

T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI



**ZIGBEE VE IOT KULLANARAK EV OTOMASYON
SİSTEMİNİN GELİŞTİRİLMESİ**

TAREK H. ALI ADEB

YÜKSEK LİSANS TEZİ

Doç. Dr. Abdulkadir KARACI

ARALIK - 2021
KASTAMONU

TAAHHÜTNAME

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bütün bilgilerin etik davranıř ve akademik kurallar çerçevesinde elde edilerek sunulduđunu; ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynađına eksiksiz atıf yapıldıđını, bilimsel etiđe uygun olarak kaynak gösterildiđini bildirir ve taahhüt ederim.

TAREK H. ALI ADEB

ÖZET

YÜKSEK LISANS TEZİ

ZİGBEE VE IOT KULLANARAK EV OTOMASYON SİSTEMİNİN GELİŞTİRİLMESİ

TAREK H. ALI ADEB

KASTAMONU ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI
DANIŞMAN: Doç. Dr. Abdulkadir KARACI

Nesnelerin İnterneti'nin (IoT) hızlı ilerlemesi, hayatı her yönden daha kolay ve basit hale getirmiştir. Günümüzde nesnelerin interneti dünya çapında yaygın olarak kullanılmaktadır. Çoğu insan otomatik sistemleri geleneksel manuel sistemlere tercih etmektedir. Nesnelerin İnterneti, insanların günlük yaşam kalitesini iyileştirmek, ekonomi ve toplum üzerinde net bir etkiye sahip olmak amacıyla birçok harika uygulamaya sahiptir. En iyi, en önemli ve en etkili Nesnelerin İnterneti uygulamaları arasında akıllı ev veya ev otomasyon sistemleri gelmektedir. Bunun yanı sıra, akıllı ev arayanların sayısı her geçen gün artmakta ve artık daha fazla şirket akıllı ev hizmetleri sunmakla ilgilenmektedir. Bu çalışmanın amacı, evi gerçek zamanlı olarak uzaktan kontrol etmek ve izlemek için düşük maliyetli bir ev otomasyon sistemi geliştirmektir. Bu amaçla, Arduino, Raspberry Pi, ZigBee ve IoT kullanarak dört düğümden oluşan bir sistem geliştirilmiştir. Geliştirilen sistemde, voltaj, akım, duman, gaz, sıcaklık ve nem seviyesi, yangın ve ışık durumu gibi bilgiler uzaktan izlenebilmektedir. Bunun yanı sıra, evin ana gücü, klima, ısıtıcı, TV ve buzdolabı uzaktan kontrol edilebilmektedir. Gerçekleştirilen deneysel çalışma sonuçlarına göre tasarlanan sistemin Web ve Android temelli uygulamalar aracılığıyla dünyanın her yerinden gerçek zamanlı olarak izleme ve kontrol işlemini başarılı bir şekilde yerine getirdiği gözlenmiştir.

ANAHTAR KELİMELER: Ev otomasyon sistemi, Akıllı ev, Zigbee, Nesnelerin İnterneti (IoT), Raspberry Pi, Arduino.

Aralık 2021, Sayfa, 121

ABSTRACT

MSC THESIS

DEVELOPMENT OF HOME AUTOMATION SYSTEM USING ZIGBEE AND IOT

TAREK H. ALI ADEB

**KASTAMONU UNIVERSITY INSTITUTE OF SCIENCE
DEPARTMENT OF MATERIALS SCIENCE AND ENGINEERING
SUPERVISOR: Assoc. Prof. Dr. Abdulkadir KARACI**

The fast progress of the Internet of Things (IoT), has made life easier and simpler in all aspects. Nowadays internet of things widely used around the world, most people prefer the automatic systems over the traditional manual systems. The Internet of Things aims to make things to be connected with anyone, anything, at anywhere and anytime in a perfectly, using any network, path, and any service. Internet of Things has many great applications which purpose is to improve the quality of people's daily life and have a clearly impact on the economy and society. One of the best and most important and effective internet of things applications is smart home or home automation systems, the number of people searching for smart homes always increases, also more companies are now interested in providing smart homes services. The purpose of the proposed project is to develop a home automation system using ZigBee and IOT technology to control and monitor the home remotely in real time, as well as used Raspberry Pi to acts as a responsible server to connect the system with the cloud, also for monitor and control the system. In additional, the system proposed provides the necessary protection to house for fire detection and alarm when there is a gas or smoke leak, and monitor if the current is stable or high, and voltage is low or high of the home. It has been observed that the system designed according to the results of the experimental study successfully performs real-time monitoring and control from all over the world through Web and Android-based applications.

KEYWORDS: Home automation system, Smart home, Zigbee, Internet of Things (IoT), Raspberry Pi.

December 2021, Page, 121

TEŐEKKÜR

Öncelikle, bu projeyi tamamlamamda bana destek olan danıőmanım Doç. Dr. Abdulkadir KARACI'ya teőekkür etmek istiyorum. Kastamonu Üniversitesi'ndeki meslektaőlarımıza ve Kastamonu'daki Libya toplumuna destekleri için teőekkür ederim. Nihayet; yetiőtirilmem ve eđitimimle ilgilenen anne ve babamın ruhlarına ve bu tezi çalıőma ve bitirme konusunda bana güven veren ailemin sürekli desteđi için őükranlarımı sunarım.

Tarek H. Ali ADEB

Kastamonu, 2021

İÇİNDEKİLER

Sayfa

TEZ ONAYI	ii
TAAHHÜTNAME	iii
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ŞEKİLLER DİZİNİ	ix
TABLolar DİZİNİ	xi
SİMGELER VE KISALTMALAR DİZİNİ	xii
1. GİRİŞ	1
1.1 Çalışmanın Amacı	1
1.2 Çalışmanın Problemi	2
1.3 Çalışmanın Önemi	3
2. LİTERATÜR İNCELEMESİ	4
3. TEORİK ARKA PLAN	8
3.1 Nesnelerin İnterneti	8
3.1.2 Nesnelerin İnterneti Uygulamaları	10
3.2 Ev Otomasyonu	12
3.2.1 Ev Otomasyon Sistemlerinin Avantajları	14
3.3 MQTT Protokolü	15
3.4 Bulut Bilişim	16
3.4.1 Bulut Bilgisayar Hizmetleri	18
3.4.1.1 Hizmet olarak yazılım (SaaS)	18
3.4.1.2 Hizmet olarak platform (PaaS)	18
3.4.1.3 Hizmet olarak altyapı (IaaS)	19
3.5 Zigbee Teknolojisi	19
3.5.2 Zigbee Mimarisi	20
3.5.3 Veri Transfer Modları	21
3.5.4 Zigbee Ağ Topolojileri	21
3.5.5 Zigbee Teknoloji Uygulamaları	22
4. UYGULAMA VE DENEYSEL SONUÇLAR	23
4.2 Ana Düğüm	26
4.2.2 XBee Modülü (Koordinatör)	27
4.2.3 Arduino Leonardo kartı	28
4.2.4 Raspberry Pi 3 B	29
4.3 Düğüm I: Ev Güç Kontrol Düğümü	29
4.3.2 SCT013-000 Bölünmüş Çekirdekli Akım Sensörü	31
4.3.2.2 Akım sensörünün Arduino Leonardo kartı ile arabirilenmesi	32
4.3.3 ZMPT101B AC Tek Fazlı Voltaj Sensörü Modülü	33
4.3.4 Kontaktör	34
4.3.5 Röle	35
4.3.6 Arduino Leonardo Kartı	35
4.3.7 XBee Uç Cihazı	36
4.3.8 20 x 4 LCD	36

4.4 Dügüm II: İklim Kontrol Dügümü	36
4.4.2 MQ2 Sensörü	39
4.4.3 DH22 Sıcaklık ve Nem Sensörü	40
4.4.4 Arduino Leonardo Kartı	41
4.4.5 Xbee Uç Cihazı.....	41
4.4.6 20 x 4 LCD	42
4.4.7 Röleler	42
4.5 Dügüm III: Yangın ve Cihaz Kontrol Dügümü	42
4.5.2 LDR Işığa Bağlı Direnç	44
4.5.3 KY-026 Alev Sensörü Modülü.....	45
4.5.4 TV ve Buzdolabı Röleleri.....	46
4.5.5 Arduino Leonardo Kartı	46
4.5.6 XBee Uç Cihazı	47
4.5.7 16 x 2 LCD	47
4.6 XBee Modüllerini Yapılandırma.....	47
4.6.2 Parametrelerin Değerlendirilmesi.....	51
4.7 Yazılım Bölümü	52
4.7.1 Düşük Seviyeli Program	52
4.7.2 Yüksek Seviyeli Program	61
4.8 IOT Sisteminde Kullanılan Bulut Hizmeti.....	69
4.8.2 Ev Otomasyon Sistemi kanalı.....	71
4.9 Tasarlanan Android Uygulaması.....	73
4.10DeneySEL Sonuçlar ve Tartışma.....	75
5. SONUÇ	78
KAYNAKLAR.....	79
EKLER.....	85
EK-1: Ana Dügümün Kodu.....	86
EK-2: Dügüm I Kodu	89
EK-3:Dügüm II Kodu.....	93
EK-4: Dügüm III Kodu	97
EK-5: Ana Programın Görsel Stüdyo Kodu.....	101
EK-6: İş İstasyonu İçin Görsel Stüdyo Kodu	114
ÖZGEÇMİŞ.....	121

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1 Nesnelerin İnterneti (Patel et al., 2016).....	8
Şekil 3.2 Her şeyin İnterneti (Srinivasan vd., 2019).....	12
Şekil 3.3 MQTT Mimarisi (Soni ve Makwana, 2017).....	16
Şekil 3.4 Zigbee Teknolojisi (Agarwal, 2014).....	19
Şekil 3.5 Zigbee İttifakı (Kinsella, 2019).....	20
Şekil 3.6 Zigbee Topolojilerinin Türleri (Kumar ve Mane, 2016).....	21
Şekil 4.1 Tasarlanmış sistemin genel görünümü	24
Şekil 4.2 Önerilen modelin genel tasarımı.....	25
Şekil 4.3 Ana düğümün genel tasarımı ve görünümü	27
Şekil 4.4 Ana düğümün elektronik tasarımı.	27
Şekil 4.5 XBee modülü	28
Şekil 4.6 Arduino Leonardo kartı	29
Şekil 4.7 Düğüm I için tasarlanmış sistem.....	30
Şekil 4.8 Düğüm I'in elektronik tasarımı.	31
Şekil 4.9 SCT013-00 Akım Trafosu.	32
Şekil 4.10 Arduino için akım sensörü durum devresi.	33
Şekil 4.11 ZMPT101B AC tek fazlı voltaj sensörü modülü.	34
Şekil 4.12 Tek Fazlı Kontaktör.	35
Şekil 4.13 Röle modülü.....	35
Şekil 4.14 Düğüm I yerel izleme görüntüsü.....	36
Şekil 4.15 Düğüm II için tasarlanmış sistemin mimarisi.	38
Şekil 4.16 Düğüm II'nin elektronik tasarımı.....	39
Şekil 4.17 MQ2 gaz sensörü	39
Şekil 4.18 DHT22 Sıcaklık ve nem sensörü.	41
Şekil 4.19 Düğüm II yerel izleme	42
Şekil 4.20 Düğüm III için uygulanan sistem.....	43
Şekil 4.21 Düğüm III'ün elektronik tasarımı.	44
Şekil 4.22 LDR Sensörü.....	45
Şekil 4.23 KY 026 sensörü.....	46
Şekil 4.24 Düğüm III yerel izlemenin çıktı örneği	47
Şekil 4.25 X-CTU program kısayolu	48
Şekil 4.26 XBee gezgini kartı	48
Şekil 4.27 XCTU yardımcı yazılımının ekran görüntüsü	49
Şekil 4.28 Seri bağlantı noktalarını ayarlama	49
Şekil 4.29 XBee Cihazını Ekleme	50
Şekil 4.30 X-CTU yazılımının XBee cihazı eklendikten sonraki ekran görüntüsü	50
Şekil 4.31 XBee parametrelerinin değiştirilmesi ve kaydedilmesi	51
Şekil 4.32 Voltaj sensöründen veri okumak için örnek kod	54
Şekil 4.33 Düğüm-I için akış şeması.	55
Şekil 4.34 MQ2 sensöründen gaz ve dumanı hesaplamak için örnek kod	57
Şekil 4.35 Düğüm-II Akış şeması.	58
Şekil 4.36 Düğüm-III akış şeması.....	59

Şekil 4.37 Ana Arduino kartından Raspberry Pi'ye gönderilen veri formatı	60
Şekil 4.38 Ana düğüme ait programının akış şemasını göstermektedir.	61
Şekil 4.39 Raspberry Pi için Tasarlanmış GUI	62
Şekil 4.40 Duman ve gaz değerlerinin normal olduğunun bildirilmesi.	63
Şekil 4.41 Yangın algılanması ve dilsel ifade olarak bildirilmesi.	63
Şekil 4.42 Işık düzeyinin algılanması ve dilsel ifade olarak bildirilmesi.	63
Şekil 4.43 Yüksek seviyede duman veya yangın algılandığında evin ana gücünün otomatik olarak kapatılması.	64
Şekil 4.44 Raspberry Pi için tasarlanmış programın akış şeması.	66
Şekil 4.45 Uzak PC için Tasarlanmış GUI.....	67
Şekil 4.46 Sensör verilerinin gösterilmesi için kullanılan örnek kod.	67
Şekil 4.47 Klima butonunun butonlar bölümünden yapılandırılması.	68
Şekil 4.48 Uzak bilgisayardan kontrol programının akış şeması	69
Şekil 4.49 Thing Speak'de oluşturulan kanallar.....	71
Şekil 4.50 Thing Speak'de oluşturulan kanalda verilerin sunumu.	72
Şekil 4.51 Thing Speak'da oluşturulan araç durum kanalında sunulan veriler.	73
Şekil 4.52 Android Uygulama.....	74
Şekil 4.53 MQTT aracısını ayarlama ve yapılandırma süreci	75

TABLolar DİZİNİ

	<u>Sayfa</u>
Tablo 3.1 Ağdaki düğümleri yapılandırmak için kullanılan parametreler	51
Tablo 3.2 Sistemin Maliyeti.....	76

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

ABI	: Uygulama İkili Arayüzü
AC	: Alternatif Akım
ADC	: Analogdan dijitale dönüştürücü
API	: Uygulama Programlama Arayüzü
AVR	: Otomatik Voltaj Regülatörü
CEO	: Baş Yönetici
CMOS	: Tamamlayıcı Metal Oksit Yarıiletken
COM	: İletişim Portu
CRM	: Müşteri ilişkileri yönetimi
DC	: Doğru Akım
DHT	: Dijital Nem ve Sıcaklık Sensörü
DynDNS	: Dinamik Alan Sistemi
GUI	: Grafiksel Kullanıcı Ara yüzü
IaaS	: Altyapı Hizmeti
ICT	: Bilgi ve İletişim Teknolojisi
IEEE	: Elektrik ve Elektronik Mühendisleri Enstitüsü
IoE	: Her şeyin İnterneti
IOS	: Uluslararası Standardizasyon Kurumu
IoT	: Nesnelerin İnterneti
IP	: İnternet Protokolü
IR	: Kızılötesi Sensör
LCD	: Sıvı Kristal Ekran
LDR	: Işığa Bağlı Direnç
LED	: Işık Yayan Diyot
LPG	: Sıvılaştırılmış Petrol Gazı
M2M	: Makineden Makineye
MIT	: Massachusetts Teknoloji Enstitüsü
MQ2	: Metal Oksit Sensör
MQTT	: Mesaj Kuyruklama Telemetri Aktarımı
OASIS	: Yapılandırılmış Bilgi Standartlarını
P2M	: Kişiden Makineye
P2P	: Kişiden Kişiye
PaaS	: Servis Olarak Platform
PAN ID	: Kişisel Alan Ağı Tanımlayıcısı
PC	: Kişisel Bilgisayar
PCB	: Baskı Devre Kartı
PDA	: Kişisel Dijital Asistan
PHP	: Hiper Metin Ön İşlemcisi
PIR	: Pasif Kızılötesi Sensör
PPM	: Milyon Başına Parça
QoS	: Hizmet Kalitesi

RF	: Radyo Frekansı
RFID	: Radyo Frekansı Tanımlama
RISC	: Azaltılmış Komut Seti Bilgisayarı
RMS	: Kök Ortalama Kare
RX	: Almak
SaaS	: Hizmet olarak Yazılım
SCL	: Seri Saat
SDA	: Seri Veri
TFT	: İnce Film Transistör
TX	: İletme
USB	: Evrensel Seri Veriyolu
VPN	: Sanal Özel Ağ
WPAN	: Kablosuz Kişisel Alan Ağı

1. GİRİŞ

Hiç şüphe yok ki, insanlar her zaman ilginç bir yaşam oluşturmaya çalışır ve en iyisini inşa edip üretmek için tüm zor koşullara meydan okurlar; tüm bunlar modern teknolojinin hızlı ilerlemesine ve gelişmesine yol açmış ve bu da günlük hayatın daha iyi olmasına katkıda bulunmuştur. Son yıllarda bilgi teknolojileri, kontrol ve elektronik teknolojileri alanını takip etmek zor olmakta ve hatta bazen ilk defa duyulan ani ve muazzam gelişmeler görülmektedir (Vogels, 2020). İnsanların daha basit ve daha kolay bir yaşama kavuşmasını sağlayan hizmetler sunan uygulamalar oluşturmak için bu teknolojileri bir arada kullanmak mümkün hale gelmiştir. Uzun yıllardan beri Nesnelerin İnterneti (Internet of Things-IoT) dünya çapında kullanılmakta ve Bilgi ve İletişim Teknolojisi (ICT) alanındaki başlıca araştırma ve endüstriyel konulardan biri haline gelmiştir (Atzori vd., 2017). Nesnelerin İnternetinin vizyonu: (i) kablolu ve kablosuz bağlantı yoluyla birbirleriyle işbirliği yapabilen yeni hizmetler ya da uygulamalar oluşturmak, (ii) ortak hedeflere ulaşmak için bu hizmetleri bir araya getirmek, analiz etmek ve içgörü sağlamaktır.

Nesnelerin İnterneti'nin amacı, herhangi bir ağı, yolu ve herhangi bir hizmeti mükemmel bir şekilde kullanarak nesnelere herhangi bir kişi, herhangi bir şey ile, herhangi bir yerde ve zamandan bağımsız olarak bağlantılandırmaktır (Vermesan ve Friess, 2015). IoT, akıllı evler ve ofisler için akıllı uygulamalar, akıllı ulaşım sistemleri, daha akıllı hastaneler, daha akıllı işletmeler ve fabrikalar gibi birçok uygulama sağlamaktadır (Sen, 2011). Akıllı ev veya ev otomasyonu terimi, büyük faydaları nedeniyle yaygınlaşmaktadır, Ev otomasyon sistemi evleri uzaktan izlemeyi ve kontrol etmeyi amaçlamaktadır (Cyril Jose ve Malekian, 2016) .

1.1 Çalışmanın Amacı

Bu çalışmadaki temel amaç, Nesnelerin İnterneti ve iletişim teknolojilerini kullanarak düşük maliyetli bir ev otomasyon sistemi tasarlamaktır. Bu amaç çerçevesindeki alt amaçlar aşağıdaki şekildedir:

- Genel olarak kullanılan yerel izleme yerine Arduino, Raspberry Pi ve Zigbee kullanarak tasarlanan kablosuz sensör ağı sistemi ile Web ve Android temelli uygulamalar aracılığıyla dünyanın her yerinden gerçek zamanlı olarak evi uzaktan izleme ve kontrol etme sistemi geliştirilmesi.
- Yük voltajı, yük akımı, sıcaklık, gaz kaçağı ve evin aydınlatma koşulları ile ilgili verileri izleme.
- Ev iklim kontrolü vasıtasıyla, ev sıcaklığını ve nem seviyesini izleme ve evdeki sıcaklığı kontrol etmek için klimayı veya ısıtıcıyı çalıştırma.
- Akıllı şehir uygulamalarından biri olan bahçe ışıklarını kontrol etme.
- Evi kontrol etme ve evdeki her elemanı anormal koşullarda kapatma.
- Evi yangınlardan ve gaz kaçağından koruma.
- Ev otomasyon uygulamalarına yön gösterme.

1.2 Çalışmanın Problemi

Olağandışı durumlarda bazen evden hızlı bir şekilde ayrılmanız ve günler veya aylar sonra eve dönmeniz gerekebilir. Bu gibi durumlarda evdeki ışıkları, ısıtmayı, klimaları veya diğer ev aletlerini açık unutabilirsiniz. Bu gibi durumları da göz önünde bulundurarak bu çalışma aşağıdaki problemlere odaklanmaktadır:

Koruma:

- Evin içinde veya dışında olduğunuzda gaz sızıntısı, yangın çıkması ve evde duman varlığı olabilir.
- Bazen evin içindeki akım ve voltajda artış olabilir.

Enerji tüketimini azaltmak:

- Dış aydınlatma özellikle gündüzleri uzun süre açık kalabilir.
- Sıcaklıklar orta seviyede iken klimaların ve ısıtmanın uzun süre çalıştırılması.

Maliyeti azaltmak:

- En düşük maliyetle akıllı ev sistemi kurmak, ana kontrol ünitesi olarak bilgisayar yerine Raspberry Pi kullanma durumunu incelemek ve test etmek.
- Aydınlatma, klima ve ısıtmadan kaynaklanan maliyet gibi gereksiz enerji tüketimini azaltmak.

1.3 Çalışmanın Önemi

Çalışmada çok yönlü bir akıllı ev sistemi geliştirilmiştir. Ayrıca geliştirilen sistem hem web hem de android temelli bir uygulamayla kontrol edilebilmektedir. Geliştirilen sistem aşağıdaki ana avantajlara sahiptir.

- Mümkün olan en düşük maliyetle ev otomasyon sistemi tasarlamak ve inşa etmek.
- Koruma ve güvenliği arttırmak.
- Evin artan güvenlik ömrü.
- Artan Enerji Tasarrufu.
- Geliştirme ve daha fazla konfor.

2. LİTERATÜR İNCELEMESİ

Günümüzde, dünya çapında farklı tekniklerle tasarlanmış ve inşa edilmiş, farklı amaçlar için kullanılan birçok ev otomasyon sistemi bulunmaktadır. Soliman vd. (2013) nesnelerin interneti ve bulut bilişim destekli ev otomasyon sistemi önermişlerdir. Çalışmalarında iki Arduino UNO mikrodenetleyici kart ve ZigBee teknolojisi kullanmışlardır. İlk kart, sistemdeki tüm elemanlara (fan, klima, masa lambası) ve Ethernet bağlantısı ile internet üzerindeki veritabanı sunucusuna bağlanan merkezi alıcı görevi görür. İkinci kart ise, sistemdeki tüm sensörlere (sıcaklık, nem, ışık) bağlı merkezi vericidir. ZigBee teknolojisi ise, ZigBee sensörleri ile merkezi Arduino kartları arasındaki iletişim için kullanılır.

Sagar ve Kusuma (2015), IoT kullanan ev otomasyon sistemi önermişlerdir ve bu model Intel Galileo kartı ve hareket, gaz, sıcaklık ve LDR gibi farklı sensörlerden oluşmaktadır. Intel Galileo internete WiFi üzerinden bağlanır, sensörlerin verilerini okumaya başlar. Web sunucusuna gönderilen ve bulutta depolanan sensör verileri, herhangi bir zamanda analiz edilebilir. Önerilen modelde gaz kaçağı, sıcaklık ve hareket ev içinde takip edilebilir. Hareketi ve sıcaklık algılaması analiz için bulutta, depolanmaktadır. Sıcaklık belirtilen seviyeyi aşarsa, klima otomatik olarak açılmaktadır. Benzer şekilde, evde bir gaz sızıntısı olduğunda, alarmı çalmak için alarm tetiklenmektedir. Ayrıca evin dışında ışık algılandığında gerekli ışıklar otomatik olarak açılıp kapanmaktadır. Kullanıcı, web sunucusu aracılığıyla internet üzerinden elektrikli cihazları izleyebilir.

Pavithra ve Balakrishnan (2015), ev otomasyonu için IoT tabanlı izleme ve kontrol sistemi önermişlerdir. Sistem; PIR sensörü, IR sensörü ve yangın algılama sensörü ve raspberry Pi'den oluşur. Kızılötesi nesne algılama ünitesi olarak kullanılan kızılötesi sensör (IR), kızılötesi LED'ler kullanılarak evde uygulanabilmekte, ışık algılandığında tetiklenmektedir. Sensör tetiklendiğinde raspberry pi'ye bir sinyal göndermektedir. Raspberry pi, wi-fi konfigürasyonu ve IoT aracılığıyla ışığı açıp/kapatabilmektedir. PIR sensörü insanın var olduğunu tespit ederse fanlar açılıp/kapatılmaktadır. Son olarak herhangi bir yangın kazası meydana gelirse yangın

algılama sensörü tetiklenerek kamerada çekilen video ve görüntü ile birlikte cep telefonuna anında uyarı mesajı göndermektedir.

Bepery vd. (2019), tüm ev aletlerini web tabanlı arayüzle IoT konseptine dayalı olarak uzaktan veya yerel bir ağ üzerinden internet olmadan yerel olarak kontrol eden bir sistem önermişlerdir. Sistem ekonomiktir, kullanımı kolaydır ve kullanıcı isteğine göre ölçeklendirilebilir. Raspberry Pi, bir web arayüzü aracılığıyla kullanıcıdan komut almak ve bu komuta göre Arduino kartını kontrol etmek için kullanılır. Arduino kontrollü röle, kontrol sinyaline göre anahtarlanır.

Sistem aşağıdaki bileşenleri içerir:

- Raspberry Pi'nin internet ile bağlantısını sağlayan bir hotspot (kablosuz internet bağlantısı veya cep yönlendiricisi).
- Bir kontrol ünitesi, web sunucusu ve ayrıca bir SQL sunucusu olarak işlev gören bir Raspberry Pi.
- Dataplicity: Raspberry Pi'ye dünyanın her yerinden VPN, statik IP, DynDNS veya Port yönlendirme olmadan internet üzerinden uzaktan erişmeyi amaçlayan yeni bir hizmettir. "Dataplicity: Raspberry Pi'yi Uzaktan Kontrol için kullanır".
- Sistemi kontrol etmek için Web tabanlı Grafik kullanıcı arayüzü (GUI) ve röle anahtarını kontrol etmek için Arduino.

Koruma, güvenlik, kontrol, izleme vb. çeşitli hedeflere ulaşmayı amaçlayan ev otomasyon sistemi ile ilgili daha önce sunulmuş ve çeşitli teknolojilerin uygulandığı çok sayıda başka araştırma ve projeler de bulunmaktadır. Akıllı telefon, Arduino kartı ve Bluetooth teknolojisini kullanan ev otomasyon sistemleri güvenli ve düşük maliyetlidir. Piyare ve Tazil (2011) tarafından önerilen ev otomasyon sisteminde donanım mimarisi Arduino BT kartı ve bir cep telefonundan oluşur. Arduino BT kartı ile cep telefonu arasındaki iletişim kablosuz olarak Bluetooth teknolojisi kullanılarak yapılır.

Piyare (2013), nesnelerin interneti (IoT) teknolojisine dayalı bir ev kontrol ve izleme sistemi ortaya koymuştur. Sistem, gömülü mikro web sunucusu, kontrol cihazları, akıllı telefon ve bir yazılım uygulaması kullanılarak tasarlanmış ve uygulanmıştır. Sistemin mimarisi üç bölümden oluşur: Ev ortamı, ev ağ geçidi ve uzak ortam. Uzak ortam, yetkili kullanıcıların ev aletlerini Wi-Fi, 3G veya 4G ve android uygulamasını destekleyen bir akıllı telefon kullanarak uzaktan kontrol etmesine ve izlemesine olanak tanır. Ev ortamı, donanım arayüz modülünü ve ev ağ geçidini içerir. Ev ağ geçidinin işlevi internet, yönlendirici ve Arduino Ethernet sunucusu arasında veri çeviri hizmeti sağlamaktır ve ev ağ geçidi, Arduino Ethernet kalkanı kullanılarak oluşturulmuş bir mikro web sunucusudur (Piyare, 2013). Sen vd. (2015), ses tanıma tabanlı bir ev otomasyon sistemi önermiş ve uygulamıştır. Bu sistemin donanım mimarisi Arduino Uno ve akıllı telefonda oluşmaktadır. Akıllı telefon ve Arduino Uno arasındaki kablosuz iletişim, Bluetooth teknolojisi aracılığıyla yapılır.

Kumar ve Pati (2016), arduino ve raspberry pi tabanlı akıllı iletişim sistemi önermişlerdir. Akıllı iletişim sistemi, iki hoparlör, bir mikrofon, bir Raspberry Pi kamera, bir LCD ekran, 16 × 2 LCD ve sesli alarm gibi çeşitli bileşenlerden oluşur. Bu sistemde, kişinin fotoğrafı kapıda çekilir ve bilgi için ev sahibinin LCD ekranında görüntülenir ve ziyaretçi konuşmak için izin alabilir veya ev sahibine sesli mesaj bırakabilir. Ev sahibi kapıdaki kişiye gel, bekle, daha sonra gel ve toplantı devam ediyor gibi bir mesaj gönderebilir ve bunlar LCD ekranda görüntülenir. Sistem Python dilinde kodlanmıştır ve aynı zamanda bazı ev aletlerini de kontrol etmektedir.

Kasmi vd. (2016), IoT temelli bir ev otomasyon sistemi önermiştir. Farklı düğümler arasında ev içi iletişim için ZigBee ağını kullanmışlar ve kullanıcının evin cihazlarını internetten veya yerel olarak uzaktan kontrol edebilmesi için PHP uygulaması geliştirmişlerdir. Önerilen sistem, sıcaklığı, aydınlatmayı otomatik olarak kontrol etme ve yangın ve şüpheli izinsiz girişler olduğunda insanları bilgilendirme özelliklerine sahiptir. Ev, kullanıcılar tarafından yapılan tüm eylemleri kaydetmek için bulut platformuna bağlıdır ve ayrıca evin tüm cihazlarının durumunu gerçek zamanlı olarak izleyebilmektedir. Wadhvani vd. (2018) ise, algılama, izleme ve kontrol olmak üzere üç önemli bölümden oluşan ev otomasyon ve güvenlik sistemi önermişlerdir. Algılama, yangın sensörü, ivmeölçer, esnek sensör vb. sensörlerle

yapılır. İzleme görevi bulut platformu üzerinden yapılır. Kontrol ise Arduino Uno tarafından gerçekleştirilir. Tüm sensörler, cihazlar ve Wi-Fi modülü Arduino Uno ile bağlantılıdır. Cihazların durumu bulut platformuna yüklenmekte ve kullanıcı cihazların durumunu dizüstü bilgisayarından ve akıllı telefonundan da izleyebilmektedir. Günümüzde nesnelerin interneti'nin evrimiyle birlikte, ev sistemlerinin izlenmesi ve kontrolü çok daha kolay hale gelmiştir ve dünyanın herhangi bir yerinden manipüle edilebilir durumdadır. Bu tez çalışması, ZigBee ve IOT kullanarak ev otomasyon sisteminin izlenmesine ve kontrol edilmesine odaklanmaktadır.

ortak altyapıya dayalı olarak nesnelere birbirine bağlamayı ve birbirleriyle iletişim kurmayı amaçlamaktadır.

Basitçe ifade etmek gerekirse, IoT, internete bağlı her cihazdır. Cihazların her zaman ve her yerde mevcut olabilecek birçok bilgiye sahip olması ilkesi ve konsepti ile başlamıştır (Kraeling ve Brogioli, 2019). Örneğin, buzdolaplarının, termostatların, nemlendiricilerin, lambaların, ilk yardım setlerinin, tıbbi ekipmanların, otomobil parçalarının ve diğer şeylerin internete bağlı olduğunu varsayıldığında, hangi ilaç ve ekipmanın kullanım süresinin dolduğunu bulmak ve stoğun tükendiğini bilmek kolaydır ve herhangi bir zamanda neyin açılıp kapattıldığını bilirsiniz, böylece kayıplar öncekinden daha az olur. Nesnelere "İnterneti" kavramı genellikle aşağıdakileri ifade etmek için kullanılır: (i) Akıllı nesnelere, genişletilmiş İnternet teknolojileri aracılığıyla birbirine bağlayan yeni ortaya çıkan küresel ağ, (ii) Bunu gerçekleştirmeye katkıda bulunan gerekli teknolojiler seti (aktüatörler / sensörler, RFID'ler, tüm makineden makineye iletişim cihazları vb. dahil) ve (iii) yeni pazarlar ve iş fırsatları oluşturmak ve inşa etmek için bu teknolojilerden yararlanan hizmetler ve uygulamalar (Miorandi et al., 2012). Cisco araştırmalarına göre 2020'ye kadar 50 milyar, 2030'a kadar 500 milyar internete bağlı cihaz olması beklenmektedir. Bu tahminlerin İnternet teknolojisindeki veya cihazlarındaki hızlı gelişmelere bakmadığını ve bu sayıların günümüzde bilinen araştırmalara bağlı olduğunu unutmamak önemlidir (Cisco, 2016).

IoT dört farklı yaygın iletişim modeli kullanmaktadır: (i) Cihazdan Cihaza İletişim: Bu model, Zigbee, Bluetooth vb. gibi kablosuz bir ağ üzerinden sunucu kullanmadan doğrudan birbiriyle iletişim kuran iki cihaza sahiptir. (ii) Cihazdan Buluta İletişim: Bu modelde , trafiği kontrol etmek ve veri alışverişi yapmak için bağlantı, doğrudan IoT cihazı ile bulut internet hizmeti arasındadır. Bu yaklaşım, cihaz ve IP ağı arasında bağlanma sürecini oluşturmak ve ardından bulut hizmetine bağlanmak için Wi-Fi veya kablolu Ethernet gibi geleneksel iletişim mekanizmalarını kullanır. (iii) Cihazdan Ağ Geçidine Modeli: Bu modelde, yerel bir ağ geçidi cihazı üzerinde çalışan, cihaz ile bulut hizmeti arasında aracı olarak kullanılan ve veri çevirisi veya protokol ve güvenlik gibi bazı işlevler sağlayan bir uygulama yazılımı mevcuttur. (iv) Arka Uç Veri Paylaşım Modeli: Bu model, kullanıcıların bir bulut hizmetinden

akıllı nesne verilerini diğerkaynaklardan başkaverilerle grup halinde almasına ve analiz etmesine olanak tanıyan bir iletişim yapısını gösterir (Rose vd., 2015).

