

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİYOMEDİKAL MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

**BİYOMEDİKAL UYGULAMALARI İÇİN NESNELERİN
İNTERNETİ TABANLI GERÇEK ZAMANLI VERİ TOPLAMA
VE İZLEME PLATFORMUNUN GELİŞTİRİLMESİ**

FARUK AKTAŞ

KOCAELİ 2018

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

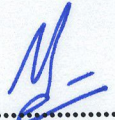
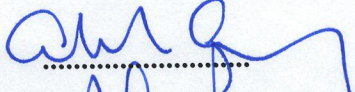
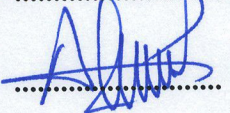
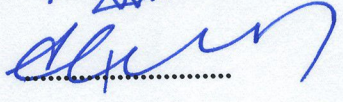
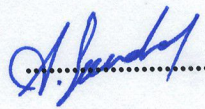
BİYOMEDİKAL MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

BİYOMEDİKAL UYGULAMALARI İÇİN NESNELERİN
İNTERNETİ TABANLI GERÇEK ZAMANLI VERİ TOPLAMA
VE İZLEME PLATFORMUNUN GELİŞTİRİLMESİ

FARUK AKTAŞ

Prof. Dr. Yunus Emre ERDEMLİ
Danışman, Kocaeli Üniversitesi
Prof. Dr. Celal ÇEKEN
Jüri Üyesi, Sakarya Üniversitesi
Prof. Dr. Ali TANGEL
Jüri Üyesi, Kocaeli Üniversitesi
Doç. Dr. Ali ÇALHAN
Jüri Üyesi, Düzce Üniversitesi
Dr. Öğr. Üyesi Adnan SONDAŞ
Jüri Üyesi, Kocaeli Üniversitesi


.....

.....

.....

.....

.....

Tezin Savunulduğu Tarih: 06.07.2018

ÖNSÖZ VE TEŞEKKÜR

Teknolojinin ve haberleşme sistemlerinin hızla gelişmesiyle birlikte günlük hayatımız içindeki birçok verinin toplanması, analiz edilmesi, saklanması ve bilgilere göre karar verme mekanizmalarının oluşturulması büyük önem kazanmıştır. Son gelişmeler ile birlikte, fiziksel dünyayı gözleme yeteneğine sahip, veri işleyebilen, karar verme tabanlı işlemleri gerçekleştirebilen kablosuz algılayıcı ve eyleyici ağları, cihazları veya kişileri tanımlayabilen sistemler, bulut bilişim ile veri tabanlı sistemlerini içeren nesnelere interneti teknolojisini ortaya çıkmıştır. Bu teknolojiyi oluşturan cihazlar internet üzerinden birbirleriyle haberleşebilmekte ve böylece bilgi paylaşımı yapabilmektedir. Bu özelliğinden dolayı nesnelere interneti teknolojisi, günümüzde, akıllı şehir, akıllı ev, akıllı enerji sistemlerinin yanı sıra bu teze de konu olan biyomedikal sistemlerinde etkin bir şekilde kullanılmaktadır.

Bu çalışmada, biyomedikal uygulamaları için nesnelere interneti tabanlı gerçek zamanlı veri toplama ve hasta izleme platformu geliştirilmiştir. Tez çalışması boyunca göstermiş olduğu sabır, verdiği sürekli destek, emek ve imkân, sağladığı huzurlu çalışma ortamı için doktora tez danışmanım Prof. Dr. Yunus Emre ERDEMLİ'ye teşekkür ederim. Doktora eğitimimin başından sonuna kadar değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarına yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran saygıdeğer hocam Prof. Dr. Celal ÇEKEN'e, özellikle web tabanlı uygulama kısmında değerli görüşleriyle tez çalışmasına katkıda bulunan Öğr. Gör. Yavuz Selim FATİHOĞLU'na, tez çalışmasının başından sonuna kadar manevi desteklerini benden esirgemeyen tüm iş arkadaşlarıma teşekkürlerimi sunarım.

Çalışmam sürecinde manevi desteğini, sabrını ve sevgisini eksik etmeyen sevgili eşim Gülay AKTAŞ'a, bugünlere gelmemi sağlayan anneme, babama ve kardeşlerime saygı, sevgi ve sonsuz teşekkürler.

Bu tez çalışmasını en değerli varlığım olan oğlum Arda Aras'a ithaf ediyorum.

Temmuz – 2018

Faruk AKTAŞ

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ	iii
TABLolar DİZİNİ	vi
SİMGELER VE KISALTMALAR DİZİNİ	vii
ÖZET.....	xi
ABSTRACT	xii
GİRİŞ.....	1
1. GELİŞTİRİLEN SİSTEM İLE İLGİLİ GENEL BİLGİLER.....	3
1.1. Literatür Özeti.....	4
1.2. Önerilen Sistemin Bileşenleri ve Tez Çalışmasının Katkıları	7
2. NESNELERİN İNTERNETİ BİLEŞENLERİ	10
2.1. Kablosuz Vücut Alan Ağları (KVAA) ve IEEE 802.15.4 Standardı.....	11
2.1.1. IEEE 802.15.4 standardı.....	12
2.2. Radyo Frekanslı Tanılama (RFID)	20
2.2.1. Uygulama katmanı	22
2.2.1.1. Seçim.....	22
2.2.1.2. Sayım.....	22
2.2.1.3. Erişim.....	24
3. GELİŞTİRİLEN SİSTEMİN BENZETİMİ	25
3.1. Benzetim Sonuçları ve Performans Değerlendirmeleri	30
3.2. RFID İletişim Sistemi Yapısı.....	41
3.3. Sonuç.....	45
4. GELİŞTİRİLEN GERÇEK ZAMANLI HASTA İZLEME SİSTEMİ	46
4.1. Donanım Bileşenleri	47
4.1.1. Mikrodenetleyici geliştirme kartı.....	47
4.1.2. Kablosuz haberleşme modülü	51
4.1.3. Kablosuz modülleri geliştirme kartına bağlama cihazları.....	54
4.1.4. Vücut sıcaklığı algılayıcısı.....	55
4.1.5. Nabız ölçer algılayıcı	57
4.1.6. RFID okuyucu kartı	59
4.2. Geliştirilen Gerçek Zamanlı Platformun Bölümleri	61
4.2.1. KVAA algılayıcı düğümü	61
4.2.2. Koordinatör düğüm	73
4.2.3. Web tabanlı uygulama yazılımı.....	77
4.2.3.1. Koordinatör düğümden gelen verilerin okunması.....	78
4.2.3.2. Web tabanlı gerçek zamanlı izleme uygulaması.....	81
4.3. Gerçek Zamanlı Sistemin QoS Değerlendirmesi.....	87
4.4. Sonuç.....	89
5. SONUÇLAR VE ÖNERİLER	90
KAYNAKLAR	92
KİŞİSEL YAYIN VE ESERLER	114
ÖZGEÇMİŞ	115

ŞEKİLLER DİZİNİ

Şekil 1.1.	IoT tabanlı veri toplama ve analiz platformunun fiziksel bileşenleri.....	8
Şekil 2.1.	Nesnelerin interneti uygulama alanları	10
Şekil 2.2.	Temel KVAA yapısı ve sistemin çalışması	12
Şekil 2.3.	Zigbee kablosuz ağ protokolü katmanları.....	13
Şekil 2.4.	Süper çerçeve formatı (superframe)	15
Şekil 2.5.	N1 düğümüne ait özellik ayar ekranı	18
Şekil 2.6.	MAC katmanı veri paketi yapısı	18
Şekil 2.7.	Kılavuzlu modda koordinatör-düğüm arası veri iletişimi.....	18
Şekil 2.8.	RFID genel çalışma prensibi.....	20
Şekil 2.9.	RFID etiketleri ve barkod arasındaki farklar	21
Şekil 2.10.	RFID etiketi ve iç yapısı	21
Şekil 2.11.	Temel RFID protokol modeli	22
Şekil 2.12.	Query komut formatı	23
Şekil 2.13.	QA komut formatı.....	23
Şekil 2.14.	QR komut formatı.....	24
Şekil 2.15.	ACK komut formatı.....	24
Şekil 3.1.	Geliştirilen nesnelerin interneti tabanlı sistemin Riverbed benzetim çalışma alanındaki görüntüsü.....	25
Şekil 3.2.	N1, N2, and N3 düğümlerinin Riverbed düğüm modelleri	26
Şekil 3.3.	RFID-GSM arayüzlerini içeren önerilen PANc (koordinatör) düğüm modeli	26
Şekil 3.4.	N1 düğümüne ait ayarlar.....	27
Şekil 3.5.	N3 düğümüne ait ayarlar.....	28
Şekil 3.6.	Önerilen KVAA'nın benzetim senaryosunun sıralı diyagramı.....	29
Şekil 3.7.	GSM_MODULE işlem modülü ve kodları.....	30
Şekil 3.8.	Baz istasyonuna gelen veriler ile Matlab ortamındaki izleme sonuçları.....	30
Şekil 3.9.	N1 düğümü senaryo 1, 2 ve 3 için EKG değerleri.....	32
Şekil 3.10.	N2 düğümü senaryo 1, 2 ve 3 için EKG değerleri.....	32
Şekil 3.11.	N3 düğümü senaryo 1, 2 ve 3 için vücut sıcaklığı değerleri.....	33
Şekil 3.12.	N1 düğümü senaryo 1, 2 ve 3 için UUG değerleri	36
Şekil 3.13.	N3 düğümü senaryo 1, 2 ve 3 için UUG değerleri	36
Şekil 3.14.	Senaryo 1, 2 ve 3 için N2 düğümü UUG değerleri.....	37
Şekil 3.15.	Senaryo 1, 2 ve 3 için N1 düğümü veri oranı değerleri.....	38
Şekil 3.16.	Senaryo 1, 2 ve 3 için N2 düğümü veri oranı değerleri.....	38
Şekil 3.17.	Senaryo 1, 2 ve 3 için N3 düğümü veri oranı değerleri.....	38
Şekil 3.18.	Senaryo 1, 2 ve 3 için N1 düğümü enerji tüketim değerleri.....	40
Şekil 3.19.	Senaryo 1, 2 ve 3 için N2 düğümü enerji tüketim değerleri.....	40
Şekil 3.20.	Senaryo 1, 2 ve 3 için N3 düğümü enerji tüketim değerleri.....	40
Şekil 3.21.	Tez çalışmasında kullanılan RFID okuyucu-etiket iletişim protokolünün sıralı diyagramı.....	41

Şekil 3.22.	Önerilen RFID haberleşme mekanizması benzetim senaryosunun sıralı diyagramı	42
Şekil 3.23.	Önerilen RFID okuyucu işlem (process) modelinin algoritması	43
Şekil 3.24.	Önerilen RFID etiket işlem (process) modelinin algoritması	43
Şekil 3.25.	Örnek bir RFID hasta tanımlama sürecinin benzetim çalışma alanındaki görüntüsü	45
Şekil 4.1.	Geliştirilen gerçek zamanlı hasta izleme sisteminin yapısı	46
Şekil 4.2.	Gerçekleştirilen gerçek zamanlı sistemin sıralı akış diyagramı.....	47
Şekil 4.3.	Arduino Uno R3 üstten görünüşü	48
Şekil 4.4.	XBee PRO S1 modülü	51
Şekil 4.5.	Basit bir Zigbee ağ yapısı	53
Şekil 4.6.	XBee shield ürünü	54
Şekil 4.7.	KVAA erişim noktası (Arduino, XBee shield ve XBee Pro S1 modülü).....	55
Şekil 4.8.	MLX 90614 algılayıcısı bacak bağlantıları	56
Şekil 4.9.	GY906 modülü	57
Şekil 4.10.	Pulse sensor nabız algılayıcısı ve bacak bağlantıları	58
Şekil 4.11.	MFRC522 RFID okuyucu kartı ve etiketler	60
Şekil 4.12.	Temel algılayıcı ve koordinatör düğüm yapısı	62
Şekil 4.13.	MLX90614 Arduino bağlantısı.....	64
Şekil 4.14.	Nabız ölçer algılayıcı Arduino bağlantısı	65
Şekil 4.15.	Nabız ölçer ve ölçüm yuvası.....	65
Şekil 4.16.	Nabız bilgisinin alınması	66
Şekil 4.17.	Arduino IDE ekranı	66
Şekil 4.18.	IDE programı örnek kodları.....	67
Şekil 4.19.	Algılayıcı düğüm tarafında verilerin seri porttan gönderilmesi.....	69
Şekil 4.20.	Verilerin seri porttan okunması	69
Şekil 4.21.	X-CTU ayarları için XBee Arduino bağlantısı	70
Şekil 4.22.	X-CTU yazılımında bulunan XBee modül bilgileri	71
Şekil 4.23.	AT modda adres bilgilerinin ayarlanması.....	72
Şekil 4.24.	X-CTU algılayıcı düğüm üzerindeki XBee modül ayarları.....	72
Şekil 4.25.	Erişim noktası tarafındaki XBee Pro S1 ayar ekranı	74
Şekil 4.26.	MF-RC522 okuyucu kartı ile Arduino bağlantısı	75
Şekil 4.27.	Koordinatör düğümü IDE kodu	76
Şekil 4.28.	Koordinatör düğümünün bağlı olduğu COM4 portundan okunan değerler.....	77
Şekil 4.29.	Node.js çalışma şekli	79
Şekil 4.30.	Node.js platformunda seri porttan gelen verilerin okunması.....	80
Şekil 4.31.	MongoDB veri tabanı ekranı	81
Şekil 4.32.	Gerçek zamanlı izleme sistemi giriş ekranı	82
Şekil 4.33.	Normal değerler için nabız grafiği ekranı	83
Şekil 4.34.	Yüksek değerler için nabız grafiği ekranı.....	84
Şekil 4.35.	Düşük değerler için nabız grafiği ekranı.....	84
Şekil 4.36.	Sıcaklık ekranı	85
Şekil 4.37.	Gerçek zamanlı sistemin çalışma ekranı (hasta ismi ve fizyolojik veriler ile birlikte).....	86
Şekil 4.38.	İzleme sistemi nabız grafiği mobil ekran görüntüsü.....	87
Şekil 4.39.	İzleme sistemi vücut sıcaklığı grafiği mobil ekran görüntüsü.....	87

Şekil 4.40. Algılayıcı düğümden gönderilen değerlerin veri tabanına yazılma zamanları 88



TABLULAR DİZİNİ

Tablo 1.1.	Farklı biyomedikal uygulamaları için QoS gereksinimleri	4
Tablo 3.1.	Benzetim parametreleri	31
Tablo 4.1.	Arduino Uno kartı üzerindeki donanımların yerleri.....	49
Tablo 4.2.	Arduino Uno teknik özellikleri	49
Tablo 4.3.	XBee S1 ve XBee Pro S1 modüllerinin özellikleri	52
Tablo 4.4.	MFRC522 kartı özellikleri	60
Tablo 4.5.	Çözünürlük değerlerine göre analog değerlerin ikili ve onlu karşılıkları	63
Tablo 4.6.	Çocuk ve erişkinler için normal vücut sıcaklığı ve kalp atım oranı değerleri	83

SİMGELER VE KISALTMALAR DİZİNİ

SpO ₂	: Oksijen saturasyonu
kHz	: KiloHertz
MHz	: MegaHertz
Ghz	: GigaHertz
Kbps	: Kilo bit per second, (Saniyede iletilen kilobit sayısı)
mW	: MiliWatt
V _{dd}	: Besleme gerilimi, (V)
V _{ss}	: Besleme gerilimi toprağı, (V)
RN16	: 16 bitlik rastgele sayı
Hz	: Hertz
sn	: Saniye
bps	: Bit per second, (Saniyede gönderilen bit sayısı)
ms	: Milisaniye
mV	: MiliVolt
°C	: Celcius
Kbit	: Kilobit
mA	: MiliAmper, (A)
J	: Joule
mm	: Milimetre
V	: Volt
DC	: Doğru akım, (V)
Gnd	: Ground (Toprak hattı), (V)
Aref	: Analog referans, (V)
KB	: Kilobayt
Km	: Kilometre
Ft	: Fit
dBm	: 1 miliwatt karşılığı desibel
µA	: Mikroamper, (A)
µm	: Mikrometre
V ₀	: Termofil çıkışı, (V)
T _a	: Metal kılıf sıcaklığı, (K)
T ₀	: Nesne sıcaklığı, (K)
A	: Hassasiyet, (%)
T _{raw}	: Hesaplanan ham sıcaklık, (K)

Kısaltmalar

ACK	: Acknowledge (Kabul)
ADC	: Analog Digital Converter (Analog Dijital Çevirici)
AES	: Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
API	: Application Programming Interface (Programlama Arayüz Uygulaması)

ASCII	: American Standard Code for Information Interchange (Bilgi Değişimi için Amerikan Standart Kodlama Sistemi)
ASSP	: Application Specific Standart Product (Uygulamaya Özgü Standart Ürün)
AT	: Application Transparent (Saydam Uygulama)
BĞİ	: Bant Genişliği İhtiyacı
BI	: Beacon Interval (Kılavuz Zamanı)
BO	: Beacon Order (Kılavuz Sırası)
C	: Coordinator (Koordinatör)
CAP	: Contention Access Period (Çekişme Erişim Periyodu)
CE	: Coordinator Enable (Koordinatör Seçili)
CFP	: Contention Free Period (Çekişme Serbest Periyodu)
CMOS	: Complementary Metal Oxide Semiconductor (Bütünleyici Metal Oksit Yarı İletken)
CRC16	: Cyclic Redundancy Check 16 bit (16 Bit Döngüsel Artıklık Denetimi)
CSMA/CA	: Carrier Sense Multiple Access / Collision Avoidedance (Taşıyıcı Algılayıcı Çoklu Erişim / Çarpışmadan Kaçınmalı)
CSS	: Cascading Style Sheets (Basamaklı Stil Şablonları)
DH	: Destination Address High (Hedefin Yüksek Adresi)
DL	: Destination Address Low (Hedefin Düşük Adresi)
DOM	: Document Object Model (Dosya Nesne Modeli)
DR	: Divide Ratio (Bölme Oranı)
DSP	: Digital Signal Processor (Dijital Sinyal İşleyici)
E	: End Device (Son Cihaz)
EEG	: Elektroansefalogram
EKG	: Elektrokardiyogram
EMG	: Elektromiyogram
EPC	: Electronic Product Code (Elektronik Ürün Kodu)
FOV	: Field of View (Görüş Açısı)
GPRS	: General Packet Radio Service (Genel Radyo Paket Servisi)
GSM	: Global System for Mobile Communication (Küresel Mobil İletişim Sistemi)
GTS	: Guaranteed Time Slot (Garanti Edilmiş Zaman Dilimi)
GTSİ	: Garanti Edilmiş Zaman Dilimi İsteği
HEX	: Hexadecimal (Onaltılı Sayı)
HTML	: Hyper Text Markup Language (Zengin Metin İşaretleme Dili)
ICSP	: In-Circuit Serial Programming (Devre Üzerinde Seri Programlama)
ID	: Identification (Tanılama)
IEEE	: The Institute of Electrical and Electronics Engineers (Elektrik ve Elektronik Mühendisleri Enstitüsü)
IoT	: Internet of Things (Nesnelerin İnterneti)
IP	: Internet Protocol (İnternet Protokolü)
IPv6	: Internet Protocol Version 6 (İnternet Protokolü Versiyon 6)
IR	: Infrared Radiation (Kızılötesi Yayınlam)
ISM	: Industrial, Scientific, Medical (Endüstriyel, Bilimsel, Tıbbi)
ISO	: International Standart Organization
I/O	: Input/Output (Giriş/Çıkış)
I ² C	: Inter-Integrated Circuit (Entegre Arası Cihaz)
KAA	: Kablosuz Algılayıcı Ağlar

KKAA	: Kablosuz Kişisel Alan Ağı
KVAA	: Kablosuz Vücut Alan Ağları
MAC	: Medium Access Control (Ortam Erişim Kontrolü)
MGTS	: Maximum Guaranteed Time Slot (Maksimum Garanti Edilmiş Zaman Dilimi)
MISO	: Master Input Slave Output (Ana Cihaz Giriş Bağımlı Cihaz Çıkış)
MOSI	: Master Output Slave Input (Ana Cihaz Çıkış Bağımlı Cihaz Giriş)
MSDU	: MAC Service Data Unit (OEK Servisi Veri Birimi)
NACK	: Negative Acknowledgement (Negatif Onay)
NFC	: Near Field Communication (Yakın Mesafe Haberleşme)
Ns-2	: Network Simulator-2 (Ağ Simülatörü-2)
N1	: node_1 Düğümü
N2	: node_2 Düğümü
N3	: node_3 Düğümü
OEK	: Ortam Erişim Kontrol
OSI	: Open System Interconnect (Açık Sistem Bağlantı)
PAN	: Personal Area Network (Kişisel Alan Ağı)
PANc	: Personal Area Network Coordinator (Kişisel Alan Ağı Koordinatörü)
PC	: Product Code (Ürün Kodu)
PDA	: Personal Data Assistant (Kişisel Veri Asistanı)
PDU	: Protocol Data Units (Protokol Veri Birimleri)
PTAT	: Proportional to Absolute Temperature Sensor (Mutlak Sıcaklık Algılayıcısına Orantılı)
PTC	: Positive Temperature Coefficient (Sıcaklıkla Direnci Artan Termistör)
PWM	: Pulse Width Modulation (Darbe Genişlik Modülasyonu)
Q	: Query (Sorgu)
QA	: Query Adjust (Sorgu Ayarı)
QoS	: Quality of Service (Servis Kalitesi)
QPSK	: Quadrature Phase Shift Keying (Dörtlü Faz Kaydırmalı Anahtarlama)
QR	: Query Repeat (Sorgu Tekrarı)
R	: Router (Yönlendirici)
RF	: Radio Frequency (Radyo Frekansı)
RFID	: Radio Frequency Identification (Radyo Frekanslı Tanımlama)
Rx	: Receiver (Radyo Sinyali Alıcı)
SCL	: Serial Clock (Seri Saat Sinyal)
SD	: Superframe Duration (Süper Çerçeve Zamanı)
SDA	: Serial Data (Seri Veri)
SH	: Serial Number High (Seri Numarası Yüksek Adresi)
SKS	: Saniyede Üretilen Kılavuz Sayısı
SL	: Serial Number Low (Seri Numarası Düşük Adresi)
SMBUS	: System Management Bus (Sistem Yönetim Veriyolu)
SMS	: Short Message Service (Kısa Mesaj Servisi)
SO	: Superframe Order (Süper Çerçeve Sırası)
SPI	: Serial Peripheral Interface (Seri Çevresel Arayüz)
SZDS	: Süper Çerçeve Zaman Dilimi Sayısı
TTL	: Transistor to Transistor Logic (Transistörden Transistör Lojiğe)
Tx	: Radio Transmitter (Radyo Sinyali Gönderici)
UART	: Universal Asenkron Receiver Transmitter (Evrensel Asenkron Alıcı Gönderici)

UHF	: Ultra High Frequence (Çok Yüksek Frekans)
UUG	: Uçtan Uca Gecikme
USB	: Universal Serial Bus (Evrensel Seri Yolu)
WBAN	: Wireless Body Area Networks (Kablosuz Vücut Alan Ağları)
Wi-Fi	: Wireless Fidelity (Kablosuz Bağlantı Alanı)
WPAN	: Wireless Personal Area Network (Kablosuz Kişisel Alan Ağı)
3G	: 3rd Generation (Üçüncü Nesil)
4G	: 4th Generation (Dördüncü Nesil)
5G	: 5th Generation (Beşinci Nesil)



BİYOMEDİKAL UYGULAMALARI İÇİN NESNELERİN İNTERNETİ TABANLI GERÇEK ZAMANLI VERİ TOPLAMA VE İZLEME PLATFORMUNUN GELİŞTİRİLMESİ

ÖZET

Biyomedikal sistemlerde, nesnelere interneti (Internet of Things - IoT) tabanlı, uzaktan sağlık hizmetleri, tanı, önleme, tedavi, izleme gibi biyomedikal uygulamalar kullanılmaktadır. Kablosuz vücut alan ağları (KVAA) ve radyo frekanslı tanımlama sistemleri (Radio Frequency Identification - RFID) IoT teknolojisinin en önemli anahtar bileşenleridir. Bu tez çalışmasında, hastane bilgi sistemleri için KVAA'lar ve RFID teknolojileri kullanılarak yeni bir IoT tabanlı veri toplama ve izleme platformu geliştirilmiştir. Tasarlanan platform öncelikle Riverbed Modeler yazılımı kullanılarak modellenmiş ve benzetimi gerçekleştirilmiştir. Ayrıca platform benzetim senaryosu ile örtüşecek şekilde gerçek zamanlı olarak gerçekleştirilmiştir. Önerilen platformun özgün özellikleri: i) ISO/IEEE 11073 standardı tarafından belirlenen servis kalitesi (Quality of Service - QoS) kriterleri (veri hızı ve gecikme süresi) enerji etkin bir sistem kullanılarak sağlanabilmektedir, ii) Hastane bilgi sistemlerinde yer alan fizyolojik ve kişisel veri takibi, geliştirilen benzetim ve gerçek zamanlı uygulama ortamı ile zaman verimli bir şekilde gerçekleştirilebilmektedir.

Anahtar Kelimeler: Kablosuz Vücut Alan Ağları, Nesnelere İnterneti, Radyo Frekanslı Tanımlama, Sağlık Hizmetleri, Servis Kalitesi.

DEVELOPMENT OF IOT-BASED REAL-TIME DATA ACQUISITION AND MONITORING PLATFORM FOR BIOMEDICAL APPLICATIONS

ABSTRACT

Internet of Things (IoT) based biomedical applications are deployed in biomedical systems such as healthcare/telecare, diagnosis, prevention, treatment, monitoring. Wireless Body Area Networks (WBANs) and Radio Frequency Identification (RFID) systems, also called as enabling technologies, are key components of the IoT concept. In this thesis, a new IoT-based data collection and monitoring platform using WBANs and RFID technologies is developed for hospital information systems. The developed platform has been modeled and simulated using Riverbed Modeler software. It has then been realized in real time based on the simulation scenario. The proposed platform has the following novel features: i) Quality of Service (QoS) criteria (data rate and latency) specified by the ISO/IEEE 11073 standard are satisfied by using the energy-aware system, ii) Physiological and personal data in hospital information systems can easily be tracked via the time-saving simulation and real time environment.

Keywords: Wireless Body Area Networks, Internet of Things, Radio Frequency Identification, Healthcare, Quality of Service.

GİRİŞ

Dünya genelinde özellikle gelişmiş ülkelerdeki nüfusun yaşlanmasıyla birlikte, kronik hastalıklara yakalanan, psikolojik sorunları olan, yaralanan ve ciddi bulaşıcı hastalıklara yakalanan kişilerin sayısının artması, sağlık hizmetlerinin kullanımını ve maliyetlerini artırmıştır. Bu nedenle, kablosuz teknolojiler kullanılarak hastanelerde veya daha küçük ölçekli medikal merkezlerde verilen sağlık hizmetlerinin bu yerler dışında da verilebilmesi bir zorunluluk haline gelmiştir. Hastalara uzaktan bakım hizmetlerinin kablosuz teknolojiler kullanılarak verilmesi, sağlık hizmeti veren yerlerin iş yükünü büyük ölçüde hafifletmiştir. Ayrıca, hastaların hastane ortamında uzun süre bakılması hem ciddi ölçüde maddi yüke neden olmakta hem de hastanede gerçekleştirilen uzun süreli tedaviler, hastada hastaneye yatış ile ilgili birçok fiziksel, ruhsal ve sosyal sorunlara neden olabilmektedir. Bu sorunların azaltılabilmesi için fizyolojik verilerin ölçümlerinin hastanın günlük yaşamına müdahale etmeden ve kişinin konforuna dikkat edilerek yapılması kablosuz sistemlerin kullanımı ile mümkün olabilmektedir [1].

Kablosuz sistemlerin hastalar açısından en büyük avantajı, hareket sınırlılığını ortadan kaldırması ve hastanın isteğine bırakılmadan fizyolojik ölçümlerin yapılabilmesidir. Uzaktan hasta izleme ile ilgili en çok kullanılan sistem Kablosuz Vücut Alan Ağı (KVAA) sistemleridir. KVAA içerisinde bulunan hasta üzerine giydirilebilen veya vücuduna yerleştirilebilen algılayıcılar ile fizyolojik veriler, hastaya acı vermeden ve günlük hayatına müdahale etmeden toplanabilmektedir. Teknolojideki son gelişmeler ile birlikte fiziksel dünyayı gözleme yeteneğine sahip, veri işleyebilen, karar verme tabanlı işlemleri gerçekleştirebilen kablosuz algılayıcı ve eyleyici ağlar, cihazları veya kişileri tanımlayabilen sistemler, bulut bilişim ile veri tabanı sistemlerini içeren nesnelerin interneti teknolojisini ortaya çıkmıştır.

Nesnelerin İnterneti (Internet of Things - IoT), fiziksel nesnelerin bilgi paylaşımını beraberce karar verebilmeleri için, birbirlerini görme, duyma, düşünme, bir araya gelerek “konuşma” olanağı vermektedir [2]. Bu tezde, nesnelerin interneti yerine genel

bir terim halini almış olan IoT kısaltması kullanılmıştır. IoT; gömülü cihazlar, haberleşme protokolleri, algılayıcı ağlar, internet protokol ve uygulamaları gibi temel teknolojilerin geleneksel olanından akıllı olanına dönüşümü olarak tanımlanabilmektedir [3]. Geniş çaplı bir bağlantı yapısına sahip IoT tabanlı sistemler ile tedavi, tanı, iyileşme süreci, envanter takibi, ilaç tedavisi gibi tüm sağlık ile ilgili bilgiler, etkin bir şekilde toplanabilmekte, yönetilebilmekte ve paylaşılabilir. Kablosuz arayüzleri ve internet bağlantıları içermesi sebebiyle IoT sistemi, sağlık hizmetleri mobilitesini desteklemektedir [4].

