

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

YÜKSEK LİSANS TEZİ

**ELEKTRONİK SU SAYACI İÇİN KABLOSUZ AĞ VE RF
TABANLI MİMARİ TASARIM**

KASIM GÖRENEKLİ

KOCAELİ 2019

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ




ELEKTRONİK SU SAYACI İÇİN KABLOSUZ AĞ VE RF
TABANLI MİMARİ TASARIM

KASIM GÖRENEKLİ

Prof. Dr. Yaşar BECERİKLİ
Danışman, Kocaeli Üniv.

Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniv.

Dr. Öğr. Üyesi Muhammed Fatih ADAK
Jüri Üyesi, Sakarya Üniv.


.....

.....

.....

Tezin Savunulduğu Tarih: 15.03.2019

ÖNSÖZ VE TEŞEKKÜR

Öncelikle yüksek lisans eğitimimi ve araştırmamı sürekli sabrı, motivasyonu, heyecanı ve muazzam bilgisi ile desteklediği için danışmanım Prof. Dr. Yaşar Becerikli'ye en içten şükranlarımı sunmak isterim. Onun rehberliği, bu tezin araştırma ve yazımının her aşamasında bana yardımcı oldu. Yüksek lisans çalışmam için daha iyi bir danışman ve rehberi hayal bile edemiyordum.

Danışmanımın yanı sıra, teşvik edici ve anlayışlı yorumları için Prof. Dr. Mehmet Melih İnal'a teşekkürlerimi sunarım.

Tez çalışmamı, bilgi, araç ve gereç olarak destekleyen İSU GENEL MÜDÜRLÜĞÜ, Genel Müdür Yardımcısı Uğur Susoy, Abone Daire Başkanı Hakan Kütük ve Bilgi İşlem Daire Başkanı Öner Denk'e içten teşekkürlerimi sunuyorum.

Son olarak, eşim Havva'ya, dünyaya gelmeme sebep olan, sabırla yetiştiren, maddi ve manevi desteklerini esirgemeyen babam Halil İbrahim ve annem Melahat'a teşekkür ediyorum.

Mart – 2019

Kasım GÖRENEKLİ

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iii
TABLolar DİZİNİ	iv
SİMGELER VE KISALTMALAR DİZİNİ	v
ÖZET.....	vi
ABSTRACT	vii
GİRİŞ	1
1. SU TEMİN VE TÜKETİM.....	4
1.1. Dünya’da Su.....	4
1.2. Kocaeli’nde Su.....	5
2. AĞ TOPOLOJİLERİ.....	7
2.1. Yol Topoloji.....	7
2.2. Yıldız Topoloji.....	7
2.3. Karmaşık Topoloji	8
2.4. Ağaç Topoloji	9
2.5. Halka Topoloji	9
3. LORA PROTOKOLÜ	12
4. DONANIM VE YAZILIMLAR	14
4.1. Raspberry Pi.....	14
4.2. Kablosuz Haberleşme	17
4.3. İşletim Sistemi	19
4.4. Programlama Dili.....	20
5. GERÇEKLENEN SENARYOLAR	21
5.1. Zayıf Pil Durumu	23
5.2. Dışarıdan Modüle Müdahale.....	23
5.3. Ağa Yeni Bir Modül Eklenmesi	23
5.4. Ağdan Bir Modülün Çıkarılması Veya Arızalanması.....	24
5.5. Erişim Noktası İletişim Modül Seçimi	24
5.6. Erişim Noktası Seçimi	24
5.7. Çok Katlı Binalardaki Veri Aktarımı.....	24
5.8. Sitelerde Veri Aktarımı.....	25
6. SİSTEMİN KURULUMU	27
6.1. İşletim Sistemin Kurulumu	27
6.2. Yerel Ağ (Hot Spot) Oluşturulması	28
6.3. Modüllerin Kablosuz Ağda Veri.....	29
6.4. Radyo Frekans Bağlantısı	33
6.5. Kullanıcı Ara Yüzü	34
7. SONUÇLAR VE ÖNERİLER	36
KAYNAKLAR	37
EKLER.....	41
KİŞİSEL YAYIN VE ESERLER	58
ÖZGEÇMİŞ	59

ŞEKİLLER DİZİNİ

Şekil 1.1	Dünya’da su kaynaklarının dağılımı	4
Şekil 2.1.	Yol Topoloji.....	7
Şekil 2.2.	Yıldız Topoloji.....	8
Şekil 2.3.	Karmaşık Topoloji	8
Şekil 2.4.	Ağaç Topoloji	9
Şekil 2.5.	Halka Topoloji	9
Şekil 3.1	Band genişliği ve mesafe grafiği.....	12
Şekil 3.2.	Kablosuz haberleşmedeki kriterler.	13
Şekil 4.1.	Raspberry Pi 3 Model B Kit	14
Şekil 4.2.	Bağlantı Uçları	15
Şekil 4.3.	Raspberry Pi 3.....	15
Şekil 4.4.	Raspberry Pi’ye Debian İşletim Sisteminin kurulması ve RF modülün eklenmesi.....	16
Şekil 4.5.	Kullanılan Malzemeler	16
Şekil 4.6.	433 MHz RF Gönderici	17
Şekil 4.7.	433 MHz RF Alıcı	17
Şekil 4.8.	433Mhz E32-TTL-100 LoRa SX1278/SX1276 RF Alıcı Verici.....	18
Şekil 4.9.	433Mhz E32-TTL-100 LoRa SX1278/SX1276 RF Uç Anlamları.....	18
Şekil 4.10.	İşletim Sistemlerinden Bazıları.....	20
Şekil 5.1	Sistem Blok Diyagramı.....	22
Şekil 5.2	Sistem Akış Diyagramı	22
Şekil 5.3	Çok Katlı Binalarda Veri Aktarımı.....	25
Şekil 5.4	Sitelerde Veri Aktarım Yönetimi , a) Az katlı bina, b) Çok katlı bina	26
Şekil 6.1.	Raspberry Kurulum Ekranı.	27
Şekil 6.2.	Rasbian İşletim Sisteminin Kurulmuş Görünümü	28
Şekil 6.3.	Bir Modülün Masaüstü Görünümü.	29
Şekil 6.4.	Çalışan Uygulamalar.....	30
Şekil 6.5.	Oluşan Verilerin Kaydedildiği Yer	30
Şekil 6.6.	Ağdaki Aktif Modüllerin IP Adreslerinin Tesbiti.....	31
Şekil 6.7.	Kendisine Bağlı Sayacın Endeks Datasını Saklar.....	32
Şekil 6.8.	Uygulama Sunucu Uygulaması.	33
Şekil 6.9.	Alıcı ve Gönderici Radyo	33
Şekil 6.10.	Alarm Yönetim Ekran Görüntüsü.....	35

TABLÖLAR DİZİNİ

Tablo 1.1. Standart Su Dengesi, IWA.....	5
Tablo 2.1. Topolojilerin Avantaj ve Dezavantajları.....	10
Tablo 3.1. LP-WAN teknolojilerinin karşılaştırılması.....	13
Tablo 4.1. 433Mhz E32-TTL-100 LoRa SX1278/SX1276 RF Alıcı Verici.....	18
Tablo 4.2. 433Mhz E32-TTL-100 LoRa Modülün Mod Anlamları	19
Tablo 5.1. Aktif Modüllerden Okunan Veriler.	23



SİMGELER VE KISALTMALAR DİZİNİ

T_{awm}	: Mikro kontrolün ortalama çalışma süresi, (ms)
$I_{avg.m}$: Mikro kontrolün ortalama akım, (mA)
$I_{avg.t}$: Göndericinin ortalama akım, (mA)
$I_{avg.losses}$: Ortalama kayıp, (mA)
T_{mc}	: Mikro kontrolün çalışma periyodu, (ms)
I_{mc}	: Mikro kontrolün çalışma devre akımı, (ms)
T_{awt}	: Göndericinin ortalama gönderim süresi, (ms)
T_{tc}	: Göndericinin çalışma süresi, (ms)
I_{mc}	: Göndericinin çalışma anındaki akımı, (mA)
C_{new}	: Pilin tam dolun kapasitesi, (mAh)
C_{10y}	: Pilin 10 yılda tahminen kendiliğinden kaybettiği enerji, (mAh)
T_{bath}	: Tahmini modül pilin ömrü, (h)

Kısaltmalar

ABYS	: Abone Bilgi Yönetim Sistemi
BM	: Birleşmiş Milletler
CBS	: Coğrafi Bilgi Sistemi
DMA	: District Metered Area (Ölçülebilir İzole Alan)
DSİ	: Devlet Su İşleri
FAO	: Food and Agriculture Organization of the United Nations (Birleşmiş Milletler Gıda ve Tarım Örgütü)
GWP	: Global Water Partnership (Küresel Su Ortaklığı)
IBYS	: İçme Suyu Bilgi Yönetim Sistemi
IOT	: Internet Of Things (Nesnelerin İnterneti)
ISU	: İzmit Su ve Kanalizasyon İdaresi
IWA	: International Water Association (Uluslararası Su Derneği)
LORA	: Long Range (Uzak Mesafe)
LORAWAN	: Long Range Wide Area Network (Uzak Mesafe Geniş Alan Haberleşme Ağı)
LPWAN	: Low Power Wide Area Network (Düşük Güç Geniş Alan Haberleşme Ağı)
SCADA	: Supervisory Control and Data Acquisition (Kontrol ve Veri Toplama Yönetim Sistemi)
TÜİK	: Türkiye İstatistik Kurumu
UNDESA	: United Nations Department of Economic and Social Affairs (Birleşmiş Milletler Ekonomik ve Sosyal İşler Dairesi)
WWAP	: World Water Assessment Programme (Dünya Su Değerlendirme Programı)

ELEKTRONİK SU SAYACI İÇİN KABLOSUZ AĞ VE RF TABANLI MİMARİ TASARIM

ÖZET

Su, olmazsa hayat olmaz felsefesi gereğince tüm canlılar için vazgeçilmez yaşam kaynağıdır. İnsanoğlu, yaratılışın kaynağı olan su ile başlar. Büyük Medeniyetler hep su çevresinde kurulmuştur. Sarı Irmak Çin'e, Ganj Hindistan'a, Nil Mısır'a, Fırat ve Dicle de Mezopotamya'ya ev sahipliği yapmıştır. Hızla büyüyen dünya nüfusunun sosyal, kültürel ve ekonomik alışkanlıkları zaman ve mekana göre değişmeye başlamıştır. Gün geçtikçe suya olan ihtiyaç artmaktadır. Suyun sürdürülebilir olması için de suyun daha çok iktisatlı kullanılması gerekmektedir. Türkiye'nin coğrafi konum anlamında kişi başına düşen su miktarının dünya ortalamasının altında olduğu düşünüldüğünde su yönünden zengin olduğu söylenemez. Mevcut su kaynaklarının korunması, verimli kullanılması ve sürdürülebilir olması gerekir. Su saatlerin kişiler tarafından okunması zaman ve kaynak israfına (insan gücü ve yakıt), yüksek hata yapma olasılığına sebep olmaktadır. Kablosuz haberleşme ile bu bahsedilenlerin oluşması engellenmektedir. Uzaktan sayaç okuma kullanıcıların anlık tüketimlerini görmesini ve bulunduğu bölgedeki ortalama tüketimle kıyaslama imkanı mevcut. Bu sayede hatlardaki su basıncı takibi ve bu sayede sistemi iyileştirme imkanı olacaktır.

Anahtar kelimeler: Akıllı Su Sayacı, Kablosuz ve Radyo Dalgalı Ağ, Su Kayıp Kaçak, Sürdürülebilir ve Kaliteli Hizmet, Uzaktan Sayaç Okuma.

WIRELESS NETWORK AND RF BASED ARCHITECTURE DESIGN FOR ELECTRONIC WATERMETER

ABSTRACT

Water is the indispensable source of life for all living beings. Human starts with water, which is the source of creation. Great civilizations have always been established around the water. The Yellow River was home to China, Ganj to India, Nile to Egypt, Euphrates and Tigris to Mesopotamia. The social, cultural and economic habits of the rapidly growing world population began to change according to time and place. The need for water is increasing day by day. Water must be used more economically in order to be sustainable. Water per capita in Turkey is below the world average in terms of geographical location under the direction said to be rich in water. Manual reading water meter is inconvenient, low accurate, time consuming, very slow and resource wasting (like human force and fuel). Using wireless communication will remove these unwanted states. A smart water meter monitoring enables users to watch their water consumption in real-time and having the opportunity to compare to the average water consumption of that district and minimize water usage. With this system water pressure of whole pipes could be observed.

Keywords: Smart Water Meter, Wireless and RF Network, Water Loss, Sustainable and Qualified Service, Remote Meter Reading.

GİRİŞ

Otomatik ölçüm sistemleri, bilim ve teknolojinin insanlığa sunduğu hayatı kolaylaştırıcı sistemlerden biridir. Dünya nüfusunun gün geçtikçe artması ve su teminin kısıtlı olması değişik çözümlere yönelmemize sebep olmaktadır. Bu sistemlerin kullanılması zamandan tasarruf sağlar, beşeri hata faktörlerini en aza indirir, maliyetleri düşürür ayrıca da analiz ve yönetim kolaylığı sağlar. Akıllı su izleme sistemi, kullanıcıların su tüketimine dikkat etmelerini sağlar ve su kullanımını azaltmalarına yardımcı olur [1]. Aynı zamanda, kullanıcılar anormal su kullanımı konusunda uyarılırlar. Su tüketiminin iktisatlı kullanımı ve kaçakların aza indirgenmesi kaçınılmaz olmaktadır [2]. Radyo frekansı ve kablosuz ağ ile veri gönderimi üzerine çalışma yapılmıştır. Bu sayede düşük enerji tüketim ile uzaktan anlık su tüketimi, geri basma ve cihaza müdahale olup olmadığını anlık olarak uzaktan takip edip, gerekli müdahalenin anında yapılabilmesi sağlanabilmektedir [3 - 4]. Bir diğer avantaj ise, elektrik, gaz ve diğer kamusal hizmetler gibi teknolojik olarak gelişmekte olan nesnelerin internetinden (Internet Of Things, IOT) faydalanılabilecektir [5-7].

Mayıs 2014'te Orman ve Su İşleri Bakanlığı tarafından yayımlanan yönetmelik "Büyükşehir Belediyeleri sorumluluk alanlarındaki içme suyu kayıplarını ilk 5 yıl içinde %30, takip eden 4 yıl içerisinde ise %25 seviyesine indirmelidir" gereğince Su kurumları çalışmaları başlatılmıştır. Türkiye su kayıp ortalaması %45'tir. Bu Yönetmelik uyarınca ISU olarak Türkiye genelinde başlatılmış ilk proje Kocaeli İli İzmit İlçesi İçme Suyu Kayıplarının Azaltılması Projesi'dir. Abonelere kesintisiz hizmet verebilmek, su dağıtım ağı yönetimini interaktif izlemek, basınç kontrolü sağlamak ve abonelere yeni hizmet sunmak üzere çalışmalar başlatılmıştır. ABYS, IBYS, CBS, SCADA ile DMA bölgelerden elde edilen veriler üzerinde çalışma yapıp durum değerlendirmesi yapılacaktır.

A.Zabasta ve N.Kunicina'nın çalışmasında su dağıtım ağlarında kablosuz sensörlerin kullanımı incelenmiş olup kablosuz haberleşmedeki enerji tüketimi ve verinin

tutarlılığı üzerine durulmuştur. Gateway'in sayaca ulaşamadığı noktalara sabit elektrikli yönlendiriciler (routerlar) eklenerek iletişim sağlanmıştır [6].

Mudumbe ve Mahfouz'un Kullanıcı merkezli tüketim ölçümü için akıllı su sayacı sistemi konulu çalışmasında suyun daha iktisatlı kullanılması kablosuz haberleşme hakkındadır. Fiziksel kablo aracılığıyla verinin alınması zaman ve maliyet açısından pratik olmadığı görülmektedir. Alternatif olarak kablosuz haberleşme ile gateway'lere bağlanmasıdır. LDBRS(Low Data Bit Rate Sense) IEEE 802.15.4 protokolü ile Zigbee'yi kullanmaktadır [7].

Zhang, Qian, Yan ve Hu'nun düşük enerji sarfiyatlı kablosuz algılama ağ kullanımı hakkındaki çalışması incelenmiştir. Mobil adhoc ağ'a kıyasla düşük enerji tüketimi için iyileştirilmiş protokol kullanılmaktadır. OPNET uygulamasında yapılan simülasyon analiz edildiğinde iyileştirilmiş protokolün kullandığı enerji tüketimin belirgin bir şekilde düştüğü gözlenmiştir [8].

Emre Yüksel'in çalışmasında ZigBee kablosuz ağları üzerinden sayaç okuması incelenmiştir. Binalardaki su sayaçların tüketim verilerini okuyup merkezi sisteme aktaran kablosuz ölçüm sistemi ZigBee kullanarak tasarlanmıştır [9].

Emre Sami Süzer'in "Uzaktan Sayaç Okuma Teknikleri ve MODBUS-TRU, IEC 61107 MOD C Protokolleri ile Örnek Yazılım" konulu tezinde RS-232, RS-485, GPRS, RF, Fiber optik kablo ve enerji hattı üzerinden haberleşme hakkında araştırmalar yapılmıştır. Uzaktan sayaç okuma sistemlerinde ve endüstriyel sistemlerde çok sık olarak kullanılan iki protokol olan Modbus-RTU ve IEC 61107 Mod C haberleşme protokolleri incelenmiştir [10].

Muhendra, Husein, Budiman, ve Khairurijal "Development of Digital Water Meter Infrastructure using Wireless Sensor Networks" konulu çalışmada mevcut sistemin zayıf taraflarının, hata, erişim zorluğu ve geçen zamanın azaltılması hedeflenmiştir. Toplanan nabız sayıları 2.4 GHz de çalışan Nordic NRF24 tarafından gönderilmektedir. Karmaşık topolojiye de oldukça uygun olduğu görülmektedir [11].