3.1.2 Nesnelerin İnterneti Uygulamaları

Nesnelerin interneti birçok uygulamaya sahiptir, bazıları toplum, çevre ve endüstriyel olmak üzere üç ana kategoriye ayırmışlardır. Bu sınıflandırma temel alınarak yapılan "Nesneler" kullanıldığı uygulama alanına bağlı olarak farklı şekillerde ayırt edilebilir. Endüstride IoT faaliyetleri, kamu hizmeti sektörünü, şirketler arasındaki ticari ve finansal işlemleri, üretim süreçlerini ve bankacılık güvenliğini vb. içerir. Çevre ile ilgili uygulamaları, insan vücudu izleme, enerji ve tarım yönetimi gibi tüm doğal kaynakların izlenmesi ve geliştirilmesi gibi faaliyetlere dayanmaktadır (Chaudary vd., 2016). Son olarak toplum ile ilgili uygulamaları, günlük yaşama yardımcı olmak için kullanılan cihazlar ve halka açık yerlerde kullanılan cihazlarla ilgilidir. Örneğin akıllı kampüsler, akıllı evler ve akıllı şehirler (Chaudary vd., 2016). Bunun dışında onu birkaç farklı kategoriye veya bölüme ayıran başka çalışmalar da vardır. Ancak çoğunlukla endüstriyel, ticari, tüketici ve altyapı şeklindedir (Rouse, 2018).

IoT, sayısız endüstri odaklı ve kullanıcıya özel IoT uygulamalarının geliştirme sürecini destekler ve kolaylaştırır. Ağlar ve cihazlar fiziksel bağlantı sağlarken, IoT uygulamaları güvenilir ve güçlü bir yöntemle cihazdan cihaza ve insandan cihaza etkileşimlere olanak tanır. IoT uygulamalarının, alınan ve doğru şekilde hareket eden verilerin zamanında hareket etmesini garanti etmesi gerekir. Örneğin, kamu hizmetleri ve ulaşım uygulamaları et, balık, süt ürünleri, meyveler ve taze ürünler gibi taşınan malların durumunu izler. Taşıma işlemi sırasında, saklama durumu sürekli olarak izlenir (örneğin, nem, sıcaklık ve şoklar) ve hasarı önlemek için uygun önlemler otomatik olarak alınır (Lee ve Lee, 2015). Daha öncede bahsedildiği gibi, nesnelerin internetinin çok sayıda türü ve uygulaması vardır. Bunlar, farklı kategorilere ayrılmıştır ve en popüler olanı akıllı evler, akıllı saat, akıllı bina, akıllı üretim, eğitim, ulaşım.

Akıllı evler birçok nedenden dolayı çok önemli bir uygulamadır, aydınlatma, ısıtma, havalandırma, kamera, TV ve buzdolabı gibi tüm cihazları ve diğer elektronik

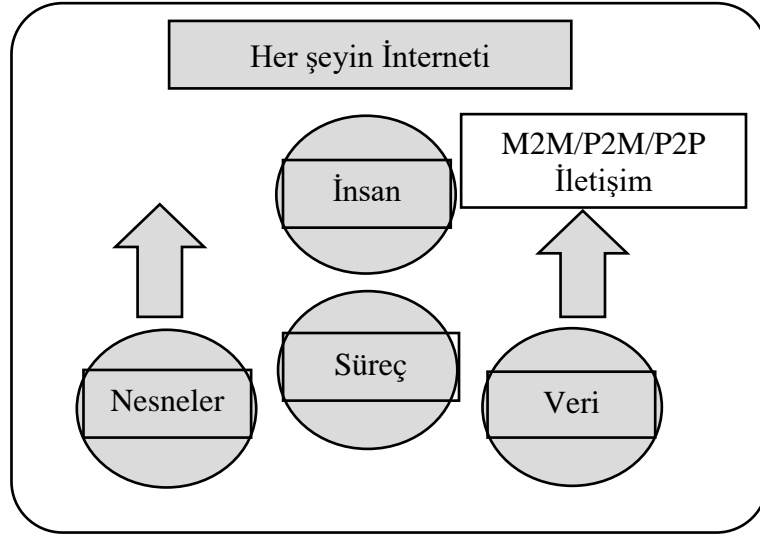
cihazları kontrol edebilir, aslında bu cihazlar IoT ile ilgilidir, böylece çok daha iyi bir yaşam kalitesi, konfor ve güvenlik sağlar (O'Brien, 2019).

Şu anda akıllı saatler yeni nesil saatlerdir ve buna olan talep tüm dünyada artmıştır. Bu teknoloji birçok büyük şirket için bir yatırım haline gelmiştir. Kullanıcı verilerini toplama, acil durumlarda analiz etme ve gönderme becerisine sahip olduğundan, kamu güvenliği için kullanılan ve kullanıcıların hayatını kolaylaştıran ve daha rahat hale getiren sensörlü ve yazılıma sahip giyilebilir cihazlardır (O'Brien, 2019; Rouse, 2018). Akıllı ulaşımında, nesnelerin interneti, sürücülerden, araçlara, parktan altyapıya ulaşım sistemlerinin tüm yönlerini kapsayacak şekilde (Mahmud vd., 2017) genişler ve çeşitli ulaşım sistemleri hakkında bilgi ve verinin iletiminin, yönetiminin ve kontrolünün entegrasyonuna yardımcı olur (Xie ve Wang, 2017). IoT ayrıca sokaklarda neler olup bittiğini ortaya koyarak trafik polislerine trafiği ve araçları takip etme ve izlemede yardımcı olur (Xie ve Wang, 2017).

Nesnelerin interneti ve akıllı sistemler, yeni ürünlerin üretim sürecini hızlandırabilir, ürün taleplerine otomatik olarak yanıt verebilir ve gerçek zamanlı üretim sürecini artırabilir (Ersue vd., 2015). Fabrika sahipleri, fabrikadaki sensörlerden izlenen verilere ve bilgilere dayanarak, denetçiye bazı bileşenlerin bitmek üzere olduğunu veya bazı cihazlarda bir sorun olacağını bildirmek için uyarılar göndererek, tüm fabrikada emek, zaman ve işten tasarruf elde edebilirler (O'Brien, 2019). Tarım alanında, tarlaları izleme, toprak içeriği, haşere istilası, yağış, rüzgar hızı, ışık, sıcaklık ve nem hakkında veri toplama (Meola, 2016) gibi IoT uygulamaları kullanılmaktadır. Ayrıca bitki büyümesi için gübre miktarını ve su kullanımını belirlemek ve böylelikle daha iyi mahsul alabilmek için kullanılan birçok IoT uygulaması da vardır (Zhang, 2015).

Nesnelerin internetinde Her Şeyin İnterneti (IoE) adı verilen yeni bir dönem bulunmaktadır. Nesnelerin interneti, nesneler / objeler (fiziksel nesneler) ile ilgili her şeydir, Her Şeyin İnterneti (IoE) ise verileri, süreçleri, nesneleri ve insanları içerir. Her Şeyin İnterneti, her şey ya da herkes için İnterneti gerektirmektedir. Şekil 3.2, insanı beslemek için endüstriyel ve ticari süreçleri kapsamı mümkün olan her şeyin İnternet şemasını, Nesneler, İnsanlar, Veriler ve Süreçler kapsamında

göstermektedir. IoE, verilerin makineden makineye (M2M), kişiden makineye (P2M) veya kişiden kişiye (P2P) aktarıldığı yerdir. Qualcomm CEO'su Steve Mollenkopf, IoE ve IoT'nin "aynı şey" olduğunu belirtmiş ve CISCO'dan Dane Evans, gelecekteki trendlerin "süreç, veriler, nesnelere ve insanlar arasındaki bağlantılar" ile ilgili olduğunu ve "nesnelere ilgili olmadığını" söylemiştir. Bu, Her Şeyin İnternetinin kalbidir (Srinivasan vd., 2019).



Şekil 3.2 Her şeyin İnterneti (Srinivasan vd., 2019)

3.2 Ev Otomasyonu

Ev otomasyonu, IoT temelli akıllı ev uygulamalarının bir parçasıdır. Ev otomasyonu, konfor, güvenlik, enerji tasarrufu sağlamak ve konutta yaşamı iyileştirmek için günlük aktivitelerde ve ev işlerinde kullanılır. Bu eylemler, ısıtma ve iklimlendirme sistemi için entegre kontrolü, kapı ve pencerelerin güvenlik kilitleri, aydınlatma ve cihazları içerir. Ev otomasyonu ayrıca yaşlılar ve sağlık bakımına ihtiyaç duyan kişiler için de daha iyi bir yaşam kalitesi sağlar (Jain vd., 2016).

Ev otomasyon sistemi, IoT uygulamaları için en önemli ve verimli olanı dikkate alır ve IoT uygulamalarında en üst sırada yer aldığı açıkça öne çıkmaktadır. IoT Analytics firması, 2015 yılında en yaygın Nesnelere İnterneti uygulamalarının ne olduğunu ölçtü ve bunun için üç farklı standart kullandı: İnsanlar Twitter'da neler konuşuyor, Google'da ne arıyor ve LinkedIn'de neler yazıyor. Ev otomasyonu %100 olarak en yüksek puanı aldı. Her ay "Akıllı Ev" terimini arayan yaklaşık 60.000 kişi

vardı. Akıllı ev için IoT Analytics firmasının veritabanı 256 şirket ve başlangıç içerir. Çoğu şirket, IoT'deki diğer uygulamalardan çok akıllı evle ilgilenmektedir (Lueth, 2015).

Akıllı ev, enerji tüketimini azaltmak için iç iklim ve çevre koşullarına yanıt veren bir yöntemle tüm cihazları soğutma, havalandırma, ısıtma, aydınlatma ve çalıştırma sistemlerini ayarlayabilen programlanabilir elektronik kontroller, sensörler içeren bir ev olarak tanımlanabilir. Akıllı evler, evin tüm yönlerini günlük olarak izlemek ve ev sahiplerine akıllı anlık bilgiler sağlamak için ev otomasyon teknolojilerini kullanır. Akıllı ev sistemi, bilgi alışverişi yapmak ve etkili kontrol uygulamak için sensör kontrol teknolojisi, iletişim teknolojisi ve çeşitli gömülü sistemlerle paylaşılan akıllı bir ev ağıdır (Kpae vd., 2018). Akıllı ev, iklim koşullarını, aydınlatmayı, ev aletlerini, güvenlik sistemlerini, multimedya ekipmanlarını ve diğer birçok işlevi kontrol etmeyi ve izlemeyi amaçlayan çok gelişmiş akıllı otomatik sistemleri içeren bir alandır (Malche ve Maheshwary, 2017).

Ev otomasyon sistemi, ortamı algılar ve kullanıcının kayıtlı olduğu bir araç üzerinden kullanıcıya uyarılar gönderebilir. Uyarı, çevreden toplanan verilerle ilgili bilgileri içerir. Bu bilgiler nem, sıcaklık, ışık yoğunluğu, ortamdaki farklı gazların seviyesi vb. içerebilir. Uyarı, düzenli olarak ve önceden ayarlanmış bir zaman diliminde kullanıcıya gönderilebilir. Ayrıca, kısa mesaj olarak e-posta yoluyla veya sosyal medya aracılığıyla uyarı gönderilebilir (Malche ve Maheshwary, 2017). Ev otomasyon sistemleri, cihazların çalışmasını uyarlayan süreç yoluyla enerji verimliliğini sağlar. Çoğu akıllı ev, konut sakinlerinin ev aletlerini uzaktan kontrol etmesine olanak tanıyan merkezi bir iletişim cihazı içerir (Lobaccaro vd., 2016).

Akıllı ev aşağıdaki temel özelliklere sahiptir:

- **Otomasyon:** Otomatik işlevleri uygulama veya otomatik cihazları barındırma yeteneği;
- **Çok işlevlilik:** Çeşitli görevleri yerine getirme veya farklı sonuçlar üretme yeteneği;

- **Uyarlanabilirlik:** Kullanıcı ihtiyaçlarını öğrenebilme, tahmin edebilme ve tatmin edebilme yeteneği;
- **Etkileşim:** Kullanıcıların birbirleriyle etkileşim kurmasına izin verme yeteneği;
- **Verimlilik:** Görevleri rahat bir şekilde gerçekleştirme yeteneği. Bu yetenek zamandan ve maliyetten tasarruf sağlar.

Nesnelerin internetine dayalı ev otomasyon sisteminin temel amacı, insan çabası ve enerji tüketiminde tasarruf sağlamaktır. Ev Otomasyon sistemi, çeşitli ev aletlerini insan müdahalesi olmadan uzaktan çalıştırmak ve kontrol etmek için kontrol sistemi teknolojilerini kullanma sürecidir (Radhika ve Menaka, 2016).

Ev otomasyon sistemlerinin üç türü vardır: i) Tek cihazdan kontrol, ii) Dağıtılmış kontrol sistemleri ve iii) Merkezi olarak kontrol edilen sistemler. Bireysel cihazları kontrol etme süreci genellikle tek bir görevde veya bir cihazda gerçekleşir, örneğin doluluk sensörleri, hareket dedektörleri, programlanabilir termostatlar, zamanlayıcılar ve aydınlatma kontrollerini kontrol eder. Dağıtılmış kontrol sistemleri, merkezi bir kontrol olmaksızın elektrik kabloları aracılığıyla tek tek cihazlar arasında dahili olarak birbirleriyle iletişim sürecini gerçekleştirir. Merkezi olarak kontrol edilen iletişim sistemleri, merkezi cihaz ile çevresel sensörler ve diğer cihazlar arasında sinyaller gönderir. Bu sistem, "akıllı" cihazları ve "aptal" cihazları kontrol edebildiği bir avantaja sahiptir, ancak kontrol ünitesi arızalandığında, tüm sistem bozulacaktır (Maksimovic ve Vujovic, 2015).

3.2.1 Ev Otomasyon Sistemlerinin Avantajları

Son zamanlarda, ZigBee, Wi-Fi gibi kablosuz sistemler ev ağlarında daha popüler hale gelmiştir. Ayrıca ev ve bina otomasyon sistemlerinde, kablosuz teknolojilerin kullanımı çeşitli avantajlar sağlar:

- Daha düşük kurulum maliyetleri: Hiçbir kablolama gerektirmediğinden kurulum maliyetlerini düşürür. Kabloyu duvarlara bağlama işlemi pahalıdır.
- Sistem ölçeklenebilirliği ve kolay genişletme: Bir kablosuz ağ kurmak, özellikle yeni ve değişen gereksinimler nedeniyle ağı genişletmeniz ve uzatmanız

gerektiğinde çok kullanışlıdır, bu, kablolu ağlar oluşturmanın aksine kablosuz ağların oluşturulmasını önemli bir yatırım haline getirir.

- Estetik faydalar: Bu özellik estetik gereksinimlerin tamamlanmasına yardımcı olur.
- Taşınabilir cihazların entegrasyonu: Kablosuz ağların mevcudiyetinde, akıllı telefonlar ve PDA'lar veya diğer taşınabilir cihazlar, her zaman ve her yerde otomasyon sistemine bağlanabilir. (Sagar ve Kusuma, 2015).

3.3 MQTT Protokolü

MQTT (Mesaj Kuyruklama Queuing Telemetry Aktarımı - Message Queuing Telemetry Transport) yayınlama/abone olma, basit ve hafif mesajlaşma protokolüdür ve nesnelerin interneti bağlantısı için bir ISO ve OASIS standardıdır. MQTT, Dr. Andy Stanford-Clark ve Arlen Nipper tarafından 1999'da icat edilen bir itme protokolüdür (MQTT.org, 2018). MQTT protokolü, ağ bant genişliğini ve cihaz kaynağı gereksinimlerini en aza indirmek amacıyla kablosuz ağlar için tasarlanmıştır ve ayrıca büyük ölçüde teslimat ve güvenilirlik güvencesi sağlamaya çalışır. MQTT, veri dağıtımının güvencesi ile ilgili üç hizmet kalitesi seviyesine sahiptir.

1. QoS0 (En fazla bir kez): Burada mesaj en fazla bir kez gönderilir ve bir mesajın teslimini garanti etmez.
2. QoS1 (En az bir kez): Bu seviyede mesaj en az bir kez gönderilir ve bir mesajı birden fazla teslim etmek mümkündür.
3. QoS2 (Tam olarak bir kez): Bu hizmet kalitesi, bir mesajın 4 yönlü sinyal alışverişi kullanılarak tam olarak bir kez gönderildiği için teslim edilmesini garanti eder.

Şekil 3.3, MQTT mimarisini göstermektedir. QoS seviyesinin seçimi sisteme bağlıdır, eğer bir sistem sürekli veri dağıtımına ihtiyaç duyuyorsa, zaman gecikmesi olsa bile veriyi iletmek için seçim QoS2'dir. MQTT'nin mimarisi, sunucu (broker) ve istemci (client) olmak üzere iki ana bileşenden oluşur. Sunucu esas olarak yayıncıdan

Ulusal Standartlar ve Teknoloji Enstitüsü, bulut bilişimin tüm genel yönlerini kapsayan bir tanım sağlamıştır: "Minimum yönetim çabası veya hizmet sağlayıcısı etkileşimi ile hızla tedarik edilebilen ve piyasaya sürülebilen, yapılandırılabilir bilgi işlem kaynaklarının (ör. Ağlar, sunucular, depolama, uygulamalar ve hizmetler) paylaşım havuzuna her yerde bulunan, kullanışlı, isteğe bağlı ağ erişimini mümkün kılan bir model" . Bu tanımdan, bulut bilişimin aşağıdaki temel özelliklere sahip olduğunu görebiliriz: (i) Geniş ağ erişimi, (ii) Kaynak havuzu (iii) İsteğe bağlı self servis, (iv) Hızlı esneklik, (v) Ölçülen Hizmet veya kullandıkça öde (Raghavendran vd., 2016). Bu, son kullanıcıların büyük bulut kaynaklarını hızlı ve kolay bir şekilde elde etme ve bu kaynakların bir kısmını kullanma yeteneğine sahip olduğu anlamına gelir.

Teknoloji şirketleri bulutu sermaye verimliliğinde önemli bir artış olarak görmektedir. Çünkü bu, daha düşük maliyetle ve eskisinden daha basit yeni bir hizmet başlatma imkânı sağlamaktadır. Bulut bilişim, son kullanıcının hizmetleri sunan sistemi ve fiziksel konumunu bilmesine gerek kalmadan veri erişimi, yazılım, hesaplama ve depolama hizmetleri sağlar. Bulut bilişim, şirketlerin, kuruluşların ve bireylerin yazılım ve uygulamaları yükleme, güncelleme ve bakım yüklerinden ve masraflarından kurtulmalarına yardımcı olur (Prasanth, 2012). Şirketlerin ve kuruluşların performansını artırmaya katkıda bulunan en önemli gelişen teknolojilerden biridir ve gelişmiş performansı zorlayan çeşitli teknolojilere sahiptir.

Bilgisayar kullanıcılarının sayısındaki muazzam artış nedeniyle, çoğu şirketin genel performans yönetimini iyileştirebilmeleri amacıyla verilerini etkili bir şekilde depolamayı amaçladıkları için, veri depolamanın nasıl artırılacağını düşünmek önemli hale geldi (Tadapaneni, 2020). Şirketler ve devletler, veri depolama, bilgi işlem, CRM ve veritabanları gibi altyapı ve uygulamalardan çeşitli ihtiyaçları karşılamak için bulut bilişim hizmetlerini kullanmaktadır. Bulut, hızlı uygulama teknolojileri isteyen kişiler için mükemmel bir çözümdür. Bulut bilişim, internette uygulama, yazılım ve donanım sağladığını gösteren yapılandırılabilir, dağıtılmış, paralel, sanal ve esnek sistemlerdir. Bulut bilişim hizmetleri yapılandırılabilir ve müşteriler, hizmetlerin ve kaynakların kullanımına göre ücret öder (Raghavendran vd., 2016).

Bulut hizmetleri sađlayan ve bulut biliřim uygulamalarını barındırmak için dünyanın farklı yerlerinde yeni veri merkezleri oluřturmaya bařlayan birçok řirket vardır. Bu řirketler arasında Sun Microsystems, Google, Microsoft, IBM, Amazon ve SaaS, PaaS ve IaaS bulut hizmetleri sađlayan diđerleri mükemmel alıřmaktadır.

3.4.1 Bulut Bilgisayar Hizmetleri

3.4.1.1 Hizmet olarak yazılım (SaaS)

SaaS aracılıđıyla, bir yazılım internette hizmet olarak (SaaS) dađıtılır. Bulut biliřimin en yaygın modelidir. Sađlayıcılar, donanım kararlılıđından uygulama alıřmasına kadar her řeyi dzenler. Bu modelde mřteriler hibir řeyden sorumlu deđildir; sadece grevlerini bitirmek için programları kullanırlar.Sađlayıcı tarafından mřterilere abonelik yoluyla, talep zerine hizmet olarak, kullanıma gre cretlendirilerek veya kullanıcı listesi satıřları veya reklam gibi kullanıcı yerine bařka kaynaklardan gelir elde etme imkanı varsa cretsiz olarak sunulur. Bu tr uygulamalar aracılıđıyla mřteriler verileri depolayabilir, analiz edebilir ve projeler zerinde iřbirliđi yapabilir (Raghavendran vd., 2016).

3.4.1.2 Hizmet olarak platform (PaaS)

PaaS sayesinde, altyapı veya yazılım ve bakım satın almayı dřnmeden web uygulamaları kolayca ve hızlı bir řekilde tasarlanabilir ve oluřturulabilir. SaaS ve PaaS arasındaki en byk fark, SaaS, web zerinden sunulan bir yazılımdır, PaaS, web zerinden sunulan yazılım oluřturmak için bir platformdur. Burada, geliřtiricilerin uygulamaları oluřturması, test etmesi, zelleřtirmesi ve yayınlamasını sađlayan aralar ve yazılım bileřenlerini ieren platform sađlanmıřtır. PaaS sađlayıcıları iřletim sistemi gncellemelerini, veritabanlarını, sunucuları, gvenlik onarımını ve yedeklemeleri ynetir. te yandan mřteriler, iřletim sistemi bakımı ve altyapısı ile ilgilenmeden veri ve uygulama geliřtirmeye odaklanır (Sakovich, 2018).

3.4.1.3 Hizmet olarak altyapı (IaaS)

Hizmet Olarak Altyapı (IaaS), işletim sistemleri, sunucular, ağ ve depolamayı talep üzerine hizmet olarak sağlama sürecidir ve bulut bilişim altyapısını dağıtmak için kullanılır. Müşteriler, yazılım, veri merkezi alanı, sunucular ve ağ gibi kaynakları talep üzerine eksiksiz bir dış kaynak hizmeti olarak satın alırlar (Prasanth, 2012). Bulut altyapısı hizmet sağlayıcıları, müşterilerine genişletilebilen ağlar, depolama ve sunucular gibi bulut bilişim araçları sunmak için sanallaştırma teknolojisini kullanır. Müşterilerin kişisel cihazlar satın almaları ve bileşenlerini yönetmeleri gerekmez. Yalnızca, uygulamalarını ve platformlarını, gerçek veri merkezinde aynı mevcut teknolojileri sunan sağlayıcının sanal makinelerine yerleştirebilirler. IaaS'de müşteriler, yazılım, uygulama, işletim sistemleri ve çalışma zamanının kurulmasından ve bakım sürecinin yanı sıra güvenlik verilerinden de sorumludur (Sakovich, 2018).

3.5 Zigbee Teknolojisi

Zigbee iletişimi (Şekil 3.4), kablosuz kişisel alan ağları (WPAN'lar) için IEEE 802.15.4 standardı üzerine inşa edilmiş kontrol ve sensör ağlarıyla başa çıkmak için özel olarak tasarlanmıştır ve Zigbee Alliance adlı bir grup şirket tarafından sunulmuştur.



Şekil 3.4 Zigbee Teknolojisi (Agarwal, 2014).

Zigbee ağı, düşük veri hızları, düşük güç tüketimi, düşük maliyet, güvenlik ve güvenilirlik gerektiren, 10-100 metreyi kapsayan menzil içinde uygulamaları kontrol etmek ve izlemek için yaygın olarak kullanılmaktadır (Gill vd., 2009). Zigbee'nin

WPAN'ları 868 MHz, 915 MHz ve 2,4 GHz frekanslarında çalışır. ZigBee standardı 250 kbps veri hızı sağlar ve sensörler ile kontrolörler arasında periyodik ve ara iki yönlü veri aktarımı için en uygundur (Kumar ve Mane, 2016). Zigbee, yöneticiden bağımlıya veya yöneticiden yöneticiye iletişim için çeşitli ağ yapılandırmalarını destekler. Ek olarak, çeşitli modlarda çalıştırılabilir, bunun sonucunda pil gücünden tasarruf sağlanır. Zigbee ağları, yönlendiriciler kullanılarak genişletilebilir ve daha geniş bir alan ağı oluşturmak için birçok düğümün birbirine bağlanmasına izin verir. Bunun yanı sıra, Zigbee ittifakı (Şekil 3.5), 1997 yılında o dönemde sekiz destekçi şirket tarafından kurulmuştur ve bu ittifak, çoğu kendi pazarlarında lider olan 200'den fazla üye şirkete yükselmiştir (Alliance, 2002).



Şekil 3.5 Zigbee İttifakı (Kinsella, 2019).

3.5.2 Zigbee Mimarisi

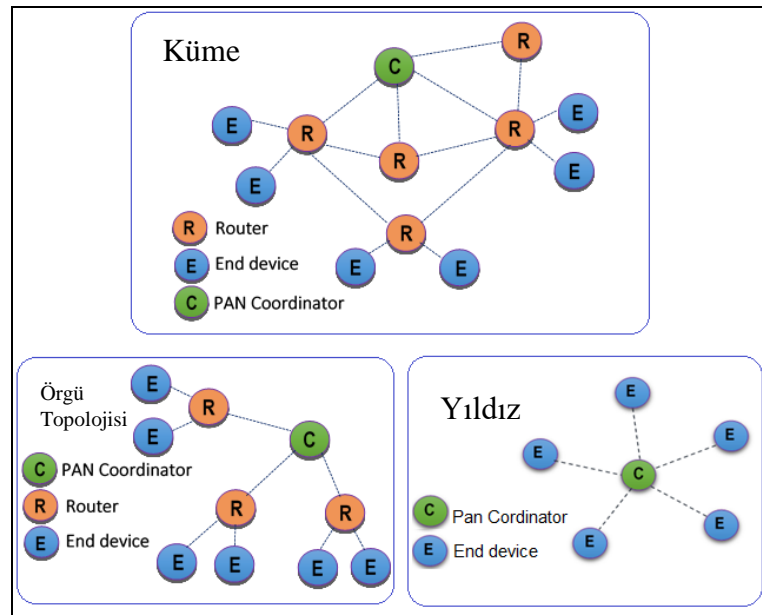
Zigbee sistem yapısı; Zigbee koordinatörü, yönlendirici ve uç cihazı gibi üç farklı türde cihaz içerir. Herhangi bir Zigbee ağı, ağın oluşturulması ve sürdürülmesinden sorumlu en az bir koordinatörden oluşmalıdır. Verileri iletme ve alma sırasında bilgileri depolamak gibi ağın işlemlerini koordine etmek ve kontrol etmek için bir yönetici olarak hareket eder. Ağdaki tüm elektronik cihazlar (yani lambalar, televizyon, klima vb.) koordinatör tarafından yönetilen bir Zigbee cihazıdır. Zigbee yönlendiricileri, verilerin farklı düğüm cihazları arasında geçişine izin veren köprü cihazları görevi görür. Uç cihazlar, orijinal düğümlerle iletişim kurmak için sınırlı işlevler gerçekleştirir (Elprocus, 2013).

3.5.3 Veri Transfer Modları

Zigbee, veri aktarımı için iki mod kullanır: İşaretçi modu ve İşaretçisiz mod. İşaretçi modunda, gelen verilerin aktif durumu koordinatörler ve yönlendiriciler tarafından sürekli olarak izlenir. Bu güç tüketimini artırır. Bu modda, yönlendiriciler ve koordinatörler sürekli çalışır. Çünkü herhangi bir zamanda herhangi bir düğüm uyanabilir ve iletişim kurabilir. Ancak, bu mod daha fazla güce ihtiyaç duyar ancak toplam güç tüketimi düşüktür. Çünkü ağdaki çoğu cihaz uzun süre uyku durumunda kalır. İşaretçisiz modda, koordinatörler ve yönlendiriciler, uç cihazlardan veri iletişimi olmadığında uyku durumuna geçer. Koordinatör periyodik olarak uyanır ve işaretçileri ağdaki yönlendiricilere ve cihazlara iletir. Bu işaretçilerde, ağlar zaman dilimleri için çalışırlar. Yani ihtiyaç duyulan iletişim daha az görev döngüsü ve daha uzun pil kullanımı gerektirdiğinde kullanılırlar (Kumar ve Mane, 2016).

3.5.4 Zigbee Ağ Topolojileri

Üç tür Zigbee ağ topolojisi vardır: yıldız topolojisi, örgü topolojisi ve küme ağacı topolojisi. Tüm bu topolojiler bir veya daha fazla koordinatörden oluşur. Şekil 3.6, Zigbee ağ topolojisi türlerini göstermektedir.



Şekil 3.6 Zigbee Topolojilerinin Türleri (Kumar ve Mane, 2016).

Bir yıldız ağ topolojisi, bir koordinatör ve birkaç uç cihazdan oluşur. Cihazların tümü tek koordinatör düğüme bağlanır ve uç cihazlar arasındaki tüm iletişim bu koordinatör aracılığıyla gerçekleşir. Bu topoloji basit ve dağıtımı kolaydır. Merkezi denetleyiciyle uç nokta cihazları arasında iletişim kurması gereken endüstrilerde kullanılabilir.

Örgü ve küme ağacı topolojilerinde, Zigbee ağı, koordinatörün onlara bakmaktan ve izlemekten sorumlu olduğu birçok yönlendirici ekleyerek genişletilebilir. Bu yapılardaki herhangi bir cihaz, verilerin fazlalığını sağlamak için herhangi bir komşu düğüm ile iletişim kurabilir. Bu topolojilerde, herhangi bir düğüm arızalandığında bilgiler otomatik olarak diğer cihaza yönlendirilir (Mahmood ve Abbas, 2017).

3.5.5 Zigbee Teknoloji Uygulamaları

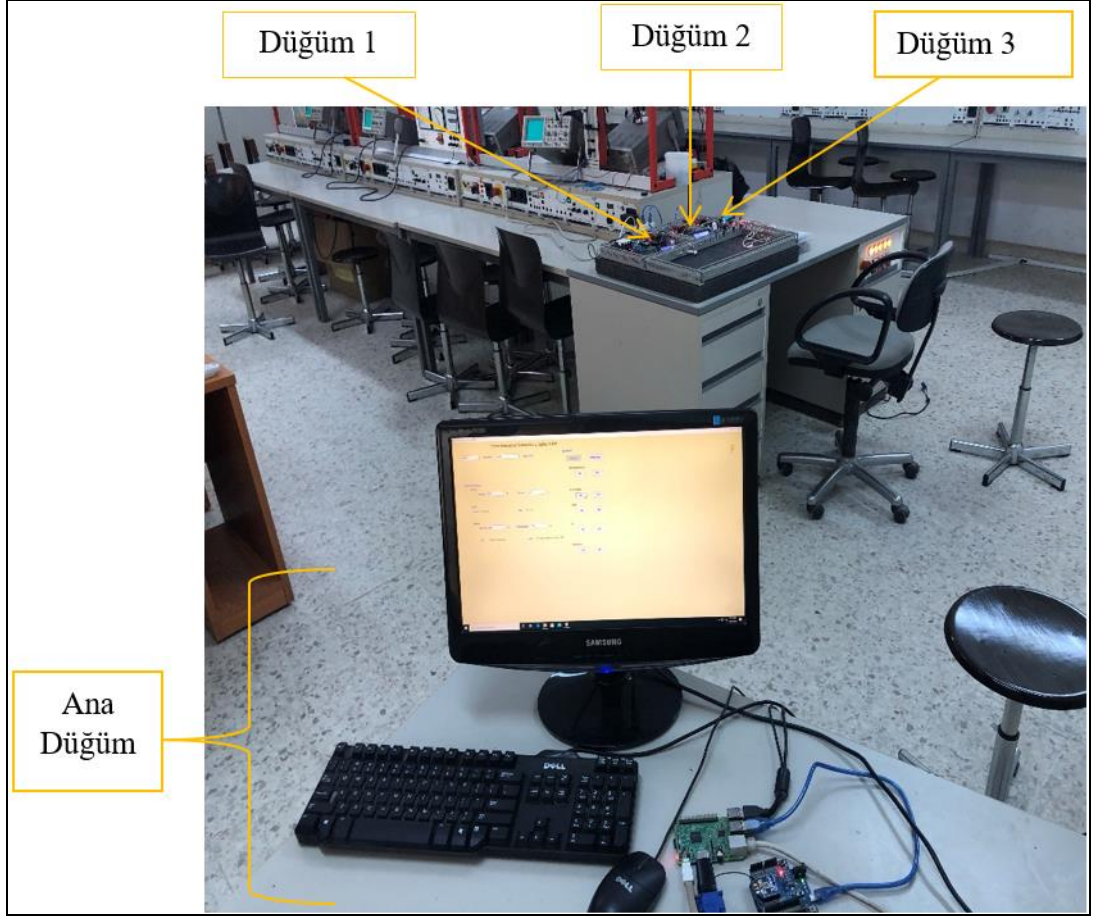
Zigbee ittifakının hedeflediği pek çok uygulama arasında "ticari, endüstriyel, tüketici ve hükümet pazarları" bulunmaktadır. Zigbee ev otomasyon sistemi, ısıtma ve iklimlendirme sistemi kontrolü, aydınlatma sistemi kontrolü, gözetim, cihaz kontrolü, güvenlik ekipmanı operasyon kontrolü gibi ev aletlerini izlemek ve uzaktan kontrol etmek için tamamen uygundur (Elprocus, 2013).

