de SPI haberleşmesi ile iletişim kurulmaktadır ve işlemciden 6 bacak kullanılmaktadır. Bu yüzden çok bacak ve çok fonksiyon gerektirdiği için STM32L052C6'da karar kılınmıştır.

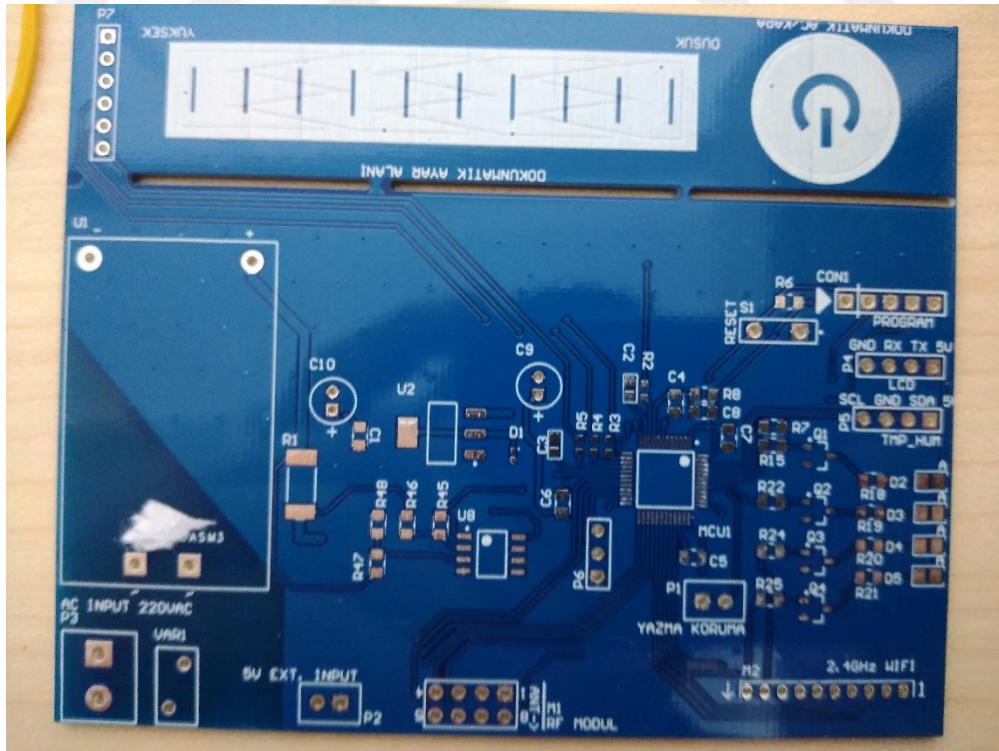
Daha sonra Altium Designer'da diğer titizlikle seçilen tüm komponentlerle beraber bu devre tasarlanmış ve ortaya çıkan PCB Çin'de PCBWAY firmasına bastırılmıştır. Komponentlerin Türkiye'de satın alınabilirliği, varsa Çin'den getirilebilirliği üzerine komponent seçimi yapılmıştır. Devrede elemanların yerleştirilmesinde otomatik yerleştirme algoritmaları kullanılmayıp, tamamen özgün, ayrıntılı ve kaliteli bir yerleşim ve tasarım uygulanmıştır. Devrenin üzerindeki bakır yolların, izlediği hat ve yol kalınlığı, akım ve gerilim parametreleri üzerinden bilimsel olarak hesaplanmış, yerleştirilmiş ve çizilmiştir.

uygulamalarından faydalanılmıştır. Altium'da komponentlerin kullanıldığı kütüphane tez için özel olarak oluşturulmuş olup, %95 oranında tek tek elle çizilmiş bu komponentlerin kütüphane çizimleri de özgündür.

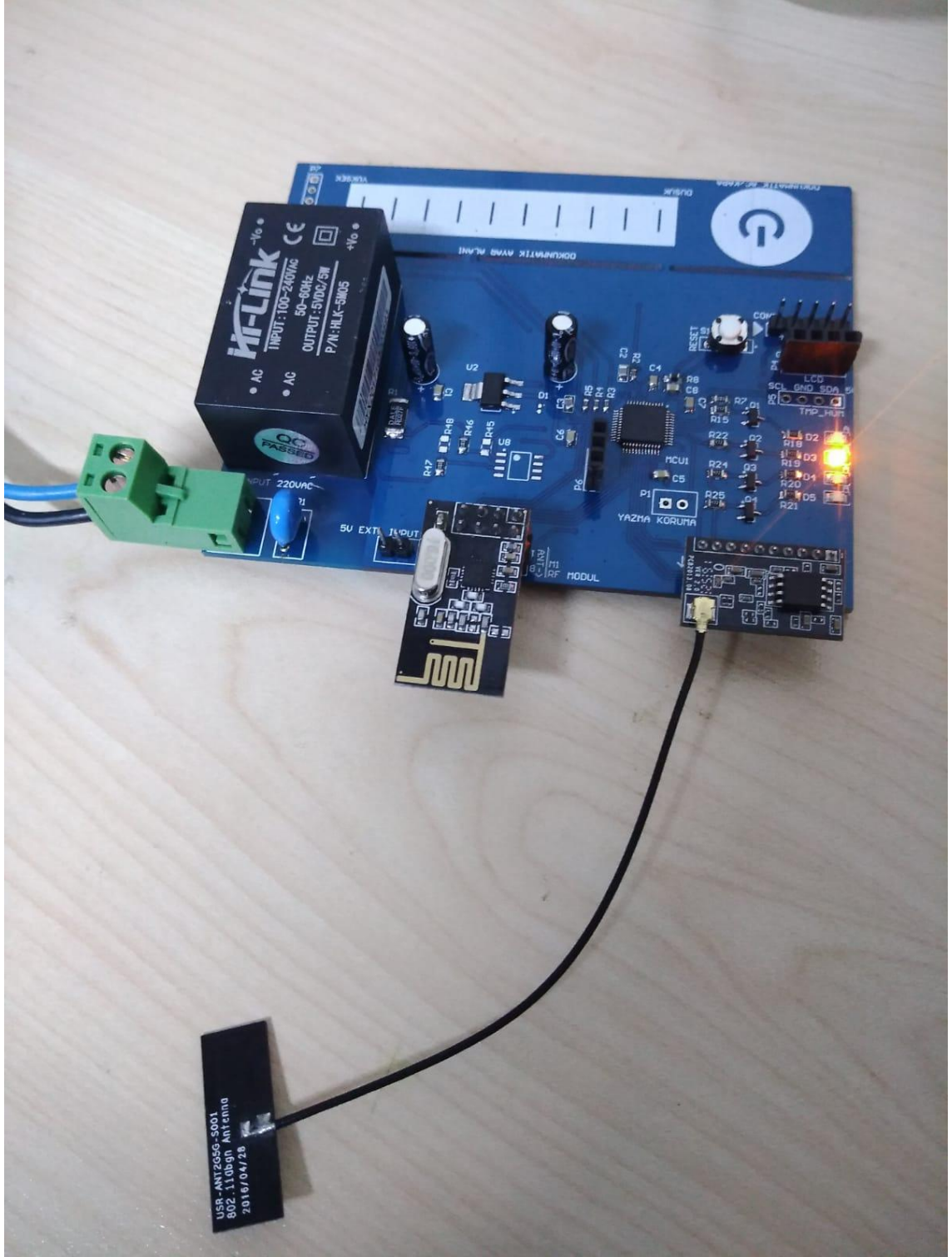
Devre, girişinden aldığı AC gerilimi 5 VDC gerilime dönüştüren küçük bir 5 W Hi-Link marka SMPS modülüne sahiptir. 5 V devrede varsa seçmeli yerleştirilen Sıcaklık-Nem modülünü, varsa seçmeli yerleştirilen LCD'yi, ve LED'leri beslemekte kullanılır, 3.3 V gereken elemanlar için AMS1117-3.3 regülatörü ile 3.3 V seviyesine düşürülür. 3.3 V işlemciyi, nRF24L01+ cihazını ve LPT-100 Wi-Fi modülünü besler. nRF24L01+ SPI ile işlemciye, LPT-100 Wi-Fi modülü UART ile işlemciyle haberleşmektedir. Haberleşme modüllerinin yerleşme düzeni, birbirlerinin haberleşme sağlığını bozmayacak şekilde, PCB üzerinde birbirine uzak yerleştirilmiştir. Devrede gürültüyü azaltmak için PCB alt ve üst plakası GND (Ground) plakasıyla örtülmüştür. PCB üzerindeki yolların kalınlığı ve kullanılan via'lar yani deliklerin çapı, geçen akımla doğrudan ilişkili olup tasarlanmış, ayrıca PCB basan firmanın verdiği sınırlamalar içinde olacak şekilde ayarlanmıştır. Tasarım sonlandığında gerber çıktı dosyaları alınmış olup bu dosyalarla PCB üretim firmasından satın alma gerçekleştirilmiştir. Devreler kurulup test edildiğinde sağlıklı bir şekilde çalıştıkları gözlemlenmiştir.



Şekil 3.2 Sunucu Devresinin Altium'da PCB çizimi görüntüsü



Şekil 3.3 Sunucu Devresinin PCB üreten firmaya bastırılmış hali



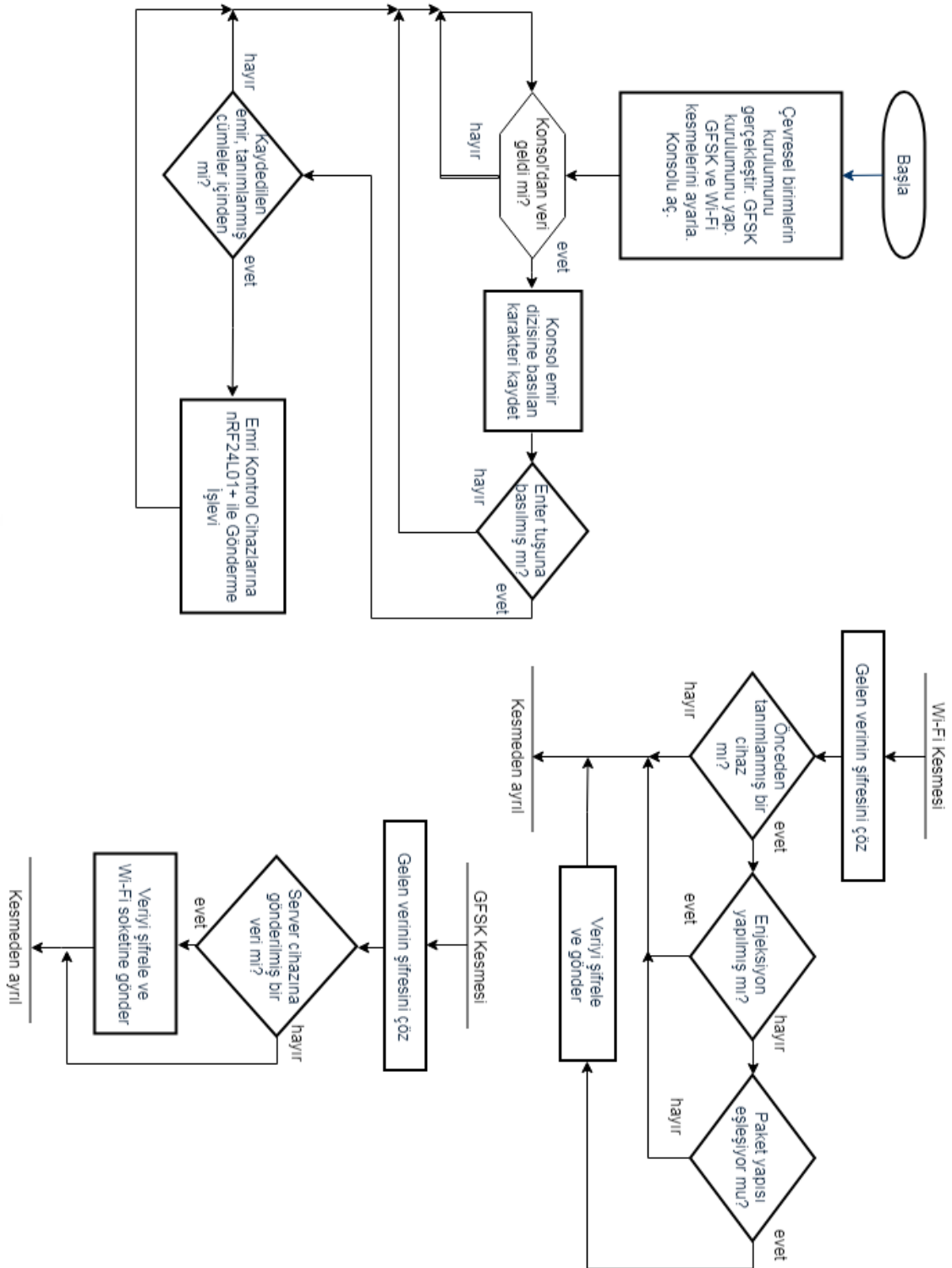
Şekil 3.4 Sunucu Cihazı

3.2 Yazılım Yaklaşımı

Sunucu cihazı, Android cihazlardan gelen emirleri kontrol cihazına iletmekle ve yabancı girişleri engelleyici güvenliği almakla görevlidir. Haberleşmenin aksamadan, hızlı ve güvenli olması ana önceliktir. Sunucu cihazı ile Android cihaz

arasındaki haberleşme Wi-Fi TCP/IP soket arayüzü, TCP/IP protokolünün getirdiği özellikler sayesinde güvenli ve hızlıdır. nRF24L01+ GFSK modülü haberleşme işlevlerine ayarlama gerektiği için daha çok yoğunlaştırılmıştır. Sistem veri alışverişi yaparken diğer işler aksatılmamalı ve işlemciyi tek iş meşgul etmemelidir. Bu yüzden (IRQ) Kesme'ler kullanılarak, nRF24L01+'e data geldiğinde çalışan EXT_IRQ ve Wi-Fi modülü için UART kesmesine öncelik verilerek, veri kayıpları engellenmeye çalışılmış, işlemcinin verimli kullanılması amaçlanmıştır. İşlemci kodları Ek 1.'de verilmiştir. Aşağıda Sunucu cihazının çalışma akış diyagramı verilmiştir.

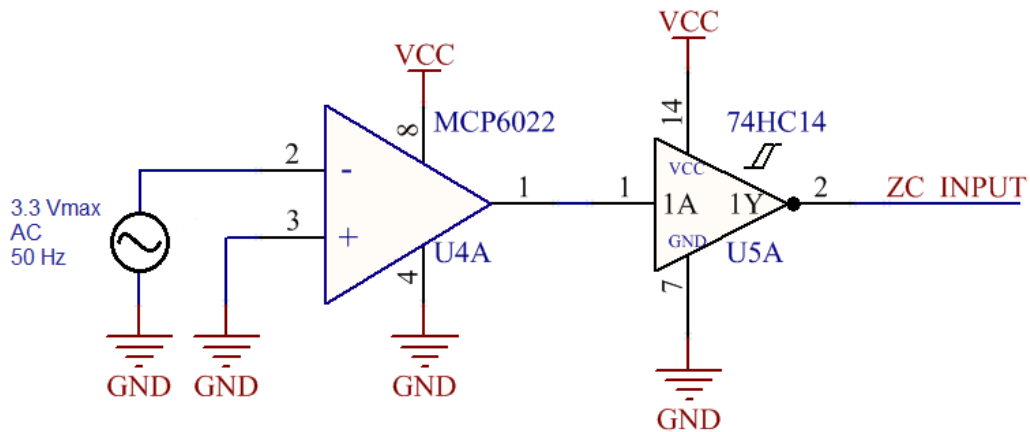
Wi-Fi modem sunucu cihazına bir IP verir. Verilen IP aynı yerel ağdaki bir Android kullanıcısı tarafından Android yazılımında görülebilecektir. Eğer ev dışından bir bağlantı olacaksa bu IP o kişiye verilir. İnternet arayüzünü sağlayan Wi-Fi modülü devreye USART üzerinden bağlı olup şeffaf data iletimine sahiptir. Yani gelen/giden veriyi direk internete açılan port üzerinden TCP/IP protokolü ile Android kullanıcılarına gönderir/alır. Android kullanıcılarına sistemin durumunu bir bütün olarak gönderir, yani bütün kontrol cihazlarının durumu Android kullanıcısından güncel olarak takip edilebilmektedir, bu devamlı (continious) çalışma modudur. Sadece kesme (interrupt) ve alarm alındığında haberleşen modda mevcuttur. Bu durumda karışıklıktan kaçınmak için bütün kontrol cihazları bu moda geçerek senkron bir davranış sergiler. Kesme ve alarm modunda çok daha az veri harcanacaktır ve güç tüketimi daha az olacaktır, bu nedenle hiçbir zaman ağ'a yük olmayacaktır.



Şekil 3.5 Sunucu Devresi Çalışma Akış Diyagramı

4.1 Dimmer Devresi

Tasarımda yer alan diğer bir devre uzaktan lambaların açılıp kapanmasını ve parlaklığın arttırılıp azaltılmasını sağlamaktadır. Devre açma/kapama işlemini sürülen bir röle yardımıyla yapmaktadır. Dimmer işlevi için triyak ve komparatör mantığıyla çalışan hassas sıfır-geçiş devresi kullanılmıştır. Diğer yapılan çalışmalarda 220VAC gerilimi DC'ye doğrultup yaklaşık 318VDC gerilim elde edilmiş, özel bir MOS yardımıyla Buck Converter (Alçaltıcı Dönüştürücü) mantığında gerilim düşürülerek lambalar bu DC gerilimle sürülmektedir [17]. Tabi bu durum kullanılan elemanlar üzerinde çok fazla ısı kaybıyla sonuçlanmaktadır. Bu durumdan kurtulmak için kayıpları çok az olan yüksek amperli triyak kullanılmıştır. Ayrıca ışık parlaklığı ayarlanamayan LED ve kompakt floresan lambalarında ışığı ayarlanabilmektedir. Tabi bu işlem lambaları zorlamakta ve ömürlerini kısaltmaktadır. Yeni sunulan devrede triyak çıkışı doğrultulmaktadır. Triyak çıkışının doğrultulduğu uygulamalar azdır. Lakin triyak çıkışının doğrultulup lambaların sürüldüğü devre yeni bir yaklaşımdır ve mevcut değildir.

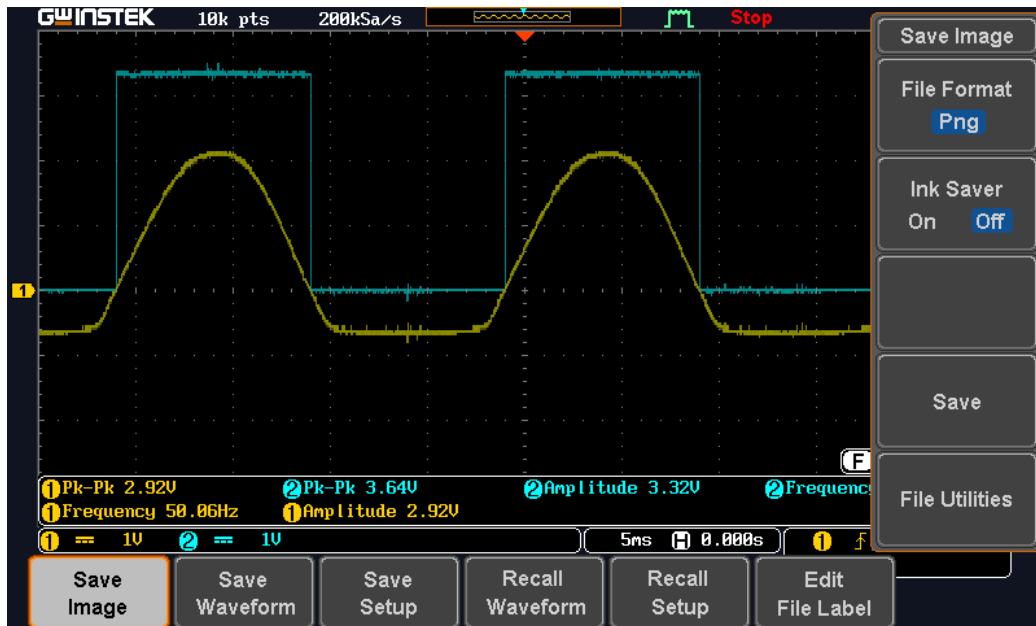


Şekil 4.1 Dimmer Sıfır Geçiş Devresi

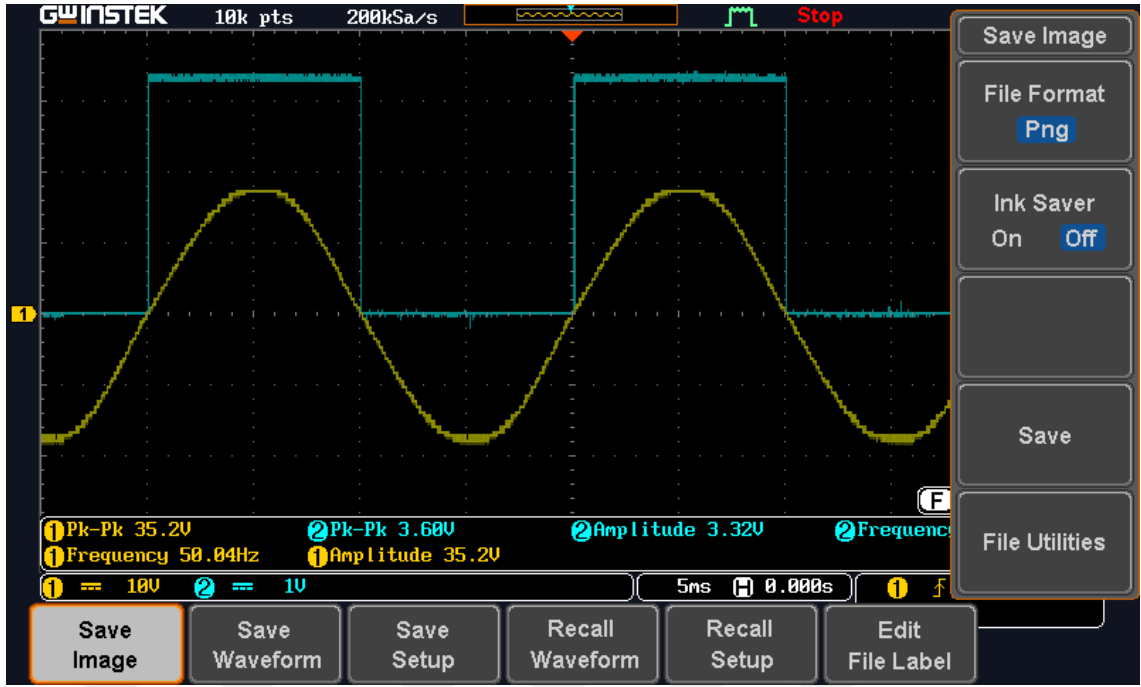
Elde edilen DC çıkış, dim edilemeyen LED ve floresandaki titreşimleri ve yanıp sönmeleri ciddi oranda azalttığı görülmüştür. Sıfır-geçiş devre şeması Şekil 4.1'de

görülmektedir. Bu yapı ile ilgili internetten birçok kaynaktan yararlanılmıştır [18]. İnternette yayımlanan bu kaynaklardaki devreler, AC gerilim olarak 3.3 Vmax AC değil, 12 V AC, 18 V AC ye göre tasarlanmıştır. Ayrıca 12 V AC veya 18 V AC kaynağın ne olduğundan, nasıl elde edildiğinden bahsedilmemiştir. Çevresel birimlerinde 5 V DC toleransına sahip olsa da kullanılan işlemci 3.3V DC ile çalışmakta, çevresel birimleri bu gerilimde kullanılmaktadır. Bu sebeple seçilen opamptan, lojik tersleyiciye kadar 3.3 V DC gerilim üzerinden tasarım yapılmıştır. Ayrıca en önemlisi ve tehlikeli olan şu ki, 220 V AC gerilimin Nötr ucunun, alçak gerilim GND'si ile kısa devre yapılmasıdır. İzolasyonsuz böyle bir davranış çok yüksek ihtimalle cihazın tamamen yanmasıyla sonuçlanacaktır.

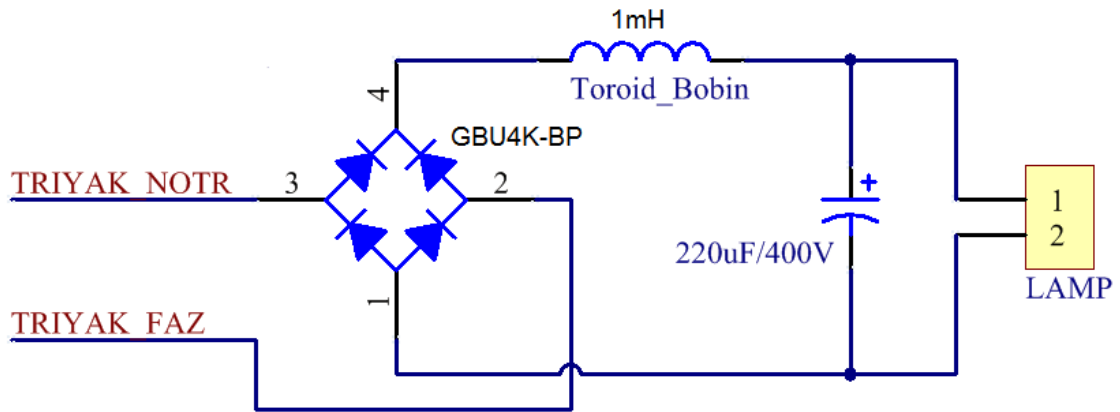
Tasarlanan devrede 3.3 Vmax AC, 220 V AC gerilim önce 5 VA gücündeki bir izolasyon trafosuyla 12 V AC gerilime düşürüldü, ardından da gerilim bölücü yardımıyla 3.3Vmax AC seviyesine getirildi. 3.3Vmax AC seviyesine getirilmesinin sebebi 3.3 V'ta çalışan mikrodenetleyiciye ve komponentlere zarar gelmemesidir. Şekil 4.1'de görüldüğü gibi MCP6022 rail-to-rail opampı komparatör olarak kullanılmıştır. Devre bu şekilde kullanıldığında Şekil 4.2'de sarı renkte gösterilen opamp giriş işaretlerinde negative alternansta bir kırılma meydana gelmektedir, lakin sıfır geçişi devresinin çalışmasına herhangi bir olumsuz etkisi olmamıştır. Sıfır geçişi devresinin tasarlanan dimmer cihazı üzerinden osiloskop görüntüleri Şekil 4.2 ve Şekil 4.3'te gösterildiği gibidir.