Bu tez çalışmasında, hastane bilgi sistemleri için yeni bir IoT tabanlı veri toplama ve izleme platformu geliştirilmiştir. Tez çalışması beş ana bölümden oluşmaktadır. Birinci bölümde, geliştirilen sistem ile ilgili genel bilgiler, ilgili literatür özeti ve tez çalışmasının katkıları sunulmaktadır. Bölüm 2’de, IoT ve IoT bileşenleri olan KVAA ve RFID sistemleri hakkında genel bilgiler verilmekte ve özel olarak KVAA içerisinde haberleşme protokolü olarak kullanılan IEEE 802.15.4 standardı anlatılmaktadır. Bölüm 3’te, IoT tabanlı olarak geliştirilen sistemin benzetimi, benzetim sonuçları, KVAA ve RFID bileşenlerinin performans değerlendirmelerine yer verilmektedir. Bölüm 4’te, geliştirilen gerçek zamanlı hasta izleme platformunun bileşenleri, çalışma yapısı ve servis kalitesi değerlendirilmektedir. Tez çalışmasının sonuçları, katkıları ve sonraki çalışmalar ile ilgili öneriler son bölümde sunulmaktadır.

1. GELİŞTİRİLEN SİSTEM İLE İLGİLİ GENEL BİLGİLER

IoT sistemleri, Kablosuz Vücut (veya Kişisel) Alan Ağları (KVAA veya KKAA), Radyo Frekanslı Tanılama (Radio Frequency Identification - RFID) sistemleri, cep telefonu, dizüstü bilgisayar, tablet, Wi-Fi, 3G/4G/5G, adresleme şemaları ve veri analitik sistemleri gibi çeşitli bileşenlere sahiptir. Nesnelerin interneti'nin sağlık hizmetleri ile uygulamaları genel olarak; nesnelerin ve insanların (hasta ve personel) takibi, kimlik tanımlama (identification) ve kimlik doğrulamaları (authentication), otomatik veri toplama ve uzaktan algılama olarak sınıflanabilmektedir [2, 5-8].

KVAA'lar, sağlık sistemlerinde IoT'nin önemli bir bileşenidir. Bir KVAA sistemi, genel olarak algılayıcı düğümleri ve erişim noktasını içerir. Gömülü bir cihaz olan algılayıcı düğüm vücut sıcaklığı, SpO₂, kan basıncı ve elektrokardiyogram (EKG) gibi birçok hayati fizyolojik veriyi, insan vücudundan algılama ve kablosuz bileşenlerini kullanarak erişim noktasına gönderme yeteneğine sahiptir [9, 10]. Ağ koordinatörü olarak da adlandırılan erişim noktası, algılayıcı düğümden gelen verileri sıkıştırma, filtreleme ve çıkarma gibi gerekli bir ön işleminden geçirdikten sonra sağlık hizmetleri görüntüleme merkezine gönderir [11-15]. IoT sistemlerde aygıtları internete bağlayan, algılayıcılardan verileri toplayan koordinatör düğümlerdir (erişim noktası). Koordinatör düğümlerin internete erişebilir olması aygıtların birbirleri ile internet üzerinden haberleşebilmelerini sağlamaktadır. KVAA sistemlerinde, düşük enerji tüketimlerinden dolayı genellikle Zigbee [16] haberleşme protokolü kullanılmaktadır. Zigbee haberleşme protokolünde aygıtların IP [17] adresleri bulunmamaktadır. IP adresleri, ağda bulunan algılayıcı düğümlerden gelen verileri toplayan ve bilgisayara bağlanabilen koordinatör düğümlerde bulunmaktadır.

IoT yapısının diğer bir önemli bileşeni de RFID sistemleridir. RFID, bir okuyucu (sorgulayıcı olarakta adlandırılabilir) ve etiket veya etiketlerden meydana gelen bir otomatik tanımlama sistemidir. Bir etiket, bir anten ve içerisinde benzersiz bir tanılayıcıyı depolayan bir mikroçip içermektedir. RFID etiketleri dahili enerji kaynağı kullanıp kullanmadıklarına göre aktif, yarı aktif ve pasif olarak sınıflandırılabilirler.

RFID okuyucular veri aktarımı ve okuma işlemlerinde çeşitli uygulamalar için 125 kHz–134,2 kHz / 13,56 MHz / 860 MHz–960 MHz / 2,45 GHz radyo frekans bantlarını kullanabilmektedirler. Okuyucu, tipik olarak, endüktif kuplaj kullanarak sorgulama aralığı içindeki bir dizi pasif etiketi bilgi toplamak için enerjiler. Yani, okuyucu bir elektromanyetik sinyal yayınlar ve daha sonra etiketin anteni, sinyal enerjisini yerleşik bir kapasitöre alır ve depolar.

IoT tabanlı sağlık uygulamalarında RFID teknolojisi etkili bir şekilde kullanılmaktadır. Bu uygulamalara; yenidoğan bebek ve hasta tanımlaması, biyomedikal envanterlerin konum takibi, tıbbi tedavilerin takibi ve doğrulanması, bir sağlık merkezinde prosedür yönetimi ve cerrahi süreç yönetimi örnek gösterilebilir [20].

Biyomedikal uygulamaları için KVAA tasarımında servis kalitesi (QoS) oldukça önemli bir yere sahiptir. Bu kapsamda, KVAA'da QoS gereksinimlerinden iletim gecikmesi ve veri iletim oranı önemli parametreler olarak karşımıza çıkmaktadır. Hastalardan toplanan fizyolojik veriler, bu metriklerin QoS gereklilikleri (Tablo 1.1) karşılanmadığı takdirde yanlış bilgi olarak kabul edilmektedir [18, 19].

Tablo 1.1. Farklı biyomedikal uygulamaları için QoS gereksinimleri

Uygulama	Veri Oranı	Gecikme
EKG (3 uçlu)	2,4 Kbps	< 300 ms
EKG (5 uçlu)	10 Kbps	< 300 ms
Nabız	2 Kbps	< 300 ms
Kan Basıncı	120 bps	< 3 s
Sıcaklık	120 bps	< 3 s
SpO ₂	120 bps	< 3 s

1.1. Literatür Özeti

Literatürde, IoT tabanlı olarak yapılan, içerisinde KVAA ile fizyolojik değerleri hastalardan toplayarak onları izleyen ve analiz eden, RFID sistemlerini beraberinde bulunduran birçok çalışma bulunmaktadır.

Yang ve diğ., içerisinde akıllı ilaç kutusu, akıllı ilaç paketi ve EKG verilerini ölçüp RFID ile haberleşebilen giyilebilir bir devre bulunan IoT tabanlı bir sağlık hizmeti sistemi önermiş ve gerçekleştirmişlerdir. Sistemdeki tüm bilgilerin yönetimini gerçekleştirmek amacıyla bir yönetim yazılımını geliştirmişlerdir [5].

Luo ve diğ., IoT altyapısı kullanarak uzaktaki hastalardan toplanan detaylı fizyolojik verileri elde etme ve bu verilerin elektronik tıbbi kayıtlarının tutulmasıyla ilgili bir sistem tasarlamışlardır. Sistemin gelen verileri bir program ile analiz ederek tanı yapabildiği ve sağlık hizmetleri ile ilgili birçok alanda uygulanabileceği belirtilmiştir [6].

Cubo ve diğ., bulut bilişim teknolojisi kullanarak IoT tabanlı akıllı bir sistem geliştirmişlerdir. Sistem içerisinde uzaktan izleme, acil durumlar için anında müdahale kısımları mevcuttur. Hastanın bulunduğu ortama göre (hastane veya ev) sistem davranışları değişmektedir [7].

Jara ve diğ., kullanılan ilaçların ciddi yan etkilerini önlemek amacıyla, ciddi yan etkileri tespit eden IoT tabanlı bir sistem tasarlamışlardır. Hastaların ve ilaçların sisteme tanıtılması için RFID-NFC ve barkod teknolojileri kullanılmıştır. Sistem akıllı eczane bilgi sistemi ile senkron halde çalışabilmektedir. Alerji profili de çıkaran sistem, gerçek zamanlı olarak bir hasta üzerinde denenmiş ve olası bir problem için hastanın sistem tarafından uyarıldığı görülmüştür. Sistem ile ilgili mobil cihaz üzerinde çalışabilecek iki uygulama geliştirilmiştir [21].

Rohokale ve diğ., şehir merkezlerinden uzakta yaşayan insanların fizyolojik verilerinin sürekli bir şekilde takip edilmesi amacıyla IoT tabanlı bir sistem önermişlerdir. Ns-2 programında benzetimi yapılan sistemde gecikme ve enerji harcamaları gibi değerler incelenmiştir [8].

Zhang ve Hu, sistemimizde de kullanılan mevcut Open-Zigbee modelini biyomedikal uygulamalarına uygun olacak şekilde modifiye etmişlerdir. Benzetim senaryosuna göre, ağdaki cihazlardan bazı fizyolojik veriler toplanmış (örn. EKG verileri) ve toplanan bu veriler koordinatör düğüm üzerinden veya aracı olmadan veri izleme merkezine iletilmiştir. Geliştirilen sistem, gelen verileri analiz ederek sistemi kullanan

doktorlara gelen verilere göre teşhis ve tanı koyma konusunda yardımcı olabilmektedir [15].

Sevin ve diğ., kişisel sağlık bilgisinin standardını tanımlayan, ISO&IEEE gibi dünyanın en önemli iki kurumunun desteklediği ISO/IEEE 11073 standardını esas alan bir Ortam Erişim Kontrol (OEK) protokolü geliştirmiştir. Bu çalışmada IEEE 802.15.6 süper çerçeve mekanizmasına benzer bir çerçeve yapısı önerilmiştir [18].

Gündoğdu ve Çalhan, veri trafiğine duyarlı olmayan bir KVAA'nın, önleyici olmayan (non-preemptive) öncelikli kuyruk disiplinini kullanarak hem benzetim modelini hem de uygulanmasını gerçekleştirmiştir. Zaman Bölümlü Çoklu Erişim (Time Division Multiple Access - TDMA) kullanan KVAA uygulaması üç farklı veri trafiği önceliği ile tasarlanmıştır. Ayrıca, bu çalışmada önleyici olmayan öncelik sırasına sahip bir koordinatör düğümü tasarlanmıştır. Uygulamayı doğrulamak amacıyla Riverbed Modeler benzetim yazılımı kullanılarak örnek ağ senaryoları modellenmiş ve sistemin benzetimi gerçekleştirilmiştir. Ağır paket yoğunluğu koşulları altında elde edilen benzetim sonuçları ile uygulama sonuçlarının tutarlı olduğu görülmüştür [14].

Celentano, çalışmasında, içinde RFID ve Kablosuz Algılayıcı Ağ (KAA) barındıran bir ilaç takip sistemi gerçekleştirmiştir. KAA kısmında, EKG verilerinin çok düşük veri kaybı ile kablosuz olarak elde edilebilmesi için kalman filtresi kullanarak bir benzetim çalışması yapılmıştır. Sisteme entegre edilmiş RFID teknolojisi ile de ilaç kutularına kimlik kazandırılmıştır. İlaç takibi gerçekleştiren sistem ilaç alınmaması ya da yanlış bir ilaç alımı durumunda hem hastayı hem de sistemde kayıtlı çalışanları mesaj ile bilgilendirmektedir [22].

Çetin, çalışmasında, KAA'ların medikal alanlar için bir uygulamasını gerçekleştirmiştir. Algılayıcı düğümler ile ölçülen nabız, plestismogram ve SpO₂ değerleri Zigbee haberleşme protokolü kullanılarak KAA ile merkezi veri tabanına yazılmıştır [23].

Atzori ve diğ., IoT teknolojisiyle ilgili geniş çaplı bir derleme çalışması yapmışlardır. Yapılan çalışmada, IoT yapısının bileşenleri, güncel hayattaki kullanım alanları ve içerisinde medikal sağlık hizmetlerinde bulunduğu birçok uygulama alanı ile ilgili bilgilere geniş olarak yer verilmiştir [2].

Lee ve diğ., hastane dışındaki yaşlı hastaların evde bakımı için teşhis ve alarm sistemi geliştirmişlerdir. Geliştirilen sistem ile gerçek zamanlı olarak EKG sinyalleri toplanmış ve bu sinyaller bir veri tabanına aktarılarak PQRS parametrelerindeki QRS parametrelerinin genişliği ve R-R parametreleri arasındaki mesafe gibi bilgiler incelenerek olası anormallikler analiz edilmiştir. Geliştirilen yazılım ile anormal durumlar sınıflandırılarak, doktorlara uyarı şeklinde bildirilmektedir [11].

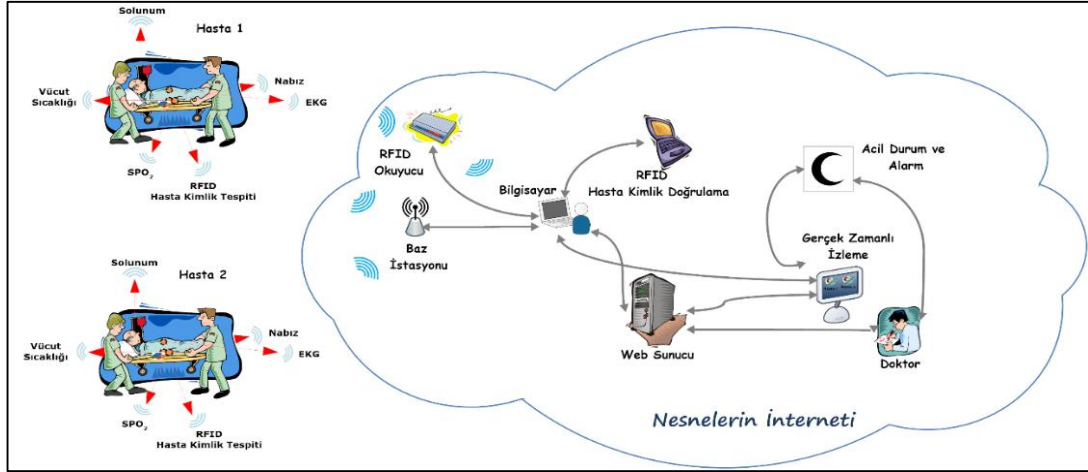
Hu ve diğ., insan vücudundaki birçok fizyolojik veriyi toplamak, toplanan bu verileri izlemek ve bu fizyolojik verilerin toplanmasından kaynaklanan ağ yığılmalarını önleme amaçlı bir sistem geliştirmişlerdir. Gerçekleştirilen sistem ile yaşlı geçmiş hastaların EKG, elektroensefalogram (EEG) ve elektromiyogram (EMG) sinyallerinin toplanıp, iletilmesine yönelik bir KVAA yapısı kurulmuştur. Gerçek zamanlı olarak toplanan fizyolojik veriler sistem yazılımı ile analiz edilmiştir. Analiz ile anormal durumlar tespit edilerek bu durumlar sınıflandırılmıştır. Ayrıca sisteme kişilerin konum ve kimlik bilgilerini tespit etmek amacıyla RFID teknolojisi entegre edilmiştir [12].

Yang ve diğ., gerçek zamanlı biyomedikal izlemeler yapmak amacıyla RFID uyumlu UHF bandında çalışan bir kablosuz algılayıcı düğüm tasarlamışlardır. Tasarımda, yüzey dielektrik katsayısı ve tanjant kaybı bilinen esnek, tekstil ve organik bir yüzeye gümüş mürekkep kullanılarak yazılmış bir RFID anten geliştirilmiştir [24].

Rahim ve diğ., bir KVAA test düzeneği tasarlamışlardır. Tasarlanan sistemdeki algılayıcı düğüm düşük güç tüketimli, giyilebilir, esnek ve küçük boyutludur. Sistemin performans değerlendirmesi için veri alım oranı ve uçtan uca gecikme (UUG) değerleri incelenmiştir. Sistemdeki yazılım ile değişik mesafelerden toplanan EKG verilerinin doğruluğu karşılaştırılmıştır [13].

1.2. Önerilen Sistemin Bileşenleri ve Tez Çalışmasının Katkıları

Doktora tez çalışmasında yeni bir veri toplama ve gerçek zamanlı hasta izleme platformu önerilmiştir. Sistemin belirli koşullar altında ve belirlenen senaryoya göre benzetimi yapılmıştır. Ayrıca tez çalışmasında belirlenen senaryoya uygun olarak gerçek zamanlı hasta izleme sistemi geliştirilmiştir (Şekil 1.1).



Şekil 1.1. IoT tabanlı veri toplama ve analiz platformunun fiziksel bileşenleri

Gerçekleştirilen IoT tabanlı bu sistemin benzetim kısmında, KVAA, RFID sistemi, hücresel ağ, veritabanı ve bir web sunucusu olmak üzere birkaç modül bulunmaktadır (Şekil 1.1). RFID ve GSM arayüzleri mevcut IEEE 802.15.4 tabanlı KKAA (Kablosuz Kişisel Alan Ağı) sistemine (open-Zigbee) entegre edilmiştir. Gerçek zamanlı izleme sisteminde ise KVAA, RFID, veri tabanı, sunucu ve web tabanlı uygulama bulunmaktadır. Benzetim ve gerçek zamanlı sistem tasarımıyla beraber sağlık hizmetleriyle ilgili bazı işlemleri basitleştirmek için IoT tabanlı bir altyapı oluşturulmuştur.

Doktora tezinin katkıları şu şekilde sıralanabilir;

- Benzetim çalışmasında içeriğinde RFID-GSM, gerçek zamanlı sistemde RFID KVAA arayüzü bulunan yeni bir Kişisel Alan Ağı (Personal Area Network - PAN) koordinatörü (PANc) geliştirilmiştir.
- Çok yönlü hastane bilgi sistemleri için bir altyapı oluşturmak üzere, çeşitli bileşenlere ve araçlara sahip yeni bir benzetim sistemi ve gerçek zamanlı izleme sistemi tasarlanmıştır. Böylece sağlık bilgi sistemleri ile ilgili işlemler zorlanmadan gerçekleştirilebilecektir.
- Benzetim çalışmasında mevcut IEEE 802.15.4 tabanlı KKAA sistemine RFID ve GSM arayüzü entegre edilmiştir. Böylece sağlık hizmetleriyle ilgili bazı işlemleri basitleştirmek için IoT tabanlı bir altyapı oluşturulmuştur. Bu kapsamda, KKAA ve GSM arayüzü ile yeni bir PAN koordinatör (PANc) cihazı geliştirilmiştir.
- IEEE 802.15.4 standardı ortam erişim kontrol (Medium Access Control - MAC)

katmanında bulunan süper çerçeve yapısı etkin biçimde kullanılarak, servis kalitesi (QoS) desteğiyle fizyolojik sinyallerin iletimi sağlanmıştır.

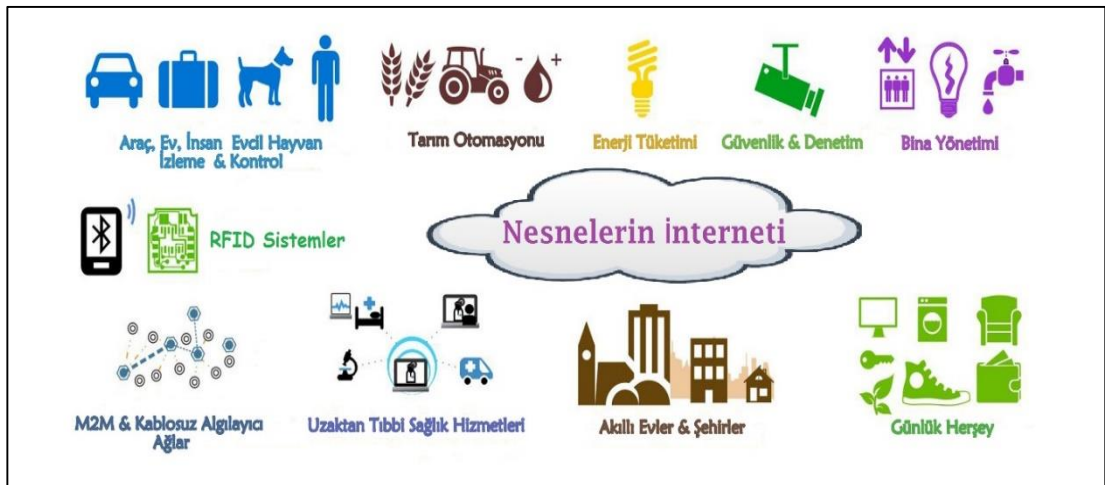
- Tasarlanan benzetim sistemi, daha gerçekçi performans sonuçları için Riverbed Modeler yazılımı kullanılarak modellenmiş ve simüle edilmiştir.
- Gerçek zamanlı izleme sisteminde KVAA erişim noktası ve RFID okuyucu birleştirilerek hibrit bir yapı oluşturulmuştur. Hastalardan gelen fizyolojik veriler servis kalitesi desteğiyle bilgisayar ortamına alınarak Node.js platformuyla etkin bir şekilde okunmuştur. Gerçek zamanlı izleme sistemi ile fizyolojik verileri ve kişisel bilgileri eşleştirilen hastaların takibi etkin bir şekilde yapılabilmektedir.



2. NESNELERİN İNTERNETİ BİLEŞENLERİ

Tez çalışmasının bu bölümünde IoT teknolojisi ile onu oluşturan bileşenler hakkında bilgi verilmektedir. Tez çalışması kapsamında kullanılan IoT bileşenlerinden olan KVAA ve RFID teknolojileri ile ilgili genel bilgiler verildikten sonra özellikle benzetim çalışmasında üzerinde çalışılan IEEE 802.15.4 standardı ve RFID uygulama katmanı ve komut yapıları hakkında detaylı bilgiler verilmektedir.

IoT’de nesnelere, fiziksel dünyanın (fiziksel nesnelere) veya bilgi dünyasının (sanal dünya) bir iletişim ağına entegre edilebilen nesnelere dir. Fiziksel nesnelere, dünya üzerinde mevcuttur ve algılanabilir, çalıştırılabilir ve bağlanabilir durumdadır (robotlar, elektronik eşya ve cihazlar vb.). Sanal nesnelere ise, bilgi dünyasında mevcuttur ve depolanabilir, işlenebilir ve erişilebilir durumdadır. IoT’de bir aygıt, iletişim yetenekleri olan ve isteğe bağlı algılama, tanılama, veri toplama, veri saklama ve veri işleme yeteneklerine sahip olan bir cihazdır. Aygıtlar, çeşitli fiziksel verileri ortamdan toplayabilmekte ve daha sonra bu verileri bilgi ve iletişim ağlarına aktarabilmektedir. Bazı aygıtlar, bilgi ve iletişim ağlarından alınan verilere dayalı olarak işlemleri de yürütebilmektedir [26, 28]. Şekil 2.1’de IoT’nin şematik gösterimi ve uygulama alanları görülmektedir [30].



Şekil 2.1. IoT uygulama alanları [30]

IoT teknolojisi Şekil 2.1’de görüldüğü gibi tarımdan enerjiye, akıllı bina yönetimlerinden uzaktan sağlık izleme hizmetlerine kadar birçok alanda kullanılmaktadır. Sağlık hizmetleri uygulamalarına örnek olarak; evde veya hastanede fizyolojik verilerinin takibi yapılan bir bebeğin ağlaması durumunda ya da ateşinin çıkması durumunda duyma engelli ailesine durumun algılanıp hem mobil cihaz üzerinden hem de SMS ile titreşim özelliği olan mobil cihaz uyumlu kol saatiyle uyarılması gösterilebilir [29, 31]. Bununla beraber IoT bileşenleri; RFID sistemleri, KAA’lar, adresleme şemaları, veri depolama ve analiz sistemleri ile görüntüleme sistemleri olarak sayılabilmektedir [32].

Bu bölümde sırasıyla tez çalışmasında kullanılan IoT bileşenleri olan KVAA ve RFID sistemleri hakkında bilgi verilmektedir. Bu bölümlerin içlerinde ise özellikle benzetim çalışmasında üzerinde çalışılan, KVAA haberleşme yapısını oluşturan Zigbee ağ protokolündeki IEEE 802.15.4 standardı ile RFID sisteminin uygulama katmanı kodları ile ilgili detaylı bilgi verilmektedir.

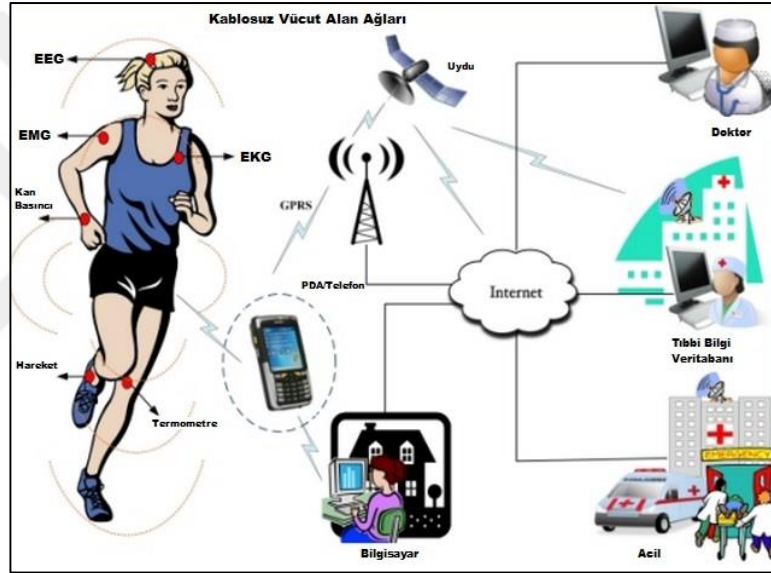
2.1. Kablosuz Vücut Alan Ağları (KVAA) ve IEEE 802.15.4 Standardı

Kablosuz Vücut Alan Ağları (KVAA), kişilere ait fizyolojik sinyalleri algılayabilen, birbirleriyle ve erişim noktasıyla kablosuz haberleşebilen, veri işleyebilen minimal boyutlardaki akıllı cihazlardan meydana gelmektedir. Sensör düğümler, insan vücudunun içerisine yerleştirilebildiği gibi elbise şeklinde giyilebilir veya üstte taşınabilir olarak kullanılabilir [1].

Şekil 2.2’de temel KVAA yapısı ve sistemin genel çalışması görülmektedir [33]. Şekil 2.2’de görülen KVAA yapısı iki kısım halinde düşünülebilir. Yapının ilk kısmında, kişi üzerinden fizyolojik parametreleri (EKG, EMG, EEG, SpO₂, kan basıncı vb.) toplayan kablosuz sensörler ve sensörlerin üst katmanında bulunan, ağ bağlantısı sayesinde haberleşmeyi sağlayan bir kablosuz ağ geçidi (PDA, akıllı cep telefonu vb.) bulunmaktadır. Bu kısımda kullanılan cihazlar genel olarak, vücut içerisine yerleştirilen sensörler, yutulabilir kameralar, yapay retina, giyilebilir sensörler, kan basıncı, ivmeölçer ve sıcaklık ölçerler gelmektedir. İkinci kısımda, kablosuz ağ geçidi üzerinden gelen verileri kablosuz olarak toplayan bir koordinatör düğüm ile bu koordinatör düğüme bağlı genellikle internet erişimi olan bir kişisel bilgisayar yer almaktadır. Bu bilgisayarda, gelen verilerin işlenmesi ve analiz

işlemlerini yapan bir takip uygulaması çalışır ve tüm veriler sisteme kayıtlı kişiler tarafından uzaktan ulaşılabilecek şekilde kayıt altında tutulur [1].

Bu alanda kullanılan KVAA'ların temel amacı, kişiler açısından acil durumlar meydana gelmeden önce önleyici tedbir alınmasını mümkün hale getirmektir. KVAA'lar hayati verilerin belirli bir seviyenin altına inmesi veya üzerine çıkması halinde hem kişinin yakınlarına hem de en yakın medikal merkezine uyarıda bulunabilirler. Bu durumda tansiyonu yükselen, kalp krizi geçiren, ateşi yükselen veya düşen bir kişinin durumu ve bulunduğu yer ile ilgili net bilgiler elde edilerek gereken tıbbi müdahale gerçekleştirilebilmektedir [1].

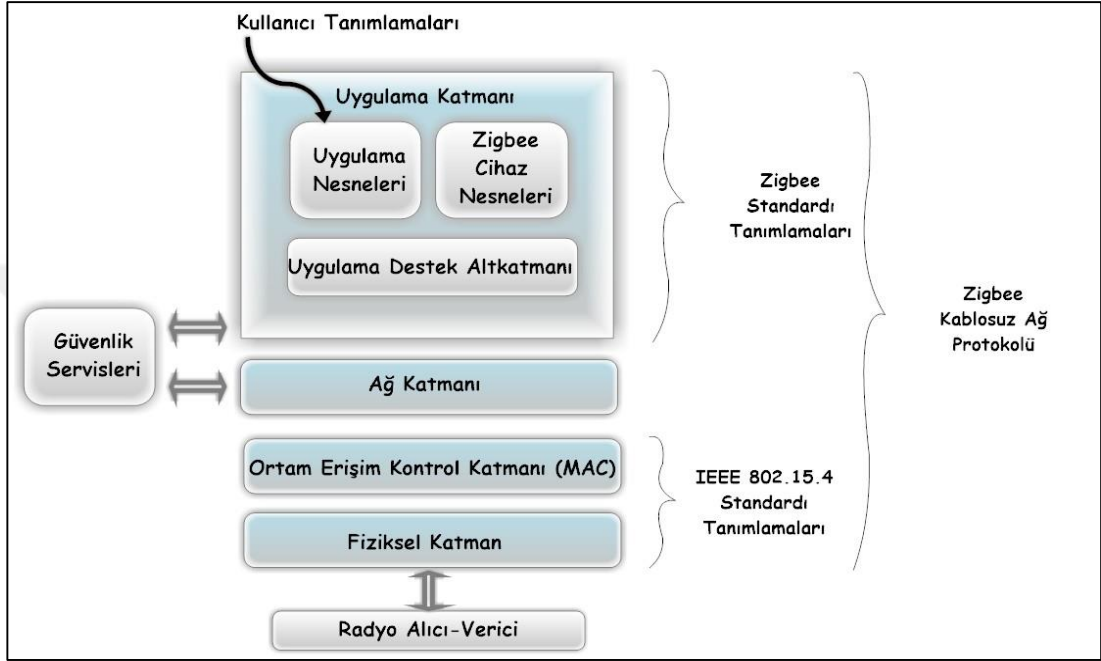


Şekil 2.2. Temel KVAA yapısı ve sistemin çalışması

2.1.1. IEEE 802.15.4 standardı

Doktora tez çalışmasının benzetim kısmında önerilen KVAA sistemi, Tablo 1.1'de belirtilen servis kalitesi (Quality of Service - QoS) gereklilikleri göz önüne alınarak herhangi bir fizyolojik sinyali iletebilmektedir. Böylece algılayıcı düğüm tarafından alınan verilerin erişim noktasına (PANc) yeterli veri oranında ve gecikme süresi içerisinde iletilmesi garanti edilmektedir. Bu kısımda, ilk olarak KVAA sisteminde kullanılan Zigbee kablosuz ağ protokolü katmanlarının IEEE 802.15.4 standardına göre nasıl çalıştığı, daha sonra IEEE 802.15.4 standardının alt bileşenlerinde tez çalışmasına yönelik olarak benzetim çalışmasında yapılan yeni modifikasyonların nasıl yapıldığı açıklanmaktadır.