4. UYGULAMA VE DENEYSSEL SONUÇLAR

Herhangi bir mikrodenetleyici kontrollü sistemi uygulamaya koymak için donanımın belirli işlemleri yapması gerekir. Bunun için de sistem donanım bileşeni ve yazılım kodları kullanılarak yapılandırılmalıdır. Bu nedenle çalışma kapsamında geliştirilen sistem iki ana bölümde (donanım ve yazılım) incelenmiştir.

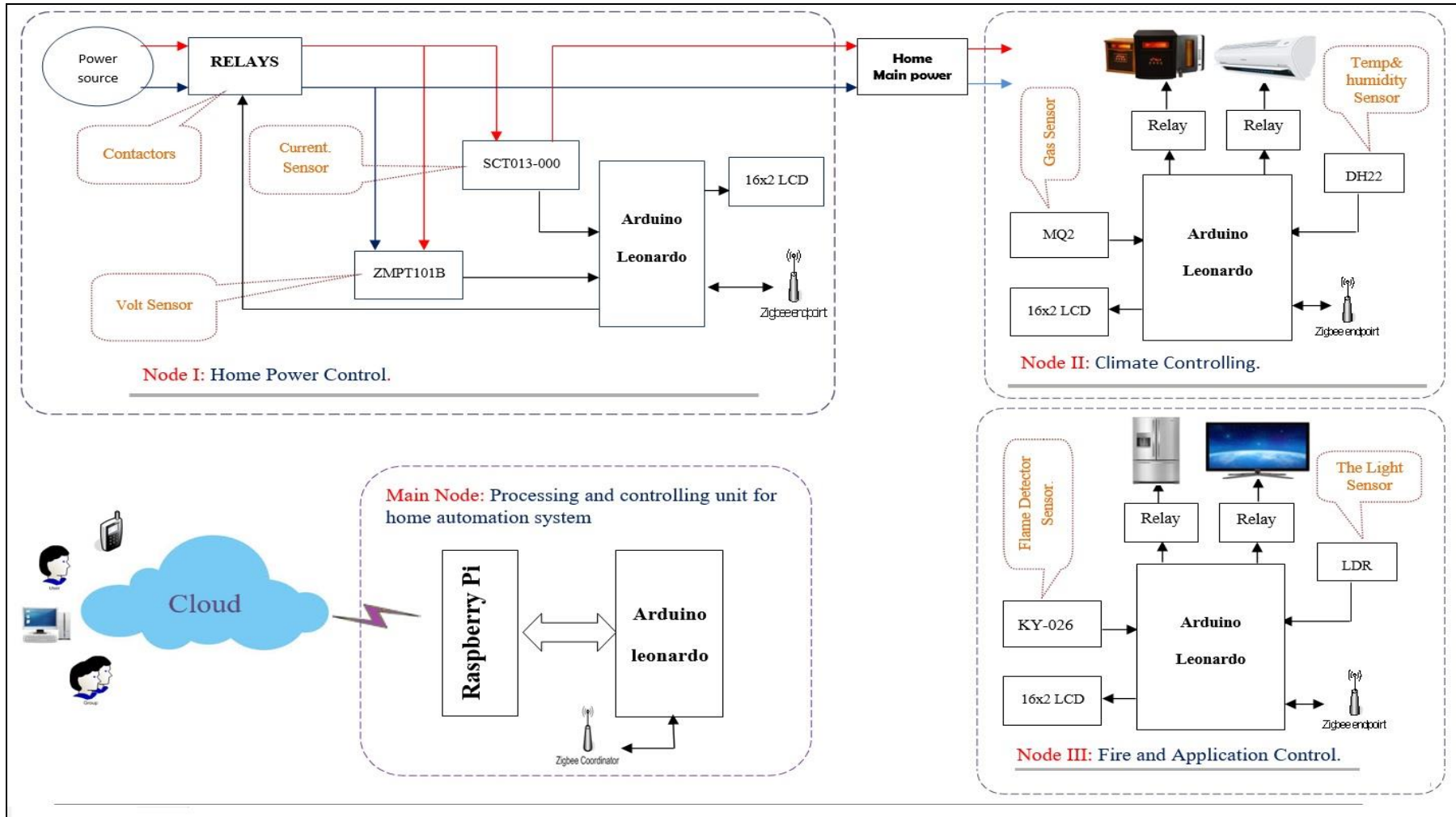
Sistem beş bölümden oluşmaktadır (Düğüm I, Düğüm II, Düğüm III, Ana Düğüm ve Raspberry Pi) ve her parçanın yapması gereken bir grup görev vardır. Geliştirilen sistemin her parçası, ATmega32u4 düşük güçlü CMOS 8-bit mikrodenetleyici temelli Arduino Leonardo kartını kullanmaktadır ve yerel bir ana birim olarak AVR geliştirilmiş RISC mimarisine ve ZigBee kablosuz iletişim sistemine dayanan XBee modülüne bağlıdır.

Ev otomasyon sistemi, evdeki ana gücü istenilen zamanda kapatacak şekilde uyarlanabilirliğe sahiptir ve ev voltajını, akımı, sıcaklığı, nemi, gaz kaçağını ve yangın alarm sistemini izlemektedir. Ayrıca, her ev aktivitesi üç likit kristal ekran (LCD) aracılığıyla gösterilir ve ölçülen değer eşzamanlı olarak daha önceden oluşturulan "Thingspeak.com" bulut alanına yüklenir. Şekil 4.1, dört düğümden oluşan ev kontrol ve izleme sisteminin tümünü göstermektedir. Bu şekilde de görüldüğü gibi bu düğümler pano üzerinde yan yana kurulmuştur.



Şekil 4.1 Tasarlanmış sistemin genel görünümü

Bu tasarım, Trablus'un Elektronik Teknolojisi kolejinde güç elektroniği laboratuvarında gerçekleştirilmiştir. Daha önce de belirtildiği gibi sistem dört bölümden oluşmaktadır: (i) Ana düğüm, (ii) Düğüm I, (iii) Düğüm II ve (iv) Düğüm III. Bu düğümlerin ait ayrıntılı tasarım mimarileri Şekil 4.2'de gösterilmektedir. Ana Düğüm, ev otomasyon sistemi için işlem ve kontrol birimi olarak kullanılır; Düğüm I, ev giriş akımını ve gerilimini izlemek ve kontrol etmek için kullanılır; Düğüm II, ev nemi, gaz kaçağı durumu ve sıcaklığı kontrol etmek için kullanılır, Düğüm III ise ev bahçe aydınlatması, yangın alarmı ve tv ve buzdolabını açma / kapamayı izleme ve kontrol için kullanılır. Bu düğümlerin donanımsal ve yazılımsal ayrıntıları aşağıdaki başlıklarda sunulmaktadır.



Şekil 4.2 Önerilen modelin genel tasarımı

4.2 Ana Dügüm

Bu düğüm, diğer düğümlerden gelen verilerin toplanması, sistemin uzaktan izlenebilmesi için sistem parametrelerinin yayınlanması, uzak kullanıcılardan gönderilen farklı uzak komutların alınması ve yürütülmesi için sistemi buluta bağlamaktan sorumludur. Şekil 4.3 ana düğüm için tasarlanmış ortamı göstermektedir. Bu düğüm, diğer üç XBee düğümüne kablosuz komutlar gönderen veya kablosuz veri alan XBee koordinatörü ve Raspberry Pi ile bağlantılıdır. Bu düğüm Bağlan, Al, Aktar, Oku ve Gönder görevleri için tasarlanmıştır.

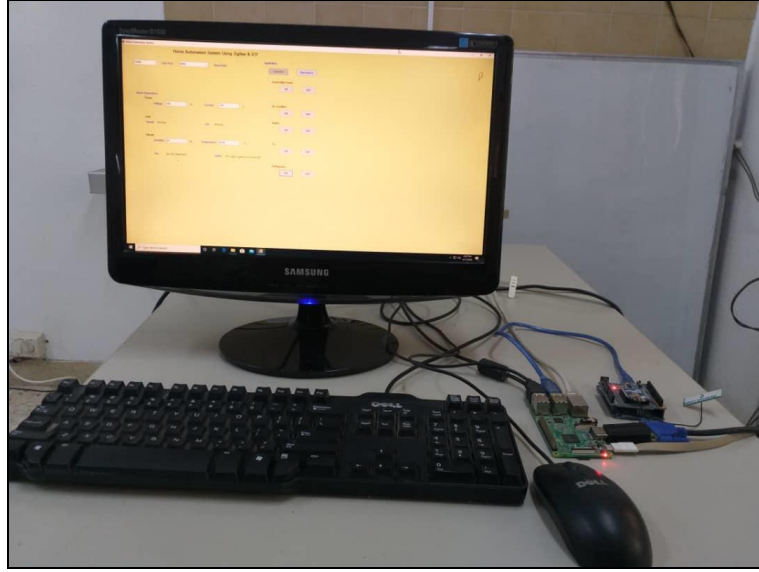
Bağlan: Bu görev, XBee kablosuz ağını kurmaktan, kurulan ağdaki diğer üç düğümü birleştirmekten ve aralarındaki bağlantıyı sağlamaktan (ana düğüm ve diğer üç düğüm) sorumludur.

Al: Oluşturulan verileri diğer üç düğümden XBee ağı üzerinden kablosuz olarak alır.

Aktar: Alınan verileri doğrudan USB COM portu üzerinden bağlanan Raspberry Pi'ye aktarır.

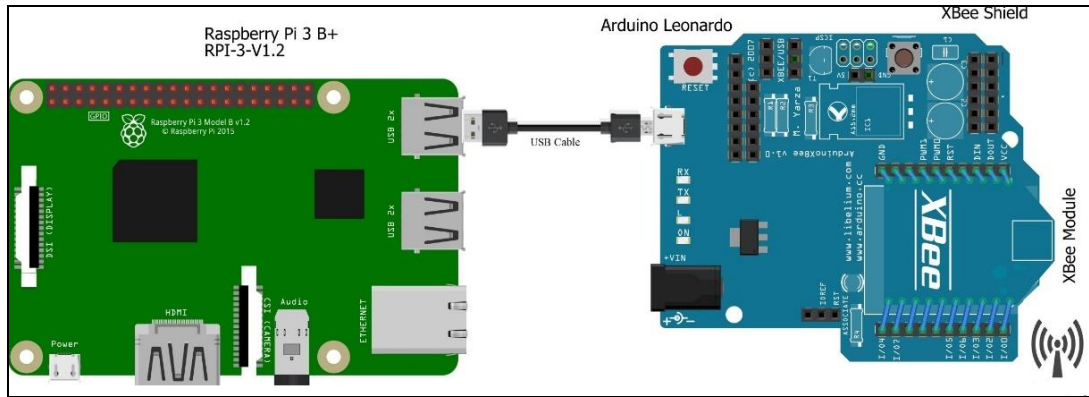
Oku: Doğrudan USB COM bağlantı noktası üzerinden bağlanan Raspberry Pi'den herhangi bir sistem veya kullanıcı komutu gelip gelmediğini okur.

Gönder: Doğrudan USB COM portu üzerinden bağlanan Raspberry Pi'den (ON/OFF komutları) gelen sistem veya kullanıcı komutlarını, ev aletlerini kontrol etmekten sorumlu diğer Düğümlere gönderir.



Şekil 4.3 Ana düğümün genel tasarımı ve görünümü

Şekil 4.4, ana düğümün elektronik tasarımını göstermektedir. Bu düğüm, Raspberry pi, Arduino Leonardo kartı ve XBee modülünü içerir. Raspberry pi, Arduino Leonardo kartına USB kablo ile bağlanır. Arduino ve Xbee modülü bağlantısı ise Arduino Xbee Shield ile gerçekleştirilir. Her bir modül aşağıdaki başlıklarda biraz daha ayrıntılı açıklanmaktadır.



Şekil 4.4 Ana düğümün elektronik tasarımı.

4.2.2 XBee Modülü (Koordinatör)

XBee koordinatörü, ZigBee ağını başlatmak ve diğer XBee düğümleri ile koordinatör arasında veri alışverişi için düğümlerin izinlerini yönetmek için kullanılmıştır. XBee modülünün bir örneği Şekil 4.5'te gösterilmektedir. Koordinatör, diğer düğümlerden kablosuz olarak alınan verilerle ilgilenir. Düğümlerden gelen verileri Arduino

Leonardo kartı üzerinden seri bağlantı yoluyla Raspberry Pi'ye ya da Raspberry Pi'den gelen verileri Arduino Leonardo kartı üzerinden diğer düğümlere aktarır. XBee modülü, X-CTU uygulama programı kullanılarak koordinatör veya son cihaz olarak çalışacak şekilde yapılandırılabilir. Bu ilerleyen bölümlerde ayrıntılı açıklanmaktadır.



Şekil 4.5 XBee modülü

4.2.3 Arduino Leonardo kartı

Arduino Leonardo kartı, ATmega32u4 tabanlı bir mikro denetleyici kartıdır dijital giriş / çıkış olarak 20 pini vardır (bunlardan 7 pini PWM çıkış olarak ve 12 pini analog giriş olarak kullanılabilir), 16 MHz kristal osilatör, mikro USB bağlantısı, güç girişi, ICSP başlığı ve bir sıfırlama düğmesi vardır. Mikro denetleyiciyi desteklemek için gereken her şeyi içerir (*Arduino Leonardo with Headers:*, 2018). Arduino Leonardo kartının bir örneği Şekil 4.6'da gösterilmektedir. Sistemde mikro denetleyici birimi olarak kullanılan bu kart, XBee koordinatöründen gelen verileri almayı, göndermeyi ve kullanılabilir veri formuna dönüştürmeyi sağlayan uygun kodlarla programlanmıştır. Ayrıca, bu karttaki kod, kullanıcıdan gelen komutları hedef düğüme aktaran XBee koordinatörüne aktarmak için kullanılmaktadır.



Şekil 4.6 Arduino Leonardo kartı

4.2.4 Raspberry Pi 3 B

Raspberry Pi, ev otomasyon sistemini gerçek zamanlı izlemek ve kontrol etmek için ve sistemi internet üzerinden buluta bağlamak için bir ağ geçidi olarak kullanılır. Tüm bu işlemler ve daha fazlası, sistem için tasarlanmış grafik kullanıcı arayüzü (GUI) olan üst düzey bir kodla yapılır. Raspberry Pi için tasarlanan GUI, Arduino Leonardo kartından veri almaktan, alınan verileri ThingSpeak bulutunda bulunan kanala gerçek zamanlı olarak göndermekten ve ayrıca uzak bilgisayarlar üzerinden aygıtları kontrol etmek için geliştirilen GUI'den gelen komutları uygulamaktan sorumludur. Bu GUI'lerin ayrıntıları çalışmanın devamında yazılım bölümünde ayrıntılı bir şekilde açıklanmaktadır.

4.3 Düğüm I: Ev Güç Kontrol Düzümü

Bu düğüm, ev ana voltajını, ev aletlerine uygulanan akımı izlemekten ve ana düğümden kablosuz olarak gelen komutlara bağlı olarak ev ana kontaktörünü etkinleştirip veya devre dışı bırakarak ana şebeke gücünü açık veya kapalı konuma getirmekten sorumludur. Şekil 4.7, Düzüm I için tasarlanmış sistemi göstermektedir. Bu düğüm; Algılama, Form, Yerel Sunma, Gönderme, Bekleme, Alma ve Yürütme görevlerini gerçekleştirmek için tasarlanmıştır.

Algılama: Eve uygulanan ev voltajını ve akımı voltaj ve akım sensörleri kullanarak algılar.

Form: Arduino Leonardo kartı içindeki kod vasıtasıyla sensörlerden elde edilen verileri yönetilebilir değerlere dönüştürerek Rasperyy pi üzerinde kurulu olan GUI'nin anlayabileceği hale getirir.

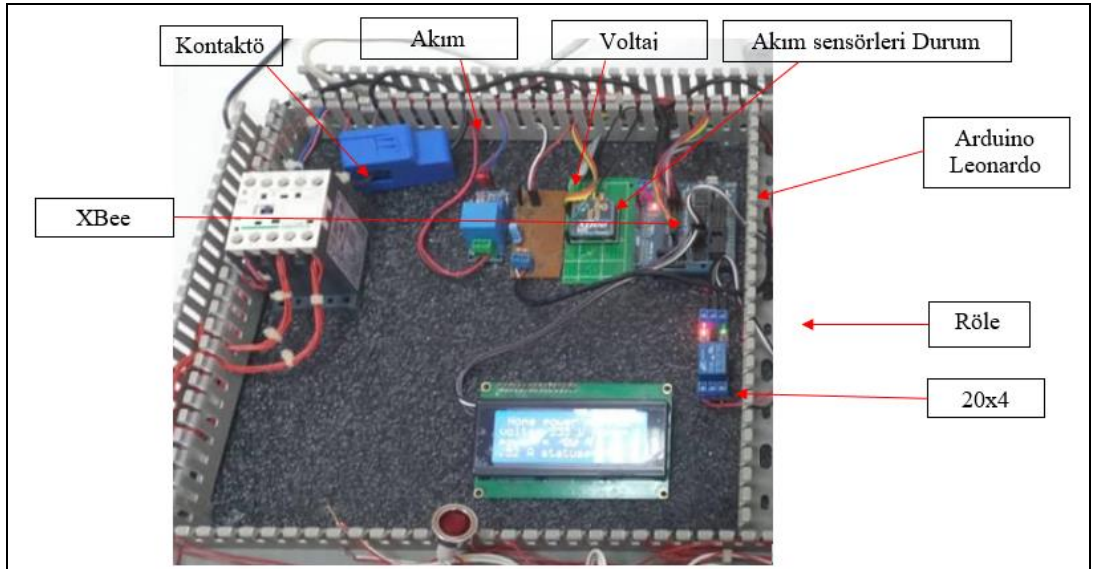
Yerel Sunma: Oluşturulan verileri yerel izleme için 20x4 LCD kullanarak gösterme.

Gönderme: Oluşturulan verileri ana düğümüne XBee modülü üzerinden kablosuz olarak gönderir.

Bekleme: Ana düğümünden XBee modülü aracılığıyla kablosuz olarak alınan herhangi bir komut olup olmadığını kontrol eder.

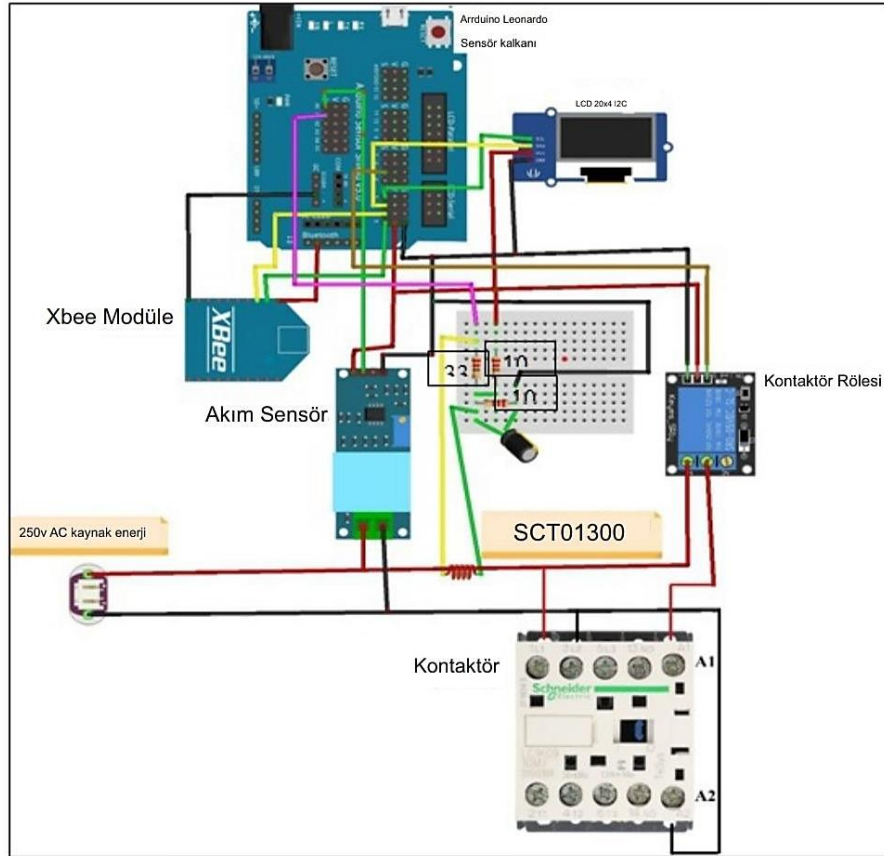
Alma: Uygunsa komutları alır.

Yürütme: XBee vasıtasıyla kablosuz olarak ana düğümünden alınan komuta bağlı olarak röleler için atanan portları etkinleştirir veya devre dışı bırakır.



Şekil 4.7 Düğüm I için tasarlanmış sistem

Şekil 4.8, Düğüm I'nin elektronik tasarımını gösterir. Bu düğüm akım sensörü, voltaj sensörü, XBee modülü, LCD 20 x 4 I2C ve Röle'den oluşur. Tüm bu cihazlar Arduino Leonardo kartına bağlıdır, ayrıca evin gücünü kontrol etmek için kullanılan kontaktör vardır. Bu elektronik elemanlar aşağıdaki başlıklarda ayrıntılı olarak açıklanmaktadır.



Şekil 4.8 Düğüm I'nin elektronik tasarımı.

4.3.2 SCT013-000 Bölünmüş Çekirdekli Akım Sensörü

Bu sensör, ev aletleri, AC motorlar, aydınlatma ekipmanları, hava kompresörü vb. için akım ölçümü, akım izleme ve akım koruması için kullanılır. SCT013-00 akım sensörü Şekil 4.9'da gösterilmiştir. Bu sensör, akımı izlemek ve ölçmek için ve ayrıca ev aletlerini korumak için kullanıldı ve devre kesiciye giden kabloya bağlandı. Akım dönüştürücünün çıkışı, voltaj çıkış tipi yerleşik örnekleme direncinin sonucu olarak sensörün halkası tarafından çevrelenen AC akımıyla orantılı bir AC voltajıdır.

Giriş akımını 0 "0A" da 50mv AC "100A" arasında temsil etmek için AC çıkış voltajını ve 0 ila 100A arasındaki AC akımını ölçebilir (InnovatorSguru, 2019).



Şekil 4.9 SCT013-00 Akım Trafosu.

4.3.2.2 Akım sensörünün Arduino Leonardo kartı ile arabirimlenmesi

Bir akım sensörünü bir Arduino'ya bağlamak için, akım sensöründen gelen çıkış sinyalinin, Arduino analog girişlerinin giriş gereksinimlerini, yani 0V ile ADC referans voltajı arasında pozitif bir voltajı karşılaması için uygun duruma getirilmesi gerekir. Bilindiği gibi akım sensörünün çıkış sinyali türü bir akım çıkışıdır, bu nedenle akım sensörünün iki çıkış hattına paralel olarak bağlanan “yük direnci” adı verilen bir direnç kullanılarak gerilim sinyaline dönüştürülmesi gerekir. Bu direncin değeri aşağıdaki formüllerden bulunabilir:

100A'da akım sensörünün maksimum çıkış değeri 50mA RMS ile sonuçlanacaktır (*SCT013-000 Split core current transformer*, 2017).

Tepeden tepeye akım değerini RMS akım değerinden hesaplanabilir, bunun için Eşitlik 4.1 kullanılır.

$$I_{ptop} = I_{RMS} * 2\sqrt{2} = 50 * 2\sqrt{2} = 141.42 \text{ mA} \quad (4.1)$$

Arduino kartının maksimum voltajının 5v olduğu bilinmektedir, böylece yük direnci ohm kanunundan hesaplanabilir:

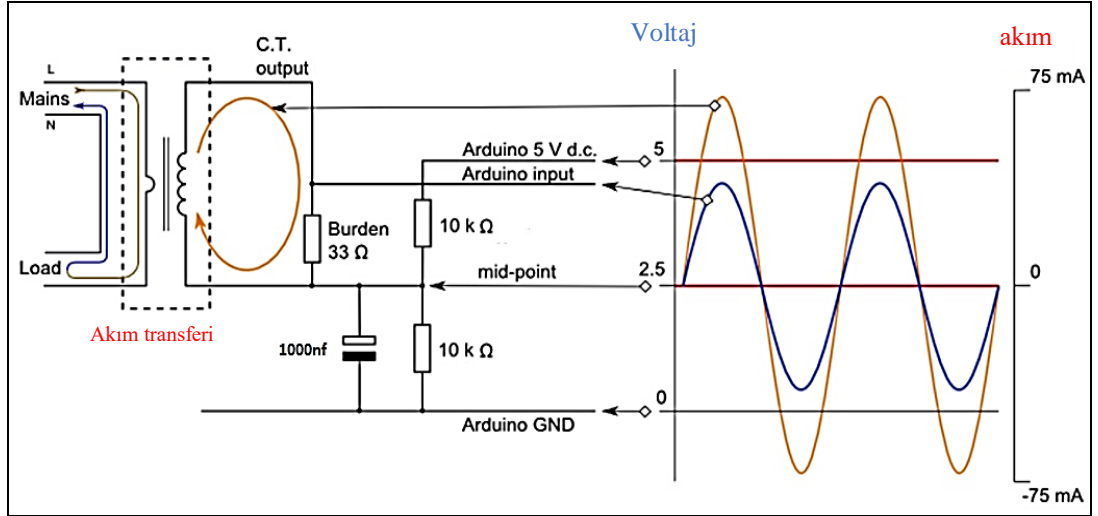
$$R = \frac{V}{I} = \frac{5}{141.42 \times 10^{-8}} = 35.35 \Omega \quad (4.2)$$

35 Ω ortak bir direnç değeri değildir. 35 Ω 'un her iki tarafındaki en yakın değerler 39 ve 33 Ω 'dur. Her zaman daha küçük değeri seçilir, % 33 \pm % 1'lik bir yük önerilir.

Bu, iki ana bölümden oluşan Şekil 4.10'da gösterilen durum devresi ile sağlanabilir:

1. CT sensörü ve 33 Ω olan yük direnci
2. R1 = R2 = 10k Ω olan *polarlama* voltaj bölücü (R1 ve R2)

Bu çalışmada akım sensörü bu durum devresiyle birlikte kullanılmıştır. Bu sensörün çıkışı Arduino Leonardo'nun A1 analog portuna bağlanarak değer okuması gerçekleştirilir.



Şekil 4.10 Arduino için akım sensörü durum devresi.

4.3.3 ZMPT101B AC Tek Fazlı Voltaj Sensörü Modülü

ZMPT101B tek fazlı gerilim sensör modülü, Arduino mikrodenetleyici kartları ile rahatlıkla kullanılmak üzere tasarlanmıştır. Çalışmada, evi güç kaynağı değişikliklerine karşı korumak, kararlı olduğundan emin olmak ve eve beslenen voltaj kaynağını ölçmek için bu voltaj sensörü kullanılmıştır. Sensör iki kısımdan oluşur. Giriş kısmı ölçülmesi gereken 0-250 AC voltajı bağlamak için kullanılır.

Çıkış kısmı vcc, toprak ve çıkış okuma değerini (DC mili-volt) içerir ve bu değerler Arduino ile işlenebilir. Geliştirilen sistemde bu sensörün çıkışı Arduino kartının A0 portundan okunmaktadır. ZMPT101B AC tek fazlı voltaj sensörü modülü, Şekil 4.11'de gösterilmektedir.

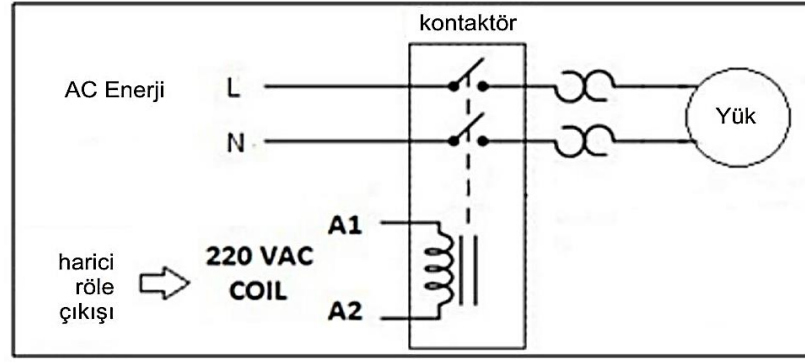


Şekil 4.11 ZMPT101B AC tek fazlı voltaj sensörü modülü.

4.3.4 Kontaktör

Bir kontaktör, bir elektrik güç devresini anahtarlama için kullanılan elektrikle kontrol edilen bir anahtardır. Daha yüksek akım derecesi dışında röleye benzer. Kontaktör, anahtarlama devresinden çok daha düşük güç seviyesine sahip bir devre tarafından kontrol edilir. Yüksek akımlı yük cihazlarına doğrudan bağlanır ve onları kontrol eder. Bu cihazlar motorlar, aydınlatma, ısıtma, hava kompresörleri, termal buharlaştırıcılar ve diğer elektrik yükleri olabilir. Kontaktörler tipik olarak birden fazla kontakta sahiptir ve bu kontaklar genellikle (ancak her zaman değil) normalde açıktır, böylece bobinin enerjisi kesildiğinde yüke giden güç kesilir (*Contactors : Electromechanical Relays*, 2016).

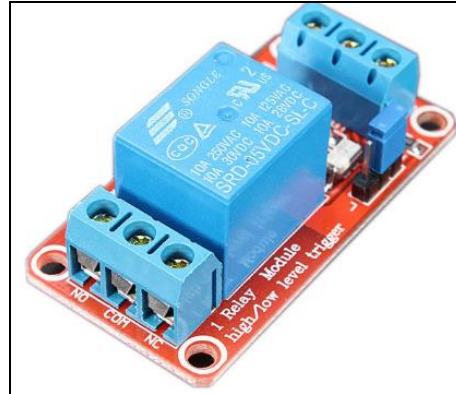
Bu çalışmadaki tasarımda kontaktöre enerji vermek / enerjisini kesmek için harici röle kullanılmıştır. Bu röle kontaktörün A1 ve A2 pinlerine bağlanmıştır. Tek fazlı kontaktör, Şekil 4.12'de gösterilmektedir.



Şekil 4.12 Tek Fazlı Kontaktör.

4.3.5 Röle

Bir röle, açılıp kapatılarak akımın geçip geçmemesini düzenleyen ve Arduino pinleri tarafından sağlanan 5V gibi düşük voltajlarla kontrol edilebilen elektrikle çalışan bir anahtardır (KonnectedInc, 2018). Röle modülü Şekil 4.13'te gösterilmektedir. Bu röle modülünün bir kanalı vardır ve bir Arduino tarafından 5V ile desteklenmelidir. Bu çalışmada röle Arduino kartının D5 pinine bağlanmıştır. Bu pinden lojik 1 veya 0 gönderildiğinde röle aktif ya da pasif hale gelir. Sonrasında A1 ve A2 fişlerine 220VAC uygulayarak kontaktörü aktif ya da pasif hale getirir.



Şekil 4.13 Röle modülü.

4.3.6 Arduino Leonardo Kartı

Bu kart, bu düğümde A0 (Gerilim sensörü) ve A1 (Akım sensörü) analog portlarından gelen verileri okumak için programlanmıştır. Bunun yanı sıra XBee Node I aracılığıyla verileri ana düğüme aktarmadan önce ana düğümün

anlayabileceği bir diziye çevirir ve kalibrasyonları yapar. Ayrıca, Düğüm I'deki XBee ile XBee koordinatörü arasında bağlantı kurulduğundan emin olmak için gerekli prosedürleri yerine getirir ve bunlar üzerinden gelen ya da giden verilerin işlenmesinden sorumludur.

4.3.7 XBee Uç Cihazı

Bu cihaz, Arduino Leonardo kartından ana düğüme kablosuz olarak verilerin gönderilmesinden veya alınmasından sorumludur. Arduino Leonardo kartına rx ve tx portları olan D0 ve D1 pinleri üzerinden bağlanır.

4.3.8 20 x 4 LCD

Bu eleman sensörler tarafından ölçülen akımın, voltajın ve evin güç durumunun yerel olarak izlenmesi için kullanılır. Arduino kartındaki dijital pin 2 ve pin 3'e bağlanır. Düğüm I'in yerel izleme çıktı örneği Şekil 4.14 gösterilmektedir.



Şekil 4.14 Düğüm I yerel izleme görüntüsü.

4.4 Düğüm II: İklim Kontrol Düzümü

Bu düğüm, ev iklim kontrolü için kullanılır. Ev sıcaklığını, nemi ve gaz kaçağı seviyelerini izlemek ve kontrol etmek için yapılandırılmıştır. Ayrıca ev sıcaklığını ve nemini kontrol etmek için klima ve ısıtıcıyı da kontrol edebilir. Yüksek gaz kaçağı seviyelerinde ev ana gücünü kapatmaya programlanmıştır. Şekil 4.15, Düğüm II için tasarlanmış sistemin mimarisini göstermektedir. Bu düğüm; Algılama, Form, Yerel

Sunma, Gönderme, Bekleme, Alma ve Yürütme görevlerini gerçekleştirmek için tasarlanmıştır.

Algılama: Bu düğümde; ev dumanı, gaz kaçağı, sıcaklık ve nemi algılamak için iki sensör (MQ2 ve DH22 sensörleri) kullanılmıştır.