Şekil 4.2 Opamp Giriş Bacağındaki Sinüzoidal Sinyalden Sıfır Geçiş Elde Edilmesi

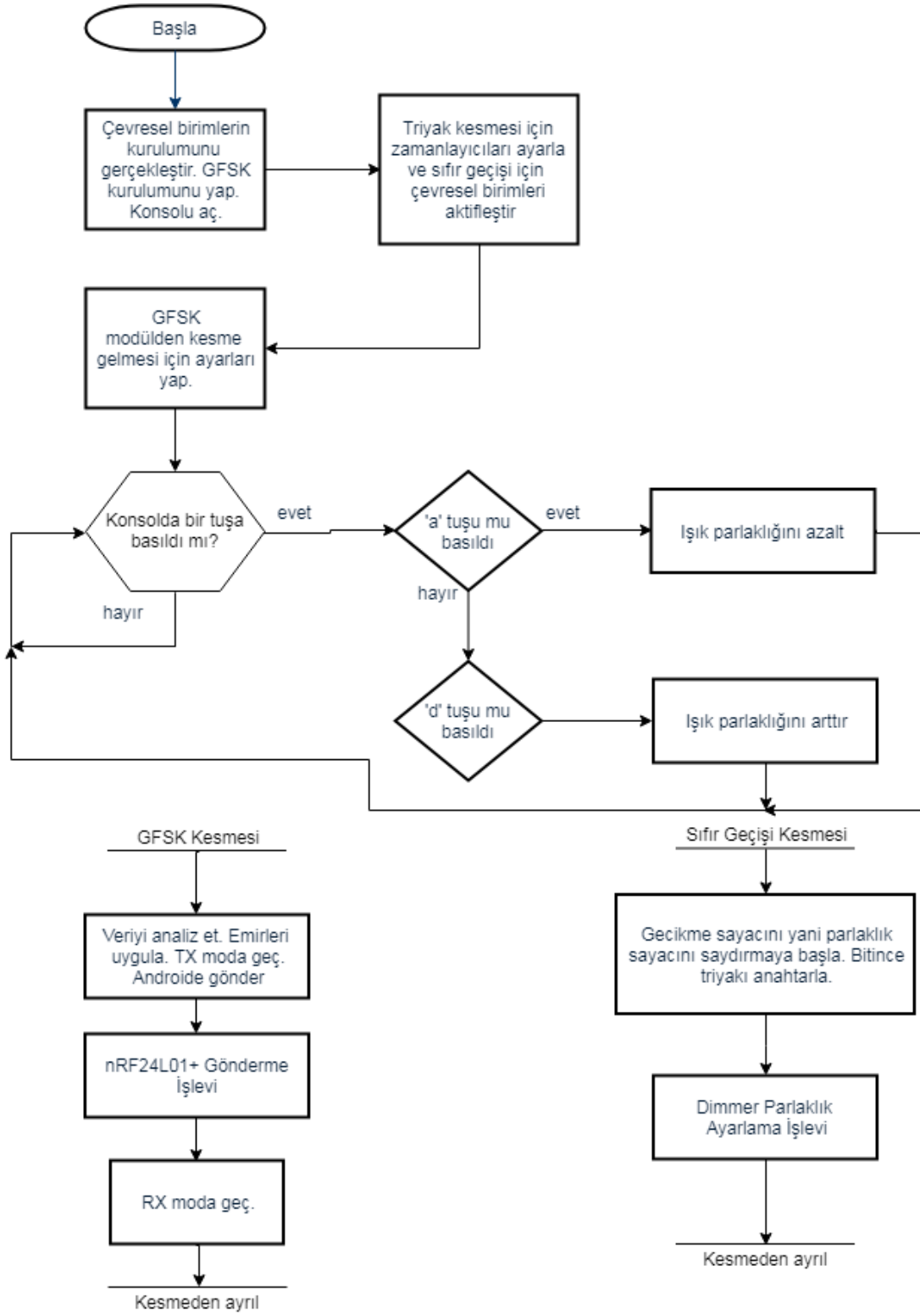


Şekil 4.3 Trafodan Alınan 12VAC Sinyali ile Sıfır Geçişinin Karşılaştırılması

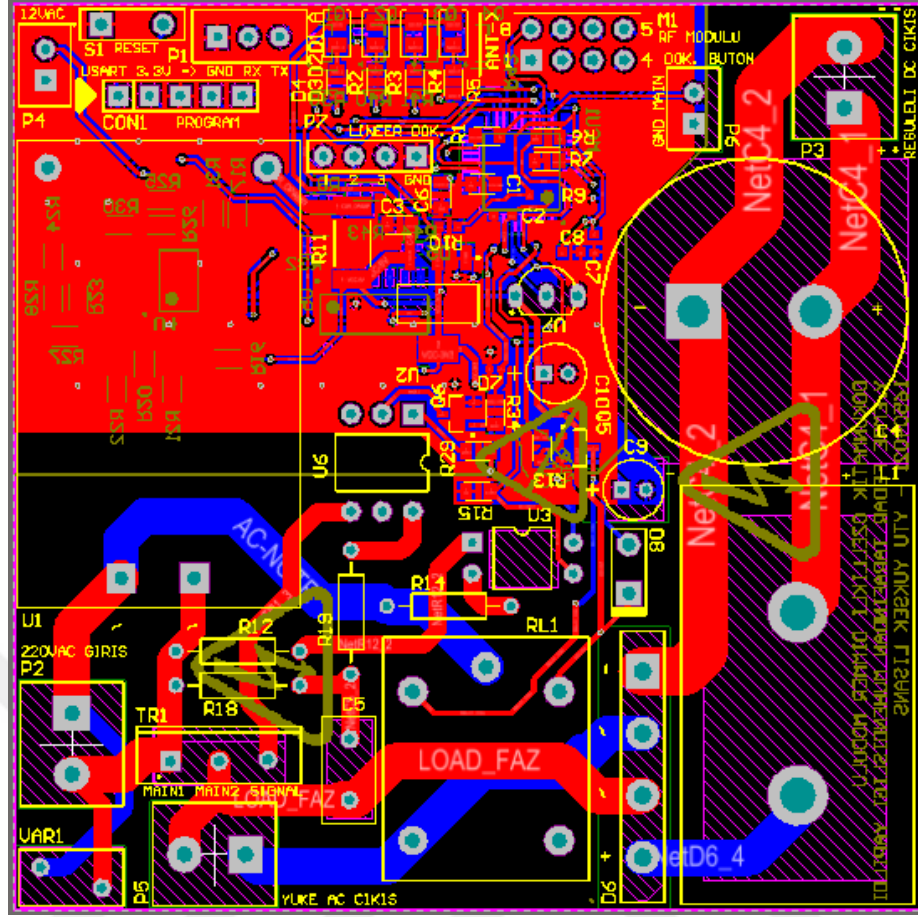


Şekil 4.4 Triyak Çıkışının Doğrultması

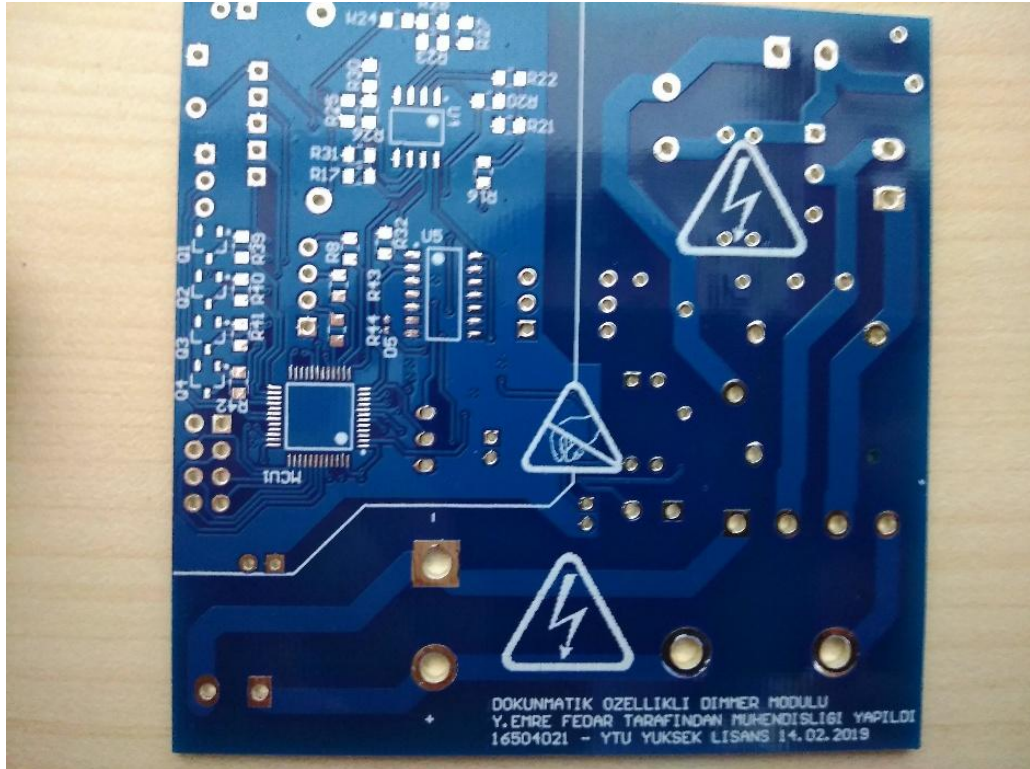
Şekil 4.4'te görüldüğü gibi Triyak çıkışının doğrultulması sonucu farklı lambalara dim edilen DC gerilim verilmiştir. Bu doğrultuda akkor ve halojen lambalarda dim edilme işlemi normal olarak gerçekleşmiştir. Dim edilemeyen kompakt florasen ve ledlerinde bu sayede dim edilmesi sağlanmış ve ışığın titremesinin %60 civarında azalmasına katkı sağlamıştır. Tabi bu durum dim edilemeyen florasen ve led lambalarda kendi iç devrelerinin aşırı ısınmasına sebep olmuş, lambayı yormuştur.



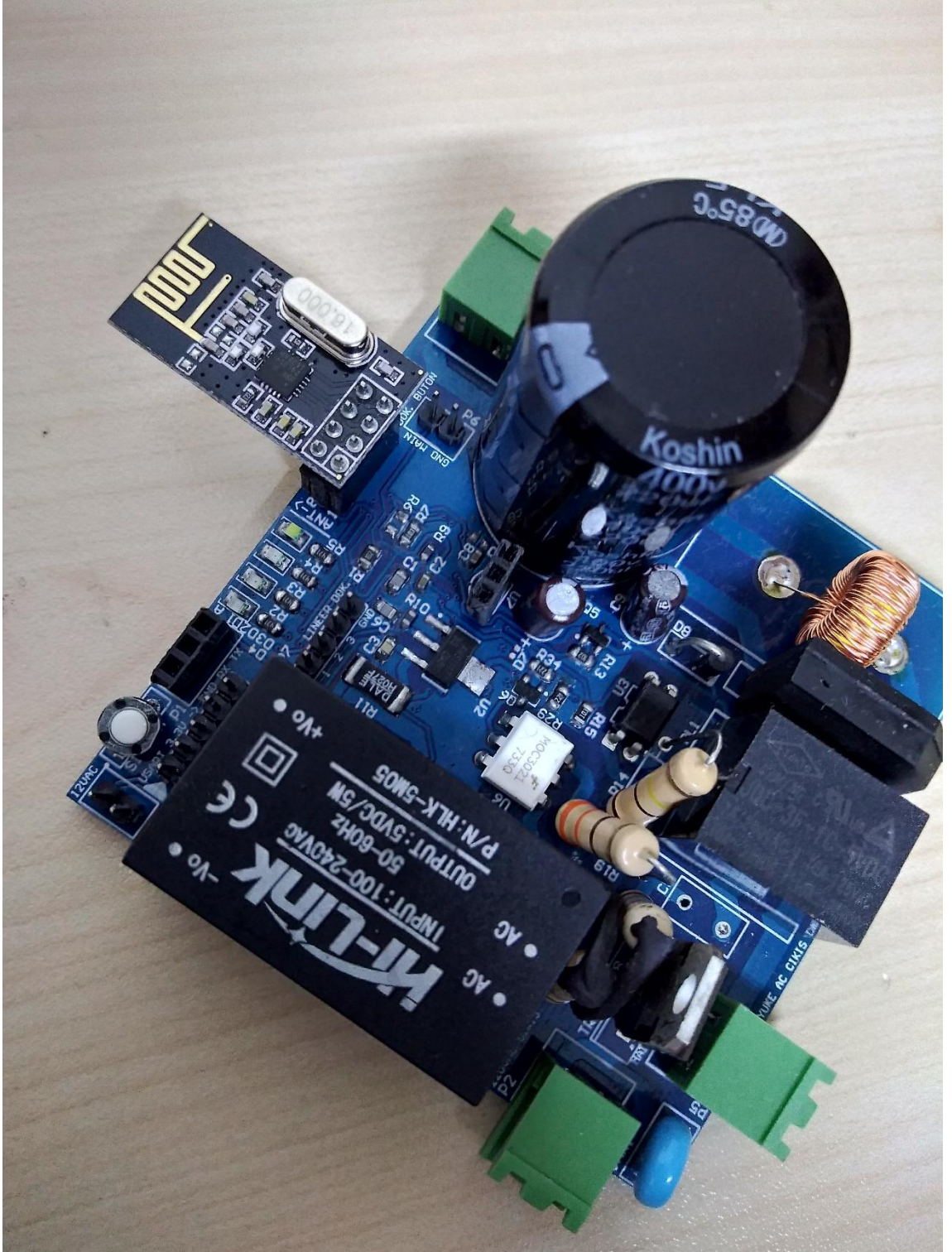
Şekil 4.5 Dimmer Kontrol Cihazı Akış Diyagramı



Şekil 4.6 Dimmer Devresi PCB Çizim Görseli



Şekil 4.7 Dimmer Devresi Bastırılmış PCB boş hali (alttan görünüm)



Şekil 4.8 Dimmer Kontrol Devresi son hali (üstten görünüm)

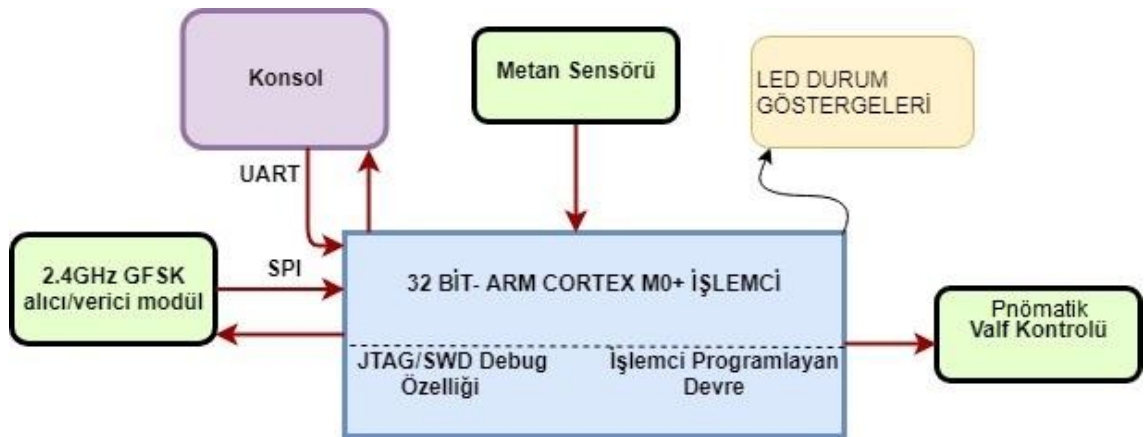
4.2 Doğalgaz Kontrol Devresi

Doğalgaz Kaçağı Tespit modülünde kullanılan Metan MEMS modülü MQ-4, Metan gazının seviyesine göre analog gerilim çıkışı verir, bu analog çıkış ADC arayüzü ile işlemcide analiz edilir. Hesaplanan değer gaz kritik seviyesinin üstünde çıkar ise, ocağa entegre edilen selenoid doğalgaz valfi, AC gerilim verilerek OFF konumuna getirilir. Bununla ilgili Android kullanıcısına bilgi mesajı gönderilir. Gönderilen bu mesaj yine şifreli ve kimlik no ile gönderilerek hangi modülden geldiği bilinir.

Selenoid valf olarak TORK doğalgaz solenoid ½ valf kullanılmıştır.

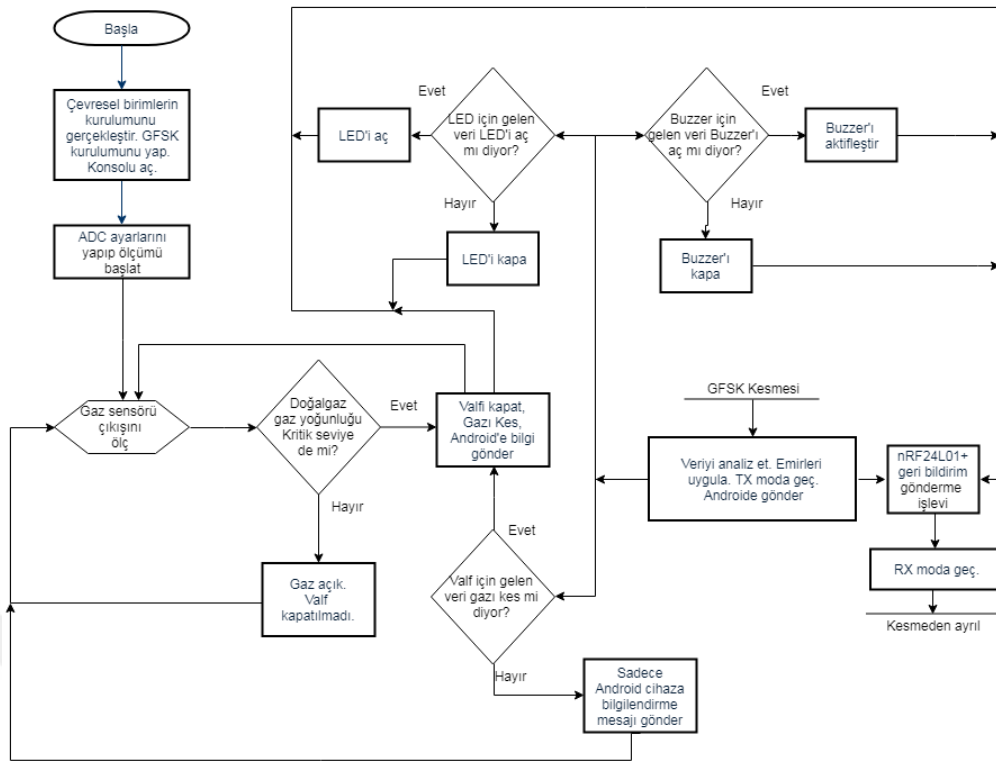


Şekil 4.9 TORK Selenoid Doğalgaz Valfi Normalde Açık ½

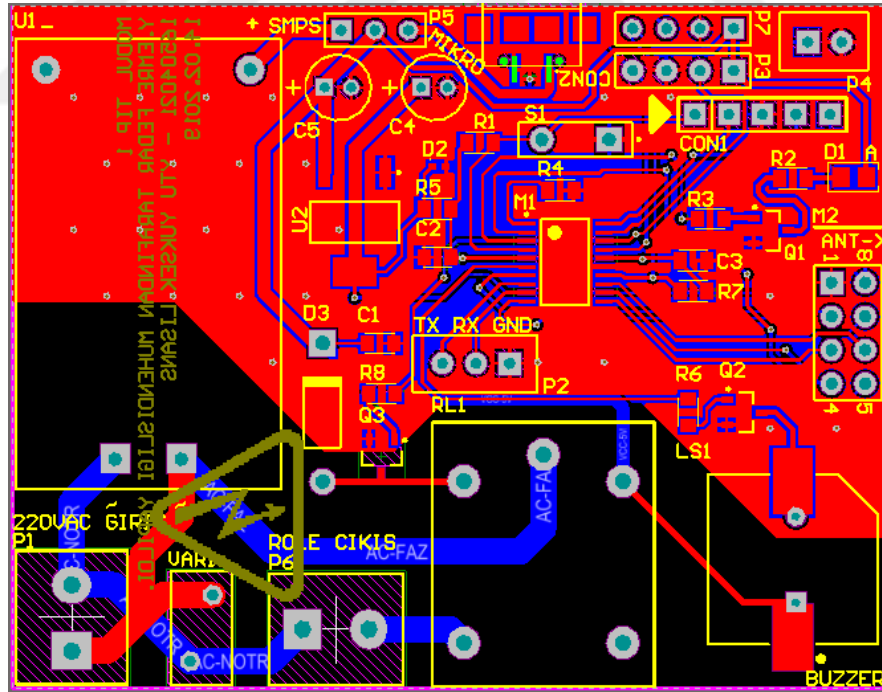


Şekil 4.10 Doğalgaz Kontrol Devresinin Şeması

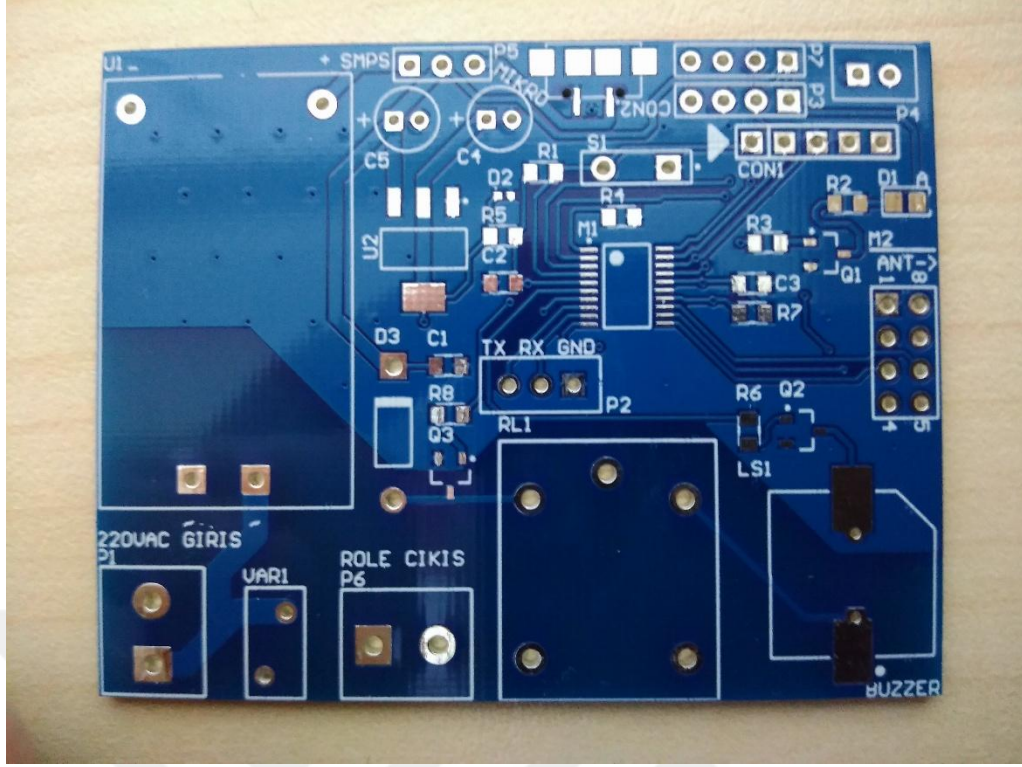
Doğalgaz Kontrol Devresinin Akış Diyagramı aşağıda verilmiştir.



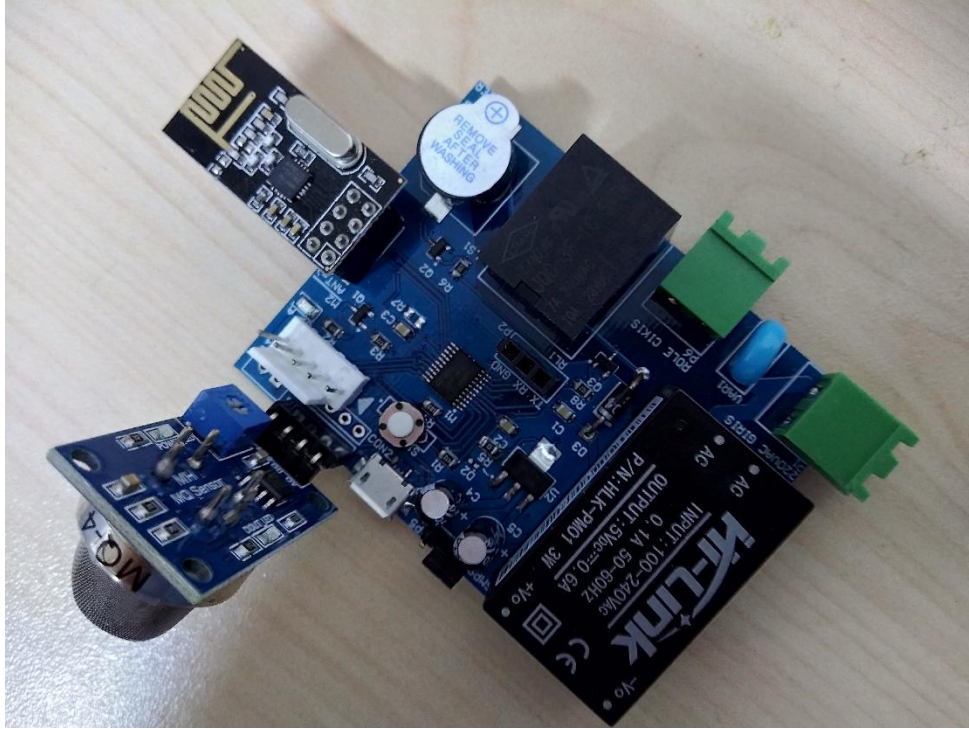
Şekil 4.11 Doğalgaz Kontrol Devresinin Akış Diyagramı



Şekil 4.12 Doğalgaz Kontrol Devresinin PCB çizim görseli



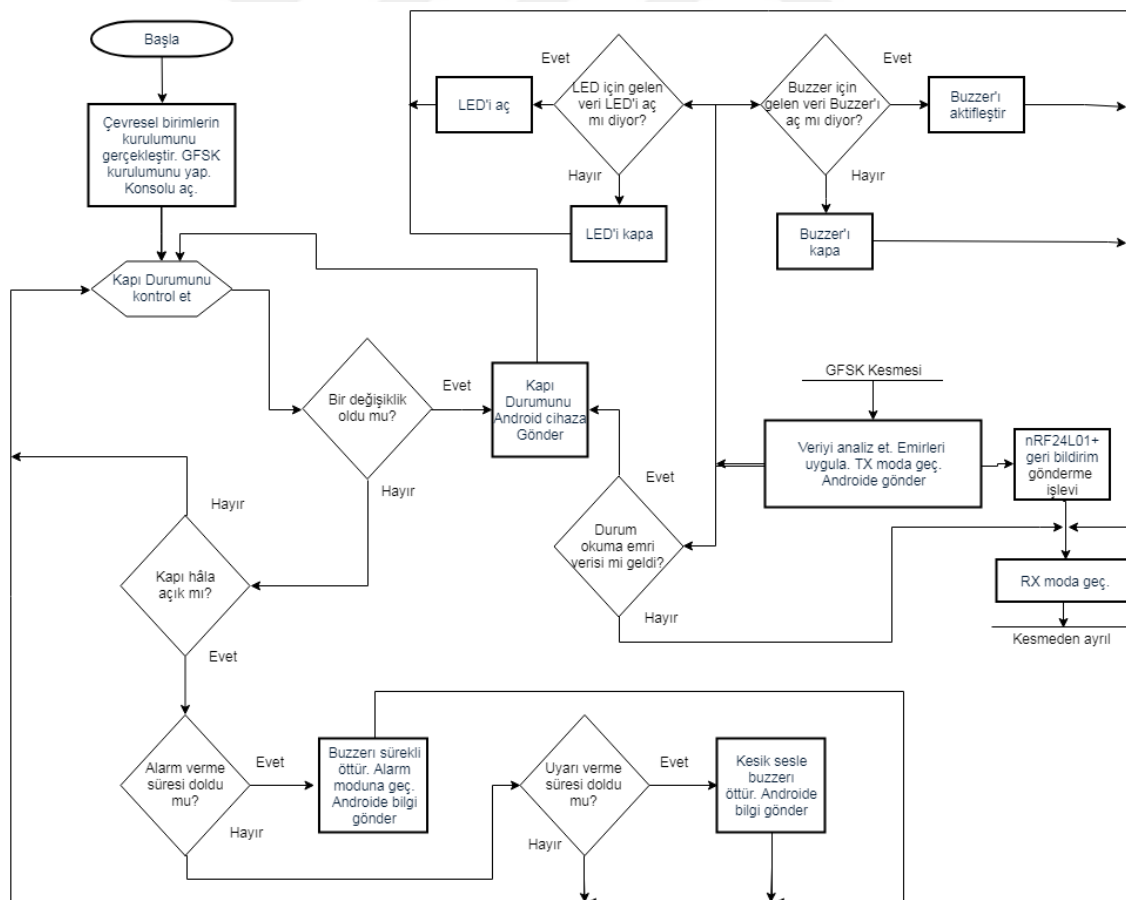
Şekil 4.13 Doğalgaz Kontrol Devresi Bastırılmış PCB boş hali (üstten görünüm)



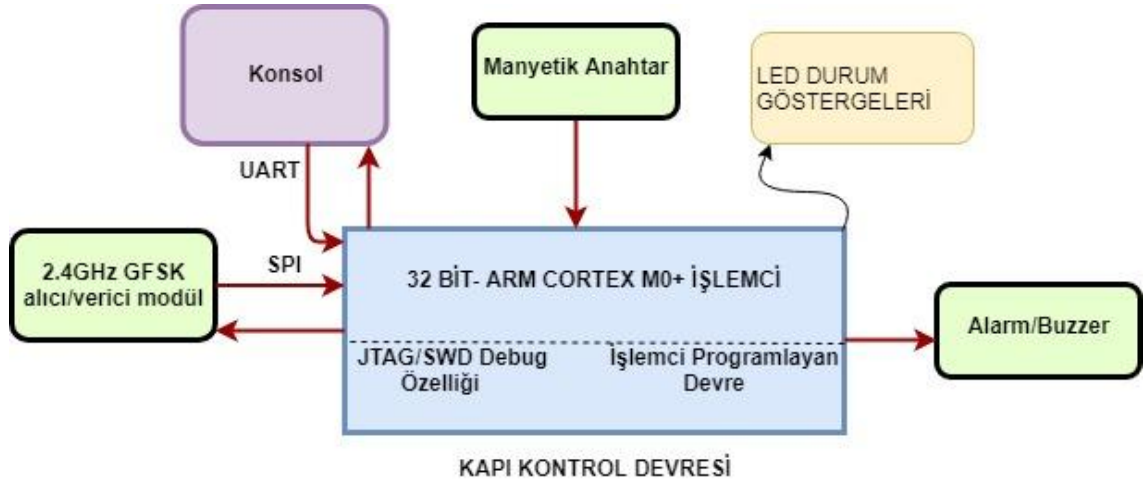
Şekil 4.14 Doğalgaz Kontrol Devresi son hali (üstten görünüm)

4.3 Kapı Kontrol Devresi

Kapı kontrol devresinde, kapının açık olup olmadığı bilgisini elde etmek için, kapı ile kapı kasasına paralel monte edilen manyetik anahtarlardan yerleştirilmiştir. Bu anahtar kapı açıldığında birbirinden uzaklaşarak açık devre olup, açık kolektör transistörün kapatılması gibi davranarak, mikrodenetleyicide giriş bacağını 3.3V'a çekmektedir, bu dijital veri işlemcide okunarak kapı durum bilgisi elde edilmektedir. Kapı açıldığı veya kapandığı zaman tasarlanan paket yapısı çerçevesinde Android cihaza anında bilgi gönderilmektedir. Bu bilgi Android kullanıcısına iletilirken sadece kapının açılmasının istenmediği gece saatlerinde veya iş saatlerinde, bildirim gönderme şeklinde ayarlanabileceği gibi, tüm gün bildirim verecek şekilde de ayarlanabilir. Ayrıca kapının son açılma tarih ve saatinin tutulduğu ayrı bir bilgilendirme çubuğu da mevcuttur. Bu sayede kullanıcı istediği zaman Android cihazından sorgulatarak, kapının durumunu ve en son kapı ne zaman açılmış bunun bilgisini elde edebilmektedir.