KVAA sistemlerinde en çok kullanılan iletişim protokollerinden biri olan Zigbee kablosuz ağ protokolünün katmanları, Açık Sistem Bağlantı (Open System Interconnect - OSI) referans modeline dayanmaktadır. OSI referans modeline göre Zigbee kablosuz ağ protokolü katmanlarının görünümü Şekil 2.3’de görülmektedir [34].



Şekil 2.3. Zigbee kablosuz ağ protokolü katmanları [34]

Şekil 2.3’de görüldüğü gibi Zigbee kablosuz ağ protokolü, Zigbee standartlarını ve IEEE 802.15.4 standart bileşenlerini içermektedir. Zigbee teknolojisi ile radyo frekansı (RF), düşük veri hızı, daha uzun pil ömrü ve ağ güvenliği gerektiren uygulamalar hedeflenmiştir. Zigbee protokolü endüstriyel, bilimsel ve tıbbi radyo (Industrial Scientific Medical - ISM) bantlarında, Avrupa’da 868 MHz, ABD ve Avustralya’da 915 MHz ve dünya çapında 2,4 GHz frekanslarında, 20 kbps ile 250 kbps arasındaki veri iletim hızlarında çalışabilmektedir. Zigbee ağ katmanı olarak yıldız, ağaç ve salkım (mesh) ağ topolojilerini desteklemektedir. Zigbee, adını bal arılarının arı kovanı etrafındaki sallanarak dönüş hareketinden almıştır [35].

IEEE 802.15.4, IEEE 802 standartlar komitesi tarafından Zigbee standardından bağımsız olarak geliştirilmiştir ve ilk olarak 2003 yılında piyasaya sürülmüştür. Bu standart, Kablosuz Kişisel Alan Ağları (Wireless Personal Area Network - WPAN) için düşük veri oranları, düşük güç tüketimi ve düşük maliyet avantajları ile

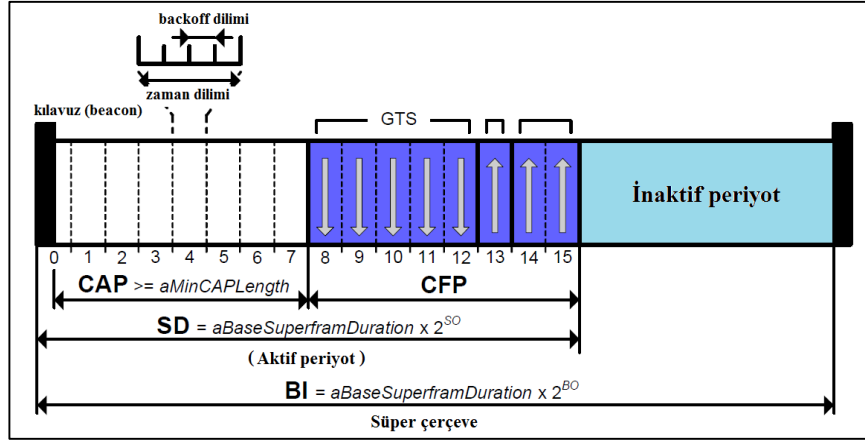
geliştirilmiştir [35]. Şekil 2.3'den de görüldüğü gibi en altta bulunan ortam erişim kontrol (MAC) katmanı ve fiziksel katman, IEEE 802.15.4 standardı tarafından tanımlanmıştır. IEEE 802.15.4 standardı daha üstteki ağ katmanları için özellik ve gereksinimleri tanımlamamaktadır.

Zigbee standardı ana protokolün sadece ağ, uygulama ve güvenlik katmanları ile ilgili özellikleri tanımlar. Fiziksel ve MAC katmanlarının özelliklerinin tanımlandığı IEEE 802.15.4 standardı ile birlikte Zigbee ağ protokolünü tamamlarlar. Böylece kullanıcılar IEEE 802.15.4 standardının tanımladığı fiziksel ve ortam erişim katmanlarının üzerinde kendi ağ (networking) ve uygulama katmanlarını geliştirebilmektedir.

IEEE 802.15.4 MAC katmanı, kılavuzlu (Beacon-enabled mode) ve kılavuzsuz (Non Beacon-enabled mode) olmak üzere iki moda çalışabilmektedir. Bu modlar PAN içerisinde bulunan erişim noktası (PANc) tarafından seçilebilmektedir.

Kılavuzsuz moda cihazlar, bölümsüz Taşıyıcı Algılayıcı Çoklu Erişim/Çarpışmadan Kaçınmalı (Carrier Sense Multiple Access/Collision Avoidance - CSMA/CA) protokolü ile veriyi ortama gönderirler. Kılavuzsuz moda süper çerçeve (superframe) yapısı kullanılmamaktadır. Bu modun avantajı ölçeklenebilirlik ve ağın kendi kendine organize olabilmesidir. Kılavuzsuz moda verilerin hedefe iletilmesi için zaman garantisi verilmemektedir [35].

Kılavuzlu moda kılavuz çerçevesi, PAN koordinatörü tarafından süper çerçevede tanımlanmış şekilde cihazlar arasındaki eşzamanlamayı sağlamak için üretilir. Erişim noktası (koordinatör) kılavuzlu modu seçtiğinde, kişisel alan ağında (PAN) birbiri ile bağlantılı aygıtların haberleşmelerini yönetmek amacıyla süper çerçeve yapısını kullanır. Süper çerçevenin formatı erişim noktası tarafından tanımlanmaktadır. Şekil 2.4'te süper çerçeve formatı görülmektedir [36].



Şekil 2.4. Süper çerçeve formatı (superframe) [36]

Süper çerçeve içerisinde tanımlanan kılavuz zamanı (Beacon Interval - BI), ardı sıra gelen iki kılavuz (beacon) arasında geçen süreyi ifade etmektedir. BI iki kısımdan oluşmaktadır. Bunlar; aktif ve pasif periyotlardır (opsiyonel). Aktif periyotta (Superframe Duration - SD) süre 16 eşit zaman dilimine bölünür ve veriler bu süre içerisinde iletilir. Her aktif periyot, çekişme erişim periyodu (Contention Access Period - CAP) ve opsiyonel olarak belirlenen çekişme serbest periyodu (Contention Free Period - CFP) kısımlarına bölünebilmektedir. Dilimli CSMA/CA yapısında, CAP kısmı kullanılarak veriler iletilmektedir. Son cihaz (End Device) olarak isimlendirilen düğümlerden erişim noktasına istek geldiğinde CFP kısmı aktif edilir. Erişim noktası (PAN_coordinator) istekte bulunan düğümlere kritik zamanlı verilerin hedefe iletimi için gerçek zamanlı garanti verir. Son cihazdan istek geldiği zaman erişim noktası eğer mümkünse ve yeterli kaynak var ise istenilen sayıda zaman dilimini cihaza tahsis etmektedir. İstenilen zaman dilimi grubu, garanti edilmiş zaman dilimi (Guaranteed Time Slot - GTS) olarak adlandırılır ve yalnızca isteyen cihaza verilir. GTS modunda, ağ içerisindeki cihazların ortam içerisinde çekişmesiz olarak birbirleriyle haberleşmelerine izin verilmektedir. GTS, sadece erişim noktası tarafından tahsis edilebilir ve yalnız son cihazlar ile erişim noktası arasında veri transferi için kullanılır. Eğer yeterli kaynak yoksa GTS tahsisi erişim noktası tarafından engellenmektedir [34]. Veri transferi (transmit direction), son cihaz (End Device) ile erişim noktası arasında karşılıklı olarak gerçekleşmektedir. CAP, minimum ($aMinCAPLength$) 440 sembol uzunluğundadır ve GTS tahsisi CAP uzunluğu etkilemez (azaltamaz). Doktora tez çalışmasında önerilen sistemde, QoS gerekliliklerine göre optimum

fizyolojik veri iletimi için open-Zigbee simülasyon modeli kılavuzlu modda kullanılmaktadır. Bundan dolayı bu modla ilgili bilgiler daha detaylı incelenmektedir.

Kılavuzlu modda, erişim noktası tarafından tanımlanan kılavuz (beacon) paketleri, ağda kimliklendirilmiş cihazlara, cihazlar arasındaki eş zamanlamayı sağlamak amacıyla belli bir periyotta gönderilmektedir. Süper çerçeve içerisindeki Beacon Order (BO) ve Superframe Order (SO) parametreleri ile süper çerçevenin ve içerisindeki aktif periyodun uzunlukları belirlenmektedir. BO'nun değeri her zaman SO'ya eşit veya daha büyüktür. BO ve SO ayarı, kılavuzlu moddaki $0 \leq SO \leq BO < 15$ ilişkisini sağlamalıdır. Böylece SD değeri, BI değerini aşmamış olacaktır (Şekil 2.4). Süper çerçevenin uzunluğu (BI) ve aktif periyodun uzunluğu (SD), Denklem (2.1) ve Denklem (2.2)'ye göre hesaplanmaktadır;

$$BI = aBaseSuperframeDuration \times 2^{BO} \quad (2.1)$$

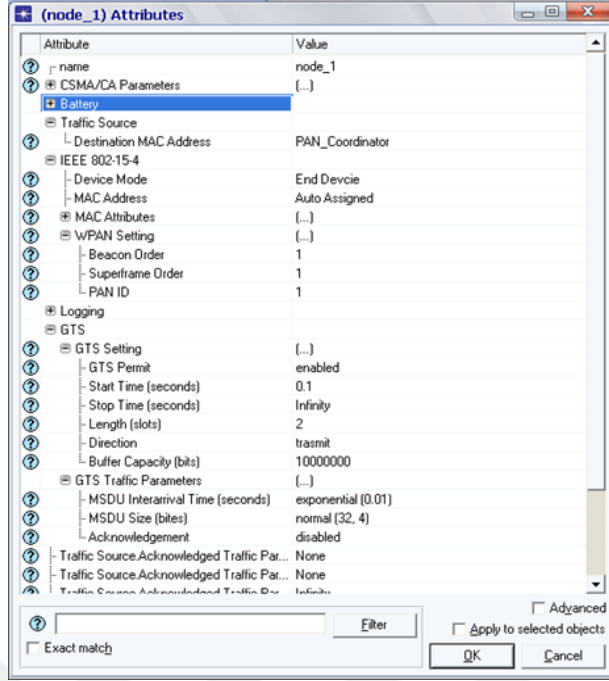
$$SD = aBaseSuperframeDuration \times 2^{SO} \quad (2.2)$$

BO'nun sıfır olduğu değer için, aBaseSuperframeDuration IEEE 802.15.4 standardına göre 960 sembol olarak sabit bir değeri ifade etmektedir. Bu durumda en düşük kılavuz süresi 960 sembol olmaktadır. 2,4 GHz iletişim için bir sembol hızı saniyede 62,5 Kiloseboldür ve her sembol 4 bit uzunluğundadır. Bu değerler için en düşük kılavuz süresi (BI) 15,36 ms olarak hesaplanmaktadır. Bu süre içerisinde (15,36 ms), $960 \times 4 = 3840$ adet bit taşınabilmektedir [34].

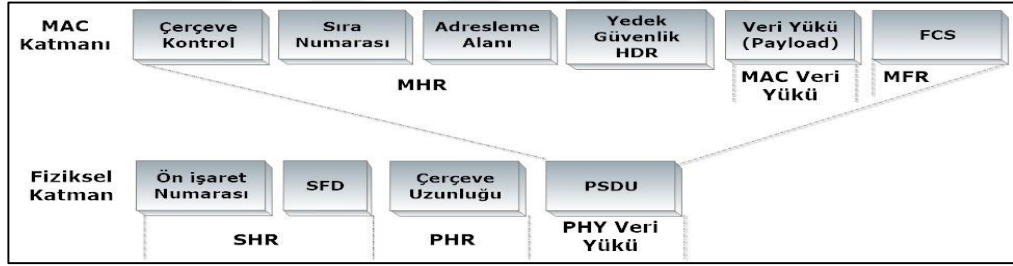
SO = BO eşitliği durumunda Denklem (2.1)'de görüldüğü gibi BI = SD olmaktadır ve süper çerçeve sürekli aktif modda çalışmaktadır. SO < BO olduğu durumlarda süper çerçevenin bir kısmı aktif, bir kısmı pasif periyotta çalışmaktadır. BO = 15 olduğunda SO değeri dikkate alınmamaktadır, süper çerçeve pasif hale gelmektedir ve ağ kılavuzsuz modda (non beacon-enabled mode) çalışmaktadır. Sonuç olarak, eğer bir kişisel alan ağı iletişim için süper çerçeve yapısını kullanacaksa BO ve SO değerlerinin 0 ile 14 arasında olması gerekmektedir [35]. Özellikle, iletimi zamansal olarak kritik olan EKG verileri aşağıdaki kısımlarda anlatılacak olan benzetim senaryosunda transfer edilirken, BO ve SO değerleri, optimize edilmiş 1 değerine ayarlanmıştır. Bu değerler seçildiğinde veri trafiğinin uçtan uca gecikme koşulu garanti edilebilmektedir.

Kılavuzlu modda, ağdaki tüm cihazlar düzenli bir şekilde uyanmakta, kılavuzu beklemekte, saat sinyali ile senkronize olmakta ve tekrar uyumaktadır. Bu durum ağdaki bir çok cihazın sadece senkronizasyon için uyanmaları ve aktif oldukları süre içinde başka hiçbir iş yapmamaları anlamına gelmektedir. Kılavuz kullanmayan ağlarda ise cihazlar bu kadar sık uyanmazlar. Bu yüzden kılavuz kullanan ağlardaki cihazların pil ömrü kılavuz kullanmayan ağlardaki cihazların pil ömrüne göre fark edilebilir şekilde daha az olmaktadır. Bu durum kılavuzlu modun dezavantajlarından biri olarak gösterilebilir [34].

Şekil 2.5’de ağ içerisinde fizyolojik verileri okuyarak erişim noktasına gönderen düğümlerin özelliklerinin ayarlandığı ekran görülmektedir. Şekil 2.5’de görüldüğü gibi, tez çalışmasına uygun olarak üzerinde geliştirmeler yapılan sistemde, cihazdan gelen istek üzerine erişim noktasının (PANc) GTS tahsisi yapıp yapmayacağına kullanıcı tarafından belirlenen ‘GTS Permit’ ayarı ile karar verilmektedir. Ayar için ‘Enable’ seçilmesi durumunda ilgili cihaza GTS tahsisi yapılmakta, ‘Disable’ seçilmesi durumunda ise GTS tahsisi yapılmamaktadır. GTS tahsis isteği yapıldıysa, paketlerin üretilme zaman aralıkları ‘Start Time’ ve ‘Stop Time’ özellikleri ile ayarlanabilmektedir. Kaç adet GTS dilimi istendiği ise ‘GTS Length’ özelliği ile ayarlanmaktadır. ‘MSDU Size’ özelliği ile asıl bilginin olduğu yararlı yük (payload) boyutu ayarlanmaktadır. ‘MSDU Interarrival Time’ birbirini takip eden iki yararlı yük arasındaki zamandır ve paket üretme sıklığını ifade etmektedir. Yararlı yük çerçevesi MAC başlığı (MAC header) ile birlikte paketlenerek geçici bellekte saklanmaktadır. Geçici belleğin boyutu ‘Buffer Capacity’ özelliği ile belirlenmektedir. MAC başlığın varsayılan değeri (MAC_HEADER_SIZE) (16 bit kısa adresleme ile) 104 bittir. Bütün paketin boyutu (veri yükü + MAC başlığı) maksimum 1016 bit olabilmektedir (aMaxPHYPacketSize) [35]. Şekil 2.6’da MAC katmanındaki veri paketi yapısı görülmektedir [30, 34].



Şekil 2.5. N1 düğümüne ait özellik ayar ekranı



Şekil 2.6. MAC katmanı veri paketi yapısı

Kılavuzlu moda koordinatör-düğüm arası veri iletişimi sırası Şekil 2.7'de görülmektedir [30, 34].



Şekil 2.7. Kılavuzlu moda koordinatör-düğüm arası veri iletişimi

Çalışılan modelde cihazlar arası iletişim için yıldız (star) ağ topolojisi kullanılmaktadır. Yıldız ağ topolojisinin en önemli avantajı basit yapıda olmasıdır.

Yıldız ağ topolojisinde merkezi haberleşme sistemi bulunmaktadır ve cihazlar merkezi kontrolör olan erişim noktası (koordinatör) üzerinden haberleşmektedir. Tez çalışmasındaki benzetim senaryosunda iki adet son cihaz (End Devices) bir tane de merkezi denetleyici olan erişim noktası (PANc) bulunmaktadır. Her düğümün kendine ait eşsiz birer adresi bulunmaktadır [35].

Tez çalışmasının benzetim modelinde bulunan IEEE 802.15.4 uyumlu algılayıcı düğümler (wpan_sensor_node) üç fonksiyonel bloktan oluşmaktadır. Bunlar;

- Fiziksel Katman: IEEE 802.15.4 özellikleri ile uyumlu kablosuz radyo sinyali gönderici (radio transmitter - Tx) ile alıcı (receiver - Rx) içermektedir. 2,4 GHz frekans bandında ve 250 kbps veri hızında çalışmaktadırlar. Veri iletim gücü 1 mW olarak ayarlanmıştır. Modülasyon tekniği olarak dörtlü faz kaydırmalı anahtarlama (Quadrature Phase Shift Keying - QPSK) kullanılmaktadır.
- Ortam Erişim Kontrol Katmanı (MAC): Ortam erişim kontrol katmanında, dilimli (slotted) CSMA/CA ile GTS mekanizmaları uygulanabilmektedir. GTS modu aktif olduğunda, veri trafiği (zaman kritik trafikler), uygulama katmanında kapasitesi belli bir geçici bellek (buffer) üzerinde depolanarak ağ katmanına gönderilir. Zaman kritik olmayan veri paketleri ise sonsuz geçici bellek üzerinde depolanarak dilimli CSMA/CA algoritması ile aktif CAP boyunca ağ katmanına gönderilir.
- Uygulama Katmanı: İki tane trafik üretici (Traffic Source ve GTS Traffic Source) ve bir tane de trafik sonlandırıcı kısım (Traffic Sink) bulunmaktadır. Trafik üretici (Traffic source) kısmı, onaylı (acknowledge) ve onaysız (unacknowledge) veri çerçevelerini üretir ve CAP süresi boyunca (dilimli CSMA/CA kullanılarak) iletir. GTS trafik üretici GTS mekanizmasını kullanarak onaylı ya da onaysız kritik zamanlı veri paketlerini üretebilmektedir. Trafik sonlandırıcı kısım ise aşağı katmandan gelen çerçeveleri alıp ağ performansı için istatistikleri tutmaktadır [35].

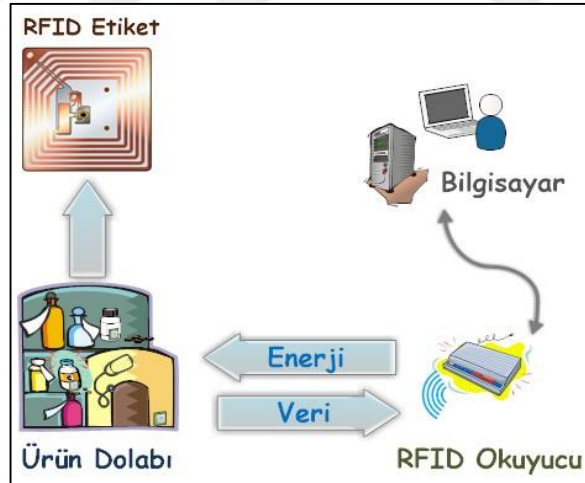
Çalışılan modelde (IEEE 802.15.4) desteklenen (gerçekleştirilen) özellikler [35] aşağıda listelenmiştir:

- Kılavuzlu mod (Beacon-enabled mode),
- Dilimli (Slotted) CSMA/CA MAC protokolü,
- Çeşitli çerçeve formatları (beacon, command, ack, mac_packet),
- Fiziksel katman karakteristikleri,

- GTS tahsis etme (allocation), tahsisi kaldırma (deallocation), ve yeniden tahsis etme (reallocation),
- Onaylı ve onaysız uygulama verisinin (MAC Frame Payload=MSDU) üretilmesi ve çekişmeli erişimi periyodu (CAP) süresi boyunca iletilmesi,
- Onaylı veya onaysız uygulama verisinin (MAC Frame Payload=MSDU) üretilmesi ve çekişmeli serbest periyodu (CFP) süresi boyunca iletilmesi.

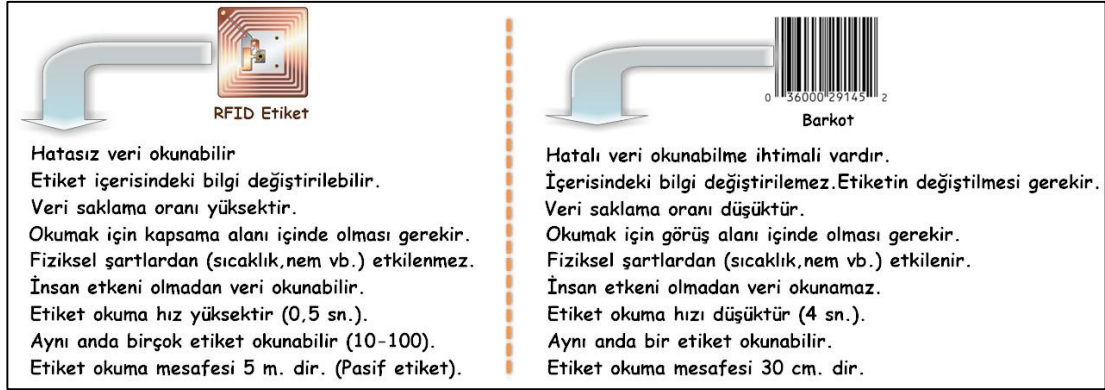
2.2. Radyo Frekanslı Tanılama (RFID)

IoT teknolojisinin en önemli bileşenlerinden biri olan RFID, radyo dalgaları kullanılarak taşınan verinin kontrol edilebilen çalışma ortamında bilgiye dönüşümüdür. RFID sistemlerinde tüm işlemler iletişim frekansına bağlı olarak değişen mesafede kablosuz olarak gerçekleşmektedir. RFID okuyucular, radyo dalgalarını farklı frekanslar üzerinden gönderirler. Frekans değeri arttıkça etiket içindeki mikroçipteki verilerin okunma uzaklığı da artmaktadır. Etiketden gelen veri dalgası okuyucu tarafından sayısal dalga biçimine dönüştürülerek bilgisayara aktarılmaktadır [37]. Şekil 2.8’de RFID genel çalışma prensibi görülmektedir [38].



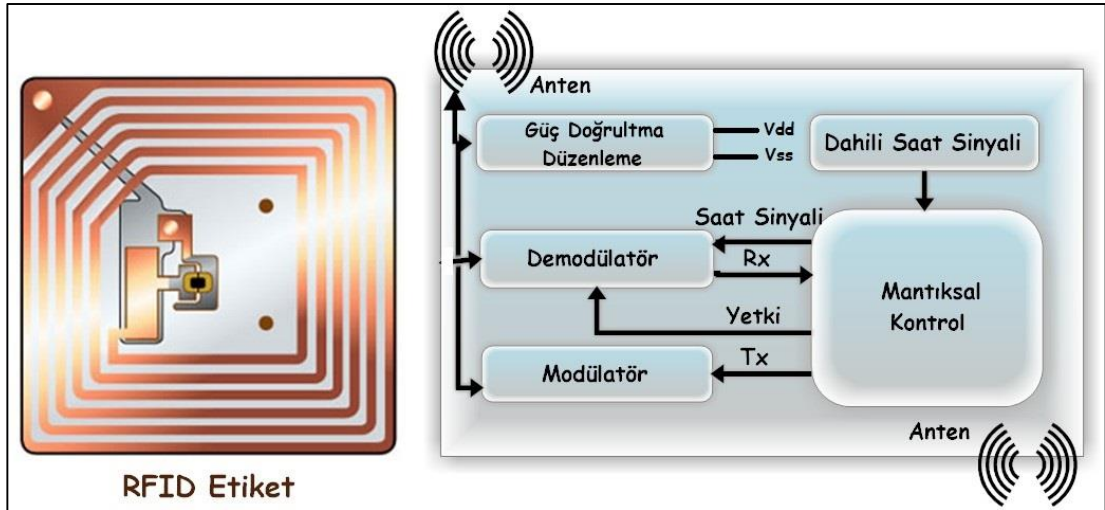
Şekil 2.8. RFID genel çalışma prensibi

Ticari ürünlerin sistemlere tanıtılmasında genel olarak RFID sistemler yerine barkod etiketler kullanılmaktadır. Barkod etiketlere göre nispeten daha pahalı bir yöntem olan RFID etiketlerin barkodlara göre birçok avantajı bulunmaktadır. Şekil 2.9’da barkodlar ile RFID etiketler arasındaki farklar liste halinde görülmektedir [38].



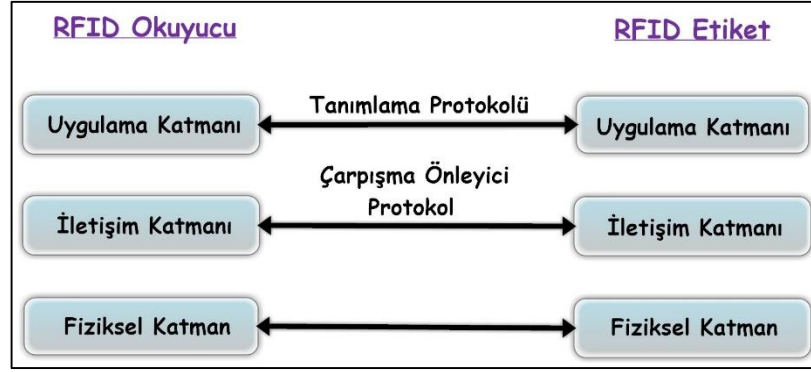
Şekil 2.9. RFID etiketleri ve barkod arasındaki farklar [38]

İçerdikleri bilgiyi koruyabilmek için dahili batarya kullanan etiketlere aktif etiket denilmektedir. Aktif etiketlerin okuma mesafesi içerdikleri bataryadan dolayı pasif etiketlere göre daha fazladır. Ancak batarya tükenme riski ve yüksek maliyetleri nedeniyle fazla tercih edilmezler. Pasif etiketler, bir RFID okuyucu tarafından enerjilendiklerinde sinyal yayar ve okuyucuya bilgi gönderirler. Yarı aktif etiketler ise mikroçipin devrelerini harekete geçirmek için dahili batarya kullanırken, iletişim kurmak için okuyucudan yayılan elektromanyetik dalgayı kullanırlar. Okuma işlemi yine okuyucunun yaydığı elektromanyetik alan yoluyla gerçekleşmektedir [39]. Şekil 2.10'da RFID etiketi ve iç yapısı görülmektedir [30, 40].



Şekil 2.10. RFID etiketi ve iç yapısı

Doktora tez çalışmasının benzetim kısmında RFID sistemin iletişim protokolü ile ilgili çalışmalar yapılmıştır. Şekil 2.11'de temel RFID iletişim protokolü ve katmanları görülmektedir [41].



Şekil 2.11. Temel RFID protokol modeli

Doktora tez çalışması kapsamının benzetimi kısmında RFID iletişim protokolünün uygulama katmanına odaklanılmıştır. Diğer kısımlar ile ilgili tanımlamalar geliştirilen sistemin benzetimi bölümünde anlatılmaktadır.

2.2.1. Uygulama katmanı

Uygulama katmanı etiket tanımlama katmanı olarak da adlandırılabilir. Bu katman, etiketlerin eşsiz olarak tanımlanmalarını yapmak için tanımlama protokollerini içermektedir. Okuyucular, etraflarında bulunan etiketleri organize etmek için üç temel işlem gerçekleştirmektedir. Bunlar seçim, sayım ve erişim işlemleridir.

2.2.1.1. Seçim

Bu işlem sadece bir seçme komutu ile gerçekleştirilmektedir. Okuyucu çevresindeki belirli bir etiket yoğunluğunu seçmek için gerçekleştirilen özel bir uygulamadır. Gerçekleştirilecek işlem sayım ve etiketlere erişimin öncelikli olduğu bir işlemdir. Ana bilgisayardaki uygun bir veri tabanından bir kaydı seçme işlemine benzerdir.