Form: Arduino Leonardo kartı içindeki kod vasıtasıyla sensörlerden elde edilen verileri yönetilebilir değerlere dönüştürerek Rasperyy pi üzerinde kurulu olan GUI'nin anlayabileceği hale getirir.

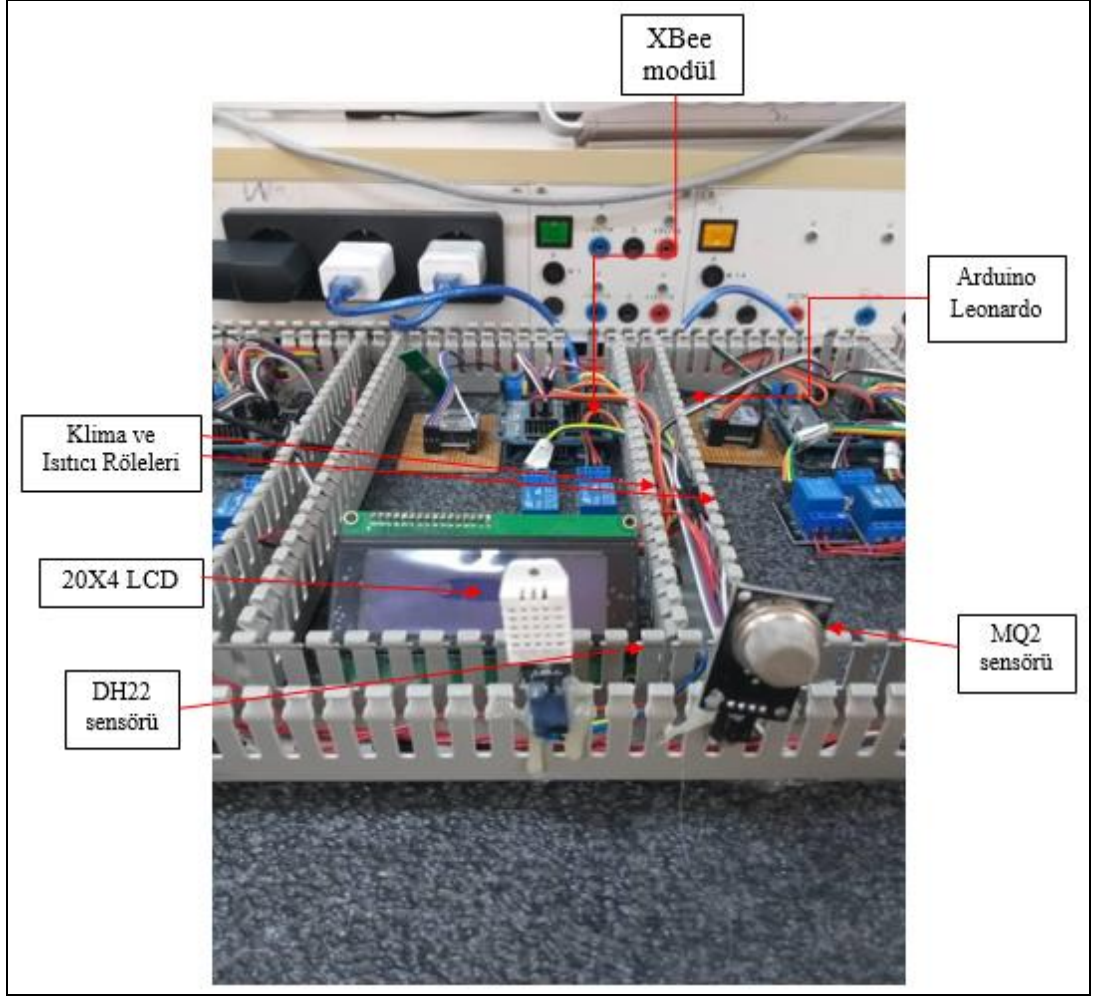
Yerel sunma: Sensörlerden okunan değerleri 20x4 LCD üzerinde göstererek yerel izleme sağlar.

Gönderme: Oluşturulan verileri XBee modülü üzerinden kablosuz olarak ana düğüme gönderir.

Bekleme: Ana düğümden XBee modülü aracılığıyla kablosuz olarak alınan herhangi bir komut olup olmadığını kontrol eder.

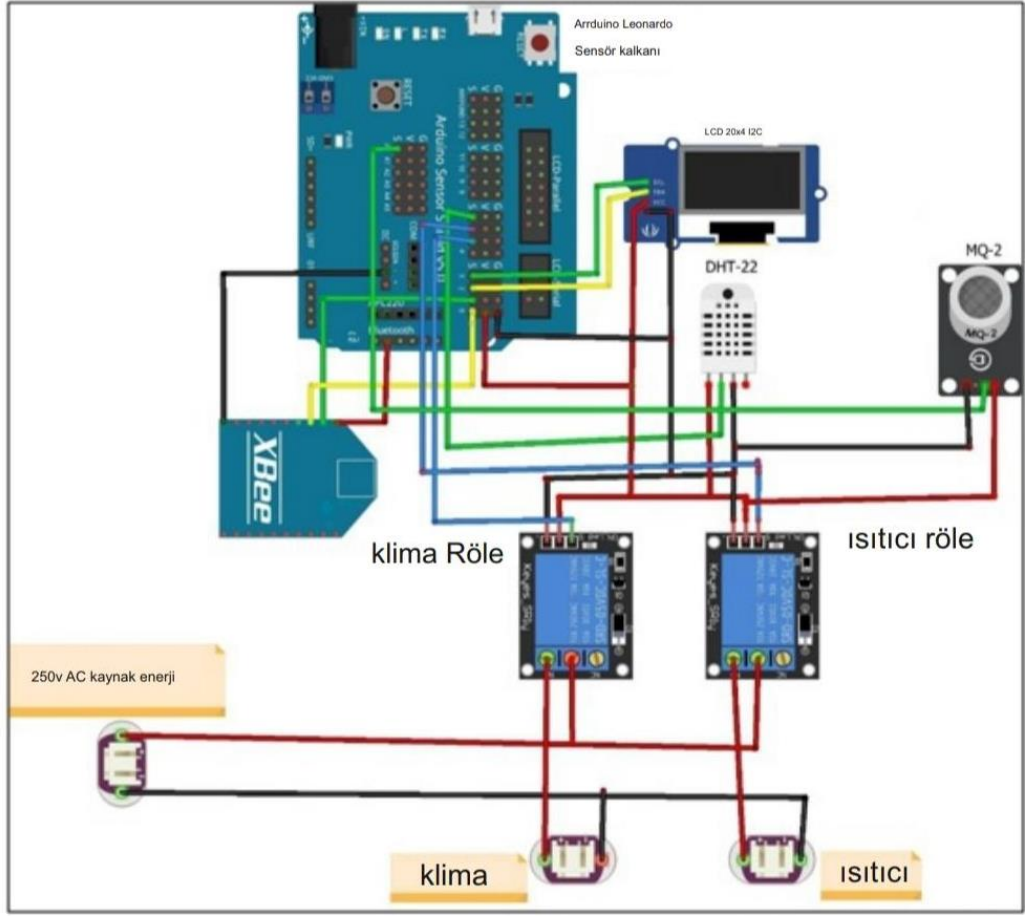
Alma: Uygunsa komutları alır.

Yürütme: XBee vasıtasıyla ana düğümden kablosuz olarak alınan komuta bağlı olarak röleler için atanan portları etkinleştirir veya devre dışı bırakır.



Şekil 4.15 Düğüm II için tasarlanmış sistemin mimarisi.

Şekil 4.16, Node II'nin elektronik tasarımını göstermektedir. Bu tasarım; DH22 sensörü, MQ2 gaz sensörü, Xbee modülü, LCD 20 x 4 I2C ve iki Röle içerir. Tüm cihazlar Arduino Leonardo kartına bağlıdır. Bu elemanlar ve düğümdeki görevleri aşağıda açıklanmaktadır.



Şekil 4.16 Dügüm II'nin elektronik tasarımı.

4.4.1 MQ2 Sensörü

Çok sayıda sensör vardır ve bu sensörlerden bazıları güvenlik sistemlerinin önemli bir parçasıdır, örneğin yangını tespit etmek ve bu yangının engellenmesinde gerekli önlemleri almak için yangın sensörleri kullanılır. MQ2 gaz sensörü, güvenlik sistemlerinde kaçak gazları tespit etmek için kullanılan bu sensörlerden biridir. Şekil 4.17, MQ2 gaz sensörünü göstermektedir.



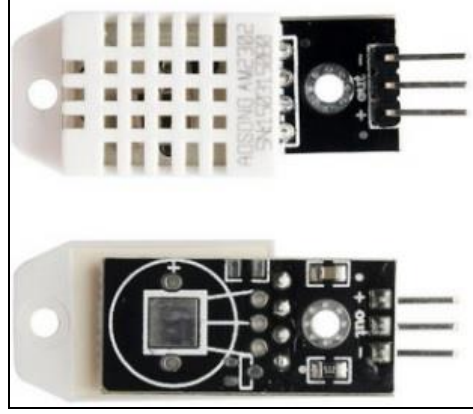
Şekil 4.17 MQ2 gaz sensörü

MQ2 gaz sensörü, kapalı mekanlarda gaz kaçağı seviyesini tespit etmek için kullanılan elektronik bir sensördür, bu sensör evde veya fabrikada gaz kaçağını izlemek için kullanılabilir. MQ2, havadaki LPG, duman, hidrojen, metan, propan, alkol ve karbon monoksit ve aynı zamanda kimyasal direnç gibi gazların seviyesini algılama özelliğine sahiptir. Bu sensör, metal oksit yarı iletken tipte bir gaz sensörüdür. MQ2, gazla temas ettiğinde direnci değişen kalay oksit (SnO_2) bir malzeme içerir. Direnç değerindeki bu değişiklik gaz kaçağını tespit eder. Gaz konsantrasyonlarının miktarı, sensörde bulunan bir voltaj bölücü devre ile ölçülür (*How MQ2 Gas/Smoke Sensor Works*, 2018). Bu sensörün uzunluğu 35 mm, genişliği 22 mm ve yüksekliği 23 mm'dir. Ana yongada LM393, ZYMQ-2 gaz sensörleri vardır. Ayrıca DC 5 voltajında, hızlı tepki ve yüksek hassasiyette çalışır (Mybotic, 2016).

Sensör bu çalışmada evdeki sigara ve gaz kaçağını ölçmek için kullanılmıştır. Geliştirilen yazılım, yüksek gaz kaçağı seviyelerinde ev ana gücünü otomatik olarak kapatmak için yapılandırılmıştır. Sensörün çıkış pini, Arduino Leonardo'nun analog giriş pini A0'a bağlanmıştır.

4.4.2 DH22 Sıcaklık ve Nem Sensörü

DHT22 düşük maliyetli bir dijital nem ve sıcaklık sensörüdür, mekanların içindeki sıcaklık ve nemi ölçmek ve kaydetmek isteyenler için önemlidir. Bu sensör, oda sıcaklığının izlenmesi veya hava durumu istasyonu gibi birçok uygulama için mükemmeldir. DHT22 sensörü iki parçadan yapılmış bir kapasitif nem sensörüdür, çevredeki havayı ölçer ve veri pini ile dijital bir sinyal iletir. Kullanımı basittir, mükemmel kalitede ve hızlı bir şekilde 2 saniyede bir yanıt verir (*DHT22 Digital Temperature and Humidity Sensor*, 2018). Şekil 4.18, DHT22 sıcaklık ve nem sensörünün bir örneğini göstermektedir. Soldaki ilk pin 3-5V güce, ortadaki pin veri giriş pinine ve son pin toprağa bağlanır. Bu sensör hem 3.3V hem de 5V'da çalıştırılabilir, bu nedenle Arduino, Raspberry Pi ve diğer tüm mikrodenetleyiciler gibi tüm standart kartlara bağlanmaya uygundur. Bu çalışmada sensörün çıkışı Arduino Leonardo kartındaki Dijital giriş D7'ye bağlanmıştır.



Şekil 4.18 DHT22 Sıcaklık ve nem sensörü.

4.4.3 Arduino Leonardo Kartı

Çalışmada bu kart, analog port A0 (MQ2 gaz sensörü) ve dijital port D7'den (DHT22 sensör) gelen verileri okumaktadır. Ayrıca D5 (Klima Rölesi) ve klima ve ısıtıcıyı kontrol etmek için kullanılan D6 (Isıtıcı Rölesi) dijital çıkış bağlantı portlarını kontrol eder. Kod, iki bağlantı noktasından gelen farklı veriler için gerekli mantık dizilerini düzenler ve XBee düğümü II aracılığıyla ana düğüme aktarmadan önce belirli bir formatta kalibre eder. Ayrıca, Düğüm II'deki XBee ile XBee koordinatörü arasında bağlantı kurulduğundan emin olmak için prosedürleri içerir ve bunlar üzerinden gelen veya giden verilerin işlenmesinden sorumludur.

4.4.4 Xbee Uç Cihazı

Bu cihaz, Arduino Leonardo kartından ana düğüme kablosuz olarak verilerin gönderilmesinden veya alınmasından sorumludur. XBee uç cihazından kablosuz olarak gönderilen bu veriler, Arduino Leonardo kartı üzerinden XBee düğümüne aktarılan MQ2 ve DH22 sensörlerinden elde edilen verileri içerir. XBee uç cihazı tarafından alınan veriler ise, Arduino Leonardo kartı kullanılarak işlenir ve yürütülecek olan bu düğümlere bağlı aletleri (klima ve ısıtıcı) başlatmak veya durdurmak için ana düğümden kablosuz olarak alınan ve uzak masaüstü uygulamasından gelen açık veya kapalı komutlarını içerir.

4.4.5 20 x 4 LCD

Bu eleman, sensörler tarafından ölçülen duman, gaz, sıcaklık ve nem seviyelerinin yerel olarak izlenmesi için kullanılmıştır. Arduino kartındaki pin 2 ve pin3'e bağlanmıştır. Şekil 4.19, düğüm II için duman, gaz, sıcaklık ve nem seviyelerinin ekrandaki izlenmesini göstermektedir.



Şekil 4.19 Düğüm II yerel izleme

4.4.6 Röleler

Bu düğümde iki röle kullanılmıştır, biri D5 dijital çıkış portuna bağlı klimayı açık veya kapalı konuma getirmek için, diğeri ise D6 dijital çıkış portuna bağlı ısıtıcıyı açık veya kapalı konuma getirmek için kullanılmaktadır. Her iki röle de yüksek seviyeli kodla ev dışından kontrol edilebilir ve gerekli komutlar XBee ağı kullanılarak ana düğüm üzerinden gönderilir.

4.5 Düğüm III: Yangın ve Cihaz Kontrol Düğümü

Bu düğümde yangın alarm sistemi, gece aydınlatma sistemi ve cihaz (tv ve buzdolabı) uzaktan açma veya kapama kontrol sistemi bulunur. Evde yangın olduğunda hoparlör çalışmaktadır. Şekil 4.20, Düğüm III için uygulanan sistemin mimarisini göstermektedir. Bu düğüm de Düğüm II'de olduğu gibi; Algılama, Form, Yerel Sunma, Gönderme, Bekleme, Alma ve Yürütme görevlerini gerçekleştirmek için tasarlanmıştır.

Algılama: Bu düğümde, ev bahçe ışıklarını ve evde yangının varlığını izlemek için iki sensörden (alev ve ışık sensörleri) değer okunmaktadır.

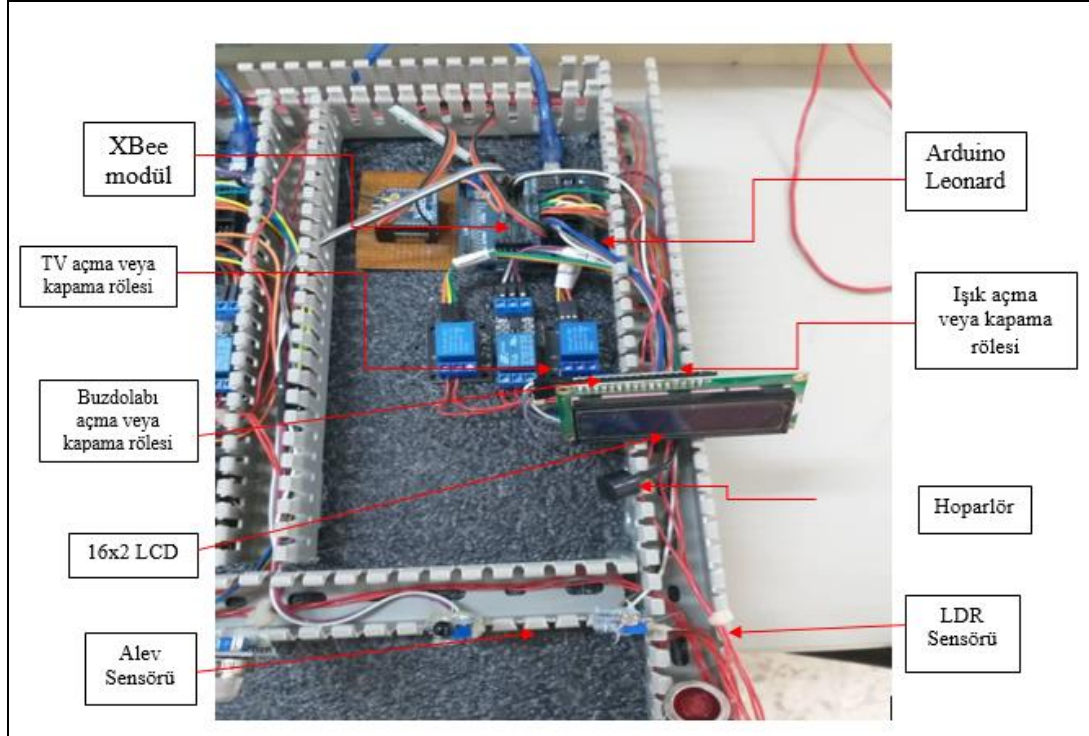
Form: Arduino Leonardo kartındaki kodu kullanarak sensörlerden elde edilen verileri, Rasperry pi'ye üzerindeki GUT'nin yönetebileceği değerlere dönüştürür.

Gönderme: Oluşturulan verileri XBee modülü üzerinden ana düğüme kablosuz olarak gönderir.

Bekleme: Ana düğümden XBee modülü aracılığıyla kablosuz olarak alınan herhangi bir komut olup olmadığını kontrol eder.

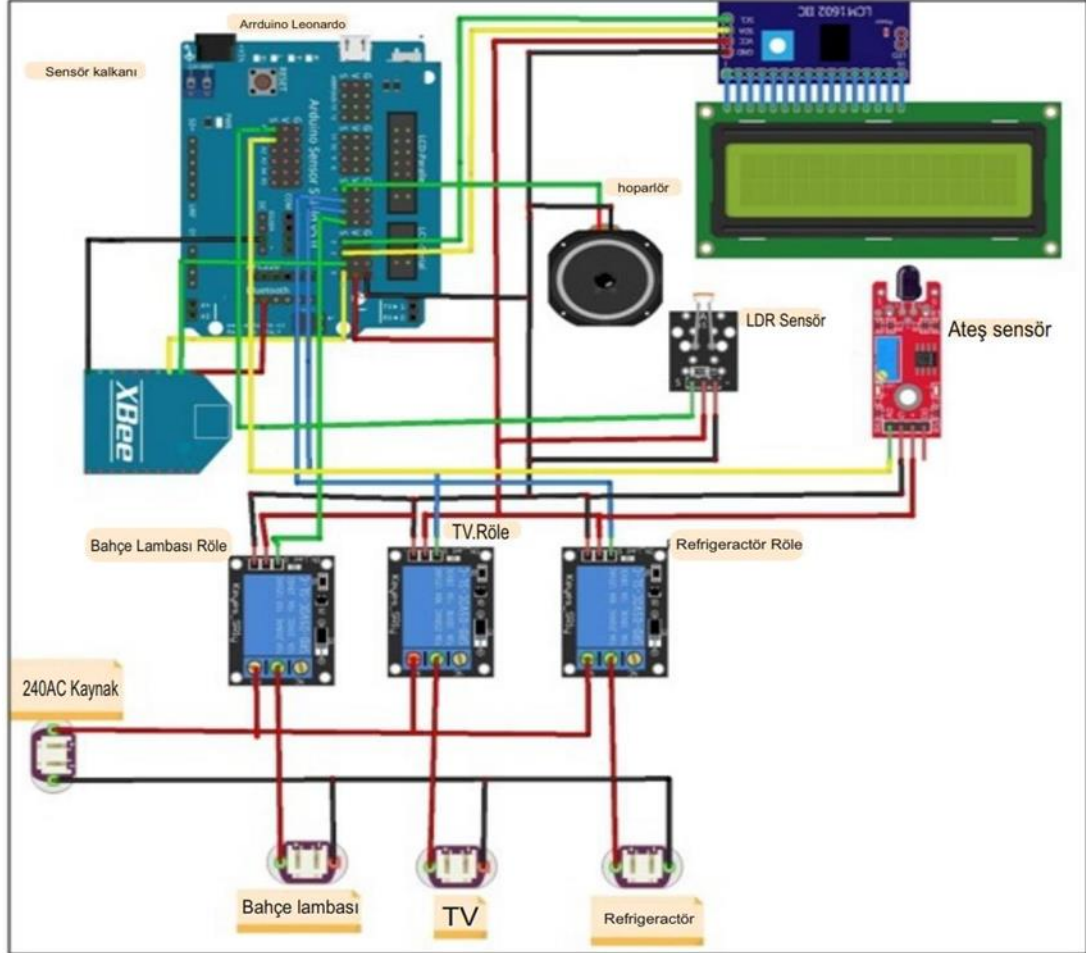
Alma: Uygunsa komutları alır.

Yürütme: Ana düğümden XBee vasıtasıyla kablosuz olarak alınan komuta bağlı olarak röleler için atanan portları etkinleştirir veya devre dışı bırakır.



Şekil 4.20 Düğüm III için uygulanan sistem

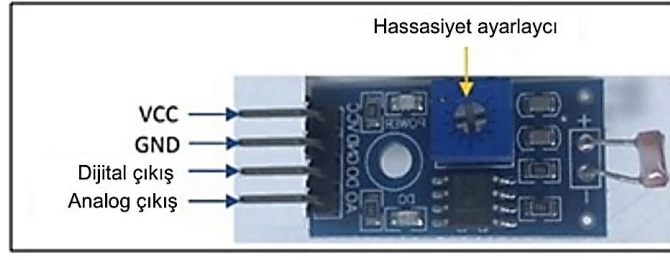
Şekil 4.21, Düğüm III'ün elektronik tasarımını göstermektedir. Bu düğüm LDR sensörü, alev sensörü, Xbee modülü, LCD 16 x 2 I2C, hoparlör ve üç röleden oluşur. Tüm bu cihazlar Arduino Leonardo kartına bağlıdır. Bu elemanlar aşağıda ayrıntılı olarak açıklanmaktadır.



Şekil 4.21 Düğüm III'ün elektronik tasarımı.

4.5.1 LDR Işığa Bağlı Direnç

Gün ışığının yoğunluğuna bağlı olarak, dış mekan ışıklarının, sokak lambalarının ve bir dizi ev aletinin otomatik olarak kapatılması için ışık sensörü devresi kullanılabilir, bu da enerji israfının üstesinden gelinmesini sağlar (WatElectronics, 2019). LDR, ışığa bağlı dirençtir ve yüzeyine düşen ışık miktarı değiştiğinde direnci değişir. Buna fotosel, foto direnç ve fotoiletken denir. LDR sensörü Şekil 4.22'de gösterilmektedir. Bu çalışmada LDR, karanlıktan ışığı açmak ve aydınlıkta ise ışığı kapatmak için kullanılmaktadır.



Şekil 4.22 LDR Sensörü

LDR üzerinde ışık olduğu zaman direnci ışığın yoğunluğuna göre azalacak, karanlıkta artacaktır. Işık yoğunluğu ne kadar yüksekse, LDR'nin direnci o kadar düşüktür. Karanlık yere bir LDR yerleştirildiğinde, direnci yüksek olurken, ışıқта direnci düşük olacaktır (WatElectronics, 2019). LDR sensöründe, ışığa karşı hassasiyeti ayarlamak ve değiştirmek için kullanılan bir potansiyometre vardır. Işık yoğunluğuna duyarlılığı artırmak için ayar düğmesi saat yönünde hareket ettirilirken, ışık yoğunluğuna karşı duyarlılığı azaltmak için ayar düğmesi saat yönünün tersine hareket ettirilir.

Çalışmada sensör çıkışı Arduino kartının analog giriş portu A1'e bağlanmıştır. Bu sensörden gelen veriye göre, Arduino kartının D4 dijital portuna bağlı ışıkları açma veya kapama Rölesi kontrol edilmektedir. Bu röle etkinleştirilerek veya devre dışı bırakılarak, bahçe ışıklarının otomatik olarak açılıp kapanması sağlanmaktadır.

4.5.2 KY-026 Alev Sensörü Modülü

KY 026 alev sensörü, yangında yayılan ışık vasıtasıyla yangını algılamak için kullanılan optik bir alev sensörüdür. Bu sensör, aleve ve radyasyona karşı en hassas olanıdır ve analog veya dijital sinyal yayabilir. 760 nm ila 1100 nm dalga boyu aralığında sıradan ışık kaynaklarını algılama yeteneğine sahiptir (VoltT, 2018). KY 026 sensörü sayesinde, yangınlar erken tespit edilir. Bu, meydana gelen yangınları önlemeye yardımcı olur, can ve mal kurtarır. Yüksek sıcaklıktan dolayı sensöre zarar gelmemesi için, en kısa test mesafesi 80 cm olduğu için alev ile sensör arasında belirli bir mesafe bırakılmalıdır. KY 026 sensörü Şekil 4.23'te gösterilmektedir.



Şekil 4.23 KY 026 sensörü

Bu sensör özellikle alevin spektrumuna duyarlıdır. Alev görünümünden farklı dalga boylarına ayarlanır ve genellikle kızılötesi ve ultraviyole sinyalleri birleştirir. Dijital potansiyometrenin ayarlanmasıyla ayarlanabilir hassasiyete sahiptir. 3,3 V ila 5 V çalışma voltajına ihtiyaç duyar. Bu çalışmada sensör evde yangının olup olmadığını tespit etmek ve yangın durumuna göre devreye alınan bir hoparlörü kontrol etmek için kullanılmıştır. Hoparlör, Arduino kartının dijital çıkış portu D7'ye bağlıdır. Sensör çıkış pini ise Arduino kartının A0 analog pinine bağlıdır.

4.5.3 TV ve Buzdolabı Röleleri

Bu röleler, XBee ağı üzerinden kablosuz olarak ana düğümden alınan uzaktan komutlara bağlı olarak TV ve Buzdolabı cihazlarını açmak veya kapatmak için kullanılır. Arduino kartının dijital çıkış pini D5 TV Rölesini etkinleştirmek veya devre dışı bırakmak için ve dijital çıkış pini D6 ise buzdolabı rölesini etkinleştirmek veya devre dışı bırakmak için atanmıştır.

4.5.4 Arduino Leonardo Kartı

Bu kart, A0 ve A1 analog pinlerinden gelen verilere göre bahçe ışıkları, TV, buzdolabı ve yük hoparlörü etkinleştirmek veya devre dışı bırakmak için kullanılan dijital portlar D4, D5, D6 ve D7'yi kontrol etmek için kullanılır. Ayrıca, iki porttan gelen farklı veriler için gerekli mantık dizilerini yapar ve Düğüm III'deki XBee aracılığıyla ana düğüme aktarmadan önce belirli bir biçimde düzenler. Bunun yanı sıra, Düğüm III'deki XBee ile XBee koordinatörü arasındaki bağlantıyı sağlamaya yönelik prosedürleri içerir ve bunlar üzerinden gelen veya giden verilerin işlenmesinden sorumludur.

4.5.5 XBee Uç Cihazı

Bu cihaz, Arduino kartından ana düğüme kablosuz olarak aktarılan verilerin gönderilmesinden veya alınmasından sorumludur. XBee uç cihazından kablosuz olarak gönderilen veriler, LDR ve KY 026 sensörlerinden elde edilen değerleri içerir. XBee uç cihazı tarafından alınan veriler ise, Arduino Leonardo kartı kullanılarak işlenecek ve çalıştırılacak olan cihazları (bahçe lambası, TV ve buzdolabı) başlatmak/durdurmak için ana düğümden kablosuz olarak alınan açık veya kapalı komutlarını içerir.

4.5.6 16 x 2 LCD

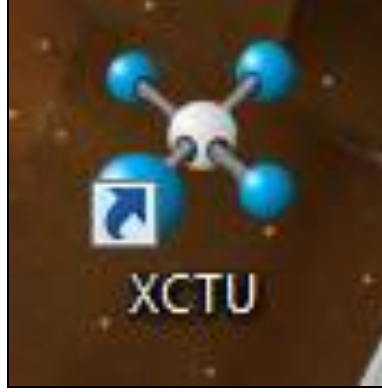
Bu ekran, yangının varlığını ve bahçe ışıklarının durumunu yerel olarak izlemek için kullanılmıştır. Yangın ve bahçe ışıklarının durumu Şekil 4.24'deki gibi ekran üzerinde izlenebilmektedir.



Şekil 4.24 Düğüm III yerel izlemenin çıktı örneği

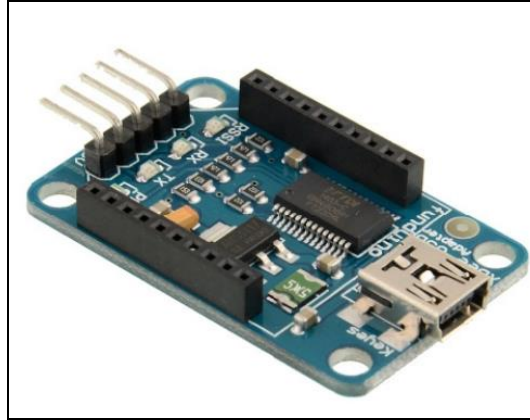
4.6 XBee Modüllerini Yapılandırma

Bu bölümde, XBee modüllerini ZigBee ağında çalışacak şekilde yapılandırmak için kullanılan yapılandırma parametreleri ve adımları sunulmaktadır. XBee'nin parametrelerini yapılandırmak için Digi uluslararası şirketi tarafından sağlanan ücretsiz Digi X-CTU 6.4.0 yardımcı yazılımı kullanılmıştır. X-CTU, geliştiricilerin digi radyo frekansı (RF) modüllerini kullanımı basit bir grafik arayüz aracılığıyla yönetmelerine olanak tanıyan ücretsiz bir çoklu platform uygulamasıdır. Uygulama, Digi RF modüllerini kurmayı, yapılandırmayı ve test etmeyi kolaylaştıran gömülü araçlar içerir (*XCTU User Guide*, 2019). X-CTU program kısayolu Şekil 4.25'te gösterilmektedir.



Şekil 4.25 X-CTU program kısayolu

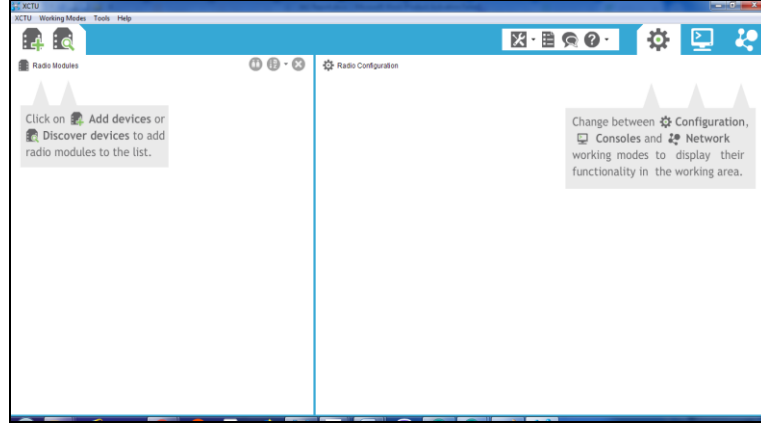
XBee'yi X-CTU kullanarak programlamadan önce, XBee explorer kartına ihtiyaç vardır. Bu kart, XBee parametrelerini yapılandırmak için kullanılır. Ayrıca USB-TTL adaptörü olarak da kullanılabilir. XBee explorer kartının bir örneği Şekil 4.26'da gösterilmektedir.



Şekil 4.26 XBee gezgini kartı

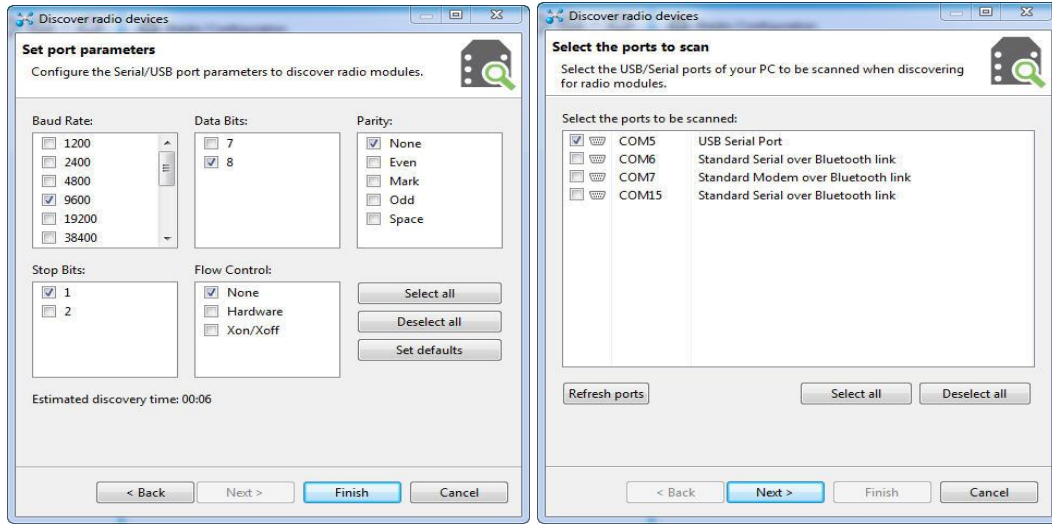
Ağda XBee cihazlarını yapılandırmak için gerçekleştirilmesi gereken adımlar aşağıda açıklanmaktadır.