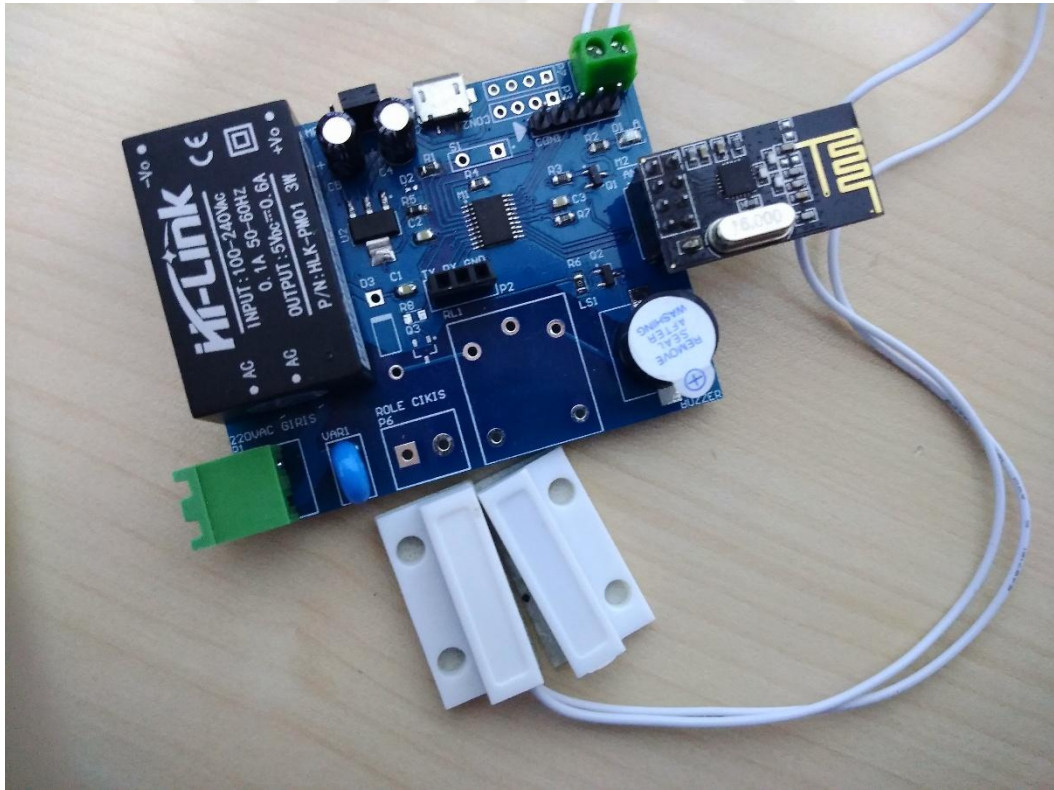


Şekil 4.15 Kapı Kontrol Devresi Akış Diyagramı



Şekil 4.16 Kapı Kontrol Devresi Şeması

Kapı Kontrol Devresi, doğalgaz kaçak kontrol devresi ile aynı PCB'yi kullanmaktadır. Sebebi şu ki, PCB bastırma maliyetleri fazla olduğu için mümkün olduğunca, ortak fonksiyonlara sahip olan cihazları birleştirerek tek kart üzerinde hepsini gerçekleştirme hedefi ile, büyük miktarda tasarruf sağladık.

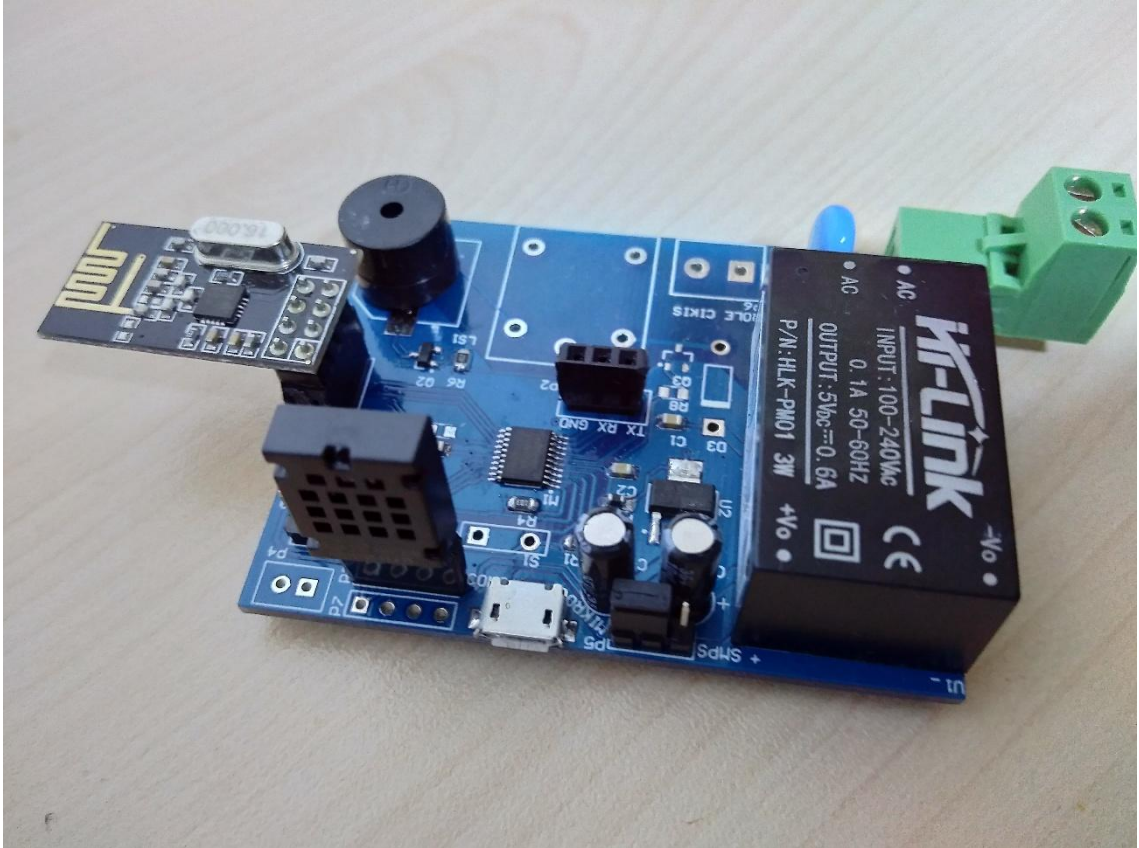


Şekil 4.17 Kapı Kontrol Devresi son hali (üstten görünüm)

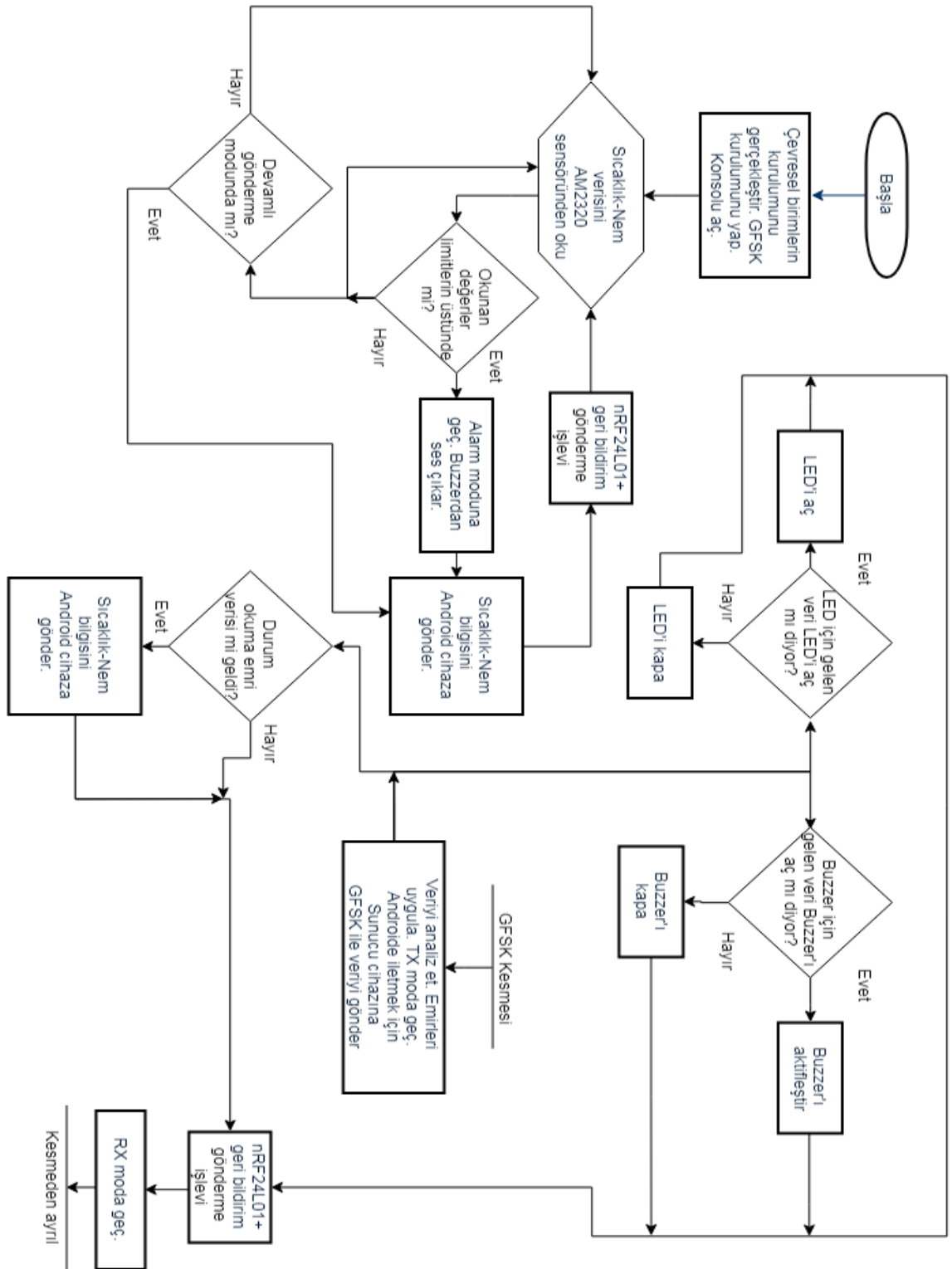
4.4 Sıcaklık-Nem Kontrol Devresi

Sıcaklık-Nem Sensörü okunan verileri Android kullanıcılarına iletmektedir. Burada devamlı iletme özelliği de mevcuttur. Bu özellik aktifleşirse ayarlanabilir saniye cinsinden periyotlar ile devre Android cihazlara veri gönderir. Saniyesi ve devamlı gönderme özelliği ayarlanabilirdir. Ayrıca herhangi bir alarm durumunda yani daha önceden belirlenmiş sıcaklık ve nem limitleri aşılsa, alarm moduna girilir. Alarm modunda da devamlı iletme özelliği açık olmasına gerek kalmadan Android cihazlara veri gönderilir. Gönderilen veri alarm bilgisi taşıdığından Android cihazlarda uyarı verilmesi sağlanır.

Sıcaklık-Nem Kontrol Devresi de, doğalgaz kaçak kontrol devresi ile aynı PCB'yi kullanmaktadır. Sebebi yukarıda kapı kontrol devresinde açıklandığı gibi yine tasarruf amaçlıdır.



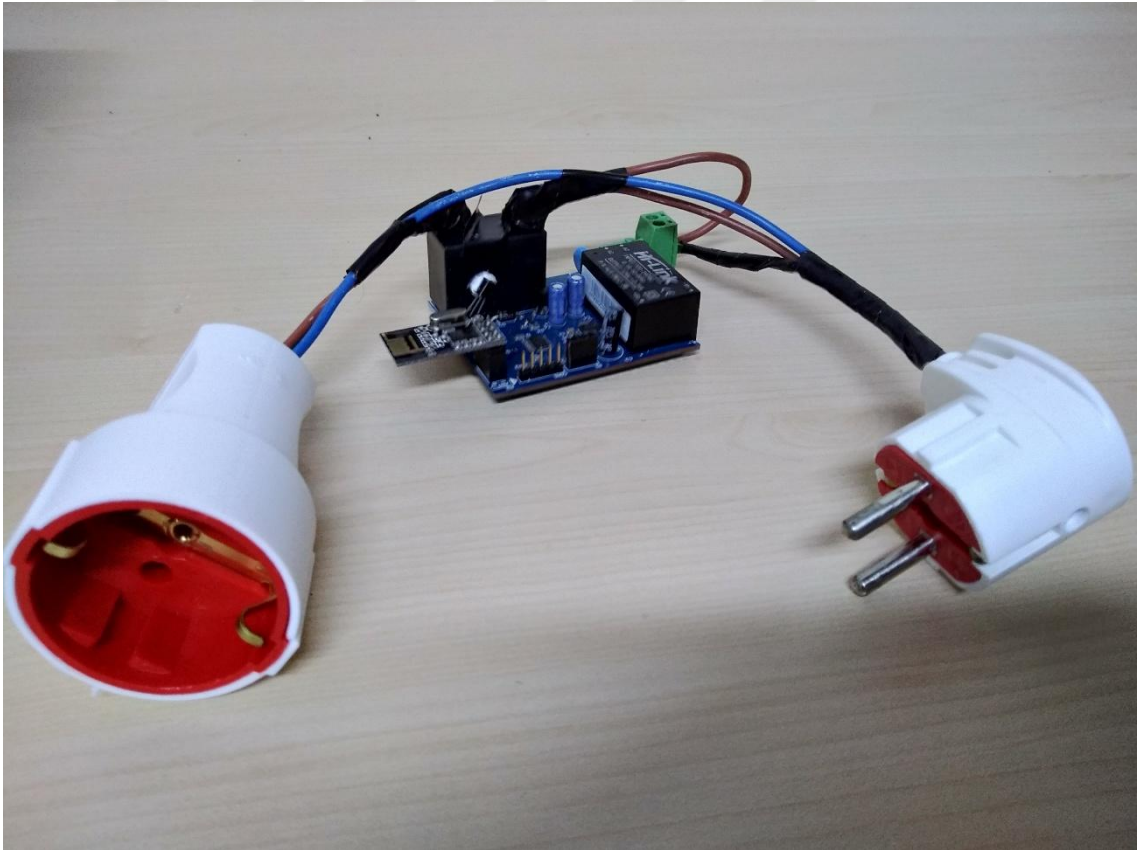
Şekil 4.18 Sıcaklık-Nem Kontrol Devresi son hali (üstten görünüm)



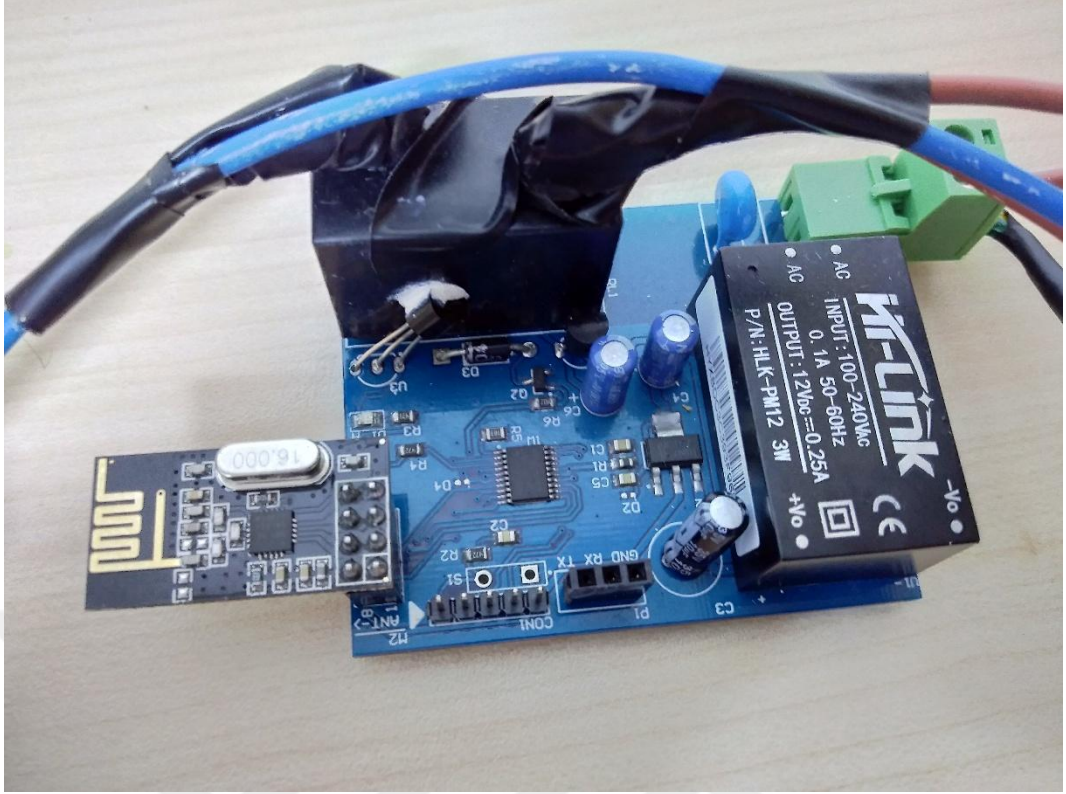
Şekil 4.19 Sıcaklık-Nem Devresi Akış Diyagramı

4.5 Ütü Kontrol Devresi

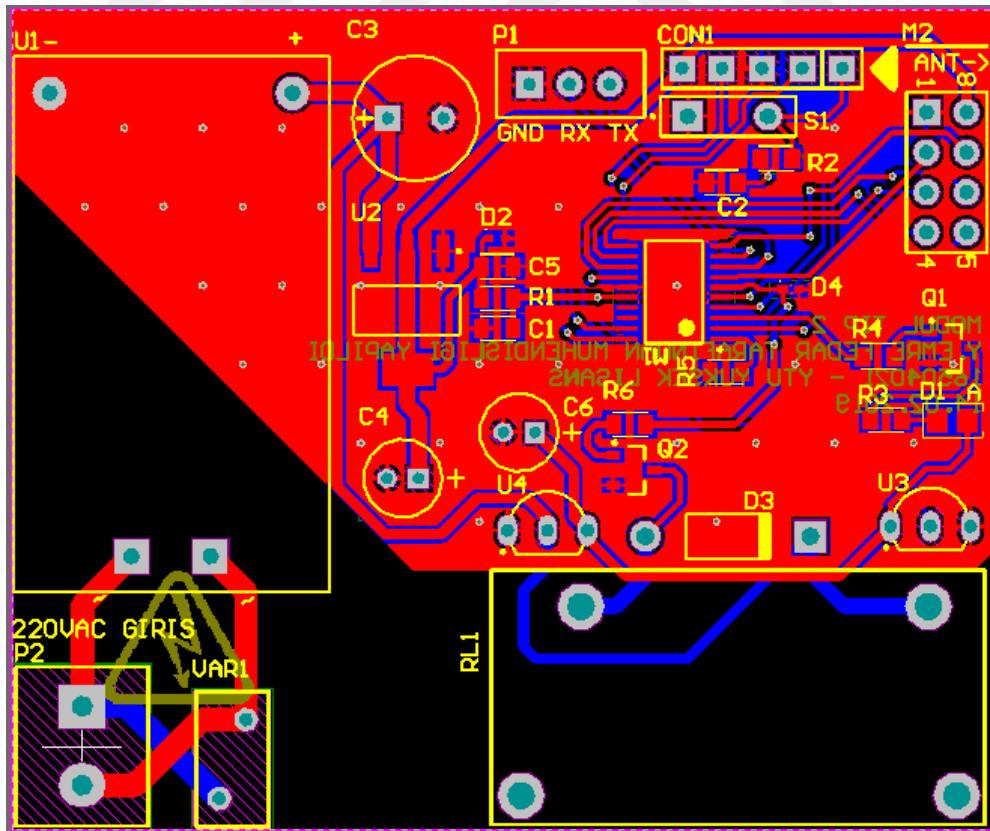
Ütünün bağlanacağı prizden önce takılarak kullanımı gerçekleştirilir. Eğer ütünün uzun bir süre kullanımı gerçekleşmez ise kontrol modülü ütüye giden enerjiyi keser. Bu işlem ütü üzerindeki sıcaklık sensörü ile anlaşılmaktadır. Sürekli kullanılan bir ütü sürekli akım çekecektir. Bu durumda rölenin ısınması kaçınılmazdır. Röle soğuduğunda ütü kullanımının bitirildiği anlaşılmaktadır. Ütünün normal olmayan bir davranışı aynı şekilde Android kullanıcılarını bilgilendirmektedir, Örneğin; ütünün sürekli akım çekmesi, bu rölede aşırı ısınmaya yol açıp tedbir amaçlı ütüye giden elektriğin kesilmesiyle sonuçlanır. Normal olarak periyodik hareketlilik ütü devresinin çalışmasına devam etmesini sağlar. Ütü kontrol devresinin resmi, sadece devresinin resmi, Altium çizimi, PCB resmi, genel şeması ve akış diyagramı aşağıda gösterilmiştir.