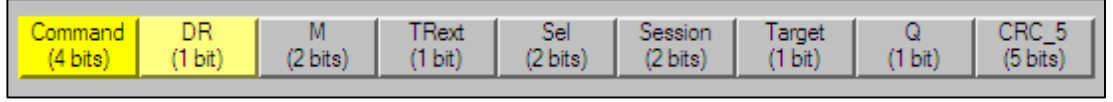
2.2.1.2. Sayım

Aşağıdaki çoklu komut seti kullanılarak etiketleri tanımlama ve sayma işlemleri yapılmaktadır. Komut seti şu şekildedir;

a. Query (Q)

Bu komut sayım döngüsünü başlatmak için kullanılır ve hangi etiketin döngüye katılacağına karar verir. Bu komut içerisinde, 'Q' olarak adlandırılan "dilim-sayıcı" parametresi etiketlere gönderilerek tüm etiketler aktif edilmektedir. Etiketler bu 'Q'

parametresine göre 0 ile 2^Q-1 arasında rastgele bir sayı üreterek kendi dilim sayıcı değerlerine sahip olurlar ve bu değeri saklarlar. Eğer ürettikleri sayı sıfıra eşit ise 'REPLY' durumuna geçerek okuyucuya cevap gönderebilirler. Ürettikleri değer sıfıra eşit değil ise diğer bir durum olan 'ARBITRATE' (karar) durumuna geçerler ve okuyucudan 'Query Adjust (QA)' veya 'Query Repeat (QR)' komutlarını beklerler [41]. Şekil 2.12'de Query komut formatı görülmektedir.



Şekil 2.12. Query komut formatı

Bu komutun toplam bit sayısı 22'dir. Komut içerisindeki alanların açıklamaları aşağıdaki gibidir;

- Komutta bulunan ilk alan olan 'Command' kısmı 4 bittir ve daima '1000_b' değerini alır.
- İkinci alan 'Bölme Oranı (Divide Ratio)' kısmıdır ve kısaca 'DR' olarak adlandırılır. Okuyucu ile etiket arasındaki hat frekansını ayarlamak için kullanılır. Tek bittir. 0: DR = 8, 1: DR = 64/3 değerlerini almaktadır.
- Sekizinci alan 'Q' değeridir. Dilim sayıcı değerinin nasıl ayarlanacağı ile ilgili bilgiyi ifade eder. 4 bitlidir. 0000_b ile 1111_b arasında (0 ile 15) değer alabilir.

b. Query Adjust (QA) komutu

Bu komut tekrar 'Query' komutudur. Bir önceki komutta gönderilen 'Q' değerini arttırabilmekte ya da azaltabilmektedir. Ancak sayım döngüsüne yeni bir etiketin katılmasına izin vermemektedir. Şekil 2.13'de 'Query Adjust (QA)' komut formatı görülmektedir.



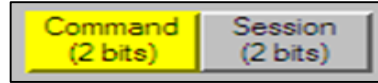
Şekil 2.13. QA komut formatı

Bu komutun üçüncü kısmı olan 'UpDn' bölümü 3 bitlidir ve 'Q' parametresinin bir önceki değerine göre arttırılması, azaltılması ya da aynı değerinde kalmasını

ayarlamaktadır. Örneğin; ‘UpDn’ değeri ‘110_b’ ise ‘Q’ parametresi 1 arttırılır, ‘000_b’ ise ‘Q’ değeri aynı kalır, ‘001_b’ ise 1 azaltılır şeklinde yorumlanabilmektedir.

c. Query Repeat (QR) komutu

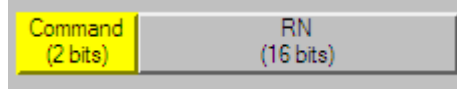
Bu komut bir önceki ‘Query’ komutunu parametrelerde herhangi bir değişiklik olmadan göndermek için kullanılır. Sayım döngüsüne yeni bir etiketin katılmasına izin verilmemektedir. Şekil 2.14’te ‘Query Repeat (QR)’ komut formatı görülmektedir.



Şekil 2.14. QR komut formatı

d. ACK komutu

Seçilen etiketin ürettiği 16 bitlik rastgele sayı (RN16) bu komut içerisinde okuyucuya gönderilir. Okuyucuda aynı RN16 değerini seçilen etikete geri gönderir. Şekil 2.15’de ACK komut formatı görülmektedir [8].



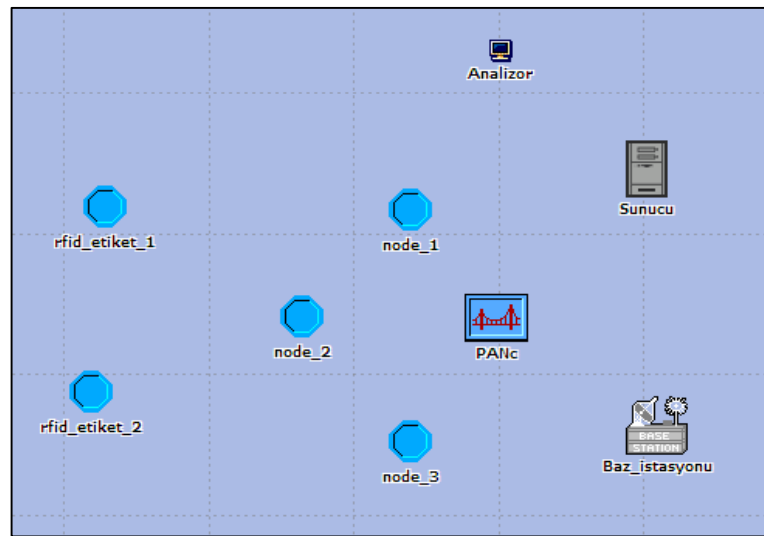
Şekil 2.15. ACK komut formatı

2.2.1.3. Erişim

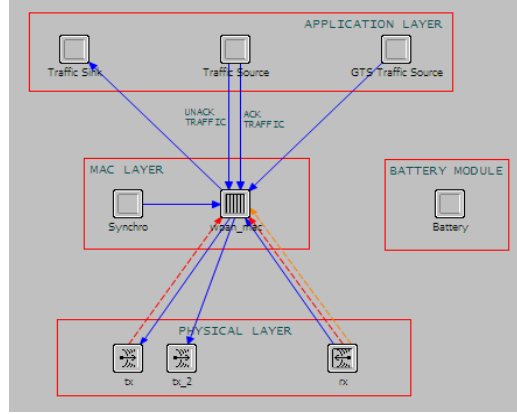
Okuyucunun etiketlerin her biriyle yaptığı işlemleri ifade etmektedir. Etiket in bu işlem den önce tanımlanması gereklidir. Okuma (read), yazma (write), iletişimi bitirme (kill) vb. komutları içermektedir.

3. GELİŞTİRİLEN SİSTEMİN BENZETİMİ

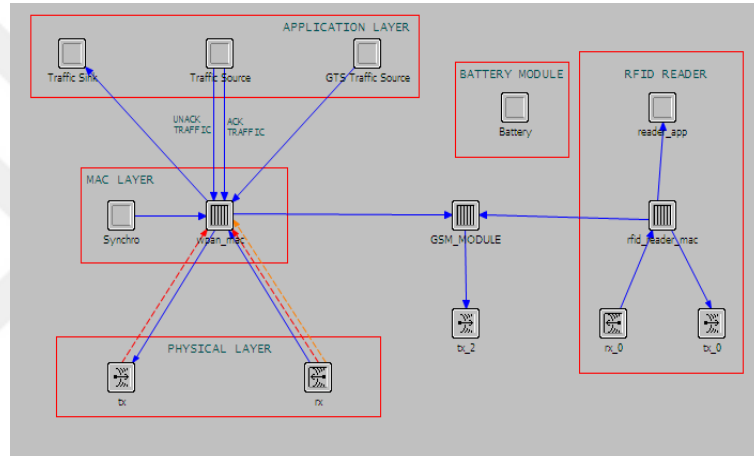
Doktora tezinde önerilen IoT tabanlı sistem, Riverbed Modeler yazılımı kullanılarak modellenmiştir ve benzetimi gerçekleştirilmiştir [42]. Geliştirilen sistem, bir KVAA sistemi (node_1 (N1), node_2 (N2), node_3 (N3) ve erişim noktası (PAN_koordinator - PANc)), bir hücresel ağ (baz istasyonu), RFID sistemi (rfid_etiket_1, rfid_etiket_2), veritabanı, sistemle ilgili istatistikleri tutan bir analiz cihazı (Analizör) ve web sunucusu bileşenlerinden oluşmaktadır. Geliştirilen sistemin Riverbed Modeler simulator çalışma alanı Şekil 3.1'de görülmektedir. Sistemde RFID okuyucu mevcut PANc'ye entegre edilmiştir. Tez çalışmasının benzetiminde, Tablo 1.1'de belirtilen QoS gerekliliklerine istinaden iki veri setinin eşzamanlı iletimi senaryosu gerçekleştirilmiştir. Bu veri setleri, (i) bir hasta için vücut ısısı ve EKG sinyali, (ii) başka bir hasta için sadece EKG sinyalidir. Her hasta için, hastaya ait fizyolojik veriler ile RFID ile elde edilen kişisel bilgiler (kimlik numarası, doğum tarihi vb.) eşleştirilmektedir. Mevcut open-Zigbee benzetim modeli, doktora tez çalışmasında belirlenen senaryoya göre modifiye edilmiştir. Bu değişiklikler ile RFID sistemi, baz istasyonu ve web sunucusu mevcut modele entegre edilmiş, ilgili sağlık bilgilerinin işlenmesinde esneklik sağlanmıştır.



Şekil 3.1. Geliştirilen nesnelere tabanlı sistemin Riverbed benzetim çalışma alanındaki görüntüsü



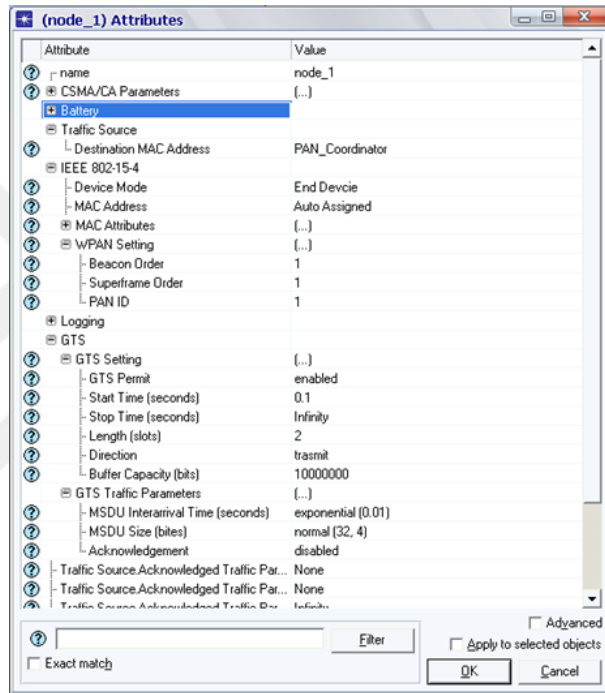
Şekil 3.2. N1, N2, and N3 düğümlerinin Riverbed düğüm modelleri



Şekil 3.3. RFID-GSM arayüzlerini içeren önerilen PANc düğüm modeli

Tez çalışmasının bu bölümünde Şekil 3.2’de görülen N1, N2 ve N3 düğümlerinin düğüm modelleri ve Şekil 3.3’de görülen PANc cihazının düğüm modelleri tanıtılmaktadır. Benzetim ortamında bulunan N1, N2 ve N3 algılayıcı düğümleri insan vücudundaki fizyolojik sinyalleri (EKG, EEG, kan basıncı, kan şekeri, pals oksimetresi vb.) algılayıp erişim noktasına (PANc) kablosuz olarak gönderebilecek şekilde tasarlanmıştır. N1 ve N2 düğümü için, düğüm editörü kısmında görülen (Şekil 3.2) paket üretme kaynağı kısmında, EKG verileri önceden kaydedilmiş dosyadan okunarak bir alt katmana paket haline getirilmesi için gönderilmektedir. N3 düğümü ise, düğüm editör kısmındaki paket üretme kaynağında her 60 saniyede bir 34 ile 38 derece arasında rastgele bir değer üreterek vücut sıcaklığı değerlerini oluşturmaktadır. Paketler her katmanda o katmana özel olarak paketlenerek Zigbee haberleşme protokolüne göre CFP ya da CAP dilimleri üzerinden erişim noktasına

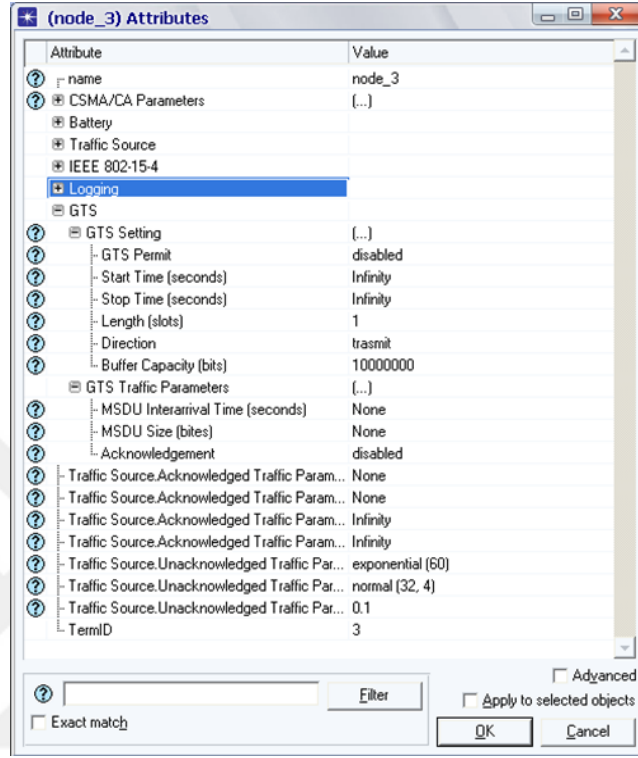
(PANc) gönderilmektedir. Eğer düğüm tarafından GTS isteğinde bulunulmamışsa, veri paketleri uygulama katmanındaki (Application Layer) ‘Traffic Source’ kısmında, GTS isteğinde bulunulmuş ise ‘GTS Traffic Source’ kısmında üretilerek PANc’ye gönderilmektedir. Düğümlerin ağ ile ilgili tüm özellikleri ‘Attributes’ kısmından yapılmaktadır. Bu ayar penceresinden GTS tahsis isteğinde bulunup bulunmayacağı, isterse kaç dilim istendiği, paket büyüklükleri vb. gibi birçok ayar yapılabilir. N1 düğümü ve N3 düğümü için ayarların yapıldığı ‘Attributes’ kısımları sırasıyla Şekil 3.4 ve Şekil 3.5’de görülmektedir.



Şekil 3.4. N1 düğümüne ait ayarlar

Şekil 3.4’de görüldüğü gibi düğümün paketleri gönderme yönü olan ‘Destination MAC Address’ kısmı ‘PAN_Coordinator’ olarak ayarlanmıştır. IEEE 802.15.4 ayarları kısmından, cihaz veri toplayıp gönderim yaptığı için ‘End Device’ seçeneği seçilmiştir. N1 düğümü için senaryo-1 de GTS tahsis isteğinde bulunulduğu için ilgili ayar olan ‘GTS Permit’ seçeneği ‘enabled’ olarak ayarlanmış ve iki dilim GTS tahsis isteğinde bulunulmuştur. ‘GTS Traffic Parameters’ kısmında, paket üretme sıklığı, yaralı yükün (payload) boyutu, onaylı (ACK) ya da onaysız (NACK) iletişim kurulup kurulmayacağı ayarları yapılmaktadır. ‘MSDU Interarrival Time’ paketlerin hangi sıklıkla üretileceği ile ilgili ayardır. N1 düğümü 100 Hz ile örneklenmiş EKG verilerini okuyup paketleyerek PANc’ye gönderdiği için bu ayar ‘exponential (0,01)’ olarak

(saniyede 100 paket) ayarlanmıştır. Yararlı yükün (payload) boyutu (MSDU size (bites)) ise EKG verilerinin kayan noktalı sayı (floating point) olmalarından dolayı 32 bit olarak ayarlanmıştır.

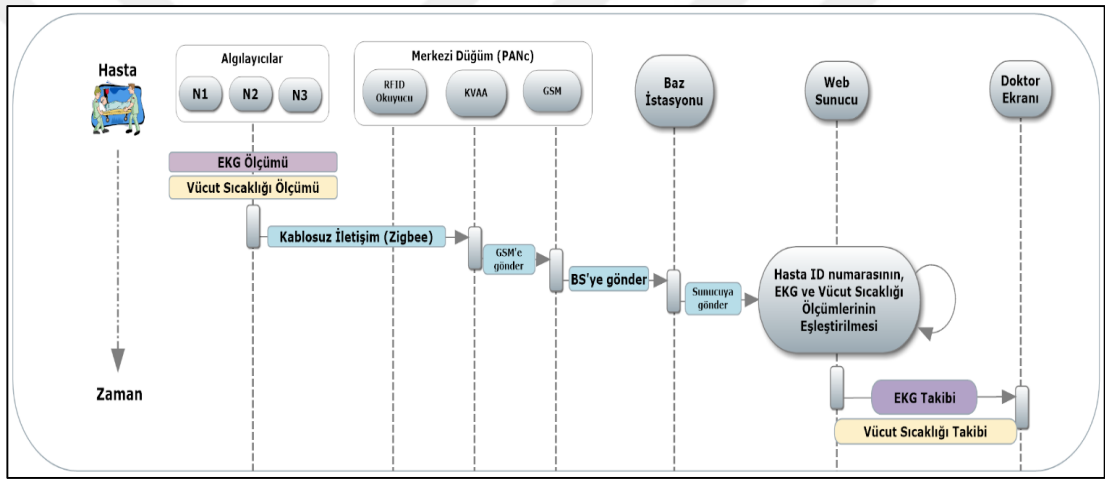


Şekil 3.5. N3 düğümüne ait ayarlar

N3 düğümü düşük veri trafiğine sahip (60 saniyede bir paket) bir düğüm olduğundan GTS tahsisine ihtiyaç duyulmamaktadır. Bu yüzden 'GTS Permit' seçeneği 'disabled' olarak ayarlanmıştır. GTS tarafında hiçbir paket üretilmeyeceği için 'GTS Setting' ayarlarındaki 'Start Time' ve 'Stop Time' kısımları 'Infinity' olarak ayarlanmıştır. N3 düğümü için GTS isteğinde bulunulmadığından, paketler uygulama katmanının (Application Layer) 'Traffic Source' kısmında üretilerek PANc'ye gönderilmektedir. Bu yüzden paket ayarları ilgili kısmın 'Traffic Source' seçeneklerinden yapılmaktadır. Vücut sıcaklığı değerleri 60 saniyede bir üretildiği için bu ayar 'exponential (60)' olarak ayarlanmıştır.

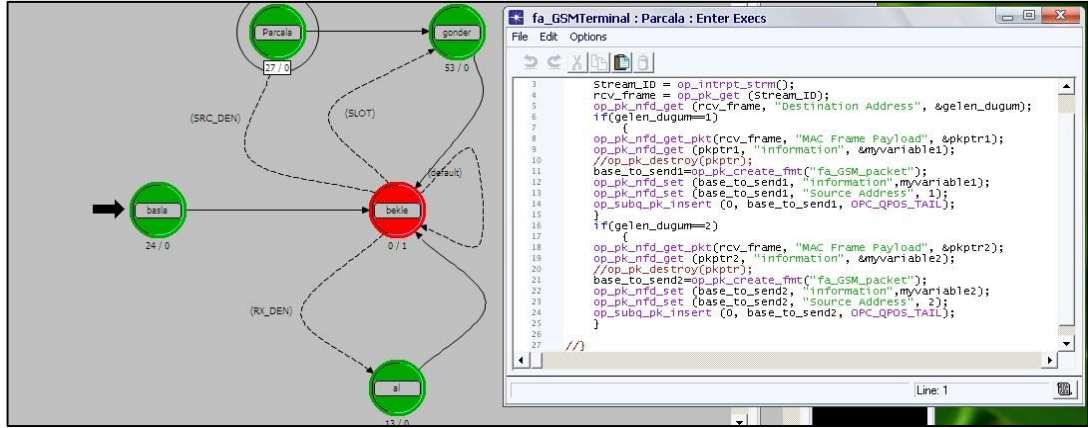
Doktora tez çalışmasında önerilen KVAA'nın benzetim senaryosunun sıralı diyagramı Şekil 3.6'da görülmektedir. Tez çalışmasının benzetim kısmında, koordinatör düğüm (PANc), düğümlerden gelen (EKG ve vücut ısı değerleri) verileri toplamak, her bir hastanın üzerindeki RFID etiketinden kişisel verilerini okumak ve her bir hastanın ID

numarasını ilgili düğüme eklemek için çok amaçlı bir entegre sistem olarak tasarlanmıştır. Koordinatör düğüm (PANc), KVAA, RFID ve GSM sistemi arasında bir köprü gibi davranmaktadır. PANc'nin temel işlevleri, KVAA aracılığıyla hastalardan toplanan fizyolojik verileri içerisinde bulunduran ZigBee mesajlarının protokol veri birimlerini (Protocol Data Units -PDU) paketlemek, aynı anda RFID okuyucu aracılığıyla hastaların bilgilerini RFID okuyucudan okumak ve daha sonra bu bilgi paketlerini GSM/GPRS çerçevelerine dönüştürmektir. Dönüştürülen bu paketler daha sonra baz istasyonu aracılığıyla web sunucusu tarafına iletilmektedir. RFID okuyucu ve GSM modülü arayüzü (Şekil 3.3) bu işlemleri yerine getirmek için PANc'ye entegre edilmiştir.



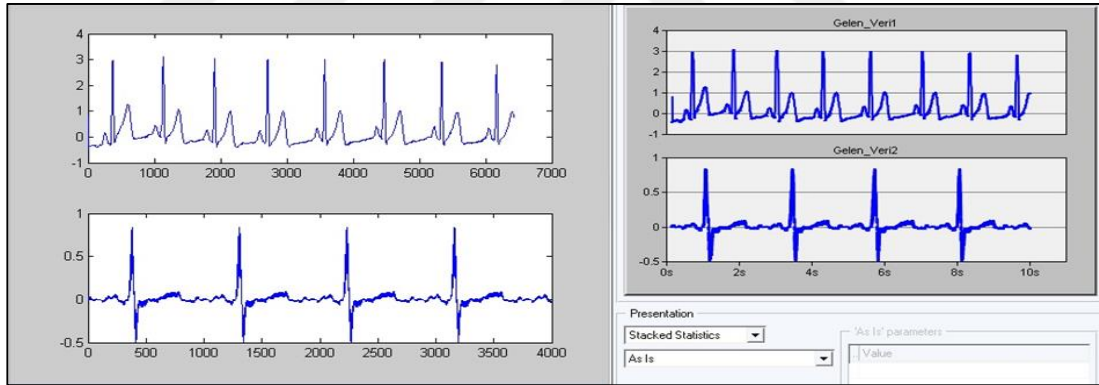
Şekil 3.6. Önerilen KVAA'nın benzetim senaryosunun sıralı diyagramı

Benzetim senaryosunda, üç düğümden gelen EKG ve vücut sıcaklığı verileri 'wpan_mac' kısmında ayrıştırılarak (pars edilerek) istatistikleri tutulmaktadır. Düğümlerden gelen paketler baz istasyonuna aktarılmak üzere 'GSM_MODULE' kısmına aktarılmaktadır (Şekil 3.3). 'GSM_MODULE' kısmında, düğümlerden gelen EKG verilerinin ham halleri buldukları paketler içerisinde ayrıştırılarak alınmaktadır. Bu veriler, baz istasyonuna gönderilmek üzere uygun şekilde başka bir paket formatında (gsm_packet_format) tekrar paketlenmektedirler. Yeni paket içine gelen EKG verisinin hangi hastaya ait olduğunu belirten bir 'ID' numarası eklenmiştir. Şekil 3.7'de 'GSM_MODULE' kısmının görüntüsü görülmektedir.



Şekil 3.7. GSM_MODULE işlem modülü ve kodları

Düğümler tarafından okunup erişim noktasına gönderilen EKG verileri sonraki çalışmalara temel oluşturması açısından Matlab ortamında izlenmiştir (monitoring). Şekil 3.8’de baz istasyonuna gelen veriler ile Matlab ortamında yapılan izleme (monitoring) sonuçlarının (sol kısım Matlab grafikleri) karşılaştırılması görülmektedir.



Şekil 3.8. Baz istasyonuna gelen veriler ile Matlab ortamındaki izleme sonuçları

Matlab izleme grafiğinde, üretilen paket sıklıklarına göre elde edilen grafikler Şekil 3.8’de görülmektedir. Okunan dosyalardaki örnekleme frekansından dolayı birinci düğümde 8 tane Q-R-S sinyali ikinci düğümde 4 tane Q-R-S sinyali görülmektedir. Daha düşük örnekleme frekansları için daha fazla Q-R-S sinyali görülebilmektedir.

3.1. Benzetim Sonuçları ve Performans Değerlendirmeleri

Doktora tezinin bu kısmında, önerilen sistemin iki ana bileşeni olan KVAA ve RFID iletişim protokolü benzetim sonuçları ve performans değerlendirmeleri ayrıntılı olarak

açıklanmaktadır. Tez çalışmasında önerilen sistem, IEEE 802.15.4 standardı veri bağı ve fiziksel katmanlarının performans değerlendirmesi için Riverbed Modeler yazılımı kullanılarak modellenmiştir ve benzetimi gerçekleştirilmiştir. Geliştirilen sistemin performans değerlendirmesi için, servis kalitesi (QoS) gereksinimleri (uçtan uca gecikme ve veri oranı) göz önüne alınarak, KVAA aracılığıyla insan vücudundan alınan fizyolojik sinyallerin aktarımına yönelik üç farklı senaryo ele alınmıştır. Bu performans ölçütlerine ek olarak, önerilen sistemin enerji verimliliğini göstermek için N1, N2 ve N3 düğümlerinin enerji tüketim oranları da incelenmiştir. Her üç senaryo için benzetim parametreleri Tablo 3.1'de listelenmiştir.

Tablo 3.1. Benzetim parametreleri

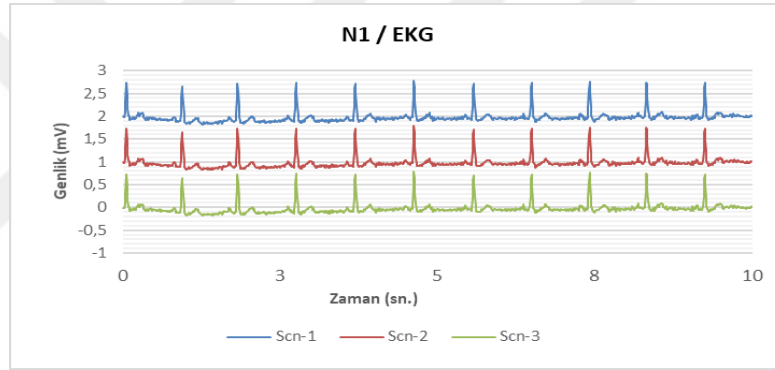
Parametre	Senaryo-1			Senaryo-2			Senaryo-3		
	N1*	N2+	N3▲	N1*	N2+	N3▲	N1	N2+	N3▲
GTS / CAP Sayısı	2 / -	7 / -	- / CAP	- / CAP	7 / -	- / CAP	- / CAP	- / CAP	- / CAP
Paket Üretme Sıklığı (s)	0,01	0,002	60	0,01	0,002	60	0,01	0,002	60
Örnekleme Oranı (Hz)	100	500	0,016	100	500	0,016	100	500	0,016
Veri Oranı	13,8 Kbps	68 Kbps	3 bps	13,8 Kbps	68 Kbps	3 bps	13,8 Kbps	68 Kbps	3 bps
Benzetim Süresi	10 dakika			10 dakika			10 dakika		
Frekans (GHz)	2,4			2,4			2,4		
T^{SuperFrame} (ms)	30,72			30,72			30,72		
Tslot (ms)	1,92			1,92			1,92		
BO & SO	1-1			1-1			1-1		
MicaZ Parametreleri	2 AA PİL (3V)			2 AA PİL (3V)			2 AA PİL (3V)		

* EKG – Node 1 + EKG – Node 2 ▲ Vücut Sıcaklığı – Node 3

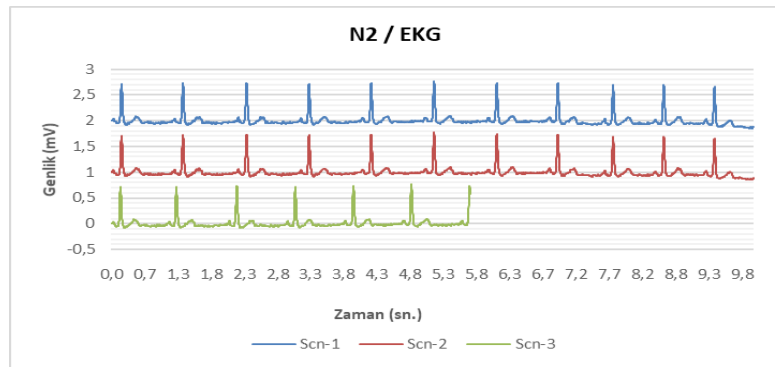
Yapılan bütün benzetimlerde, iki ayrı düğümden üç uçtan farklı frekanslarda örneklenecek alınmış EKG verileri ile vücut sıcaklığı verisi kullanılmıştır. EKG verilerinden ilki 100 Hz (N1), diğeri ise ağ performansını daha iyi değerlendirebilmek amacıyla 500 Hz (N2) ile örneklenecek kaydedilmiştir. Vücut sıcaklığı verileri ise benzetim süresi boyunca her 60 saniyede (dakikada bir) bir örneklenecektir. EKG verileri zaman açısından kritik olduğundan, bütün senaryolar için IEEE 802.15.4 MAC protokolünün BO ve SO değeri (uyku durumu yok, sürekli aktif periyotta çalışma) 1

olarak (optimum değer) ayarlanarak süper çerçeve ve zaman dilimi süreleri elde edilmiştir.