Ahmad, Asad, Muhammad ve Anwar "Development of a Low-Power Smart Water Meter for Discharges in Indus Basin Irrigation Networks" konulu tezinde toplanan

veriler ultrasonic algılayıcı tabanlı ve GPRS vasıtasıyla merkeze gönderilmesi geliştirilmiştir. Pakistan'daki Indus Nehri Havzası, dünyanın en büyük bitişik sulama ağına sahiptir ve 90.000 Km'nin üzerinde su yolu ve yaklaşık 25 milyon dönümlük sulaması yapılmaktadır. Bu kadar büyük bir sistem, otomasyon ve karar destek sistemlerinin kullanımı olmadan yüksek verimlilikle yönetilemez [12].

Spinsante, Pizzichini, Mencarelli, Squartini, ve Gambi “Evaluation of the Wireless M-Bus standard for future smart water grids” konulu çalışmalarında özellikle kablosuz haberleşmede güç tüketimine değinilmiştir. Borular boyunca kullanılacak algılayıcıların (sensör) çalışması için gerekli enerjiyi asgariye düşürme üzerine çalışma yapılmıştır. Düşük enerji kullanımı olarak 868 Mhz, 468 Mhz ve 169 Mhz de çalışabilen WM-Bus üzerine odaklanılmıştır. [13].

Domoney, Ramli, Alarefi, ve Walker “Smart city solutions to water management using self-powered, low-cost, water sensors and apache spark data aggregation” konulu tezlerinde akıllı şehir oluşumundan bahsedilmiştir. Özellikle ihtiyaç duyulan enerjinin sağlanması adına su akıntısından istifade edilerek 4 Watt'a kadar üretilebilen çalışma yapılmıştır [14].

Phuong, N.M., Schappacher, M., Sikora, I.A., Ahmah, Z., and Muhammad, D.A. “Real-time Water Level Monitoring Using Low-Power Wireless Sensor Network” konulu çalışmada $\frac{3}{4}$ 'ü ekim arazisi olan pakistanda suyun oldukça iktisatlı kullanılması için gerekli teknolojiden bahsedilmiştir. Gelen sinyal ile uyanma (Wake-On-Radio, WOR) destekli çok düşük enerji kullanan IEEE-802.15.4 tabanlı kablosuz haberleşme üzerine çalışma yapılmıştır [15].

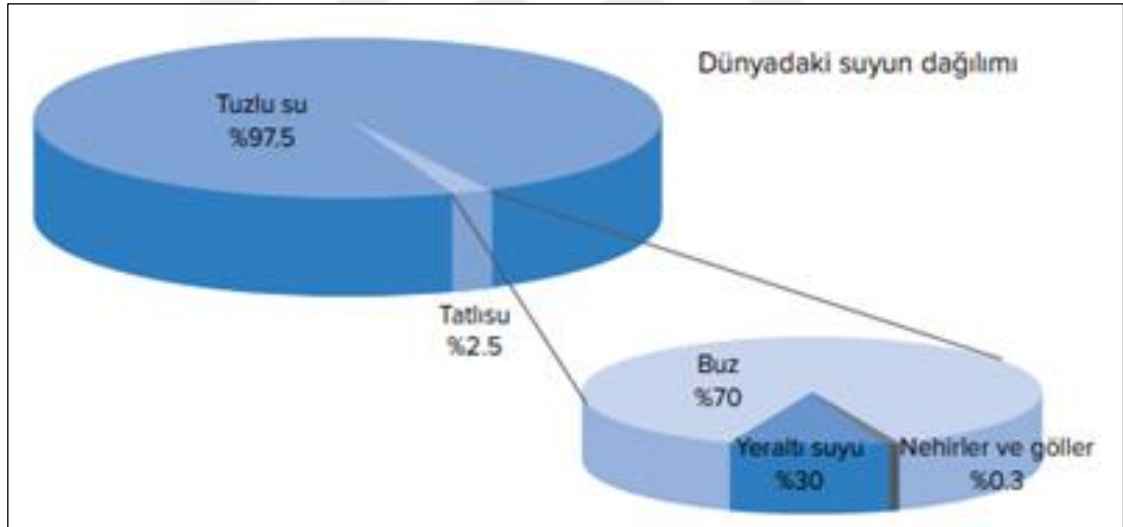
Bölüm 1'de, hayat için zaruri olan su hakkında temel bilgiler anlatılmıştır. Bölüm 2'de, Ağ topolojilerden bazıları ele alınmış, avantaj ve dezavantajlar kıyaslanmıştır. Data gönderim için en önemli kriter olan pil ve uzak mesafe için LoRa Protokolü Bölüm 3'de incelenmiştir. Sistemi oluşturan cihaz, haberleşme, işletim sistemi ve programlama dili olan 4 temel konuya Bölüm 4'de değinilmiştir. Bölüm 5'de ise, oluşabilecek senaryolar ve çözümler anlatılmıştır. Uygulamalı olarak sistemi çalışır hale getirmek için uygulanacak adımlar Bölüm 6'da anlatılmıştır. Bölüm 7'de ise, bu tezden çıkarılan sonuç belirtilmiş ve gelecekte iyileştirme olarak yapılabileceklerle değinildi.

1. SU TEMİN VE TÜKETİM

1.1. Dünya’da Su

Dünyanın 4 te 3ü sularla kaplı olmasına rağmen %1 den azı ekosistem ve insan kullanımına elverişlidir. Şekil 1.1’de gösterilen FAO Aquastat’ın 2013 yılındaki araştırmasına göre su kaynakların %70’i tarım amaçlı, %19’u sanayi ve %11’i de evsel de kullanılmaktadır. UNESCO’nun 2000 yılındaki raporuna göre gelişmiş ülkelerde kişi başı su tüketimi yaklaşık 650 m3 iken gelişmekte olan ülkelerde bu rakam yaklaşık 50m3 tür [16] .

Dünyada ki suların sadece %2,5’u tatlı su. Bununda yaklaşık %70’i buz, %30’u yeraltı suyu %0,3ü ise nehirler ve göllerdir.



Şekil 1.1 Dünya’da su kaynaklarının dağılım

GWP’nin 2000 yılındaki araştırmasına göre tüm uzmanlar su kaynaklarının korunması hakkında Entegre Su Kaynakları Yönetimi’nin vazgeçilmez üzerine hem fikir olmalarıdır. WWAP’nin 2012 yılındaki araştırmasına göre son 50 yılda su kaynakların miktarı değişmemesine rağmen su çekimi üç katına çıkmıştır. 2030 yılında gıda, su ve enerji ihtiyaçları %50 civarında artacağı tahmin edilmektedir. Su kaynakların su miktarları zamanla azalmakta ve hatta yok olmakta. 1990’lı yıllarda

350 km² lik bir alanı kaplayan Akşehir gölü günümüzde tamamen kurumuştur. 2012 yılında 7.1 milyar olan Dünya nüfusu 2030 yılında 8.3 milyar olması tahmin edilmektedir. UNDESA'nın 2009 yılındaki araştırmasına göre nüfusunun %60'ının kentlerde yaşaması tahmin edilmektedir.

2013 yılını “Uluslararası Su Dayanışma Yılı” olarak ilan eden Birleşmiş Milletler (BM) Genel Sekreteri Ban Ki-Moon “Sağlık, gıda güvenliği ve ekonomik kalkınma için suya ihtiyacımız var” söyleyerek sürdürülebilir kalkınmanın temel anahtarının suda olduğunu vurgulamıştır.

DSİ verilerine göre 2015 yılında ülkemizde kişi başına düşen su miktarı yıllık 1.519m³ iken 2030 yılında TÜİK verilerine göre 1.120m³ tahmin edilmektedir[17]. 2050 yılında Dünya nüfusu yaklaşık 9 Milyar civarında olacaktır. Dolayısıyla su tüketimindeki kayıplar en aza indirgenmelidir [18].

1.2. Kocaeli’nde Su

Kocaeli’nde durum nasıl diye bir bakalım. TÜİK’e göre Kocaeli’nin nüfusu 2015 yılında 1.7 milyon iken 2023 yılında 2 milyon olması beklenmektedir. 2013 yılında 130 milyon m³ su temin edilmiştir. 2006 yılında 491 bin abonesi var iken şuan 794.000 abonesi mevcut.

Tablo 1.1. Standart Su Dengesi, IWA

Sisteme Giren Su	İzinli Tüketim	Faturalandırılmış İzinli Su Tüketimi	Faturalandırılmış Ölçülmüş Kullanım	Gelir Getiren Su Miktarı
			Faturalandırılmamış Ölçülmemiş Kullanım	
		Faturalandırılmamış İzinli Su Tüketimi	Faturalandırılmamış Ölçülmüş Kullanım	Gelir Getirmeyen Su Miktarı

Tablo 1.2. (Devam) Standart Su Dengesi, IWA

Sisteme Giren Su	Su Kayıpları	İdari Kayıplar	Faturalandırılmamış Ölçülmemiş Kullanım	
			İzinsiz Tüketim	
		Fiziki Kayıplar	Sayaçlardaki Ölçüm Hataları	
			Şebekede Oluşan Kayıp-Kaçaklar	
Depolarda Meydana Gelen Kaçak ve Taşmalar				

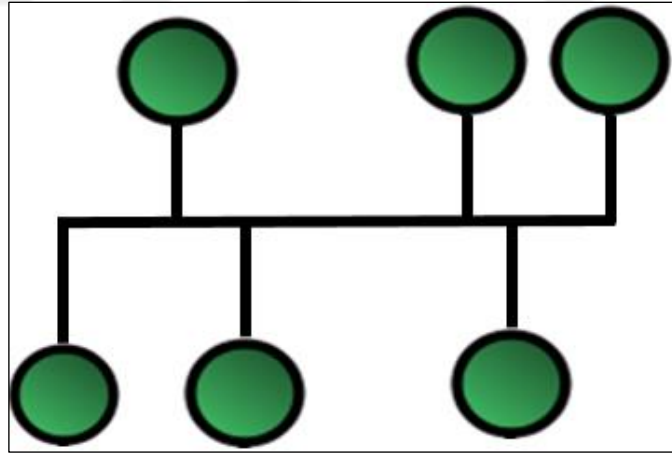
Tablo 1.1’de ISU kurumunun Su Dengesi gösterilmiştir. 1991 yılında Uluslararası Su Sevisi Birliği’nin (IWSA) yaptığı çalışmada; Gelişmişlik durumuna göre ülkelerin su kayıp oranları karşılaştırılmıştır. Buna göre gelişmiş ülkelerde su kaybı oranı %8-24, gelişmekte olan ülkelerde %25-45 seviyesindedir. TÜİK verilerine göre ülkemizde 2012 yılı verilerine göre çeşitli nedenlerle su kayıp oranı %43 civarında dır.

2. AĞ TOPOLOJİLERİ

Akıllı su sayaçları hassas ve yüz binlerce abone verilerinin sorunsuz transfer edilebilmesi için oldukça hızlı olmalı. Çift yönlü iletişim destekli olan bu sistemi sağlayacak firmalar coğrafi yüzey şekillerini göz önünde bulundurmaları. Gecikmesiz, hızlı, hassas, güvenilebilen ve şifreleme özelliklerinin olması kaçınılmaz. Cihazlar birbirleriyle değişik şekillerde iletişim kurabilir. Ağ topolojileri denilen bu bağlantı şekillerden en çok kullanılan 5 tanesine burada değineceğiz [19-21].

2.1. Yol Topoloji

Şekil 2.1’de gösterildiği gibi bu topolojide tüm cihazlar bir ana hatta bağlıdır. Hat üzerinde olan cihazlar veri gönderebilmeleri için hattın boş olduğundan emin olmalı. Yoksa veri çakışması oluşur. Bu sistemde veri tüm cihazlara gönderilir. Ancak veriyi hedef cihaz alır. Bu durumda da performans kaybı söz konusudur.

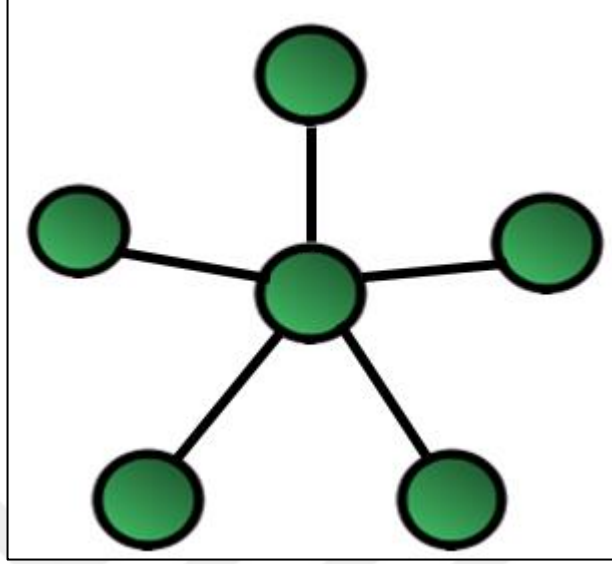


Şekil 2.1. Yol Topoloji

2.2. Yıldız Topoloji

Şekil 2.2’de gösterildiği gibi bu topolojide tüm cihazlar bir cihaza, switch (anahtar) veya çoklayıcı’ya (hub) bağlanır. Çoklayıcılar veriyi Yol topolojisinde olduğu gibi tüm bağlı cihazlara gönderirler. Anahtarlar ise kendine bağlı tüm cihazların makine

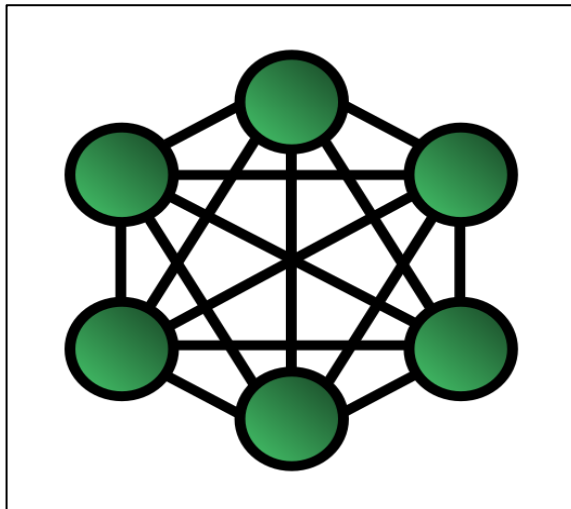
adreslerini (MAC) kaydeder ve veriyi sadece hedef cihaza gönderir. Bu yüzden anahtarlar daha güvenilir ve hızlıdır.



Şekil 2.2. Yıldız Topoloji

2.3. Karmaşık Topoloji

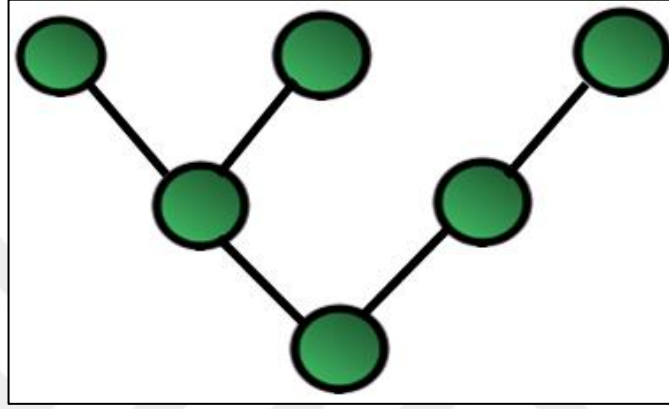
Bu topolojide her cihazlar birbirlerine bağlıdır. Şekil 2.3’de gösterildiği gibi her hangi bir bağlantı kopsa bile sistem bütünlüğü bozulmaz. Bu yapı topolojiler içinde en güvenilir olarak da bilinir. 2 nokta arasında veri alış verişinden ağdaki diğer cihazların haberi olmaz. Cihaz sayısı çok fazla ise bağlantı sayısı da logaritmik olarak artacağından çok fazla cihazı olan ağlarda bu topoloji tercih edilmemektedir.



Şekil 2.3. Karmaşık Topoloji

2.4. Ağaç Topoloji

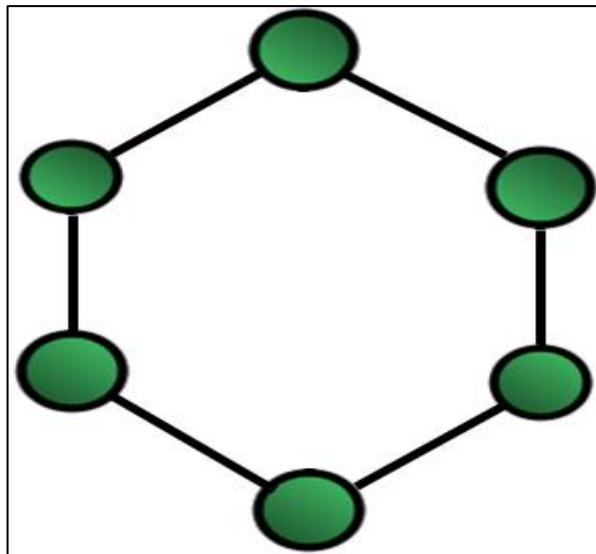
İsminden de anlaşılacağı gibi Şekil 2.4 deki ağaç yapısında olan bu topoloji diğer topolojilerle birlikte çalışabilir. Ağacın her bir dalında farklı bir yapı kullanılabilir. Örneğin bir dalında karmaşık topoloji kullanılırken diğer bir dalda yıldız topolojisi kullanılabilir. Ancak ana gövde de oluşan her hangi bir sorun tüm sistemin çalışmasını etkiler.