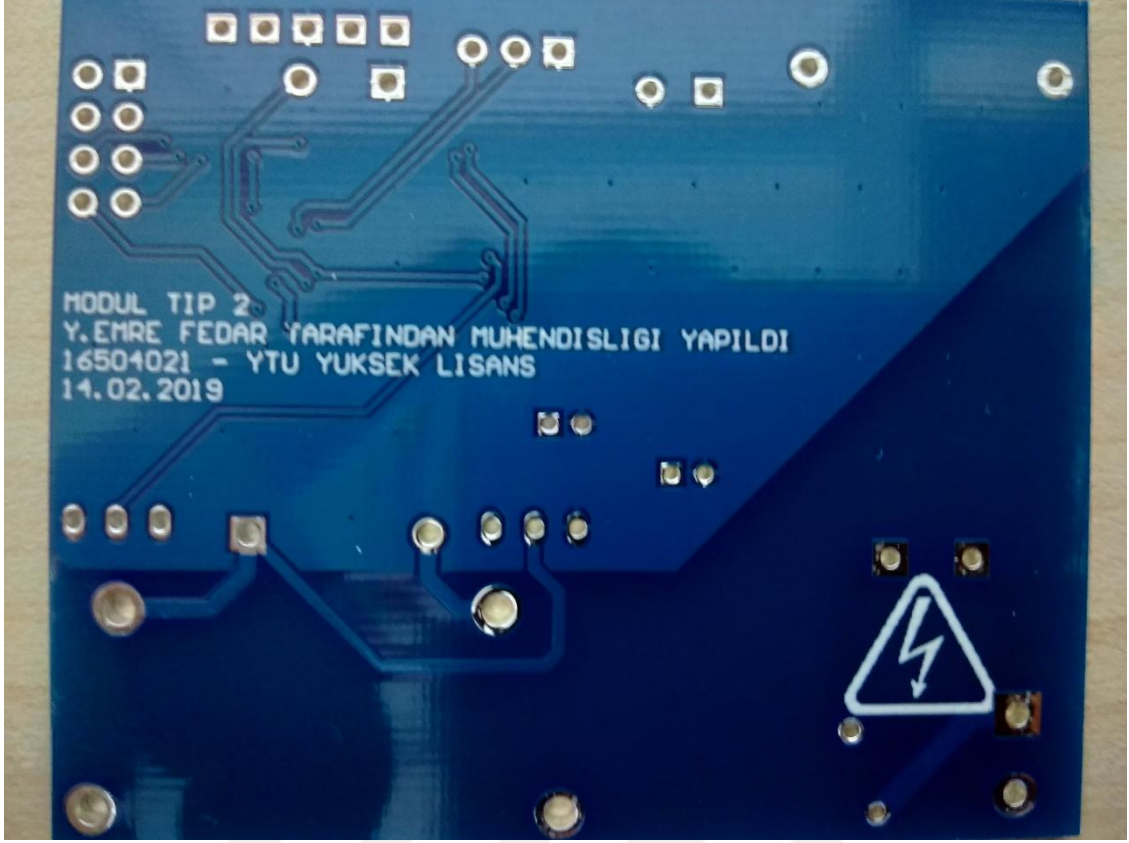
Şekil 4.20 Ütü Kontrol Devresinin resmi



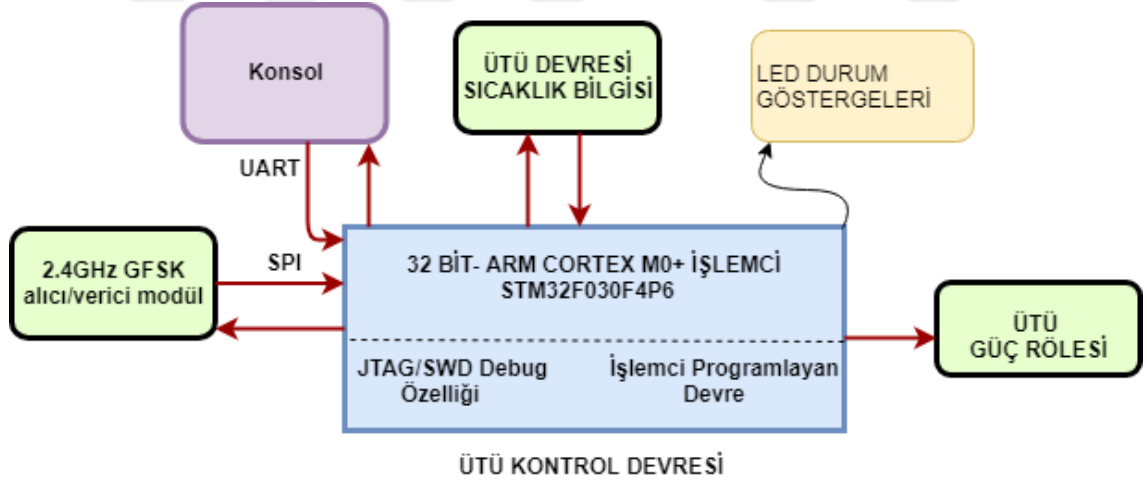
Şekil 4.21 Ütü Kontrol Devresinin resmi



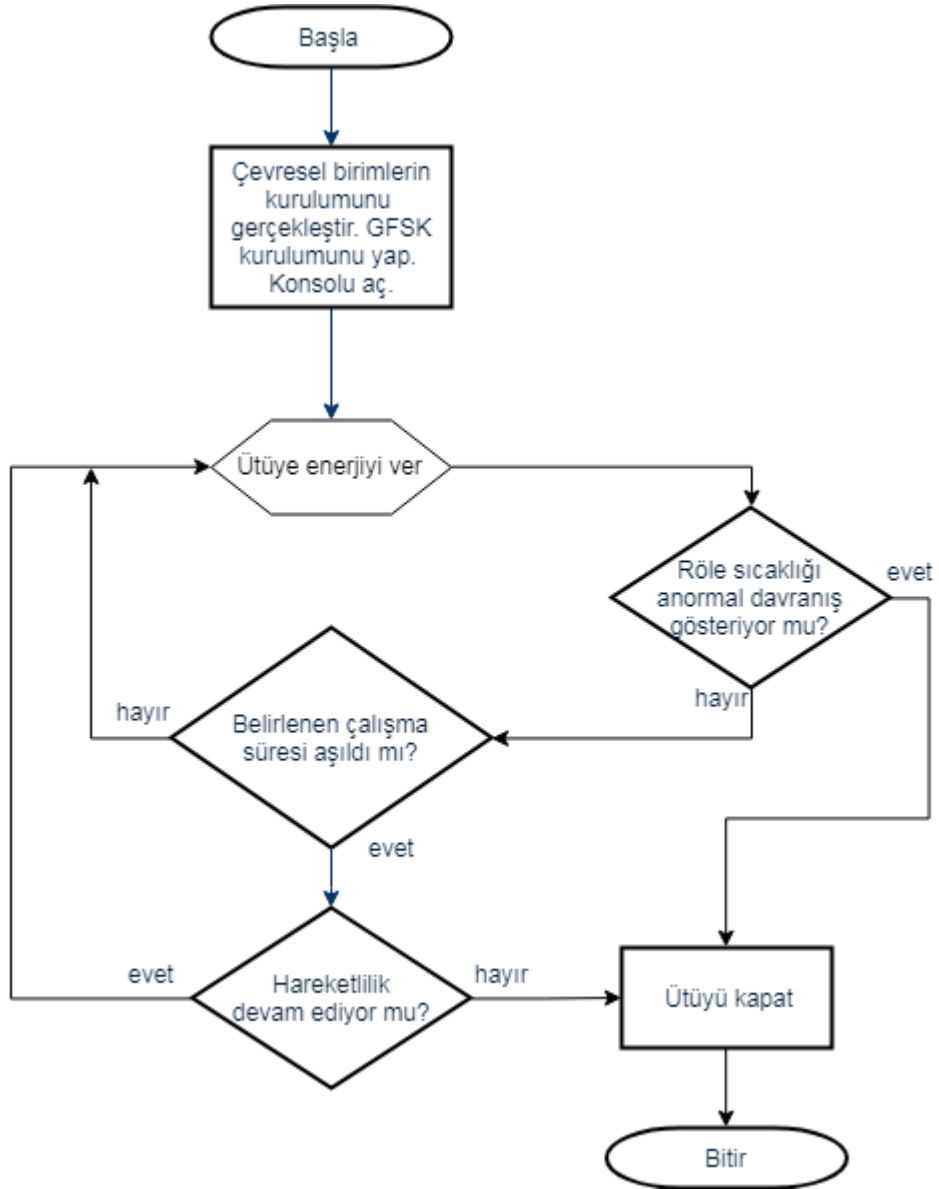
Şekil 4.22 Ütü Devresinin Altium'da çizilmiş hali



Şekil 4.23 Ütü Devresinin PCB'sinin bastırılmış hali



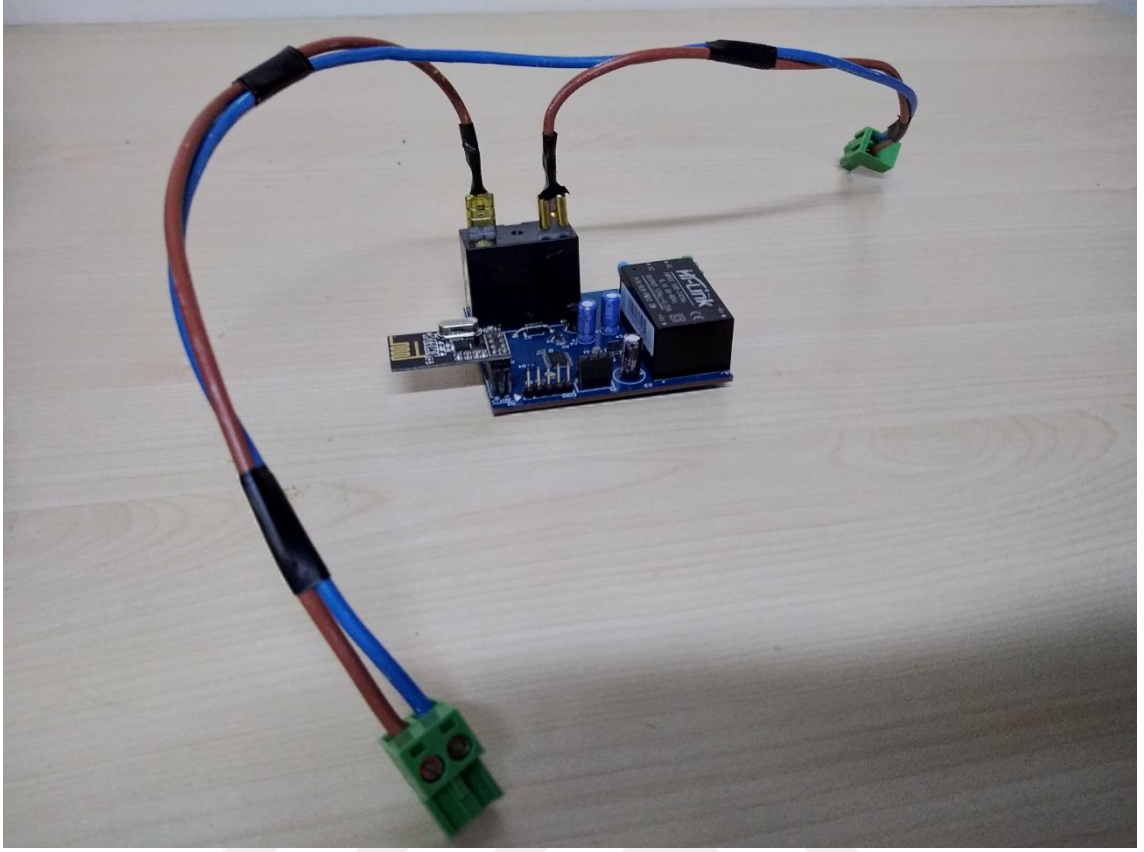
Şekil 4.24 Ütü Kontrol Devresinin genel şeması



Şekil 4.25 Ütü Kontrol Devresi Akış Diyagramı

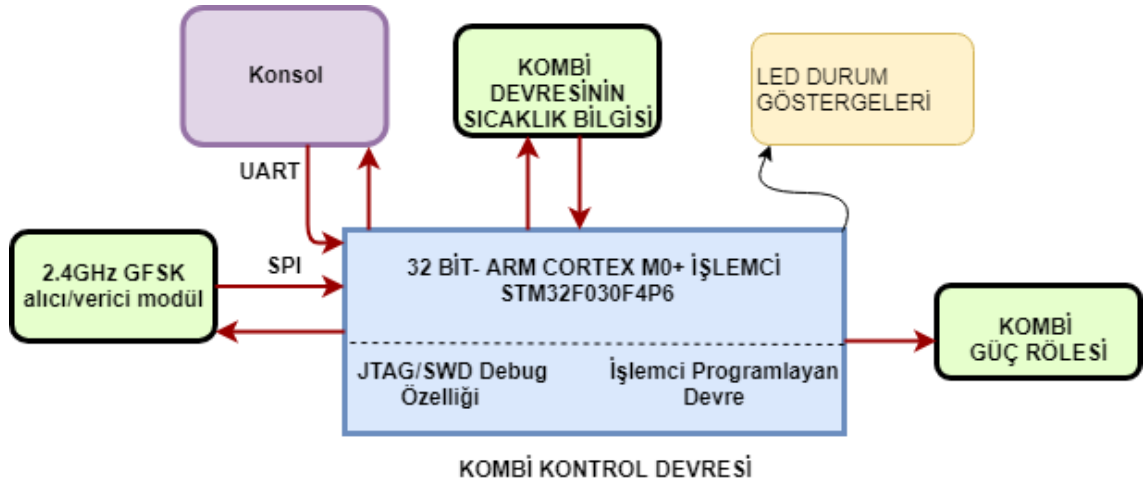
4.6 Kombi Kontrol Devresi

Doğalgaz kombisi kontrol modülü, kombi sigortası ile kombi arasında bulunan güç hattına seri bağlanmaktadır. Bu modül kombinin enerjisinin kesilip, verilmesini sağlar. Bu fonksiyon evde olunmadığı zamanda kombi kapatılmamışsa, bu erişimi sağlayarak kapatıp açmayı, eğer kapatılmışsa eve gelmeden önce açılarak evin ısınmasını sağlar. Kapatma açma işleminin kısa sürede çok sık olması kombiyi bozabileceğinden 5dk'da bir açma kapama isteği gönderilecek şekilde ayarlanmıştır.

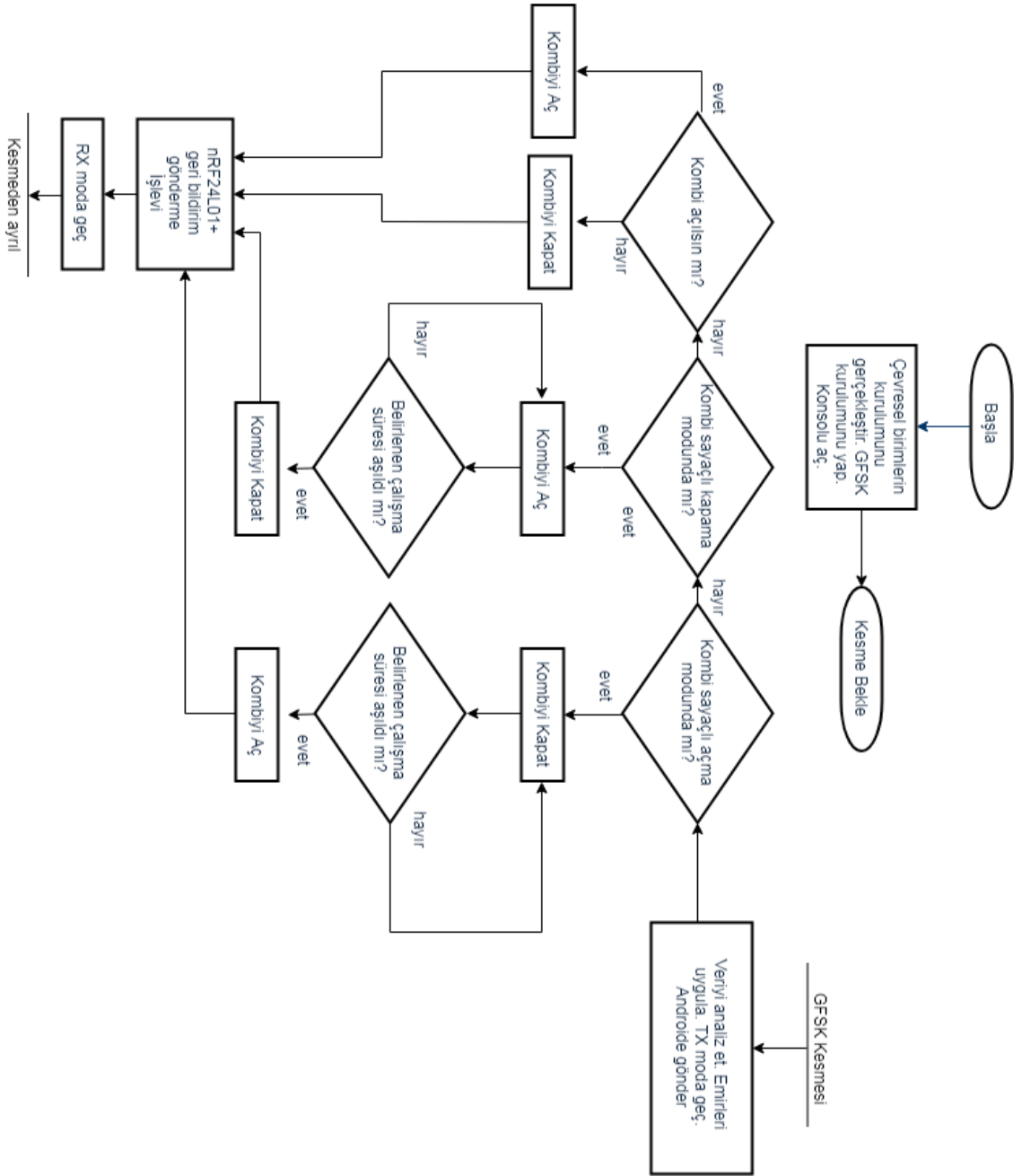


Şekil 4.26 Kombi Kontrol Devresi

Kombi Kontrol Devresi’de modül tip 2’dir. Ütü kontrol kartıyla aynı PCB’ye sahiptir. Devreleri aynı olmakla birlikte kodlanması diğerlerinden farklıdır.



Şekil 4.27 Kombi Kontrol Devresi genel şeması

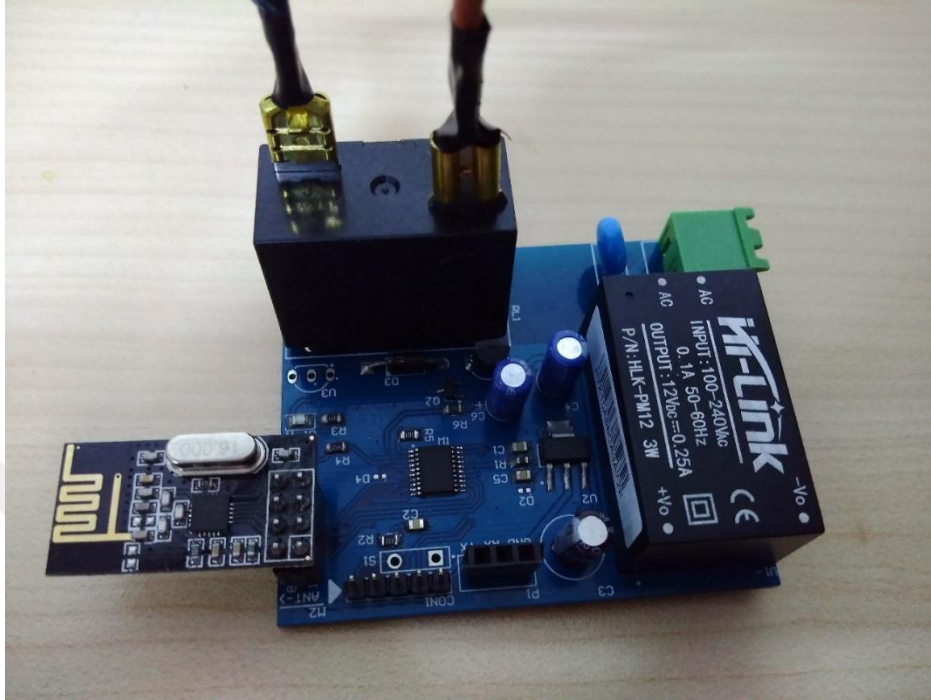


Şekil 4.28 Kombi Kontrol Devresi Akış Diyagramı

4.7 Klima Kontrol Devresi

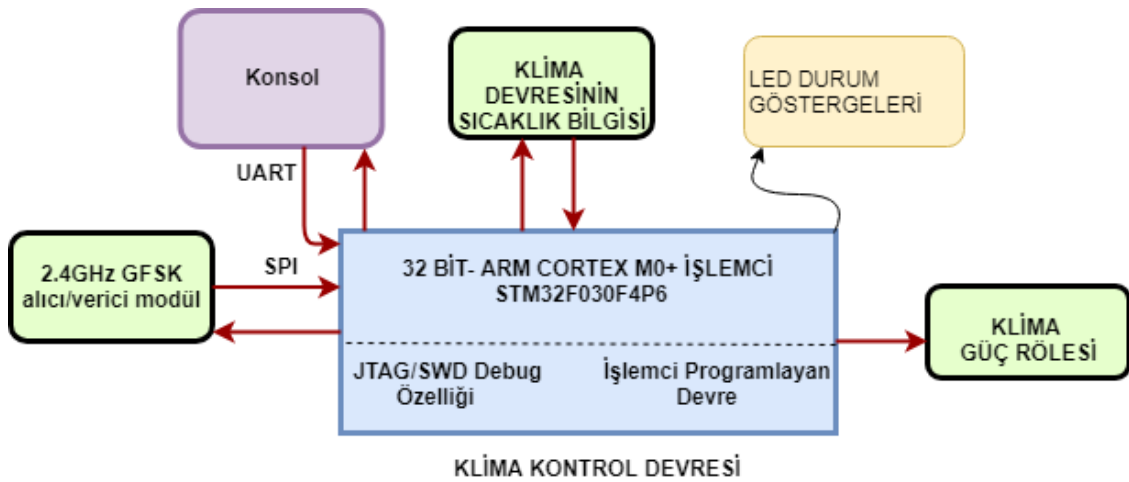
Klima sigortasından sonraki güç hattına seri olarak bağlanan kontrol modülü, klimaya giden gücü röleler vasıtası ile kesmekte, rölenin açık olup olmadığı ve rölenin sıcaklığının ne seviyede olduğunu Android kullanıcılarına iletmektedir. Bu modülde aynı şekilde klimanın enerjisinin kesilip, verilmesini sağlar. Aynı fonksiyonlara sahip bu devre, evde olunmadığı zamanda klima kapatılmamışsa, bu erişimi sağlayarak kapatıp açmayı, eğer kapatılmışsa eve gelmeden önce açılarak

evin ısınmasını sağlar. Kapatma açma işleminin kısa sürede çok sık olması klimayı olumsuz etkileyebileceğinden 1dk'da bir açma kapama isteği gönderilecek şekilde ayarlanmıştır.



Şekil 4.29 Klima Kontrol Devresi

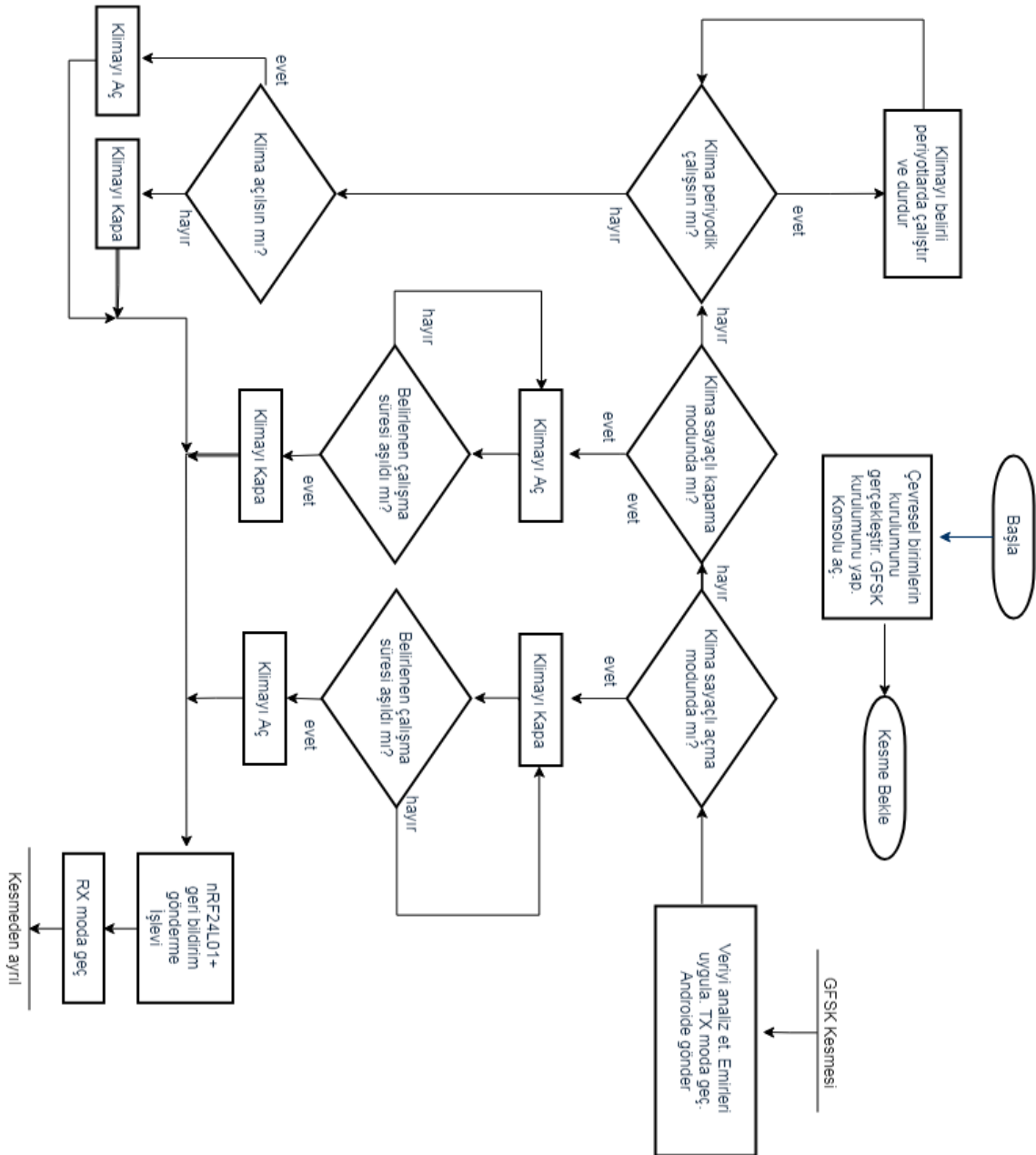
Klima Kontrol Devresi'de modül tip 2'dir. Ütü kontrol kartıyla aynı PCB'ye sahiptir. Devreleri aynı olmakla birlikte kodlanması diğerlerinden farklıdır. Aynı PCB'ye sahip olma sebepleri daha önce doğalgaz kaçak kontrol devresinde ifade edildiği gibi, ekonomik sebepler ve ortak fonksiyonları içermesi sebebiyle tasarruf amaçlıdır.



Şekil 4.30 Klima Kontrol Devresinin genel şeması

Klima kontrolünde aynı şekilde Androidde belirlenen bir süre çerçevesinde açma ve kesme işlemlerini yapmaktadır. Belirli bir süre sonra aç, belirli bir süre sonra kes ve istediğin zaman aç veya kes, özellikleri mevcuttur. Klima kontrolünde opsiyonlu olarak belirli periyotlarda belirli sürelerde sürekli çalışmada mevcuttur. Yani eğer sürekli çalışma moduna ayarlandıysa, Örneğin; 1 saat periyodunda 10dk çalış, 50dk çalışma ya da 30dk çalış, 30dk çalışma gibi işlevler uygulanabilmektedir. Periyotlar ve zaman aralıkları değiştirilebilmektedir.

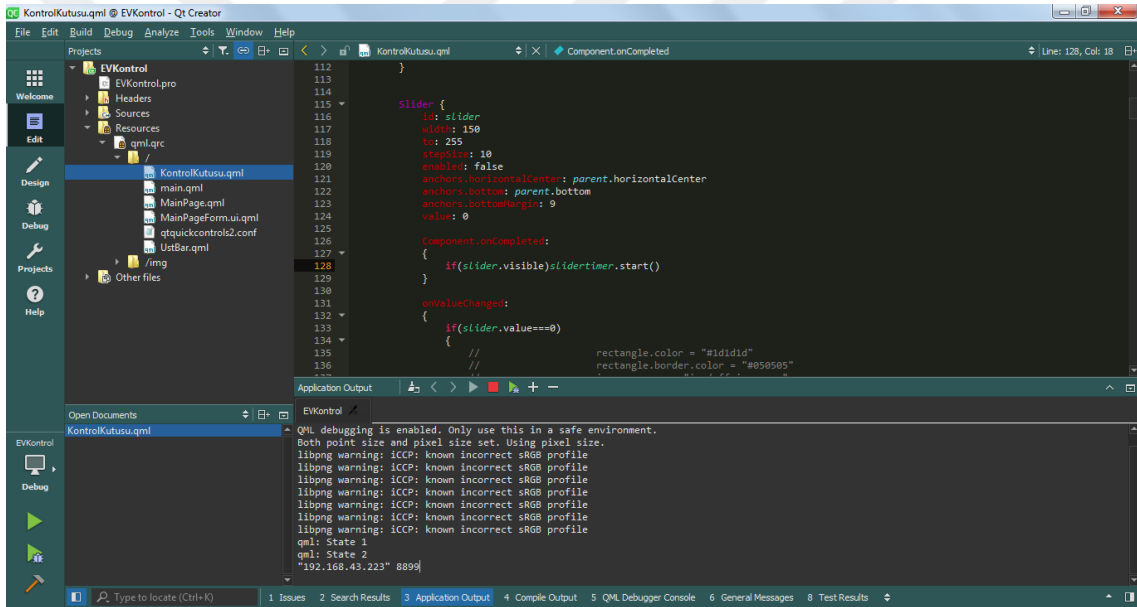
Klima kontrol devresinin akış diyagramı aşağıda mevcuttur.



Şekil 4.31 Klima Kontrol Devresi Akış Diyagramı

5.1 Android Yazılım Geliştirme Arayüzü QtCreator

Qt Creator, Qt GUI uygulama geliştirme çerçevesi için SDK'nın bir parçası olan platformlar arası bir C ++, JavaScript ve QML tümleşik geliştirme ortamıdır. Görsel bir hata ayıklayıcı ve entegre bir GUI düzeni ve form tasarımcısı içerir. Editörün özellikleri sözdizimi vurgulama ve otomatik tamamlama içerir.



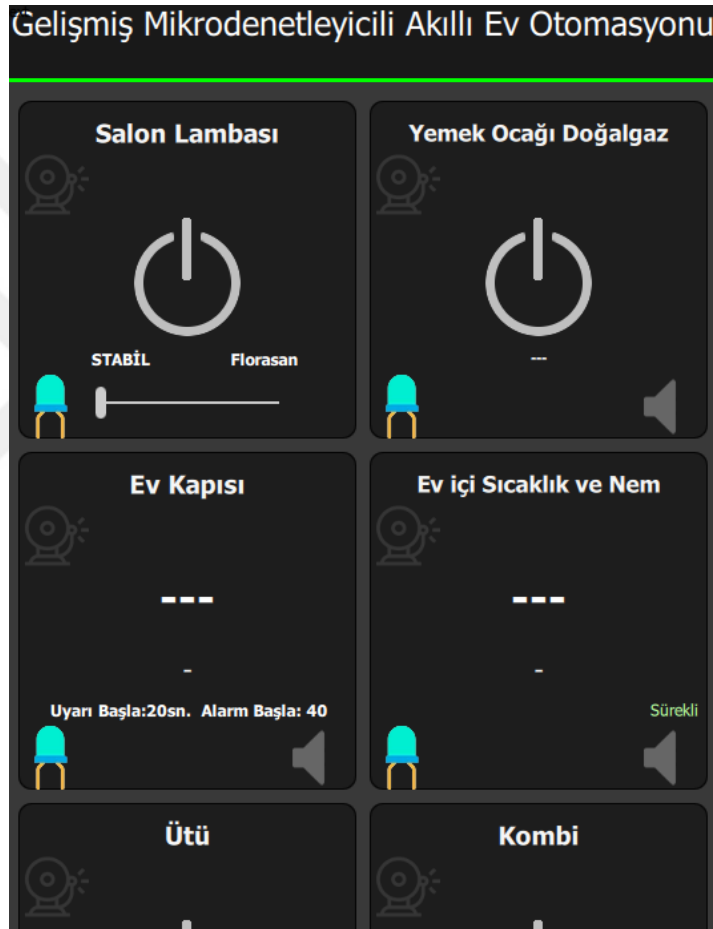
Şekil 5.1 QtCreator Arayüzü

Qt Creatorda Android yazılımı geliştirmek için QT'nin ihtiyacı olan birçok Android SDK'leri yüklendi. Bu sayede QML ve C++ dilleri üzerinden ARM-v7 kütüphanesi ile Android uygulamaları geliştirme aşamaları gerçekleştirildi.