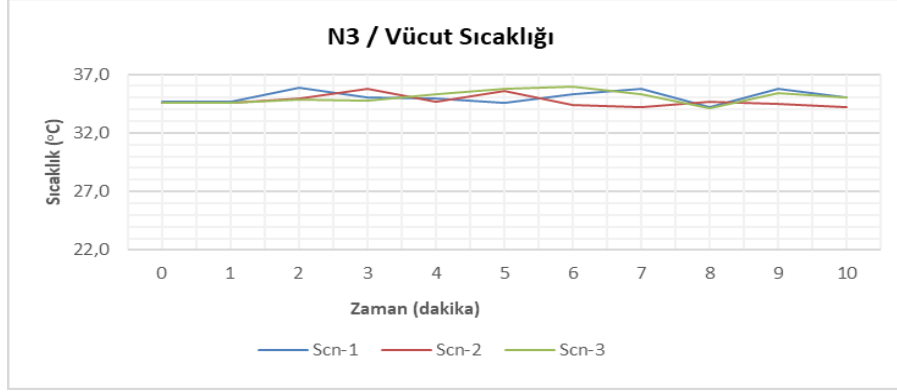
Tablo 3.1'de görüldüğü gibi, düşük yük trafiğine sahip N1 (100 Hz/ECG) düğümü için, birinci senaryoda, verilerin iletiminde 2 GTS dilimi kullanılmakta iken diğer senaryolarda, verilerin iletimi için CAP (1 - 4 zaman dilimi) dilimleri kullanılmaktadır. Ağır yük trafiğine sahip N2 (500 Hz/ECG) düğümü ise, senaryo-1 ve 2 için 7 GTS dilimi kullanırken, senaryo-3 için CAP dilimlerini kullanmaktadır. N1 ve N2'nin aksine, N3 vücut sıcaklığı verilerini 16 mHz'lik (miliHertz) çok düşük bir veri hızı ile iletmektedir. Bu nedenle N3 düğümü için dikkate alınan tüm senaryolarda CAP dilimleri kullanılmaktadır. N1, N2 ve N3 tarafından işlenen gerçekçi fizyolojik veriler sırasıyla Şekil 3.9, Şekil 3.10 ve Şekil 3.11'de görülmektedir.



Şekil 3.9. N1 düğümü senaryo 1, 2 ve 3 için EKG değerleri



Şekil 3.10. N2 düğümü senaryo 1, 2 ve 3 için EKG değerleri



Şekil 3.11. N3 düğümü senaryo 1, 2 ve 3 için vücut sıcaklığı değerleri

Düğümlerden alınan EKG ve vücut sıcaklığı verilerinin QoS gereksinimlerini karşılayarak etkili bir şekilde iletimi için gerekli olan veri oranı ve bu verilerin iletimi için gerekli olan dilim sayısı hesabı genel olarak şu şekilde yapılmıştır;

Hesaplamalar yapılırken Süper çerçeve Zaman Dilimi Sayısı (SZDS) = 16, CAP dilimlerinin minimum uzunluğu 'aMinCAPLength' = 440 sabit değeri, Symbolrate = 62,5 Kilosembol, MAC_HEADER_SIZE = 104 bit (MAC başlığının varsayılan boyutu) ve yararlı yük (payload) = 32 bit (IEEE-754 kayan noktalı sayı için) değerleri kullanılmıştır. BO ve SO değerleri 1 olarak seçildiğinde bir süper çerçevenin uzunluğu Denklem (3.1)'e göre hesaplanmaktadır. Her bir dilimin uzunluğu ise Denklem (3.2)'ye göre hesaplanmaktadır;

$$BI = aBaseSuperframeDuration \times 2^{BO} = BI = SD = 960 \times 2^1 \Rightarrow 1920 \quad (3.1)$$

$$\text{Dilim Genişliği: } Slot_{sembol} = BI/SZDS = 1920/16 = 120 \text{ sembol} \quad (3.2)$$

PANc'den talep edilebilecek maksimum GTS sayısı (MGTS) Denklem (3.3)'e göre hesaplanmaktadır;

$$MGTS = \frac{BI - aMinCAPLength}{Slotsembol} = \frac{1920 - 440}{120} = 12 \quad (3.3)$$

Bu hesaba göre ortamdaki tüm düğümlerin istekte bulunabilecekleri maksimum GTS sayısı 12 olabilmektedir (SO=1 ve BO=1). Herhangi bir düğümün ürettiği trafiğe göre kaç GTS isteği yapabileceği şu şekilde hesaplanmaktadır; Bir süper çerçevenin toplam süresi (Tsuperframe) seçilen BO değerine göre değişir ve bu değer

Denklem (3.4)'e göre hesaplanmaktadır;

$$T_{\text{superframe}} = BI / \text{Symbolrate} = 1920 / 62,5 \text{ Kilosembol} = 30,72 \text{ ms.} \quad (3.4)$$

Bulunan bu değere göre bir saniyede üretilen kılavuz sayısı (SKS) Denklem (3.5)'e göre hesaplanmaktadır;

$$\text{SKS} = 1 / T_{\text{superframe}} = 1 / 30,72 \text{ ms.} \approx 30 \quad (3.5)$$

Zigbee protokolünde her sembol 4 bitle temsil edilmektedir. Bu durumda her bir dilimin bit uzunluğu (Slotbit) Denklem (3.6)'ya göre hesaplanmaktadır;

$$\text{Slotbit} = \text{Slotsembol} \times 4 = 120 \times 4 = 480 \text{ bit} \quad (3.6)$$

Düğümün ürettikleri trafiklere göre bant genişliği ihtiyacı (BGİ) Denklem (3.7)'ye göre hesaplanmaktadır;

$$\text{BGİ} = (\text{MAC_HEADER_SIZE} + \text{Yararlı Yük (Payload)}) \times \text{Örnekleme} \quad (3.7)$$

Örneğin; N2 düğümünün ürettiği trafiğe göre hesaplama yapıldığında (Denklem (3.8)), verilerin sorunsuz bir şekilde karşıya gönderimi için gerekli olan bant genişliği 68 Kbit çıkmaktadır;

$$\text{BGİ} = (104 + 32) \times 500 = 68000 \text{ bit} = 68 \text{ Kbit} \quad (3.8)$$

Hesaplamalar doğrultusunda N1 düğümü için gerekli olan bant genişliği 13,6, N3 düğümü için gerekli bant genişliği ise 3 bit çıkmaktadır. Böylece benzetim süresi içerisinde gerekli olan toplam bant genişliği 81,6 Kbit/saniye olarak hesaplanmıştır.

Zamanda kritik verilerin iletimi için düğümler GTS isteğinde bulunabilirler. GTS tahsisi isteği değeri (GTSİ) düğümün ürettiği trafik yoğunluğuna göre değişebilmektedir. Her düğüm için minimum kaç GTS dilimi isteği yapılabileceği Denklem (3.9) ile bulunmaktadır;

$$\text{GTSİ} = \frac{\text{BGİ} \times \text{SZDS}}{\text{SKS} \times \text{MGTS} \times \text{Slotbit}} \quad (3.9)$$

Örneğin, N2 düğümü için bu değer hesaplandığında bir saniye boyunca verilerin tamamının karşıya gönderilmesi için gereken minimum GTS dilimi 7 olarak hesaplanmıştır (Denklem (3.10)). Bu değer altında istenecek GTS tahsisi, gönderilmesi gereken verilerin karşıya gönderilmesi için gerekli bant genişliğini sağlamayacak ve verilerin bir kısmı karşıya gönderilmeyecektir. Bu durumda uçtan uca gecikmeler artacak, veri oranı değeri ise azalacaktır;

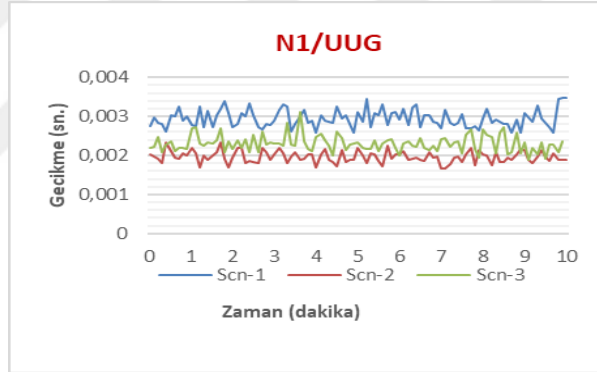
$$GTS\dot{I} = \frac{68000 \times 16}{30 \times 12 \times 480} \approx 7 \quad (3.10)$$

Zamanda kritik verilerin aktarımı için, düğümler PANc'ye bir GTS talebi gönderir ve bu talep mevcut kaynaklar dikkate alınarak PANc tarafından değerlendirilir. Herhangi bir uygulama için GTSİ, düğümün trafik türüne bağlı olarak değişebilmekte ve yukarıda verilen formül kullanılarak belirlenebilmektedir. Örneğin, QoS gerekliliklerini sağlaması için N1 ve N2 düğümleri için gerekli minimum GTSİ sırasıyla 2 ve 7 olarak hesaplanmıştır. Doktora tez çalışmasının en önemli katkılarından biri de sunulan formüller aracılığıyla zamanda kritik fizyolojik verilerin QoS gerekliliklerini sağlayarak iletilmesi için gerekli olan BGİ ve GTSİ hesaplamalarının etkin bir şekilde yapılabilmesidir.

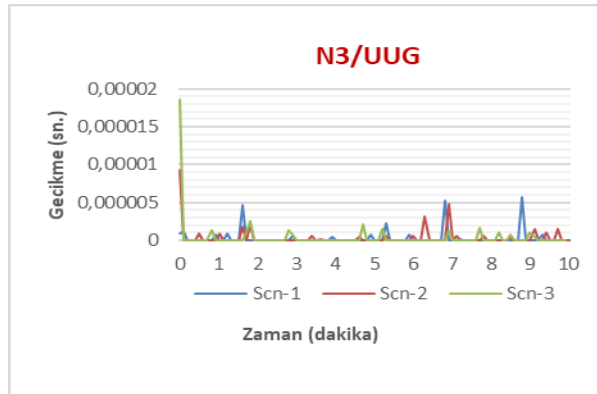
Tablo 3.1'de hesaplanan parametreler, değişen senaryolar altında sistem performansını değerlendirebilmek için benzetim ortamında kullanılmıştır. Benzetim sonuçlarından ilk olarak uçtan uca gecikme (UUG) süreleri incelenmiştir. Uçtan uca gecikmeler paketin ilk üretildiği zaman ile hedef tarafından alındığı zaman arasındaki farktır. Şekil 3.12'de ve Şekil 3.13'de N1, N3 düğümlerinin, Şekil 3.14'de ise N2 düğümünün bütün senaryolar için UUG değerleri görülmektedir. Şekil 3.12'de görüldüğü gibi N1 düğümünün her üç senaryo için UUG değerleri 0,002 ile 0,003 saniye arasında değişmektedir. N1 düğümünün paket trafiği düşük veri hızına (100 Hz EKG) sahip olduğundan, senaryo-2 ve senaryo-3'te CAP dilimleri kullanılsa bile sonucun değişmediği görülmektedir. Çok düşük veri trafiğine sahip N3'ün UUG'si, Şekil 3.13'de görüldüğü gibi üç senaryonun her biri için 5 mikrosaniyeden az çıkmıştır. N2 düğümü için (yoğun yük trafiği), GTS dilimlerinin kullanıldığı senaryo-1 ve senaryo-2 de UUG değerleri beklenildiği gibi ortalama 0,01 saniye olarak

gerçekleşmiştir. CAP dilimlerinin kullanıldığı senaryo-3'te ise UUG değerleri sürpriz olmayacak şekilde 3 dakika civarlarına çıkmıştır (Şekil 3.14).

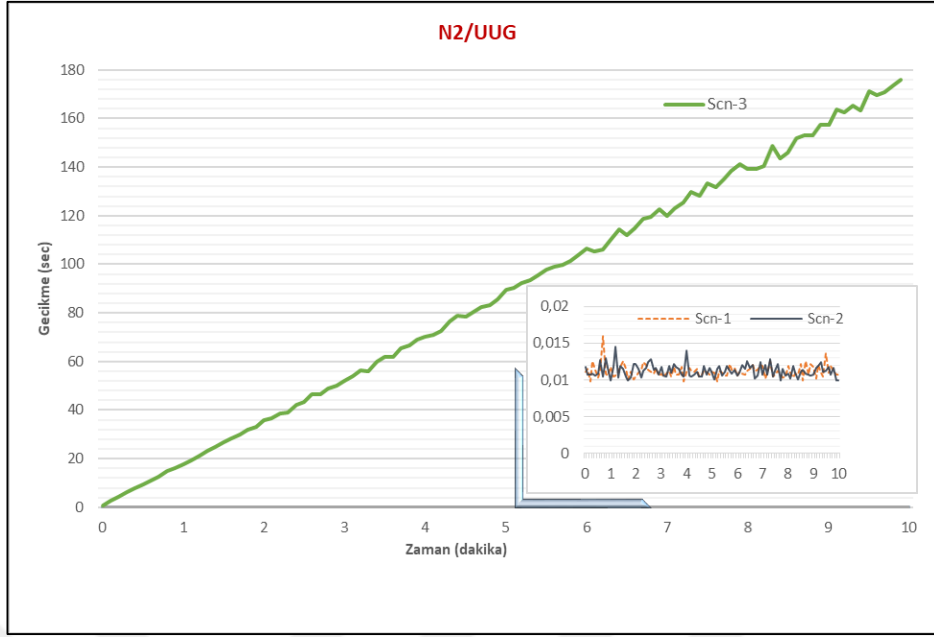
Bu sonuçlara göre, N1 ve N3 düğümleri için bütün benzetim senaryolarında QoS gereksinimleri sağlanmakla beraber alınan verilerin erişim noktasına aktarımında herhangi bir sorun görülmemektedir. N2 düğümünde ise senaryo-1 ve senaryo-2'de GTS dilimleri kullanıldığı için UUG değerleri QoS gereksinimlerini sağlayacak şekilde gerçekleşmiştir. Senaryo-3 de ise GTS dilimleri yerine CAP dilimleri kullanıldığından UUG değerleri Tablo 1.1'de görülen servis kalitesi gereksinimlerinin çok üzerinde gerçekleşmiştir. Bu durumdan dolayı Şekil 3.10'da görüldüğü gibi, verilerin erişim noktasına iletiminde gecikmeler meydana gelmiştir. Gecikmelerden dolayı benzetim süresi boyunca karşı tarafa iletilmesi beklenen EKG verilerinin neredeyse yarısı iletilmemiştir. Meydana gelen bu gecikmeler hastalardan gelen verilerin yanlış değerlendirilmesine sebep olabilmektedir.



Şekil 3.12. N1 düğümü senaryo 1, 2 ve 3 için UUG değerleri

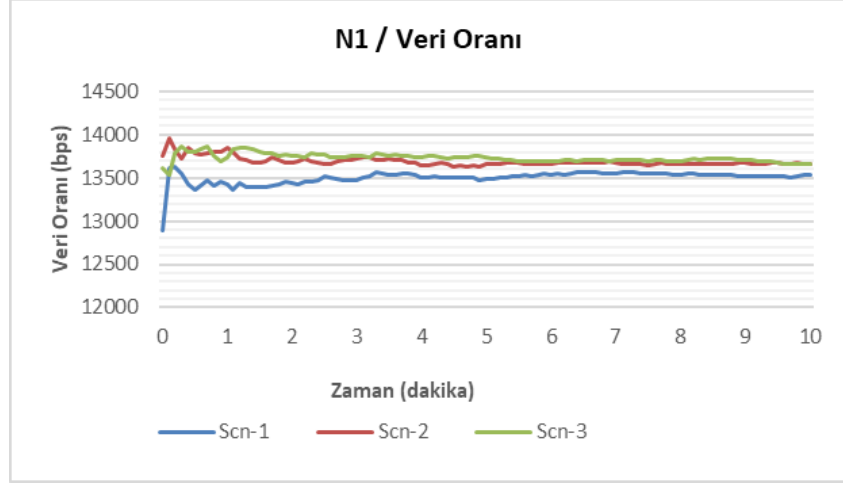


Şekil 3.13. N3 düğümü senaryo 1, 2 ve 3 için UUG değerleri

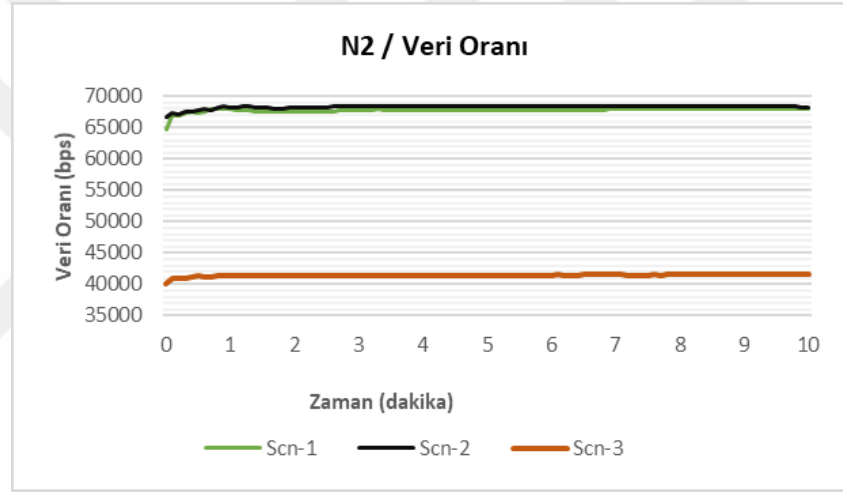


Şekil 3.14. Senaryo 1, 2 ve 3 için N2 düğümü UUG değerleri

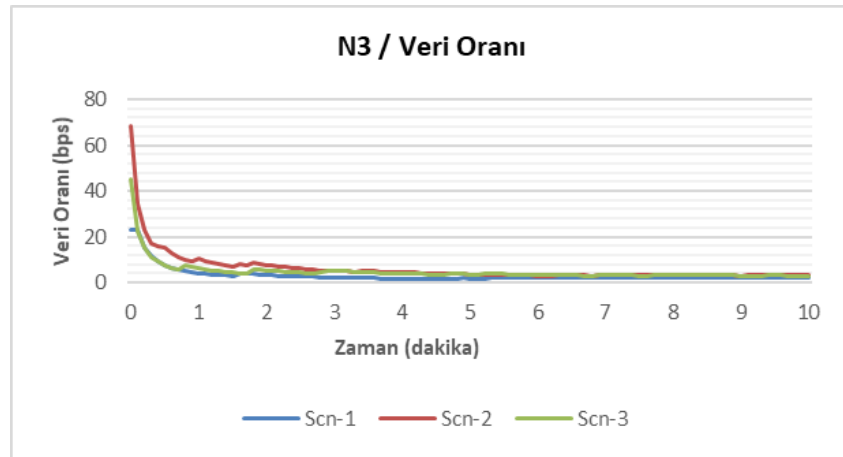
Servis kalitesi kriterleri bağlamında iş çıkarma oranı (veri oranı), ağ performansının değerlendirilmesinde bakılması gereken diğer önemli parametredir. İş çıkarma oranının ne oranda gerçekleştiğini değerlendirmek amacıyla düğümlerin veri oranı değerleri incelenmiştir. Veri oranı değerleri incelendiğinde N1 ve N3 düğümleri için her üç seneryoda da analitik olarak hesaplanmış ($N1=13,8$ Kbps, $N3=3$ bps) değere göre karşıya gönderilmesi gereken tüm bitlerin gönderildiği görülmektedir (Şekil 3.15, Şekil 3.17). N2 düğümü için ise senaryo-1 ve 2 de veri oranı değerleri analitik olarak hesaplanan değerle örtüşerek 68 Kbit olarak gerçekleşmiş ve karşıya gönderilmesi gereken tüm bilgiler gönderilmiştir. Üçüncü senaryoda ise veri trafik yoğunluğunun büyük olması (500 Hz EKG) ve GTS isteği bulunulmamasından dolayı veri oranı, Şekil 3.16'da görüldüğü gibi yaklaşık %40 oranında azalmıştır (42 Kbps). Bu yüzden benzetim süresi boyunca karşıya iletilmesi gereken verilerin neredeyse yarısı iletilenmemiştir (Şekil 3.10). Benzetim sonuçlarına göre düşük trafik yüklerine sahip N1 ve N3 düğümleri için tüm benzetim senaryolarında yüksek iş çıkarma oranları elde edildiği görülmüştür. Yüksek trafik yoğunluğuna sahip N2 düğümü için ise, GTS dilimlerinin kullanıldığı senaryo-1 ve senaryo-2 değerlerinde yüksek iş çıkarma oranı elde edilirken, GTS dilimleri yerine CAP dilimlerinin kullanıldığı üçüncü senaryoda iş çıkarma oranı %40 oranında azalmıştır.



Şekil 3.15. Senaryo 1, 2 ve 3 için N1 düğümü veri oranı değerleri



Şekil 3.16. Senaryo 1, 2 ve 3 için N2 düğümü veri oranı değerleri

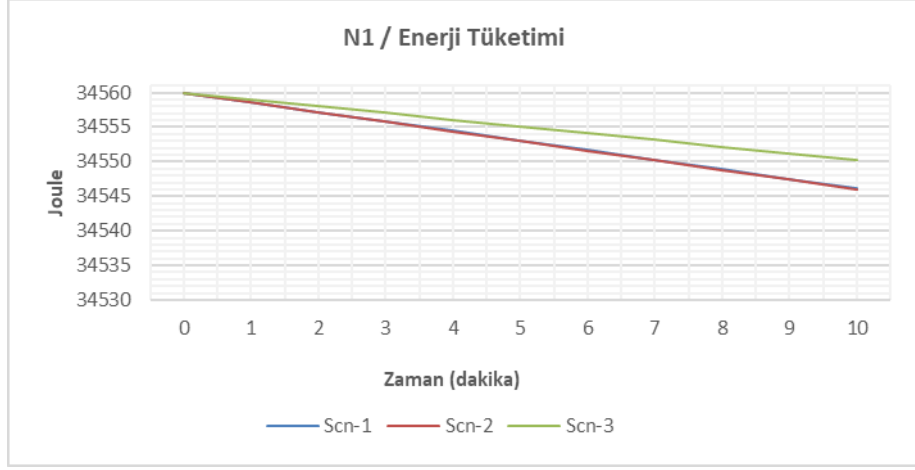


Şekil 3.17. Senaryo 1, 2 ve 3 için N3 düğümü veri oranı değerleri

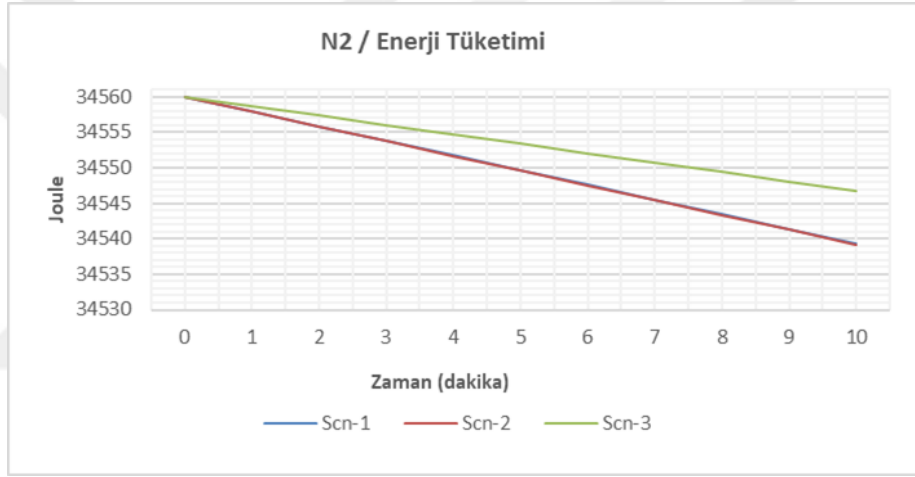
Düğümlerdeki enerji tüketimi, KVAA yapısının performans değerlendirmesinde anahtar tasarım ölçülerinden birisidir. Doktora tezi çerçevesinde geliştirilen sistemin KVAA kısmında, Wi-Fi, GSM/GPRS, Bluetooth gibi kablosuz teknolojilere kıyasla oldukça düşük enerji tüketimi ve daha uzun pil ömrüne sahip olan Zigbee haberleşme protokolü kullanılmaktadır [43, 44]. İletim modunda Wi-Fi, Bluetooth ve Zigbee için mevcut tüketim değerleri sırasıyla 400 mA, 60 mA ve 25–35 mA'dır. Ayrıca Wi-Fi, Bluetooth ve Zigbee için ilgili pil ömürleri, sırasıyla 0,5–5 gün, 1–7 gün, 100–1000 gün ve üzerinde şeklinde gerçekleşmektedir. Bu nedenle, tez çalışmasında önerilen sistem, uzun pil ömrü ve düşük enerji tüketimi nedeniyle enerji verimli olarak ta düşünülebilir.

Sistemin on dakikalık benzetim süresi boyunca enerji tüketimini ölçmek için, micaZ [45] güç modeli seçilmiştir. MicaZ düğümlerinin başlangıç enerji değeri, $2 \times 1,5$ volt AA pil ile üretilen enerjiye karşılık gelen 34560 Joule (J)'dir. Düğümler için enerji tüketimi benzetim sonuçları Şekil 3.18, Şekil 3.19 ve Şekil 3.20'de görülmektedir.

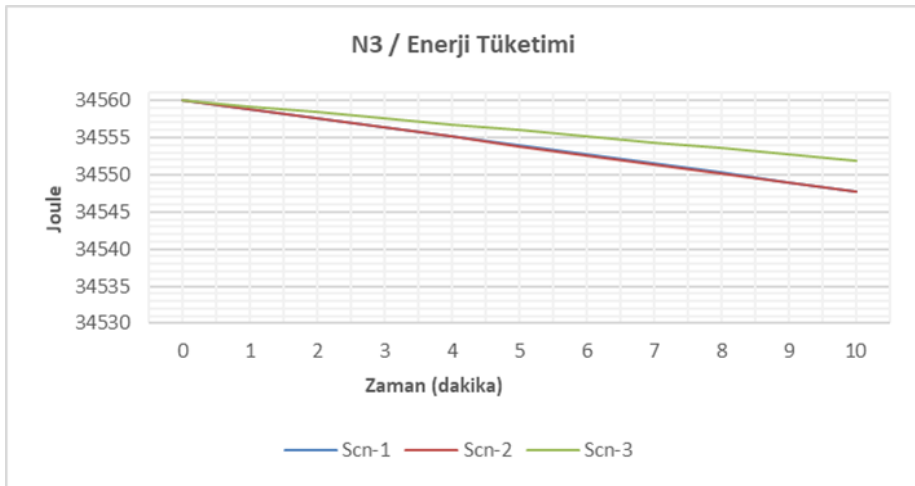
Şekil 3.18'de görüldüğü gibi on dakikalık benzetim süresi boyunca N1 düğümü, senaryo-1 ve 2'de 15 J, senaryo-3'te 10 J enerji tüketmiştir. Diğer taraftan N2 düğümü, senaryo-1 ve 2'de 20 J, senaryo-3'te ise 13 J enerji tüketmiştir (Şekil 3.19). N1 ve N2 düğümlerinin enerji tüketimleri arasındaki fark esas olarak N2'nin (500 Hz / EKG) veri trafiği yoğunluğunun N1 (100 Hz/EKG) düğümünün veri trafiği yoğunluğundan daha yüksek olmasından kaynaklanmaktadır. N3 düğümünün veri trafik yoğunluğu (16 mHz/vücut sıcaklığı) N1 ve N2 düğümlerine kıyasla çok daha düşüktür. Bu nedenle, N3 düğümü için enerji tüketimi, beklendiği gibi N1 ve N2'den daha düşük gerçekleşmiştir. N3 düğümü için senaryo-1 ve 2'de 13 J, senaryo-3'te 7 J enerji tüketimi meydana gelmiştir (Şekil 3.20). Bu iki parametre arasında doğrusal bir ilişki olmamakla birlikte, ağ içindeki trafik yoğunluğu yükseldikçe, düğümlerin enerji tüketiminin arttığı sonucuna varılabilmektedir. Ek olarak, N2 düğümü, senaryo-1 ve 2'den farklı olarak senaryo-3'te veri paketlerinin büyük bir kısmını iletemediğinden, sistemdeki ağ yoğunluğu azalmaktadır. Bu durumdan dolayı her düğümün enerji tüketimi senaryo-3 için azalmaktadır. Sistemdeki düğümlerin batarya ömrünün, sürekli veri iletimi (uyku moduna geçilmemekte) modunda 15–30 gün (üç senaryodan her birine bağlı olarak) arasında değiştiği hesaplanmıştır.



Şekil 3.18. Senaryo 1, 2 ve 3 için N1 düğümü enerji tüketim değerleri



Şekil 3.19. Senaryo 1, 2 ve 3 için N2 düğümü enerji tüketim değerleri

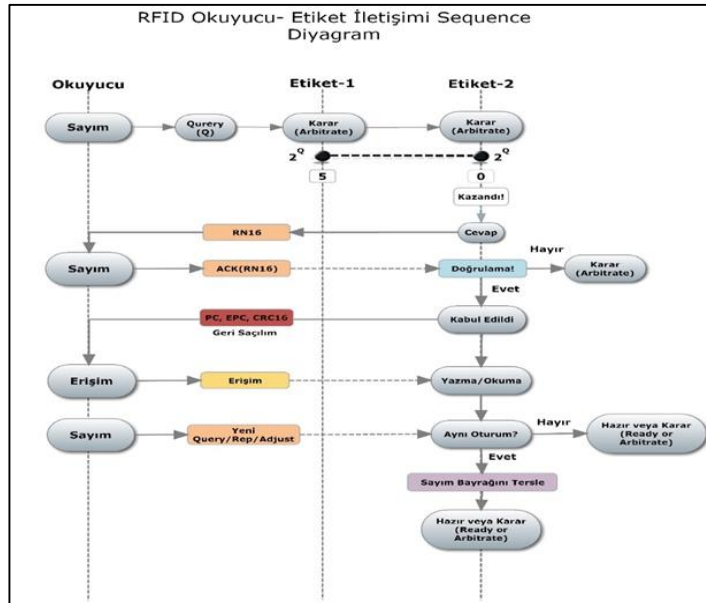


Şekil 3.20. Senaryo 1, 2 ve 3 için N3 düğümü enerji tüketim değerleri

3.2. RFID İletişim Sistemi Yapısı

Gerçekleştirilen platformun benzetiminde RFID sistemi mevcut IEEE 802.15.4 open-Zigbee protokolü içerisine senaryoya uygun olarak eklenmiştir. Tez çalışmasının benzetiminde iletişim katmanı olarak, RFID çoklu etiket anti-çarpışma problemlerini etkin bir şekilde çözdüğü için, adaptif dilimli ALOHA (adaptive slotted ALOHA) protokolü [46] seçilmiştir. Fiziksel katman için ise 13,56 MHz'lik iletişim frekansı kullanılmıştır.