Şekil 2.4. Ağaç Topoloji

2.5. Halka Topoloji

Bu topolojide her bir cihaz sadece 2 cihaza bağlanarak bir çember oluştururlar. Gönderilen veri hedefe varana kadar cihazlar üzerinden aktarılır. Şekil 2.5’de gösterildiği gibi her hangi bir cihaz bozulursa tüm ağ bağlantısı kesilir.



Şekil 2.5. Halka Topoloji

Tablo 2.1. Topolojilerin Avantaj ve Dezavantajları

Topolojiler	Avantaj	Dezavantaj
Yol	<ul style="list-style-type: none"> - Bilgisayarlar ve cihazlar ağa kolayca bağlanır - Az kablo kullanılır. - Kolay tasarım ve genişletme imkanı. - Kolay kurulum için ideal, daimi ve geçici ağ kullanım için uygun. - Cihaz, anahtar ve dağıtıcı kullanımı kolay olduğundan ucuz maliyetlidir. - Bir istasyonun çalışmaması ağın çalışmasını engellemez. - Geliştirme açısından en ucuz topolojidir. 	<ul style="list-style-type: none"> - Yönetim ve sorun giderme zordur. - Kısıtlı sayıda istasyon ve kısa mesafeli kablo kullanılabilir.. - Ana kabloda sorun olursa tüm sistem çalışmaz, tamamen devre dışı kalır. - Her ilave istasyon genel ağ performansını kötü etkiler. - Kablonun her iki ucunda da sonlandırma yapılması zorunludur.
Yıldız	<ul style="list-style-type: none"> - Yeni istasyonlar sisteme kolay eklenir. - Yönetim ve hata tespiti oldukça kolay ve kısa sürede hatanın giderimi mümkün. - Değişik kablolama metotlarıyla uyum sağlar. - Bir istasyonda sorun olduğunda sistem etkilenmez. - Yeni istasyon eklendiğinde de ağ performansı etkilenmez. 	<ul style="list-style-type: none"> - Diğer topolojilere kıyasla oldukça fazla kablo kullanılır. - Anahtar veya dağıtıcıdaki sorun tüm istasyonları etkiler. - Bu cihazları kullanmak Yol topolojisini kullanmaktan daha fazla maliyetlidir.
Karmaşık	<ul style="list-style-type: none"> - Her bir istasyon sondan sona diğer istasyonlarla bağlantı kurduğu için çoklu erişim mümkün olmakta. Şayet bir bağlantı koparsa erişim diğer bağlantılardan sağlandığı için oldukça kullanılışıdır. - Şayet iki istasyon arasındaki veri transfer zamanı çok ise diğer alternatif bağlantılar kullanılarak bu süre aza indirgenmiş olur. 	<ul style="list-style-type: none"> - İstasyon sayıları arttıkça bağlantı sayıları logaritmik olarak artmaktadır. Çoklu istasyonlarda bu topoloji önerilmez.

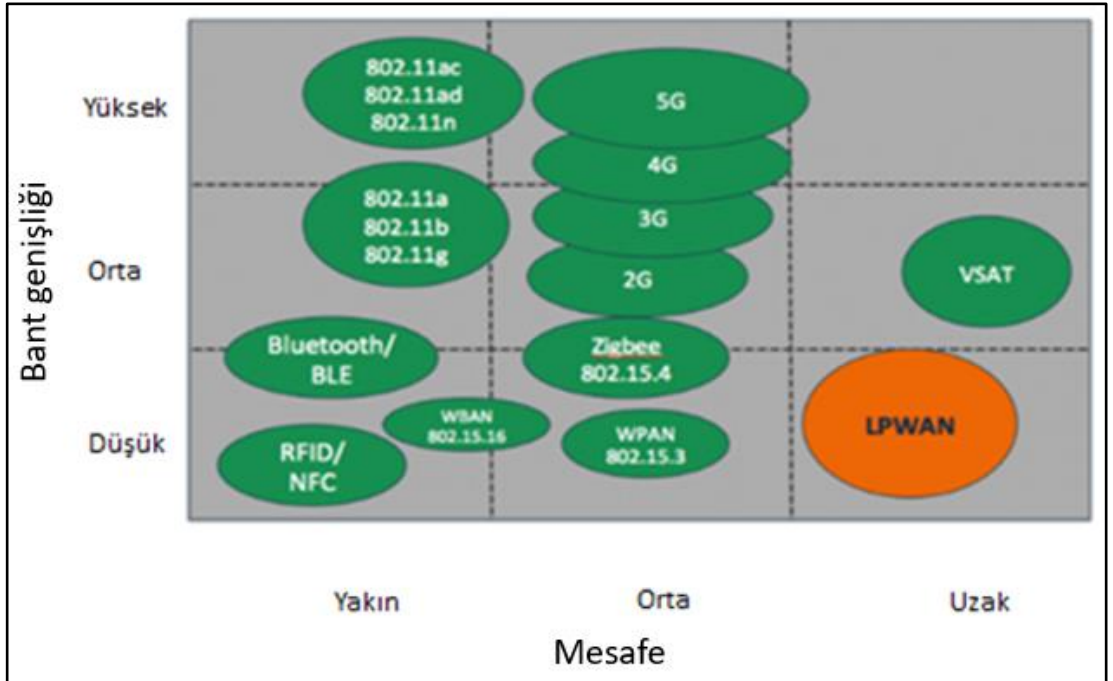
Tablo 2.2. (Devam) Topolojilerin Avantaj ve Dezavantajları

Ağaç	<ul style="list-style-type: none">- Noktadan noktaya bağlantı olduğundan her hangi bir sorunda diğer noktalar etkilenmez.- Farklı donanım ve yazılım firmalarına ait donanımlar birlikte sorunsuz çalışır.	<ul style="list-style-type: none">- Kablolardaki bağlantı tiplerine göre cihazlar arasında kullanılan kabloların uzunluğu belirli bir rakamdan daha uzun olamaz.- Şayet ana omurgada bir sorun olursa tüm sistem devre dışı kalır.- Kablolama konfigürasyonu diğer topolojilerden daha çok zordur.
Halka	<ul style="list-style-type: none">- İstasyon sayısının artırımı genel ağ performansını oldukça az etkiler.- Her istasyonun eşit erişim hakkı vardır.	<ul style="list-style-type: none">- Bilinen en çok maliyetli topolojidir.- Oldukça karmaşık yapıya sahiptir.- Her hangi bir istasyonda sorun olursa tüm sistem etkilenir.

Tablo 2.1’de bu topolojilerin avantaj ve dezavantajları gösterilmektedir. Kablosuz haberleşmede yıldız, ağaç ve karmaşık topoloji üzerine çalışma yapılmıştır [23].

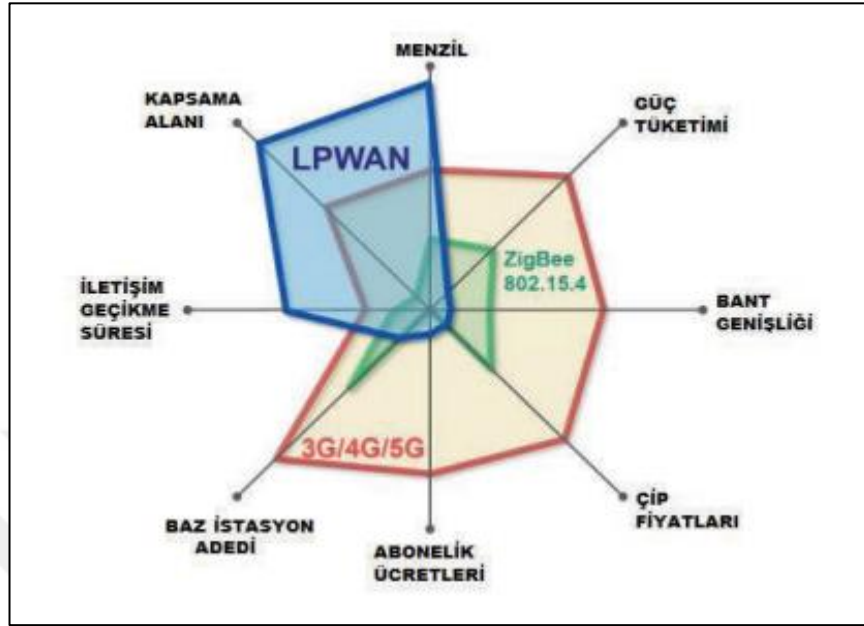
3. LORA PROTOKOLÜ

Lora, uzak mesafenin kısaltılması (Long Range) anlamına gelir. Kendine has özelliği ile 169Mhz, 433Mhz, 868Mhz ve 915Mhz frekansta uzak mesafelere veri gönderebilen fiziksel katman iletişim protokolüdür. Açık bir platform olarak LoRaWAN'ı duyurmuş. Lora'ya ortam yönetimi, güvenlik, GPS'siz lokasyon ve benzeri bir çok özellik eklenmiş. LoRaWAN bir LPWAN (Low Power Wide Area Network) teknolojisidir. Bu network çok az güç kullanarak uzun menzilli iletişim sağlar. 0,3kbps'den 50kbps ye kadar veri hızı sağlayarak 2 yönlü güvenli bir haberleşme sağlar. Ek-B de gösterildiği gibi pilin yaklaşık ömrü, modülün 15 dakikada 23 byte uzunluğunda paket gönderilmesi baz alındığında, 21 yıldan fazladır. IoT (Internet of Things) için vaz geçilmez bir alt yapıdır. Amsterdam, LoraWAN ile tam kapsama sağlayan 10 adet ağ geçitli IoT veri ağı kullanan ilk şehirdir[6].



Şekil 3.1 Band genişliği ve mesafe grafiği.

Link Labs'ın 2015 yılı araştırmasına göre Şekil 3.1.'de gösterildiği üzere en önemli özelliği düşük enerji ile uzak mesafelerle haberleşmenin yapılabilmesidir.



Şekil 3.2. Kablosuz haberleşmedeki kriterler.

Şekil 3.2'de görüldüğü üzere modül tasarımı yüksek binalar arasından iletişim kurulacağı için engel aşımı, uzak mesafe iletişimi ve gecikme parametreleri ön plana çıkmaktadır. Bu sebeplerden ötürü LPWAN tercih edilmiştir [24].

Tablo 3.1. LP-WAN teknolojilerinin karşılaştırılması

Özellik / Teknoloji	LTE-M	NB-IoT	LoRA	Sigfox
Düşük Güç	Orta	İyi	Orta	İyi
Bant Genişliği	İyi	İyi	İyi	Orta
Açık Standart	Orta	Orta	Orta	Orta
Tedarikçi genişliği	İyi	İyi	Orta	İyi
Güvenlik	İyi	İyi	Orta	Orta
Yayılm	Yok	Orta	İyi	İyi

Tablo 3.1'de görüldüğü üzere yayılım olarak LoRA ve Sigfox teknolojileri öne çıkmaktadır. Dezavantaj olarak Sigfox kullanıcıya sertifikalı modem ve web tabanlı uygulama sunmaktadır [25].

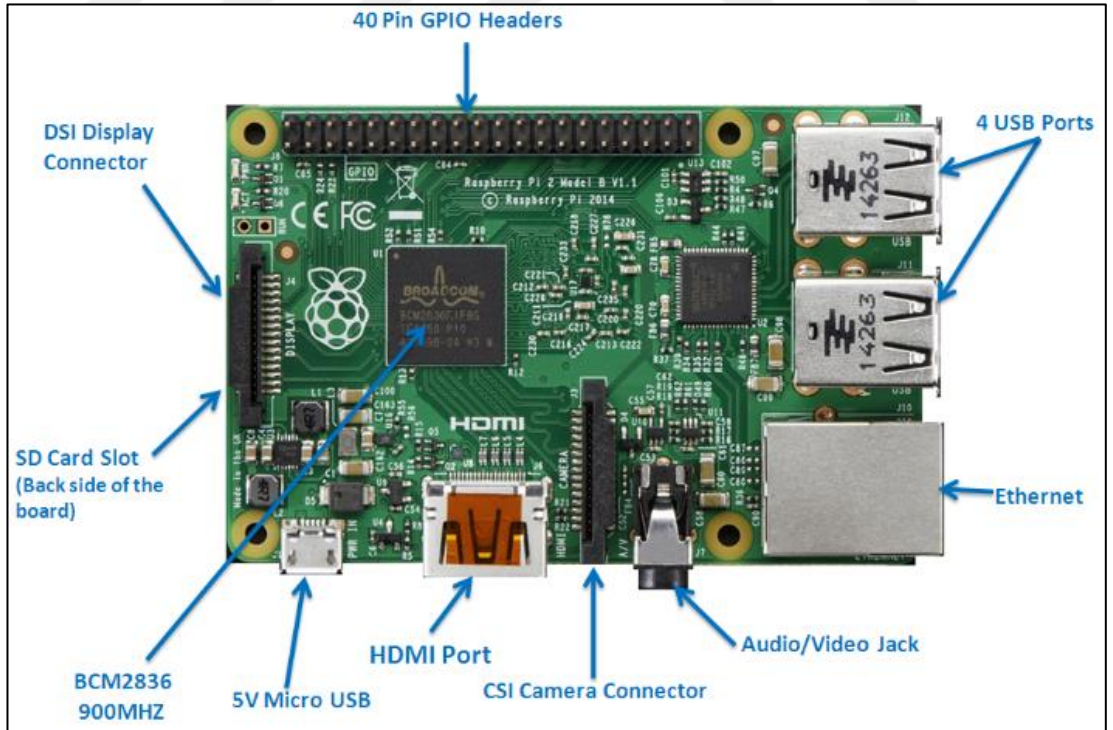
4. DONANIM VE YAZILIMLAR

Sistemi oluşturan donanım ve yazılımlar 4 başlık halinde anlatılmıştır.

4.1. Raspberry Pi

PIC, Arduino ve buna benzer bir çok geliştirme donanımları mevcut. Bu çalışmada geniş bir kütüphaneye sahip ve oldukça esnek olan Raspberry Pi kullanılmıştır. Raspberry Pi, Birleşik Krallıkta Raspberry Pi Vakfı tarafından 2009 yılında ilk üretilmeye başlanılmış, bankamatik kartı boyutlarında bir bilgisayardır. En ucuz bilgisayar modülü ünvanı alan bu ürün 2016 Mart ayında WIFI ve Bluetooth özelliğiyle Raspberry Pi 3 model B sürümünü piyasaya sürmüştür[26-27].

HDMI görüntü çıkışı yuvasıyla monitör veya televizyona bağlanabilir. Ayrıca 4 adet USB ile 1 adet Ethernet portu mevcut.

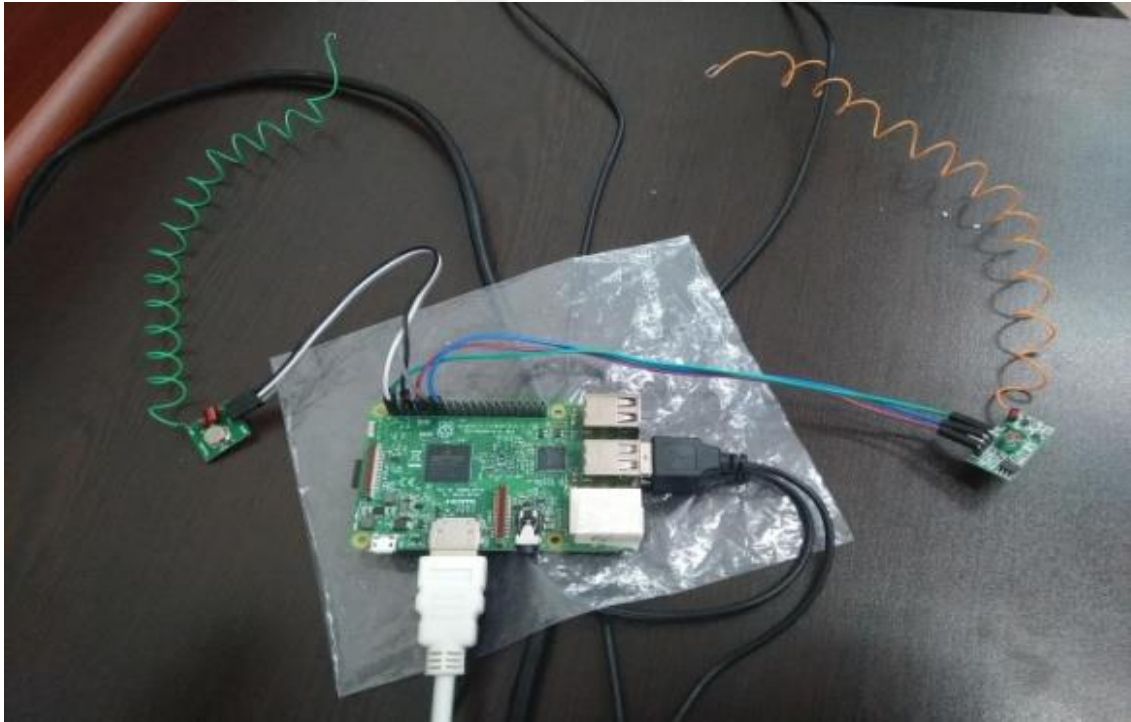


Şekil 4.1. Raspberry Pi 3 Model B Kit

5V Power	2	5V Power	4	Ground	6	GPIO14 UART0_TXD	8	GPIO15 UART0_RXD	10	GPIO18 PCM_CLK	12	Ground	14	GPIO23	16	GPIO24	18	Ground	20	GPIO25	22	GPIO8 SPI0_CEB_N	24	GPIO7 SPI0_CEL_N	26	ID_SD DC ID EEPROM	28	Ground	30	GPIO12	32	Ground	34	GPIO16	36	GPIO20	38	GPIO21	40
3V3 Power	1	GPIO2 SDA1 I2C	3	GPIO3 SCL1 I2C	5	GPIO4	7	Ground	9	GPIO17	11	GPIO27	13	GPIO22	15	3V3 Power	17	GPIO10 SPI0_MOSI	19	GPIO9 SPI0_MISO	21	GPIO11 SPI0_CSCLK	23	Ground	25	ID_SD DC ID EEPROM	27	GPIO5	29	GPIO6	31	GPIO13	33	GPIO19	35	GPIO26	37	Ground	39

Şekil 4.2. Bağlantı Uçları

Şekil 4.1 ve Şekil 4.2 de Raspberry pi 3'ün resmi ve genişletme için kullanılan 40 adet uçların açıklamaları gösterilmektedir. Sistem kurulumu için öngörülen asgari 4GB kapasiteli MicroSD kart. Bu tezde 16GB hafıza kartı kullanıldı. Bir paketin boyutunun 45 Byte olduğu ve saatte bir veri saklandığı düşünüldüğünde 100MB'lık bir diskte 100 yıllık veri saklanması mümkün.