5.2 Akıllı Ev Otomasyonu Android Yazılımı

Tasarlanan Akıllı Ev Otomasyonu yazılımı ev içindeki cihazların kontrolünü sağlıklı bir biçimde sağlamaktadır. Anlaşılır ve kullanımı gayet basit bir arayüzü olması da pozitif bir etkidir.

Butonlar herbiri bir sınıf olarak tasarlanmış olup, kendine has bir kod bloğu ve parametreleri vardır. Android yazılımına bir veri geldiği zaman buton sınıflarının hepsinde işlenir. Gelen veri hangi butona aitse, o buton üzerinde düzenlemeler yapılır. Buton tıklandığı anda veri şifrelenip gönderilir. Eğer modül cevap vermez ise arayüzde buton güncellenmez. Böylece gönderilen emrin uygulanıp uygulanmadığına dair bildirim alınmış olur. Gönderilen emrin uygulanmamasının başlıca sebebi, cihaz üzerindeki GFSK modülün uzak mesafelerde sinyal seviyesinin yetmemesi hasebiyle iletişim kurulamamasıdır.



Şekil 5.2 Ev Otomasyonu Uygulaması Görseli

Butona basıldığı andan itibaren gönderilen 32 byte emir 16 byte'lara bölünerek şifrelenir ve birleştirilir. Birleştirilen bu byte dizisi C++'ta QT kütüphanesiyle oluşturulmuş olan TCP/IP sokete gönderilir. Soketi dinlemekte olan anakart üzerindeki Wi-Fi modülü gelen 50mir çözer ve güvenlik testine sokar, güvenlik testini geçen veri kontrol cihazlarına gönderilir ve işlem uygulanır. Kontrol cihazı gelen işlemi uygular ve cihazın son halini geri gönderir. Gönderilen veri anakart

üzerinden direk Akıllı Ev Otomasyonu yazılımına gönderilir. Yazılım aldığı veriyi inceler ve uygulandığını doğrularsa emrin gönderildiği butonun son halini günceller.



Şekil 5.3 Kontrol Kutuları

Emir verilen kontrol cihazlarının butonlarının son hali ve içindeki fonksiyonlar alarm simgesine basıldığında veya bir emir gönderildiğinde güncellenir, geri bildirim gelirse butona yansır, gelmezse yine sanki değişmiş gibi arka planda fonksiyonlar çalışır.

Android arayüzündeki verilerin güncelliğide önemlidir. Otomatik olarak belirli periyotlarla Android arayüz güncellenecek şekilde ayarlanabilir, bu isteğe bağlı olarak yazılıma kolayca eklenip sunulabilmektedir.

Bu tez çalışması kapsamında kontrol cihazlarını kontrol eden bir sunucu cihazı, doğalgaz kaçak kontrol, kapı kontrol, sıcaklık nem kontrol, ütü kontrol, klima ve kombi kontrol cihazları olan bir akıllı ev otomasyonu sistemi geliştirilmiştir. Tasarlanan bu gelişmiş ev otomasyonu tanımlanan emirleri başarı ile yerine getirmekte ve kullanıcıya büyük bir rahatlık sunmaktadır. Android üzerinden gönderilen emirlerin uygulanıp geri gönderilmesi 100-200ms civarında sürmekte ve işleyişi bozmamaktadır. Basit anlaşılır Android arayüzü kullanımda büyük kolaylık sağlamaktadır. Hassas elektronik tasarımı cihazın uzun ömürlü olmasını sağlamıştır. Cihazların geri bildirimleri ve alarmları kullanıcıyı şaşırtmamaktadır, olayın ne olduğu bilgisinin kolay kavranmasını sağlamaktadır. Sistemin sağladığı bilgiler kullanıcının ihtiyaçlarını karşılamaktadır.

Kullanılan 32-bit ARM Cortex M0+ işlemciler ile ARM çekirdeğinin sunduğu yüksek hız ve çok fonksiyonluluk, işlemci maliyetini ciddi oranda düşürmekle birlikte yüksek performans kazandırmıştır.

Tasarladığımız bu sistem yeni bir haberleşme paket yapısı sunmakla birlikte, AES-128bit şifreleme ile yüksek güvenlik vaad etmektedir. Haberleşmenin korunmasının yanı sıra farklı cihazların sızmasının kontrolü ve engellemesi de her safhada yapılmıştır. Tabi bu pozitif etkilerin yanında bu kriptoloji hesaplamalarına harcanan gecikmeler çok fazla hissedilmese bile göz ardı edilmemelidir. Sızma ve kontrol cihazı doğrulaması işlemi önemsenmeyecek derecede kısa sürmektedir. Bu gecikmelerin düşük olmasında yüksek hızlı ARM çekirdeğinin büyük faydası görülmüştür.

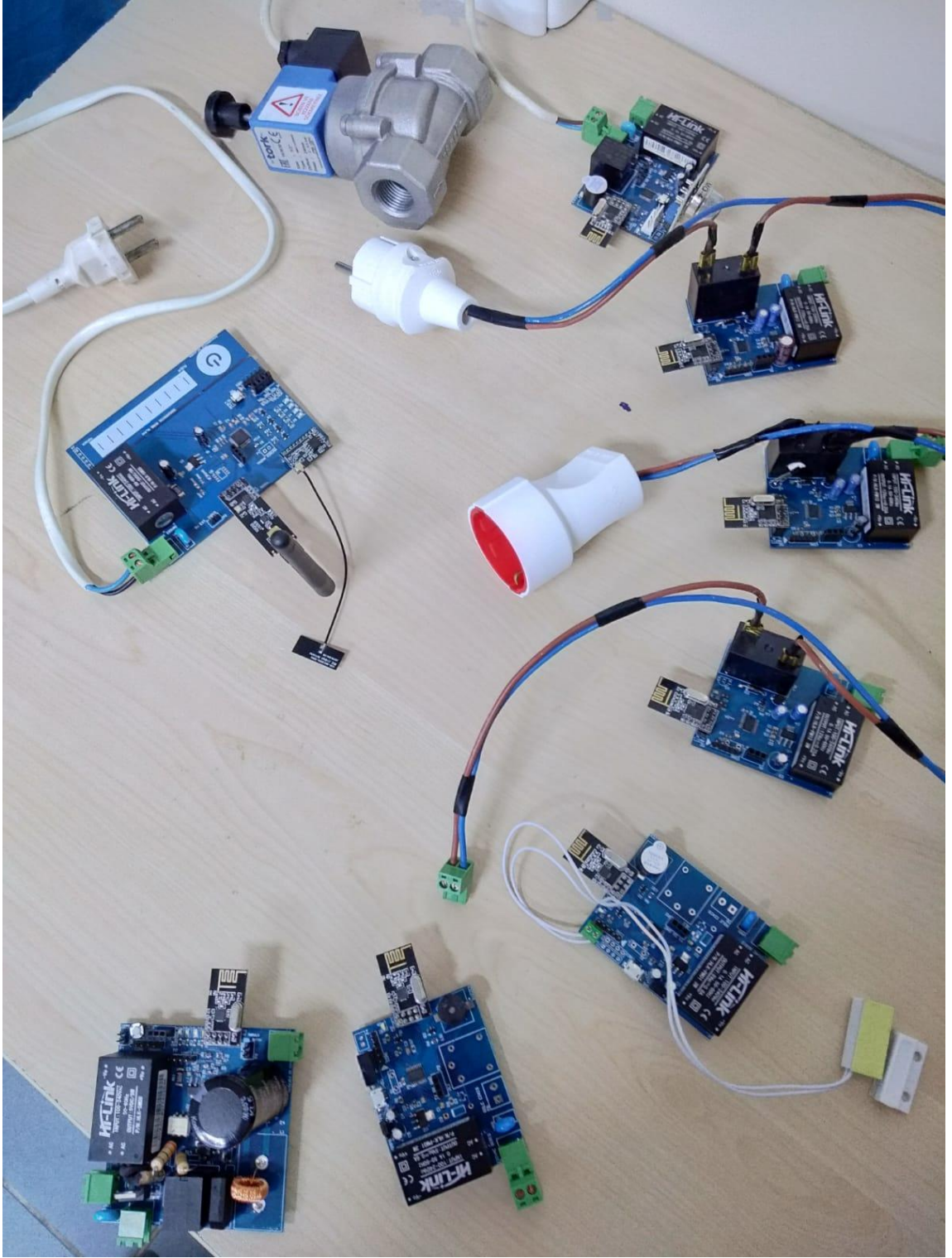
Dimmer mantığına yeni bir yaklaşım kazandırılıp 220 VAC gerilimi, florasen ve LED lambalarda aşırı titreşimden dolayı doğrultmanın faydalı olacağı düşüncesiyle 320 VDC gerilime doğrultulmuştur. Anahtarlama MOS'la yapılırsa 320 VDC'de çok pahalı entegrelerin gerekeceği görülüp, triyak anahtarlamasından faydalanılmıştır. Her ne kadar dim etme işlemi başarılı olsa da, dim edilmiş AC ve DC gerilim verilmesinden dolayı dim edilme özelliği bulunmayan kompakt florasen ve led lambaların kendi devrelerinin çok

fazla ısındığı görülmüştür. Bu tarz lambaların AC veya DC şekilde dim edilmesinin lambayı yormakta olduğu görülmüştür, bu durum lambanın ömrünü azaltmaktadır.

Kurulan ev otomasyonu sistemi Android işletim sistemli akıllı telefon üzerinden rahatlıkla kontrol edilmiş olup, tamamen kablosuz oluşundan dolayı göze hoş gelen bir yapıdadır. Bu sistem sadece elektrik hattı ve önceden kurulmuş ticari modem Wi-Fi ağı ile evin içine kolayca entegre edilebilecek tarzda hazır bir yapıdadır. Seri üretim yapılabilmesi için bazı bir düzenlemeler gereklidir. Yine de seri üretime yatkın tasarımından dolayı yeni sistem ve parametrelerin entegrasyonu çok kolaydır. Android kısmında kontrol edilen cihazlar önceden tanımlı olmuş olsalar bile, kolayca yenileri eklenebilecek şekilde hazırdır. Bu özgün tasarımı ile geliştirilen haberleşme paket yapısı sayesinde.

Sisteme ev içinde hem uzaktan kontrol edilebilecek hemde kendi akıllı algoritmasıyla çalışacak yeni parametreler, sensörler ve kontrol cihazları eklenebilir. Örneğin; perde kontrol cihazı, mutfakta tencere karıştırma cihazı, yemek pişme kontrol cihazı, pencere kontrol cihazları, kettıl kontrol cihazı, mikrodalga kontrol cihazı, fırın kontrol cihazı, televizyon kontrol cihazı, su sayacı kontrol cihazları eklenebilir. Hayati öneme sahip sunucu cihazı ve bazı kontrol cihazları UPS ile elektrik kesilme anında erişime açık tutulabilir.

Genelde ucuz elektronik malzemelerle tasarlanmaya çalışılan bu sistem yaklaşık AR-GE aşamasında 2500-3000TL civarında bir para harcanmıştır. Yaklaşık 1500 aktif geçen saat AR-GE'ye ve yapılan işlemlerin tümüne harcanmıştır. Ek-3'te minimum Maliyet Analizi yapılmıştır. Maliyet Analizine göre AR-GE süresi ve verilen emek hariç, sistemin komponent bazında yaklaşık maliyeti 1233TL civarındadır.



Şekil 6.1 Projedeki bütün cihazların toplu resmi

Ek-1

Cihazların Mikrodenetleyici Kodları

Anakart işlemci kodları

“Main.c”

```
#include "main.h"
#include "stm32l0xx_hal.h"
#include "touchsensing.h"

/* USER CODE BEGIN Includes */
#include "uart.h"
#include "nrf24.h"
#include "aes.h"
#include "protocol.h"
/* USER CODE END Includes */

/* Private variables -----
-*/
ADC_HandleTypeDef hadc;

I2C_HandleTypeDef hi2c1;

UART_HandleTypeDef hlpuart1;
UART_HandleTypeDef huart1;
UART_HandleTypeDef huart2;

SPI_HandleTypeDef hspi2;

TIM_HandleTypeDef htim6;

TSC_HandleTypeDef htsc;

/* USER CODE BEGIN PV */
/* Private variables -----
-*/
#define UART_INFO_ACTIVATED 1

// Buffer to store a payload of maximum width
uint8_t nRF24_payload[32] = {0,0,0,4,//destination modul ID
    0,0,0,0,//source modulID
    0,0,0,0,0,0,//RTC
    4,//Modul type
    1,//1:write enab, 0:read mode
    8,0,//command 1
    0,0,//command 2
    0,0,//command 3
    0,0,//command 4
    0,0,0,//16-bit answer
    255,//LED
```



```

        0,0,0,0};
// Length of received payload
uint8_t payload_length= 32;
uint8_t IRQHandled=0;

uint8_t pipeadres=0;

uint8_t wifuartirq=0;
uint8_t wifuartbuffer[32];
uint8_t consolebuffer[1];
uint8_t consoletrigger=0;
uint8_t consolecommand[32];
uint8_t consolecounter=0;
uint8_t ordersended=0;
#define STRING_SAYISI 31
#define MAX_STRING_SIZE 40
char arr[STRING_SAYISI][MAX_STRING_SIZE] =
{ "ledkapa0000000",
  "ledyanipsonsun",
  "ledac0000000000",
  "buzzerkapa0000",
  "buzzerac0000000",
  "sicakliknem000",
  "dimmermod000000",
  "sicaknemmod000",
  "dogalgazmod000",
  "kapimod00000000",
  "utumod000000000",
  "kombimod000000",
  "klimamod000000",
  "dim250000000000",
  "dim500000000000",
  "dim750000000000",
  "dim100000000000",
  "gazroleac00000",
  "gazrolekapa000",
  "uturoleac00000",
  "uturolekapa000",
  "kapidurum00000",
  "kapialarmcrpan",
  "kapiuyaricrpan",
  "dimz cstbl00000",
  "dimzcnormal000",
  "halojen0000000",
  "florasan000000",
  "led000000000000",
  "dim000000000000",
  "dimiki00000000",

};

uint8_t buffermodulID=255;
uint8_t buffermodultype=255;
uint8_t bufferwritemode=1;
uint8_t moddegisikligi=0;
uint8_t gonder=0;
/* USER CODE END PV */

```

```

/* Private function prototypes -----
-*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC_Init(void);
static void MX_I2C1_Init(void);
static void MX_LPUART1_UART_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_SPI2_Init(void);
static void MX_TIM6_Init(void);
static void MX_TSC_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----
-*/
void nRF_SendPayload(void);
void nRF_SenderBuffReset(void);
void AndroidSifrele(void);
void AndroidSifreCoz(void);
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 *
 * @retval None
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
    -----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_ADC_Init();
    MX_I2C1_Init();

```

```

MX_LPUART1_UART_Init();
MX_USART1_UART_Init();
MX_USART2_UART_Init();
MX_SPI2_Init();
MX_TIM6_Init();
MX_TSC_Init();
MX_TOUCHSENSING_Init();
/* USER CODE BEGIN 2 */
HAL_GPIO_WritePin(LED2_GPIO_Port,LED2_Pin, GPIO_PIN_SET);
HAL_TIM_Base_Start_IT(&htim6);

if (UART_INFO_ACTIVATED)
{
    UART_SendStr("\r\nCommand Set created by Y.Emre FEDAR. YTU -
TEZLI Y.L. 16504021");
    UART_SendStr("\r\nSystem started.STM32L052");
    UART_SendStr("\r\nGELISMIS MIKRODENETLEYICILI AKILLI EV
OTOMASYONU SERVER CIHAZI");
    UART_SendStr("\r\nnRF24L01 check: ");
}
do {

    if (nRF24_Check())
    {}
    else
    {
        UART_SendStr("CONNECTING TO NRF24L01.\r\n");
    }
    HAL_Delay(500);
    //UART_SendHex32(HAL_SPI_GetError(&hspi1));
}while (!nRF24_Check());

UART_SendStr("OK.\r\n");
// Initialize the nRF24L01 to its default state
nRF24_Init();
nRF_RX_Init();
if (UART_INFO_ACTIVATED)
{
    UART_SendStr("Ready. Looking for IRQ.\r\n");
}

UART_SendStr("\r\nroot>");
HAL_UART_Receive_IT(&hlpuart1,consolebuffer,1);
HAL_UART_Receive_IT(&huart1,wifiuartbuffer,32);
HAL_GPIO_WritePin(WIFI_RESET_GPIO_Port,WIFI_RESET_Pin,GPIO_PIN_SET);
/*****
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(consoletrigger)
    {
        if (consolebuffer[0]==13)
        {
            for(int i=0;i<STRING_SAYISI;i++)
            {

```

```

        if
        ((memcmp(arr[i],consolecommand,consolecounter)==0)&&(consolecounter>0))
        {
            if(i==0)
            {
                UART_SendStr("\r\nLEDKapa.\r\n");

                ordersended=1;
                nRF_SenderBuffReset();
                nRF24_payload[16]=2;
                nRF24_payload[17]=0;
            }
            else if(i==1)
            {
                UART_SendStr("\r\nLEDYanipSondur.\r\n");
                ordersended=1;
                nRF_SenderBuffReset();
                nRF24_payload[16]=2;
                nRF24_payload[17]=1;
            }
            else if(i==2)
            {
                UART_SendStr("\r\nLEDAc.\r\n");
                ordersended=1;
                nRF_SenderBuffReset();
                nRF24_payload[16]=2;
                nRF24_payload[17]=255;
            }
            else if(i==3)
            {
                UART_SendStr("\r\nBuzzerKapa.\r\n");
                ordersended=1;
                nRF_SenderBuffReset();
                nRF24_payload[16]=7;
                nRF24_payload[17]=0;
            }
            else if(i==4)
            {
                UART_SendStr("\r\nBuzzerAc.\r\n");
                ordersended=1;
                nRF_SenderBuffReset();
                nRF24_payload[16]=7;
                nRF24_payload[17]=1;
            }
            else if(i==5)
            {
                UART_SendStr("\r\nSicaklikNemDegeri.\r\n");
                ordersended=1;
                nRF_SenderBuffReset();
                nRF24_payload[16]=8;
            }
            else if(i==6)
            {
                UART_SendStr("\r\nDimmer
modu.\r\n");

```

```

modu.\r\n");

//ordersended=1;
moddegisikligi=1;
nRF_SenderBuffReset();
buffermodulID=1;//modul ID
buffermodultype=1;//Modul type
//nRF24_payload[16]=8;
}
else if(i==7)
{
    UART_SendStr("\r\nSicaklikNem

//ordersended=1;
moddegisikligi=1;
nRF_SenderBuffReset();
buffermodulID=4;//modul ID
buffermodultype=4;//Modul type
//nRF24_payload[16]=8;
}
else if(i==8)
{
    UART_SendStr("\r\nDogalgaz

//ordersended=1;
moddegisikligi=1;
nRF_SenderBuffReset();
buffermodulID=2;//modul ID
buffermodultype=2;//Modul type
//nRF24_payload[16]=8;
}
else if(i==9)
{
    UART_SendStr("\r\nKapiKontrol

//ordersended=1;
moddegisikligi=1;
nRF_SenderBuffReset();
buffermodulID=3;//modul ID
buffermodultype=3;//Modul type
//nRF24_payload[16]=8;
}
else if(i==10)
{
    UART_SendStr("\r\nUtuKontrol

//ordersended=1;
moddegisikligi=1;
nRF_SenderBuffReset();
buffermodulID=5;//modul ID
buffermodultype=5;//Modul type
//nRF24_payload[16]=8;
}
else if(i==11)
{
    UART_SendStr("\r\nKombiKontrol

//ordersended=1;
moddegisikligi=1;
nRF_SenderBuffReset();
buffermodulID=6;//modul ID

```

```

        buffermodultype=6;//Modul type
        //nRF24_payload[16]=8;
    }
    else if(i==12)
    {
        UART_SendStr("\r\nKlimaKontrol
modu.\r\n");

        //ordersended=1;
        moddegisikligi=1;
        nRF_SenderBuffReset();
        buffermodulID=7;//modul ID
        buffermodultype=7;//Modul type
        //nRF24_payload[16]=8;
    }
    else if(i==13)
    {
        UART_SendStr("\r\nDim Ayari
%25.\r\n");

        ordersended=1;
        nRF_SenderBuffReset();
        nRF24_payload[16]=1;
        nRF24_payload[17]=25;
    }
    else if(i==14)
    {
        UART_SendStr("\r\nDim Ayari
%50.\r\n");

        ordersended=1;
        nRF_SenderBuffReset();
        nRF24_payload[16]=1;
        nRF24_payload[17]=122;
    }
    else if(i==15)
    {
        UART_SendStr("\r\nDim Ayari
%75.\r\n");

        ordersended=1;
        nRF_SenderBuffReset();
        nRF24_payload[16]=1;
        nRF24_payload[17]=180;
    }
    else if(i==16)
    {
        UART_SendStr("\r\nDim Ayari
%100.\r\n");

        ordersended=1;
        nRF_SenderBuffReset();
        nRF24_payload[16]=1;
        nRF24_payload[17]=255;
    }
    else if(i==17)
    {
        UART_SendStr("\r\nGazRolesiAc.\r\n");

        ordersended=1;
        nRF_SenderBuffReset();
        nRF24_payload[16]=5;
        nRF24_payload[17]=1;
    }
}

```

```

else if(i==18)
{
UART_SendStr("\r\nGazRolesiKapa.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=5;
nRF24_payload[17]=0;
}
else if(i==19)
{
UART_SendStr("\r\nUtuRolesiAc.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=9;
nRF24_payload[17]=1;
}
else if(i==20)
{
UART_SendStr("\r\nUtuRolesiKapa.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=9;
nRF24_payload[17]=0;
}
else if(i==21)
{
UART_SendStr("\r\nKapiDurum.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=6;
nRF24_payload[17]=0;
}
else if(i==22)
{
UART_SendStr("\r\nKapiAlarmCarpani.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=19;
nRF24_payload[17]=10;
}
else if(i==23)
{
UART_SendStr("\r\nKapiUyariCarpani.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=20;
nRF24_payload[17]=10;
}
else if(i==24)
{
UART_SendStr("\r\nZC_STBL.\r\n");
ordersended=1;

```

```

nRF_SenderBuffReset();
nRF24_payload[16]=21;
nRF24_payload[17]=0;
}
else if(i==25)
{
UART_SendStr("\r\nZC_NORMAL.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=21;
nRF24_payload[17]=1;
}
else if(i==26)
{
UART_SendStr("\r\nHALOJEN
TIP.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=22;
nRF24_payload[17]=0;
}
else if(i==27)
{
UART_SendStr("\r\nFLORASAN.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=22;
nRF24_payload[17]=1;
}
else if(i==28)
{
UART_SendStr("\r\nLED.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=22;
nRF24_payload[17]=2;
}
else if(i==29)
{
UART_SendStr("\r\nDim %0.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=1;
nRF24_payload[17]=0;
}
else if(i==30)
{
UART_SendStr("\r\nDim %2.\r\n");
ordersended=1;
nRF_SenderBuffReset();
nRF24_payload[16]=1;
nRF24_payload[17]=2;
}
nRF24_payload[3]=buffermodulID;//modul
ID
nRF24_payload[14]=buffermodultype;//Modul type

```



```

nRF24_payload[15]=bufferwritemode;//WriteMode
    }
    }
    if(ordersended==0)
    {
        if(!moddegisikligi)UART_SendStr("\r\nBilinmeyen Komut.");
    }
    else
    {
        UART_SendChar('#');
        for(int i=0;i<32;i++)
        {
            UART_SendHex8(nRF24_payload[i]);
            UART_SendChar('-');
        }
        UART_SendChar('#');
        PayloadSifrele();
        nRF_SendPayload();
        PayloadSifreCoz();
    }
    ordersended=0;
    consolecounter=0;
    for(int i=0;i<10;i++)
    {
        consolecommand[i]=0;
    }
    UART_SendStr("\r\nroot>");
}
else
{
    consolecommand[consolecounter]=consolebuffer[0];
    consolecounter++;
    UART_SendChar(consolebuffer[0]);
}

if(consolecounter>15)consolecounter=0;
HAL_UART_Receive_IT(&hlpuart1,consolebuffer,1);
consoletrigger=0;
}
if(IRQHandled)
{
    if (nRF24_GetStatus_RXFIFO() !=
nRF24_STATUS_RXFIFO_EMPTY)
    {
        if(UART_INFO_ACTIVATED){UART_SendStr("RX_IRQ.\r\n");}
        nRF24_RX_IRQCallBack();
        HAL_UART_Transmit_IT(&huart1,nRF24_payload,32);

        PayloadSifreCoz();
        for (int i=0;i<32;i++)
        {
            UART_SendHex8(nRF24_payload[i]);

```

```

        UART_SendChar(' ');
    }
    UART_SendStr("#\r\n");
    UART_SendStr("Telefona gönderildi.");
    //KOMUTLAR FONKSIYONU GELECEK.
}
IRQHandled=0;
}

if(wifiuartirq)
{
    AndroidSifreCoz();
    pipeadresi=wifiuartbuffer[3];
    if(pipeadresi>5)pipeadresi=5;
    pipeadresi=pipeadresi - 1;
    AndroidSifrele();

    nRF_TX_Init();

    for(int a=0;a<32;a++)nRF24_payload[a]=wifiuartbuffer[a];
    HAL_UART_Receive_IT(&huart1,wifiuartbuffer,32);
    nRF_TX_Send();

    nRF24_FlushTX();
    nRF_RX_Init();

    wifiuartirq=0;
}
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

}
/* USER CODE END 3 */

}
void HAL_TSC_ConvCpltCallback(TSC_HandleTypeDef* htsc)
{
}

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    HAL_GPIO_TogglePin(LED2_GPIO_Port,LED2_Pin);
    HAL_GPIO_TogglePin(LED1_GPIO_Port,LED1_Pin);
    gonder=1;

}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    IRQHandled=1;

}

```

```

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if(huart->Instance==LPUART1)consoletrigger=1;
    else if(huart->Instance==USART1)wifiuartirq=1;

}

void nRF_SendPayload(void)
{
    nRF_TX_Init();

    nRF_TX_Send();
    nRF24_FlushTX();
    nRF_RX_Init();
    if (UART_INFO_ACTIVATED)
    {
        UART_SendStr("Waiting RX IRQ.\r\n");
    }
}

void nRF_SenderBuffReset(void)
{
    for(int i=0;i<32;i++)
    {
        nRF24_payload[i] = 0;
    }
    nRF24_payload[7] = 0xFF;
}

void AndroidSifrele(void)
{
    uint8_t temp[16];
    for (int i=0;i<16;i++)
    {
        temp[i]=wifiuartbuffer[i];//İLK 16 BYTE AYRILIYOR.
    }
    //UART_SendStr("Ilk 16 byte sifreleniyor.\n\r");
    test_encrypt_ecb(temp);//İLK 16 BYTE ÇÖZÜLÜYOR.
    for (int i=0;i<16;i++)
    {
        wifiuartbuffer[i]= temp[i];//İLK 16 BYTE GERİ YÜKLENİYOR.
    }
    //////////////////////////////////////
    for (int i=0;i<16;i++)
    {
        temp[i]=wifiuartbuffer[i + 16];//SON 16 BYTE AYRILIYOR.
    }
    test_encrypt_ecb(temp);//SON 16 BYTE ÇÖZÜLÜYOR.
    for (int i=0;i<16;i++)
    {
        wifiuartbuffer[i + 16]= temp[i];//SON 16 BYTE GERİ YÜKLENİYOR.
    }
    //UART_SendStr("Son 16 byte sifrelendi.\n\r");
}

void AndroidSifreCoz(void)
{
    uint8_t temp[16];