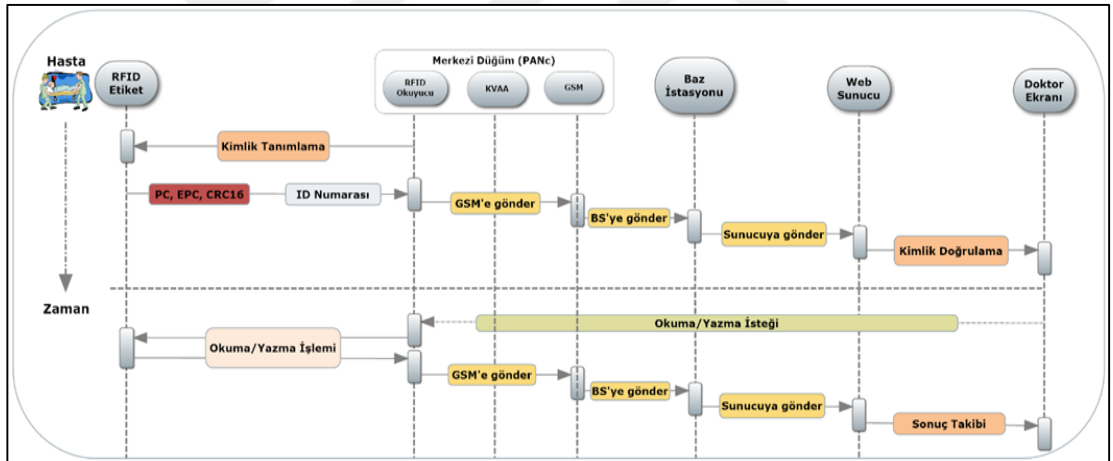
Doktora tez çalışması için gerçekleştirilen platformun benzetimi çalışmasında RFID iletişim protokolünün uygulama katmanına odaklanılmıştır. RFID okuyucu, seçme, envanter ve erişim işlemlerini kullanarak etiket yoğunluğunu düzenler. Seçme işlemi, okuyucunun yakınındaki belirli bir etiket grubuna bir seçme komutu gönderilerek uygulanmaktadır. Bu işlem, web sunucusunda mevcut olan veritabanından bir kaydın seçilmesine benzerdir. Envanter işlemi, okuyucunun yakınındaki farklı hasta veya nesnelere tanımlayan etiketleri sınıflandırmaktadır. Seçme işlemi ya da envanter işlemini bitirdikten sonra okuyucu, ilgili etikete erişmeyi seçebilmektedir. Olası bir erişim işleminden sonra, okuyucu ya iletişimi durdurur ya da yeni bir oturum başlatır. Şekil 3.21'de, tez çalışmasında kullanılan RFID okuyucu-etiket iletişim protokolünün sıralı diyagramı görülmektedir [47, 48].



Şekil 3.21. Tez çalışmasında kullanılan RFID okuyucu-etiket iletişim protokolünün sıralı diyagramı [47]

Okuyucu ve etiket arası iletişimde en önemli parametre, okuyucu tarafında etikete gönderilen 'Query' komutu içerisindeki 'Q' değeridir. 'Q' değerinin çok büyük ya da küçük seçilmesi, etiketlerin tümünün ya çok geç cevap vermesine ya da etiketlerin tamamının cevap verememesine neden olabilmektedir. Bu durumda 'Q' değerini arttırmak ya da azaltmak için 'QA' komutu gönderilmektedir. Bu komut içerisinde tanımlanacak bir değere göre 'Q' değeri artırılıp azaltılabilmektedir.

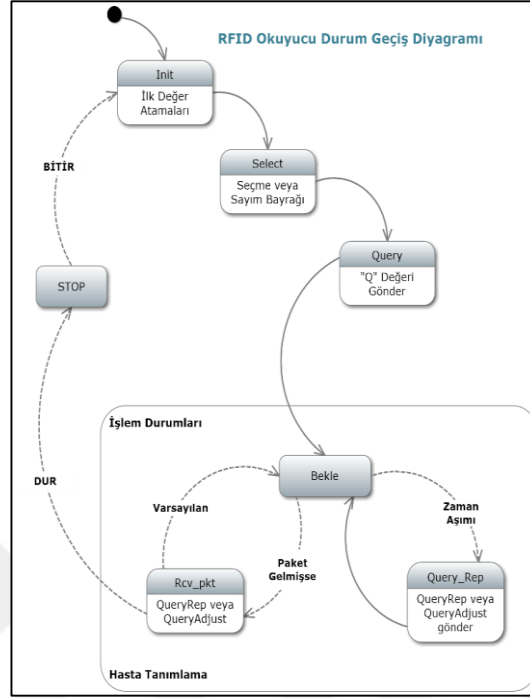
PC, EPC ve CRC16 kodunu gönderen etiketlere erişim yapıp işlem yapılmak isteniyorsa bunun için okuma, yazma, kilitleme vb. için istek gönderilir. Bu istek geldiğinde etiket okuyucunun bu isteğine cevap vermektedir. Eğer okuyucu isterse bu işlemden sonra iletişimi bitirebilir ya da yeni bir oturum açabilir. Örneğin; hastane ortamında RFID ile ilk önce hastalara ait bilgilere ulaşip yazma ya da okuma yapıldıktan hemen sonra başka bir oturum yapılarak hastane içerisinde üzerinde RFID etiket bulunan ilaçların sayımı yapılabilir.



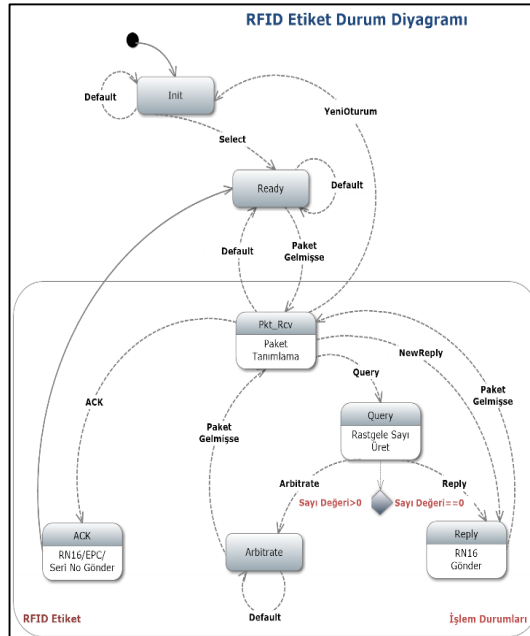
Şekil 3.22. Önerilen RFID haberleşme mekanizması benzetim senaryosunun sıralı diyagramı

Doktora tez çalışmasında gerçekleştirilen benzetim çalışmalarındaki senaryoda, hastalar üzerinde bulunan etiketlere erişilerek etikette bulunan hastalara ait kişisel bilgiler okunmaktadır. Okunan bu bilgiler PANc içerisinde bulunan 'rfid_reader_mac' kısmından baz istasyonuna gönderilmek üzere 'GSM_MODULE' kısmına aktarılmaktadır. Bu kısımda hasta bilgileri, gelen paket içerisinde parçalanarak (ayrıştırılarak) baz istasyonuna gönderilmek üzere uygun formatta tekrar paketlenmektedir. Şekil 3.1'de verilen benzetim senaryosu için RFID iletişim mekanizmasının sıralı diyagramı Şekil 3.22'de gösterilmektedir. RFID okuyucu-etiket

düğümü modellerinde kullanılan algoritmalar (proses modelleri), sırasıyla Şekil 3.23 ve Şekil 3.24'de gösterilmektedir.



Şekil 3.23. Önerilen RFID okuyucu işlem (process) modelinin algoritması



Şekil 3.24. Önerilen RFID etiket işlem (process) modelinin algoritması

Benzetimi gerçekleştirilen RFID sistemi senaryosunda iki hasta ve bu hastalara ait kişisel bilgileri bulunduran iki adet RFID etiket bulunmaktadır. Benzetim çalışmasında, sistemdeki etiket yoğunluğunun az olmasından dolayı 'Q' değeri üç olarak seçilmiştir. 'Q' değeri okuyucu tarafından etiketlere gönderildikten sonra bu etiketler aldıkları bu değere göre rastgele bir sayı (0 ile 2^Q-1 arasında) üretmektedir. Ürettikleri sayı sıfıra eşit değilse etiketler, 'ARBITRATE' durumuna geçerek 'QR' komutu beklemektedirler. 'Query' komutunu gönderen okuyucu belirli bir süre etiketlerden cevap gelmesini beklemektedir. Daha önceden belirlenen bir süre içinde (ms) herhangi bir etiketten cevap gelmemişse, her etikette üretilen rastgele sayıyı güncellemek için RFID okuyucusu tarafından 'QR' komutu gönderilmekte ve bu işlem rastgele sayı sıfır olana kadar devam etmektedir. 'QR' gönderilmesine rağmen etiketlerden çok uzun süre cevap gelmezse, okuyucu tarafından etiketlere, 'Q' değerini azaltıp yeni bir rastgele sayı üretmeleri ve cevap verme olasılıklarını arttırmak için 'QA' komutu gönderilmektedir. Bu işlem herhangi bir etiketten cevap gelene kadar devam etmektedir. Herhangi bir etiketin rasgele sayı değeri sıfır olduğunda, etiket okuyucuya 16 bitlik rastgele bir sayı (RN16) ile yanıt vermektedir. Okuyucu daha sonra ilgili rastgele sayıyı (RN16) 'ACK' komutuyla etikete geri göndermektedir. Okuyucu tarafından etikete geri gönderilen değer (ACK değeri) doğrulanırsa, etiket içeriğinde seri numarasının bulunduğu elektronik ürün kodunu (electronic product code - EPC) okuyucuya göndermektedir. Belirlenen senaryoya göre okuyucu, etiketlerden okuma işlemi gerçekleştirmektedir (Şekil 3.22). EPC kodunu gönderen etikete 'READ' isteği gönderilmektedir. 'READ' isteğini alan etiket, kendi içerisinde tanımlı olan kişisel bilgileri (kimlik numarası ve doğum yılı) PANc içerisine entegre edilmiş okuyucuya göndermektedir. PANc daha sonra bu bilgiyi 'GSM_MODULE'ü kullanarak baz istasyonu üzerinden tekrar web sunucuya göndermektedir (Şekil 3.1 ve Şekil 3.3). Web sunucuda hasta kimlik numarası ile veri tabanındaki bilgiler eşleştirilerek hastaya ait tüm kişisel bilgilere erişilmektedir. Daha sonra KVAA aracılığıyla alınan fizyolojik veriler (EKG ve vücut sıcaklığı) ile hastaya ait kişisel veriler web sunucu tarafında eşleştirilmektedir. Şekil 3.25'de RFID okuyucu-etiket iletişiminin aşamaları ve detayları görülmektedir.

```
--*--*--*Patient 1 Surname : Aktas
--*--*--*Patient 1 Birthdate :1983
--*--*--*Identification Completed--*--*--*
**1. etiket select geldi: **
**2. etiket select geldi: **
1. etiket Uretilen deger: 2**
2. etiket Uretilen deger: 0**
2. etiket RN16 degeri: 17276**
**ACK gonderildi:17276 **
2. Etiket EPC gonderildi: **
Gelen Seri no: 112233**
**2. etiket okuma verileri gonderildi**
+--+2. verileri Baz istasyonuna yönlendirildi:++-
--*--*--* istatistik yazildi
--*--*--*Patient 2 ID Number :1111111111

--*--*--*Patient 2 Name : Haluk
--*--*--*Patient 2 Surname : Aktas
--*--*--*Patient 2 Birthdate :1990
[Mode: PAN coordinator, t=0.42090, S: 1111111111, EPC OF TAG]
```

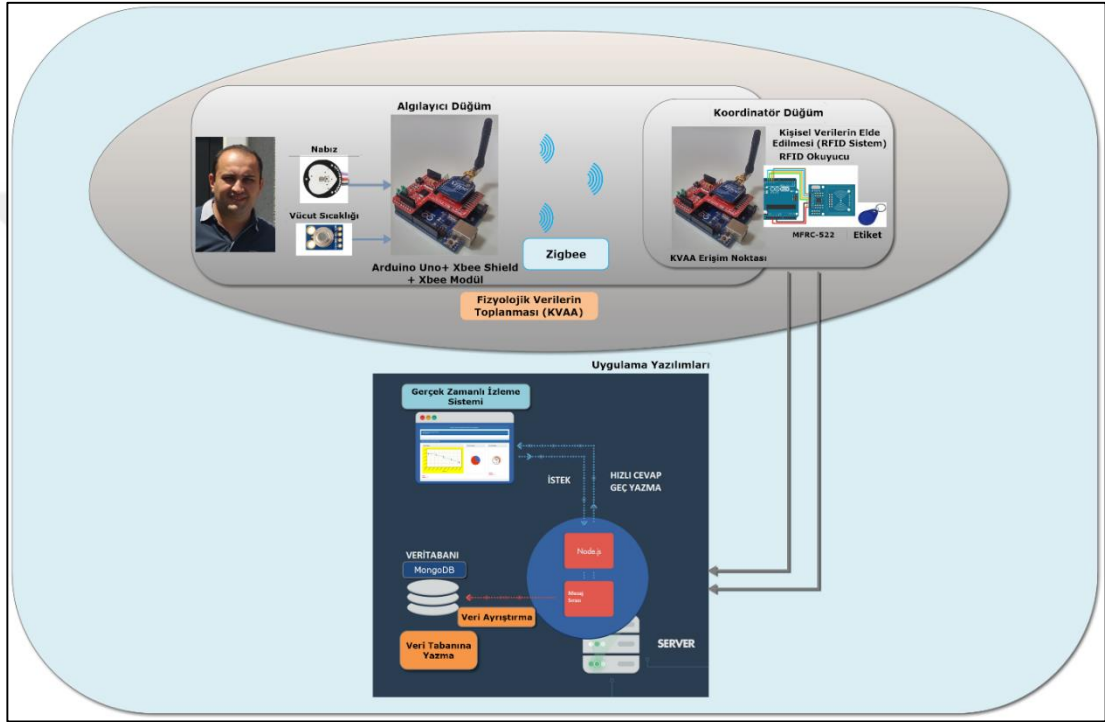
Şekil 3.25. Örnek bir RFID hasta tanımlama sürecinin benzetim çalışma alanındaki görüntüsü

3.3. Sonuç

Doktora tez çalışması kapsamında gerçekleştirilen benzetim sonuçlarına göre, birinci ve ikinci senaryolar için uçtan uca gecikme değerleri göz ardı edilebilir seviyelerdedir. Hastalardan gelen verilerin sorunsuz bir biçimde alıcıya ulaştırılması için gerekli olan bant genişliği bütün senaryolar için toplamda 81,6 Kbps olarak hesaplanmıştır. Veri oranı açısından birinci ve ikinci senaryolardaki benzetim sonuçları ile hesaplanan veri oranı değerleri örtüşmektedir. Üçüncü senaryoda, N1 ve N3 düğümleri için gerçekleşen veri oranı değerleri hesaplanan veri oranı değerleri ile uyumlu olduğu, ancak N2 düğümü için (beklendiği üzere) veri oranı değerlerinin sağlanmadığı görülmüştür. GTS tahsisi için istekte bulunulmadığı zamanlarda, veri trafik yükünün yoğun olmadığı durumlar için CAP dilimlerinin veri oranı ve iletim gecikmeleri yönünden kullanılabilir olduğu görülmektedir. Veri trafik yükünün boyutlarının arttığı durumlar için ise, gecikmelerin aşırı arttığı görülmektedir. Dolayısıyla, trafik yoğunluğunun az olduğu düğümler için GTS tahsisine gerek olmadığına, yoğunluğun fazla olduğu durumlar için ise GTS tahsisine gerek olduğu sonucuna varılmıştır. RFID okuyucu mevcut PANc içerisine entegre edilerek hastalar üzerinde bulunduğu düşünülen etiketlerden kişisel bilgilerinin okuma işlemi başarı ile gerçekleştirilmiştir.

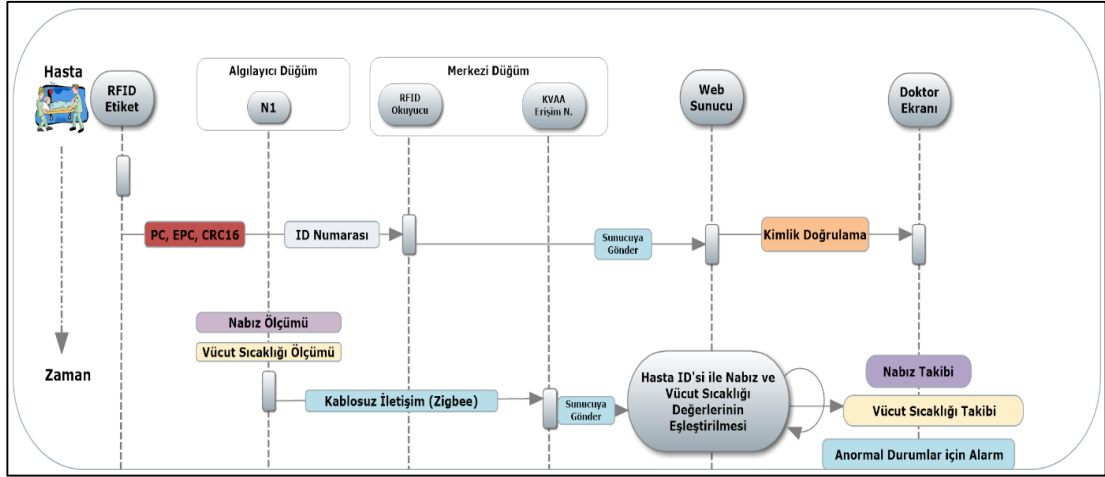
4. GELİŞTİRİLEN GERÇEK ZAMANLI HASTA İZLEME SİSTEMİ

Doktora tez çalışmasına yönelik olarak belirlenen senaryonun gerçek zamanlı uygulaması gerçekleştirilmiştir. İlgili sistemin blok şeması Şekil 4.1’de görülmektedir.



Şekil 4.1. Geliştirilen gerçek zamanlı hasta izleme sisteminin yapısı

Doktora tez çalışmasının gerçek zamanlı uygulaması Şekil 4.1’de görüldüğü gibi üç kısımdan oluşmaktadır. Bunlardan birincisi hastalardan fizyolojik verilerin toplanabilmesi için geliştirilen KVAA yapısının algılayıcı düğüm kısmıdır. İkinci kısım, algılayıcı düğümden gelen verileri toplayan KVAA’nın erişim noktası ile hastanın kişisel verileri elde etmek için geliştirilen RFID sistemin okuyucu kartının birleştiği koordinatör düğümdür. Üçüncü kısım ise koordinatör düğüm tarafından alınan tüm verilerin bilgisayar ortamına aktarılarak veri tabanına kaydedildiği, fizyolojik verilerin ve kişisel verilerin eşleştirildiği ve bu verilerin tamamının grafiklerle görselleştirildiği web tabanlı yazılım uygulama kısmıdır.



Şekil 4.2. Gerçekleştirilen gerçek zamanlı sistemin sıralı akış diyagramı

Gerçekleştirilen gerçek zamanlı sistemin çalışmasının sıralı akış diyagramı Şekil 4.2’de görülmektedir. Sistemin çalışması alt bölümlerde detaylı bir şekilde anlatılmaktadır.

Sistemin donanım kısmında Arduino Uno mikrodenetleyici geliştirme kartı, iki adet fizyolojik veri (nabız ölçer ve vücut sıcaklığı ölçer) algılayıcıları ve kablosuz iletişimi sağlamak için Zigbee haberleşme altyapısını kullanan XBee modülleri, bu modülleri Arduino kartına bağlamak için XBee kalkanları (shield), hastaların kişisel bilgilerine ulaşmak için gerekli olan RFID okuyucu ve etiketler kullanılmıştır. Tez çalışmasının bir sonraki kısmında uygulamanın donanım kısmındaki bileşenler hakkında bilgiler verilmektedir.

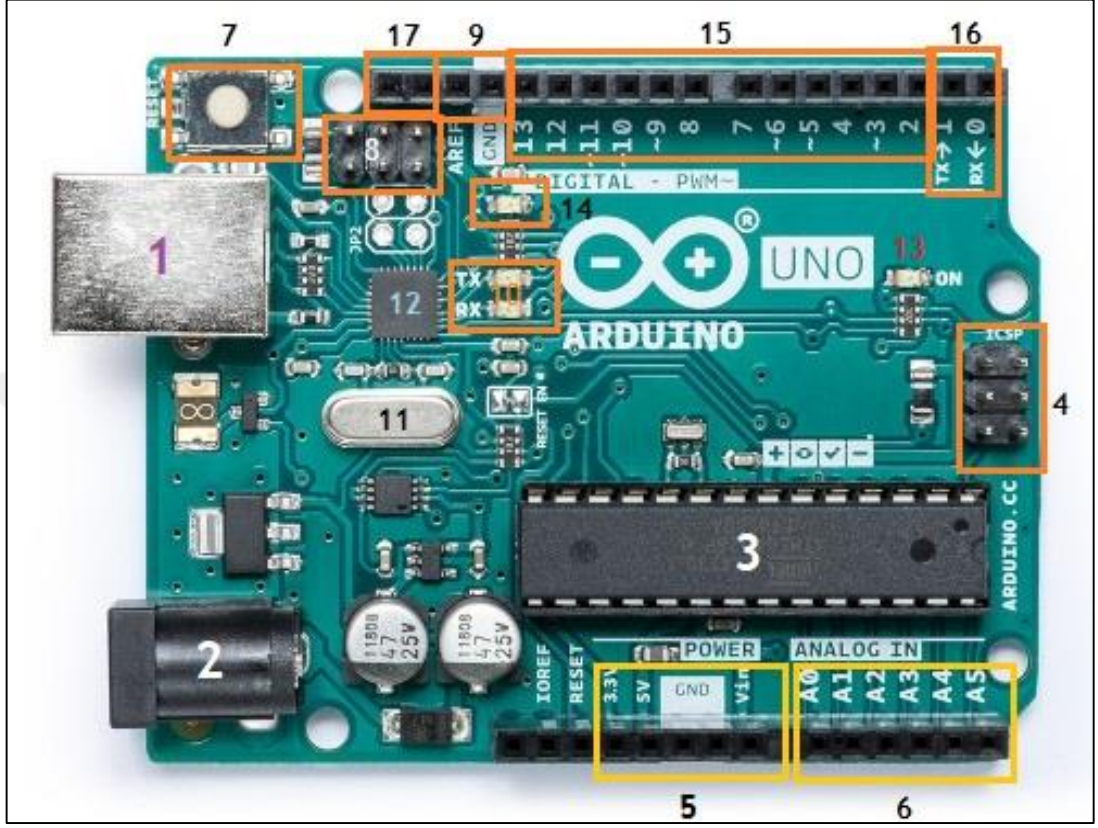
4.1. Donanım Bileşenleri

4.1.1. Mikrodenetleyici geliştirme kartı

Gerçek zamanlı sistemin tasarımında kullanılan Arduino Uno, Atmel firmasının ürettiği AVR mikrodenetleyicisi (ATmega328) içeren donanım ve yazılım tabanlı bir uygulama geliştirme kartıdır. Uno serisi Arduino kartlarının en yaygın kullanılan kartlarından biridir [49].

Arduino Uno kartının 14 tane dijital giriş / çıkış ucu bulunmaktadır. Bunlardan 6 tanesi darbe genişlik modülasyonu (Pulse Width Modulation - PWM) çıkışı olarak kullanılabilir. Ayrıca 6 adet analog girişi, bir adet 16 MHz kristal osilatörü, evrensel seri veri yolu (Universal Serial Bus - USB) bağlantısı, 2,1 mm’lik güç girişi, devre

üzerinde seri programlama başlığı (In-Circuit Serial Programming - ICSP) ve sıfırlama (reset) butonu bulunmaktadır. Şekil 4.3’de Arduino Uno R3 kartının üstten görünüşü ve uygulama kartı üzerindeki donanımların ve uçların yerleri görülmektedir [49, 50].



Şekil 4.3. Arduino Uno R3 üstten görünüşü

Arduino Uno bir USB kablosu ile bilgisayara bağlanarak ya da pil gibi harici bir güç kaynağı ile çalıştırılabilir. Adaptörün bağlantı ucunun merkezi pozitif olmalıdır ve Arduino Uno'nun güç girişine takılmalıdır. Pil veya batarya bağlanacaksa güç kabloları Gnd ve V_{in} uçlarına bağlanmalıdır. Tablo 4.1’de Arduino Uno kartı üzerindeki donanımların yerleri ve görevleri görülmektedir [49]. Tablo 4.2’de ise Arduino Uno kartının teknik özellikleri görülmektedir [50].

Tablo 4.1. Arduino Uno kartı üzerindeki donanımların yerleri ve görevleri [49]

1	USB Bağlantı Ucu
2	Harici Güç Giriş Ucu (7-12 V DC)
3	ATmega328 Mikrodenetleyicisi
4	ATMega328 ICSP (Harici Programlama Uçları)
5	Güç Uçları (Gnd, +5 V, +3,3 V)
6	Analog Giriş Uçları (6 Adet)
7	Sıfırlama (Reset) Butonu
8	USB için ICSP uçları
9	Aref ve Gnd (toprak) uçları
10	Tx-Rx Ledleri
11	16 Mhz Kristal
12	Haberleşme Entegresi
13	Güç ledi
14	Dahili led (13 nolu bacağına bağlı)
15	Dijital giriş/çıkış uçları (yanında ~ işareti olan uçlar PWM çıkışı olarak kullanılabilir.)
16	Tx-Rx (Gönderme-Alma) uçları
17	SCL/SDA (I ² C) haberleşme uçları

Tablo 4.2. Arduino Uno teknik özellikleri [50]

Mikrodenetleyici	ATmega328
Çalışma gerilimi	+5 V
Tavsiye edilen besleme gerilimi ve limitleri	7-12 V DC / 6-20 V (Limit)
Dijital giriş / çıkış pinleri	14 tane (6 tanesi PWM)
Analog giriş pinleri	6 tane
Giriş/çıkış pini başına düşen DC akım	40 mA
3,3 V pini için akım	50 mA
Flash hafıza	32 KB (0,5 KB bootloader)
Statik RAM - EEPROM	2 KB – 1 KB
Saat frekansı (Clock Speed)	16 MHz
Uzunluk – Genişlik – Ağırlık	68,6 mm – 53,4 mm – 25 g

Arduino Uno kartında bulunan 14 tane dijital giriş/çıkış uçlarının tamamı, `pinMode()`, `digitalWrite()` ve `digitalRead()` fonksiyonları ile giriş ya da çıkış olarak kullanılabilir. Bu uçlar mantıksal 5 V ile çalışır. Her uç maksimum 40 mA çekebilme ya da sağlayabilir. Bu uçlara bağlı 20–50 Kohm dahili pull-up dirençleri bulunmaktadır.

Arduino Uno kartı üzerinde bulunan bazı uçların birden fazla özel fonksiyonu bulunmaktadır;

- Serial 0 (Rx) ve 1 (Tx) : Bu uçlar seri veri alma (receive - Rx) ve gönderme (transmit - Tx) uçlarıdır.
- Harici kesmeler (2 ve 3) : Bu uçlar düşük seviyede, yükselen kenarda, düşen kenarda veya uçtaki değer değiştiğinde bir kesmeyi tetiklemek için kullanılabilir.
- PWM: 3, 5, 6, 9, 10 ve 11: Bu uçlar `analogWrite()` fonksiyonu ile 8-bit PWM çıkışı sağlamaktadır.
- SPI 10: (Slave Select - SS), komut alan cihazın seçim bağlantısı, 11: (Master Out Slave Input - MOSI), komut veren cihazın komut alan cihaza gönderdiği seri veri sinyali, 12: (Master Input Slave Output - MISO), komut alan cihazın komut veren cihaza gönderdiği seri veri sinyali, 13: (Serial Clock - SCK) komut veren cihazın ürettiği SPI iletişiminin hızını belirleyen saat frekansı. Bu uçlar, SPI kütüphanesi aracılığıyla SPI iletişimini desteklemektedir.
- LED 13: 13 nolu dijital uca bağlı bir leddir. Ucun değeri 'HIGH' olduğunda yanmakta, 'LOW' olduğunda ise sönmektedir.

Arduino Uno 'nun A0 dan A5 e kadar etiketlenmiş 10 bitlik çözünürlüğe sahip 6 adet analog girişi bulunmaktadır. Varsayılan ayarlar ile toprak seviyesi ile 5 V arası gerilim değerlerini ölçebilirler. Ancak, `AREF` ucu ve `analogReference()` fonksiyonu kullanılarak üst limit değiştirmek mümkün olabilmektedir.