1. XBee'yi, XBee gezgini kartına taktıktan ve kartı PC'nin USB portuna bağladıktan sonra X-CTU yazılımı çalıştırılır. XCTU yardımcı yazılımının ekran görüntüsü Şekil 4.27'de gösterilmektedir
2. Karta bağlı XBee'yi eklemek için “Discover devices” sekmesi seçilir.



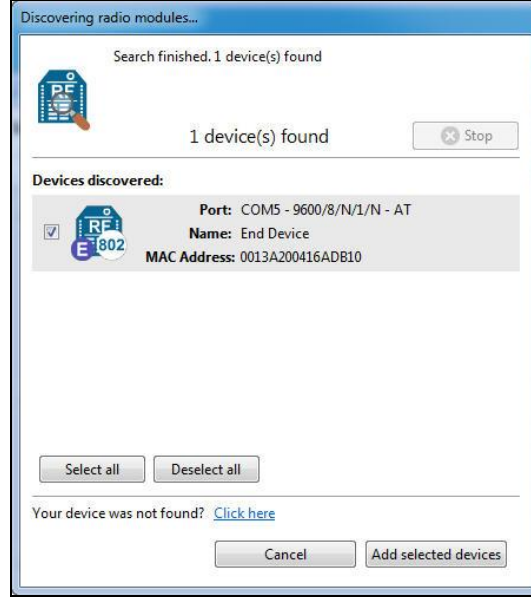
Şekil 4.27 XCTU yardımcı yazılımının ekran görüntüsü

3. Kartın bağlandığı seri port seçilir ve seri port parametreleri Şekil 4.28'de gösterildiği gibi ayarlanır.



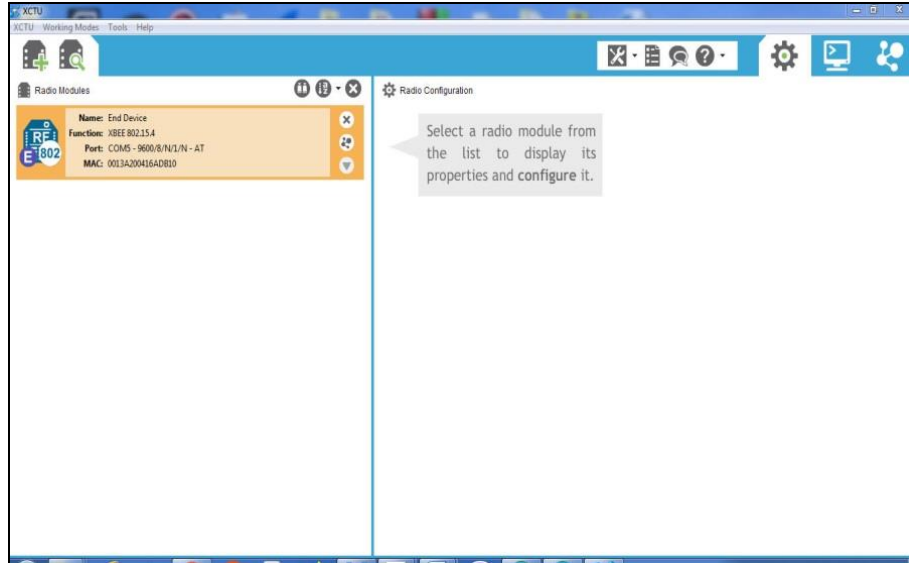
Şekil 4.28 Seri bağlantı noktalarını ayarlama

4. X-CTU yazılımı, XBee'yi buluncaya kadar arayacaktır. XBee bulunduktan sonra seçili cihazları ekle düğmesine tıklanarak XBee cihazı Şekil 4.29'da gösterildiği gibi eklenir.



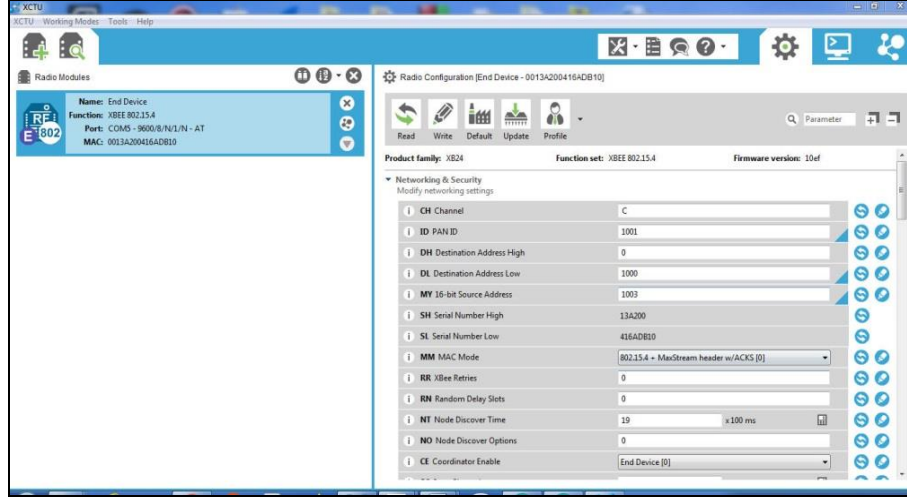
Şekil 4.29 XBee Cihazını Ekleme

5. XBee cihazı eklendikten sonraki X-CTU yazılımının ekran görüntüsü, Şekil 4.30'da gösterilmektedir.



Şekil 4.30 X-CTU yazılımının XBee cihazı eklendikten sonraki ekran görüntüsü

6. XBee cihazına tıkladığında, XBee parametreleri gösterilir ve ardından Şekil 4.31'de gösterildiği gibi yeni parametreleri XBee'ye yazmak için yaz düğmesine tıklanır.



Şekil 4.31 XBee parametrelerinin değiştirilmesi ve kaydedilmesi

Düğümle yapılandırmak için kullanılan parametreler Tablo 4.1'de gösterilmektedir.

Tablo 4.1 Ağdaki düğümleri yapılandırmak için kullanılan parametreler

Parametre	Koordinatör	Düğüm I	Düğüm II	Düğüm III
Kullanılan kanal	C	C	C	C
PAN ID	3332	3332	3332	3332
Yüksek Hedef Adresi (HD)	0	0	0	0
Düşük Hedef Adresi (LD)	FFFF	1111	1111	1111
Cihaz Tipi	Koordinatör	Uç Cihaz	Uç Cihaz	Uç Cihaz
16-bit Adres	1111	1112	1113	1114
Mod	AT	AT	AT	

4.6.1 Parametrelerin Değerlendirilmesi

Kullanılan kanal: Tüm düğümler birbirine bağlanmak için aynı kanala ve PAN (Personal Area Network) ID'ye sahip olmalıdır. PAN Kimliği, kişisel alan ağı kimlik numarasını temsil eden 16 bitlik (4 onaltılık sayı) bir adrestir, her ağ için benzersizdir.

Yüksek Hedef Adresi (HD): 48 bitlik XBee modülünün 24 bitlik yüksek adresini, yani hedef düğümün adresinin yüksek kısmını temsil eder.

Düşük Hedef Adresi (LD): Hedef XBee modülünün 48 bit uzun adresinin alt 24 bitini, "adresin alt kısmını" temsil eder. Düğümün hedef düşük adresi, koordinatörün 16 bitlik adresi olmalıdır, çünkü düğümler tek noktaya yayın iletim modunda

yalnızca koordinatörle iletişim kurarken, koordinatörün hedef adresi FFFF'dir, bu da çok noktaya yayın modunu kullanarak ağdaki tüm düğümlerle iletişim kurmak anlamına gelir.

Cihaz tipi: XBee cihazının ağda son cihaz veya koordinatör olarak çalışma durumunu ortaya koyar. Her XBee ağında yalnızca bir PAN koordinatörü ve 216 uç cihaza kadar en az bir uç cihazı olmalıdır.

16-bit Adres: 4 haneli onaltılık sayı ile temsil edilen XBee modülünün kısa adresidir ve kişisel alan ağında benzersiz olmalıdır.

4.7 Yazılım Bölümü

Önerilen sistemi uygulamak için kullanılan yazılım iki bölüme ayrılmıştır: Düşük seviyeli ve yüksek seviyeli program.

4.7.1 Düşük Seviyeli Program

Düşük seviyeli program, mikrodenetleyiciye bağlı farklı birimlerden gelen veya giden sinyalleri ele almak, işlemek ve mikro denetleyicideki farklı pinleri yapılandırmak için mikro denetleyici üzerinde çalışan komutlar grubunu ifade eder. Bu program ayrıca, Arduino kartının veri ve komutları XBee düğümleri arasında gönderip almasını sağlamak için uygun kodu içerir. Bu kodlar, veri ve komutların kullanılabilir veri formuna dönüştürülerek gönderilip alınmasından sorumludur. Arduino Microcontroller'ı programlamak için açık kaynaklı Arduino Software (IDE) derleyicisi "Arduino.cc" kullanılmıştır (*Arduino - The open-source Arduino Software (IDE) Software*, 2018). Geliştirilen sistemdeki her Arduino kartı, ona bağlı sensörlerden gelen verileri ele almak ve işlemek üzere de programlanmıştır. Her düğüm için geliştirilen kodlar aşağıda ayrıntılı olarak açıklanmaktadır.

Düğüm-I için program kodu:

Daha önce de belirtildiği gibi, Düğüm-I, eve beslenen gücü izlemek ve kontrol etmek için tasarlanmıştır. İzleme süreci, iki güç sensörü vasıtasıyla gerçekleştirilmektedir:

- Ev voltajını ölçmek için Arduino Leonardo analog portu A0'a bağlı tek fazlı AC voltaj sensörü.
- Ev akımını ölçmek için Arduino Leonardo analog portu A1'e bağlı AC akım sensörü.

Bu sensörlerin tüm okumaları, Arduino kartının dijital D0 (RX pin) ve D1 (TX pin) pinine bağlanan XBee modülüne aktarılmaktadır. Aynı zamanda, sensörlerden okunan tüm değerler, SDA pini Arduino kartının D2 pinine ve SCL pini D3 pinine bağlı 20x4 LCD üzerinde gösterilmektedir.

Kontrol işlemi, ana kontaktörü kontrol etmek için kullanılan Röleler tarafından gerçekleştirilmektedir. Arduino kartının dijital portu D5'den 0 volt uygulanarak röle etkinleştirilebilir. Ayrıca aynı dijital porttan 5 volt uygulanarak devre dışı bırakılabilir. Röleleri etkinleştirip devre dışı bırakma ile ilgili özel komutlar XBee modülü üzerinden RF kullanılarak ana düğümden alınmaktadır. Böylece kullanıcı isterse sistemi uzaktan kontrol edebilmektedir. Arduino Leonardo kartına yüklenen kod, voltaj ve akım sensörleri tarafından algılanan gerçek fiziksel okuma değerlerini temsil etmek için farklı analog girişlerden (A0 ve A1) gelen elektrik sinyallerini dönüştürmek için uygun talimatları içerir ve son olarak bunları XBee modülünü kullanarak RF sinyalleri aracılığıyla dizi biçiminde ana düğüme transfer eder. Dize formu, voltaj (3 haneli tam sayı) ve akım (3 haneli gerçel sayı) bilgilerini içerir. Şekil 4.32, Düğüm I için örnek kodu ve Şekil 4.33 ise akış şemasını göstermektedir. Bunun yanı sıra çalışma kapsamında geliştirilen yazılımlara ait kodların tümü Ekler bölümünde sunulmaktadır.

```
node1L
lcd.print("ON ");
}

else if(l=='d')
{
digitalWrite(5,LOW);
digitalWrite(6,LOW);
// c="M";
lcd.setCursor(0,3);
lcd.print("power status=");
lcd.print("Off");
}

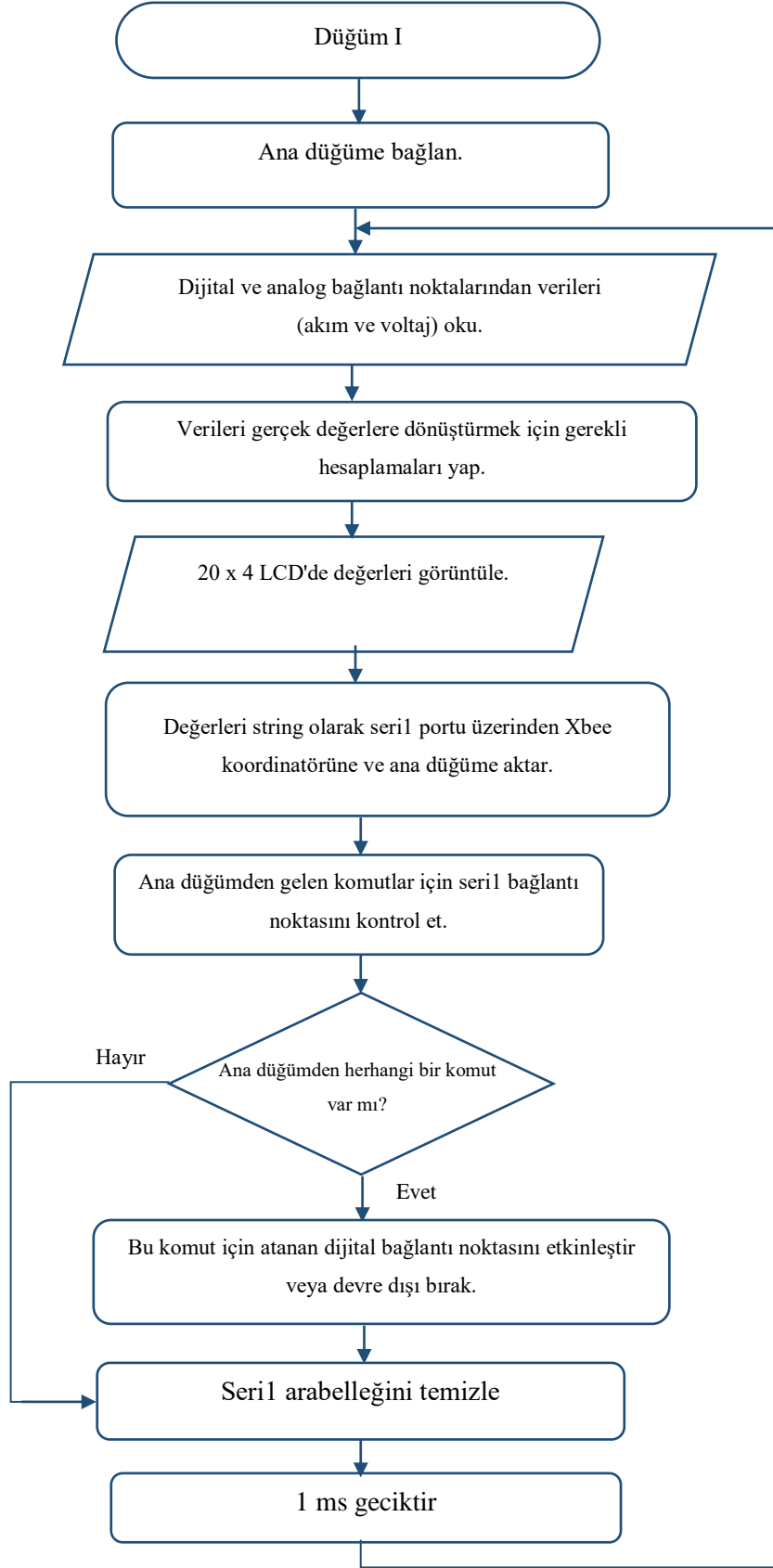
delay(15);

ok=false;
String j=String(a)+r+String(b)+r ;

String c = address+String(j);
char send_to_master[c.length() + 1];
c.toCharArray(send_to_master, c.length() + 1);

Serial1.write(send_to_master, sizeof(send_to_master));
}
}
```

Şekil 4.32 Voltaj sensöründen veri okumak için örnek kod



Şekil 4.33 Düğüm-I için akış şeması.

Düğüm-II için program kodu:

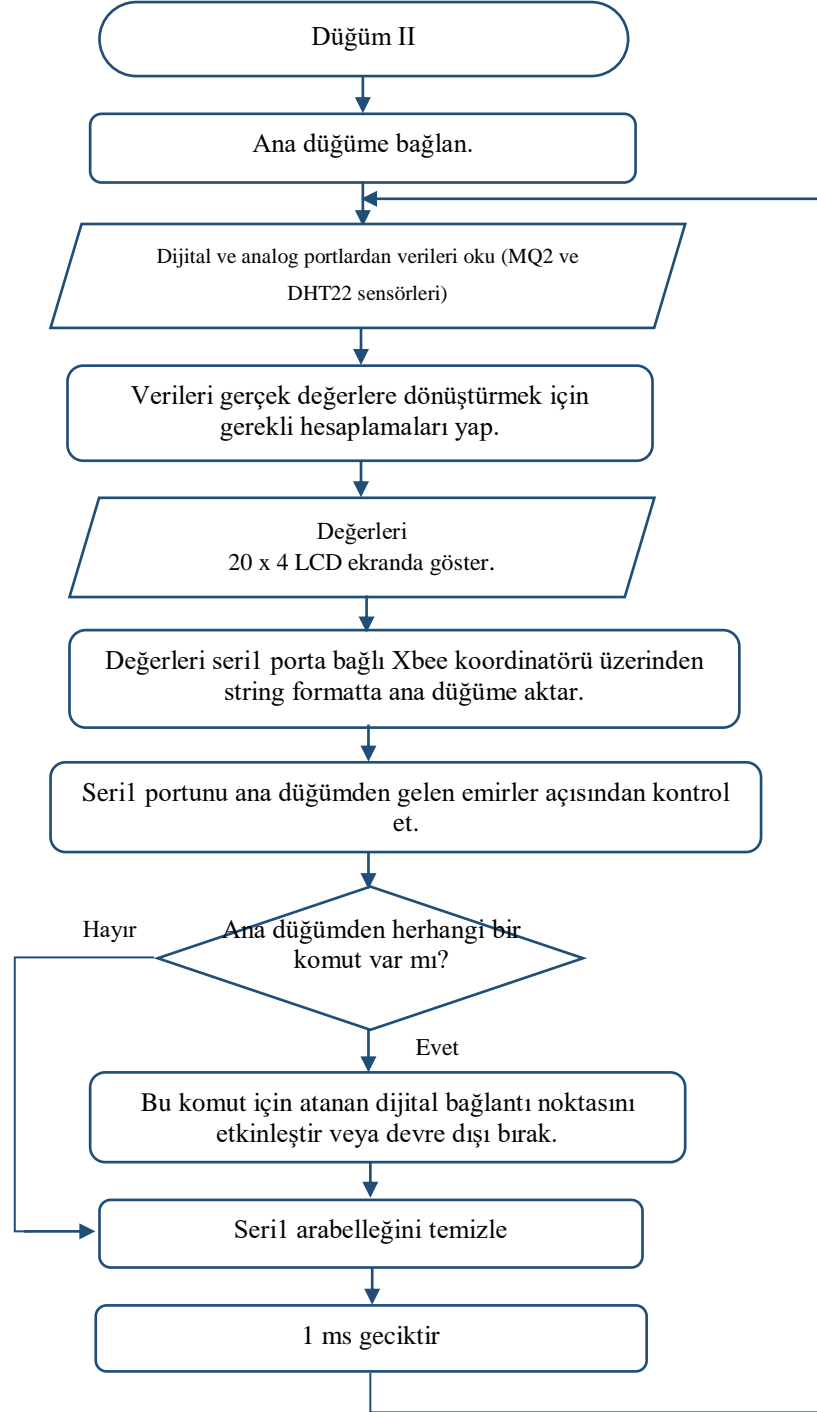
Düğüm-II; duman, gaz kaçağı, sıcaklık ve nem seviyelerini ölçmek ve ısıtıcı ve klimayı uzaktan açık veya kapalı konuma getirmek için tasarlanmıştır. Bu düğüm, iklim kontrol düğümü olacak şekilde organize edilmiştir. Duman ve gaz sızıntısını izlemek için MQ2 sensörü kullanılmıştır. Arduino kartındaki A0 pini analog giriş olarak yapılandırılarak, Arduino kartının A0 pini MQ2 sensör modülünün "in" pinine bağlanmıştır. Ayrıca DH22 sıcaklık ve nem sensörünün "in" pini Arduino kartının dijital D7 pinine bağlanmıştır.

Bu düğümdeki kontrol işlemi, ısıtıcıyı ve klimayı kontrol eden iki röle tarafından yerine getirilmektedir. Isıtıcıyı kontrol etmek için kullanılan röle dijital pin D6'ya, klimayı kontrol etmek için kullanılan röle ise dijital pin D5'e bağlanmıştır. Ayrıca, Arduino kartındaki D2 ve D3 pinlerine bağlanan 20x4 LCD'de gaz, duman, sıcaklık ve nem değerleri gösterilmektedir. Farklı portlardan toplanan tüm veriler, dijital pin D0 ve D1'e seri olarak bağlanan XBee modülü kullanılarak RF üzerinden gönderilmektedir. XBee modülünün Dout pini Arduino kartının D0 "RX" pinine ve XBee modülünün Din pini Arduino kartının D1 "TX" pinine bağlanmıştır.

Arduino kartına yüklenen kod, verileri analog pin A0'dan okumak, gaz ve duman seviyelerini hesaplamak için uygun talimatları içerir. Ayrıca, DH22 sıcaklık ve nem sensörü tarafından algılanan gerçek fiziksel okuma değerleri dijital pin D7'den alınır. Sensörlerden alınan değerler, XBee modülü üzerinden RF sinyalleri aracılığıyla dizi biçiminde ana düğüme aktarılır. Dizi formu şu bilgileri içerir: duman "4 haneli tam sayı" / LPG Kaçağı "4 haneli tam sayı" / Nem "3 haneli tam sayı" / sıcaklık "3 haneli gerçel sayı". Ayrıca, ısıtıcı ve klimayı kontrol eden iki röleyi etkinleştirmek/devre dışı bırakmak için XBee ağı üzerinden alınan harici uzaktan komutları işleyecek şekilde yapılandırılan kod da sisteme eklenmiştir. Şekil 4.34, MQ2 sensöründen gaz ve dumanı hesaplamak için örnek kodu göstermektedir. Şekil 4.35 ise Düğüm-II için akış şemasını göstermektedir.

```
File Edit Sketch Tools Help
node2L
{
  //MQ2 Sensor read//////////
  float values= mq2.read(true); //set it false if you don't want to print the values in the Serial
  //lpg = values[0];
  lpg = mq2.readLPG();
  //co = values[1];
  co = mq2.readCO();
  //smoke = values[2];
  smoke = mq2.readSmoke();
  // Print LPG , CO and Smoke sensor values to LCD monitor
  lcd.setCursor(0,1);
  lcd.print("LPG:");
  lcd.print(lpg);
  lcd.print(" CO:");
  lcd.print(co);
  lcd.setCursor(0,2);
  lcd.print("SMOKE: ");
  lcd.print(smoke);
  lcd.print(" PPM");
  // Print LPG , CO and Smoke sensor values to serial monitor
  Serial.print("LPG: ");
  Serial.println(lpg);
  Serial.print(" CO: ");
  Serial.println(co);
  Serial.print("SMOKE: ");
  Serial.print(smoke);
  Serial.println(" PPM");
  delay(100);
}
```

Şekil 4.34 MQ2 sensöründen gaz ve dumanı hesaplamak için örnek kod

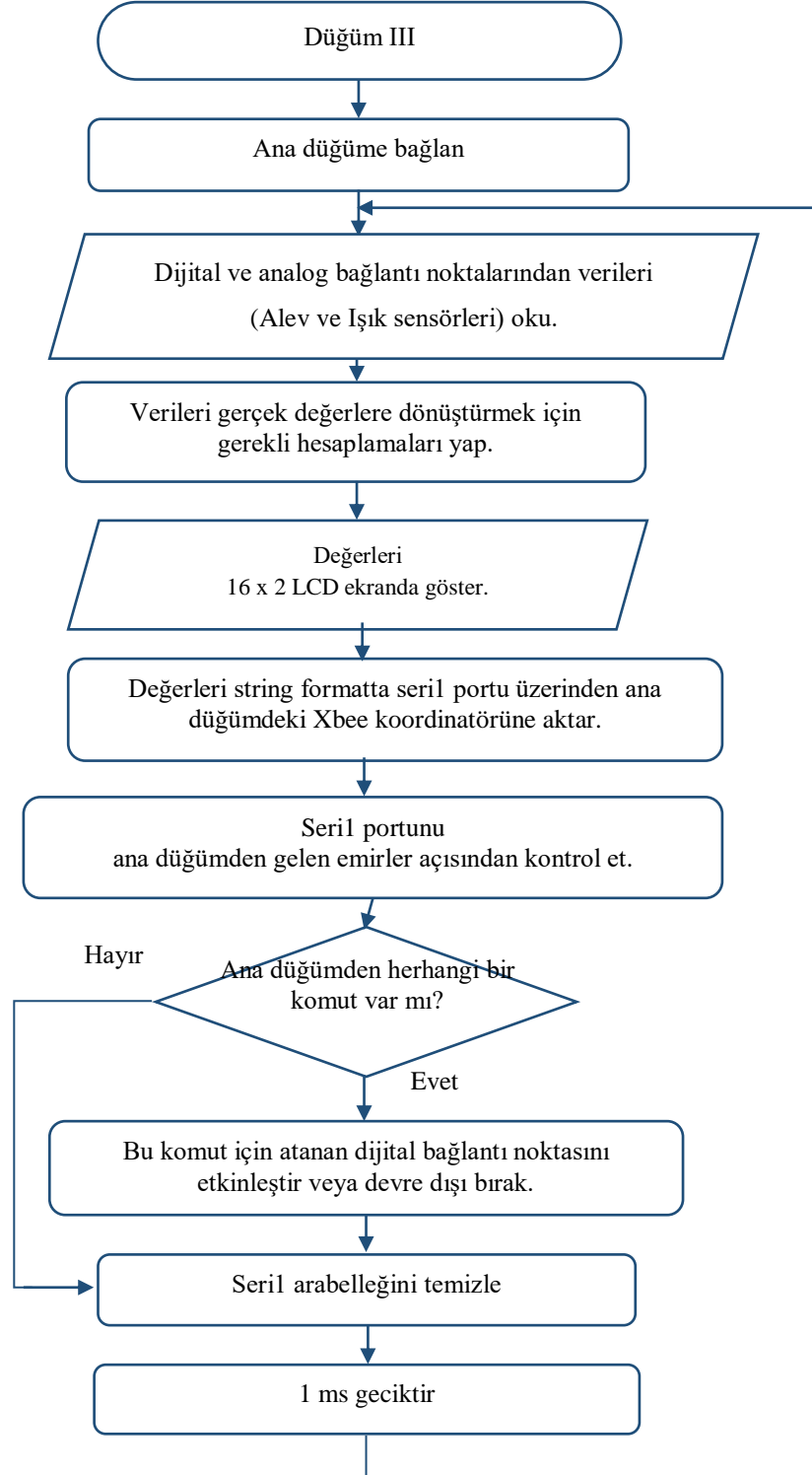


Şekil 4.35 Düğüm-II Akış şeması.

Düğüm-III için program kodu:

Bu düğüm için geliştirilen kod, alev sensörü için atanan A0 pininden gelen verileri izler ve değerlendirir. Ayrıca bahçe ışıklarını açık veya kapalı konuma getirmek için ışık sensörünün bağlı olduğu A1 pininden gelen verileri okur ve değerlendirir. Ek

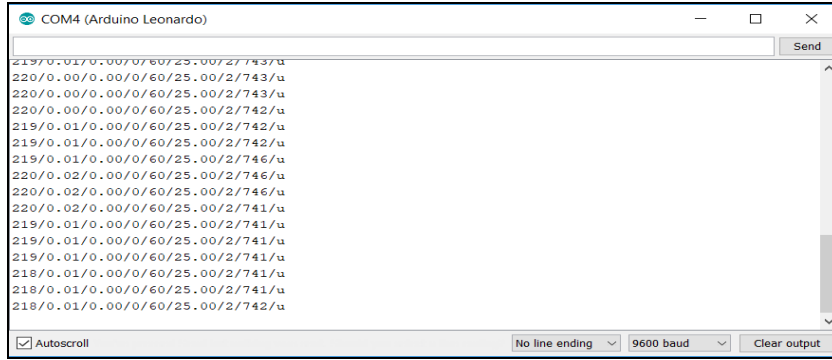
olarak, D5 ve D6 dijital pinlerine baęlı rleler ile buzdolabı ve televizyon kontrol edilmektedir. Bu cihazları kontrol etmek iin XBee aęı zerinden alınan harici uzaktan komutlar kullanılmaktadır. Őekil 4.36, Dęm III programının akıř Őemasını gsterir.



Őekil 4.36 Dęm-III akıř Őeması.

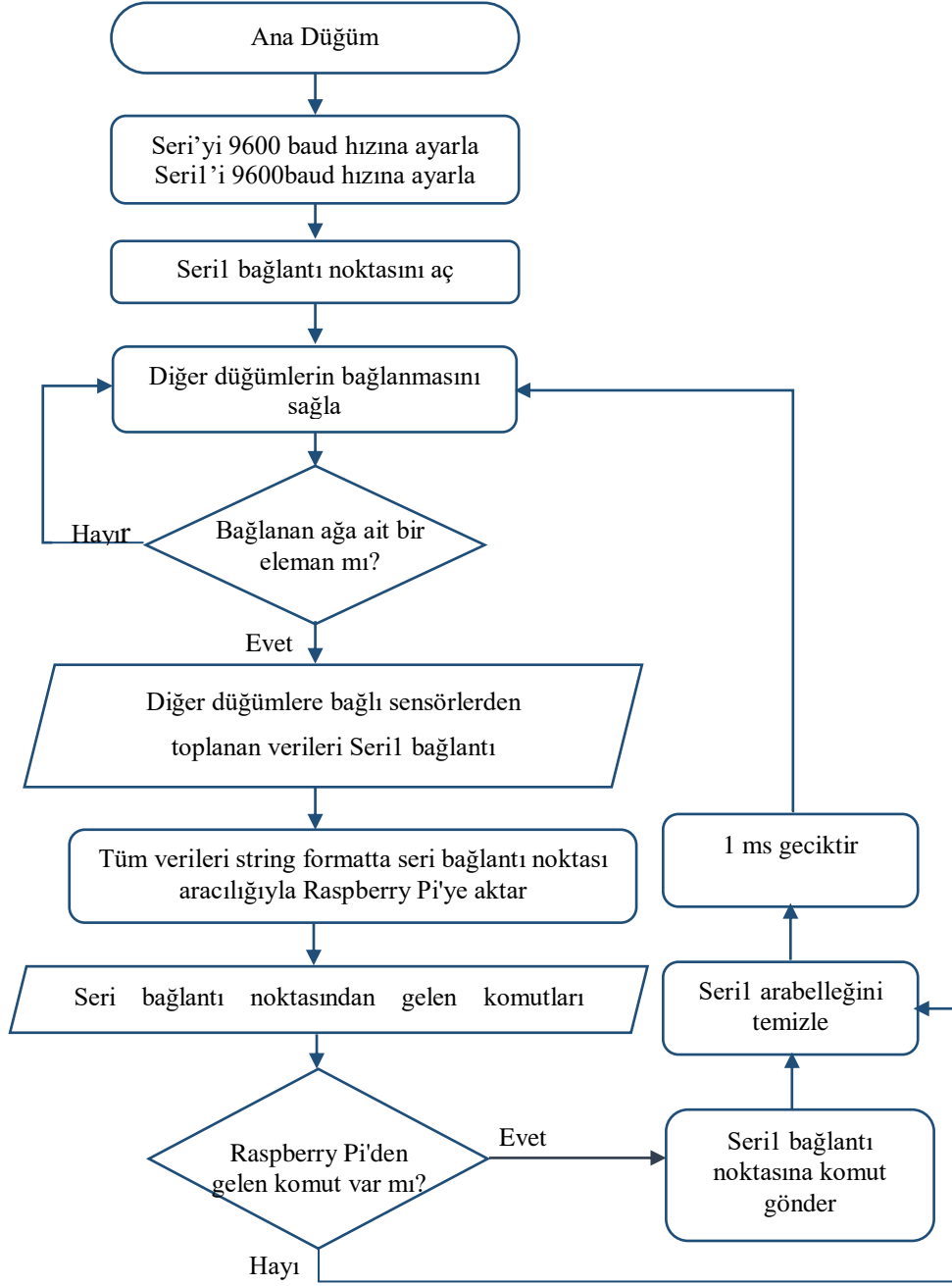
Ana Dügüm için program kodu

Bu düğümdeki Arduino Leonardo kartı, diğer düğümlere bağlı farklı sensörlerden toplanan string verileri XBee düğümlerinden almak ve alınan bu verileri seri port vasıtasıyla Raspberry Pi'ye aktarmak için programlanmıştır. String veri formu şu şekildedir: Ev voltajı "3 haneli tamsayı", Ev akımı "3 haneli gerçel sayı", Duman "4 haneli tam sayı", LPG kaçak "4 haneli tam sayı", Nem "3 haneli tamsayı", Sıcaklık "3 haneli gerçel sayı", Yangın "1 haneli tam sayı", Işık "3 haneli tam sayı". Ayrıca, program, ana XBee düğümü ile her düğüm arasındaki bağlantıyı doğrulamak için bir kod içerir. Şekil 4.37, ana Arduino kartından Raspberry Pi'ye gönderilen veri formatını göstermektedir.



```
COM4 (Arduino Leonardo)
219/0.017/0.00/0/60/25.00/2/743/u
220/0.00/0.00/0/60/25.00/2/743/u
220/0.00/0.00/0/60/25.00/2/743/u
220/0.00/0.00/0/60/25.00/2/742/u
219/0.01/0.00/0/60/25.00/2/742/u
219/0.01/0.00/0/60/25.00/2/742/u
219/0.01/0.00/0/60/25.00/2/746/u
220/0.02/0.00/0/60/25.00/2/746/u
220/0.02/0.00/0/60/25.00/2/746/u
220/0.02/0.00/0/60/25.00/2/741/u
219/0.01/0.00/0/60/25.00/2/741/u
219/0.01/0.00/0/60/25.00/2/741/u
219/0.01/0.00/0/60/25.00/2/741/u
218/0.01/0.00/0/60/25.00/2/741/u
218/0.01/0.00/0/60/25.00/2/741/u
218/0.01/0.00/0/60/25.00/2/742/u
```

Şekil 4.37 Ana Arduino kartından Raspberry Pi'ye gönderilen veri formatı



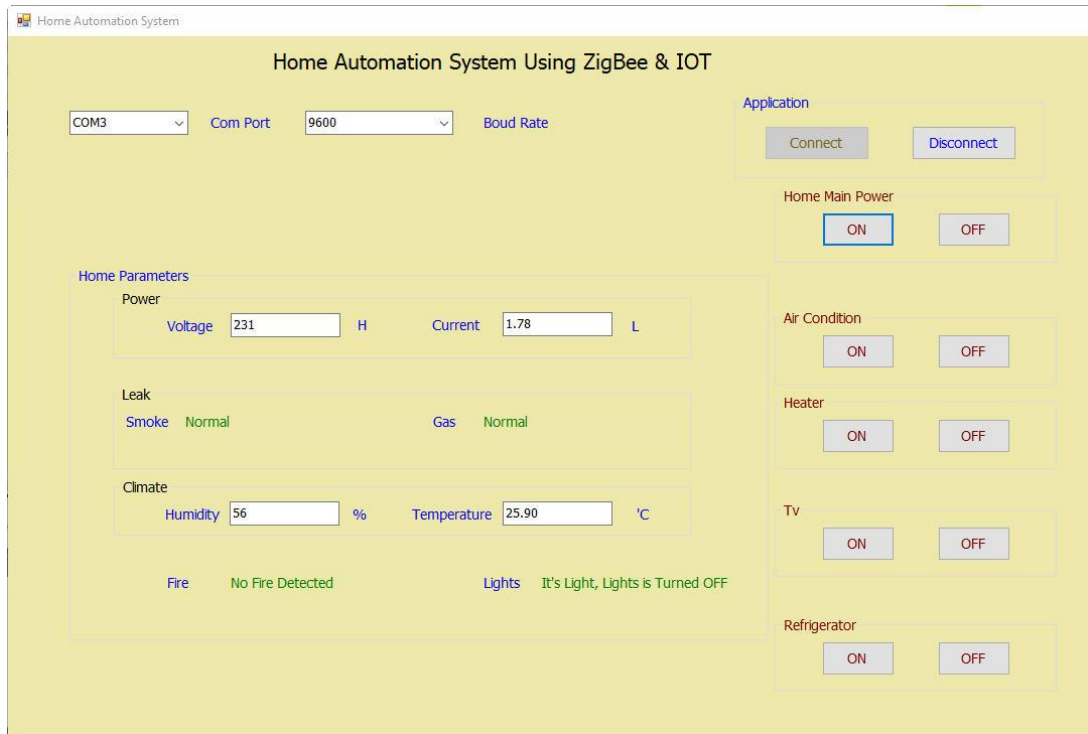
Şekil 4.38 Ana düğüme ait programının akış şemasını göstermektedir.