```

```

    for (int i=0;i<16;i++)
    {
        temp[i]=wifuartbuffer[i];//İLK 16 BYTE AYRILIYOR.
    }
    //UART_SendStr("Ilk 16 byte cozuluyor.\n\r");
    test_decrypt_ecb(temp);//İLK 16 BYTE ÇÖZÜLÜYOR.
    for (int i=0;i<16;i++)
    {
        wifuartbuffer[i]= temp[i];//İLK 16 BYTE GERİ YÜKLENİYOR.
    }
    //////////////////////////////////////
    for (int i=0;i<16;i++)
    {
        temp[i]=wifuartbuffer[i + 16];//SON 16 BYTE AYRILIYOR.
    }
    test_decrypt_ecb(temp);//SON 16 BYTE ÇÖZÜLÜYOR.
    for (int i=0;i<16;i++)
    {
        wifuartbuffer[i + 16]= temp[i];//SON 16 BYTE GERİ YÜKLENİYOR.
    }
    //UART_SendStr("Son 16 byte cozuldu.\n\r");
}
/* USER CODE END 4 */

```

Dimmer Cihazı Kodları

“Main.c”

```

#include "main.h"
#include "stm32l0xx_hal.h"

/* USER CODE BEGIN Includes */
#include "uart.h"
#include "protocol.h"
#include "nrf24.h"
/* USER CODE END Includes */

/* Private variables -----
-*/
ADC_HandleTypeDef hadc;

SPI_HandleTypeDef hspi1;

```

```

TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim6;
TIM_HandleTypeDef htim21;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
/* Private variables -----
-*/
uint8_t nRF24_payload[32];

// Length of received payload
uint8_t payload_length= 32;
uint8_t IRQHandled=0;

uint8_t alarmX=0;
uint8_t lm35temp=0;
uint8_t ledstatus;

uint8_t dimmervalue=210;
uint8_t ZCsource=0;//0:stb1,1:normal
uint8_t lamptype=1;//0:halojen,dimmable. 1:florasan, 2:led
//uint16_t fire_trigger=220;
uint8_t lamba_onoff=0;
uint8_t lamba_onoffchanged=0;
uint8_t htim2_IRQ=0,htim21_IRQ=0,htim6_IRQ=0;
uint8_t syc1 =0,syc2=0;
uint8_t UartRXBuff[1];
uint8_t uartrecieved=0;
uint16_t firestorm;
uint8_t working=0;
/* USER CODE END PV */

/* Private function prototypes -----
-*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_SPI1_Init(void);
static void MX_ADC_Init(void);
static void MX_TIM6_Init(void);
static void MX_TIM2_Init(void);
static void MX_TIM21_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----
-*/
void TakeAndSend(void);
void nRF_SendPayload(void);
void nRF_PeriodicSend(void);
void nRF_SenderBuffAyarla(uint8_t alarmA);
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**

```

```

    * @brief The application entry point.
    *
    * @retval None
    */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_USART1_UART_Init();
    MX_SPI1_Init();
    MX_ADC_Init();
    MX_TIM6_Init();
    MX_TIM2_Init();
    MX_TIM21_Init();
    /* USER CODE BEGIN 2 */
    TIM2->CR1 |= TIM_CR1_ARP;
    HAL_TIM_Base_Start_IT(&htim6);
    HAL_GPIO_WritePin(LAMP_ONOFF_GPIO_Port,LAMP_ONOFF_Pin,GPIO_PIN_SET);
    HAL_UART_Receive_IT(&huart1, UartRXBuff,1);

    do {

        if (nRF24_Check())
        {}
        else
        {
            UART_SendStr("CONNECTING TO NRF24L01.\r\n");
        }
        HAL_Delay(500);
        //UART_SendHex32(HAL_SPI_GetError(&hspi1));
    }while (!nRF24_Check());

    UART_SendStr("OK.\r\n");
    // Initialize the nRF24L01 to its default state
    nRF24_Init();

```

```

nRF_RX_Init();
if (UART_INFO_ACTIVATED)
{
    UART_SendStr("Ready. Looking for IRQ.\r\n");
}

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(IRQHandled)
    {
        TakeAndSend();
        IRQHandled=0;
    }

    if(uartrecieved)
    {
        if (UartRXBuff[0]==97)
        {
            dimmervalue=dimmervalue+2;
            UART_SendInt(dimmervalue);
        }
        else if (UartRXBuff[0]==100)
        {
            dimmervalue=dimmervalue-2;
            UART_SendInt(dimmervalue);
        }
        uartrecieved=0;
        HAL_UART_Receive_IT(&huart1, UartRXBuff,1);
    }

    if (htim2_IRQ)
    {
        htim2_IRQ=0;
    }
    else if (htim21_IRQ)
    {
        htim21_IRQ=0;
    }
    else if (htim6_IRQ)
    {
        if(ledstatus==0)HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,GPIO_PIN_RESET);
        else
        if(ledstatus==1)HAL_GPIO_TogglePin(LED1_GPIO_Port,LED1_Pin);
        else
        if(ledstatus>1)HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,GPIO_PIN_SET);

        htim6_IRQ=0;
    }
}
/* USER CODE END WHILE */

```

```

/* USER CODE BEGIN 3 */

}
/* USER CODE END 3 */

}

/* USER CODE BEGIN 4 */

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Instance == TIM2)
    {
        //htim2_IRQ=1;

        if(lamptype==0)//halojen,dimmable
        {
            firestorm=
            3500+((255-dimmervalue)*15);
        }
        else
        {
            if(lamptype==1)//florasan
            {
                firestorm=
                7975+((255-dimmervalue)*4);
            }
            else if(lamptype==2)//led
            {
                firestorm=
                7975+((255-dimmervalue)*4);
            }

            __HAL_TIM_SET_AUTORELOAD(&htim2, firestorm);

            if(dimmervalue!=0)
            {
                HAL_GPIO_WritePin(LAMP_ONOFF_GPIO_Port,LAMP_ONOFF_Pin,GPIO_PIN_SET);
                HAL_GPIO_WritePin(TRIAC_CTRL_GPIO_Port,TRIAC_CTRL_Pin,GPIO_PIN_SET);

                HAL_TIM_Base_Start_IT(&htim2);
                HAL_TIM_Base_Stop(&htim2);
            }
            else
            {
                }
            }
        }
    }
    else if (htim->Instance == TIM21)
    {
        //htim21_IRQ=1;

```



```

    HAL_GPIO_WritePin(TRIAC_CTRL_GPIO_Port, TRIAC_CTRL_Pin, GPIO_PIN_RESET);

    HAL_GPIO_TogglePin(LED4_GPIO_Port, LED4_Pin);
                                HAL_TIM_Base_Stop(&htim21);
    }
    else if (htim->Instance == TIM6)
    {
        htim6_IRQ=1;
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_2)
    {
        if(ZCsource==0)HAL_TIM_Base_Start_IT(&htim2);

        {
            syc1=0;
            HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
        }
        syc1++;*/
    }
    else if(GPIO_Pin == GPIO_PIN_13)
    {
        //ZC_INPUT
        if(ZCsource)HAL_TIM_Base_Start_IT(&htim2);
        /*if ((syc2%50)==0)
        {
            syc2=0;
            HAL_GPIO_TogglePin(LED3_GPIO_Port, LED3_Pin);
        }
        syc2++;*/
    }
    else if(GPIO_Pin == GPIO_PIN_6)
    {
        IRQHandled=1;
    }
}

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    uartrecieved=1;
}

void nRF_SenderBuffAyarla(uint8_t alarmA)
{
    if(alarmA)
    {
        for(int i=0;i<32;i++)
        {
            nRF24_payload[i] = 0;
        }
        nRF24_payload[16] = 1; //DIMMER VALUE
        nRF24_payload[17] = dimmervalue; //DIMMER VALUE
        nRF24_payload[18] = 12;//alarm belirteç
        nRF24_payload[19] = alarmX;//alarm bilgisi
    }
}

```

```

        nRF24_payload[20] = 21;//kaynak bilgisi
        nRF24_payload[21] = ZCsource;
        nRF24_payload[22] = 22;//lambatipi
        nRF24_payload[23] = lamptype;
        nRF24_payload[24] = 3;//lm35 derece
        nRF24_payload[25] = lm35temp;
        nRF24_payload[26] = 2;//led durumu
        nRF24_payload[27] = ledstatus;
    }
    nRF24_payload[0] = serverid3;
    nRF24_payload[1] = serverid2;
    nRF24_payload[2] = serverid1;
    nRF24_payload[3] = serverid0;

    nRF24_payload[4] = 0;
    nRF24_payload[5] = 0;
    nRF24_payload[6] = 0;
    nRF24_payload[7] = DEVICE_ID;

    nRF24_payload[14] = MODULE_TYPE_ID;
}

void nRF_PeriodicSend(void)
{
    nRF_SenderBuffAyarla(1);
    PayloadSifrele();
    nRF_SendPayload();
    PayloadSifreCoz();
}

void TakeAndSend(void)
{
    if (nRF24_GetStatus_RXFIFO() != nRF24_STATUS_RXFIFO_EMPTY)
    {
        //if(UART_INFO_ACTIVATED){UART_SendStr("RX_IRQ.\r\n");}
        nRF24_RX_IRQCallBack();
        //KOMUTLAR FONKSIYONU RX.
        if(RX_Analysis())
        {
            PayloadSifrele();
            nRF_SendPayload();
            PayloadSifreCoz();
        }
        //UART_SendStr((char *)nRF24_payload);

    }
    else
    {
        //if(UART_INFO_ACTIVATED){UART_SendStr("Another
        IRQ_PIN_ASSERTED.\r\n");}
    }
}

void nRF_SendPayload(void)
{
    nRF_TX_Init();

```

```

        nRF_TX_Send();
        nRF24_FlushTX();
        nRF_RX_Init();
        /*if (UART_INFO_ACTIVATED)
        {
            UART_SendStr("Wait RX IRQ.\r\n");
        }*/
    }
    /* USER CODE END 4 */

```

Doğalgaz Kontrol Cihazı Kodları

“Main.c”

```

#include "main.h"
#include "stm32f0xx_hal.h"

/* USER CODE BEGIN Includes */
#include "nrf24.h"
#include "uart.h"
#include "protocol.h"
#include "aes.h"
/* USER CODE END Includes */

/* Private variables -----
-*/
ADC_HandleTypeDef hadc;

SPI_HandleTypeDef hspi1;

TIM_HandleTypeDef htim14;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
/* Private variables -----
-*/

// Buffer to store a payload of maximum width
uint8_t nRF24_payload[32];

// Length of received payload

```

```

uint8_t payload_length= 32;
uint8_t IRQHandled=0;

uint8_t roleacildi=0;
uint8_t hassasiyet=20;
uint8_t ledstatus=1;
uint8_t mq4analog=0;
uint8_t mq4digital=0;
uint8_t mq4aktif=1;
uint8_t gazrole=0;
uint8_t alarmX=0;
uint8_t buzzer=1;
uint8_t contunioussend=0;
uint8_t timertrigger=0;
uint8_t triggerdelayer=0;
uint8_t conversioncompleted=0;
uint16_t adccounter=0;
/* USER CODE END PV */

/* Private function prototypes -----
*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_TIM14_Init(void);
static void MX_ADC_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----
*/

void TakeAndSend(void);
void nRF_SendPayload(void);
void nRF_PeriodicSend(void);
void nRF_SenderBuffAyarla(uint8_t alarmA);
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 *
 * @retval None
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
    -----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    Systick. */
    HAL_Init();

```

```

/* USER CODE BEGIN Init */

/* USER CODE END Init */

/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_SPI1_Init();
MX_USART1_UART_Init();
MX_TIM14_Init();
MX_ADC_Init();
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim14);
HAL_ADC_Start_IT(&hadc);
// Configure the nRF24L01+
if (UART_INFO_ACTIVATED)
{
    //UART_SendStr("\r\nCommand Set created by Y.Emre FEDAR. YTU -
TEZLI Y.L. 16504021");
    //UART_SendStr("\r\nSystem started.STM32F030 - SICAKLIK - NEM -
MODULU");
    //UART_SendStr("\r\nnRF24L01 check: ");
}
do {

    if (nRF24_Check())
    {}
    else
    {
        UART_SendChar('E');
    }
    HAL_Delay(500);
    //UART_SendHex32(HAL_SPI_GetError(&hspi1));
}while (!nRF24_Check());

//UART_SendStr("OK.\r\n");
// Initialize the nRF24L01 to its default state
nRF24_Init();
/*****
nRF_RX_Init();
if (UART_INFO_ACTIVATED)
{
    UART_SendStr("Wait RX IRQ.\r\n");
}
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(IRQHandled)

```

```

    {
        TakeAndSend();
        IRQHandled=0;
    }
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    if(timertrigger)
    {
        if(alarmX)
        {
            if(triggerdelayer>2)//6sn de bir
            {
                nRF_PeriodicSend();
                triggerdelayer=0;
            }

            triggerdelayer++;
        }
        else
        {
            if(triggerdelayer>10)//20 sn de bir
            {
                nRF_PeriodicSend();
                triggerdelayer=0;
            }
            triggerdelayer++;
        }
        timertrigger=0;
    }

    if(conversioncompleted)
    {
        conversioncompleted=0;
        mq4analog=HAL_ADC_GetValue(&hadc);
        adccounter++;
        if(adccounter>15000)
        {
            UART_SendChar('#');
            UART_SendHex8(mq4analog);
            if (mq4analog>hassasiyet)
            {
                alarmX=1;
                if(roleacildi)
                {
                    HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,GPIO_PIN_RESET);

                    gazrole=0;
                }
                else
                {
                    HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,GPIO_PIN_SET);
                    roleacildi=1;
                    gazrole=1;
                }
            }
        }
    }
}

```

```

        else
        {
            alarmX=0;

            HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,GPIO_PIN_RESET
);
            roleacildi=0;
            gazrole=0;
        }
        mq4analog=0;
        adccounter=0;
    }

}

/* USER CODE END 3 */

}

/* USER CODE BEGIN 4 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    conversioncompleted=1;
}
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance==TIM14)
    {
        if(ledstatus==1) HAL_GPIO_TogglePin(LED1_GPIO_Port,LED1_Pin);
        if(buzzer)
        {
            if(alarmX)
            HAL_GPIO_TogglePin(BUZZER_GPIO_Port,BUZZER_Pin);
            else
            HAL_GPIO_WritePin(BUZZER_GPIO_Port,BUZZER_Pin,GPIO_PIN_RESET);
        }
        else
        {
            HAL_GPIO_WritePin(BUZZER_GPIO_Port,BUZZER_Pin,GPIO_PIN_RESET);
        }
        timertrigger=1;
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    IRQHandled=1;
}

void nRF_SenderBuffAyarla(uint8_t alarmA)
{
    for(int i=0;i<32;i++)
    {
        nRF24_payload[i] = 0;
    }
    nRF24_payload[16] = 5;//GAZ ROLESI
    nRF24_payload[17] = gazrole;//GAZ ROLESI
}

```

```

nRF24_payload[18] = 12;//alarm belirteç
nRF24_payload[19] = alarmX;//alarm bilgisi
nRF24_payload[20] = 15;//MQ4 analog gaz verisi
nRF24_payload[21] = mq4analog;
nRF24_payload[22] = 18;//MQ4 acikmi?
nRF24_payload[23] = mq4aktif;

nRF24_payload[0] = serverid3;
nRF24_payload[1] = serverid2;
nRF24_payload[2] = serverid1;
nRF24_payload[3] = serverid0;

nRF24_payload[4] = 0;
nRF24_payload[5] = 0;
nRF24_payload[6] = 0;
nRF24_payload[7] = DEVICE_ID;

nRF24_payload[14] = MODULE_TYPE_ID;
}

void nRF_PeriodicSend(void)
{
    nRF_SenderBuffAyarla(alarmX);
    PayloadSifrele();
    nRF_SendPayload();
    PayloadSifreCoz();
}

void TakeAndSend(void)
{
    if (nRF24_GetStatus_RXFIFO() != nRF24_STATUS_RXFIFO_EMPTY)
    {
        if(UART_INFO_ACTIVATED){UART_SendStr("RX_IRQ.\r\n");}
        nRF24_RX_IRQCallBack();
        //KOMUTLAR FONKSIYONU GELECEK.
        if(RX_Analysis())
        {
            PayloadSifrele();
            nRF_SendPayload();
            PayloadSifreCoz();
        }
    }
    else
    {
        //if(UART_INFO_ACTIVATED){UART_SendStr("Another
        IRQ_PIN_ASSERTED.\r\n");}
    }
}

void nRF_SendPayload(void)
{
    nRF_TX_Init();

    nRF_TX_Send();
    nRF24_FlushTX();
    nRF_RX_Init();
    if (UART_INFO_ACTIVATED)
    {
        UART_SendStr("Wait RX IRQ.\r\n");
    }
}

```



```

    }
}
/* USER CODE END 4 */

```

Kapı Kontrol Cihazı Kodları

“Main.c”

```

#include "main.h"
#include "stm32f0xx_hal.h"

/* USER CODE BEGIN Includes */
#include "nrf24.h"
#include "uart.h"
#include "protocol.h"
/* USER CODE END Includes */

/* Private variables -----
-*/
SPI_HandleTypeDef hspi1;

TIM_HandleTypeDef htim14;
TIM_HandleTypeDef htim16;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
/* Private variables -----
-*/
uint8_t nRF24_payload[32];

// Length of received payload
uint8_t payload_length= 32;
uint8_t IRQHandled=0;

uint8_t ledstatus=1;
uint8_t alarmX=0;
uint8_t buzzer=1;
uint8_t contunioussend=0;
uint8_t timertrigger=0;
uint8_t triggerdelayer=0;

uint8_t kapiaciksayici=0;
uint8_t kapidurumu=0;
uint8_t kapialarmcarpani=60;
uint8_t kapiuyaricarpani=30;
uint8_t kapiacildi=0;

```

```

/* USER CODE END PV */

/* Private function prototypes -----
-*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_TIM14_Init(void);
static void MX_TIM16_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----
-*/
void TakeAndSend(void);
void nRF_SendPayload(void);
void nRF_PeriodicSend(void);
void nRF_SenderBuffAyarla(uint8_t alarmA);
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 *
 * @retval None
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
    -----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_SPI1_Init();
    MX_USART1_UART_Init();
    MX_TIM14_Init();
    MX_TIM16_Init();

```

```

/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim14);
HAL_TIM_Base_Start_IT(&htim16);
// Configure the nRF24L01+
if (UART_INFO_ACTIVATED)
{
    //UART_SendStr("\r\nCommand Set created by Y.Emre FEDAR. YTU -
TEZLI Y.L. 16504021");
    //UART_SendStr("\r\nSystem started.STM32F030 - SICAKLIK - NEM -
MODULU");
    //UART_SendStr("\r\nnRF24L01 check: ");
}
do {

    if (nRF24_Check())
    {}
    else
    {
        UART_SendChar('E');
    }
    HAL_Delay(500);
    //UART_SendHex32(HAL_SPI_GetError(&hspi1));
}while (!nRF24_Check());

//UART_SendStr("OK.\r\n");
// Initialize the nRF24L01 to its default state
nRF24_Init();
/*****
*****/
nRF_RX_Init();
if (UART_INFO_ACTIVATED)
{
    UART_SendStr("Wait RX IRQ.\r\n");
}
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(IRQHandled)
    {
        TakeAndSend();
        IRQHandled=0;
    }
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    if(timertrigger)
    {
        if(alarmX)
        {
            triggerdelay=11;
            if(triggerdelay>10)
            {
                nRF_PeriodicSend();
                triggerdelay=0;
            }
        }
    }
}

```

```

        triggerdelayer++;
    }

    timertrigger=0;
}
if(kapiaciksayici>1 && (!kapiacildi))
{
    nRF_PeriodicSend();
    kapiacildi=1;
}
if(kapiacildi && kapiaciksayici<1)
{
    nRF_PeriodicSend();
    kapiacildi=0;
}
}
/* USER CODE END 3 */

}

/* USER CODE BEGIN 4 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance==TIM14)
    {
        timertrigger=1;
    }
    else if(htim->Instance==TIM16)
    {
        kapidurumu =
        HAL_GPIO_ReadPin(KAPI_DURUM_GPIO_Port,KAPI_DURUM_Pin);
        if(kapidurumu==1)
        {
            kapiaciksayici++;

            if(ledstatus>0)HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,GPIO_PIN_SET)
;
            }
            else
            {
                alarmX=0;
                kapiaciksayici=0;

                HAL_GPIO_WritePin(BUZZER_GPIO_Port,BUZZER_Pin,GPIO_PIN_RESET);

                HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,GPIO_PIN_RESET);
            }
            if(kapiaciksayici>(kapialarmcarpani*2))//60 saniye
            {
                if(buzzer)HAL_GPIO_WritePin(BUZZER_GPIO_Port,BUZZER_Pin,GPIO_PIN_SET);
                alarmX=1;
                kapiaciksayici=(kapialarmcarpani*2)+1;
            }

            else if(kapiaciksayici>(kapiuyaricarpani*2))//30 saniye

```

```

        {

            if(buzzer)HAL_GPIO_TogglePin(BUZZER_GPIO_Port,BUZZER_Pin);
        }

    }
}
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    IRQHandled=1;
}
void nRF_SenderBuffAyarla(uint8_t alarmA)
{
    for(int i=0;i<32;i++)
    {
        nRF24_payload[i] = 0;
    }
    nRF24_payload[16] = 6;//KAPI DURUMU
    nRF24_payload[17] = kapidurumu;//KAPI DURUMU
    nRF24_payload[18] = 12;//alarm belirteç
    nRF24_payload[19] = alarmX;//alarm bilgisi
    nRF24_payload[20] = 2;
    nRF24_payload[21] = ledstatus;
    nRF24_payload[22] = 7;
    nRF24_payload[23] = buzzer;
    nRF24_payload[24] = 20; //KAPI UYARI ÇARPANI
    nRF24_payload[25] = kapiuyaricarpani;
    nRF24_payload[26] = 19; //KAPI ALARM ÇARPANI
    nRF24_payload[27] = kapialarmcarpani;

    nRF24_payload[0] = serverid3;
    nRF24_payload[1] = serverid2;
    nRF24_payload[2] = serverid1;
    nRF24_payload[3] = serverid0;

    nRF24_payload[4] = 0;
    nRF24_payload[5] = 0;
    nRF24_payload[6] = 0;
    nRF24_payload[7] = DEVICE_ID;

    nRF24_payload[14] = MODULE_TYPE_ID;
}

void nRF_PeriodicSend(void)
{
    nRF_SenderBuffAyarla(alarmX);
    PayloadSifrele();
    nRF_SendPayload();
    PayloadSifreCoz();
}

void TakeAndSend(void)
{
    if (nRF24_GetStatus_RXFIFO() != nRF24_STATUS_RXFIFO_EMPTY)
    {
        if(UART_INFO_ACTIVATED){UART_SendStr("RX_IRQ.\r\n");}
        nRF24_RX_IRQCallBack();
        //KOMUTLAR FONKSIYONU GELECEK.
    }
}

```

```

        if(RX_Analysis())
        {
            PayloadSifrele();
            nRF_SendPayload();
            PayloadSifreCoz();
        }
        //UART_SendStr((char *)nRF24_payload);

    }
    else
    {
        //if(UART_INFO_ACTIVATED){UART_SendStr("Another
IRQ_PIN_ASSERTED.\r\n");}
    }
}

void nRF_SendPayload(void)
{
    nRF_TX_Init();

    nRF_TX_Send();
    nRF24_FlushTX();
    nRF_RX_Init();
    if (UART_INFO_ACTIVATED)
    {
        UART_SendStr("Wait RX IRQ.\r\n");
    }
}
/* USER CODE END 4 */

```

Sıcaklık-Nem Cihazı Kodları

“Main.c”

```
/* Includes -----
-*/
#include "main.h"
#include "stm32f0xx_hal.h"

/* USER CODE BEGIN Includes */
#include "nrf24.h"
#include "uart.h"
#include "aes.h"
#include "protocol.h"
/* USER CODE END Includes */

/* Private variables -----
-*/
I2C_HandleTypeDef hi2c1;

SPI_HandleTypeDef hspi1;

TIM_HandleTypeDef htim14;
TIM_HandleTypeDef htim16;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
/* Private variables -----
-*/

// Buffer to store a payload of maximum width
uint8_t nRF24_payload[32];

// Length of received payload
uint8_t payload_length= 32;
uint8_t IRQHandled=0;
//am2320 variables

//#define AM2320_CMD_READREG      0x03 ///< read register command
//#define AM2320_REG_TEMP_H      0x02 ///< temp register address
//#define AM2320_REG_HUM_H      0x00 ///< humidity register address
uint8_t bufTxRx[8];
uint8_t sensoroku=0;
uint16_t rawT, rawH;
//float Temperature; float Humidity;
uint8_t alarmX=0;

uint8_t buzzer=1;
uint8_t ledstatus=1;
uint8_t sicaknemmodulaktif=1;
uint8_t contunioussend=1;
uint8_t contunioussendtrigger=0;
/* USER CODE END PV */

/* Private function prototypes -----
-*/
void SystemClock_Config(void);
```

```

static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
static void MX_SPI1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_TIM14_Init(void);
static void MX_TIM16_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----
-*/
void SicaklikNemUartGonder(void);
int ReadSens(void);
void nRF_SenderBuffAyarla(void);
void nRF_PeriodicSend(void);
void TakeAndSend(void);
void nRF_SendPayload(void);
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 *
 * @retval None
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----
    ---*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_I2C1_Init();
    MX_SPI1_Init();
    MX_USART1_UART_Init();
    MX_TIM14_Init();
    MX_TIM16_Init();
    /* USER CODE BEGIN 2 */

```



```

HAL_TIM_Base_Start_IT(&htim14);
HAL_TIM_Base_Start_IT(&htim16);
// Configure the nRF24L01+
if (UART_INFO_ACTIVATED)
{
    //UART_SendStr("\r\nCommand Set created by Y.Emre FEDAR. YTU - TEZLI
Y.L. 16504021");
    //UART_SendStr("\r\nSystem started.STM32F030 - SICAKLIK - NEM -
MODULU");
    //UART_SendStr("\r\nnRF24L01 check: ");
}
do {

    if (nRF24_Check())
    {}
    else
    {
        UART_SendChar('E');
    }
    HAL_Delay(500);
    //UART_SendHex32(HAL_SPI_GetError(&hspi1));
}while (!nRF24_Check());

//UART_SendStr("OK.\r\n");
// Initialize the nRF24L01 to its default state
nRF24_Init();

/*****
nRF_RX_Init();
if (UART_INFO_ACTIVATED)
{
    UART_SendStr("RX waiting.\r\n");
}
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(IRQHandled)
    {
        TakeAndSend();
        IRQHandled=0;
    }
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
    if(contunioussendtrigger)
    {
        if(contunioussend)
        {
            nRF_PeriodicSend();
        }
        else
        {
            if(alarmX || (!sicaknemmodulaktif))
            {
                nRF_PeriodicSend();
            }
        }
    }
}