Arduino Uno içerisindeki ATmega328 mikrodenetleyicisi, Rx ve Tx uçlarından erişilebilen evrensel asenkron alıcı gönderici (Universal Asenkron Receiver Transmitter - UART) transistörden transistör lojiğe (Transistor to Transistor Logic - TTL) 5V seri haberleşmeyi desteklemektedir. Kart üzerindeki ATmega16U2 entegresi USB üzerinden seri haberleşmeyi sağlar ve cihaz bilgisayar üzerinde sanal bir

'Comport' olarak görünür. ATmega16U2 entegresi standart USB sürücülerini kullanmaktadır ve harici sürücü gerektirmemektedir. Ancak, Windows platformunda çalışması için '.inf' uzantılı bir dosya gerekmektedir. Kart üzerindeki Rx ve Tx ledleri cihazlar arası haberleşme olduğunda yanıp sönerler. Arduino içerisine yüklenecek 'SoftwareSerial' kütüphanesi ile Arduino Uno'nun sayısal uçlarının herhangi biri üzerinden seri haberleşme gerçekleştirilebilmektedir. Ayrıca ATmega328, entegre arası cihaz (Inter Integrated Circuit - I²C) ve seri çevresel arayüz (Serial Peripheral Interface - SPI) haberleşmelerini de desteklemektedir.

4.1.2. Kablosuz haberleşme modülü

Tez çalışmasında belirlenen senaryoya uygun olarak gerçek zamanlı olarak geliştirilen sistemin KVAA kısmında Zigbee haberleşme altyapısını kullanan XBee haberleşme modülleri kullanılmıştır. XBee modülleri, Digi firması tarafından geliştirilmiş 2,4 GHz frekansında çalışan 802.15.4, ZigBee örgü (mesh) ve çoklu nokta (Multi-Point) haberleşme protokolüne uygun kablosuz haberleşme modülleridir. Gelişmiş tasarımı ve kolay kullanımı sayesinde bilgisayar ve mikrodenetleyicili sistemlerle güvenilir ve kolay bir seri haberleşme yapmaktadır. Noktadan noktaya (Point to Point) ve bir noktadan çok noktalı ağlar (Point to Multi-Point) içerisinde haberleşmeyi desteklemektedir.

XBee modüllerin çıkış güçleri 1mW'tan 315mW'a kadar ayarlanabilmekte olup, üzerindeki özel anten bağlantısıyla yönlü dipol anten bağlanarak 40 Km'ye (40000 metre) kadar haberleşmesi sağlanabilmektedir. Şekil 4.4'de XBee PRO S1 modülü görülmektedir [51].



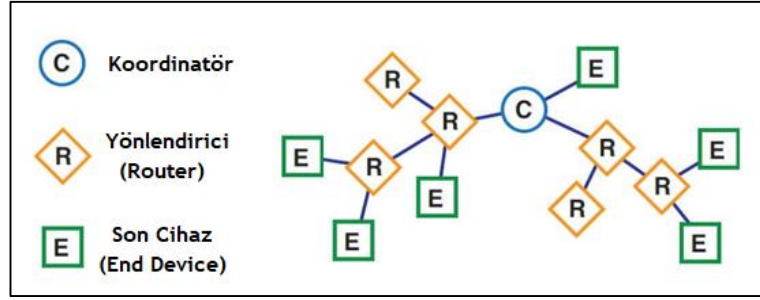
Şekil 4.4. XBee PRO S1 modülü

Tez çalışmasının KVAA kısmının algılayıcı düğüm tarafında XBee S1, erişim noktası tarafında ise XBee PRO S1 modülü kullanılmıştır. Bu modüller arasındaki farklar Tablo 4.3'te görülmektedir.

Tablo 4.3. XBee S1 ve XBee Pro S1 modüllerinin özellikleri [51]

Özellikler	Legacy Digi XBee® S1 802.15.4	Legacy Digi XBee- PRO® S1 802.15.4
RF Veri Oranı	250 Kbps	250 Kbps
Kapalı/Şehirsal Alan Görüş Açısı Mesafesi	100 ft (30 m)	300 ft (100 m)
Açık Alan Görüş Açısı (Line of Sight) Mesafesi	300 ft (100 m)	1 mil (1,6 Km)
Gönderim Gücü	1 mW (+0 dBm)	60 mW (+18 dBm)
Alıcı Hassasiyeti (1% Her)	-92 dBm	-100 dBm
Digi Donanım İsmi	S1	PRO S1
Alın-Gönderici Entegresi	Freescale MC13212	Freescale MC13212
Seri Veri Arayüzü	3.3V CMOS UART	3.3V CMOS UART
Dijital Giriş/Çıkış	8	8
Şifreleme	128-bit AES	128-bit AES
ADC Girişleri	10-bitlik 6 Adet	10-bitlik 6 Adet
Besleme Gerilimi	2,8-3,4 V DC	2,8-3,4 V DC
Gönderim Akımı	45 mA-3,3 V DC	215 mA-3,3 V DC
Alım (Receive) Akımı	50 mA-3,3 V DC	55 mA-3,3 V DC
Düşük Güç Akımı	< 10 µA 25°C	< 10 µA 25°C

Modüllerin kullanılabilmesi için en az 2 adet XBee modüle ihtiyaç vardır. XBee modülleri Zigbee ağ altyapısını kullanmaktadır. Basit bir Zigbee ağ yapısı Şekil 4.5'te görülmektedir [52, 65].



Şekil 4.5. Basit bir Zigbee ağ yapısı

XBee modüllerinde tüm haberleşme işlemleri bir kişisel alan ağı (PAN) üzerinden yapılmaktadır. Haberleşmenin gerçekleşebilmesi için kesinlikle bir PAN yapısı oluşturulmalıdır. Bir PAN ancak bir koordinatör düğüm etrafında oluşturabilmektedir. Bir PAN içinde bulunan tüm cihazların 'PAN ID' parametresi aynı olmak zorundadır. Erişim noktası ("coordinator") tarafından oluşturulan PAN'a katılmak isteyen tüm cihazların 'PAN ID' parametreleri mutlaka 'koordinatör' cihazın 'PAN ID'si ile aynı olmalıdır. Aksi durumda cihaz ağa kabul edilmemektedir.

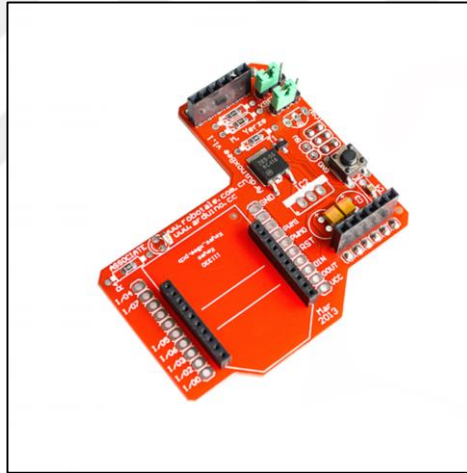
Şekil 4.5'de görülen ağ yapısına göre cihazların özellikleri şu şekildedir; 'C', erişim noktasını (Coordinator) ifade etmektedir ve haberleşme kanalını ve 'PAN ID' parametresini belirlemektedir. Ağın başlatılmasından sorumlu olan 'C' PAN'ı başlattıktan sonra, 'E' (Son cihaz) ve 'R'lerin (yönlendiricilerin) PAN'a katılmasına izin verebilir. 'C', veri alıp gönderebilir, ya da örgü ağ (mesh network) yapısında verinin bir sonraki cihaza iletilmesi işlemlerini gerçekleştirebilir. 'C', tüm ağı yönettiğinden pil ile beslenmesi ve/veya düşük güç modlarında çalıştırılması uygun olmamaktadır. 'C' sabit bir güç kaynağı ile beslenmelidir ve standart güç modunda (main power mode) çalıştırılmalıdır. 'C'nin herhangi bir şekilde işlevini kaybetmesi, tüm ağın işlevsiz hale gelmesine neden olmaktadır [65].

'R', yönlendiriciyi (Router) ifade etmektedir. Bir 'R'nin çalışabilmesi için öncelikle bir PAN'a katılması gerekir. PAN'a katıldıktan sonra 'R', diğer 'R'lerin ve 'E'lerin PAN'a katılmasına izin verebilmektedir. 'R', veri alıp gönderebilir ya da örgü ağ (mesh network) üzerinde veri aktarımını sağlayabilir. 'R'ler diğer cihazların PAN'a katılmasına izin verdikleri için 'R'lerin düşük güç modlarında çalıştırılmaları uygun olmamaktadır.

'E', son cihazı (End Device) temsil etmektedir. 'R' gibi 'E'de çalışabilmek için öncelikle bir PAN'a dahil olmalıdır. 'E'ler diğer cihazların PAN'a katılmasına izin veremezler, ancak ağ üzerinde veri iletimi yapabilirler [65].

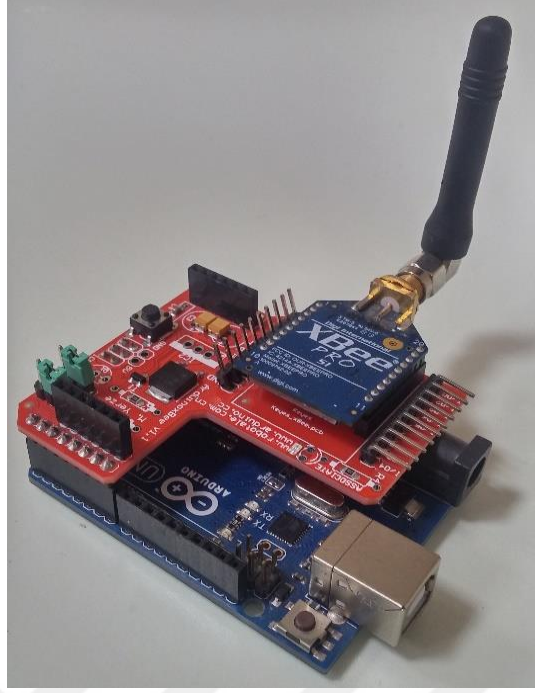
4.1.3. Kablosuz modülleri geliştirme kartına bağlama cihazları

XBee modüllerini Arduino karta bağlayabilmek için Arduino uyumlu XBee kalkanları (shield) kullanılmıştır. XBee shield ürünü, XBee modüllerin Arduino ile beraber çalışmasını yani arayüzlendirme işlevini basit bir hale getirmektedir. XBee shield, düşük maliyetli, düşük güçlü kablosuz algılayıcı ağlarının özel ihtiyaçlarını karşılamak için tasarlanmış uyumlu bir çözüm olarak gösterilebilmektedir. Modülün kullanımı kolaydır ve düşük güç tüketimi ile cihazlar arasında kritik verilerin güvenli aktarımını sağlayabilmektedir. Şekil 4.6'da XBee shield ürünü görülmektedir.



Şekil 4.6. XBee shield ürünü

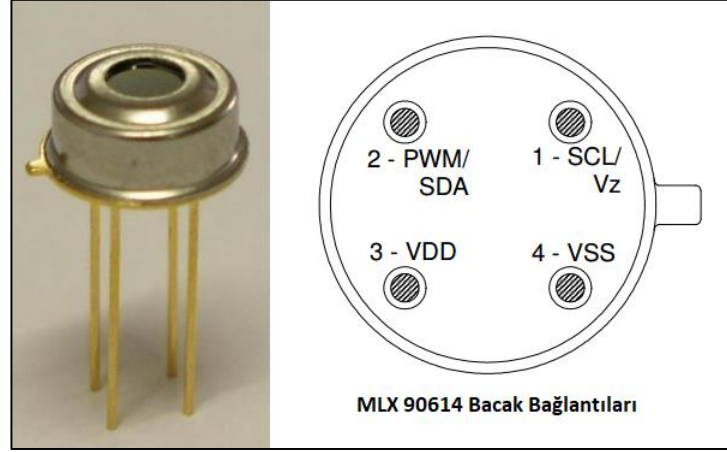
XBee shield, S1 (Series 1), S2 (Series 2 ve 2,5), standart ve Pro sürümleri ile birlikte tüm XBee modülleri ile sorunsuz bir şekilde çalışmaktadır. XBee shield ürünü, Arduino Uno, Arduino Mega, Freeduino, Seeduino vb. gibi birçok geliştirme kartıyla uyumludur. Şekil 4.7'de tez çalışmasında erişim noktası olarak kullanılan Arduino Uno, XBee shield ve XBee Pro S1 modülü görülmektedir.



Şekil 4.7. KVAA erişim noktası (Arduino, XBee shield ve XBee Pro S1 modülü)

4.1.4. Vücut sıcaklığı algılayıcısı

Tez çalışmasının senaryosuna uygun olarak hasta üzerinden vücut sıcaklığı verisi toplanmıştır. Sıcaklık verisinin toplanması için Melexis firmasının ürettiği MLX90614 kızılötesi sıcaklık algılayıcısı kullanılmıştır. Bu algılayıcı, sıcaklığına göre cisimlerin yüzeyinden yansıyan kızılötesi radyasyona (Infrared Radiation - IR) bağlı olarak temassız bir şekilde sıcaklıkları ölçmektedir. Belli bir mesafeden algılayıcı doğrultusunda bulunan cisimlerin sıcaklıklarını ölçmekle birlikte, aynı zamanda ortam sıcaklığını da ölçebilmektedirler. MLX90614 sıcaklık ölçer algılayıcı ve bacak bağlantıları Şekil 4.8’de görülmektedir [53].



Şekil 4.8. MLX 90614 algılayıcısı bacak bağlantıları

Algılayıcı, termofil dedektör ve sinyal uyarlamalı uygulamaya özgü bir standart üründen (Application Specific Standart Product - ASSP) meydana gelmektedir. Algılayıcı 90° lik görme açısına (Field of View - FOV) sahiptir ve herhangi bir nesnenin ve ortamın sıcaklığını teknik olarak 0,01°C hassasiyette ölçmektedir. Ancak ortam ve cisim sıcaklıklarının birbirine etkisinden dolayı gerçek ortam ve cisim sıcaklığının ürün rehberinde yazan formüller ile yeniden hesaplanması gerekmektedir. Bu hesaplamalara göre optimum ölçüm sıcaklığı hassasiyeti 0,5°C olarak söylenebilmektedir. Penceresinde bulunan optik filtre ile kızılötesi ışınımın 5,5–14 µm aralığını geçirmektedir. Bu algılayıcı -40°C/+125°C ortam sıcaklıklarını ve -70°C/+380°C cisim sıcaklıklarını ölçebilmektedir. MLX90614 algılayıcısı, iki ayrı sıcaklık ölçme elemanından oluşmaktadır. Birincisi metal kılıf üzerinde bulunan sıcaklık algılayıcısı (Proportional to Absolute Temperature Sensor - PTAT veya Positive Temperature Coefficient Sensor - PTC) ve kızılötesi algılayıcısıdır [54]. Algılayıcı 17-Bit çözünürlüğünde bir ADC ile IR detektör ile analog sıcaklık bilgisini elde etmektedir ve bu elde edilen bilgiyi dijital sinyal işleyici (Digital Signal Processor - DSP) ile hesaplamaktadır. Algılayıcı, I²C haberleşmenin bir versiyonu olan sistem yönetim veriyolu (System Management Bus - SMBus) iletişim protokolü ile haberleşmektedir [53].

MLX90614 algılayıcısının termofil çıkışı şu şekilde hesaplanmaktadır;

$$V_0 = (T_a, T_0) = A \times (T_0^4 - T_a^4) \quad (4.1)$$

Bu hesaplamada T_0 ; Kelvin cinsinden nesne sıcaklığını, T_a ; Kelvin cinsinden metal kılıf sıcaklığını, A ise hassasiyeti göstermektedir. Algılayıcı içerisindeki DSP gerekli hesaplamaları yaptıktan sonra hesaplanan ortam ve nesne sıcaklığını sayısal olarak çıkışa göndermektedir. Hesaplanan ham değerler;

$$T [^{\circ}\text{C}] = T_{\text{raw}} \times 0,02 - 273,15 \quad (4.2)$$

Formülü yardımı ile 0,02 hassasiyetinde derece cinsinden sıcaklık değerine dönüştürülmektedir [54].

MLX90614 algılayıcısının çalışma voltajları modeline göre farklılık göstermektedir (3,3–3,6 V). Bu algılayıcıları Arduino gibi geliştirme kartlarıyla daha kolay kullanabilmek için GY906 modülü geliştirilmiştir. Tez çalışmasının uygulama kısmında da bu modül kullanılmıştır. Şekil 4.9’da GY906 modülü görülmektedir.



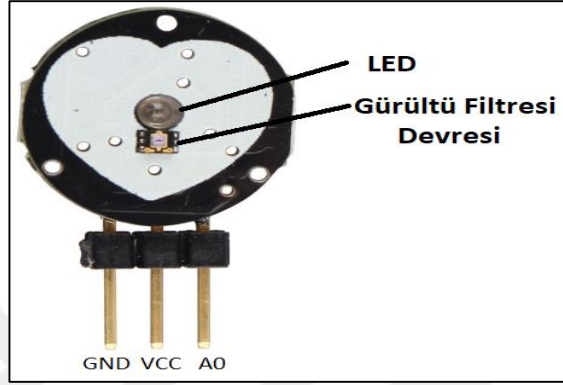
Şekil 4.9. GY906 modülü

Şekil 4.9’da görüldüğü gibi modül üzerinde MLX90614 algılayıcısı ile birlikte bir 3,3 V voltaj ayarlayıcısı (regülatör) bulunmaktadır. Bu ayarlayıcı ile algılayıcı hem 3,3 V destekli hem de 5 V destekli mikrodenetleyici kartlar ile kolaylıkla çalışabilmektedir.

4.1.5. Nabız ölçer algılayıcı

Hastane veya uzak sağlık hizmetlerinde hastalardan takip edilmesi gereken en önemli parametrelerden birisi de kalp atım oranı yani nabız bilgileridir. Kalp atım oranı bilgisi EKG verisi üzerinden tepe noktaları belirlenerek hesaplanabileceği gibi hastaların parmak uçlarına hariçten (non-invasive) bağlanabilen algılayıcılarla da alınabilir. Bu amaçla tez çalışmasının senaryosuna uygun olarak hastalardan toplanan diğer

fizyolojik veri dakikadaki kalp atım oranı bilgisidir. Özellikle bebeklerdeki kalp atım düzensizlikleri, bebeklerin ani ölümlerine yol açabilmektedir [55]. Beklenmedik acil durumlarda hastaların nabız ölçümlerinin düzenli olarak yapılması gerekmektedir. Tez çalışmasında nabız verisinin toplanması için World Famous Electronics llc. firmasının ürettiği ‘Pulse sensor’ ürünü kullanılmıştır [56]. Şekil 4.10’da nabız ölçümünde kullanılan algılayıcı görülmektedir [57].



Şekil 4.10. “Pulse-sensor” nabız algılayıcısı ve bacak bağlantıları

Bu algılayıcı özellikle Arduino kartları için geliştirilmiş tak-çalıştır kalp atış oranı algılayıcısıdır. Parmak ucundan veya kulağa sabitlenebilen aparatla kolaylıkla ölçüm yapabilmektedir. Algılayıcı kılcal (kapiller) kan damarlarının genişlemesinden dolayı meydana gelen ışıktaki değişime göre ölçüm işlemini gerçekleştirmektedir. Şekil 4.10’da görüldüğü gibi modülün merkezinde kalp atımını tespit etmeye yardımcı olan bir led bulunmaktadır. Led’in altında, gürültülerin okumaları etkilemesini önlemek için bir gürültü giderme devresi bulunmaktadır [57].

Algılayıcının temel çalışma prensibi şu şekildedir; Bir kalp atışı oluştuğunda, insan vücudu üzerinden kan pompalanır ve bu kan kılcal dokulara sıkışır. Bu kılcal dokuların hacmi, kalp atışının sonucunda artar. Ancak kalp atımları arasında (iki ardışık kalp atışı arasındaki süre), kılcal dokuların içindeki bu hacim azalır. Kalp atışları arasındaki kılcal dokulardaki hacimdeki bu değişim, bu dokularda iletilecek ışık miktarını etkiler. Algılayıcı modülün üzerinde nabız hızını ölçmeye yardımcı olan bir ışık (led) vardır. Nabız algılayıcı parmağa yerleştirildiğinde, yansıyan ışık kılcal kan damarlarının içindeki kan hacmine göre değişmektedir. Bir kalp atışı sırasında, kılcal kan damarlarının içindeki hacim yükselmektedir. Bu durum, ışığın yansımaları

etkilemekte ve kalp atışı sırasında yansıyan ışık, kalp atışının olmadığı (kalp atımının olmadığı zaman periyodu veya kalp atışları arasında geçen süre boyunca, kılcal damarların içindeki hacim daha az olmakta ve bu durum ışığın daha yüksek yansımaya yol açmaktadır) zamana göre daha az olmaktadır. Işık iletimi ve yansımadaki bu değişim, nabız algılayıcısının çıkışından bir darbe olarak elde edilebilmektedir. Bu bilgi daha sonra kalp atımını ölçmek ve ardından kalp atışı sayısı belirlemek üzere kodlanarak programlanabilir [57].

4.1.6. RFID okuyucu kartı

Tez çalışmasının gerçek zamanlı uygulamasındaki senaryoya uygun olarak hastaların kişisel bilgilerini toplayabilmek ve uzak sunucu tarafında fizyolojik veriler ile eşleştirebilmek amacıyla RFID sistem, koordinatör düğüm üzerinde KVAA'nın erişim noktasıyla birleştirilmiştir. RFID okuyucu olarak MFRC522 RFID okuyucu kartı kullanılmıştır.

MF-RC522 RFID okuyucu kartı, yakın alan haberleşme (Near Field Communication - NFC) frekansı olan 13,56 MHz frekansında çalışan etiketler üzerinde okuma ve yazma işlemini yapabilen, düşük güç tüketimli, ufak boyutlu bir karttır [58].

Arduino başta olmak üzere birçok mikrodenetleyici içeren geliştirme kartı platformu ile beraber rahatlıkla kullanılabilir. 424 kbit/s haberleşme hızında iletişim yapabilirler. RFID üzerinde farklı şifreleme türlerini desteklemektedir. Mifare1 S50, mifare1 S70 mifare ultralight, mifare pro ve mifare desfire gibi kart türlerini desteklemektedir. Şekil 4.11'de tez çalışmasında kullanılan MFRC522 RFID okuyucu kartı ve etiketler görülmektedir.



Şekil 4.11. MFRC522 RFID okuyucu kartı ve etiketler

Seri haberleşme (SPI, I²C) kullanım kolaylığı sağlamaktadır. MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 ve MIFARE Plus RF tanıma (identification) protokollerinin tümünü desteklemektedir. MFRC522 RFID kartı, okuma/yazma (read/write) modunda 40 mm'ye kadar desteklemektedir. MFRC522 RFID modülünün kullanılabileceği çeşitli uygulamalar bulunmaktadır. Bunlar; erişim yönetimi, ürün izleme, hayvan veya insan takibi, otomatik geçiş sistemleri ve temassız ödeme, havalimanı bagaj takip ve bagaj lojistiği, kişisel kimlikler, depo stok takibi gibi sistemlerdir [59]. Tablo 4.4'te MFRC522 kartının özellikleri görülmektedir.

Tablo 4.4. MFRC522 kartı özellikleri [58]

Çalışma Gerilimi	2,5–3,3 V
Çalışma Frekansı	13,56 MHz
Çalışma Akımı	13–26 mA
Uyku Akımı	< 80 µA
Haberleşme Protokolü	SPI, I ² C arayüzü
Desteklenen Kartlar	mifare1 S50, mifare1 S70, mifare ultralight, mifare pro ve mifare desfire
Kart Boyutları	40×60 mm

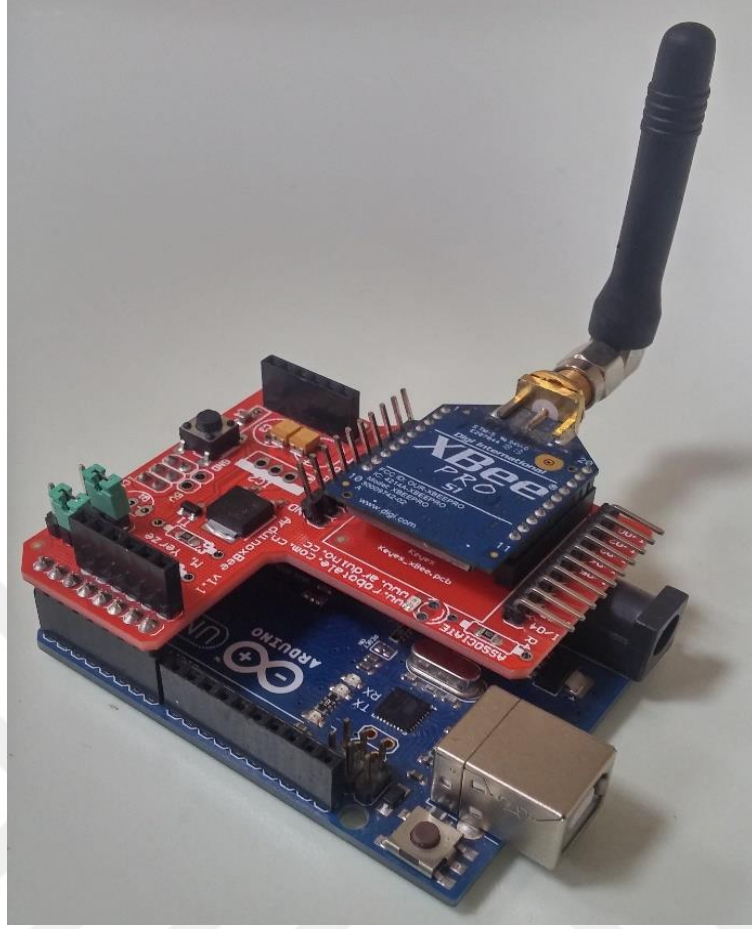
4.2. Geliştirilen Gerçek Zamanlı Platformun Bölümleri

Tez çalışmasının bu kısmında gerçekleştirilen sistemin kısımları ve çalışmaları hakkında detaylı bilgiler bulunmaktadır.

4.2.1. KVAA algılayıcı düğümü

Tez çalışmasının gerçek zamanlı uygulamasında belirlenen senaryoya uygun olarak insan vücudundan fizyolojik veriler oluşturulan KVAA kullanılarak toplanmaktadır. Gerçek zamanlı uygulamada EKG verisi yerine benzetim çalışmasından farklı olarak nabız bilgisi toplanmıştır. Gerçek zamanlı EKG verilerinin toplanmasının zor olması, EKG verilerinin şebeke ve çalışma elemanlarının gürültülerinden etkilenmesi ve ufak müdahalelerde bile bozulmasından dolayı veri toplanması daha kolay olan ve hastanın kalp atım oranıyla ilgili kesin bilgi veren nabız bilgisi ölçülerek QoS kriterlerine göre verilerin uygulama kısmına iletilip iletilmediği bilgileri incelenmiştir. Nabız bilgisi ile birlikte vücut sıcaklığı bilgisi hasta üzerinden toplanarak kablosuz ortamdan erişim noktasına gönderilmiştir.

Geliştirilen gerçek zamanlı sistemin KVAA ve RFID sistemleri, Arduino Uno, Arduino uyumlu algılayıcılar ve kartlar ile gerçekleştirilmiştir. KVAA yapısının algılayıcı düğümü, iki adet algılayıcıdan veri alıp bunları kablosuz ortamda erişim noktasına gönderebilecek şekilde tasarlanmıştır. Bu tasarımda, algılayıcılar olarak MLX90614 kızılötesi sıcaklık ölçer algılayıcı ve 'Pulse sensor' nabız ölçer algılayıcısı kullanılmıştır. Verileri kablosuz olarak gönderebilmek için Zigbee haberleşme altyapısını (IEEE 802.15.4) kullanan XBee S1 modülü kullanılmıştır. XBee modülleri Arduino uyumlu XBee kalkan (shield) modülleri kullanılarak Arduino geliştirme kartına bağlanmıştır. Geliştirilen sistemdeki temel algılayıcı ve koordinatör düğüm yapısı Şekil 4.12'de görülmektedir.



Şekil 4.12. Temel algılayıcı ve koordinatör düğüm yapısı

Vücut sıcaklığını ölçmek için kullanılan MLX90614 sıcaklık algılayıcısı, GY-906 modülü üzerinden Arduino kartına bağlanmıştır. MLX90614 algılayıcısı, bir seri haberleşme protokolü olan I²C (SMBUS) protokolü ile mikrodenetleyiciler ile haberleşebilmektedir. I²C protokolünde ‘Master-Slave’ şeklinde haberleşme yapılmaktadır. Haberleşme için iki hat bulunmaktadır. Bunlar; seri veri hattı (SDA) ve seri saat sinyal (SCL) hattıdır. SCL hattı mikrodenetleyici tarafından çeşitli frekanslara göre ayarlanır ve cihazların senkronizasyonu için kullanılır. SCL hattındaki sinyallerin yükselen kenar ve düşen kenar durumlarına göre SDA hattına da bakılarak veri iletimi gerçekleştirilmektedir [60]. Tasarlanan sistemde Arduino kartı ‘master’, algılayıcı ise ‘slave’ modundadır. Bu durumda iletişimin gerçekleşmesi için gerekli olan ‘START’, ‘STOP’ durumlarını ve iletişim için gerekli olan saat sinyallerini Arduino üretmektedir. Slave modundaki algılayıcı ölçtüğü sıcaklık bilgisini seri veri hattından (SDA) çok hızlı bir şekilde değerlendirmek üzere mikrodenetleyiciye göndermektedir. Kullanılan algılayıcının en büyük avantajı kızılötesi algılayıcısı sayesinde meydana

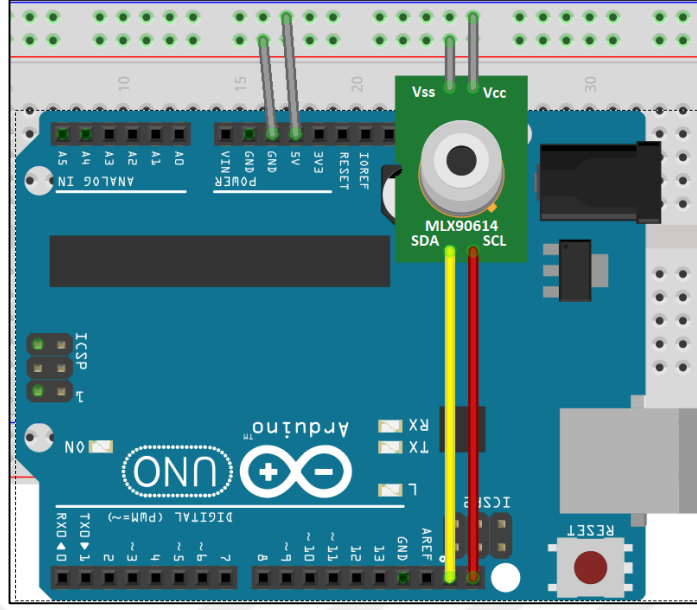
gelen sıcaklık deęişimlerini çok hızlı bir şekilde algılaması ve mikrodenetleyiciye sayısal veri hattından göndermesidir.