4.7.2 Yüksek Seviyeli Program

Yüksek seviyeli program, sistemi farklı cihazlardan gerçek zamanlı olarak izlemek ve kontrol etmek için kullanılan kullanıcı arayüzüdür. Daha önce de bahsedildiği gibi, düğümlerden toplanan tüm veriler, string veri formunda COM portu üzerinden Raspberry Pi'ye gönderilmektedir. Grafik kullanıcı arayüzünü geliştirmek için

ücretsiz bir açık kaynaklı yazılım olan Visual Studio 2019 topluluğu (*Visual Studio 2019 Community free download*, n.d.) "Visual Basic. Net" kullanılmıştır.

Çalışmada iki farklı grafiksel kullanıcı arayüzü (GUI) sistemi geliştirilmiştir. İlk GUI sistemi, Raspberry Pi üzerinde sunucu olarak çalışmaktadır. Bu sistem doğrudan ana Arduino kartına bağlanarak tasarlanan tüm sisteme ana hizmetleri sağlamaktadır. İkinci GUI sistemi ise uzak bilgisayarlara kurulmak üzere geliştirilmiştir. Gerçek zamanlı uzaktan izleme ve uzaktan kontrol için kullanılmaktadır. Şekil 4.39, Raspberry Pi için tasarlanmış GUI'yi göstermektedir.



Şekil 4.39 Raspberry Pi için Tasarlanmış GUI

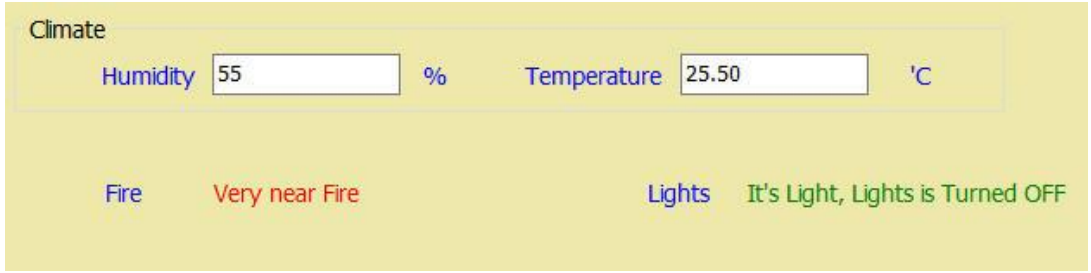
Raspberry Pi için tasarlanan GUI sistemi aşağıdaki görevleri gerçekleştirmektedir:

1. Arduino Leonardo'dan gönderilen bir dizi string bilgiyi almak için seri portu yapılandırma.
2. Seri port üzerinden alınan ve sensörlerden gelen verilerimizi içeren veri dizisini, GUI sistemimizdeki belirlenmiş metin kutularında gösterilecek bilgilere göre parçalara ayırmak.

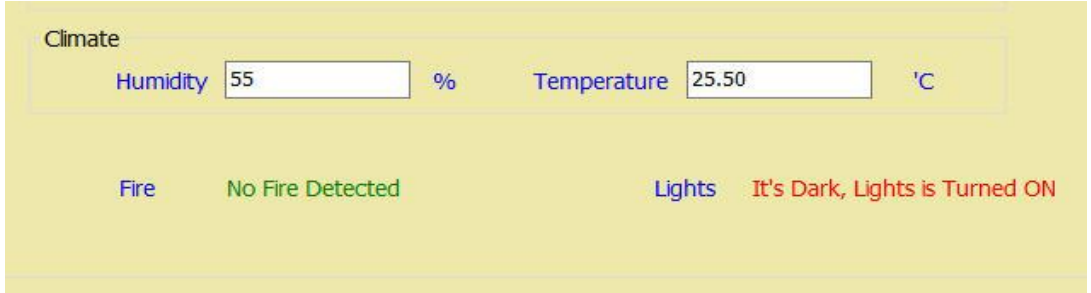
3. Tüm ev parametrelerini temsil eden parçalara ayrılmış bu verileri form üzerinde göstermek.
4. Sensörlerden gelen verilerin normal değerlerde olup olmadığını değerlendirerek değerlendirme sonucunu dilsel ifade olarak kullanıcıya sunmak. Şekil 4.40, 4.41 ve 4.42 sistemin normal ve normal olmayan değerlerle nasıl başa çıktığını göstermektedir.



Şekil 4.40 Duman ve gaz değerlerinin normal olduğunun bildirilmesi.



Şekil 4.41 Yangın algılanması ve dilsel ifade olarak bildirilmesi.



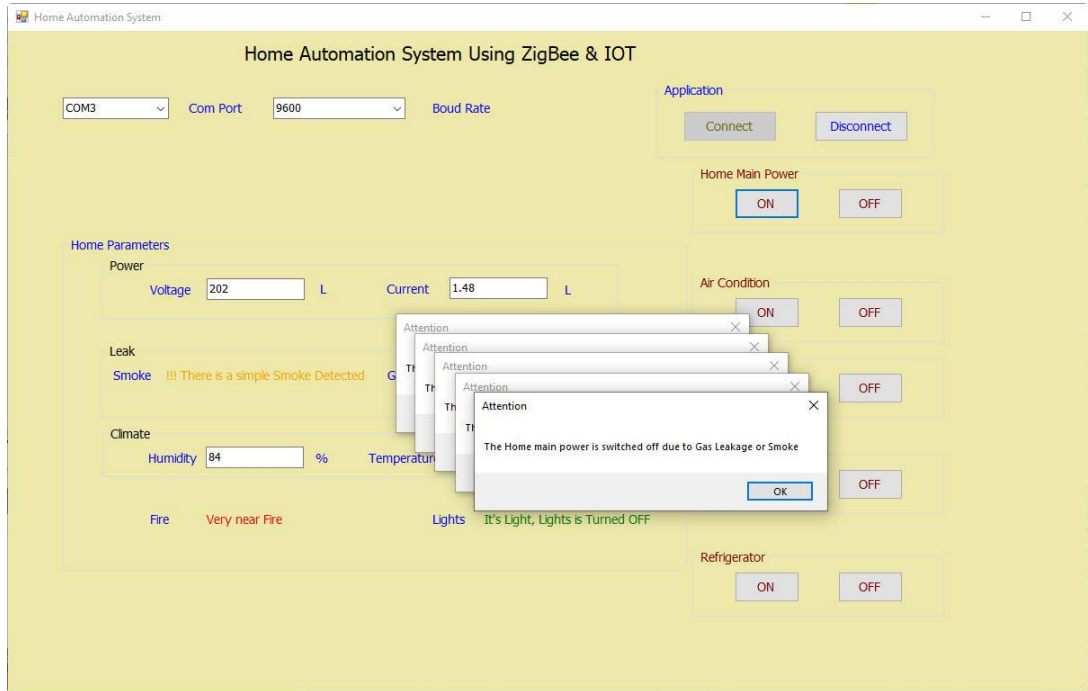
Şekil 4.42 Işık düzeyinin algılanması ve dilsel ifade olarak bildirilmesi.

5. Kullanıcıların uzaktan komutlar göndererek aşağıdaki birimleri açmasına veya kapatmasına izin vermek.
 - Evin ana gücü,
 - Klima,

- Isıtıcı,
- TV,
- Buzdolabı.

6. Sistemi buluta bağlayarak IOT uygulamasıyla çalışacak şekilde yapılandırmak. Böylece veriler bulut üzerinde yayınlanabilir veya indirilebilir.

7. Gaz kaçağı, yüksek düzeyde sigara ve yangına maruz kalma gibi durumlarda evin ana gücünün otomatik olarak kapatılması. Şekil 4.43, yüksek seviyede duman veya yangın algılandığında sistemin otomatik olarak kapalı konuma geçme durumunu göstermektedir.



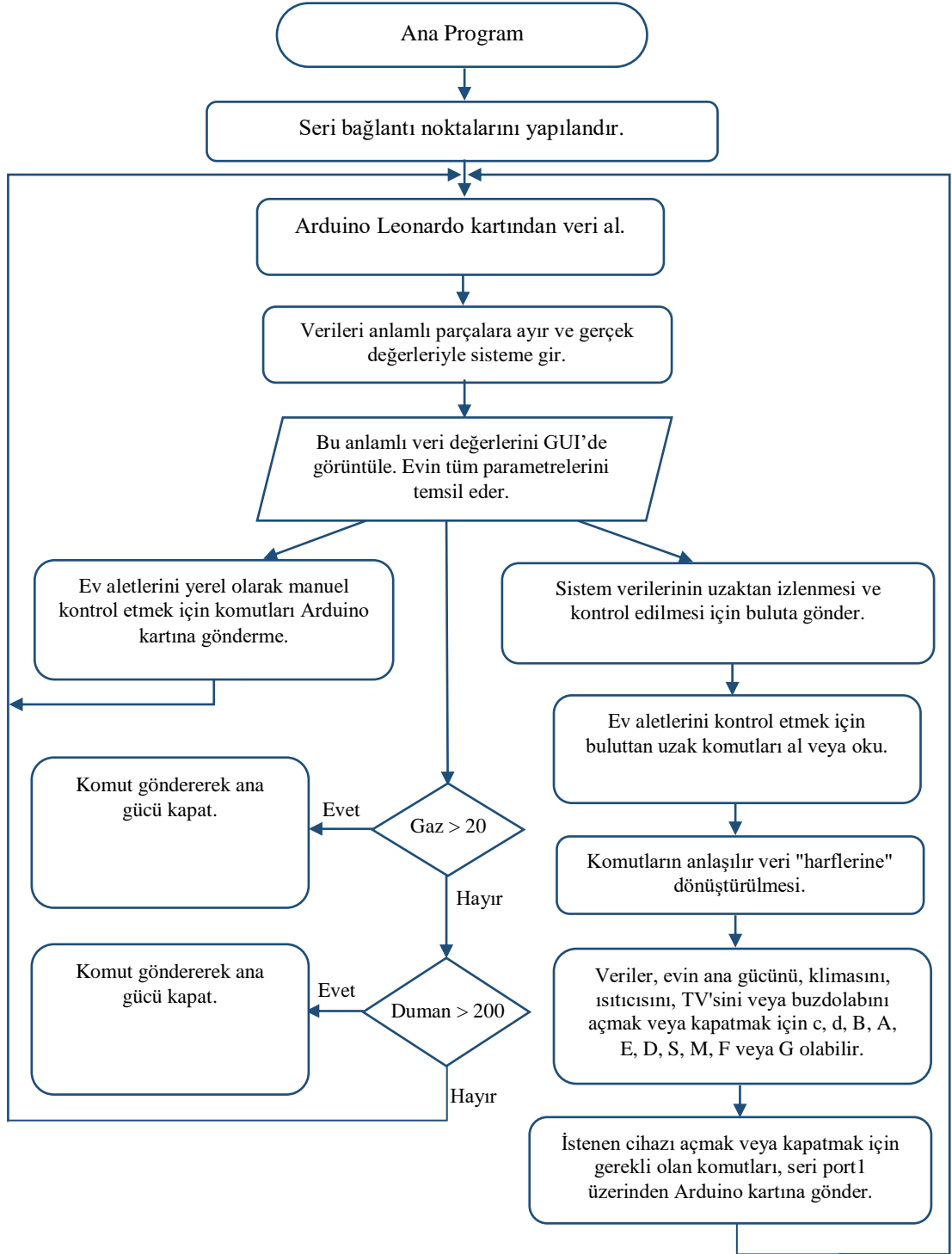
Şekil 4.43 Yüksek seviyede duman veya yangın algılandığında evin ana gücünün otomatik olarak kapatılması.

Şekil 4.44, ana programın akış şemasını göstermektedir.

Bu akış şemasının algoritması aşağıdaki gibidir:

- Seri bağlantı noktasını seçili bağlantı noktası için yapılandır.

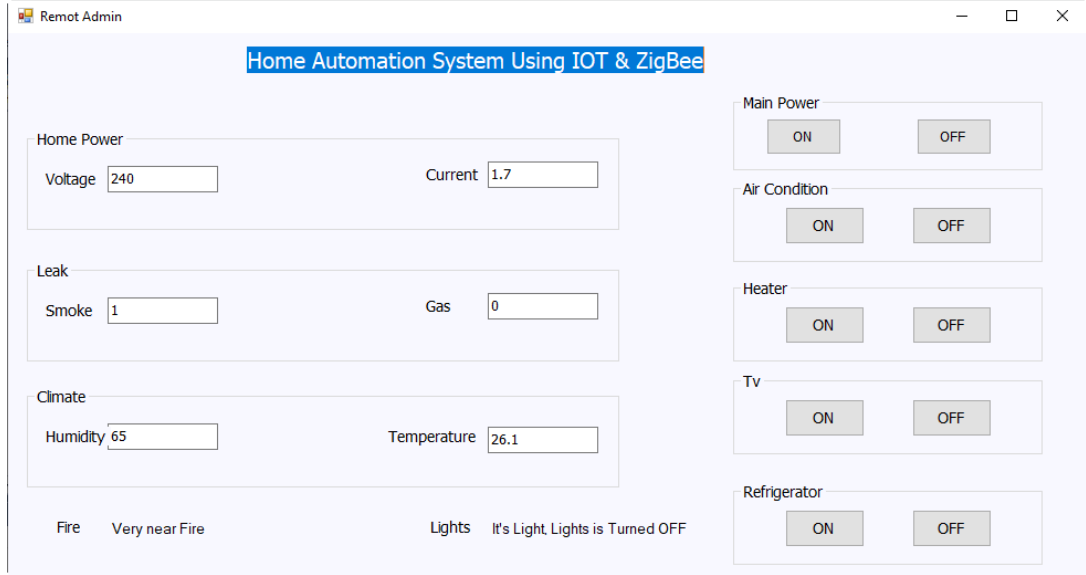
- Arduino kartından string formatta veri al.
- String veri dizisindeki verileri anlamlı verilere parçala ve bu verileri gerçek değerlere dönüştür.
- Verileri GUI'de ev aletlerinin değerleri olarak görüntüle. Sistem bu işlemleri aynı anda yapabilir.
- 20 ppm'den fazla gaz kaçağı veya 200 ppm'den fazla duman varsa evin ana gücünü kapatma komutunu Arduino kartına gönder.
- Ev aletlerini yerel olarak kontrol etmek için komutları Arduino kartına gönder.
- Uzaktan izleme ve kontrol için verileri buluta gönder ve buluttan komutları al ve ardından evdeki herhangi bir cihazı kapatmak veya açmak için Arduino kartına komut gönder.



Şekil 4.44 Raspberry Pi için tasarlanmış programın akış şeması.

Şekil 4.45, uzak bilgisayar için tasarlanmış ikinci GUI'yi göstermektedir. Bu program ev otomasyon sistemini dünyanın her yerinden izlemek ve kontrol etmek için tasarlanmıştır. ThingSpeak bulutu üzerindeki kanallarımızla bağlantılı olarak çalışmaktadır. Program metin kutuları ve butonların yer aldığı iki bölümden

oluşmaktadır. Metin kutularının yer aldığı bölümde sensörlerden gelen veriler gösterilmektedir.



Şekil 4.45 Uzak PC için Tasarlanmış GUI

Program, ThinkSpeak üzerinden verileri çekip metin kutularında göstermek için kanala ait olan API Okuma anahtarını kullanmaktadır. Şekil 4.46, verilerin çekilmesi ve metin kutularında gösterilmesine ait örnek kodu göstermektedir.

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    Dim webClient As WebClient = New WebClient()
    Dim data = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field1/last")
    Ph1V.Text = data
    Dim data1 = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field2/last")
    Ph1C.Text = data1
    Dim data2 = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field3/last")
    Smoke.Text = data2
    Dim data3 = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field4/last")
    Gas.Text = data3
    Dim data4 = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field5/last")
    Humidity.Text = data4
    Dim data5 = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field6/last")
    Temperature.Text = data5
    Dim data6 = webClient.DownloadString("https://api.thingspeak.com/channels/884057/fields/field7/last")
    Temp.Text = data6
    If data6 = 0 Then
        Label19.Text = "Very near Fire"
    End If
End Sub
```

Şekil 4.46 Sensör verilerinin gösterilmesi için kullanılan örnek kod.

Programda butonların bulunduğu bölümden, ev aletlerinin her biri kontrol edilebilmektedir. Cihaz kontrolü için 1 ve 0 mantığı kullanılır. Şekil 4.47, düğmeler bölümünden Klima düğmesine ait örnek kodu göstermektedir.