Şekil 4.3. Raspberry Pi 3



Şekil 4.4. Raspberry Pi'ye Debian İşletim Sistemin kurulması ve RF modülün eklenmesi



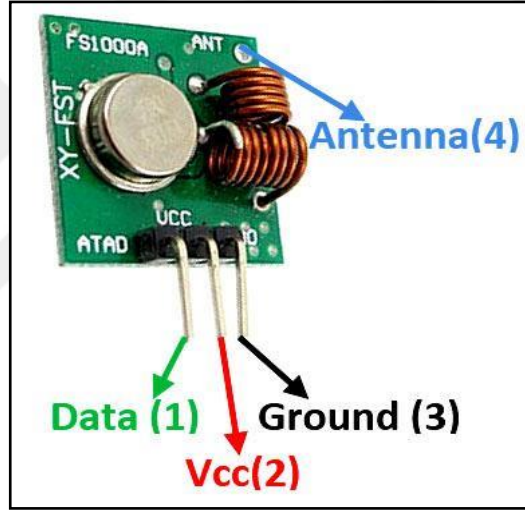
Şekil 4.5. Kullanılan Malzemeler

Şekil 4.3, Şekil 4.4 ve Şekil 4.5'de Raspberry Pi 3'ün montajı gösterilmiştir.

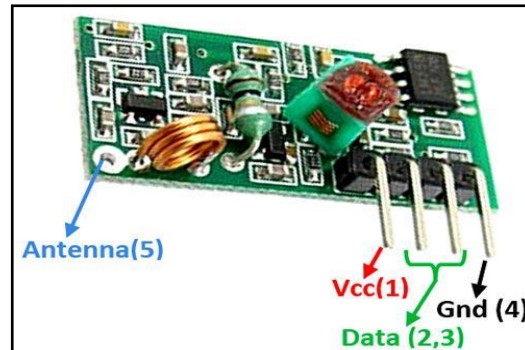
4.2. Kablosuz Haberleşme

Kablosuz haberleşme, her hangi bir kablo olmadan iki veya daha fazla nokta arasında bilginin aktarılması. Kızılötesi (IrDA), Mikrodalga, Radyo Yayını, Uydu, WIFI, RF, Bluetooth, NFC kablosuz haberleşme tekniklerin bazıları. Tezde WIFI ve RF kullanılmıştır.

Şekil 4.6'da RF gönderici, Şekil 4.7'da RF dinleyici gösterilmiştir. Şekil 4.8'de SEMTECH firmasının ürettiği, 433Mhz frekansında çalışan E32-TTL-100 LoRa SX1278/SX1276 RF alıcı ve verici modülü kullanılmıştır. Bu modülün uç tanımları Şekil 4.9'da gösterilmiştir. Bu modülün teknik özellikleri Tablo 4.1'de ve nasıl programlandığı ise Tablo 4.2'deki tablolarda gösterilmiştir.



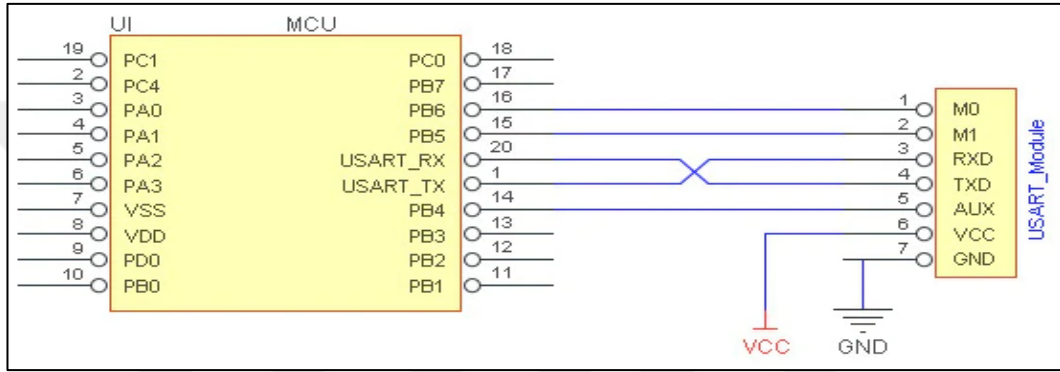
Şekil 4.6. 433 MHz RF Gönderici (Transmitter) Modül



Şekil 4.7. 433 MHz RF Alıcı (Receiver) Modül



Şekil 4.8. 433Mhz E32-TTL-100 LoRa SX1278/SX1276 RF Alıcı Verici



Şekil 4.9. 433Mhz E32-TTL-100 LoRa SX1278/SX1276 RF Uç Anlamları

Tablo 4.1. 433Mhz E32-TTL-100 LoRa SX1278/SX1276 RF Alıcı Verici

RF Parametre	Değeri	Açıklama
Çalışma frekansı	410 ~ 441 MHz	Varsayılan: 433 MHz
Verici güç	10 ~ 20 dBm	Varsayılan: 20dBm (Yaklaşık 100 mW)
Alıcı hassasiyeti	-146 dBm	Hava veri hızı: 0.3 kbps
Hava veri hızı	0.3k ~ 19.2 kbps	Varsayılan: 2.4 kbps
Testi mesafesi	3000 metre	Açık ve net hava, ile maksimum güç, 5dBi anten kazanç, yüksekliği 2 m, Hava veri hızı: 2.4 kbps
Verici Akımı	110mA (20dBm)	
Alma Akımı	14.5mA (mod 0 veya mod 1)	Minimum 30uA (mod 2 + 2 s uyandırma süresi)
Hassasiyet	-130dbm'de 0.3 kbps'de	
Anten Tipi	SMA-K	harici iplik deliği, 50 ohm empedans
Boyutu	21*36mm	Olmadan SMA
Paketi	DIP	
Tampon	512 bayt tampon	Otomatik olarak alt-58 ile ambalaj bayt başına paketi

Tablo 4.2. 433Mhz E32-TTL-100 LoRa Modülün Mod Anlamları

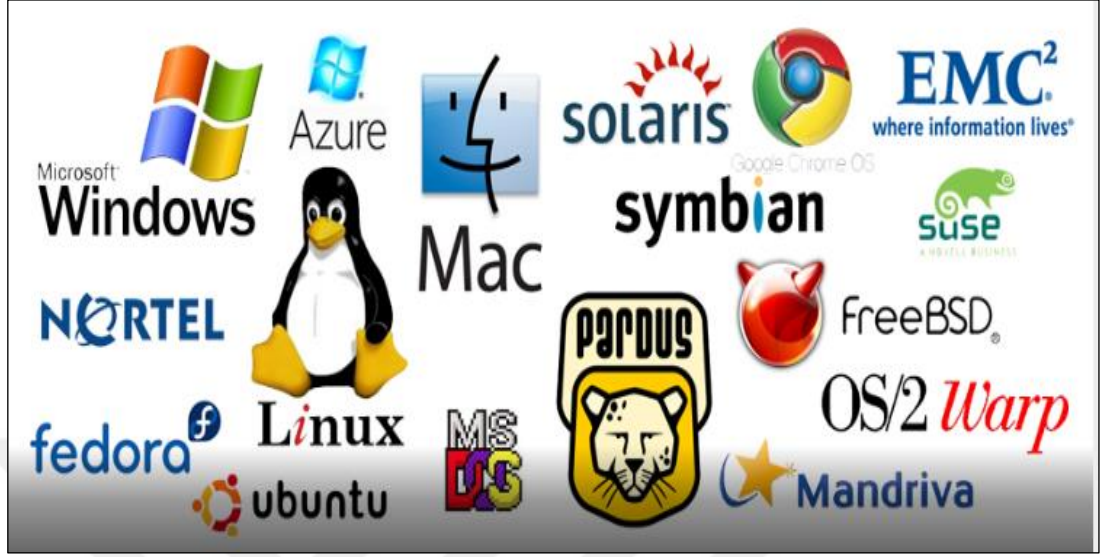
Sıra	Pin Adı	Uç (Pin) Oryantasyon	Uç (Pin) kullanımı
1	MD0	Giriş (çok zayıf pull-up)	MD1 ile işbirliği karar 4 çalışma modülleri olamaz sol havada
2	MD1	Giriş (çok zayıf pull-up)	MD0 ile işbirliği karar 4 çalışma modülleri Havada sol olabilir
3	RXD	Giriş	TTL seri giriş, bağlı harici TXD çıkış pin. Açık drenaj veya pull-up giriş olarak konfigüre edilebilir, görmek parametre ayarları
4	TXD	Çıkış	TTL seri çıkış, bağlı harici RXD girişi pin. Açık drenaj veya pull-up çıkış olarak konfigüre edilebilir, görmek parametre ayarları
5	AUX	Çıkış	Çalışma durumunu belirtmek için kullanılır modülü, kullanıcı uyanmak harici MCU, güç-on başlatma sırasında süre öz-testi, bu çıkış düşük güç seviyesi, açık drenaj çıkışı için konfigüre edilebilir, veya push-pull çıkış, görmek parametre ayarları
6	VCC		Güç girişi, voltaj 2.0-4.5 v (Not: daha az 3.6 v çıkış güç azalacak, ama üzerinde etkisini resepsiyon performansı çok küçük.)
7	GND		Zemin hattı, güç kaynağına bağlı referans zemin

E32-TTL-100 modeli Raspberry cihazına seri olarak ayarlama yapılır. Gönderilecek veri Seri porta bir nevi kopyalanır, diğer taraftan da seri porttan gelen veri alınır. Yukarıdaki Tablolarda belirtildiği üzere kartın montajı ve mod seçimi yapılır.

4.3. İşletim Sistemi

İşletim sistemi belirli kurallar çerçevesinde kendine bağlı donanımlar ile programlar arasındaki iletişimi sağlar. İşletim sistemlerin başında Windows, Mac ve Linux gelmekte. Şekil 4.10'da bazı işletim sistemlerin simgeleri gösterilmiştir. Bunlardan sadece Linux işletim sistemi tamamen açık kodlu ücretsiz bir işletim sistemidir. Kişi ve kuruluşlar tarafından geliştirilen bu işletim sistemlerin bazıları Pisi Linux, Pardus, Linux Mint, Ubuntu, Debian, Mageia, Fedora, Arch Linux, PCLinuxOS, openSUSE, Puppy Linux, CentOS ve Manjaro dur[28]. İşletim sistemi seçiminden sonra kullanılacak grafik arayüz seçilmelidir. Bunlardan bazıları GNOME, Unity, KDE, Xfce, Cinnamon, Mate, LXDE ve Enlightenment dir. Yüksek performans için görsellikten taviz verilebilir. Raspberry ile çalışabilen değişik işletim sistemleri

mevcut. Bunlardan bazıları Pardus ARM, OpenELEC, Kodi, RetroPie, Kano OS ve RASpbian (Debian tabanlı). Tezde Ubutuntu tabanlı Raspbian kullanılmıştır.



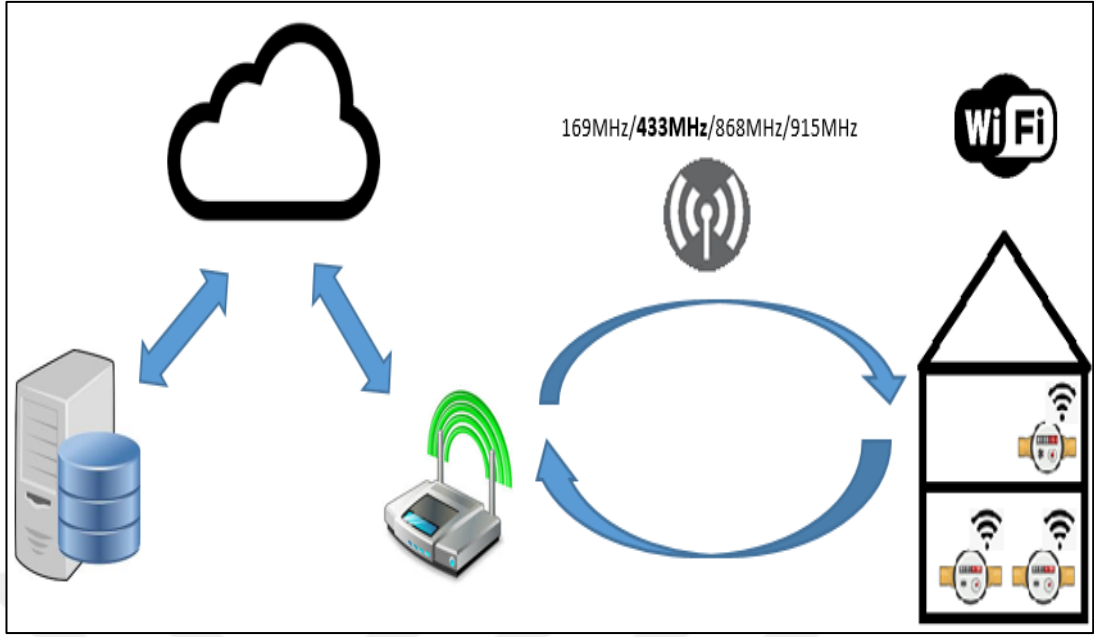
Şekil 4.10. İşletim Sistemlerinden Bazıları

4.4. Programlama Dili

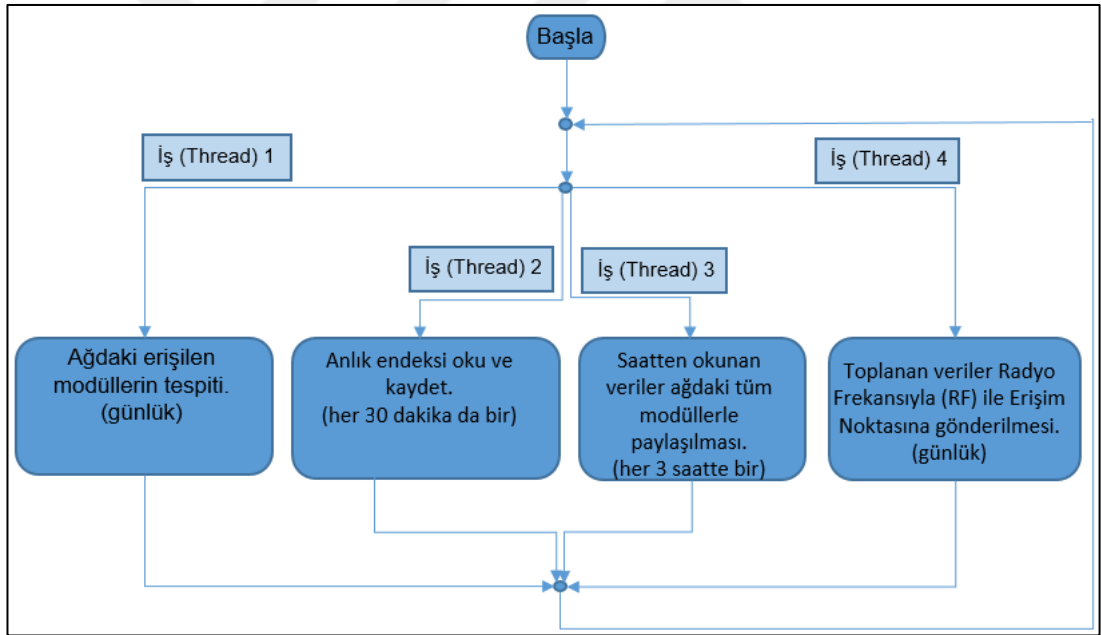
Programlama dili, şartlara ve düzene göre standart bir notasyona göre yapılması gereken işlemlerin bütününe denir[29]. Sistemi en aktif kullanmak için şemasal gösterilen akışa Algoritma denir. Şuana kadar 3000 den fazla programlama dili yapılmıştır. Kullanılan işletim sistemine ve yapılan işin cinsine göre yazılım seçilir. Bu tezde Raspberry için işletim sistemi olarak RASBIAN seçilmiştir. Bu işletim sisteminde desteklenen dillerden bazıları C, C#, C++, Java, Ada, Fortran, Python, Vala, Genie ve Rubby dir. Geniş kütüphaneye sahip, esnek ve kullanışlı olan Python kullanıldı.