```

```

        }
        contunioussendtrigger=0;
    }

    if(sensoroku)
    {
        ReadSens();
        SicaklikNemUartGonder();
        sensoroku=0;
    }
}
/* USER CODE END 3 */

}

/* USER CODE BEGIN 4 */

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance==TIM14)
    {
        sensoroku=1;
        if(ledstatus==1) HAL_GPIO_TogglePin(LED1_GPIO_Port,LED1_Pin);
        if(buzzer)
        {
            if(alarmX)
                HAL_GPIO_TogglePin(BUZZER_GPIO_Port,BUZZER_Pin);
            else
                HAL_GPIO_WritePin(BUZZER_GPIO_Port,BUZZER_Pin,GPIO_PIN_RESET);
        }
        else
        {
            HAL_GPIO_WritePin(BUZZER_GPIO_Port,BUZZER_Pin,GPIO_PIN_RESET);
        }

    }
    if(htim->Instance==TIM16)
    {
        contunioussendtrigger=1;
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    IRQHandled=1;
}

int ReadSens(void)
{
    // step 1:wakeup
    if(HAL_I2C_IsDeviceReady(&hi2c1,0x5C<<1, 50 , 100)!=HAL_OK)
    {
        UART_SendStr("\r\nNS I2C.");
        sicaknemmodulaktif=0;
        //nRF_PeriodicSend();
        return 0;
    }
    else
    {
        sicaknemmodulaktif=1;
    }
}

```

```

    }

    //
    HAL_I2C_Master_Transmit(&hi2c1,0x5C<<1,NULL,0,0xFFFF);
    HAL_Delay(3); // 800us - 3ms

    // step 2:command
    bufTxRx[0] = 0x03; // Get Humidity and Temperature
    bufTxRx[1] = 0x00; // Start address
    bufTxRx[2] = 0x04; // Length
    HAL_I2C_Master_Transmit(&hi2c1,0x5C<<1,bufTxRx,3,0xFFFF);
    HAL_Delay(2); // 1.5ms

    // step 3:data
    HAL_I2C_Master_Receive(&hi2c1,0x5C<<1,bufTxRx,8,0xFFFFFFFF);
    // UART_SendStr("\r\nGelen:");
    /*for(int i=0;i<8;i++)
    {
        UART_SendHex8(bufTxRx[i]);
        UART_SendStr("-");
    }*/
    rawH = ((uint16_t)(bufTxRx[2]<<8))|bufTxRx[3];
    //Humidity = rawH/10;
    rawT = ((uint16_t)(bufTxRx[4]<<8))|bufTxRx[5];
    //Temperature = rawT/10;
    if(rawT>400 || rawT <50 || rawH <150 || rawH > 600)
    {
        /*if(alarmX==0)
        {
            HAL_TIM_Base_Stop(&htim14);
            htim14.Init.Period = 750;
            HAL_TIM_Base_Init(&htim14);
            HAL_TIM_Base_Start_IT(&htim14);

        }*/
        alarmX=1;
    }
    else
    {
        /*if(alarmX)
        {
            HAL_TIM_Base_Stop(&htim14);
            htim14.Init.Period = 5000;
            HAL_TIM_Base_Init(&htim14);
            HAL_TIM_Base_Start_IT(&htim14);

        }*/
        alarmX=0;
    }
    return 1;
}

void SicaklikNemUartGonder(void)
{
    /*UART_SendStr("\r\noC:");
    UART_SendInt((rawT/10));
    UART_SendChar('.');
    UART_SendInt(rawT%10);
    UART_SendStr(" -- H:");
    UART_SendInt(rawH/10);
    UART_SendChar('.');
    UART_SendInt(rawH%10);*/
}

```

```

    UART_SendChar('#');
    UART_SendInt((rawT/10));
    UART_SendChar('-');
    UART_SendInt(rawH/10);
    UART_SendChar('#');
}

void nRF_SenderBuffAyarla(void)
{
    for(int i=0;i<32;i++)
    {
        nRF24_payload[i] = 0;
    }
    nRF24_payload[0] = serverid3;
    nRF24_payload[1] = serverid2;
    nRF24_payload[2] = serverid1;
    nRF24_payload[3] = serverid0;

    nRF24_payload[4] = 0;
    nRF24_payload[5] = 0;
    nRF24_payload[6] = 0;
    nRF24_payload[7] = DEVICE_ID;

    nRF24_payload[14] = MODULE_TYPE_ID;

    nRF24_payload[16] = 8;//SICAKLIK VERILERININ OKUNMASINI SAGLAYACAK
    nRF24_payload[18] = 12;//alarm belirteç
    nRF24_payload[19] = alarmX;//alarm bilgisi
    nRF24_payload[20] = 16;//sicakliknem sensörü aktifmi belirteç
    nRF24_payload[21] = sicaknemmodulaktif;
    nRF24_payload[22] = 17;//devamlı gönderilsin mi?
    nRF24_payload[23] = contunioussend;

    if(sicaknemmodulaktif)
    {
        nRF24_payload[28]= (uint8_t)(rawT>>8);           //TEMP HIGH
        nRF24_payload[29]= (uint8_t)(rawT&0x00FF);        //TEMP LOW
        nRF24_payload[30]= (uint8_t)(rawH>>8);           //HUMIDITY HIGH
        nRF24_payload[31]= (uint8_t)(rawH&0x00FF);        //HUMIDITY LOW
    }
    else
    {
        nRF24_payload[28]= 0;           //TEMP HIGH
        nRF24_payload[29]= 0;           //TEMP LOW
        nRF24_payload[30]= 0;           //HUMIDITY HIGH
        nRF24_payload[31]= 0;           //HUMIDITY LOW
    }
}

void nRF_PeriodicSend(void)
{
    nRF_SenderBuffAyarla();
    PayloadSifrele();
    nRF_SendPayload();
    PayloadSifreCoz();
}

```

```

void TakeAndSend(void)
{
    if (nRF24_GetStatus_RXFIFO() != nRF24_STATUS_RXFIFO_EMPTY)
    {

if(UART_INFO_ACTIVATED){UART_SendStr("RX_IRQ.\r\n");}
        nRF24_RX_IRQCallBack();
        //KOMUTLAR FONKSIYONU GELECEK.
        if(RX_Analysis())
        {
            //HAL_Delay(500);
            PayloadSifrele();
            nRF_SendPayload();
            PayloadSifreCoz();
        }
        //UART_SendStr((char *)nRF24_payload);

    }
    else
    {
        //if(UART_INFO_ACTIVATED){UART_SendStr("Another
        IRQ_PIN_ASSERTED.\r\n");}
    }
}

void nRF_SendPayload(void)
{
    nRF_TX_Init();

    nRF_TX_Send();

    nRF_RX_Init();
    if (UART_INFO_ACTIVATED)
    {
        UART_SendStr("Wait RX_IRQ.\r\n");
    }
}

/* USER CODE END 4 */

```

Ütü, Klima, Kombi Kontrol Cihazları Kodları

“Main.c”

```

/* Includes -----
-*/
#include "main.h"
#include "stm32f0xx_hal.h"
/* USER CODE BEGIN Includes */
#include "nrf24.h"
#include "uart.h"
#include "protocol.h"
/* USER CODE END Includes */

/* Private variables -----
-*/

```

```

ADC_HandleTypeDef hadc;

SPI_HandleTypeDef hspi1;

TIM_HandleTypeDef htim14;
TIM_HandleTypeDef htim16;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
/* Private variables -----
-*/

// Buffer to store a payload of maximum width
uint8_t nRF24_payload[32];

// Length of received payload
uint8_t payload_length= 32;
uint8_t IRQHandled=0;

uint8_t timer14irq=0;
uint8_t triggerdelayer=0;
uint8_t alarmX=0;
uint8_t conversioncompleted=0;
uint16_t adccounter=0;
uint8_t lm35temp=0;
uint8_t ledstatus=1;

uint8_t carpmasayisi=0;
uint8_t timer16irq=0;
uint8_t timeraktif=0;
uint8_t highwattrelay=0;
uint32_t acmakesmesuresi=10;//saniye cinsinden
uint32_t acmakesmesayaci=0;//saniye
uint8_t acmakesmestartstop=0;//0:herseyidurdur//1:acma sayacı başlat//2:kesme
sayacı başlat
uint8_t yenedurumgeldi=0;
/* USER CODE END PV */

/* Private function prototypes -----
-*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_ADC_Init(void);
static void MX_TIM14_Init(void);
static void MX_TIM16_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----
-*/
void TakeAndSend(void);
void nRF_SendPayload(void);
void nRF_PeriodicSend(void);
void nRF_SenderBuffAyarla(uint8_t alarmA);
/* USER CODE END PFP */

```

```

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 *
 * @retval None
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_SPI1_Init();
    MX_USART1_UART_Init();
    MX_ADC_Init();
    MX_TIM14_Init();
    MX_TIM16_Init();
    /* USER CODE BEGIN 2 */
    HAL_TIM_Base_Start_IT(&htim14);
    //HAL_ADC_Start_IT(&hadc);
    // Configure the nRF24L01+
    if (UART_INFO_ACTIVATED)
    {
        //UART_SendStr("\r\nCommand Set created by Y.Emre FEDAR. YTU -
        TEZLI Y.L. 16504021");
        //UART_SendStr("\r\nSystem started.STM32F030 - SICAKLIK - NEM -
        MODULU");
        //UART_SendStr("\r\nnRF24L01 check: ");
    }
    do {

        if (nRF24_Check())
        {}
        else
        {
            UART_SendChar('E');

```

```

    }
    HAL_Delay(500);
    //UART_SendHex32(HAL_SPI_GetError(&hspi1));
}while (!nRF24_Check());

//UART_SendStr("OK.\r\n");
// Initialize the nRF24L01 to its default state
nRF24_Init();
/*****
nRF_RX_Init();
if (UART_INFO_ACTIVATED)
{
    UART_SendStr("Ready.IRQ waiting.\r\n");}
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(IRQHandled)
    {
        TakeAndSend();
        IRQHandled=0;
    }
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    if(yenidurumgeldi){//Yeni data gelince sayacı baştan başlat
        acmakesmesayaci=0;
        yenidurumgeldi=0;
        acmakesmesuresi=10;
        HAL_TIM_Base_Stop(&htim16);

        for(int i=0;i<carpmasayisi;i++)//Androide göre yazıldı
        {

            if(acmakesmesuresi>86399)acmakesmesuresi=acmakesmesuresi+86400;//1 gün
ekle
                else acmakesmesuresi=acmakesmesuresi*2;//1 güne
gelmemişse saniyeyi 2 yle çarp
        }

        if(acmakesmestartstop>0)HAL_TIM_Base_Start_IT(&htim16);
        else if(acmakesmestartstop==0)
HAL_TIM_Base_Stop(&htim16);
        if(acmakesmestartstop==1)
        {
            highwattrelay=0;

            HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,
GPIO_PIN_RESET);

            nRF_PeriodicSend();
        }
        else if(acmakesmestartstop==2)
        {
            highwattrelay=1;

```



```

        HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,
GPIO_PIN_SET);
        nRF_PeriodicSend();
    }
}

if(timer14irq)
{
    if(triggerdelayer>30)//0 olunca if hep 1
    {
        nRF_PeriodicSend();
        triggerdelayer=0;
    }
    triggerdelayer++;

    timer14irq=0;
    /*if(highwattrelay)
    {
        HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,
GPIO_PIN_SET);
    }
    else
    {
        HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,
GPIO_PIN_RESET);
    }*/
}
if(timer16irq)//timer16 sayacı hedefe ulaşmışsa
{
    timer16irq=0;
    HAL_TIM_Base_Stop(&htim16);
    if(acmakesmestartstop==1)
    {
        highwattrelay=1;

        HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,GPIO_PIN_SET);
        nRF_PeriodicSend();
    }
    else if(acmakesmestartstop==2)
    {
        highwattrelay=0;

        HAL_GPIO_WritePin(RELAY_ONOFF_GPIO_Port,RELAY_ONOFF_Pin,GPIO_PIN_RESET
);
        nRF_PeriodicSend();
    }
    acmakesmesayaci=0;
    acmakesmestartstop=0;
}

if(conversioncompleted)
{
    conversioncompleted=0;
}

```

```

lm35temp=HAL_ADC_GetValue(&hadc); //(lm35temp/255)*3.3
~=237mV =237mV/10mV=23.7C
adccounter++;
if(adccounter>10000)
{
    //UART_SendChar('#');
    UART_SendHex8(lm35temp);
    if (lm35temp>50)
    {
        alarmX=1;
    }
    else
    {
        alarmX=0;
    }
    lm35temp=0;
    adccounter=0;
}

}

}
/* USER CODE END 3 */

}

/* USER CODE BEGIN 4 */
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    conversioncompleted=1;
}
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Instance==TIM14)
    {
        if(ledstatus==0)
        HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,GPIO_PIN_RESET);
        else if(ledstatus==1)
        HAL_GPIO_TogglePin(LED1_GPIO_Port,LED1_Pin);
        else if(ledstatus>1)
        HAL_GPIO_WritePin(LED1_GPIO_Port,LED1_Pin,GPIO_PIN_SET);
        timer14irq=1;
    }
    if(htim->Instance==TIM16)
    {
        if(acmakesmesayaci==acmakesmesuresi)
        {
            timer16irq=1;
        }
        else acmakesmesayaci++;
    }
}
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    IRQHandled=1;
}

void nRF_SenderBuffAyarla(uint8_t alarmA)
{

```

```

for(int i=0;i<32;i++)
{
    nRF24_payload[i] = 0;
}
nRF24_payload[16] = 9;//modul tip2 role
nRF24_payload[17] = highwattrelay;//modul tip2 role
nRF24_payload[18] = 12;//alarm belirteç
nRF24_payload[19] = alarmX;//alarm bilgisi
nRF24_payload[20] = 3;//lm35 verisi
nRF24_payload[21] = lm35temp;//lm35 verisi
nRF24_payload[22] = 10;//kesmestartstop
nRF24_payload[23] = acmakesmestartstop;//acmakesmestartstop
if(acmakesmestartstop>0)
{
    nRF24_payload[28] = (uint8_t)((acmakesmesayaci&0xFF000000)>>24);
    nRF24_payload[29] = (uint8_t)((acmakesmesayaci&0x00FF0000)>>16);
    nRF24_payload[30] = (uint8_t)((acmakesmesayaci&0x0000FF00)>>8);
    nRF24_payload[31] = (uint8_t)( acmakesmesayaci&0x000000FF);
}
else
{
    nRF24_payload[28] = (uint8_t)((acmakesmesuresi&0xFF000000)>>24);
    nRF24_payload[29] = (uint8_t)((acmakesmesuresi&0x00FF0000)>>16);
    nRF24_payload[30] = (uint8_t)((acmakesmesuresi&0x0000FF00)>>8);
    nRF24_payload[31] = (uint8_t)( acmakesmesuresi&0x000000FF);
}
nRF24_payload[0] = serverid3;
nRF24_payload[1] = serverid2;
nRF24_payload[2] = serverid1;
nRF24_payload[3] = serverid0;

nRF24_payload[4] = 0;
nRF24_payload[5] = 0;
nRF24_payload[6] = 0;
nRF24_payload[7] = DEVICE_ID;

nRF24_payload[14] = MODULE_TYPE_ID;
}

void nRF_PeriodicSend(void)
{
    nRF_SenderBuffAyarla(alarmX);
    PayloadSifrele();
    nRF_SendPayload();
    PayloadSifreCoz();
}

void TakeAndSend(void)
{
    if (nRF24_GetStatus_RXFIFO() != nRF24_STATUS_RXFIFO_EMPTY)
    {
        //if(UART_INFO_ACTIVATED){UART_SendStr("RX_IRQ.\r\n");}
        nRF24_RX_IRQCallBack();
        //KOMUTLAR FONKSIYONU GELECEK.
        if(RX_Analysis())
        {
            PayloadSifrele();
            nRF_SendPayload();
        }
    }
}

```

```

        PayloadSifreCoz();
    }
    //UART_SendStr((char *)nRF24_payload);

}
else
{
    //if(UART_INFO_ACTIVATED){UART_SendStr("Another
IRQ_PIN_ASSERTED.\r\n");}
}
}

void nRF_SendPayload(void)
{
    nRF_TX_Init();

    nRF_TX_Send();
    nRF24_FlushTX();
    nRF24_FlushRX();
    nRF_RX_Init();
    if (UART_INFO_ACTIVATED)
    {
        UART_SendStr("Wait RX IRQ.\r\n");
    }
}

/* USER CODE END 4 */

```

Ek-2

Android Yazılımının Kodları

QML’de yazılan Kontrol Kutusu kodu

“KontrolKutusu.qml”

```
import QtQuick 2.0
import QtQuick.Controls 2.3
import com.fedartech.qmlsockets 1.0
import com.fedartech.qmlprotocol 1.0

Item {
    id: root

    width: 264
    height: 264
    property alias alarmmousearea: alarmmousearea
    property alias lm35tempmousearea: lm35tempmousearea
    property alias txt_durumaltsag: txt_durumaltsag
    property alias txt_durumaltsol: txt_durumaltsol
    property alias ledmousearea: ledmousearea
    property alias alarmimage: alarmimage
    property alias zcSourceMouseArea: zcSourceMouseArea
    property alias lamptypeMouseArea: lamptypeMouseArea
    property alias buzzerMouseArea: buzzerMouseArea
    property alias buzzerImage: buzzerImage
    property alias lm35temp: lm35temp
    property alias ledimage: ledimage
    property alias guncellenmeTarih: guncellenmeTarih
    property alias mouseArea: mouseArea
    property alias txt_durumAlt: txt_durumAlt
    property alias txt_durumAna: txt_durumAna
    property alias image: image
    property alias slider: slider
    property alias txt_baslik: txt_baslik

    // Sonradan esitlenen degiskenler
    property FDRTCPSocket socket
    property int modulId: 0
    property int modulType: 0
    property double sicaklik: 27.3
    property double nem: 54.1
    property bool butonaktif:false
    property int lambatipi: 1
    property int zckaynak: 0
    property int ledstatus: 1
    property bool newsliderdata:false
    property bool slidersifirdan255:false
    property bool buzzerstatus: true
    property int kapiuyaricarpani: 20
    property int kapialarmcarpani: 40
    property bool sicaklikcontunious: true
    property bool acmakesmeyenibilgi: false
    property int acmakesmestartstop: 0 //0:dur,1:açma sayacı,2:kesme
    sayacı
```

```

property int acmakesmesuresi: 10
property int acmakesmesayaci: 0
property int dakika: 0
property int saat: 0
property int gun: 0
property int temporary: 0
property int temporary2: 0
property int carpmasayisi: 0

Rectangle {
    id: rectangle
    x: 5
    y: 5
    color: "#1d1d1d"
    radius: 10
    anchors.rightMargin: 5
    anchors.leftMargin: 5
    anchors.bottomMargin: 5
    anchors.topMargin: 5

    border.color: "#050505"
    border.width: 1
    anchors.fill: parent

    MouseArea {
        id: mouseArea
        x: 11
        width: 254
        height: 156
        anchors.top: parent.top
        anchors.topMargin: 5
        anchors.verticalCenterOffset: -33
        anchors.horizontalCenterOffset: 0
        anchors.horizontalCenter: parent.horizontalCenter
        anchors.verticalCenter: parent.verticalCenter

        onClicked: {
            if(modulType===4 || modulType==3){}
            else
            {
                if(butonaktif===false){
                    // rectangle.color =
                    "#004356"
                    //
                    rectangle.border.color = "#0076b2"
                    // image.source =
                    "img/on_icon.png"
                    // image.update()
                    //slider.enabled=true
                    butonaktif=true
                    if((slider.value===0) && slider.visible){
                        slidersifirdan255=true
                        slider.value=255}
                }else{
                    // rectangle.color =
                    "#1d1d1d"
                    //
                    rectangle.border.color = "#050505"
                    // image.source =
                    "img/off_icon.png"
                    // image.update()

```

```

        //slider.enabled=false
        butonaktif=false

    }
    sendCommand(1)

}

}

}

Slider {
    id: slider
    width: 150
    to: 255
    stepSize: 10
    enabled: false
    anchors.horizontalCenter: parent.horizontalCenter
    anchors.bottom: parent.bottom
    anchors.bottomMargin: 9
    value: 0

    Component.onCompleted:
    {
        if(slider.visible) slidertimer.start()
    }

    onValueChanged:
    {
        if(slider.value===0)
        {
            // rectangle.color = "#1d1d1d"
            // rectangle.border.color =
"#050505"
            // image.source =
"img/off_icon.png"
            // image.update()
            //slider.enabled=false
            //butonaktif=false
        }else
        {
            // rectangle.color = "#004356"
            // rectangle.border.color =
"#0076b2"
            // image.source =
"img/on_icon.png"
            // image.update()
            slider.enabled=true
            butonaktif=true
        }
        newsliderdata=1
    }

}

Text {
    id: txt_baslik
    width: 200
    height: 30
    color: "#f6f6f6"
    text: "-- İsim --"

```



```

fontSizeMode: Text.HorizontalFit

Component.onCompleted:
{
    if(modulType===4 || modulType===3){
        txt_durumAna.visible=true
    }
    else{
        txt_durumAna.visible=false
    }
}

Text {
    id: txt_durumAlt
    x: -9
    y: 1
    width: 200
    height: 30
    color: "#f6f6f6"
    text: "% -"
    anchors.horizontalCenterOffset: 0
    anchors.horizontalCenter: parent.horizontalCenter
    anchors.top: txt_durumAna.bottom
    horizontalAlignment: Text.AlignHCenter
    font.pointSize: 15
    verticalAlignment: Text.AlignVCenter
    fontSizeMode: Text.HorizontalFit
    font.bold: false

Component.onCompleted:
{
    if(modulType===1 || modulType===2){
        txt_durumAlt.visible=false
    }
    else{
        txt_durumAlt.visible=true
        txt_durumAlt.text = "-"
    }
}

Text {
    id: guncellenmeTarih
    x: 27
    width: 200
    height: 30
    color: "#ffffff"
    text: qsTr("---")
    anchors.horizontalCenter: parent.horizontalCenter
    anchors.top: parent.top
    anchors.topMargin: 178
    font.bold: true
    fontSizeMode: Text.HorizontalFit
    wrapMode: Text.NoWrap
    verticalAlignment: Text.AlignVCenter
    horizontalAlignment: Text.AlignHCenter
    font.pixelSize: 10
    Component.onCompleted: {

```

```

        if (modulType===2)
        {
            guncellenmeTarih.visible=true
        }
        else
        {
            guncellenmeTarih.visible=false
        }
    }

    AnimatedImage {
        id: ledimage
        width: 48
        height: 48
        anchors.left: parent.left
        anchors.leftMargin: 0
        anchors.bottom: parent.bottom
        anchors.bottomMargin: 0
        source: "img/led-anim.webp"

        MouseArea {
            id: ledmousearea
            anchors.fill: parent
            onClicked:
            {
                ledstatus = (ledstatus+1) % 3;
                sendCommand(1)
                //
                if(ledimage.source==="img/led-on.png")
                {
                    //
                    ledimage.source==="img/led-anim.webp"
                    //
                }
                else
                if(ledimage.source==="img/led-anim.webp")
                {
                    //
                    ledimage.source==="img/led-off.png"
                    //
                }
                else
                {
                    //
                    ledimage.source==="img/led-on.png"
                    //
                }
            }
        }
    }

    Text {
        id: lm35temp
        width: 50
        height: 25
        color: "#b9ffa2"
        text: qsTr("")
        anchors.right: parent.right
        anchors.rightMargin: 0
        anchors.bottom: parent.bottom
    }

```

```

anchors.bottomMargin: 48
wrapMode: Text.WordWrap
verticalAlignment: Text.AlignVCenter
horizontalAlignment: Text.AlignHCenter
font.pixelSize: 12
Component.onCompleted:
{
    if(modulType===2 || modulType===4)
    {
        lm35temp.visible=true
    }
    else
    {
        lm35temp.visible=false
    }
    if(modulType===4)
    {

if(sicaklikcontunious===true) lm35temp.text="Sürekli"
        else lm35temp.text="Tıklayınca"
    }
}

MouseArea {
    id: lm35tempmousearea
    anchors.fill: parent

    onClicked:{
        if(modulType===4)
        {
if(sicaklikcontunious===true) sicaklikcontunious=false
            else sicaklikcontunious=true
        }
        sendCommand(1)
    }
}

}

AnimatedImage {
    id: buzzerImage
    width: 48
    height: 48
    anchors.right: parent.right
    anchors.rightMargin: 0
    anchors.bottom: parent.bottom
    anchors.bottomMargin: 0
    source: "img/buzzer-off2.png"
    Component.onCompleted:
    {
        if (modulType===2 || modulType===3 ||
modulType===4) buzzerImage.visible=true
        else buzzerImage.visible=false
    }

    MouseArea {
        id: buzzerMouseArea
        anchors.fill: parent
        onClicked:
        {
            if(buzzerstatus===true)

```

```

        {
            buzzerstatus=false
        }
        else
        {
            buzzerstatus=true
        }
        sendCommand(1)
    }
}
}

AnimatedImage {
    id: alarmimage
    width: 48
    height: 48
    anchors.left: parent.left
    anchors.leftMargin: 5
    anchors.top: txt_baslik.bottom
    anchors.topMargin: 0
    source: "img/alarm-off.png"

    MouseArea {
        id: alarmmousearea
        anchors.fill: parent
        onClicked:
        {
            sendCommand(0)
        }
    }
}

Text {
    id: txt_durumaltsol
    x: 27
    y: 184
    width: 100
    height: 20
    color: "#ffffff"
    text: qsTr("---")
    visible: true
    font.bold: true
    horizontalAlignment: Text.AlignHCenter
    verticalAlignment: Text.AlignVCenter
    font.pixelSize: 12

    Component.onCompleted:{
        if(modulType===1)
        {
            txt_durumaltsol.visible=true
            txt_durumaltsol.text = "STABİL"
        }
        else if(modulType===3)
        {
            txt_durumaltsol.visible=true
            txt_durumaltsol.text = "Uyarı Başla:" +
kapiuyaricarpani + "sn."
        }
        else if(modulType===5 || modulType===6 ||
modulType===7)
        {

```

```

        txt_durumaltsol.visible=true
        txt_durumaltsol.text = "Durdu"
    }
    else
    {
        txt_durumaltsol.visible=false
    }
}

MouseArea {
    id: zcSourceMouseArea
    anchors.fill: parent
    onClicked: {
        if(modulType===1)
        {
            if(zckaynak===1)
            {
                //zcSource.text="STABİL"
                zckaynak=0
            }
            else
            {
                //zcSource.text="NORMAL"
                zckaynak=1
            }
        }
        else if(modulType===3)
        {
            if(kapiuyaricarpani>=10)kapiuyaricarpani=
kapiuyaricarpani * 2
            if((kapiuyaricarpani>254) ||
(kapiuyaricarpani<10))kapiuyaricarpani=10
            //txt_durumaltsol.text = "Uyarı Çarpanı:" +
kapiuyaricarpani
        }
        else if(modulType===5 || modulType===6 ||
modulType===7)
        {
            if(acmakesmestartstop===0)
            {
                acmakesmestartstop=1
                acmakesmeyenibilgi=1
                //txt_durumaltsol.text = "Açma için Say:"
            }
            else if(acmakesmestartstop===1)
            {
                acmakesmestartstop=2
                acmakesmeyenibilgi=1
                //txt_durumaltsol.text = "Kapama için
Say:"
            }
            else if(acmakesmestartstop===2)
            {
                acmakesmestartstop=0
                acmakesmeyenibilgi=1
                //txt_durumaltsol.text = "Durdu"
            }
        }
    }
    else

```

```

        {
            txt_durumaltsol.visible=false
        }
        sendCommand(1)
        //SEND CODE
    }
}

Text {
    id: txt_durumaltsag
    x: 130
    y: 184
    width: 110
    height: 20
    color: "#ffffff"
    text: qStr("Florasan")
    font.bold: true
    wrapMode: Text.NoWrap
    verticalAlignment: Text.AlignVCenter
    horizontalAlignment: Text.AlignHCenter
    font.pixelSize: 12

    Component.onCompleted:{
        if(modulType===1)
        {
            txt_durumaltsag.visible=true
        }
        else if(modulType===3)
        {
            txt_durumaltsag.visible=true
            txt_durumaltsag.text = "Alarm Başla: " +
kapialarmcarpani
        }
        else if(modulType===5 || modulType===6 ||
modulType===7)
        {
            txt_durumaltsag.visible=true
            txt_durumaltsag.text = acmakesmesuresi +"sn."
        }
        else
        {
            txt_durumaltsag.visible=false
        }
    }
    MouseArea {
        id: lamptypeMouseArea
        anchors.fill: parent
        z: 3
        onClicked: {

            if(modulType===1)
            {
                lambatipi = (lambatipi+1) % 3;
                sendCommand(1)
            }
            else if(modulType===3)
            {
                if(kapialarmcarpani>=10) kapialarmcarpani=
kapialarmcarpani * 2