```

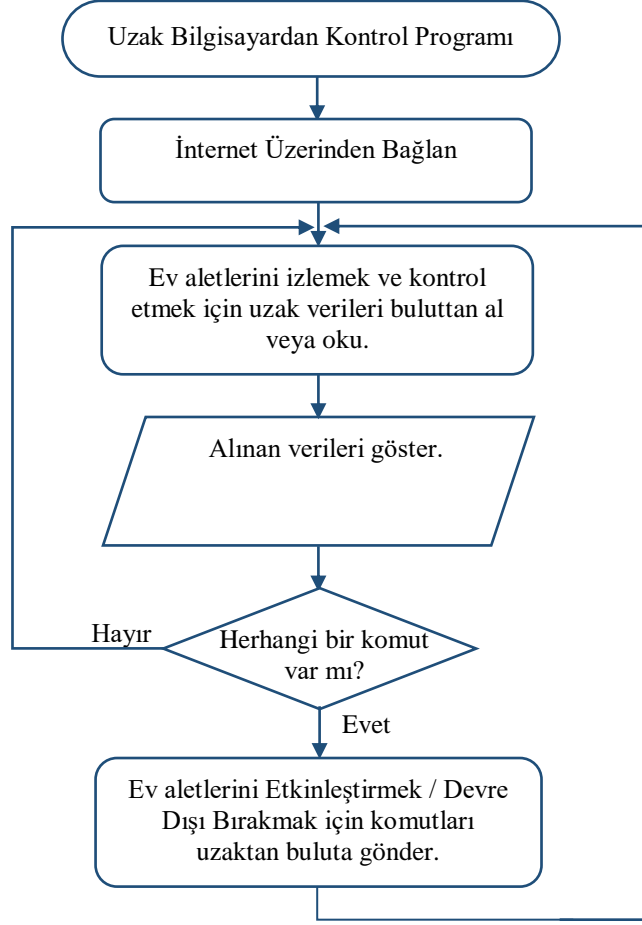
Private Sub Airon_Click(sender As Object, e As EventArgs) Handles Airon.Click
    Airon.Enabled = False           'enable Connect button
    Airoff.Enabled = True
    Dim Request1 As WebRequest = WebRequest.Create("https://api.thingspeak.com/update?api_key=M5BXHCCVODCD6S36&field2=1")
    'set the method property of the request to post
    Request1.Method = "POST"
    'creat post data And convert it to byte Array
    Dim postdata1 As String = "55"
    Dim byteArray1 As Byte() = Encoding.UTF8.GetBytes(postdata)
    'set the content type property of the webrequest
    Request1.ContentType = "application/x-www-form-urlencoded"
    'set the content length property of webrequest
    Request1.ContentLength = byteArray1.Length
    'get the request stream
    Dim datastream1 As Stream = Request1.GetRequestStream()
    ' write the data to the request stream
    datastream1.Write(byteArray1, 0, byteArray1.Length)
    'close the stream object
    datastream1.Close()

```

Şekil 4.47 Klima butonunun butonlar bölümünden yapılandırılması.

Daha önce de belirtildiği üzere uzak bilgisayar için kullanılan geliştirilmiş GUI, doğrudan Raspberry Pi ile bağlantılı değildir, ThingSpeak bulutu üzerinden ev IOT sistemine bağlanmaktadır. Şekil 4.48, bu programa ait akış şemasını göstermektedir. Bu akış şemasının algoritması aşağıdaki gibidir:

- İnternet üzerinden bağlantı kur.
- Verileri buluttan al.
- Verileri ev aletlerinin değerleri olarak görüntüle.
- Verileri izle ve kontrol et.
- Evdeki cihazları açmak veya kapatmak için komutları uzak bilgisayardan buluta gönder.



Şekil 4.48 Uzak bilgisayardan kontrol programının akış şeması.

4.8 IOT Sisteminde Kullanılan Bulut Hizmeti

Çalışmada, ücretsiz bir web bulut hizmeti olan ve tasarımcıların sensör verilerini bulutta toplayıp depolamasına ve IoT uygulamaları oluşturmasına olanak sağlayan Thing Speak kullanılmıştır. Ayrıca, bu web bulut hizmeti, tasarımcıların verilerini analiz etmelerine, görselleştirmelerine ve ardından veriler üzerinde işlem yapmalarına olanak tanıyan uygulamalar sağlar. Thing Speak, Arduino, Raspberry Pi, BeagleBone Black, bilgisayar ve diğer donanımlardan gönderilen sensör verilerini işleyebilir (*Started with Thing speak*, 2019). Thing Speak ile kanal oluşturarak IoT uygulaması sensör verileri izlenebilir veya sensörlere komut gönderilebilir. Her kanal, maksimum sekiz sensör verisi için alan sunar. Her alan tek bir sensör verisi için özelleştirilir. Think Speak, ücretsiz hesapta 3 milyona kadar mesaj yayınlamaya izin verir. Sınır aşıldığında bulut hizmetleri için ödeme yapılması gerekir.

Thing speak cloud'u kullanmak için <https://www.mathworks.com> adresinde hesap oluşturulması gerekir. Kanal oluşturduktan sonra, alanları kanalınıza ya da kanalınızdan veri yayınlamak ve okumak için kullanmak üzere aşağıdaki bilgilere sahip olacaksınız:

Kanalı oluşturduktan sonra, kanaldaki alanlarda veri yayınlamak veya okumak için aşağıdaki bilgiler gerekecektir:



- **Kanal Kimliği (Channel ID):** Thing speak kanalınız üzerindeki alanlarda işlem yapmak için Thinkspeak.com'dan sizin için atanan benzersiz bir numaradır.

- **Veri Okuma API Anahtarı (Read API key):** Önünde kanal kimliği ve ardında alan numarası bulunan bu anahtar, kanalınızın Thing speak alanlarından veri okumak için kullanılır.

- **Veri Yazma API Anahtarı (Write API Key):** Önünde kanal kimliği ve ardında alan numarası bulunan bu anahtar, sensörden kanalınızın Thing speak alanlarına veri yazmak için kullanılır.

- **MQTT API Anahtarı:** Önünde kanal kimliği ve ardında alan numarası bulunan bu anahtar, kanalınızın Thing speak alanlarına/alanlarından veri yayınlamak/abone olmak için kullanılır. Ayrıca kanalınızı herkesle paylaşılacak, seçilen kullanıcılarla paylaşılacak veya "özel görünümü" paylaşmayacak şekilde yapılandırabilirsiniz.

Çalışma kapsamında geliştirilen IOT sistemi için Thing'de iki ücretsiz kanal oluşturulmuştur. Bu kanallar, kanal kimliği: 884057 ("Ev Otomasyon Sistemi" olarak adlandırılmıştır ve sekiz alan içerir) ve 884075 ("Cihaz Durumu" olarak adlandırılmıştır ve beş alan içerir) şeklindedir. Şekil 4.49, Thing Speak'de oluşturulan kanalları göstermektedir.

Name	Created	Updated
 Home Automation System <small>home automation system, iot, xbee, zigbee</small> Private Public Settings Sharing API Keys Data Import / Export	2019-10-13	2019-12-29 10:00
 Appliances Status <small>home automation system, mqtt, iot, xbee, zigbee</small> Private Public Settings Sharing API Keys Data Import / Export	2019-10-13	2019-12-29 10:03

Şekil 4.49 Thing Speak'de oluşturulan kanallar

4.8.1 Ev Otomasyon Sistemi kanalı

Bu kanal, ev parametrelerini gerçek zamanlı olarak göndermek, kaydetmek ve izlemek için oluşturulmuştur. Bu kanalda sekiz alan etkinleştirilmiştir. Bu alanlar aşağıdaki gibidir:

- Alan 1, Eve beslenen voltaj için atanmıştır.
- Alan 2, Eve beslenen akım için atanmıştır.
- Alan 3, Evdeki duman seviyesi için atanmıştır.
- Alan 4, Evdeki gaz kaçağı için atanmıştır.
- Alan 5, Evdeki nem için atanmıştır.
- Alan 6, Ev sıcaklığı için atanmıştır.
- Alan 7, Evdeki yangın durumu için atanmıştır.
- Alan 8, Evin bahçesindeki ışık durumu için atanmıştır.

Şekil 4.50'de bu alanlardaki verilerin grafiksel olarak sunumu gösterilmektedir.



Şekil 4.50 Thing Speak'de oluşturulan kanalda verilerin sunumu.

IoT sisteminde, evin ana gücü, klima, ısıtıcı, TV ve buzdolabını kapalı veya açık duruma getirmek için atanan ikinci kanal beş alana sahiptir. Ayrıca bu kanal da bu cihazların durumları da Şekil 4.51'deki gibi gösterilmektedir.



Şekil 4.51 Thing Speak'da oluşturulan araç durum kanalında sunulan veriler.

4.9 Tasarlanan Android Uygulaması

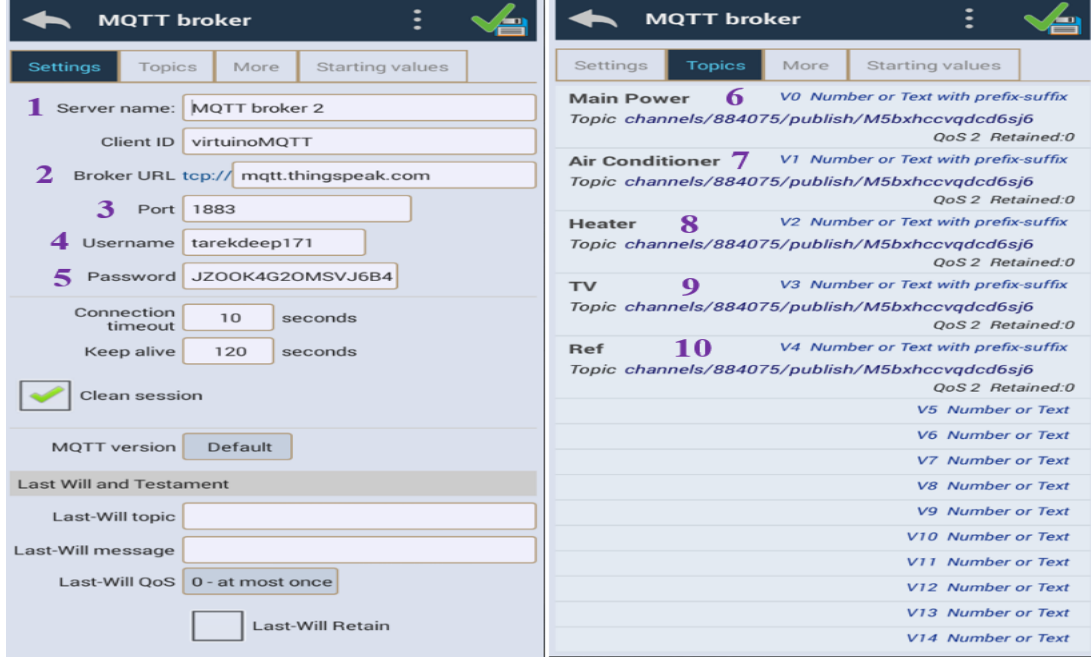
Ev aletlerini dünyanın her yerinden izlemek ve kontrol etmek için Android platform cihazlarında kullanılacak bir Android Uygulaması tasarlanmıştır. Bu uygulamayı oluşturmak için VirtuinoMQTT Application Basic yardımcı yazılımı kullanılmıştır. Bu uygulama, sistemdeki MQTT aracısı olan Thing Speak tarafından sağlanan MQTT anahtarı aracılığıyla Thing Speak'deki kanala abone olmak (Oku) veya

verileri yayınlamak (Yazmak) için MQTT protokolünü kullanır. Thing Speak'deki MQTT aracısındaki kanalda veri yayınlamak için, veri formu “topics channels/channel number/publish/channel password” biçiminde olmalıdır. Kanaldaki verilere abone olmak için ise, veri formu “sub-topic channels/channel number/subscribe/fields/field number” biçiminde olmalıdır. Şekil 4.52, Android Platform cihazları için tasarlanan uygulamayı göstermektedir.



Şekil 4.52 Android Uygulama

Thing Speak, MQTT aracısından alınan verileri işlemek için Virtuino MQTT Uygulamasını yapılandırma ayarları Şekil 4.53’de gösterilmektedir. Thing Speak MQTT aracısından gelen verileri işlemek için Virtuoso MQTT uygulamasının ayarlarının nasıl yapılandırılacağı Şekil 4.53’de gösterilmektedir.



Şekil 4.53 MQTT aracısını ayarlama ve yapılandırma süreci

4.10 Deneysel Sonuçlar ve Tartışma

Çalışma kapsamında geliştirilen sistem ve yazılımlar gerçek zamanlı olarak test edilmiştir. Hem uzak bilgisayar için hem de android için geliştirilen uygulamanın ev aletlerini uzaktan kontrol etme ve verileri kullanıcıya sunma açısından başarılı olduğu gözlenmiştir. Bu nedenle sistem, ev güvenliğini izlemek, yangınlardan korunmak ve ev aletlerini uzaktan kontrol etmek için uygundur. Gaz kaçağının 20 ppm'den fazla olması durumunda veya evdeki duman seviyesi 200 ppm'nin üzerine çıktığında evin ana gücü otomatik olarak kapatılabilmekte, ayrıca kullanıcı isterse manuel olarak kapatabilmektedir. Evde yangın çıkması durumunda yangın alarmı devreye girmektedir. Klima, ısıtma sistemi, TV ve buzdolabı hem uzak bilgisayar hem de android için geliştirilen uygulamalar vasıtasıyla kullanıcı tarafından başarılı bir şekilde açılıp kapatılabilmektedir. Bunun yanı sıra, bahçe ışıklarının karanlıkta otomatik olarak başarılı bir şekilde yanması ve gündüzleri sönmesi sağlanmıştır. Bunun yanı sıra, ev otomasyon sisteminin maliyeti düşüktür ve Tablo 4.2'deki gibidir:

Tablo 4.2 Sistemin Maliyeti

Hayır	Ürün Adı	Ürün Fiyatı	Miktar	Toplam
1	Raspberry Pi 3 Model B + Dahili Broadcom 1,4 GHz dört çekirdekli 64 bit işlemci Wifi Bluetooth ve USB Bağlantı Noktası	40\$	1X	40\$
2	Arduino Leonardo v3	5 \$	X4	20\$
3	RF Alıcı-Verici Modülleri WIRE ANT Zigbee / 802.15.4 XBee Radyo frekanslı modülü)	17\$	X4	68\$
4	XBee Pro V03 Shield Modülü Kablosuz Kontrol Genişleme Kartı	2\$	X1	2\$
5	Xbee kalkanı	1,5\$	X3	4,5\$
6	5V tek kanallı Röle	1 \$	X5	5\$
7	5v çift kanallı Röle	1,5\$	X1	1,5\$
8	20x4 LCD Modül Kalkanı IIC I2C LCD2004 Seri LCD Modül Ekranı Mavi Ekran.	3\$	X2	6\$
9	LCD1602 Mavi Yeşil ekran IIC / I2C 1602 16X2 LCD ekran modülü	2\$	X1	2\$
10	Gerilim trafosu sensör modülü tek fazlı AC aktif çıkış ZMPT101B	2\$	X1	2\$
11	SCT-013 0-100A Açık-Kapalı Akım Trafosu SCT013 Non-invaziv AC Akım Sensörü Bölünmüş Çekirdekli Akım Trafosu	6\$	X1	6\$
12	MQ-2 MQ2 Duman Gazı LPG Bütan Hidrojen Gazı Sensör Dedektör Modülü Arduino için	1.5\$	X1	1.5\$
13	Foto direnç modülü KY-018	1\$	X1	1\$
14	Alev sensörü modülü KY-026 Kızılötesi alıcı Alev yangın algılama alarmı.	1\$	X1	1\$
15	Sıcaklık ve Bağıl Nem Sensörü DHT11 Modülü	1\$	X1	1\$
16	SFM-20B DC3-24V Dayanıkl 3-24V Piezo Elektronik Buzzer Alarm 95DB Arduino İçin Sürekli Ses Beeper	0,5\$	X1	0,5\$
17	Bağlantı kablosu seti		X50	10\$
			Toplam Fiyat	172\$

Önerilen sistem literatürdeki önceki çalışmalarla karşılaştırıldığında bu sistemin ve diğer sistemlerin gereği gibi görevlerini yerine getirdikleri görülmektedir. Bununla birlikte, sistemin tasarlanma yöntemi, sistemi kurmak için kullanılan teknolojiler, cihazlar, yazılımlar ve evde hangi cihazların kontrol edildiği noktasında az da olsa bazı farklılıklar vardır. Sistemimizde, verileri toplamak ve sensör ağı oluşturmak için

Zigbee teknolojisi kullanılmıştır. Arduino Leonardo, sistemin merkezi işlem birimi olarak çalışmaktadır. Raspberry Pi gerçek zamanlı kontrol, izleme ve sistemi internete bağlamak için bir ağ geçidi görevi görmektedir. Ek olarak, evdeki akımı, voltajı, sıcaklığı, nemi ve bahçe aydınlatmasını izlemek için kullanılan bir dizi sensör ve ev aletlerini açmak veya kapatmak için kullanılan bazı röleler mevcuttur. Yazılımda, Visual Studio 2019 topluluğu, biri Raspberry Pi için diğeri uzak PC'ler için iki GUI programı oluşturmak amacıyla kullanılmıştır. Ayrıca, Android platformunda çalışan bir uygulama tasarlamak için VirtuinoMQTT kullanılmıştır. Sistemdeki verileri buluta taşımak için ücretsiz web bulut hizmeti olan ThingSpeak kullanılmıştır.

Pavithra ve Balakrishnan (2015) önerdikleri sistemde sistemi programlamak için Python programlama dilini kullanmışlardır ve ayrıca ev aletlerini kontrol etmek ve izlemek için bir web sayfası tasarlamışlardır. Ek olarak, tüm sensörler doğrudan Raspberry Pi'ye bağlamışlardır. Bepery vd. (2019) geliştirdikleri sistemde Raspberry Pi'yi esas olarak bir web sunucusu olarak kullanmışlar ve Arduino Uno için kontrol sinyali oluşturmuşlardır. Raspberry Pi, kullanıcı arayüzünü içeren web sitesini barındırmaktadır. Arduino kartı, cihazların rölelerini Açık veya Kapalı olarak değiştirmektedir. Vinay ve Kusuma (2015), hareket, gaz ve LDR gibi bir dizi sensör kullanan bir sistem geliştirmişlerdir. Intel Galileo anakartı ve WiFi kartı, internete bağlanmak için web sunucusu olarak çalışır ve verileri depolamak için bulut bilişim olarak Gmail'i kullanır. Soliman vd. (2013), iki Arduino Uno kartı kullanan bir sistem geliştirmişlerdir. Bu kartlardan biri merkezi alıcı olarak çalışarak sistemdeki tüm aktüatörlere ve internet üzerindeki veritabanı sunucusuna bağlanır. Diğeri ise verici görevi görür ve tüm sensörler ona bağlıdır. Ayrıca, sensörler/aktüatörler ile merkezi Arduino kartı arasındaki iletişim için ZigBee teknolojisi kullanırlarken, bulut hizmeti olarak Google uygulama motoru platformunu kullanmışlardır. Kumar ve Pati (2016) tarafından önerilen sistemde kameralı Raspberry Pi2 model B, Arduino Mega2560 ve Python dili kullanılmıştır.

5. SONUÇ

Bu çalışmada, ucuz ve kurulumu kolay teknolojiler kullanarak, gerçek zamanlı olarak evi kontrol etmek ve izlemek için ZigBee ağı üzerinden birbirine bağlanan IoT ve akıllı düğümlere dayanan düşük maliyetli ve esnek bir ev otomasyon sistemi tasarımı sunulmuştur. Önerilen tasarım, aşırı akım ve düşük veya aşırı voltaj olduğunda evi korumak, ayrıca yangın algılama, gaz kaçağı veya duman olduğunda uyarı vermek ve elektrik enerjisi tüketimini en aza indirmek için kullanılabilir.

Raspberry Pi, tüm sistemi izlemek ve kontrol etmek ve sistemi buluta bağlamak için sorumlu bir sunucu olarak kullanılmıştır. Bu sistemin temel işlevi, bir Web uygulaması ve android aracılığıyla dünyanın her yerinden evi gerçek zamanlı olarak uzaktan izlemek ve kontrol etmektir. Sistem, evi kontrol ederek ve anormal koşullarda tüm cihazları kapatarak güvenlik ve emniyet sağlama özelliğine sahiptir.

Sistem donanım ve yazılıma ayrılmıştır, donanım bölümü önerilen sistemi uygulamak için gerekli gereksinimleri içerir. Bunlar, evdeki duman ve gaz sızıntıları algılamak ve sıcaklık, nem, akım, gerilim bilgilerini ölçmek için kullanılan sensörlerdir (akım, voltaj, nem, dış aydınlatma ve gaz). Ayrıca, evin ana gücünü ve cihazlarını açmak/kapatmak için kullanılan röleler ve kontaktörlerdir. Arduino Leonardo ile veriler sensörlerden toplanır, daha sonra Raspberry Pi'ye aktarılır. Yazılım kısmında birden fazla program tasarlanmıştır. İlki düşük seviyeli bir programdır, çeşitli sensörlerden veri okumak için kod içermektedir. Bu program verileri ana düğüme aktarmaktadır. İkincisi üst düzey programdır. Verileri ayıklamak, ayırmak ve görüntülemek ve ardından Thing Speak bulutunda yayınlamak için kullanılmaktadır. Üçüncüsü, izleme ve kontrol amacıyla kullanılan android üzerinde çalışan VirtuinoMQTT temelli olan bir uygulamadır. Çalışmada geliştirilen tüm donanım ve yazılım bileşenlerinin ev otomasyon sisteminin düzgün bir şekilde çalıştığı test edilerek doğrulanmıştır. Bu çalışma bu alanda yapılacak gelecek çalışmalara yön gösterebilir.

KAYNAKLAR

- A. ZMPT101B AC single phase voltage sensor module specifications and features,. (2018). https://www.aliexpress.com/item/32347399136.html?src=google&albc h=search&acnt=479-062-3723&isdl=y&aff_short_key=UneMJZVf&albc p=266121556&albag=7593673036&slnk=&trgt=aud-30186416528:dsa-42862830006&plac=&crea=64152518476&netw=g&device=c&mtctp=b&memo1=1t1&a
- Agarwal, T. (2014). Interfacing Zigbee Module With the Microcontroller. <https://www.efxkits.co.uk/interfacing-zigbee-module-with-microcontroller/>
- Alliance, Z. (2002). The Zigbee Alliance Celebrates 15 Years and A Decade of Standards - Zigbee Alliance. https://zigbeealliance.org/news_and_articles/the-zigbee-alliance-celebrates-15-years-and-a-decade-of-standards/
- Arduino - The open-source Arduino Software (IDE) Software. (2018). <https://www.arduino.cc/en/Main/Software>
- Arduino Leonardo with Headers, (2018). <https://store.arduino.cc/arduino-leonardo-with-headers>
- Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56(October 2017), 122–140. <https://doi.org/10.1016/j.adhoc.2016.12.004>
- Bepery, C., Baral, S., Khashkel, A., & Hossain, F. (2019). ADVANCED HOME AUTOMATION SYSTEM USING RASPBERRY-PI AND ARDUINO. www.iaset.us
- Chaudary, A. I., Usman, M., Farhad, A., & Khan, W. U. (2016). Towards the Development of an Efficient and Cost Effective Intelligent Home System Based on the Internet of Things. *International Journal of Computer Science and Information Security*, 14(6), 343–350. https://search.proquest.com/docview/1807061942?accountid=134127%0Ahttp://link.periodicos.capes.gov.br/sfxlcl41?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&genre=article&sid=ProQ:ProQ%3Acomputerinfo&atitle=Towards+the+Development+of+an+Eff
- Cisco (2016). Internet of Things At-a-Glance. CISCO Software Engineering for Embedded Systems, 465–499. <https://doi.org/10.1016/b978-0-12-809448-8.00013-8>

- Contactors : Electromechanical Relays,. (2016). <https://www.allaboutcircuits.com/textbook/digital/chpt-5/contactors/>
- Cyril Jose, A., & Malekian, R. (2016). Smart Home Automation Security: A Literature Review. *The Smart Computing Review*, Rtdm, 6–10. <https://doi.org/10.6029/smartcr.2015.04.004>
- DHT22 Digital Temperature and Humidity Sensor. (2018). <https://robuin/product/dht22-digital-temperature-humidity-sensor-temperature-humidity-module-am2302/>
- Dr. Vermesan, O., & Dr. Friess, P. (2015). The internet of things: Converging Technologies for Smart Environments and Integrated Ecosystems. http://www.worldcat.org/title/internet-of-things/oclc/952553238&referer=brief_results
- Elprocus. (2013). What is ZigBee Technology, Architecture and its Applications? <https://www.elprocus.com/what-is-zigbee-technology-architecture-and-its-applications/>
- Ersue, M., Romascanu, D., Schoenwaelder, J., & Sehgal, A. (2015). Management of Networks with Constrained Devices: Use Cases. <https://tools.ietf.org/html/rfc7548>
- Gill, K., Yang, S. H., Yao, F., & Lu, X. (2009). A ZigBee-based home automation system. *IEEE Transactions on Consumer Electronics*, 55(2), 422–430. <https://doi.org/10.1109/TCE.2009.5174403>
- How MQ2 Gas/Smoke Sensor Works. (2018). <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/>
- InnovatorSguru. (2019). SCT-013-000 Split Core Current Transformer, 100A, 50A, 30A. <https://innovatorsguru.com/sct-013-000/>
- Jain, P. S. A., Maineka, S., & Nimgade, P. (2016). APPLICATION OF IOT-WSN IN HOME AUTOMATION SYSTEM: A LITERATURE SURVEY. 3(1), 916–922.
- Kasmi, M., Bahloul, F., & Tkitek, H. (2016). Smart home based on Internet of Things and cloud computing. 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, SETIT 2016, 82–86. <https://doi.org/10.1109/SETIT.2016.7939846>
- Kevin Asthon. (2010). That ' Internet of Things ' Thing. *RFID Journal*, 4986. <https://doi.org/10.1038/nature03475>

- Kinsella, B. (2019). Apple and Zigbee Alliance to be Big Winners in New Smart Home Announcement with Amazon and Google - Voicebot.ai. <https://voicebot.ai/2019/12/18/apple-and-zigbee-alliance-to-be-big-winners-in-new-smart-home-announcement-with-amazon-and-google/>
- KonnectedInc. (2018). Relay Modules with High/Low Level Trigger. <https://konnected.io/products/1-channel-5v-relay-module-with-high-low-level-trigger>
- Kpae, P., Michael, M., Nsikak D, E., Chigozinum T, W., Abdullahi, A., & Friday O.N, K. (2018). Design and Implementation of a Smart Home (Smoke , Fire , Gas and Motion Detector). 1120–1153.
- Kraeling, M., & Brogioli, M. C. (2019). Internet of Things. In *CTO Office, GE Transportation, Melbourne, FL, United States †Polymathic Consulting, Austin, TX, United States (Second Edi). Elsevier Inc. <https://doi.org/10.1016/b978-0-12-809448-8.00013-8>
- Kumar, T., & Mane, P. B. (2016). ZigBee topology: A survey. 2016 International Conference on Control Instrumentation Communication and Computational Technologies, ICCICCT 2016, 164–166. <https://doi.org/10.1109/ICCICCT.2016.7987937>
- Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440. <https://doi.org/10.1016/j.bushor.2015.03.008>
- Lobaccaro, G., Carlucci, S., & Löfström, E. (2016). A Review of Systems and Technologies for Smart Homes and Smart Grids. *Energies*, 1–33. <https://doi.org/10.3390/en9050348>
- Lueth, K. L. (2015). The 10 most popular Internet of Things applications right now. <https://iot-analytics.com/10-internet-of-things-applications/>
- Macmanus, R. (2009). *FedEx Joins the Internet of Things With SenseAware*. ReadWrite. https://readwrite.com/2009/12/04/fedex_joins_the_internet_of_things_with_senseaware/
- Mahmood, N. T., & Abbas, E. I. (2017). Wireless Controlling and Monitoring the Consumption Energy Meter based on ZigBee Technology. *International Journal of Current Engineering and Technology*, 410677(55). <http://inpressco.com/category/ijcet>
- Mahmud, K., Town, G. E., Morsalin, S., & Hossain, M. J. (2017). Integration of electric vehicles and management in the internet of energy. *Renewable and Sustainable Energy Reviews*, August, 0–1. <https://doi.org/10.1016/j.rser.2017.11.004>

- Maksimovic, M., & Vujovic', V. (2015). Raspberry Pi as a Sensor Web node for home automation q. <https://doi.org/10.1016/j.compeleceng.2015.01.019>
- Malche, T., & Maheshwary, P. (2017). Internet of Things (IoT) for building Smart Home System. 65–70.
- Meola, A. (2016). Why IoT, big data & smart farming are the future of agriculture. Business Insider UK. <https://uk.news.yahoo.com/why-iot-big-data-smart-192155797.html>
- Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. In *Ad Hoc Networks* (Vol. 10, Issue 7, pp. 1497–1516). Elsevier B.V. <https://doi.org/10.1016/j.adhoc.2012.02.016>
- More Than 30 Billion Devices. (2013). ABI Research-Oyster Bay, New York. <https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne/>
- MQTT.org. (2018). FAQ MQTT. <https://mqtt.org/faq/>
- Mybotic. (2016). How to detect concentration of gas by using MQ2 sensor. *Instructables*. <http://www.instructables.com/id/How-to-Detect-Concentration-of-Gas-by-Using-MQ2-Se/>
- O'Brien, B. (2019). *Internet of Things (IoT) Applications*. UKDiss. https://www.ukdiss.com/examples/internet-of-things-applications.php#_Toc482816936
- Patel, K. K., Patel, S. M., & Scholar, P. G. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, 6(5), 1–10. <https://doi.org/10.4010/2016.1482>
- Prasanth, A. (2012). Cloud Computing Services : A Survey. May 2012, 24–29.
- Radhika, C., & Menaka, M. (2016). Survey on IoT Technologies for Home Automation System Abstract: *International Journal of Engineering and Techniques*, 2(6), 159–165.
- Raghavendran, V., Satish, N., Varma, S., & Moses, J. (2016). A Study on Cloud Computing Services. *Advances in Intelligent Systems and Computing*, 248 VOLUME(October 2017), 605–612. https://doi.org/10.1007/978-3-319-03107-1_66
- Rose, K., Eldridge, S., & Chapin, L. (2015). THE INTERNET OF THINGS: AN OVERVIEW. *Internet Society (ISOC)*, 64(9), 54–55.

- Rouse, M. (2018). internet of things (IoT). IOT Agenda. <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- Sagar K N, V., & Kusuma, S. M. (2015). Home Automation Using Internet of Things. *Lecture Notes in Networks and Systems*, 02(03), 293–301. https://doi.org/10.1007/978-981-13-0277-0_24
- Sakovich, N. (2018). IaaS vs. PaaS vs. SaaS. Advantages and Disadvantages | SaM Solutions. <https://www.sam-solutions.com/blog/iaas-vs-paas-vs-saas-whats-the-difference/>
- Sakovich, N. (2019). Fog Computing vs. Cloud Computing: Key Differences | SaM Solutions. <https://www.sam-solutions.com/blog/fog-computing-vs-cloud-computing-for-iot-projects/>
- SCT013-000 Split core current transformer. (2017). <https://www.poweruc.pl/collections/split-core-current-transformers2/products/split-core-current-transformer-sct013-rated-input-5a-100a>
- Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1), 49–69. <https://doi.org/10.1007/s11277-011-0288-5>
- Sidi Ahmed, S.-M., & Zuhuda, S. (2015). The Concept of Internet of Things and Its Challenges to Privacy. *South East Asia Journal of Contemporary Business, Economics and Law*, 8(4), 1–5.
- Soni, D., & Makwana, A. (2017). A survey on mqtt: a protocol of internet of things(IoT). *International Conference on Telecommunication, Power Analysis and Computing Techniques (Ictpact - 2017)*, April, 0–5. https://www.researchgate.net/publication/316018571_A_SURVEY_ON_MQTT_A_PROTOCOL_OF_INTERNET_OF_THINGS_IOT
- Srinivasan, C. R., Rajesh, B., Saikalyan, P., Premsagar, K., & Yadav, E. S. (2019). A review on the different types of internet of things (IoT). *Journal of Advanced Research in Dynamical and Control Systems*, 11(1), 154–158.
- Started with Thing speak.* (2019). <https://www.mathworks.com/help/thingspeak/>
- Tadapaneni, N. R. (2020). Cloud computing: An emerging technology+. 2010 International Conference on Computer Design and Applications, ICCDA 2010, 5(3), 37–40. <https://doi.org/10.1109/ICCDA.2010.5541105>
- Visual Studio 2019 Community free download.*, (n.d.). Retrieved December 13, 2019, from <https://visualstudio.microsoft.com/downloads/>

- Vogels, W. (2020). Eight Predictions on How Technology Will Continue to Change Our Lives in the Coming Year Table of Contents. December 2020.
- VoltT, C. (2018). Flame Detector With Arduino. 1–6.
- Wadhvani, S., Singh, U., Singh, P., & Dwivedi, S. (2018). Smart Home Automation and Security System using Arduino and IOT. *International Research Journal of Engineering and Technology*. www.irjet.net
- WatElectronics. (2019). *Light Dependent Resistor (LDR) - Working Principle and Its Applications*. <https://www.watelectronics.com/light-dependent-resistor-ldr-with-applications/>
- XCTU User Guide. (n.d.). Retrieved May 13, 2019, from <https://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm>
- Xie, X.-F., & Wang, Z.-J. (2017). Integrated In-Vehicle Decision Support System for Driving at Signalized Intersections: A Prototype of Smart IoT in Transportation. <https://trid.trb.org/view/1437314>
- Zhang, Q. (2015). Precision Agriculture Technology for Crop Farming. CRC Press, 2015.

EKLER

EK-1: Ana Düğümün Kodu

```
String Node = "NODE";
#define numbr 3
String Node_receive[numbr],receive_data[numbr];
String      a="0/0/0/d/u/u/u/",      b="0/0/0/d/d/u/u/u/",      c="0/0/0/d/d/u/u/u/",
x="0/0/0/d/d/u/u/u/",      y="0/0/0/d/d/u/u/u/",      z="0/0/0/d/d/u/u/u/",
w="0/0/0/d/d/u/u/u/0/0/0/d/d/u/u/u/0/0/0/d/d/u/u/u/B/B/B";
boolean slv_stat[numbr];
int selektor, slv_countr,slvError[numbr],slvError2[numbr],slvError3[numbr];
char d='r',v;
char s1='B',s2='B',s3='B';
char re_Node[45];
int m=0;
int maximum=30;

void tranceiver_data(int time_to_delay, int number_of_slave)
{
  int i = 0;
  volatile long  delay_data_time = millis();
  Node_receive[slv_countr] = "";
  receive_data[slv_countr] = "";
  while (millis() - delay_data_time <= time_to_delay)
  {
    if (Serial1.available() > 0)
    {
      while (Serial1.available() > 0 && i <= 45)
      {
        re_Node[i] = Serial1.read();
        delay(1);

        if (i<=7)Node_receive[slv_countr] += String(re_Node[i]);
        else if (i<=45)receive_data[slv_countr] += String(re_Node[i]);
        i++;
      }
    }
    Serial1.flush();
  }
}

void setup()
{
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop()
{
```

```

if (d=='+')
maximum++;
else if(d=='-')
maximum--;
for (slv_countr = 0; slv_countr < numbr; slv_countr++)
{
    if(Serial.available()>0)
    d=Serial.read();
    if(d!='r')
    m++;
    if(m>=6)
    {
        d='r';
        m=0;
    }

    String c = Node + String(slv_countr + 1)+String(d);
    char send_to_slave[c.length() + 1];
    c.toCharArray(send_to_slave, c.length() + 1) ;
    Serial1.write(send_to_slave, sizeof(send_to_slave));
    delay(50);
    trancever_data(500, slv_countr);

    if(Node_receive[slv_countr]==Node+String(slv_countr+1))
{
        slvError[slv_countr]=0;
}

    if(Node_receive[slv_countr]== "")
{
    slvError[slv_countr]+=1;
    slvError[slv_countr]=constrain(slvError[slv_countr],0,6);
}

    if (slvError[0]<=1)
        s1='A';
    else
        s1='B';
    if (slvError[1]<=1)
        s2='A';
    else
        s2='B';
    if (slvError[2]<=1)
        s3='A';
    else
        s3='B';

if (Node_receive[slv_countr] == Node + String(1))
{
        x=receive_data[slv_countr];
}

```

```
else if (Node_receive[slv_countr] == Node + String(2))
{
    y=receive_data[slv_countr];
}
else if (Node_receive[slv_countr] == Node + String(3))
{
    z=receive_data[slv_countr];
}
a=x;
b=y;
c=z;
String r="/";
w=a+b+c;
Serial.println(w);
}
}
```

EK-2: Dügüm I Kodu

```
String Node = "NODE1";
#include <Wire.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include "EmonLib.h" // Emon Library
EnergyMonitor emon1,emon2,emon3;

LiquidCrystal_I2C lcd(0x27,20,4);
const int pinvoltage=A0;
const float Ref=55.3;
float V, VLrms;
double sensrVal=0;
double sensrVal1=0;
double sensrVal2=0;
double Prms;
double VOmax ;
double VOmin;
double Irms;
int crosscount=0;
int climbhill=0;
char re_Node[20],v;
String z,q="A";
String r="/",s="u",o="u",t="u";
String Node_receive="";
String receive_data="";
boolean ok = false;

void tranceiver_data()
{
  int i = 0;
  Node_receive = "";
  receive_data="";
  if (Serial1.available() > 0)
  {
    while (Serial1.available() > 0)
    {
      re_Node[i] = Serial1.read();
      delay(1);
      if(i<=7) Node_receive += String(re_Node[i]);
      else if(i<=10) receive_data+=String(re_Node[i]);
      i++;
    }
    Serial1.flush();
  }
  if(Node_receive==Node)
```

```

ok=true;
}

void setup()
{
    lcd.init();           // initialize the lcd
    lcd.backlight();
    lcd.setCursor(1,0);
    lcd.print("Home power monitor");

    pinMode(5,OUTPUT); // Power relay
    pinMode(6,OUTPUT); //
    emon1.current(1, Ref); // for Current calibration.
    pinMode(13,OUTPUT);
    digitalWrite(6,HIGH);
    digitalWrite(5,HIGH);
    Serial.begin(9600);
    Serial1.begin(9600);
}
void loop()
{
    if (!ok)
        trancever_data();
    else
    {

        //----- voltage read -----
        Vmax=0;
        Vmin=1023;
        for(int i1=0; i1<1000; i1++)
        {
            sensrVal1=Vmax;
            sensrVal2=Vmin;
            sensrVal= analogRead(pinvoltage);
            if(sensrVal>sensrVal1)
            {
                Vmax=sensrVal;
                goto bailout;
            }
            if (sensrVal<sensrVal2)
            {
                Vmin=sensrVal;
                goto bailout;
            }
            Vmax=sensrVal1;
            Vmin=sensrVal2;
            bailout:
            delay(0.1);
        }
    }
}

```

```

float k=(V0max-V0min);
V=(0.00000412*k*k*k-0.000857*k*k+2.675*k-3.198);
int a=V/(2*sqrt(2));

lcd.setCursor(0,1);
lcd.print("volts= ");
lcd.print(a);
lcd.print(" V");
Serial.print("voltage= ");
Serial.print(a);
Serial.print(" V");
delay(2);

//----- current read -----
Irms = emon1.calcIrms(1480); // Calculate Irms
Prms = VLrms*Irms;
float b =Irms;
int w =Prms;
lcd.print(" Amps= ");
lcd.print(b);
lcd.print(" A");
Serial.print(" Current = ");
Serial.print(b);
Serial.println(" A");
lcd.setCursor(0,2);
lcd.print("power = ");
lcd.print(w);
lcd.print("W");
Serial.print(" power = ");
Serial.print(w);
Serial.println(" W");

Serial.println(q);

String x= receive_data;
char l= x.charAt(0);
Serial.println(l);

if(l=='c')
{
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
lcd.setCursor(0,3);
lcd.print("power stautuse=");
lcd.print("ON ");
}
else if(l=='d')
{
digitalWrite(5,LOW);

```

```
digitalWrite(6,LOW);
lcd.setCursor(0,3);
lcd.print("power statuse=");
lcd.print("Off");
}
delay(15);

ok=false;
String j=String(a)+r+String(b)+r ;
String c = Node+String(j);
char send_to_master[c.length() + 1];
c.toCharArray(send_to_master, c.length() + 1) ;
Serial1.write(send_to_master, sizeof(send_to_master));
}
}
```

EK-3:Düğüm II Kodu

```
String Node = "NODE2";
#include <DHT.h>
#include <MQ2.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);
#define DHTPIN 7 // what pin we're connected to
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor for normal 16mhz Arduino

int chk;
int hum; //Stores humidity value
float temp; //Stores temperature value
int Analog_Input = A0;
int lpg, co, smoke;
String z,q="A";
String r="/" ,s="u",o="u",t="u";
char re_Node[20],v;
String Node_receive="";
String receive_data="";
boolean ok = false;

MQ2 mq2(Analog_Input);
void trancever_data()
{
    int i = 0;
    Node_receive = "";
    receive_data="";
    if (Serial1.available() > 0)
    {
        while (Serial1.available() > 0)
        {
            re_Node[i] = Serial1.read();
            delay(1);
            if(i<=7) Node_receive += String(re_Node[i]);
            else if(i<=10) receive_data+=String(re_Node[i]);
            i++;
        }
        Serial1.flush();
    }
    if(Node_receive==Node)
        ok=true;
}
```

```

void setup()
{
    lcd.init();           // initialize the LCD
    lcd.backlight();
    lcd.setCursor(0,0);
    lcd.print("Climte & gas monitor");
    mq2.begin();
    dht.begin();
    pinMode(5,OUTPUT); // relay of Air
    pinMode(6,OUTPUT); // relay of Heater
    Serial.begin(9600);
    Serial1.begin(9600);
}
void loop()
{
    if (!ok)
        trancever_data();
    else
    {
        //----- MQ2 Sensor read -----
        float* values= mq2.read(true); //set it false if you don't want to print the values
in the Serial
        lpg = mq2.readLPG();
        co = mq2.readCO();
        smoke = mq2.readSmoke();
        /// Print LPG , CO and Smoke sensor values to LCD monitor
        lcd.setCursor(0,1);
        lcd.print("LPG:");
        lcd.print(lpg);
        lcd.print(" CO:");
        lcd.print(co);
        lcd.setCursor(0,2);
        lcd.print("SMOKE: ");
        lcd.print(smoke);
        lcd.print(" PPM");
        /// Print LPG , CO and Smoke sensor values to serial monitor
        Serial.print("LPG: ");
        Serial.println(lpg);
        Serial.print(" CO: ");
        Serial.println(co);
        Serial.print("SMOKE: ");
        Serial.print(smoke);
        Serial.println(" PPM");
        delay(100);

        float a= lpg;
        int b= smoke;
        //----- DH22 Sensor read -----
    }
}

```

```

//Read data and store it to variables hum and temp
    hum = dht.readHumidity();
    temp= dht.readTemperature();
//Print temp and humidity values to serial monitor
    Serial.print("Humidity: ");
    Serial.print(hum);
    Serial.print(" %, Temp: ");
    Serial.print(temp);
    Serial.println(" Celsius");
////////Print temp and humidity values to LCD monitor //////////
    lcd.setCursor(0,3);
    lcd.print("HUM:");
    lcd.print(hum);
    lcd.print("%");
    lcd.print(" Tmp:");
    lcd.print(temp);
    lcd.print("C");
    delay(100); //Delay 2 sec.

    int w= hum;
    float k= temp;

    Serial.println(q);

    String x= receive_data;
    char l= x.charAt(0);
    Serial.println(l);

    if(l=='A')
    {
        digitalWrite(5,LOW); // TV ON
    }
    else if(l=='B')
    {
        digitalWrite(5,HIGH); // TV OFF
    }
    else if(l=='E')
    {
        digitalWrite(6,HIGH); // AIR CONDITIONER ON
    }
    else if(l=='D')
    {
        digitalWrite(6,LOW); // AIR CONDITIONER OFF
    }
    delay(15);

    ok=false;
    String j=String(a)+r+String(b)+r+String(w)+r+String(k)+r ;
    String c = Node+String(j);

```

```
char send_to_master[c.length() + 1];  
c.toCharArray(send_to_master, c.length() + 1);  
Serial1.write(send_to_master, sizeof(send_to_master));  
    }  
}
```

EK-4: Düğüm III Kodu

```
String Node = "NODE3";
#include <SoftwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

const int ledPin = 4;
const int ldrPin = A1;
const int sensorMin = 0; // sensor minimum
const int sensorMax = 1024; // sensor maximum
int chk;
char re_Node[20],v;
String z,q="A";
String r="/" ,s="u",o="u",t="u";
String Node_receive="";
String receive_data="";
boolean ok = false;

void tranceiver_data()
{
    int i = 0;
    Node_receive = "";
    receive_data="";
    if (Serial1.available() > 0)
    {
        while (Serial1.available() > 0)
        {
            re_Node[i] = Serial1.read();
            delay(1);
            if(i<=7) Node_receive += String(re_Node[i]);
            else if(i<=10) receive_data+=String(re_Node[i]);
            i++;
        }
        Serial1.flush();
    }
    if(Node_receive==Node)
        ok=true;
}

void setup()
{
    lcd.init(); // initialize the LCD
    lcd.backlight();

    pinMode(ledPin, OUTPUT);
```

```

pinMode(ldrPin, INPUT);
pinMode(7,OUTPUT); // flame alarm
pinMode(5,OUTPUT); // TV relay
pinMode(6,OUTPUT); // Ref. relay
Serial.begin(9600);
Serial1.begin(9600);

}

void loop()
{
  if (!ok)
  trncever_data();
  else
  {
    ////////// Flame Sensor read//////////
    // read the sensor on analog A0:
    int sensorReading = analogRead(A0);
    // map the sensor range (four options):
    // ex: 'long int map(long int, long int, long int, long int, long int)'
    int range = map(sensorReading, sensorMin, sensorMax, 0, 3);
    // range value:
    switch (range)
    {
      case 0: { // A fire closer than 1.5 feet away.
        lcd.setCursor(1,0);
        lcd.print("Fire:");
        lcd.print(" V.Close");
        Serial.println(" V.Close");
        digitalWrite(7,HIGH);
        break;}
      case 1: { // A fire between 1-3 feet away.
        lcd.setCursor(1,0);
        lcd.print("Fire:");
        lcd.print(" Distant");
        Serial.println("*** Distant Fire ***");
        digitalWrite(7,HIGH);
        break; }
      case 2: { // No fire detected.
        lcd.setCursor(1,0);
        lcd.print("Fire:");
        lcd.print(" No Fire");
        Serial.println("No Fire");
        digitalWrite(7,LOW);
        break; }

    }
    Serial.println(range);
    delay(100); // delay between reads
  }
}

```

```
int a= range;
```

```
///----- LDR Sensor read -----
```

```
int ldrStatus = analogRead(ldrPin);
if (ldrStatus <= 300) {
    digitalWrite(ledPin, LOW);
    Serial.print("Its BRIGHT, Turn off the LIGHTS : ");
    lcd.setCursor(1,1);
    lcd.print("Lights:");
    lcd.print(" OFF");
    Serial.println(ldrStatus);
}
else {
    Serial.print("Its DARK, Turn on the LIGHTS : ");
    Serial.println(ldrStatus);
    lcd.setCursor(1,1);
    lcd.print("Lights:");
    lcd.print(" ON ");
    digitalWrite(ledPin, HIGH);
}
delay(100); //Delay 2 sec.
```

```
int b= ldrStatus;
Serial.println(q);
String x= receive_data;
char l= x.charAt(0);
Serial.println(l);
```

```
if(l=='S')
{
    digitalWrite(5,HIGH); // TV ON
}
else if(l=='M')
{
    digitalWrite(5,LOW); // TV OFF
}
else if(l=='F')
{
    digitalWrite(6,HIGH); // REF. ON
}
else if(l=='G')
{
    digitalWrite(6,LOW); // REF OFF
}
delay(15);
ok=false;
String j=String(a)+r+String(b)+r+o ;
String c = Node+String(j);
char send_to_master[c.length() + 1];
```