5. GERÇEKLENEN SENARYOLAR

Şekil 5.1’de sistemin genel çalışma şeması gösterilmiştir. Ek-A’da ki 1 nolu getkeys.py kaynak kodu gerçek elektronik sayaç simülasyonu olarak klavyeden değer okuyup ilgili endeks dosyasına ekler. Ek-A deki 2 nolu checkterminals.py kaynak kodu ise kablosuz ağdaki aktif modüllerin IP adreslerini tutar. Günde bir defa yürütülmesi önerilmektedir. Şekil 5.2’de gösterildiği gibi her bir modül Ek-A da belirtilen 3 nolu myThreads.py kaynak kod ile belirli zaman aralıklarla saatteki veriyi (ileri akış, geri akış, pil seviyesi) bir dosyaya kaydeder. Parametrik olan bu süre yarım saatte birdir. Bir binadaki aboneler kendi aralarında bir ağ oluşturur. Oluşan bu kablosuz ağ yerine alternatif olarak Radyo Frekansıyla (RF) bağlantıda kullanılabilir. Ancak tez konusu olan WIFI teknolojisi kullanılmıştır. Ağdaki modüller parametrik olan her altı saatte bir kendilerinde tuttıkları verileri komşularıyla paylaşır. Ek-A da belirtilen 4 nolu server_pi.py kaynak kodu ile kablosuz yerel ağ daki modüllerden toplanan verilerin yayınlanmasını sağlayan bir web uygulamasıdır. Gateway olarak kullanılan modül ise Ek-A’da ki 5 nolu rf_listen.py kaynak kodu ile dinleme yapar. Ek-A’da ki 6 nolu rf_send.py kaynak kodu ile komşularından topladığı günlük verileri yine parametrik olan günde bir defa erişim noktasına (Gateway). Sistemin sürdürülebilir, hızlı ve güvenli veri aktarımı için değişik senaryolar hazırlanmıştır. Web sayfasında yayınlanan çıktı, Tablo 5.1’de gösterilmiştir Kurulum esnasında modüllere buldukları kat numaraları da kaydedilir. Zemin seviyesinin altında bulunan katlardaki modüller, yüksek katlı binalarda ki gibi kendinden daha yukarıdaki katta bulunan modül üzerinden veri transferi gerçekleştirilir.



Şekil 5.1 Sistem Blok Diyagramı



Şekil 5.2 Sistem Akış Diyagramı

Tablo 5.1. Aktif Modüllerden Okunan Veriler

ter504:192.168.4.4

ter507:192.168.4.7

ter500:192.168.4.1

Time	Counter	Time	Counter	Time	Counter
2018/05/07 17:09:45	750	2018/05/07 17:10:06	23242	2018/05/07 17:10:02	11017
2018/05/07 17:09:49	750	2018/05/07 17:10:09	23242	2018/05/07 17:10:05	11017
2018/05/07 17:09:53	751	2018/05/07 17:10:12	23242	2018/05/07 17:10:08	11018
2018/05/07 17:09:57	751	2018/05/07 17:10:15	23242	2018/05/07 17:10:11	11020
2018/05/07 17:10:01	751	2018/05/07 17:10:18	23244	2018/05/07 17:10:14	11020
2018/05/07 17:10:05	753	2018/05/07 17:10:21	23244	2018/05/07 17:10:17	11020
2018/05/07 17:10:09	754	2018/05/07 17:10:24	23245	2018/05/07 17:10:20	11020
2018/05/07 17:10:13	752	2018/05/07 17:10:27	23245	2018/05/07 17:10:23	11020
2018/05/07 17:10:17	753	2018/05/07 17:10:30	23249	2018/05/07 17:10:26	11020
2018/05/07 17:10:21	755	2018/05/07 17:10:33	23250	2018/05/07 17:10:29	11021

Aşağıda 8 senaryo anlatılmıştır.

5.1. Zayıf Pil Durumu

Modüller Ek-A kaynak kodunda belirtildiği gibi her yarım saatte bir okunan pil ölçüm değeri kabul edilebilir sınırın altında ise merkezdeki sisteme pil zayıf alarmı gönderir. Merkezdeki sistem ise pil değiştirilmesi için otomatik bir iş emri üretir. Bu iş emri neticesinde ilgili kişi veya ekip öngörülen zaman diliminde modüle müdahale eder.

5.2. Dışarıdan Modüle Müdahale

Dışarıdan modüle yetkisiz kişi veya kişiler tarafından müdahale edildiğinde merkeze müdahale alarmı gönderilir. Bölüm 5.1' de olduğu gibi bu durumda da otomatik bir iş emri üretilerek bir kişi veya ekibin modülün kontrolü sağlanır.

5.3. Ağa Yeni Bir Modül Eklenmesi

Her modül belirli aralıklarla Ek-A da belirtilen kaynak kodunu çalıştırarak kablosuz yerel ağdaki tüm aktif modüllerin IP adreslerini bir dosyada tutar. Parametrik olarak değiştirilebilen bu süre genelde günde 1 defa çalıştırılır. Böylece ağdaki tüm aktif modüller müteakip gün içinde bir birlerine veri transferini sağlar.

5.4. Ağdan Bir Modülün Çıkarılması Veya Arızalanması

Kablosuz yerel ağdaki tüm modüller günde bir defa ağdaki aktif komşularını tarayarak komşu listelerini yenilerler. Komşu listesinde olan bir modülün çıkarılması veya arızalanmasının akabinde neden hemen liste güncellenmiyor diye akla bir soru gelebilir. Modüller arasına kısa süreliğine iletişimi kesecek bir engel veya sinyal bozucu bir cihaz gelmiş olabilir. Kısa süreli arızanın akabinde o modül tekrar aktif hale gelebilir. Her hâlükârda bu işlemin neticesinde sisteme bir alarm gönderilir.

5.5. Erişim Noktası İletişim Modül Seçimi

Ağdaki modüllerden bir tanesi komşu modüllerden topladığı verileri günde bir defa RF ile erişim noktasına gönderir. Ancak bu gönderi esnasında başarısız olduğunda parametrik olarak tanımlanan belirli bir sayı kadar tekrar göndermeye çalışır. Şayet bu sayıyı aşmasına rağmen gönderi başarısız olmuş ise ağdaki bir başka modül bu gönderim işi görevini devralır. Sıralama olarak yerel ağdaki IP numaralarına göre küçükten büyüğe doğru yapılmakta. Ayrıca gönderi başarılı olsa bile şayet birden fazla deneme yapmış ise tutulan bir puanlama ile ağdaki başka bir modül bu görevi üstlenmesi sağlanır. Bu sayede sistem otomatik olarak en kaliteli bağlantıyı zamana bağlı olarak bulmuş olur.

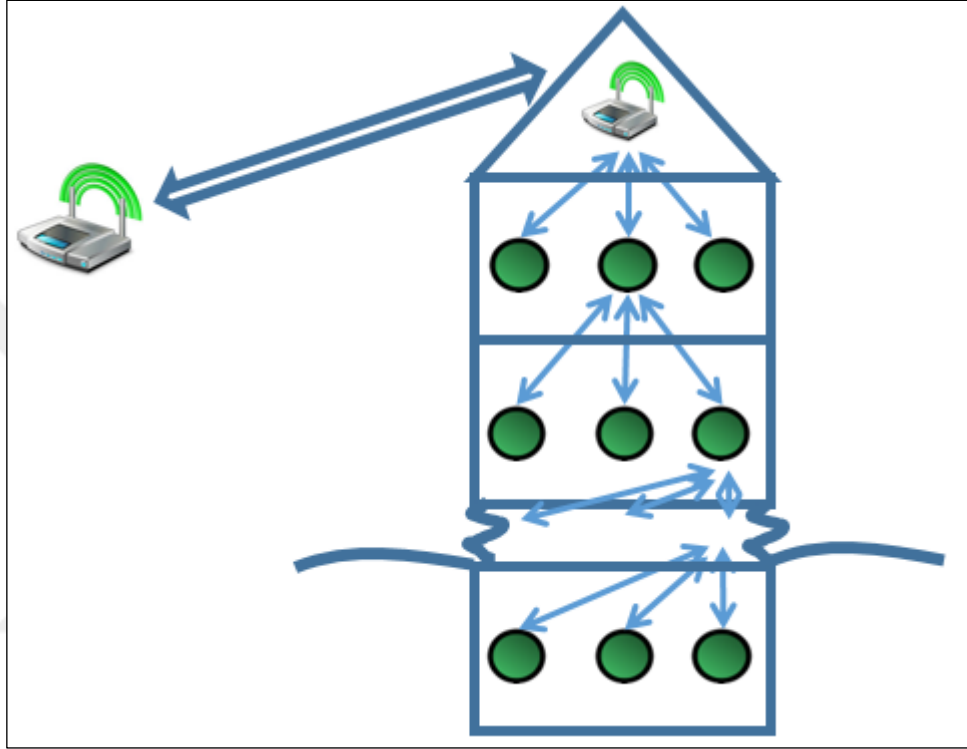
5.6. Erişim Noktası Seçimi

Bazı Modüller birden fazla Erişim noktasıyla iletişim kurabilmektedir. Bu durumda bu iki erişim noktasından hangisinin sinyal gücü yüksek ise onunla veri alış verişi yapar. Bu kontrol haftada bir yapılır. Parametrik olan bu haftalık değer değiştirilebilir. Sinyal gücünün yanında bu erişim noktasından veri alan modül sayıları da göz önünde bulundurulmaktadır. Aksi takdirde bazı erişim noktalarına oldukça az modül bağlantı kurarken bazılarında oldukça fazla modül bağlantı kurar. Buda sistemin genel anlamda sağlıklı çalışmasını engeller.

5.7. Çok Katlı Binalardaki Veri Aktarımı

Bu çalışmada karmaşık topoloji (mesh topology) kullanıldı. Ancak Şekil 5.3'de gösterildiği gibi 5 ve daha fazla katlı binalarda seri aktarım mümkün. Şöyle ki, erişim noktası için bu yüksek katlı binaların en üstüne birer erişim noktası eklenir. En alttaki abone modülleri bir üstteki abone modülüyle veri alış verişini yapar. Üstteki

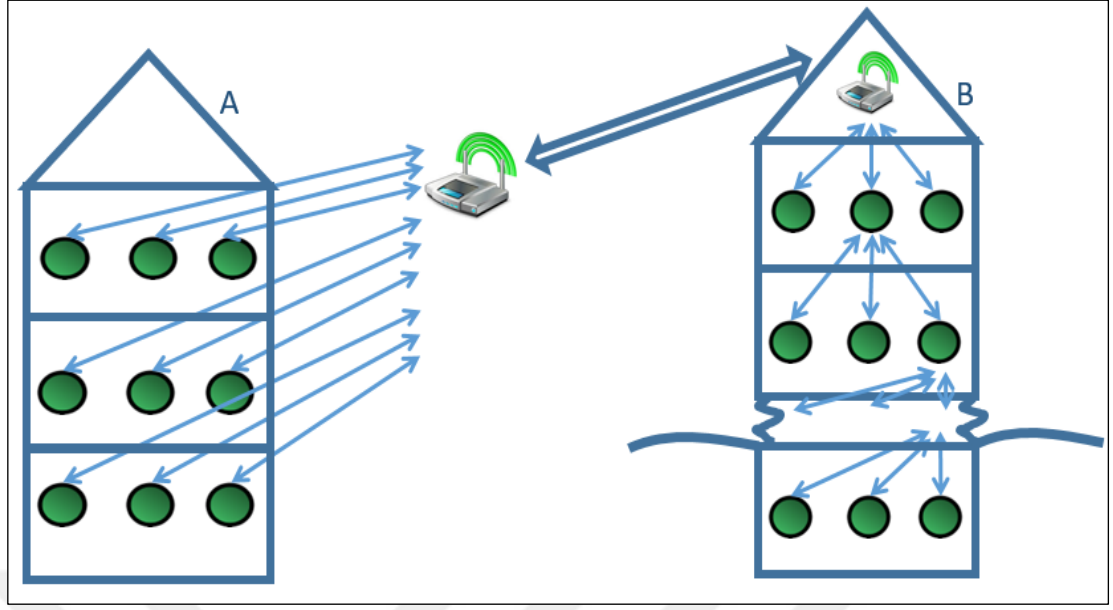
abone modülü de bir üstteki modülle veri alış verişinde bulunur. Bu şekilde her kattaki modül bir üstteki modülle veri alış verişi yaparak ana dağıtıcıya ulaşır. Ana dağıtıcıda uzaktaki bir erişim noktasıyla veri alış verişini yapar. Bu sayede zemin altında bulunan katlardaki abone modüllerin veri transfer problemleri bu sayede çözülmüş olur.



Şekil 5.3 Çok Katlı Binalarda Veri Aktarımı

5.8. Sitelerde Veri Aktarımı

Bölgedeki erişim noktasıyla erişim kurabilecek sitenin uygun yerine erişim noktası konur. Bu erişim noktasıyla o sitede bulunan tüm aboneler veri alış verişinde bulunur. Bu erişim noktası ise uzakta bulunan ana erişim noktası ile iletişim kurarak veri alış verişinde bulunur. Bu durumda binalarda birer erişim noktası aracılığıyla sitedeki ana erişim noktasına bağlanır. Diğer yöntem ise abone modüllerin direkt site ana erişim noktasına Şekil 5.4’de gösterildiği gibi erişim sağlamasıdır.

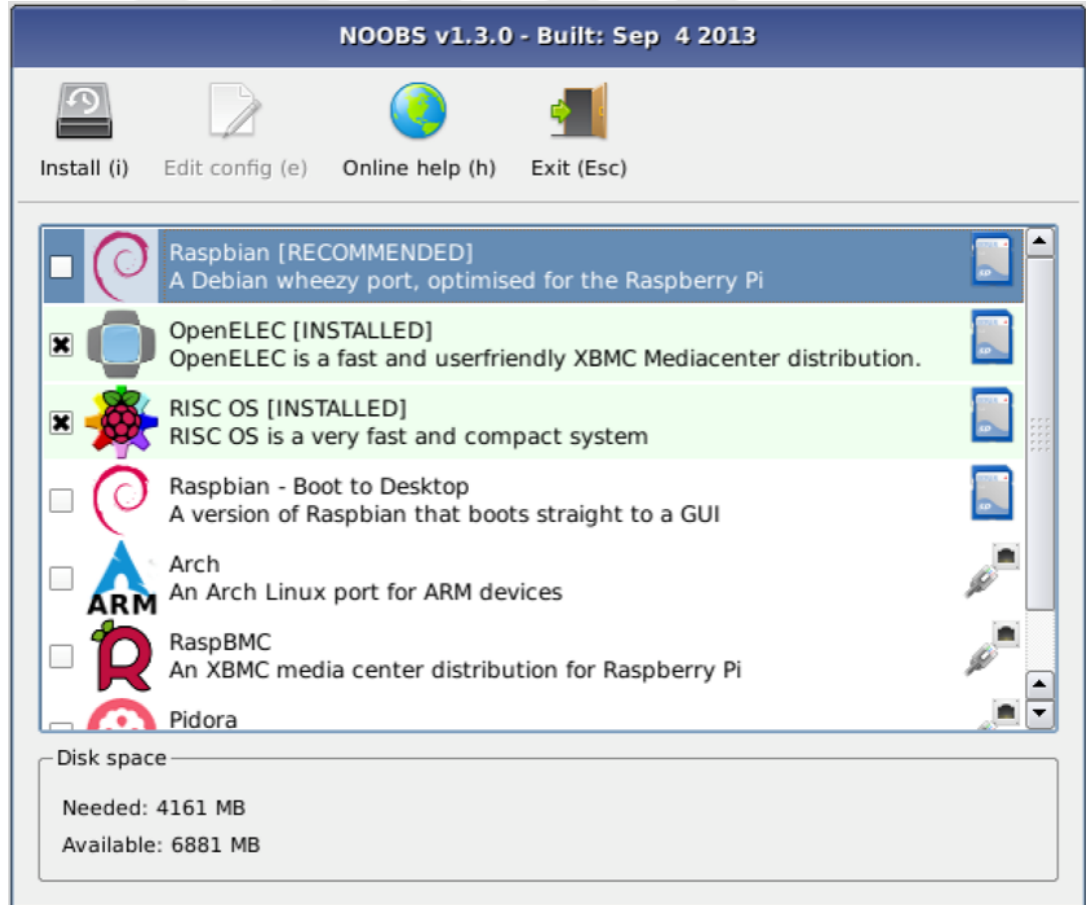


Şekil 5.4 Sitelerde Veri Aktarım Yönetimi , a) Az katlı bina, b) Çok katlı bina

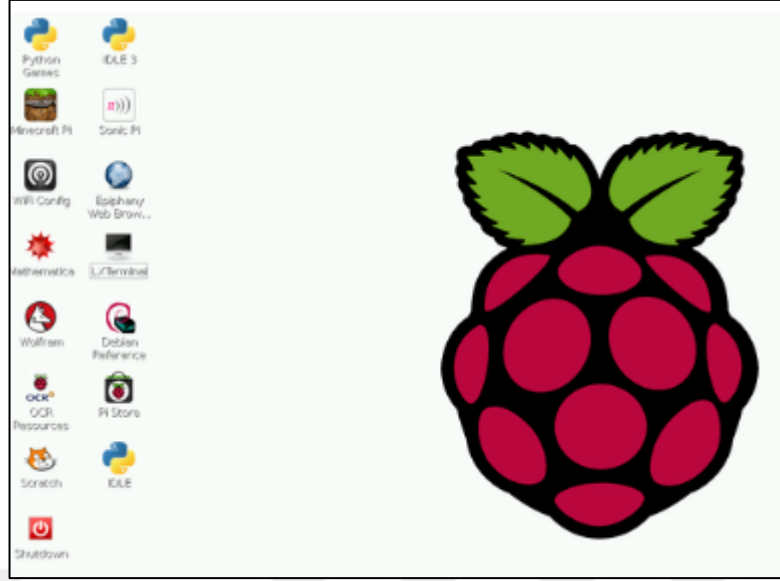
6. SİSTEMİN KURULUMU

6.1. İşletim Sistemin Kurulumu

4 adet Raspberry Pi 3, 6 adet RF modül ve 4 adet 16GB Class-10 tipi MicroSD Hafıza kartı temin edildi. Micro SD kartları [33] dosyası ile formatlanır. 1.6GB boyutlu NOOBS dosyasını zip halinde bilgisayara indirilir. Bu zip dosyası içindeki tüm dosyalar bir dosya yönetici yardımıyla MicroSD kart içine direkt kopyalanır. Kopyalama bitince MicroSD kartı Raspberry Pi'ye takılır. Monitor, Klavye, Fare takıldıktan sonra elektriğe bağlanarak sistem başlatılır. [30-32] deki yönergeler takip edilir. Şekil 6.1'de başta gözükten Raspbian [RECOMMENDED] seçilerek kurulum başlatılır.