```

```

        if((kapialarmcarpani>254) ||
(kapialarmcarpani<10))kapialarmcarpani=10
        sendCommand(1)
    }
    else if(modulType===5 || modulType===6 ||
modulType===7)
    {
        carpmasayisi=carpmasayisi+1
        txt_durumaltsag.text=""
        var temp3

if(acmakesmesuresi>86399)acmakesmesuresi=acmakesmesuresi+86400//1 gün
ekle
        else acmakesmesuresi=acmakesmesuresi*2//1 güne
gelmemişse saniyeyi 2 yle çarp

        if(acmakesmesuresi>2592000)
        {
            carpmasayisi=0
            acmakesmesuresi=10 // 30 günü aştıysa 10sn
ye geri dön
        }
        if(acmakesmesuresi>86399)//kaç gün var
        {
            temporary2=(acmakesmesuresi/86400)
            txt_durumaltsag.text= temporary2 +"gün,"
            temporary2=(acmakesmesuresi%86400)
        }
        else temporary2=acmakesmesuresi
        if(acmakesmesuresi>3599)//kaç saat var
        {
            temporary2=(temporary2/3600)
            temp3=temporary2
            txt_durumaltsag.text=txt_durumaltsag.text
+ temp3 + "saat,"
var
            temporary2=(acmakesmesuresi%3600)//kaç dk

        }
        if(temporary2>59)//kaç saniye var
        {
            temporary2=(temporary2/60)
            temp3=temporary2
            txt_durumaltsag.text=txt_durumaltsag.text
+ temp3 + "dk.,"
            temporary2=acmakesmesuresi%60
        }
        txt_durumaltsag.text=txt_durumaltsag.text +
temporary2 + "sn."
    }
    else
    {
        txt_durumaltsag.visible=false
    }
}
}
}

function sendCommand(writeenable)
{

```

```

if (socket.state !== 3) {
    return 0
}
var senderarray = [];
for(var i=0;i<32;i++)
{
    senderarray[i]=0;
}
senderarray[3]=modulId
senderarray[7]=255//anakarttan gelmiş gibi algılayacak
modüller
senderarray[14]=modulType
senderarray[15]=writeenable//WRITENABLE

if(modulType===1) //DIMMER
{
    senderarray[16]= 1 //dimmervalue
    if(butonaktif){
        if(slider.value < 128) {
            senderarray[17] = Math.round(slider.value * 48 /
128)
        } else {
            senderarray[17] = Math.round((slider.value-128) *
206 / 128) + 48
        }

    }else senderarray[17] = 0
    senderarray[18] = 12 //alarmbelirteç
    senderarray[19]= 0 //okuma
    senderarray[20]= 21 //kaynak bilgisi
    senderarray[21] = zckaynak
    senderarray[22] = 22 //lambatipi
    senderarray[23] = lambatipi
    senderarray[24] = 3 //lm35
    senderarray[25] = 0 //okuma
    senderarray[26] = 2 //LED
    senderarray[27] = ledstatus
}
else if(modulType===2) //DOGALGAZ
{
    senderarray[16] = 5;//GAZ ROLESİ
    senderarray[17] = butonaktif;//GAZ ROLESİ
    senderarray[18] = 12;//alarm belirteç
    senderarray[19] = 0;//alarm bilgisi okuma
    senderarray[20] = 4;//MQ4 analog gaz verisi
    senderarray[21] = 0; // MQ4 analog gaz verisi okuma
    senderarray[22] = 2 //LED
    senderarray[23] = ledstatus
    senderarray[24] = 7 //Buzzer
    senderarray[25] = buzzerstatus
}
else if(modulType===3) //KapıKontrol
{
    senderarray[16] = 6;//KAPI DURUMU
    senderarray[17] = 0;//kapı durumu okuma
    senderarray[18] = 12;//alarm belirteç
    senderarray[19] = 0;//alarm bilgisi okuma
    senderarray[20] = 2 //LED
    senderarray[21] = ledstatus
    senderarray[22] = 7 //Buzzer

```



```

        senderarray[23] = buzzerstatus
        senderarray[24] = 20 //KAPI UYARI
ÇARPANI
        senderarray[25] = kapiuyaricarpani
        senderarray[26] = 19 //KAPI ALARM
ÇARPANI
        senderarray[27] = kapialarmcarpani
    }
    else if(modulType==4) //Sıcaklık-Nem
    {
        senderarray[16] = 8;//SICAKLIK VERILERININ OKUNMASINI
SAGLAYACAK
        senderarray[18] = 12;//alarm belirteç
        senderarray[19] = 0;//alarm bilgisi
        senderarray[20] = 16;//sicakliknem sensörü aktifmi
belirteç
        senderarray[21] = 0;
        senderarray[22] = 17;//devamlı gönderilsin mi?
        senderarray[23] = sıcaklikcontunious;
        senderarray[24] = 2 //LED
        senderarray[25] = ledstatus
        senderarray[26] = 7 //Buzzer
        senderarray[27] = buzzerstatus
    }
    else if(modulType==5) //ÜTÜ
    {
        senderarray[16] = 9;//modul tip2 role
        senderarray[17] = butonaktif;//modul tip2 role
        senderarray[18] = 12;//alarm belirteç
        senderarray[19] = 0;//alarm bilgisi
        senderarray[20] = 3;//lm35 verisi
        senderarray[21] = 0;//lm35 verisi
        if(acmakesmeyenibilgi || (!writeenable))
        {
            senderarray[22] = 10;//kesmestartstop
            senderarray[23] =
acmakesmestartstop;//acmakesmestartstop
            acmakesmeyenibilgi=false
        }
        if(writeenable)
        {
            senderarray[31] = carpmasayisi
        }
    }
    else if(modulType==6) //KOMBI
    {
        senderarray[16] = 9;//modul tip2 role
        senderarray[17] = butonaktif;//modul tip2 role
        senderarray[18] = 12;//alarm belirteç
        senderarray[19] = 0;//alarm bilgisi
        senderarray[20] = 3;//lm35 verisi
        senderarray[21] = 0;//lm35 verisi
        if(acmakesmeyenibilgi || (!writeenable))
        {
            senderarray[22] = 10;//kesmestartstop
            senderarray[23] =
acmakesmestartstop;//acmakesmestartstop
            acmakesmeyenibilgi=false
        }
        if(writeenable)
        {

```

```

        senderarray[31] = carpmasayisi
    }
}
else if(modulType===7) //KLIMA
{
    senderarray[16] = 9;//modul tip2 role
    senderarray[17] = butonaktif;//modul tip2 role
    senderarray[18] = 12;//alarm belirteç
    senderarray[19] = 0;//alarm bilgisi
    senderarray[20] = 3;//lm35 verisi
    senderarray[21] = 0;//lm35 verisi
    if(acmakesmeyebilgi || (!writeenable))
    {
        senderarray[22] = 10;//kesmestartstop
        senderarray[23] =
acmakesmestartstop;//acmakesmestartstop
        acmakesmeyebilgi=false
    }
    if(writeenable)
    {
        senderarray[31] = carpmasayisi
    }
} //console.log(senderarray)
//senderarray = protocolid.sifrele(senderarray)
console.log("JS: " + senderarray)
tcpsocket.write(senderarray)
//console.log(senderarray)
//senderarray = protocolid.sifreCoz(senderarray)
//console.log(senderarray)
}
function butonsondur()
{
    rectangle.color = "#1d1d1d"
    rectangle.border.color = "#050505"
    image.source = "img/off_icon.png"
    image.update()
}

function butonyak()
{
    rectangle.color = "#004356"
    rectangle.border.color = "#0076b2"
    image.source = "img/on_icon.png"
    image.update()
}

function commanderSub(bytenumber, veri)
{
    if(veri[bytenumber]===0) //NO OPERATION
    {
    }
    else if (veri[bytenumber]===2) //LED AYAR - GERİ DÖNÜŞLÜ SON
HALİ
    {
        var ledstatusx = veri[bytenumber+1]
        if(ledstatusx===0)
        {
            ledimage.source = "img/led-off.png"
            ledimage.update()
        }
    }
}

```

```

else if(ledstatusx===1){
    ledimage.source = "img/led-anim.webp"
    ledimage.update()
    ledimage.playing=true
}
else if(ledstatusx===2){
    ledimage.source = "img/led-on.png"
    ledimage.update()
}
}

else if(veri[bytenumber]===7)//BUZZER--GERİ DÖNÜŞLÜ SON HALİ
{
    if (veri[bytenumber+1]>0)
    {
        buzzerImage.source="img/buzzer-on2.png"
        buzzerImage.update()
        //buzzerstatus=1
    }
    else
    {
        buzzerImage.source="img/buzzer-off2.png"
        buzzerImage.update()
        //buzzerstatus=0
    }
}

else if(veri[bytenumber]===1)//DIMMER AC KAPA AYAR--GERİ
DÖNÜŞLÜ SON HALİ
{
    var dimmervalue = veri[bytenumber+1]
    if(dimmervalue===0)
    {
        //                rectangle.color = "#1d1d1d"
        //                rectangle.border.color = "#050505"
        //                image.source = "img/off_icon.png"
        //                image.update()
        butonsondur()
        // slider.enabled=false
        butonaktif=false
    }
    else if(dimmervalue>0)
    {
        //                rectangle.color = "#004356"
        //                rectangle.border.color = "#0076b2"
        //                image.source = "img/on_icon.png"
        //                image.update()
        butonyak()
        slider.enabled=true
        if(slider.value===0){slider.value=255}
        butonaktif=true
        // slider.value=dimmervalue
    }
}

else if(veri[bytenumber]===3)//LM35 - GERİ DÖNÜŞLÜ SON HALİ
{
    lm35temp.text= veri[bytenumber+1]
    //lm35temp.visible=true
}

else if(veri[bytenumber]===21)//ZC kaynak değiştir--GERİ
DÖNÜŞLÜ SON HALİ

```

```

{
    if (veri[bytenumber+1]==0)
    {
        txt_durumaltsol.text="STABİL"
    }
    else
    {
        txt_durumaltsol.text="NORMAL"
    }
}

else if(veri[bytenumber]==22)//Lamba Tipi değiştir--GERİ
DÖNÜŞLÜ SON HALİ
{
    var lambatipix =veri [bytenumber+1]
    if(lambatipix==0)
    {
        txt_durumaltsag.text="Halojen (Dim) "
    }
    else if(lambatipix==1)
    {
        txt_durumaltsag.text="Florasan (Reg) "
    }
    else if(lambatipix==2)
    {
        txt_durumaltsag.text="LED (Reg) "
    }
}

else if(veri[bytenumber]==4)//EV-ICI ANALOG GAZ VERISI - GERİ
DÖNÜŞLÜ SON HALİ
{
    lm35temp.text = "Gaz:" + veri[bytenumber+1]//lm35temp
    yerine gaz datası yazılıyor tekrardan değişken tanımlamamak için
    lm35temp.visible = true
}
//          else if(veri[bytenumber]==15)//EV-ICI DIJITAL GAZ
VERISI
//          {
//              mq4digital = veri[bytenumber+1]
//          }
else if(veri[bytenumber]==5)//GAZ ROLESİ ON-OFF --GERİ
DÖNÜŞLÜ SON HALİ
{
    var gazrole = veri[bytenumber+1];
    if(gazrole==0){
        //          image.source="off_icon"
        //          image.update()
        butonaktif=false
        butonsondur()
    }
    else
    {
        //          image.source="on_icon"
        //          image.update()
        butonaktif=false
        butonsondur()
        guncellenmeTarih.text= "Son Kesilme:" + new
        Date().toLocaleDateString(Qt.locale("tr_TR"),"dd-MM-yyyy")
    }
}

```

```

        + new
Date().toLocaleTimeString(Qt.locale("tr_TR")," hh:mm:ss")
    }
}
//          else if(veri[bytenumber]==18)//MQ4 calisiyormu
//          {
//              veri[bytenumber+1] = mq4aktif;
//          }
//      #endif

else if(veri[bytenumber]==6)//KAPI DURUM VERISI --GERİ
DÖNÜŞLÜ SON HALİ
{
    var kapidurumu=veri[bytenumber+1]
    if (kapidurumu==0){
        txt_durumAna.text="KAPALI"
        txt_durumAna.color = "#008f00"
        //butonsondur()
    }else{
        txt_durumAna.text="AÇIK"
        txt_durumAna.color = "#fd7262"
        txt_durumAlt.text= "Son Açılma:" + new
Date().toLocaleDateString(Qt.locale("tr_TR"),"dd-MM-yyyy")
        + new
Date().toLocaleTimeString(Qt.locale("tr_TR")," hh:mm:ss")
        guncellenmeTarih.visible= false
        //butonyak()
    }
}
else if(veri[bytenumber]==19)//KAPI ALARM ÇARPANI --GERİ
DÖNÜŞLÜ SON HALİ
{
    kapialarmcarpani=veri[bytenumber+1]
    txt_durumaltsag.text = "Alarm Başla:" + kapialarmcarpani +
"sn."
}
else if(veri[bytenumber]==20)//KAPI UYARI ÇARPANI --GERİ
DÖNÜŞLÜ SON HALİ
{
    kapiuyaricarpani=veri[bytenumber+1]
    txt_durumaltsol.text = "Uyarı Başla:" + kapiuyaricarpani +
"sn."
}

else if(veri[bytenumber]==8)//SICAKLIK-NEM VERISI --GERİ
DÖNÜŞLÜ SON HALİ
{
    sicaklik = (veri[28]* 255+veri[29])/10
    nem = (veri[30]* 255+veri[31])/10
    txt_durumAna.text = sicaklik + " C"
    txt_durumAlt.text = "% " + nem
    //          veri[28]=
(uint8_t)(rawT>>8);          //TEMP HIGH
    //          veri[29]= (uint8_t)(rawT&0x00FF);
    //TEMP LOW
    //          veri[30]=
(uint8_t)(rawH>>8);          //HUMIDITY HIGH
    //          veri[31]= (uint8_t)(rawH&0x00FF);
    //HUMIDITY LOW

```

```

    }

    else if(veri[bytenumber]==16)//SICAKLIK-NEM MODULU AKTIFMI --
GERİ DÖNÜŞLÜ SON HALİ
    {
        if(veri[bytenumber+1]==0)
        {
            txt_durumAna.text = "Modül yanıtız!"
            txt_durumAlt.text = "Modül yanıtız!"
        }
        else
        {
            txt_durumAna.text = sicaklik + " C"
            txt_durumAlt.text = "% " + nem
        }
    }
    else if(veri[bytenumber]==17)//SICAKLIK-NEM MODULU CONTUNIOUS
DATA --GERİ DÖNÜŞLÜ SON HALİ
    {
        if(veri[bytenumber+1]==1)
        {
            lm35temp.text="Sürekli"
        }
        else
        {
            lm35temp.text="Tıklayınca"
        }
    }

    else if(veri[bytenumber]==9)//MODUL TİP 2 RÖLE --GERİ DÖNÜŞLÜ
SON HALİ
    {
        if(veri[bytenumber+1]>0)
        {
            butonyak()
            butonaktif=true
        }
        else
        {
            butonsondur()
            butonaktif=false
        }
    }

    else if(veri[bytenumber]==10)//KESME START STOP BİLGİSİ
    {
        var temp2
        acmakesmestartstop=veri[bytenumber+1]

        if(acmakesmestartstop==0) {
            acmakesmesayaci=0
            txt_durumaltsol.text="Durdu"
            txt_durumAlt.text="Sayma Bitmiş."
            return 0
        }
        else if(acmakesmestartstop==1)
        {
            txt_durumaltsol.text="Açma için Say:"
        }
        else if(acmakesmestartstop==2)
        {

```

```

        txt_durumaltsol.text="Kapama için Say:"
    }
    acmakesmesayaci= veri[28]*255*255*255 + veri[29]*255*255 +
    veri[30]*255 + veri[31]
    txt_durumAlt.text="Geçen Süre:"
    if(acmakesmesayaci>86399)
    {
        temporary=(acmakesmesayaci/86400)
        txt_durumAlt.text=txt_durumAlt.text + temporary
        temporary=(acmakesmesayaci%86400)
    }
    else temporary=acmakesmesayaci
    if(acmakesmestartstop>3599)
    {
        temporary=(temporary/3600)
        temp2=temporary
        txt_durumAlt.text=txt_durumAlt.text + temp2 + "saat,"
        temporary=(acmakesmesayaci%3600)
    }
    if(temporary>59)
    {
        temporary=(temporary/60)
        temp2=temporary
        txt_durumAlt.text=txt_durumAlt.text + temp2 + "dk.,"
        temporary=acmakesmesayaci%60
    }
    txt_durumAlt.text=txt_durumAlt.text + temporary + "sn."
}

else if(veri[bytenumber]==12)//BÜTÜN ALARMLAR
{
    if(veri[bytenumber+1])
    {
        alarmimage.source="img/alarm-yanson.webp"
        alarmimage.update()
        alarmimage.playing=true
        guncellenmeTarih.text="Son Alarm:" + new
        Date().toLocaleDateString(Qt.locale("tr_TR"),"dd-MM-yyyy")
        + new
        Date().toLocaleTimeString(Qt.locale("tr_TR")," hh:mm:ss")
        //ALARM BİLDİRİM KODU
    }
    else
    {
        alarmimage.source="img/alarm-off.png"
        alarmimage.update()
        // ALARM BİLDİRİM KODU
    }
}

}

Timer {
    id:slidertimer
    interval: 400
    repeat: true
    onTriggered: {
        if(newsliderdata){
            if(slidersifirdan255===true)

```

```

        {
            slidersifirdan255=false
        }else{
            sendCommand(1)
        }
        newsliderdata=0
    }
}

FDRProtocol {
    id: fdrprotocol
}

Connections {
    target: socket
    onDecrypted:{
        //console.log("Kontrollegeldi: " + veri)
        // console.log(veri)
        //var cozulenVeri = protocolid.sifreCoz(veri)
        //console.log("COZULEN: " + cozulenVeri)
        var gelenModulId = fdrprotocol.getModulId(veri)
        var gelenModulType = veri[14]
        //console.log(gelenModulId, gelenModulType)

        if( (gelenModulId === modulId) && (gelenModulType ===
modulType) ){
            // MESAJ BIZE GELMIS
            console.log("Mesaj bize gelmis.")
            console.log(gelenModulId, gelenModulType)
            //console.log("Cozulen " + cozulenVeri)
            for(var i=16; i<28; i += 2) {
                commanderSub(i, veri)
            }
        }
    }
}
}

```


Ek-3

Akıllı Ev Otomasyonu için Maliyet Analizi

Tablo E3.1. Tasarlanan Uygulamanın Birim Maliyet Analizi

Komponent	Tipi	Parametreleri	Üretici Firma	Adet	Birim Fiyatı(TL)
STM32L052C6 T6	32-bit ARM işlemci	32KB Flash memory, 48MHz max. clock freq.	STMicroelectroni cs	2	17,9229
100mmx100mm PCB Siparişi(8 adet)	Çift- taraflı, Mavi	FR-4 Aluminum	PCBWay	1	291,54
STM32F030F4P 6	32-bit ARM işlemci	16KB Flash memory, 48MHz max. clock freq.	STMicroelectroni cs	6	5,37690
Kapasite	0805 SMD	100nF	-	6	0,17857
Kapasite	0603 SMD	10pF	-	2	0,05861
Alüminyum Kapasite	RADIAL	220UF 20% 10V	-	2	0,225
LED	0805 SMD	Kırmızı, Amber, Yeşil, Beyaz	-	4+4+3+3	0,24929
nRF24L01+	Modül	2.4GHz GFSK, SPI	Nordic Semiconductor	8	7,51763

LPT-100	Wi-Fi modülü	2.4GHz 802.11b/g/n	Waveshare	1	70,38
Header 1x2	Erkek	5.08mm	-	1+3+6+3	0,69
Header 1x2	Erkek	2.54 mm	-	2+2+3	1,90676
Header 1x5	Erkek	2.54mm	-	1+1+3+3	0
Header 1x4	Erkek	2.54mm	-	2+1+6	0
Header 1x3	Erkek	2.54mm	-	2+6+3	0
Header 1x6	Erkek	2.54mm	-	1	0
AM2320	Sıcaklık-Nem Sensörü	±%3RH sapma, ±0.5°C sıcaklık sapması, I2C	AOSONG	1	19,61
BC817-25	NPN transistör	45V 0.8A SOT-23	-	4+6+9+6	0,12072
Direnç	2512 SMD	0.027R	-	1+1	5,08809
Direnç	0603 SMD	1K	-	4	0,02207
Direnç	0805 SMD	4.7K, 10K, Ferrit Bead, 4.7K, 470, 470, 470, 470, 4.7K, 4.7K, 4.7K, 100K, 10K, 10K, 100K(Herbirinden birer tane)	-	15	0,02131
Switch	SPST	-	-	1+1+3+3	0,46715

HLK5M-05	AC/DC Converter	5V 5W	Hi-Link	1+1	77,61
AMS1117-3.3	Lineer Regulator	3.3V 1A SOT-223	-	1+1+3+3	0,84296
MCP6022-I/SN	Opamp	10MHz Rail-to-Rail Output SOIC-8	MicroChip	1	5,45840
Varistör	-	220VAC Disk 7mm	-	1+1+3+3	0,48
Kapasite	0805 SMD	10pF	-	2	0,10931
Kapasite	0603 SMD	100nF	-	3	0,41071
Kapasite	0805 SMD	100nF	-	1	0,17857
Elektrolitik Kapasite	Radial-0.4	220uF , 400V	Rubicon	1	9,53854
Polyester Kapasite	Radial-5mm	47nF , 250V	-	1	1,36883
GBU4K-BP	Köprü Diyot	4A, 800V	MCC	1	1,93
1N4004	Diyot	1A, 400V	-	1+3+3	0,34572
Bobin	Toroid	330uH, 10A	-	1	24,37
Direnç	Axial 0.4	100K, 100K, 470R, 330R(Herbirinden birer tane)	-	4	0,05708

BTB12-600	Triyak	12A, 600V	STMicroelectroni cs	1	2,9
PC814	Optokopl ör	220VAC Input	-	1	1,16
74HC14D	Schmitt Trigger Inverter	3.3V, 5V	-	1	0,77
MOC3023M	Triyak Sürücü Çıkışı	400V peak	ON Semiconductor	1	1,63
Direnç	0805 SMD	1K, 470R, 470R, 470R, 470R, 4.7K, 10K, 1K, 1K, 10K, 10K, 340K, 100K, 10K, 10K, 10K, 10K, 33K, 4.7K, 100K, 10K, 680R, 4.7K, 4.7K, 33K, 220R, 4.7K, 4.7K, 4.7K, 4.7K, 1K, 1K(Hepsinden birer adet)	-	32	0,03577
Ferrit Bead	0603 SMD	100R	-	2	0,08999
JQC-3F(T73)- 5VDC	Röle	5V	-	1+3	1,82

LM35	Sıcaklık Sensörü	0.5°C sapma, 5V, 10-mV/°C lineer	Texas Instruments	1+3	8,39
Kapasite	0805 SMD	100nF	-	9+9	0,17857
Kapasite	Radial	220uF – 10V	-	6+6	0,225
USB Micro SMT	Type-B	-	-	3	0,89144
Buzzer	Magnetic	5V	-	3	1,28
Direnç	0805 SMD	4.7K, 470, 4.7K, 10K, FERRIT BEAD, 1K, 1K, 1K(Herbirinden 3er adet)	-	24	0,03577
HLK-PM01	AC/DC converter	5V, 3W	-	3	30,35
Kapasite	Radial	330uF – 35V	-	3	0,63998
Direnç	0805 SMD	FERRIT BEAD, 4.7K, 470, 4.7K, 10K, 1K (Herbirinden 3er adet)	-	18	0,03577
JQX-102F	Röle	220VAC, 25A	-	3	11,80
HLK-PM12	AC/DC converter	12V, 3W	Hi-Link	3	51,74
MCP1702-500E	Lineer Regülatör	3.3V, 1A, SOT233	MicroChip	3	2,86

TORK Doğalgaz Solenoid Valf	Valf	220VAC bobin gerilimi	TORK	1	65
Dişi Priz	Priz	-	AY-KA	2	1
Erkek Priz	Priz	-	AY-KA	2	1
Trafo 3VA	İzoleli	12VAC dual çıkış	MERVESAN	1	15,34
nRF24L01+PA	Modül	2.4GHz GFSK, SPI, Power Amp	Nordic Semiconductor	2	25,96
Toplam Kalem, Toplam Fiyat				178	1214,6556 2

- [1] Honeywell, "The Kitchen Computer", 1967.
<https://www.afcdud.com/fr/smart-city/422-how-the-history-of-smart-homes.html>.
- [2] K. Kayaer, "Mikrodenetleyicili ev otomasyonu ve güvenlik sistemi", Yüksek Lisans Tezi, YTÜ, 2002.
- [3] P. Karataş, "Mikrodenetleyici tabanlı ev otomasyonu", Yüksek Lisans Tezi, Çukurova Üni., 2004.
- [4] F. S. Karakaş, "Bilgisayar kontrollü akıllı ev otomasyonu", Yüksek Lisans Tezi, İstanbul Üni., 2006.
- [5] H. Kongaz, "Akıllı Ev Otomasyonun Mikrodenetleyici ile gerçekleştirilmesi", Yüksek Lisans Tezi, YTÜ, 2007.
- [6] N. E. Karagöz, "Home Area Networking with universal plug and play", Yüksek Lisans Tezi, İTÜ, 2007.
- [7] L. Birgül, "Bilgisayar Destekli Ev Otomasyonu", Yüksek Lisans Tezi, YTÜ, 2007.
- [8] G. Arslanoğlu, "RF ev otomasyonu", Yüksek Lisans Tezi, Gazi Üni., 2009.
- [9] B. Cincirop, "GSM kontrollü akıllı ev otomasyonu", Yüksek Lisans Tezi, Sakarya Üni., 2009.
- [10] K. S. b. Bahir, "Arduino tabanlı akıllı ev otomasyon sistemi", Yüksek Lisans Tezi, Yüzüncüyıl Üni., 2016.
- [11] S. Akkuş, "Gömülü Sistem Tabanlı, kriptolu TCP/IP veri haberleşmesi ve uygulaması", Yüksek Lisans Tezi, Marmara Üni., 2016.
- [12] Niçin ARM çok popüler ve ARM Mimarisi, Elprocus.com
<https://www.elprocus.com/arm-architecture/>.
- [13] Nordic Semiconductor, nRF24L01+ datasheet,
https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf.
- [14] LonelyWolf, nRF24L01+ kütüphanesi,
<https://github.com/LonelyWolf/stm32/tree/master/nrf24l01>.
- [15] Brad, PCB Trace Width Calculator Tool, Circuitcalculator.com
<http://circuitcalculator.com/wordpress/2006/01/31/pcb-trace-width-calculator/>.
- [16] L. Rozenblat, PCB Trace Spacing Calculator, Smmps.us
<https://www.smmps.us/pcbtracespacing.html>.

- [17] R. Barış Dai, M. Turhan, D. Yıldırım and C. Uçak, "A dimmer circuit for various lighting devices," *2013 8th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, 2013, pp. 278-282.
- [18] Google Ev Asistanı Hakkında, Google LLC,
<https://assistant.google.com/platforms/speakers/>.
- [19] Mi Ev Asistanı Hakkında, <https://xiaomi-mi.com/mi-smart-home/>.



Tezden Üretilmiş Yayınlar

İletişim Bilgisi: yefedar@gmail.com

Konferans Bildirileri

1. Y.E. Fedar, H. Sedef, “Gelişmiş Mikrodenetleyicili Akıllı Ev Otomasyonu Sistemi ve Uygulaması”, *Ereğli Uluslararası Bilim ve Akademi Kongresi*, pp. 286-292, 2019.