```
        c.toCharArray(send_to_master, c.length() + 1);  
        Serial1.write(send_to_master, sizeof(send_to_master));  
    }  
}
```

EK-5: Ana Programın Görsel Stüdyo Kodu

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports
Imports System.Text
Imports System.Net
Imports System.IO
Imports System.Linq
Imports System.Web

Public Class Home
    Dim valu1 As Integer
    Dim valu2 As Decimal
    Dim valu3 As Decimal
    Dim valu4 As Decimal
    Dim valu5 As Decimal
    Dim valu6 As Decimal
    Dim valu7 As Decimal
    Dim valu8 As Decimal
    Dim counter As Integer
    Dim postdata As String = "55"
    Dim mydata As String = ""
    Dim s As String
    Dim a, b, c, d, f, j, w, r, Remote As Decimal
    Dim myPort As Array 'store COM Ports detected on the system here

    Private Sub btnConnect_Click(sender As Object, e As EventArgs)
        Handles btnConnect.Click
        SerialPort1.PortName = cmbPort.Text
        SerialPort1.BaudRate = cmbBaud.Text
        SerialPort1.Parity = IO.Ports.Parity.None
        SerialPort1.StopBits = IO.Ports.StopBits.One
        SerialPort1.DataBits = 8
        SerialPort1.Open()

        btnConnect.Enabled = False
        btnDisconnect.Enabled = True
    End Sub

    Private Sub btnDisconnect_Click(sender As Object, e As EventArgs)
        Handles btnDisconnect.Click
```

```

    Try
        SerialPort1.DiscardInBuffer()
        SerialPort1.DiscardOutBuffer()
        SerialPort1.Close()
        btnConnect.Enabled = True
        btnDisconnect.Enabled = False
    Catch ex As Exception

End Try
End Sub

Private Sub Home_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    myPort = IO.Ports.SerialPort.GetPortNames()
                                   available
    cmbBaud.Items.Add(9600) 'Store common baud rates in Combo box of cmbBaud
    cmbBaud.Items.Add(19200)
    cmbBaud.Items.Add(38400)
    cmbBaud.Items.Add(57600)
    cmbBaud.Items.Add(115200)

    For i = 0 To UBound(myPort)
        cmbPort.Items.Add(myPort(i))
    Next

    cmbPort.Text = cmbPort.Items.Item(0) 'Assign cmbPort text to the first COM port detected
    cmbBaud.Text = cmbBaud.Items.Item(0) 'Assign cmbBaud text to the first Baud rate on list

    btnDisconnect.Enabled = False      'At first Disable the Disconnect Button
End Sub

Private Sub cmbBaud_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
    cmbBaud.SelectedIndexChanged

    If SerialPort1.IsOpen = False Then
        SerialPort1.BaudRate = cmbBaud.Text

    Else
        MsgBox("Valid only if port is Closed", vbCritical)
    End If
End Sub

Private Sub cmbPort_SelectedIndexChanged(sender As Object, e As EventArgs)
    Handles cmbPort.SelectedIndexChanged
    If SerialPort1.IsOpen = False Then
        SerialPort1.PortName = cmbPort.Text

    Else
        MsgBox("Valid only if port is Closed", vbCritical)
    End If
End Sub

Private Sub datareceived(ByVal sender As Object, ByVal e As SerialDataReceivedE
ventArgs) Handles SerialPort1.DataReceived

    Try

```

```

mydata = SerialPort1.ReadExisting()
If TextBox9.InvokeRequired Then
    TextBox9.Invoke(DirectCast(Sub() TextBox9.Text &= mydata, MethodInvocation
r))
Else
    TextBox9.Text = mydata

End If

Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try

```

End Sub

```

Private Sub pwron_Click(sender As Object, e As EventArgs)Handles pwron.Click
If (SerialPort1.IsOpen()) Then
    SerialPort1.Write("d")

```

```

Dim Request As WebRequest = WebRequest.Create("https://api.
thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field1=1")
Request.Method = "POST"
Dim postdata As String = "55"
Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)
Request.ContentType = "application/x-www-form-urlencoded"
Request.ContentLength = byteArray.Length
Dim datastream As Stream = Request.GetRequestStream()
datastream.Write(byteArray, 0, byteArray.Length)
datastream.Close()
Dim response As WebResponse = Request.GetResponse()
Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
datastream = response.GetResponseStream()
Dim reader As New StreamReader(datastream)
Dim responsefromserver As String = reader.ReadToEnd()
Console.WriteLine(responsefromserver)
reader.Close()
datastream.Close()
response.Close()

```

```

End If
End Sub

```

```

Private Sub pwroff_Click(sender As Object, e As EventArgs)Handles pwroff.Click
If (SerialPort1.IsOpen()) Then
    SerialPort1.Write("c")

```

```

Dim Request As WebRequest = WebRequest.Create("https://api.
thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field1=0")
Request.Method = "POST"

```

```

    Dim postData As String = "55"
    Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)
    Request.ContentType = "application/x-www-form-urlencoded"
    Request.ContentLength = byteArray.Length
    Dim datastream As Stream = Request.GetRequestStream()
    datastream.Write(byteArray, 0, byteArray.Length)
    datastream.Close()
    Dim response As WebResponse = Request.GetResponse()
    Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
    datastream = response.GetResponseStream()
    Dim reader As New StreamReader(datastream)
    Dim responsefromserver As String = reader.ReadToEnd()
    Console.WriteLine(responsefromserver)
    reader.Close()
    datastream.Close()
    response.Close()
End If
End Sub

Private Sub Airon_Click(sender As Object, e As EventArgs)Handles Airon.Click
    If (SerialPort1.IsOpen()) Then
        SerialPort1.Write("A")

        Dim Request1 As WebRequest = WebRequest.Create("https://api.
            thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field2=1")
        Request1.Method = "POST"
        Dim postData1 As String = "55"
        Dim byteArray1 As Byte() = Encoding.UTF8.GetBytes(postdata)
        Request1.ContentType = "application/x-www-form-urlencoded"
        Request1.ContentLength = byteArray1.Length
        Dim datastream1 As Stream = Request1.GetRequestStream()
        datastream1.Write(byteArray1, 0, byteArray1.Length)
        datastream1.Close()
        Dim response1 As WebResponse = Request1.GetResponse()
        Console.WriteLine(CType(response1, HttpWebResponse).StatusDescription)
        datastream1 = response1.GetResponseStream()
        Dim reader1 As New StreamReader(datastream1)
        Dim responsefromserver1 As String = reader1.ReadToEnd()
        Console.WriteLine(responsefromserver1)
        reader1.Close()
        datastream1.Close()
        response1.Close()
    End If
End Sub

Private Sub Airoff_Click(sender As Object, e As EventArgs)Handles Airoff.Click
    If (SerialPort1.IsOpen()) Then
        SerialPort1.Write("B")
    End If
End Sub

```

```

Dim Request1 As WebRequest = WebRequest.Create("https://api.
    thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field2=0")
Request1.Method = "POST"
Dim postData1 As String = "55"
Dim byteArray1 As Byte() = Encoding.UTF8.GetBytes(postdata)
Request1.ContentType = "application/x-www-form-urlencoded"
Request1.ContentLength = byteArray1.Length
Dim datastream1 As Stream = Request1.GetRequestStream()
datastream1.Write(byteArray1, 0, byteArray1.Length)
datastream1.Close()
Dim response1 As WebResponse = Request1.GetResponse()
Console.WriteLine(CType(response1, HttpWebResponse).StatusDescription)
datastream1 = response1.GetResponseStream()
Dim reader1 As New StreamReader(datastream1)
Dim responsefromserver1 As String = reader1.ReadToEnd()
Console.WriteLine(responsefromserver1)
reader1.Close()
datastream1.Close()
response1.Close()
End If
End Sub

Private Sub Heateron_Click(sender As Object, e As EventArgs) Handles
    Heateron.Click

If (SerialPort1.IsOpen()) Then
    SerialPort1.Write("D")

    Dim Request2 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field3=1")
    Request2.Method = "POST"
    Dim postData2 As String = "55"
    Dim byteArray2 As Byte() = Encoding.UTF8.GetBytes(postdata2)
    Request2.ContentType = "application/x-www-form-urlencoded"
    Request2.ContentLength = byteArray2.Length
    Dim datastream2 As Stream = Request2.GetRequestStream()
    datastream2.Write(byteArray2, 0, byteArray2.Length)
    datastream2.Close()
    Dim response2 As WebResponse = Request2.GetResponse()
    Console.WriteLine(CType(response2, HttpWebResponse).StatusDescription)
    datastream2 = response2.GetResponseStream()
    Dim reader2 As New StreamReader(datastream2)
    Dim responsefromserver2 As String = reader2.ReadToEnd()
    Console.WriteLine(responsefromserver2)
    reader2.Close()
    datastream2.Close()
    response2.Close()
End If
End Sub

```

Private Sub Heateroff_Click(sender As Object, e As EventArgs) Handles Heateroff.Click

If (SerialPort1.IsOpen()) Then
SerialPort1.Write("E")

Dim Request2 As WebRequest = WebRequest.Create("https://api.
thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field3=0")

Request2.Method = "POST"

Dim postData2 As String = "55"

Dim byteArray2 As Byte() = Encoding.UTF8.GetBytes(postData2)

Request2.ContentType = "application/x-www-form-urlencoded"

Request2.ContentLength = byteArray2.Length

Dim datastream2 As Stream = Request2.GetRequestStream()

datastream2.Write(byteArray2, 0, byteArray2.Length)

datastream2.Close()

Dim response2 As WebResponse = Request2.GetResponse()

Console.WriteLine(CType(response2, HttpWebResponse).StatusDescription)

datastream2 = response2.GetResponseStream()

Dim reader2 As New StreamReader(datastream2)

Dim responsefromserver2 As String = reader2.ReadToEnd()

Console.WriteLine(responsefromserver2)

reader2.Close()

datastream2.Close()

response2.Close()

End If

End Sub

Private Sub TVon_Click(sender As Object, e As EventArgs) Handles TVon.Click

If (SerialPort1.IsOpen()) Then

SerialPort1.Write("M")

Dim Request3 As WebRequest = WebRequest.Create("https://api.
thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field4=1")

Request3.Method = "POST"

Dim postData3 As String = "55"

Dim byteArray3 As Byte() = Encoding.UTF8.GetBytes(postData3)

Request3.ContentType = "application/x-www-form-urlencoded"

Request3.ContentLength = byteArray3.Length

Dim datastream3 As Stream = Request3.GetRequestStream()

datastream3.Write(byteArray3, 0, byteArray3.Length)

datastream3.Close()

Dim response3 As WebResponse = Request3.GetResponse()

Console.WriteLine(CType(response3, HttpWebResponse).StatusDescription)

datastream3 = response3.GetResponseStream()

Dim reader3 As New StreamReader(datastream3)

Dim responsefromserver3 As String = reader3.ReadToEnd()

Console.WriteLine(responsefromserver3)

reader3.Close()

datastream3.Close()

```

        response3.Close()
    End If
End Sub

```

```

Private Sub TVoff_Click(sender As Object, e As EventArgs) Handles TVoff.Click

```

```

    If (SerialPort1.IsOpen()) Then
        SerialPort1.Write("S")

        Dim Request3 As WebRequest = WebRequest.Create("https://api.
            thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field4=0")
        Request3.Method = "POST"
        Dim postData3 As String = "55"
        Dim byteArray3 As Byte() = Encoding.UTF8.GetBytes(postData3)
        Request3.ContentType = "application/x-www-form-urlencoded"
        Request3.ContentLength = byteArray3.Length
        Dim datastream3 As Stream = Request3.GetRequestStream()
        datastream3.Write(byteArray3, 0, byteArray3.Length)
        datastream3.Close()
        Dim response3 As WebResponse = Request3.GetResponse()
        Console.WriteLine(CType(response3, HttpWebResponse).
            StatusDescription)
        datastream3 = response3.GetResponseStream()
        Dim reader3 As New StreamReader(datastream3)
        Dim responsefromserver3 As String = reader3.ReadToEnd()
        Console.WriteLine(responsefromserver3)
        reader3.Close()
        datastream3.Close()
        response3.Close()
    End If
End Sub

```

```

Private Sub Refon_Click(sender As Object, e As EventArgs) Handles Refon.Click

```

```

    If (SerialPort1.IsOpen()) Then
        SerialPort1.Write("F")

        Dim Request4 As WebRequest = WebRequest.Create("https://api.
            thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field5=1")
        Request4.Method = "POST"
        Dim postData4 As String = "55"
        Dim byteArray4 As Byte() = Encoding.UTF8.GetBytes(postData4)
        Request4.ContentType = "application/x-www-form-urlencoded"
        Request4.ContentLength = byteArray4.Length
        Dim datastream4 As Stream = Request4.GetRequestStream()
        datastream4.Write(byteArray4, 0, byteArray4.Length)
        datastream4.Close()
        Dim response4 As WebResponse = Request4.GetResponse()
        Console.WriteLine(CType(response4, HttpWebResponse).
            StatusDescription)
        datastream4 = response4.GetResponseStream()
    End If
End Sub

```

```

Dim reader4 As New StreamReader(datastream4)
Dim responsefromserver4 As String = reader4.ReadToEnd()

Console.WriteLine(responsefromserver4)
reader4.Close()
datastream4.Close()
response4.Close()
End If
End Sub

Private Sub Refoff_Click(sender As Object, e As EventArgs) Handles Refoff.Click
If (SerialPort1.IsOpen()) Then
SerialPort1.Write("G")
Dim Request4 As WebRequest = WebRequest.Create("https://api
.thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field5=0")
Request4.Method = "POST"
Dim postdata4 As String = "55"
Dim byteArray4 As Byte() = Encoding.UTF8.GetBytes(postdata)
Request4.ContentType = "application/x-www-form-urlencoded"
Request4.ContentLength = byteArray4.Length
Dim datastream4 As Stream = Request4.GetRequestStream()
datastream4.Write(byteArray4, 0, byteArray4.Length)
datastream4.Close()
Dim response4 As WebResponse = Request4.GetResponse()
Console.WriteLine(CType(response4, HttpWebResponse)
.StatusDescription)
datastream4 = response4.GetResponseStream()
Dim reader4 As New StreamReader(datastream4)
Dim responsefromserver4 As String = reader4.ReadToEnd()
Console.WriteLine(responsefromserver4)
reader4.Close()
datastream4.Close()
response4.Close()
End If
End Sub

```

=====

```

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
Dim tagz As Integer
If (SerialPort1.IsOpen()) Then
If (mydata IsNot vbNullString) Then
s = mydata + "/" + "/" + "/" + "/" + "/" + "/" + "/"
Dim splstring() As String
Dim ML As Integer = 0
Dim VH As Integer = 0
Dim TH As Integer = 0
Dim IH As Integer = 0
Dim VL As Integer = 0
Dim IL As Integer = 0

```

```

Dim TL As Integer = 0
spltstring = s.Split(New Char() {"c"})
volt.Text = spltstring(0)
valu1 = Val(volt.Text)
a = valu1
If valu1 > 225 Then

    Label11.Text = "H"
    VH = VH + 1
ElseIf valu1 < 217 Then
    Label11.Text = "L"
    VL = VL + 1
Else
    Label11.Text = "N"
End If
curnt.Text = spltstring(1)
valu2 = Val(curnt.Text)
b = valu2
If valu2 > 3.17 Then
    Label12.Text = "H"
    IH = IH + 1
ElseIf valu2 < 3.14 Then
    Label12.Text = "L"
    IL = IL + 1
Else
    Label12.Text = "N"
End If
Gas.Text = spltstring(2)
valu4 = Val(Gas.Text)
d = valu4
If valu4 > 10 And valu4 < 20 Then
    Label14.ForeColor = Color.Orange
    Label14.Text = "!!! There is a simple Gas Leakage"
ElseIf valu4 < 10 Then
    Label14.ForeColor = Color.Green
    Label14.Text = "Normal"
    IL = IL + 1
Else

    Label14.ForeColor = Color.Red
    Label14.Text = "!!!There is a High Gas Leakage or Smoke"

```

```

hh:    tagz += 1
        SerialPort1.Write("c")
        If tagz = 1 Then
            MessageBox.Show("The Home main power is switched off
                due to Gas Leakage or Smoke", "Attention")
        End If

```

```

Dim Request5 As WebRequest = WebRequest.Create("https://
    api.thingspeak.com/update?api_key
    =M5BXHCCVQDCD6SJ6&field1=0")
Request5.Method = "POST"
Dim postData5 As String = "55"
Dim byteArray5 As Byte()=Encoding.UTF8.GetBytes(postdata5)
Request5.ContentType = "application/x-www-form-urlencoded"
Request5.ContentLength = byteArray5.Length
Dim datastream5 As Stream = Request5.GetRequestStream()
datastream5.Write(byteArray5, 0, byteArray5.Length)
datastream5.Close()
Dim response5 As WebResponse = Request5.GetResponse()
Console.WriteLine(CType(response5, HttpWebResponse)
    .StatusDescription)
datastream5 = response5.GetResponseStream()
Dim reader5 As New StreamReader(datastream5)
Dim responsefromserver5 As String = reader5.ReadToEnd()
Console.WriteLine(responsefromserver5)
reader5.Close()
datastream5.Close()
response5.Close()
End If

```

```

Smoke.Text = splstring(3)
valu3 = Val(Smoke.Text)
c = valu3
If valu3 > 200 Then
    Label13.ForeColor = Color.Red
    Label13.Text = "!!! There is a High Smoke Detected"
    GoTo hh

ElseIf valu3 < 200 And valu3 > 10 Then
    Label13.ForeColor = Color.Orange
    Label13.Text = "!!! There is a simple Smoke Detected"

Else
    Label13.ForeColor = Color.Green
    Label13.Text = "Normal"
End If

```

```

Humidity.Text = splstring(4)
valu5 = Val(Humidity.Text)
f = valu5
Label15.Text = "%"

```

```

TEMP.Text = splstring(5)
valu6 = Val(TEMP.Text)
j = valu6
Label16.Text = "C"

```

```

Fire.Text = splstring(6)
valu8 = Val(Fire.Text)
r = valu8
If valu8 = 0 Then
    Label19.ForeColor = Color.Red
    Label19.Text = "Very near Fire"
ElseIf valu8 = 1 Then
    Label19.ForeColor = Color.Orange
    Label19.Text = "Near Fire Detected"
ElseIf valu8 = 2 Then
    Label19.ForeColor = Color.Green
    Label19.Text = "No Fire Detected"
End If
Lights.Text = splstring(7)
valu7 = Val(Lights.Text)
w = valu7
If valu7 < 300 Then
    Label17.ForeColor = Color.Green
    Label17.Text = "It's Light, Lights is Turned OFF"
Else
    Label17.ForeColor = Color.Red
    Label17.Text = "It's Dark, Lights is Turned ON"
End If

```

```

Dim Request As WebRequest = WebRequest.Create("https://api.
thingspeak.com/update?api_key=67UY9R1BOOAL2C3E&field1="
& a & "&field2=" & b & "&field3=" & c & "&field4="
& d & "&field5=" & f & "&field6=" & j & "&field7="
& r & "&field8=" & w & "")
Request.Method = "POST"
Dim postdata As String = "55"
Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)
Request.ContentType = "application/x-www-form-urlencoded"
Request.ContentLength = byteArray.Length
Dim datastream As Stream = Request.GetRequestStream()
datastream.Write(byteArray, 0, byteArray.Length)
datastream.Close()
Dim response As WebResponse = Request.GetResponse()
'Display the status
    Console.WriteLine(CType(response, HttpWebResponse)
.StatusDescription)
datastream = response.GetResponseStream()
Dim reader As New StreamReader(datastream)
Dim responsefromserver As String = reader.ReadToEnd()
Console.WriteLine(responsefromserver)
reader.Close()
datastream.Close()
response.Close()

```

```

Dim webClient As WebClient = New WebClient()
    Dim data = webClient.DownloadString("https://api.thingspeak
        .com/channels/884075/fields/field1/last")
Dim Remote = Val(data)
If Remote = 0 Then
    SerialPort1.Write("c")
ElseIf Remote = 1 Then
    SerialPort1.Write("d")
End If

Dim webClient1 As WebClient = New WebClient()
    'Read only the last data Entered to filed 2 which is
    The Air condition status

    Dim data1 = webClient1.DownloadString("https://api.thingspeak.
        com/channels/884075/fields/field2/last")
Dim Remote1 = Val(data1)
If Remote1 = 0 Then
    SerialPort1.Write("B")
ElseIf Remote1 = 1 Then
    SerialPort1.Write("A")
End If

Dim webClient2 As WebClient = New WebClient()

Dim data2 = webClient2.DownloadString("https://api.
    thingspeak.com/channels/884075/fields/field3/last")

Dim Remote2 = Val(data2)
If Remote2 = 0 Then
    SerialPort1.Write("E")

ElseIf Remote2 = 1 Then
    SerialPort1.Write("D")
End If

Dim webClient3 As WebClient = New WebClient()
'Read only the last data Entered to filed 4
    which is the TV status

    Dim data3 = webClient3.DownloadString("https://api.thingspeak
        .com/channels/884075/fields/field4/last")
Dim Remote3 = Val(data3)
If Remote3 = 0 Then
    SerialPort1.Write("S")

ElseIf Remote3 = 1 Then
    SerialPort1.Write("M")

```

End If

```
Dim webClient4 As WebClient = New WebClient()  
Dim data4 = webClient4.DownloadString("https://api.  
thingspeak.com/channels/884075/fields/field5/last")  
Dim Remote4 = Val(data4)  
If Remote4 = 0 Then  
    SerialPort1.Write("G")  
ElseIf Remote4 = 1 Then  
    SerialPort1.Write("F")  
End If  
    End If  
End If  
End Sub
```

EK-6: İş İstasyonu İçin Görsel Stüdyo Kodu

```
Imports System.IO
Imports System.Text
Imports System.Net
Imports System.IO.Ports
```

```
Public Class MainForm
```

```
    Dim postdata As String = "55"
```

```
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
```

```
        Dim webClient As WebClient = New WebClient()
```

```
        Dim data = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field1/last")
```

```
        volt.Text = data
```

```
        Dim data1 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field2/last")
```

```
        curtn.Text = data1
```

```
        Dim data2 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field3/last")
```

```
        Smoke.Text = data2
```

```
        Dim data3 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field4/last")
```

```
        Gas.Text = data3
```

```
        Dim data4 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field5/last")
```

```
        Humidity.Text = data4
```

```
        Dim data5 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field6/last")
```

```
        Temperature.Text = data5
```

```
        Dim data6 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field7/last")
```

```
        If data6 = 0 Then
```

```
            Label9.Text = "Very near Fire"
```

```
        ElseIf data6 = 1 Then
```

```
            Label9.Text = "Near Fire Detected"
```

```
        ElseIf data6 = 2 Then
```

```
            Label9.Text = "No Fire Detected"
```

```
        End If
```

```
        Dim data7 = webClient.DownloadString("https://api.thingspeak.com/
channels/884057/fields/field8/last")
```

```
        If data7 < 300 Then
```

```
            Label10.Text = "It's Light, Lights is Turned OFF"
```

```
        Else
```

```
Label10.Text = "It's Dark, Lights is Turned ON"
```

```
End If
```

```
End Sub
```

```
Private Sub pwron_Click(sender As Object, e As EventArgs) Handles  
    pwron.Click  
    Dim Request As WebRequest = WebRequest.Create("https://api.  
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field1=1")  
    Request.Method = "POST"  
    Dim postdata As String = "55"  
    Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)  
    Request.ContentType = "application/x-www-form-urlencoded"  
    Request.ContentLength = byteArray.Length  
    Dim datastream As Stream = Request.GetRequestStream()  
    datastream.Write(byteArray, 0, byteArray.Length)  
    datastream.Close()  
    Dim response As WebResponse = Request.GetResponse()  
    Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)  
    datastream = response.GetResponseStream()  
    Dim reader As New StreamReader(datastream)  
    Dim responsefromserver As String = reader.ReadToEnd()  
    Console.WriteLine(responsefromserver)  
    reader.Close()  
    datastream.Close()  
    response.Close()
```

```
End Sub
```

```
Private Sub pwroff_Click(sender As Object, e As EventArgs) Handles  
    pwroff.Click  
    Dim Request As WebRequest = WebRequest.Create("https://api.  
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field1=0")  
    Request.Method = "POST"  
    Dim postdata As String = "55"  
    Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)  
    Request.ContentType = "application/x-www-form-urlencoded"  
    Request.ContentLength = byteArray.Length  
    Dim datastream As Stream = Request.GetRequestStream()  
    datastream.Write(byteArray, 0, byteArray.Length)  
    datastream.Close()  
    Dim response As WebResponse = Request.GetResponse()  
    Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)  
    datastream = response.GetResponseStream()  
    Dim reader As New StreamReader(datastream)  
    Dim responsefromserver As String = reader.ReadToEnd()  
    Console.WriteLine(responsefromserver)  
    reader.Close()  
    datastream.Close()
```

```
response.Close()  
End Sub
```

```
Private Sub Airon_Click(sender As Object, e As EventArgs) Handles  
    Airon.Click  
    Airon.Enabled = False      'Enable Connect button  
    Airoff.Enabled = True  
    Dim Request1 As WebRequest = WebRequest.Create("https://api.  
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field2=1")  
    Request1.Method = "POST"  
    Dim postData1 As String = "55"  
    Dim byteArray1 As Byte() = Encoding.UTF8.GetBytes(postdata)  
    Request1.ContentType = "application/x-www-form-urlencoded"  
    Request1.ContentLength = byteArray1.Length  
    Dim datastream1 As Stream = Request1.GetRequestStream()  
    datastream1.Write(byteArray1, 0, byteArray1.Length)  
    datastream1.Close()  
    Dim response1 As WebResponse = Request1.GetResponse()  
    Console.WriteLine(CType(response1, HttpWebResponse).StatusDescription)  
    datastream1 = response1.GetResponseStream()  
    Dim reader1 As New StreamReader(datastream1)  
    Dim responsefromserver1 As String = reader1.ReadToEnd()  
    Console.WriteLine(responsefromserver1)  
    reader1.Close()  
    datastream1.Close()  
    response1.Close()
```

```
End Sub
```

```
Private Sub Airoff_Click(sender As Object, e As EventArgs) Handles  
    Airoff.Click  
    Airon.Enabled = True      'Enable Connect button  
    Airoff.Enabled = False  
    Dim Request1 As WebRequest = WebRequest.Create("https://api.  
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field2=0")  
    Request1.Method = "POST"  
    Dim postData1 As String = "55"  
    Dim byteArray1 As Byte() = Encoding.UTF8.GetBytes(postdata)  
    Request1.ContentType = "application/x-www-form-urlencoded"  
    Request1.ContentLength = byteArray1.Length  
    Dim datastream1 As Stream = Request1.GetRequestStream()  
    datastream1.Write(byteArray1, 0, byteArray1.Length)  
    datastream1.Close()  
  
    Dim response1 As WebResponse = Request1.GetResponse()  
    Console.WriteLine(CType(response1, HttpWebResponse).StatusDescription)  
    datastream1 = response1.GetResponseStream()  
    Dim reader1 As New StreamReader(datastream1)  
    Dim responsefromserver1 As String = reader1.ReadToEnd()
```

```
Console.WriteLine(responsefromserver1)
reader1.Close()
datastream1.Close()
response1.Close()
```

End Sub

```
Private Sub Heateron_Click(sender As Object, e As EventArgs) Handles
    Heateron.Click
    Heateron.Enabled = False
    Heateroff.Enabled = True
    Dim Request2 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field3=1")
    Request2.Method = "POST"
    Dim postdata2 As String = "55"
    Dim byteArray2 As Byte() = Encoding.UTF8.GetBytes(postdata2)
    Request2.ContentType = "application/x-www-form-urlencoded"
    Request2.ContentLength = byteArray2.Length
    Dim datastream2 As Stream = Request2.GetRequestStream()
    datastream2.Write(byteArray2, 0, byteArray2.Length)
    datastream2.Close()
    Dim response2 As WebResponse = Request2.GetResponse()
    Console.WriteLine(CType(response2, HttpWebResponse).StatusDescription)
    datastream2 = response2.GetResponseStream()
    Dim reader2 As New StreamReader(datastream2)
    Dim responsefromserver2 As String = reader2.ReadToEnd()
    Console.WriteLine(responsefromserver2)
    reader2.Close()
    datastream2.Close()
    response2.Close()
```

End Sub

```
Private Sub Heateroff_Click(sender As Object, e As EventArgs) Handles
    Heateroff.Click
    Heateron.Enabled = True
    Heateroff.Enabled = False
    Dim Request2 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field3=0")
    Request2.Method = "POST"
    Dim postdata2 As String = "55"
    Dim byteArray2 As Byte() = Encoding.UTF8.GetBytes(postdata2)
    Request2.ContentType = "application/x-www-form-urlencoded"
    Request2.ContentLength = byteArray2.Length
    Dim datastream2 As Stream = Request2.GetRequestStream()
    datastream2.Write(byteArray2, 0, byteArray2.Length)
    datastream2.Close()
    Dim response2 As WebResponse = Request2.GetResponse()
    Console.WriteLine(CType(response2, HttpWebResponse).StatusDescription)
    datastream2 = response2.GetResponseStream()
```

```

Dim reader2 As New StreamReader(datastream2)
Dim responsefromserver2 As String = reader2.ReadToEnd()
Console.WriteLine(responsefromserver2)
reader2.Close()
datastream2.Close()
response2.Close()
End Sub

```

```

Private Sub TVon_Click(sender As Object, e As EventArgs) Handles
    TVon.Click

```

```

    TVon.Enabled = False
    TVoff.Enabled = True
    Dim Request3 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field4=1")
    Request3.Method = "POST"
    Dim postdata3 As String = "55"
    Dim byteArray3 As Byte() = Encoding.UTF8.GetBytes(postdata3)
    Request3.ContentType = "application/x-www-form-urlencoded"
    Request3.ContentLength = byteArray3.Length
    Dim datastream3 As Stream = Request3.GetRequestStream()
    datastream3.Write(byteArray3, 0, byteArray3.Length)
    datastream3.Close()
    Dim response3 As WebResponse = Request3.GetResponse()
    Console.WriteLine(CType(response3, HttpWebResponse).StatusDescription)
    datastream3 = response3.GetResponseStream()
    Dim reader3 As New StreamReader(datastream3)
    Dim responsefromserver3 As String = reader3.ReadToEnd()
    Console.WriteLine(responsefromserver3)
    reader3.Close()
    datastream3.Close()
    response3.Close()
End Sub

```

```

Private Sub TVoff_Click(sender As Object, e As EventArgs) Handles
    TVoff.Click

```

```

    TVon.Enabled = True        'enable Connect button
    TVoff.Enabled = False
    Dim Request3 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field4=0")
    Request3.Method = "POST"
    Dim postdata3 As String = "55"
    Dim byteArray3 As Byte() = Encoding.UTF8.GetBytes(postdata3)
    Request3.ContentType = "application/x-www-form-urlencoded"
    Request3.ContentLength = byteArray3.Length
    Dim datastream3 As Stream = Request3.GetRequestStream()
    datastream3.Write(byteArray3, 0, byteArray3.Length)
    datastream3.Close()
    Dim response3 As WebResponse = Request3.GetResponse()
    Console.WriteLine(CType(response3, HttpWebResponse).StatusDescription)

```

```

datastream3 = response3.GetResponseStream()
Dim reader3 As New StreamReader(datastream3)
Dim responsefromserver3 As String = reader3.ReadToEnd()
Console.WriteLine(responsefromserver3)
reader3.Close()
datastream3.Close()
response3.Close()
End Sub

```

```

Private Sub Refon_Click(sender As Object, e As EventArgs) Handles
    Refon.Click

```

```

    Refon.Enabled = False      'enable Connect button
    Refoff.Enabled = True
    Dim Request4 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field5=1")
    Request4.Method = "POST"
    Dim postdata4 As String = "55"
    Dim byteArray4 As Byte() = Encoding.UTF8.GetBytes(postdata4)
    Request4.ContentType = "application/x-www-form-urlencoded"
    Request4.ContentLength = byteArray4.Length

```

```

    Dim datastream4 As Stream = Request4.GetRequestStream()

```

```

    datastream4.Write(byteArray4, 0, byteArray4.Length)
    datastream4.Close()
    Dim response4 As WebResponse = Request4.GetResponse()
    Console.WriteLine(CType(response4, HttpWebResponse).StatusDescription)
    datastream4 = response4.GetResponseStream()
    Dim reader4 As New StreamReader(datastream4)
    Dim responsefromserver4 As String = reader4.ReadToEnd()

```

```

    Console.WriteLine(responsefromserver4)
    reader4.Close()
    datastream4.Close()
    response4.Close()

```

```

End Sub

```

```

Private Sub Refoff_Click(sender As Object, e As EventArgs) Handles
    Refoff.Click

```

```

    Refon.Enabled = True
    Refoff.Enabled = False
    Dim Request4 As WebRequest = WebRequest.Create("https://api.
        thingspeak.com/update?api_key=M5BXHCCVQDCD6SJ6&field5=0")
    Request4.Method = "POST"
    Dim postdata4 As String = "55"
    Dim byteArray4 As Byte() = Encoding.UTF8.GetBytes(postdata)
    Request4.ContentType = "application/x-www-form-urlencoded"
    Request4.ContentLength = byteArray4.Length
    Dim datastream4 As Stream = Request4.GetRequestStream()

```

```
datastream4.Write(byteArray4, 0, byteArray4.Length)
datastream4.Close()
Dim response4 As WebResponse = Request4.GetResponse()
Console.WriteLine(CType(response4, HttpWebResponse).StatusDescription)
datastream4 = response4.GetResponseStream()
Dim reader4 As New StreamReader(datastream4)
Dim responsefromserver4 As String = reader4.ReadToEnd()

Console.WriteLine(responsefromserver4)
reader4.Close()
datastream4.Close()
response4.Close()
End Sub
End Class
```