Şekil 6.1. Raspberry Kurulum Ekranı.



Şekil 6.2. Rasbian İşletim Sistemin Kurulmuş Görünümü

Kurulum bittiğinde Şekil 6.2'deki gibi masaüstü gelir. İnternet erişimi varken sistem güncellemeleri ve yükseltmeleri için terminal moda geçilir. Aşağıdaki komutlar işletilir.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

6.2. Yerel Ağ (Hot Spot) Oluşturulması

[34-35] ' deki yönergelerle göre yerel ağ oluşturulur. Modüllerin yerel ağda birbirlerine bilgi göndermeyi aşağıdaki kod ile aktif hale getirilir. Sudo raspi-config ile ayar ekranı açılır. Buradan Interfacing Options (ara opsiyonlar) tıklanır. SSH seçeneği aktif hale getirilerek kaydedilir. Modüllerin birbirlerine güvenli bir şekilde veri göndermek için aşağıdaki komutlar işletilir [36].

```
rm ~/.ssh/id* (mevcut güvenlik anahtarını kaldırmak için)
```

```
ssh-keygen (yeni bir anahtar oluşturmak için)
```

```
ssh-copy-id <kullanıcı adı>@<ip-adres> ( oluşturulan güvenlik anahtarı karşı tarafa göndermek için)
```

```
ssh <kullanıcı adı>@<ip-adres> ( sorunsuz olduğu test edilir)
```

Yazılım yaptığımız python dilinin kurulumu için aşağıdaki kod işletilir.

sudo apt-get install python

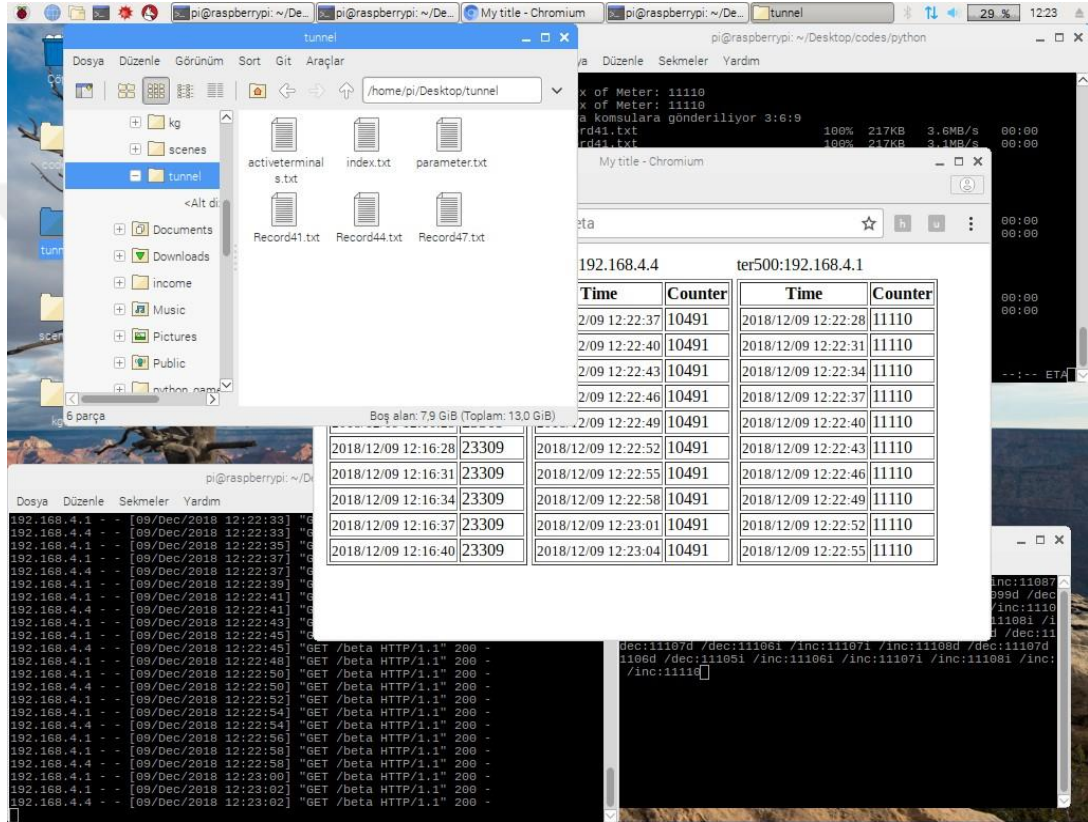
Ağa bağlı olan bilgisayarları bulmak için aşağıdaki kod işletilir.

sudo apt-get install arp-scan

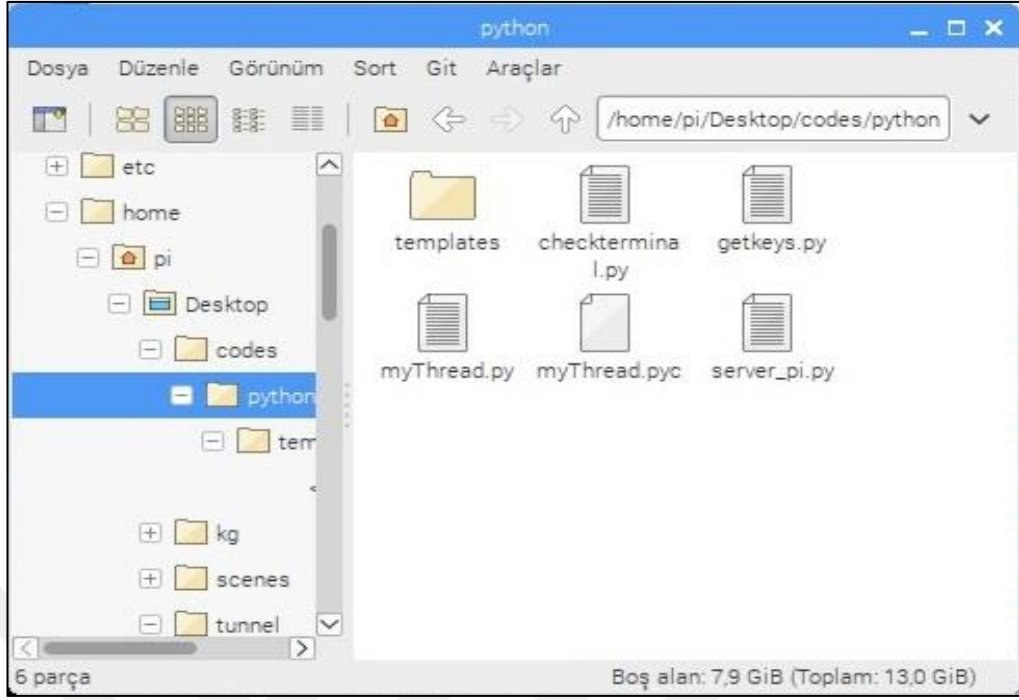
sudo arp-scan -l -- interface=wlan0 (kablosuz ağdakiler aranır)

sudo arp-scan -l -- interface=eth0 (yerel kablolu ağdakiler aranır)

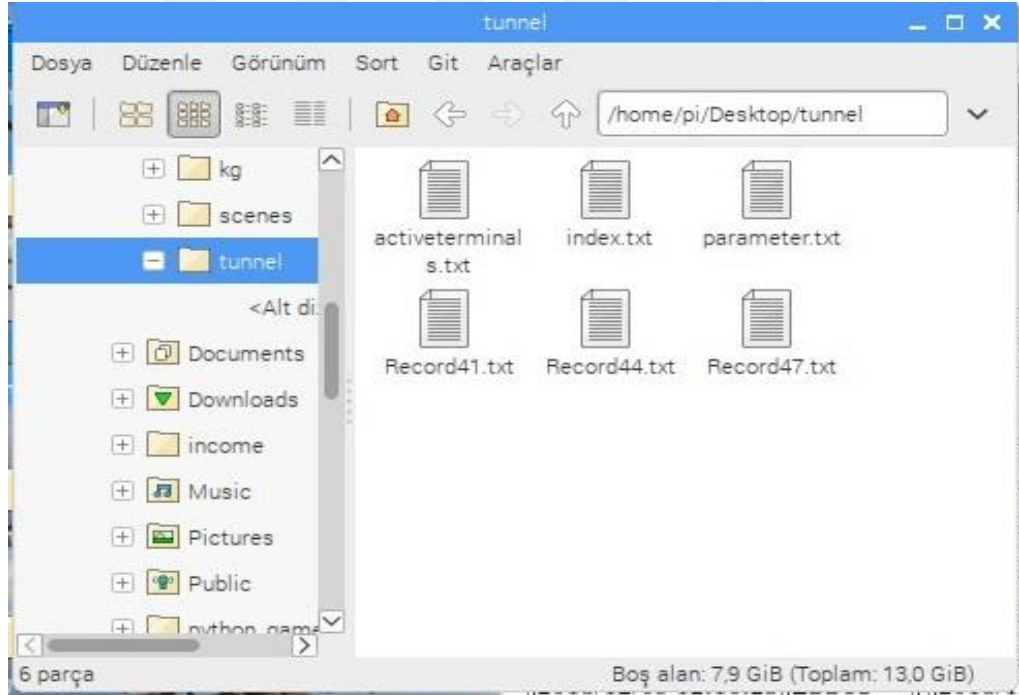
6.3. Modüllerin Kablosuz Ağda Veri



Şekil 6.3. Bir Modülün Masaüstü Görünümü.



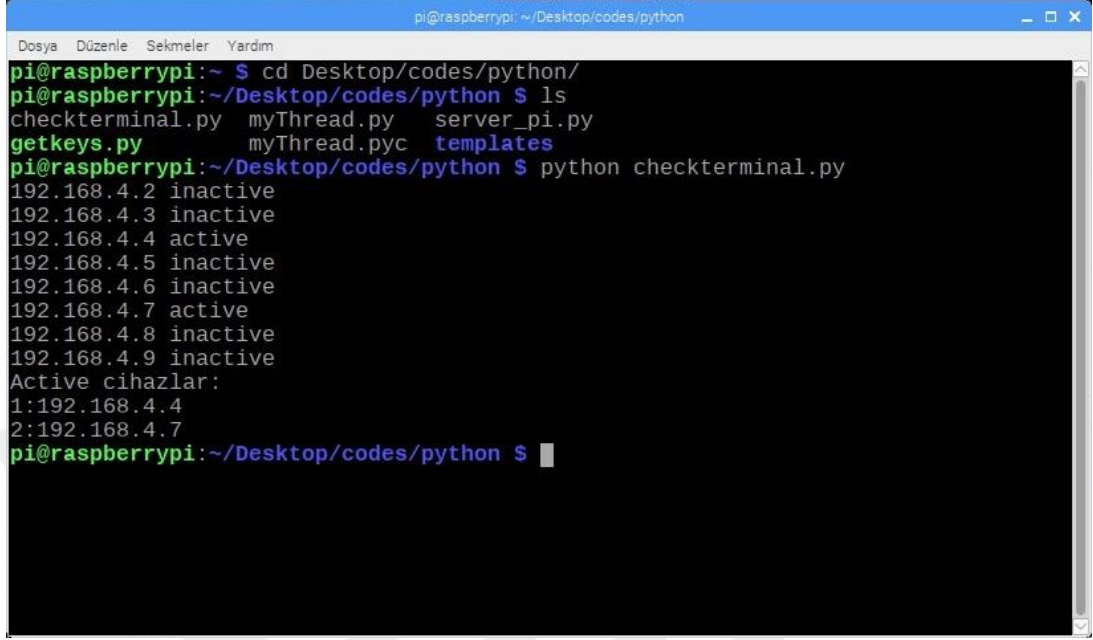
Şekil 6.4. Çalışan Uygulamalar



Şekil 6.5. Oluşan Verilerin Kaydedildiği Yer

Şekil 6.3’de uygulamaların çalıştırılmış hali gözükmektedir. Çalışan kodlar Şekil 6.4’de listelenmiştir. Çalışan kodlar tarafından oluşan çıktılar ise Şekil 6.5’de gözükmektedir. Önce Ek-A’deki checkTerminals.py kodu çalıştırılarak mevcut

ağdaki aktif modüllerin Şekil 6.6’de gösterildiği gibi IP adresleri bulunur. Bu işlem günde bir defa yapılır.

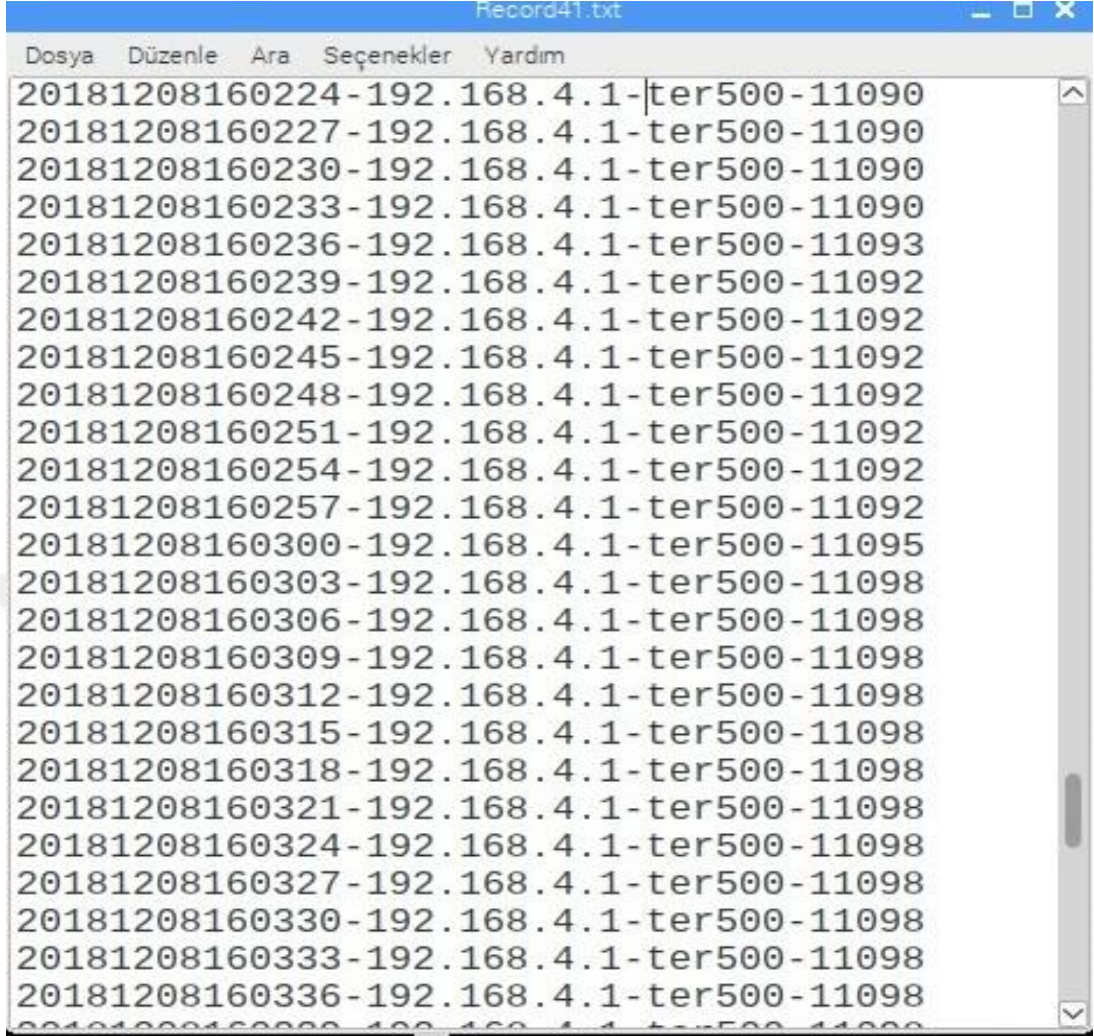


```
pi@raspberrypi: ~/Desktop/codes/python
Dosya Düzenle Sekmeler Yardım
pi@raspberrypi:~ $ cd Desktop/codes/python/
pi@raspberrypi:~/Desktop/codes/python $ ls
checkterminal.py  myThread.py  server_pi.py
getkeys.py       myThread.pyc  templates
pi@raspberrypi:~/Desktop/codes/python $ python checkterminal.py
192.168.4.2 inactive
192.168.4.3 inactive
192.168.4.4 active
192.168.4.5 inactive
192.168.4.6 inactive
192.168.4.7 active
192.168.4.8 inactive
192.168.4.9 inactive
Active cihazlar:
1:192.168.4.4
2:192.168.4.7
pi@raspberrypi:~/Desktop/codes/python $
```

Şekil 6.6. Ağdaki Aktif Modüllerin IP Adreslerin Tesbiti

Su saat endeksini okuyan Ek-A’daki kaynak kodu olan getkeys.py simülasyon uygulaması çalıştırılır. Bu uygulama saatin ileri akış, geri akış verisini okur.

Dataların kablosuz ağdaki tüm modüllerle paylaşımı için çoklu kesmenin bulunduğu Ek-A’daki kaynak kodu olan myThread.py uygulaması çalıştırılır. Bu uygulama parameter.txt parametre dosyasındaki sürelerle göre endeks dosyanın oluşturulmasını, aynı kablosuz ağdaki diğer modüllere datanın aktarılmasını sağlar. Şekil 6.7’de gösterildiği gibi periyodik olarak endeks değerlerini saklar.



```
Record41.txt
Dosya  Düzenle  Ara  Seçenekler  Yardım
20181208160224-192.168.4.1-ter500-11090
20181208160227-192.168.4.1-ter500-11090
20181208160230-192.168.4.1-ter500-11090
20181208160233-192.168.4.1-ter500-11090
20181208160236-192.168.4.1-ter500-11093
20181208160239-192.168.4.1-ter500-11092
20181208160242-192.168.4.1-ter500-11092
20181208160245-192.168.4.1-ter500-11092
20181208160248-192.168.4.1-ter500-11092
20181208160251-192.168.4.1-ter500-11092
20181208160254-192.168.4.1-ter500-11092
20181208160257-192.168.4.1-ter500-11092
20181208160300-192.168.4.1-ter500-11095
20181208160303-192.168.4.1-ter500-11098
20181208160306-192.168.4.1-ter500-11098
20181208160309-192.168.4.1-ter500-11098
20181208160312-192.168.4.1-ter500-11098
20181208160315-192.168.4.1-ter500-11098
20181208160318-192.168.4.1-ter500-11098
20181208160321-192.168.4.1-ter500-11098
20181208160324-192.168.4.1-ter500-11098
20181208160327-192.168.4.1-ter500-11098
20181208160330-192.168.4.1-ter500-11098
20181208160333-192.168.4.1-ter500-11098
20181208160336-192.168.4.1-ter500-11098
```

Şekil 6.7. Kendisine Bağlı Sayacın Endeks Datasını Saklar.

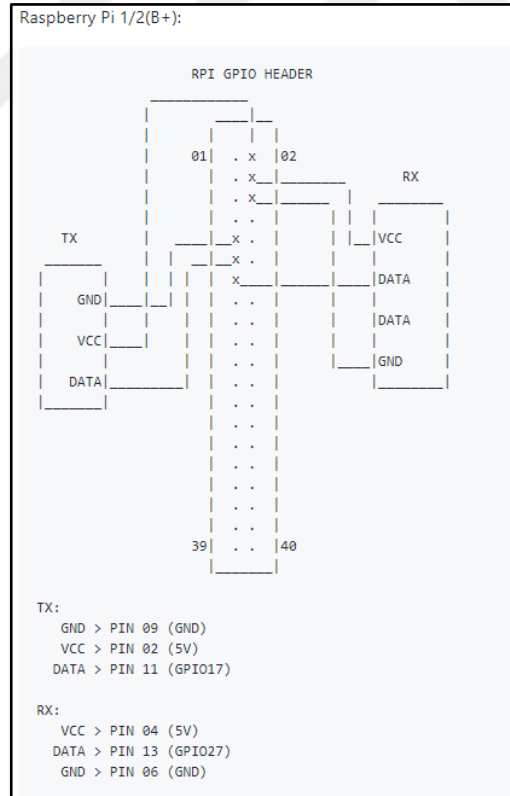
Uygulama sunucuyu aktif hale getiren Ek-A'de kaynak kodu olan server_pi.py çalıştırılır. Bu aşda olan tüm modüllerin tarayıcısından aşağıdaki linke tıklanır. Bu sayede şekil 6.8'deki gibi güncel veriler gösterilir.

Url: <http://x.x.x.x:2000/beta>.

ter504:192.168.4.4		ter507:192.168.4.7		ter500:192.168.4.1	
Time	Counter	Time	Counter	Time	Counter
2018/12/09 11:32:02	10486	2018/12/09 11:31:14	23300	2018/12/09 11:31:51	11098
2018/12/09 11:32:05	10486	2018/12/09 11:31:17	23300	2018/12/09 11:31:54	11098
2018/12/09 11:32:08	10486	2018/12/09 11:31:20	23301	2018/12/09 11:31:57	11098
2018/12/09 11:32:11	10486	2018/12/09 11:31:23	23308	2018/12/09 11:32:00	11098
2018/12/09 11:32:14	10486	2018/12/09 11:31:26	23309	2018/12/09 11:32:03	11109
2018/12/09 11:32:17	10486	2018/12/09 11:31:29	23306	2018/12/09 11:32:06	11108
2018/12/09 11:32:20	10493	2018/12/09 11:31:32	23306	2018/12/09 11:32:09	11108
2018/12/09 11:32:23	10492	2018/12/09 11:31:35	23306	2018/12/09 11:32:12	11108
2018/12/09 11:32:26	10491	2018/12/09 11:31:38	23306	2018/12/09 11:32:15	11108
2018/12/09 11:32:29	10491	2018/12/09 11:31:41	23309	2018/12/09 11:32:18	11110

Şekil 6.8. Uygulama Sunucu Uygulaması.

6.4. Radyo Frekans Bağlantısı



Şekil 6.9. Alıcı ve Gönderici Radyo Frekans Bağlama Şeması

Alıcı ve gönderici Şekil 6.9'deki gibi Raspberry'yle bağlantıları yapılır. Yukarıdaki şekle göre radyo frekanslı alıcı ve verici Raspberry cihazına bağlanır.

Aşağıdaki gibi güncelleme yapılmalı

```
sudo apt-get install python3-pip
```

```
pip3 install rpi-rf
```

```
git clone https://github.com/milaq/rpi-rf.git (github dan ilgili dosyalar indirilir)
```

İndirilen dosyalardan alıcı ve gönderici kodlar aşağıdaki şekilde çalıştırılır.

```
python3 rpi-rf_receive
```

```
python3 rpi-rf_send
```

Yukarıdaki alıcı (dinleyici) ve gönderici kodlarda gerekli ayarlamalar yapıldıktan sonra aynı Raspberry cihazında sorunsuz çalıştı. Fakat farklı Raspberry cihazlarında sağlıklı çalışmadı. Aşağıdaki yönerge ile farklı cihazlar arasında veri gönderimi başarıyla sonuçlandı [37].

Göndericinin uç 6(GND), uç 1 (3.3V VCC) ve uç 11 (DATA) ya bağlanır.

Alıcıda ise uç 9 (GND), uç 2 (5V VCC) ve uç 13 (DATA) bağlanır.

Dinleyici tarafında aşağıdaki kod işletilir

```
rpi-rf_receive -g 27
```

Gönderici tarafında ise aşağıdaki kod işletilir.

```
rpi-rf_send -g 17
```

Yukarıdaki kodlarla rakamlar sorunsuz gönderilebiliyor.

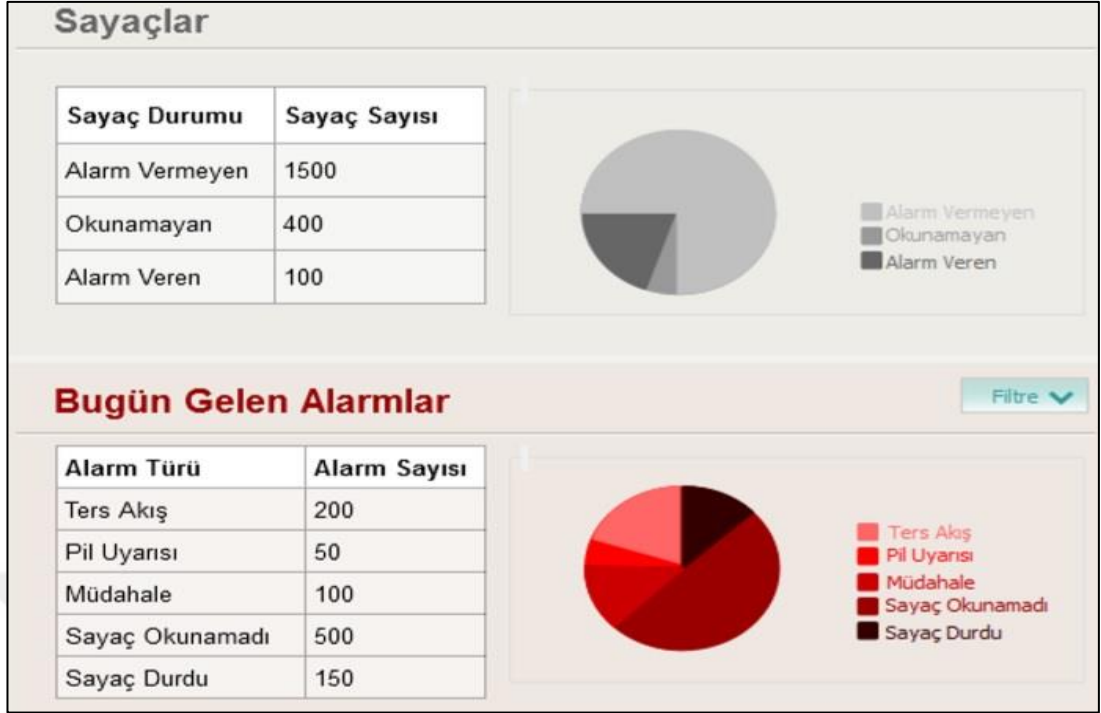
Alfa nümerik gönderim için aşağıdaki bağlantı şekli yapıldı [38].

Göndericinin uç 14(GND), uç 4 (3.3V VCC) ve uç 11 (DATA) ya bağlanır.

Alıcıda ise uç 6 (GND), uç 2 (5V VCC) ve uç 38 (DATA) bağlanır.

6.5. Kullanıcı Ara Yüzü

Modüllerden gelen alarmlar Şekil 6.10'daki gibi bir ara yüz de gösterilir. Aynı zamanda belirtilen yetkililere eposta ve/ veya sms ile bilgi gönderilir. Oluşturulacak senaryolara göre iş emirleri oluşturulup modül'ün bulunduğu yere kişi veya ekip gönderilir. Gerekli işlemlerin ardından ilgili alarm kapatılır.



Şekil 6.10. Alarm Yönetim Ekran Görüntüsü

7. SONUÇLAR VE ÖNERİLER

Lisanssız düşük enerjiyle uzak mesafelere veri aktarımından dolayı LORA teknolojisi tercih edilmiştir.

Tez çalışmasında İzmit ilçesinde bulunan 150.000 abonenin radyo frekans tabanlı su sayaçlarına takılacak modüller aracılığıyla, sayaçtan alınacak verilerin ağ geçidi (gateway) üzerinden merkeze ulaştırılması sağlanacaktır. Bu sistemin çalışabilir duruma getirilmesi planlanmaktadır.

İhtiyaç duyulan verilerin çeşitlilikleri zaman geçtikçe çoğalmaktadır. Bu da artık büyük boyutlu verilerin (Big Data) toplanması anlamına gelmektedir. Halka açık alanlarda bu verilerin aktarılmasıyla ilgili nesnelerin interneti (Internet Of Things, IOT) üzerine çalışmalar yapılmaktadır. Bu sistemlerden faydalanacak şekilde geliştirilmesi gerekmektedir [39 - 41].

IDC Nesnelerin İnterneti, Analitik, Büyük Veri & Yapay Zeka konulu 2018 yılı sonunda yapılan konferansta belirtildiği üzere günde 2 Milyondan fazla algılayıcıdan veri okuyan KoçDigital firması şüana kadar IoT için 1 Milyar \$ yatırım yapmıştır. Dünya genelinde 2021 yılına kadar IoT için 2 Trilyon \$ yatırım yapılacağı öngörülmektedir [42].

Enerji kaynağı için önerilerden bir tanesi, gündüzleri gün ışığından aldığı enerjiyi depolayarak gece kullanabilmesi. Diğeri ise tesisata, abone su sayaçlardan önce enerji üreten su tribünleri ilave edilmesi. Bu durumda aboneye gelen suyun basıncında bir düşüş meydana gelecektir. Bunu önlemek adına su akış basıncını artırmak için su dağıtım yeri olan depoların konumlarını gerekli seviyede yükseltmek gerekecektir. Üçüncü yöntem ise yukarıda bahsedilen güneş enerji ve su tribününü birlikte kullanabilmektir. Yani bataryanın gücü belli bir değerin altına düştüğünde direkt tribünden faydalanabilmesi üzerine çalışma öngörülmektedir.

KAYNAKLAR

- [1] Daldal, N., İçme Suyu Şebeke Otomasyonunun Tasarımı Ve Gerçekleştirilmesi, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 2018, **24**(5), 831–836.
- [2] Benzi F., Anglani N., Bassi E., Frosini L., Electricity Smart Meters Interfacing The Households, *IEEE Transactions On Industrial Electronics*, DOI: 10.1109/TIE.2011.2107713.
- [3] Li M., Lin H., Design And Implementation Of Smart Home Control Systems Based On Wireless Sensor Networks And Power Line Communications, *IEEE Transactions On Industrial Electronics*, DOI: 10.1109/TIE.2014.2379586.
- [4] Abu-Mahfouz A.M., Olwal T.O., Kurien A.M., Munda J.L., Djouani K., Toward Developing A Distributed Autonomous Energy Management System (DAEMS), DOI:10.1109/Afrcon.2015.7332008.
- [5] Nasirian A., Maghrebi M.F., Yazdani S., Leakage Detection İn Water Distribution Network Based On A New Heuristic Genetic Algorithm Model, *Journal Of Water Resource And Protection*, 2013, **5**, 294-303.
- [6] Kondratjevs K., Zabasta A., Kunicina N., Ribickis L., Development Of Pseudo Autonomous Wireless Sensor Monitoring System For Water Distribution Network, *2014 IEEE 23rd International Symposium On Industrial Electronics (ISIE)*, İstanbul, 1-4 Haziran 2014.
- [7] Mudumbe M.J., Abu-Mahfouz A.M., Smart Water Meter System For User-Centric Consumption Measurement, *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, Cambridge, 22-24 July 2015.
- [8] Hu G., Zhang X., Qian W., Yan P., Hu Z., A Routing Protocol For Low Energy Wireless Sense Network, *In 2006 IET International Conference On Wireless, Mobile And Multimedia Networks*, China, 6-9 November 2006.
- [9] Yüksel E., Uzaktan Ölçüm İçin Zigbee Mimarisi Kullanan Sistem Tasarımı, Yüksek Lisans Tezi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2010, 282571.
- [10] Süzer E.S., İkizoğlu S., Uzaktan Sayaç Okuma Teknikleri Ve Modbus-Rtu, Iec 61107 Mod C Protokolleri İle Örnek Yazılım, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006, 222155.
- [11] Muhendra R., Husein, Budiman M. Ve Khairurrijal, Development Of Digital Water Meter Infrastructure Using Wireless Sensor Networks, *AIP Conference Proceedings*, Yogyakarta, Indonesia, 25-26 January 2016.

- [12] Ahmad Z., Asad E. U., Muhammad A., Ahmad W. Ve Anwar A., Development Of A Low-Power Smart Water Meter For Discharges In Indus Basin Irrigation Networks, *Wireless Sensor Networks For Developing Countries*, DOI: 10.1007/978-3-642-41054-3_1.
- [13] Spinsante S., Pizzichini M., Mencarelli M., Squartini S., And Gambi E., Evaluation Of The Wireless M-Bus Standard For Future Smart Water Grids. *2013 9th International Wireless Communications And Mobile Computing Conference (IWCMC)*, Sardinia, Italy, 1-5 July 2013.
- [14] Domoney W.F., Ramli N., Alarefi S., And Walker, Smart City Solutions To Water Management Using Self-Powered, Low-Cost, Water Sensors And Apache Spark Data Aggregation, *3rd International Renewable And Sustainable Energy Conference (IRSEC)*, Marrakech, Morocco, 10-13 December 2015.
- [15] Phuong N.M., Schappacher M., Sikora I.A., Ahmah Z., And Muhammad D.A., Real-Time Water Level Monitoring Using Low-Power Wireless Sensor Network, *EmbeddedWorld 2015 Conference*, Nürnberg, Germany, 24-26 February 2015.
- [16] Çakmak B. Ve Gökalp Z., Kuraklık Ve Tarımsal Su Yönetimi, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 2013, **1**(4), 1-11.
- [17] <http://www.skdturkiye.org/yayin/turkiyede-suyun-durumu-ve-su-yonetiminde-yeni-yaklasimlar-cevresel-perspektif/>, (Ziyaret tarihi: 13 Aralık 2018)
- [18] Kulkarni P., And Farnham T., Smart City Wireless Connectivity Considerations And Cost Analysis: Lessons Learnt From Smart Water Case Studies. *IEEE Access*, 2016, **4**, 660–672.
- [19] Çölkesen R., And Örencik B., Bilgisayar Haberleşmesi Ve Ağ Teknolojileri, *Papatya Yayıncılık*, İstanbul, 2008.
- [20] Bykowski M., Tracey D., Graham B., Timmons N. Ve Morrison J., A Schema For The Selection Of Network Topology For Wireless Body Area Networks, İçinde *2011 IEEE Radio And Wireless Symposium*, Phoenix, AZ, USA, 16-19 January 2011.
- [21] Ganesh A., Massoulie L., And Towsley D., The Effect Of Network Topology On The Spread Of Epidemics, *Proceedings IEEE 24th Annual Joint Conference Of The IEEE Computer And Communications Societies.*, Miami, FL, USA, 13-17 March 2005.
- [22] <https://fossbytes.com/tree-topology-advantage-disadvantage/>, (Ziyaret tarihi: 6-Kasım-2018)
- [23] Marais J., Malekian R., Ye N., And Wang R., A Review Of The Topologies Used In Smart Water Meter Networks, *Journal of Sensors, A Wireless Sensor Network Application*, **2016**(1),1-12.
- [24] Keskin Ü., Tarik Ö., Ioterm Şirketi'nin Nesnelerin İnterneti Konusuna Ve

Uygulamalarına Bakışı, *Mühendis Ve Makine Güncel Dergisi Özel Sayı Çağruları*, 2018, **59**(698), 8-23

- [25] Mermer G. B. Ve Zeydan E., A Comparison Of LP-WAN Technologies: An Overview From A Mobile Operators' Perspective, *25th Signal Processing And Communications Applications Conference (SIU)*, Antalya, 15-18 May 2017.
- [26] <https://serifgungor.com/blogs/10-raspberry-pi-hakkinda/>, (Ziyaret tarihi: 19 Kasım 2018)
- [27] <https://www.raspi-tr.com/2012/08/01/raspberry-pi-nedir/>, (Ziyaret tarihi: 19 Kasım 2018)
- [28] <https://www.linuxdunyam.com/2017/10/linux-rehberi.html/>, (Ziyaret tarihi: 25 Kasım 2018)
- [29] http://Www.Robotiksistem.Com/Programlama_Dilleri_Ozellikleri.Html/, (Ziyaret tarihi: 25 Kasım 2018)
- [30] <https://cahitsoyman.blogspot.com/2015/12/raspberry-pi-kurulumu-ve-kullanimi.html/>, (Ziyaret tarihi: 6 Aralık 2018)
- [31] <https://www.raspi-tr.com/2012/08/03/debian-raspbian-kurulumu/>, (Ziyaret tarihi: 6 Aralık 2018)
- [32] <https://ch-film.net/v-raspberry-pi-3-isletim-sistemi-kurulum-raspbian-kurulumu-qwhyfbsr58k.html/>, (Ziyaret tarihi: 6 Aralık 2018)
- [33] https://save-download.com/d/i4zhf983ee/sd_cardformatter0500setupen_exe/, (Ziyaret tarihi: 6 Aralık 2018)
- [34] <https://howtoraspberrypi.com/create-a-wi-fi-hotspot-in-less-than-10-minutes-with-pi-raspberry/>, (Ziyaret tarihi: 7 Aralık 2018)
- [35] <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md/>, (Ziyaret tarihi: 7 Aralık 2018)
- [36] <https://www.raspberrypi.org/documentation/remote-access/ssh/passwordless.md/>, (Ziyaret tarihi: 7 Aralık 2018)
- [37] http://www.knight-of-pi.org/home-automation-with-rpi-rf_send-and-rpi-rf_receive-for-433mhz-rc-switches/, (Ziyaret tarihi: 7 Aralık 2018)
- [38] <https://quadmeup.com/virtualwire-support-for-raspberry-pi/#comment-7013/>, (Ziyaret tarihi: 7 Aralık 2018)
- [39] Koo D., Piratla K., And Matthews C.J., Towards Sustainable Water Supply: Schematic Development Of Big Data Collection Using Internet Of Things (Iot). *Procedia Engineering*, 2015, 118, 489–497.
- [40] Muhendra R., Rinaldi A., Budiman M., And Khairurrijal, Development Of Wifi

Mesh Infrastructure For Internet Of Things Applications. *Procedia Engineering*, 2017, **170**(C), 332–337.

- [41] Cherukutota N., And Jadhav S., Architectural Framework Of Smart Water Meter Reading System In Iot Environment, *International Conference On Communication And Signal Processing (ICCSP)*, Melmaruvathur, India, 6-8 April 2016.
- [42] https://www.idc.com/mea/events/65260-idc-iot-analytics-big-data-cognitive-ai-computing-conference-2018?g_clang=tr/, (Ziyaret tarihi: 11 Aralık 2018)





EKLER

Ek-A : Kullanılan Yazılım Kaynak Kodları

1) getkeys.py : Gerçek elektronik sayaç simülasyon uygulama

```
# -*- coding: utf-8 -*-
import curses
import sys
import os

screen = curses.initscr()
curses.cbreak()
screen.keypad(1)

screen.addstr("q:quit i:increment d:decrement ")
screen.refresh()

key = "
s=5

q = ord('q')
i = ord('i')
d = ord('d')

os.chdir("/home/pi/Desktop/tunnel/")
dosya = "index.txt"

#f= open(dosya,"w+")
#f.write("500")
#f.close()

while key != q:
    key = screen.getch()
    if (key == d):
        f= open(dosya,"r")
        f1=f.read()
        s=int(f1)
        f.close()
        s=s-1
        f= open(dosya,"w+")
        f.write(str(s))
        f.close()
```

```
        sys.stdout.write(" /dec:"+str(s))
        sys.stdout.flush();
elif key == i:
    f= open(dosya,"r")
    f1=f.read()
    s=int(f1)
    f.close()
    s=s+1
    f= open(dosya,"w+")
    f.write(str(s))
    f.close()
    sys.stdout.write(" /inc:"+str(s))
    sys.stdout.flush();
else:
    sys.stdout.write(" /no action:"+str(s))
    sys.stdout.flush();
    #screen.addch(key)
    #screen.refresh()
curses.endwin()
```

2) checkterminals.py : Yerel Ağdaki aktif modüllerin IP değerlerini tesbit eden uygulama

```
import subprocess
import os
#import time
import socket
#import datetime
import fcntl
import struct

def get_ip_address(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    return socket.inet_ntoa(fcntl.ioctl(
        s.fileno(),
        0x8915, # SIOCGIFADDR
        struct.pack('256s', ifname[:15])
    )[20:24])

p=[]
#locIp=get_ip_address("eth0")      # "eth0" "lo" "wlan0"
locIp=get_ip_address("wlan0")     # "eth0" "lo" "wlan0"
b=locIp.split(".")
n1=int(b[0])
n2=int(b[1])
n3=int(b[2])
n4=int(b[3])
c=str(n1)+ "." +str(n2)+ "." +str(n3)+ ".{0}"
with open(os.devnull, "wb") as limbo:
    for n in xrange(1,10):
        if (n==n4 ):
            pass
        else:
            ip=c.format(n)
            #ip="172.16.211.{0}".format(n)
```

```

        result=subprocess.Popen(["ping", "-c", "1", "-n", "-W", "2",
ip],
                                stdout=limbo, stderr=limbo).wait()
    if result:
        print ip, "inactive"
    else:
        print ip, "active"
        p.append(ip)

os.chdir("/home/pi/Desktop/tunnel/")
dosya = "activeterminals.txt"
c=1
t=""
for ip in p:
    print str(c)+":"+ip
    t+=ip+", "
    c+=1
t=t[:-1]
f=open(dosya,"w+")
f.write(str(t))
f.close()

```

3) myThreads.py : Periyodik aralıklarda yerel ağdaki aktif komşular arasında veri transferini sağlayan uygulama

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#!/usr/bin/python
import thread
import time
import socket
import datetime
import fcntl
import struct
import os
import glob
locIp=""
os.chdir("/home/pi/Desktop/tunnel/")
param = "parameter.txt"
index = "index.txt"
recName=""
ipfile="activeterminals.txt"
s=-1
x=-1 #save current consume,time period,terminal id on local
y=-1 #time interval for sending datas via wifi
z=-1 #time interval for sending datas via rf
terminalId="" # terminal id
ips=[]
htmltop="<!DOCTYPE html><html lang=\"en\"><head><title>My
title</title></head><meta http-equiv=\"refresh\" content=\"2\" /><body>"
htmlbottom="</body></html>"
htmlinner="<table><tr>"
datadirs="/home/pi/Desktop/tunnel/"
#recs=[]
def appendFile(par):
```

```

a=open(recName,"a+")
now=datetime.datetime.now().strftime("%Y%m%d%H%M%S")
yaz=now+"-"+locIp+"-"+terminalId+"-"+par+"\n"
a.write(yaz)
a.close()
print yaz
return;

def readIndex():
    a=open(index,"r")
    global s
    s=a.read()
    a.close()
    return;

def get_ip_address(ifname):
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    return socket.inet_ntoa(fcntl.ioctl(
        s.fileno(),
        0x8915, # SIOCGIFADDR
        struct.pack('256s', ifname[:15])
    )[20:24])

def readInitials():
    global x
    global y
    global z
    global recName
    global locIp
    global terminalId
    global ips
    # read active ips
    f=open(ipfile,"r")
    lines=f.read().split("\n")
    ips=lines[0].split(",")
    f.close()

```



```

print("Neighbour ip # :"+str(len(ips)))
# read time slices and terminal id
f=open(param,"r")
lines=f.read().split("\n")
datas=lines[0].split(",")
f.close()
if (len(datas)==4):
    x=datas[0]
    y=datas[1]
    z=datas[2]
    terminalId=datas[3]
print "initial values "+str(x)+':'+str(y)+':'+str(z)+':'+terminalId
#locIp=socket.gethostbyname(socket.gethostname()) # gives 127.0.1.1
#locIp=get_ip_address("eth0") # "eth0" "lo" "wlan0"
locIp=get_ip_address("wlan0") # "eth0" "lo" "wlan0"
b=locIp.split(".")
n3=int(b[2])
n4=int(b[3])
recName="Record"+str(n3)+str(n4)+".txt"
return;
def callx(x):
    count=0
    while (1==1):
        readIndex()
        appendFile(s)
        time.sleep(int(x))
        print "Index of Meter: %s" % (s)
        count+=1
def readRecords():
    global datadirs
    global htmlinner
    recs=glob.glob(datadirs+"Record*.txt")
    htmlinner=""

```

```

print len(recs)
heads=""
for a in recs:
    f=open(a,"r")
    #b=f.read()
    b = f.readlines()[-10:]
    f.close()
    strTable = " <table border='1\"><tr><th>Time</th><th>Counter</th></tr>"     metername=""     meterip=""     for i in b:         rec=i.split("-")         rdate=rec[0]         rd=rdate[0:4]+"/"+rdate[4:6]+"/"+rdate[6:8]+ +rdate[8:10]+":"+rdate[10:12]+":"+rdate[12:]         strRW = "<tr><td><font style='font-size:13px;'>"+rd+ "</font></td><td>"+rec[3]+</td></tr>"         strTable += strRW         meterip=rec[1]         metername=rec[2]         strTable += "</table></td>"         heads+="<td>"+metername+": "+meterip+"</td>"         htmlinner+=strTable     htmlinner="<table><tr>"+heads+"</tr><tr>"+htmlinner     htmlinner+="</tr></table>"  def update_html():     readRecords()     global htmltop     global htmlbottom     global htmlinner     global s |
```

```

    direct="/home/pi/Desktop/codes/python/templates"
    fpath = os.path.join(direct, "myfile.html")
    f = open(fpath,"w+")
    f.write(htmltop)
    f.write(htmlinner)
    f.write(htmlbottom)
    f.close()
    print "html is called:"
def cally(y):
    while (1==1):
        update_html()
        time.sleep(int(y))
        print "Dosya komsulara gönderiliyor %s:%s:%s" % (x,y,z)
        for a in ips:
            os.system("scp "+recName+"
pi@"+a+":/home/pi/Desktop/tunnel/")
            print("scp "+recName+" pi@"+a+":/home/pi/Desktop/tunnel/")

## Beginning of program
readInitials()
try:
    #thread.start_new_thread( print_time, ("Thread-1", 2, ) )
    #thread.start_new_thread( print_time, ("Thread-2", 4, ) )
    thread.start_new_thread( callx, (x, ) )    #x: parameter
    thread.start_new_thread( cally, (y, ) )    #y: parameter
except:
    print "Error: unable to start thread"
while 1:
    pass

```

4) server_pi.py : Yerel ağdaki aktif modüllerden toplanan verileri tarayıcıda yayınlayan uygulama

```

from flask import Flask, render_template

```

```

from flask import request
app = Flask(__name__)
c=1
a="j"
b="l"
@app.route('/')
def display():
    global b
    global c
    c+=1
    if(c%2==1):
        b="<br/>"
    else:
        b=""
    return "Looks like "+b+" it works!"
@app.route('/alpha')
def alpha():
    return "This is the alpha version"

@app.route('/beta')
def beta():
    return render_template('myfile.html')
@app.route('/search')
def index():
    query = request.args.get('query',"", type=str);
    return query

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True, port=2000)

```

5) rf_listen.py : RF ile gelen verileri dinyelen uygulama

```
#!/usr/bin/env python
```

```

# -*- coding: utf-8 -*-
import serial
from datetime import datetime
from time import sleep
#kg port = serial.Serial("/dev/ttyAMA0", baudrate=115200, timeout=3.0)
port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=3.0)
header="@P#*"
tail="@Q"
oku=""
tobesent=[]
def fill_examples(): # Test Verileri
    tobesent.append("Bugün Cuma Akşamı oldu")
    tobesent.append("Aylardan Ağustos ayı")
    tobesent.append("İsu Genel Müdürlüğü Bilgi İşlem Daire Başkanlığı")
    tobesent.append("Rf yöntemiyle gönderilen Verilerin testi!")
    tobesent.append("Gönderim işlerinde yapılması gerekenler belirtilmeli")
c=0
while True:
    c=c+1
    data=port.readline()
    while (len(data)<6):
        data=port.readline()
    sleep(0.1)
# t2=datetime.now().strftime("%d/%m/%Y %H:%M:%S.%f")[:-3]
    print("Son Durum:"+data+": "+str(c)+"\n")
    port.write("Okunan:"+str(c))

```

6) rf_send.py : RF ile verileri gönderen uygulama

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import serial
import os
import glob
import thread
import time
from datetime import datetime
from time import sleep
#kg port = serial.Serial("/dev/ttyAMA0", baudrate=115200, timeout=3.0)
port = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=3.0)
header="@P#*"
tail="@Q"
oku=""
datadirs="/home/pi/Desktop/tunnel/"
os.chdir("/home/pi/Desktop/tunnel/")
param = "parameter.txt"
z=-1 #time interval for sending datas via rf
terminalId="" # terminal id
tobesent=[]
def readInitials():
    global z
    f=open(param,"r")
    lines=f.read().split("\n")
    datas=lines[0].split(",")
    f.close()
    if (len(datas)==4):
        z=datas[2]
        terminalId=datas[3]
    print "initial values "+str(z)+':'+terminalId
    return;
```

```

def readAndSendRecords():
    global datadirs
    global htmlinner
    recs=glob.glob(datadirs+"Record*.txt")
    htmlinner=""
    print len(recs)
    heads=""
    for a in recs:
        f=open(a,"r")
        b = f.readlines()[-5:]
        f.close()

        for rec in b:
            rdate=rec[0]
            uz=str(len(rdate))
            uz=uz.rjust(3,'0')
            port.write(header+uz+rdate+tail+"\n")
            sleep(0.1)
            print('Giden:'+rdate)

def callz(z):
    while (1==1):
        print "Datarlar Gateway'e %s" % (z)
        readAndSendRecords()
        bekle=3
        #sleep(0.1)
        data=port.readline()
        while (len(data)<6) and (bekle>0):
            # sleep(0.01)
            bekle=bekle-1
            print("."+str(bekle))
            data=port.readline()
        t2=datetime.now().strftime("%d/%m/%Y %H:%M:%S.%f")[:-3]
        print("Teyid Durum:"+data)

```

```
def fill_examples(): # Örnek Veriler
    tobesent.append("Bugün Cuma Akşamı oldu")
    tobesent.append("Aylardan Ağustos ayı")
    tobesent.append("İsu Genel Müdürlüğü Bilgi İşlem Daire Başkanlığı")
    tobesent.append("Rf yöntemiyle gönderilen Verilerin testi!")
    tobesent.append("Gönderim işlerinde yapılması gerekenler belirtilmeli")

readInitials()
try:
    thread.start_new_thread( callz, (z, ) )    #z: parameter
except:
    print "Hata: Thread başlatılamıyor!"

while 1:
    pass
```


Ek-B

Modüllerin Pil Ömrü Hesabı;

$$I_{avg} = I_{avg.m} + I_{avg.t} + I_{avg.losses}$$

$$I_{avg} = \frac{T_{awm}}{T_{mc} \times 1000} \times I_{mc} \times 1000 + \frac{T_{awt}}{T_{tc} \times 1000} \times I_{tc} \times 1000 + I_{avg.losses}$$

$$T_{bat.h} = \frac{C_{new}}{I_{avg}} \times 1000 \times \frac{100 - \left(1 - \frac{C_{10y}}{C_{new}}\right) \times 100}{T_{tc} \times 1000} = \frac{1000 \times C_{10y}}{I_{avg}}$$

Yukarıdaki formülleri kullanarak örneğin 15 dakikalık aralıklarla 23 byte'lık bir veri paketini 4800bps hızında gönderim yapıldığını hesaplayalım.

$$T_{awt} = \frac{23 \times 8}{4800} \times 1000 = 38 \text{ ms} \approx 40 \text{ ms}$$

$$T_{tc} = 24 \text{ h} = 24 \times 60 \times 60 \times 1000 = 900.000 \text{ ms}$$

$$I_{avg.t} \approx 80 \text{ mA}$$

$$T_{awm} \approx 20 \text{ ms}$$

$$T_{mc} \approx 16 \text{ ms}$$

$$I_{mc} \approx 3 \text{ mA}$$

$$I_{avg.losses} \approx 5 \mu A$$

$$I_{avg} = \frac{20}{16 \times 1000} \times 3 \times 1000 + \frac{40}{900000} \times 80 \times 1000 + 5 \approx 12.25 \mu A$$

$$T_{bat.h} \approx \frac{1000 \times 2250}{12.25} \approx 183673 \text{ h} \approx 21 \text{ yıl}$$

KİŞİSEL YAYINLAR VE ESERLER

- [1] **Görenekli K.**, Becerikli Y., Development Of Electronic Water Meter Based On Wireless Network And RF, Engineering, *Technology & Applied Science Research*, 2019, **9**(2), 3905-3908.



ÖZGEÇMİŞ

İlkokulu Almanya'da tamamladıktan sonra 1983 yılında Türkiye'ye geldi. Ortaokulu ve Liseyi İstanbul'da bitirdikten sonra yılında Yıldız Teknik Üniversitesi Bilgisayar Bilimleri ve Mühendisliği bölümüne başladı. 1996 yılında Bilgisayar Mühendisi oldu. 2016 yılında Kocaeli Üniversitesi Bilgisayar Mühendisliği'nde Yüksek Lisans eğitimine başladı. 2010 yılından beri Kocaeli Büyükşehir İsu Genel Müdürlüğü Bilgi İşlem Daire Başkanlığı'nda çalışmaktadır.