Klasik analog sıcaklık algılayıcılar meydana gelen sıcaklık deęişikliklerini sayısal algılayıcılara göre daha yavaş algılamaktadırlar. Ayrıca, analog kanallar üzerinden mikrodenetleyiciye gönderilen ham sıcaklık deęerleri çeşitli formüller ile gerçek sıcaklık deęerine çevrilmektedir. Bu durum mikrodenetleyiciye ek hesaplama yükü getirmektedir. Mikrodenetleyici içerisindeki ADC modülünün çözünürlüğüne göre ölçülen ham deęerin sayısal karşılığı deęişmektedir. 8 bitlik çözünürlüğe sahip bir ADC modülü olan mikrodenetleyici de analog veriler, $2^8=256$ farklı deęer alabilmekteyken 10 bitlik çözünürlüğe sahip mikrodenetleyici de $2^{10}=1024$ farklı deęer alabilmektedir. ADC biriminin çözünürlük deęerine göre 0 ile 5 V arasındaki analog gerilim deęerlere karşılık gelen ikili sayı ve onlu sayı karşılıkları Tablo 4.5'te görülmektedir.

Tablo 4.5. Çözünürlük deęerlerine göre analog deęerlerin ikili ve onlu karşılıkları

Analog Deęer	Çözünürlük			
	8 Bit		10 Bit	
	İkili (Binary)	Onlu (Decimal)	İkili (Binary)	Onlu (Decimal)
0 V	00000000	0	0000000000	0
1,25 V	01000000	64	0100000000	256
2,5 V	10000000	128	1000000000	512
5 V	11111111	255	1111111111	1023

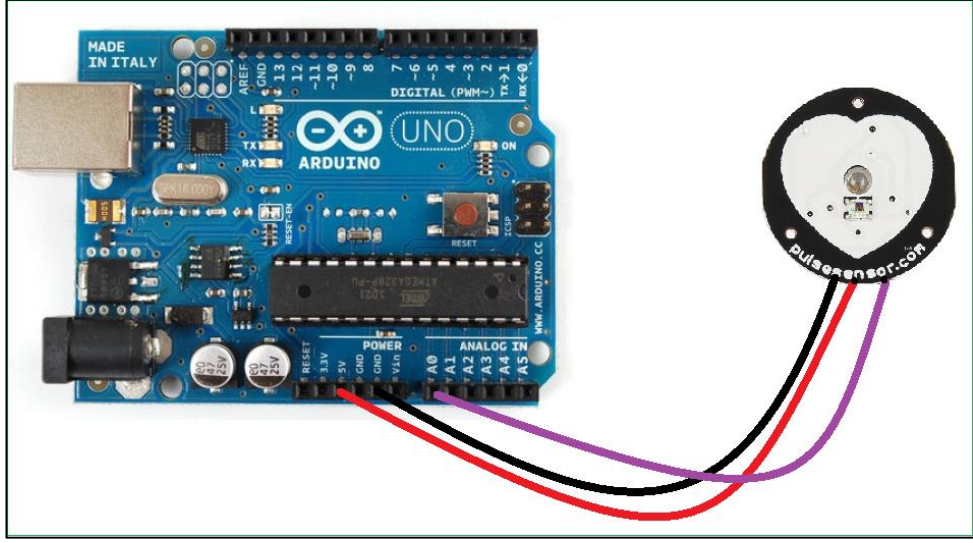
Tablo 4.5'te görüldüğü gibi, ADC'nin çözünürlük deęerlerine göre aynı gerilim deęerleri farklı sayısal deęerlere karşılık gelmektedir. Bu durum, ölçülen ham deęer kullanılarak hesaplanan gerçek sıcaklık deęerinin bulunmasını zorlaştırmaktadır. Farklı mikrodenetleyici özellikleri için farklı hesaplama formülleri kullanılması gerekliliği doğmaktadır. Hem deęişen sıcaklık deęerlerinin ölçümünün yavaşlığı hem de yukarıda belirtilen zorluklardan dolayı analog sıcaklık ölçer algılayıcı kullanılmamıştır. Literatürde yapılan çalışmalarda tez çalışmasında kullanılan algılayıcının sıcaklık deęişimlerine dięer analog algılayıcılara göre çok daha hızlı tepki vererek ölçüm yaptığı tespit edilmiştir [54]. MLX90614 algılayıcısının Arduino kartına bağlantısı Şekil 4.13'de görüldüğü gibi yapılmıştır.



Şekil 4.13. MLX90614 Arduino bağlantısı

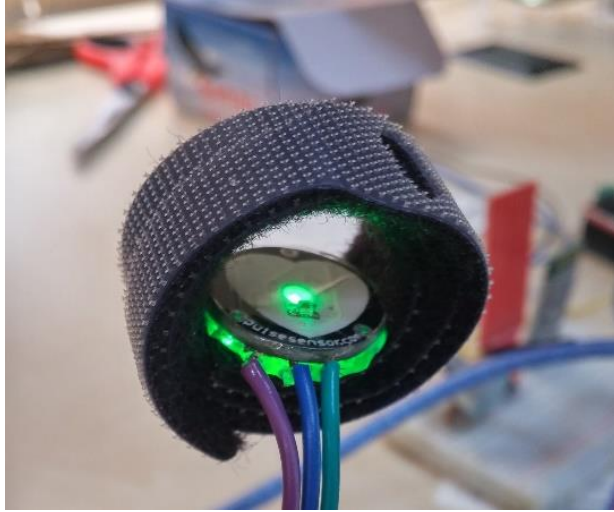
Şekil 4.13’de görüldüğü gibi algılayıcının seri veri hattı (SDA) ucu Arduino üzerindeki SDA ucuna, seri saat sinyali ucu da SCL ucuna bağlanmıştır. Algılayıcı çalışması için gerekli enerjiyi Arduino geliştirme kartı üzerinden almaktadır.

Hasta üzerinden alınan diğer bir fizyolojik veri, kalp atım oranıdır (nabız). Nabız bilgisini almak için ‘Pulse sensor’ nabız ölçer algılayıcısı kullanılmıştır. Nabız ölçer algılayıcı üzerinde bulunan led ve gürültü önleyici devre yardımıyla analog olarak nabız bilgisini ölçüp Arduino kartına analog giriş uçlarından göndermektedir. Şekil 4.14’de nabız ölçer algılayıcının Arduino geliştirme kartına bağlantıları görülmektedir [61].



Şekil 4.14. Nabız ölçer algılayıcı Arduino bağlantısı

Ölçüm sonuçlarının dış etkenlerden daha az etkilenmesi için algılayıcı üzerine şeffaf bir di-elektrik malzeme kaplanmıştır. Yine algılayıcı üzerindeki diğer devre yapılarını elektriksel gürültülerden korumak için algılayıcının altına silikon malzeme kaplaması yapılmıştır. Nabız ölçümleri parmaktan gelen bilgiye göre yapılacağı için algılayıcının etrafına parmağın yerleşebileceği ergonomik bir yuva yapılmıştır. Şekil 4.15'de algılayıcı düğüm ve tasarlanan algılayıcı ölçüm yuvası, Şekil 4.16'da ise nabız bilgisinin parmaktan alınması görülmektedir.

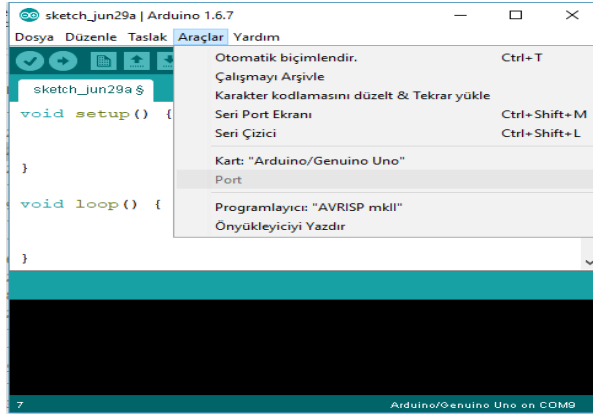


Şekil 4.15. Nabız ölçer ve ölçüm yuvası



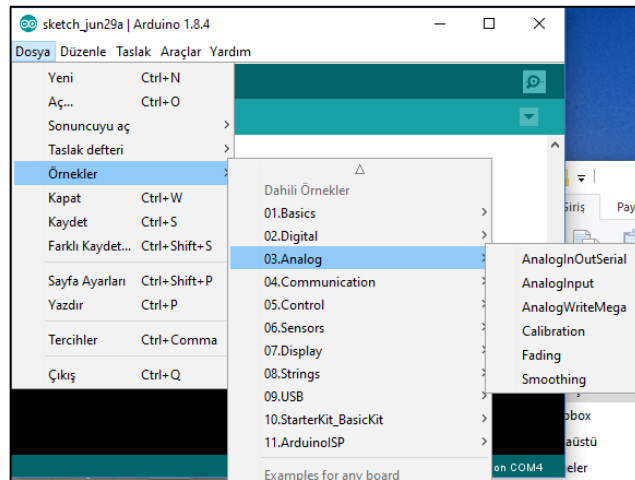
Şekil 4.16. Nabız bilgisinin alınması

Sıcaklık ve nabız ölçer algılayıcılar Arduino geliştirme kartına yukarıda belirtilen şekilde bağlanmışlardır. Bilgilerin doğru bir şekilde alınıp aktarılabilmesi için Arduino içerisindeki ATmega328 mikrodenetleyicisi programlanmıştır. Mikrodenetleyiciyi programlamak için Arduino'nun kendi derleyicisi olan Arduino IDE programı kullanılmıştır. Arduino IDE, Arduino geliştirme kartları için geliştirilen; komut yazılmasına, derleme işlemi yapılmasına ve son olarak derlenen kodların doğrudan (bilgisayarın USB portuna bağlı olan) Arduino kartlarına yüklenmesine olanak sağlayan bir yazılım geliştirme platformudur [62]. IDE programı Arduino'nun resmi sitesinden ücretsiz olarak indirilebilmektedir. Şekil 4.17'de Arduino IDE programı örnek ekranı görülmektedir.



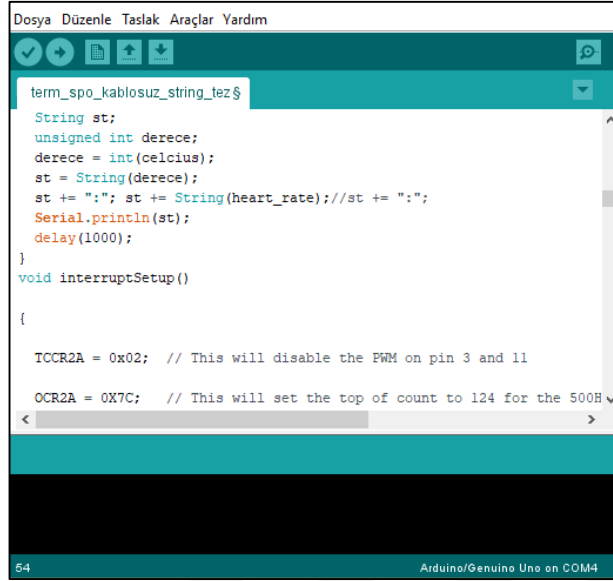
Şekil 4.17. Arduino IDE ekranı

Şekil 4.17’de görüldüğü gibi IDE programı ilk açıldığında ‘sketch’ ile başlayan bir dosya ismi almaktadır. Bu dosya ismi istenilen şekilde değiştirilebilmektedir. Değiştirilen isime göre çalışma dosyası kaydedilmek istendiğinde, dosya ismiyle aynı isimde otomatik olarak bir klasör oluşturularak dosya istenilen yere kaydedilmektedir. IDE programı temel olarak iki kısımdan oluşmaktadır. Bunlardan ilki olan ‘setup’ kısmı, Arduino kartına ilk defa enerji verildiğinde veya sıfırlandığında (reset) sadece bir defa çalışacak kodların bulunduğu kısımdır. Genelde tüm tanımlamalar bu kısımda yapılmaktadır. Enerji kesilene kadar başa dönerek çalışacak kodların olduğu kısım ise ‘loop’ kısmıdır. IDE programında kodlar C programlama dili tabanlı olarak yazılmaktadır. IDE programı içerisine Arduino ile uyumlu çalışan birçok modülün kütüphanesi eklenebilmektedir ve bu sayede modüller istenildiği şekilde kullanılabilir. Arduino geliştirme kartları USB üzerinden bilgisayara bağlandığında, işletim sistemi tarafından takılan kart için sanal bir seri port atanmaktadır. Bilgisayar ile Arduino kartı bu sanal seri port üzerinden haberleşmektedir. Bu portlar ‘COM’ şeklinde ön isimlendirmenin ardından numarası şeklinde isimlendirilmektedir (örneğin, COM5). Araçlar menüsü içerisinde Arduino kartının bağlı olduğu seri port ayarları, hangi Arduino kartına yükleme yapılacağı, seri port çıkışının izlenmesi gibi ayarlar ve seçenekler bulunmaktadır. Yine IDE programı içerisinde birçok amaç için yazılmış örnek kod parçacıkları bulunmaktadır. Dosya menüsü içerisinde ulaşılabilen bu kodlar Arduino içerisine kolaylıkla yüklenebilmektedir. Şekil 4.18’de IDE programı örnek kodlarının bulunduğu kısım görülmektedir.



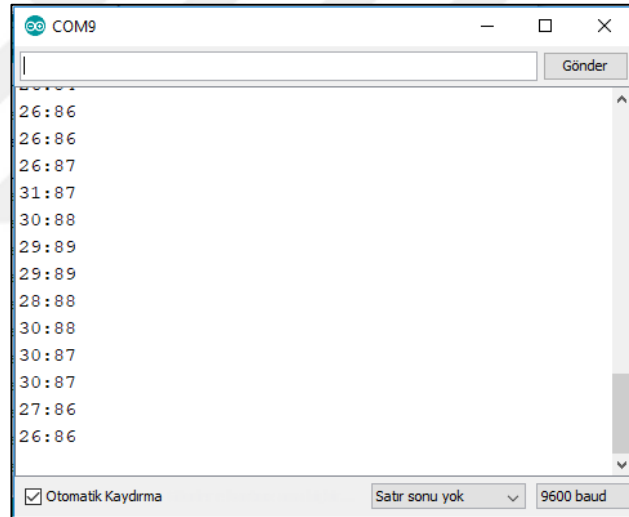
Şekil 4.18. IDE programı örnek kodları

Gerçek zamanlı geliştirilen sistemin algılayıcı düğüm kısmında sıcaklık algılayıcısı, ölçtüğü sıcaklık bilgisini I²C haberleşme protokolüne göre Arduino kartına gönderdiği için, I²C ile ilgili tanımlamaların yapıldığı 'i2cmaster.h' kütüphanesi çalışma dosyasına eklenmiştir. Kod içerisinde birçok kısımda bu kütüphane içerisindeki fonksiyonlar kullanılarak sıcaklık bilgisi Arduino'ya sorunsuz biçimde aktarılmıştır. Nabız ölçer algılayıcısı ise geliştirme kartının analog girişine (A0) bağlanmıştır. Analog uçtan gelen bilgiyi okumak için zamanlayıcı kesmesi kullanılmıştır. Zamanlayıcı kesmesi ile belirlenen zamanlarda mikrodenetleyici de kesme meydana geldiğinde analog uçtan gelen veri okunarak Arduino'ya değerlendirmek üzere aktarılmıştır. Okunan bu değerler farklı değerlere atanarak seri port üzerinden gönderilmek üzere hazır hale gelmektedir. 'integer' değişken olarak tanımlanan sıcaklık ve nabız bilgisi değişkenleri seri porta gönderilmeden önce 'string' formatına dönüştürülmektedir. IDE programındaki Serial.print() ve Serial.println() fonksiyonları serileri seri porta göndermek için kullanılan komutlardır. Bu iki fonksiyon arasındaki fark olarak, Serial.println() fonksiyonu ile veri gönderildikten sonra otomatik olarak bir alt satıra geçilmesi gösterilebilmektedir. Kod içerisinde ekstradan '/n' parametresi yazılmasına gerek kalmamaktadır. Bu fonksiyonlar içerisine yazılan veriler seri porta onlu (desimal) kodlanmış ASCII formatta gönderilmektedir. Fonksiyon içerisine yazılan ek parametreler ile veriler değişik formatlarda gönderilebilmektedir. Örneğin Serial.println(23,HEX) şeklinde veri seri porta gönderilirse seri porttan okunan değer 23 değerinin onaltılı sayı formatındaki (Hexadesimal) karşılığı yani 17 değeri olacaktır. Veriler karşı tarafa Zigbee haberleşme protokolüne sahip XBee modülleri ile gönderildiği için verilerin gönderimi ile ilgili birçok deneme yapılmış, en uygun çözümün verileri 'string' formatında karşıya göndermek olduğu sonucuna varılmıştır. 'string' formatına çevrilen bilgiler herhangi bir format değişikliğine uğramadan seri portta okunmaktadır. XBee modülleri ile gönderilen bu bilgiler erişim noktası tarafında 'string' formatında kolaylıkla alınabilmektedir. Uygulama yazılımı tarafında verileri birbirlerinden ayırabilmek için veri aralarına ':' işareti konmuştur. Gönderim formatı 'sıcaklık:nabız' şeklindedir. Uygulama yazılımı tarafından ':' işareti görüldüğünde veriler birbirinden ayrılıp bir dizi içerisinde farklı indislere aktarılmaktadır. Şekil 4.19'da IDE programından seri porta 'string' formatında verilerin gönderilmesi, Şekil 4.20'de ise gönderilen verilerin seri porttan nasıl okunduğu görülmektedir.



```
term_spo_kablosuz_string_tez $
String st;
unsigned int derece;
derece = int(cecius);
st = String(derece);
st += ":"; st += String(heart_rate);//st += ":";
Serial.println(st);
delay(1000);
}
void interruptSetup()
{
  TCRCR2A = 0x02; // This will disable the PWM on pin 3 and 11
  OCR2A = 0X7C; // This will set the top of count to 124 for the 500H
```

Şekil 4.19. Algılayıcı düğüm tarafında verilerin seri porttan gönderilmesi



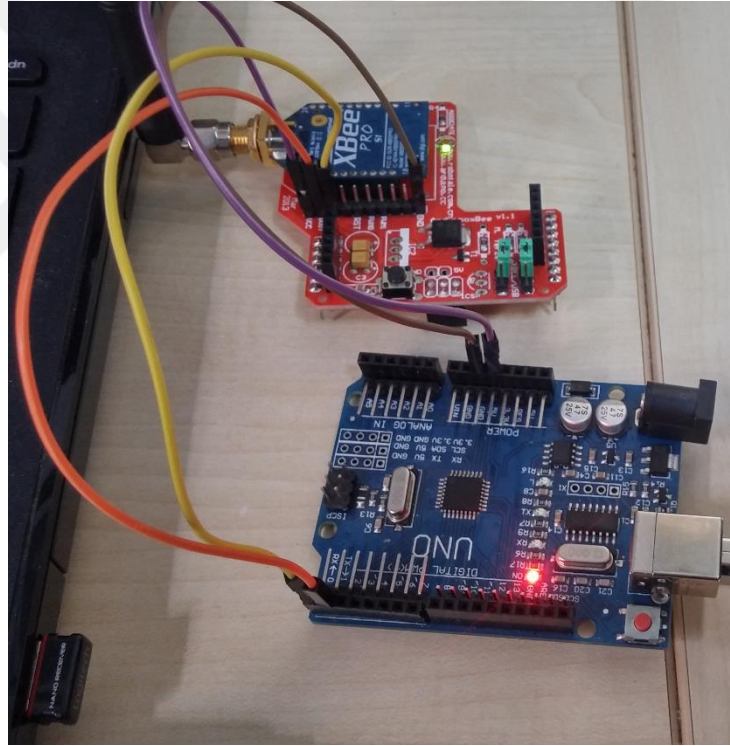
```
COM9
Gönder
26:86
26:86
26:87
31:87
30:88
29:89
29:89
28:88
30:88
30:87
30:87
27:86
26:86
Saturasyonu
9600 baud
```

Şekil 4.20. Verilerin seri porttan okunması

Algılayıcı düğüm üzerinde bulunan sıcaklık ve nabız ölçer algılayıcılardan alınan veriler kablosuz olarak erişim noktasına gönderilmektedir. Verilerin kablosuz gönderimi için Digi firmasının ürettiği XBee modülleri kullanılmıştır. XBee modüllerini kullanarak kablosuz haberleşme yapabilmek için bir ağ yapısı kurmak gereklidir. Ağ yapısında algılayıcı düğüm Zigbee haberleşme protokolüne uygun olarak son cihaz (End Device), erişim noktası ise merkezi düğüm olarak ayarlanmıştır. XBee modüller ile ilgili konfigürasyon ayarlamalarının yapılabilmesi için birçok araç bulunmaktadır ancak bunların içerisinde en çok kullanılan ve en güvenilir olanı yine Digi firmasının kendi ürettiği X-CTU yazılımıdır. X-CTU yazılımı kullanılarak modül

adreslemeleri yapılabilmekte ve diğer X-CTU modülleri ile haberleşme sağlanabilmektedir. Ücretsiz olan yazılım Digi firmasının internet sitesi üzerinden indirilebilmektedir [63].

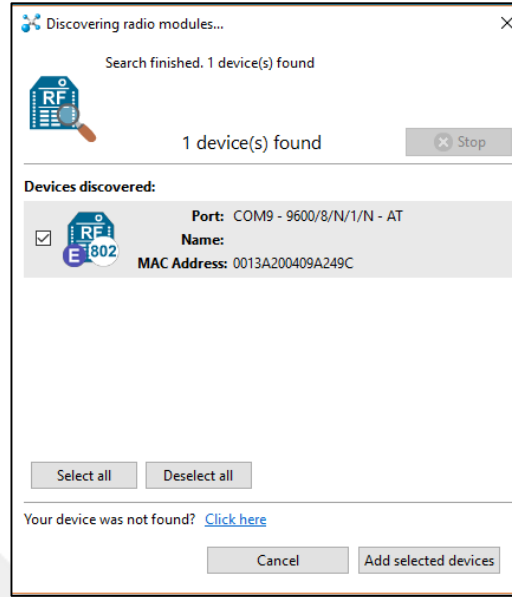
XBee modüller ile ilgili ayarların yapılabilmesi için X-CTU yazılımının XBee modüllerini bilgisayar üzerinde görmesi gerekmektedir. XBee, USB modülleri veya Arduino uyumlu XBee shield'ler üzerinden de gerekli bağlantılar yapılarak bilgisayara bağlanabilmektedir. Tez uygulaması için yapılan çalışmada XBee modüllerinin X-CTU ayarlamasının yapılabilmesi için Arduino ile XBee shield arasında gerekli bağlantılar yapılmıştır. Şekil 4.21'de Arduino ile XBee shield arasındaki bağlantı görülmektedir.



Şekil 4.21. X-CTU ayarları için XBee Arduino bağlantısı

Şekil 4.21'de görüldüğü gibi XBee modülünün çalışması için gerekli olan besleme gerilimleri Arduino üzerinden alınmaktadır. XBee modülünün veri alım ucu (Rx) ile Arduino üzerindeki Rx ucu, XBee üzerindeki gönderim ucu (Tx) ile Arduino üzerindeki Tx uçları birbirlerine bağlanmıştır. Gerekli bağlantılar yapıldıktan sonra Arduino kartı bilgisayara bağlandığında sanal bir seri port atanmaktadır. X-CTU yazılımı üzerinden herhangi bir radyo modülü olup olmadığı taraması yapıldığında

dođru bađlantılar yapılmıřsa yazılım XBee modülünü görmektir. řekil 4.22'de arama sonucunda bulunan XBee modülle ilgili bilgiler görülmektedir.

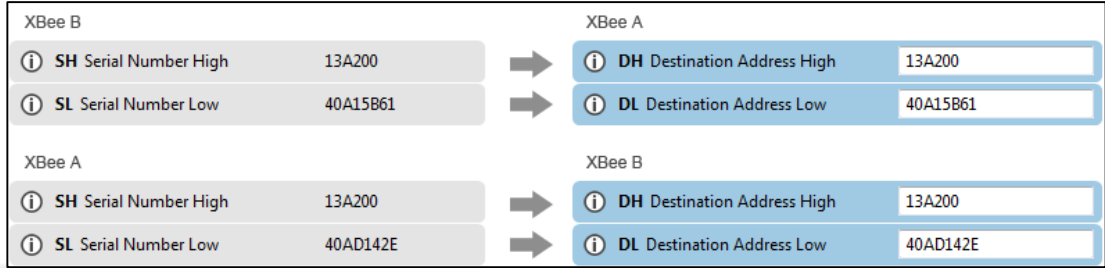


řekil 4.22. X-CTU yazılımında bulunan XBee modül bilgileri

XBee modülü X-CTU tarafından tanındıktan sonra kullanıcı tarafından ayarlamaların yapılabileceđi ekran gelmektedir. Ayarlamalar içerisinde benzetim alıřmasında bulunan GTS ve CAP ayarları bulunmamaktadır. XBee modülleri iki řekilde haberleřebilmektedir. Bunlar AT (Application Transparent) mod ve API (Application Programming Interface) modlarıdır. API modu birden ok XBee düđüm ile kurulan büyük ađlar için kullanılan haberleřme modudur. API modunda veriler bir paket formatı (bařlık+veri uzunluđu+hedef adresi+payload+checksum) içerisinde karřı tarafa gönderilir.

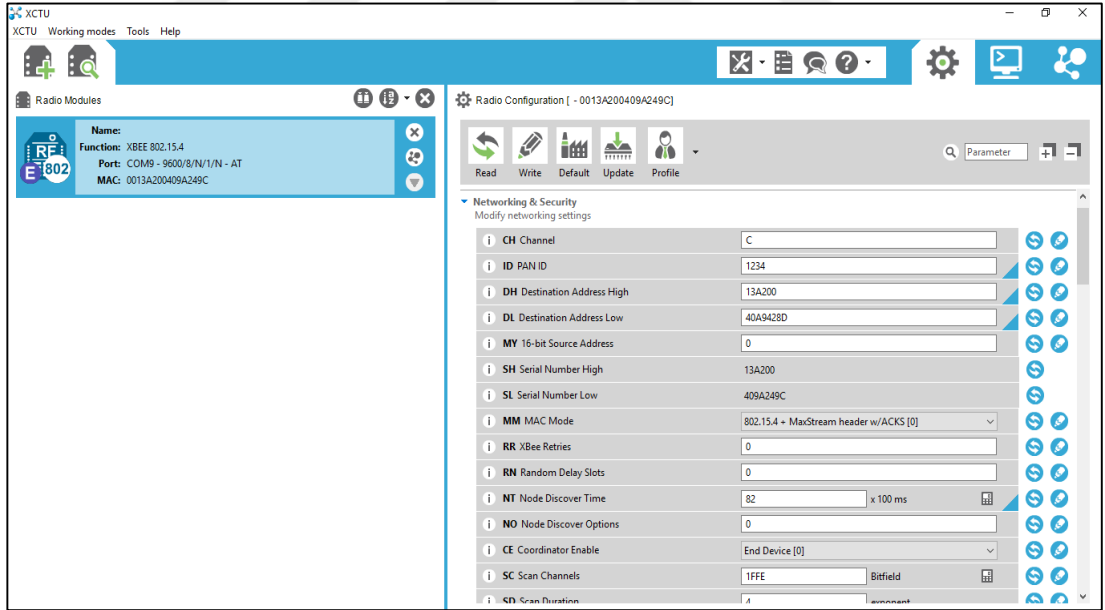
Gerekleřtirilen sistemde iki adet XBee düđüm birbirleri ile haberleřtiđi için, AT modda (saydam) iletiřim gerekleřtirilmiřtir. XBee modülleri AT modda alıřırken seri hat deđiřimi görevi görmekteirler. Seri giriřten alınan tüm veriler hemen kablosuz ortam üzerinden iletilmektedir. AT modda iletiřim, iki modül bir tel ile birbirine bađlıymıř gibi sonu vermektir. Bir XBee'nin bařka bir XBee ile iletiřim kurması için, gönderen XBee'nin alıcının adresine ihtiyaı bulunmaktadır. AT modda alıřırken, bu adresin iletiřim halindeki XBee tarafında yapılandırılması gerekmektedir. XBee modüller, hedef modülün tam 64 bit adresini

saklayabilmektedirler. Bu adresin iki parametresi bulunmaktadır; Bunlar, Hedefin yüksek adresi (Destination High - DH) ve hedefin düşük adresidir (Destination Low - DL) [64]. A ve B modüllerinin iletişim kurabilmesi için, XBee A'nın hedef adresini (DH + DL) XBee B'nin MAC adresi (SH + SL) olarak ayarlamak ve bunun tersini karşı tarafta yapmak gerekmektedir. Örnek ayarlama Şekil 4.23'de görülmektedir [64].



Şekil 4.23. AT modda adres bilgilerinin ayarlanması

Algılayıcı düğüm tarafında kullanılan XBee modülünün genel ağ ayarlarının yapıldığı ekran Şekil 4.24'de görülmektedir.



Şekil 4.24. X-CTU algılayıcı düğüm üzerindeki XBee modül ayarları

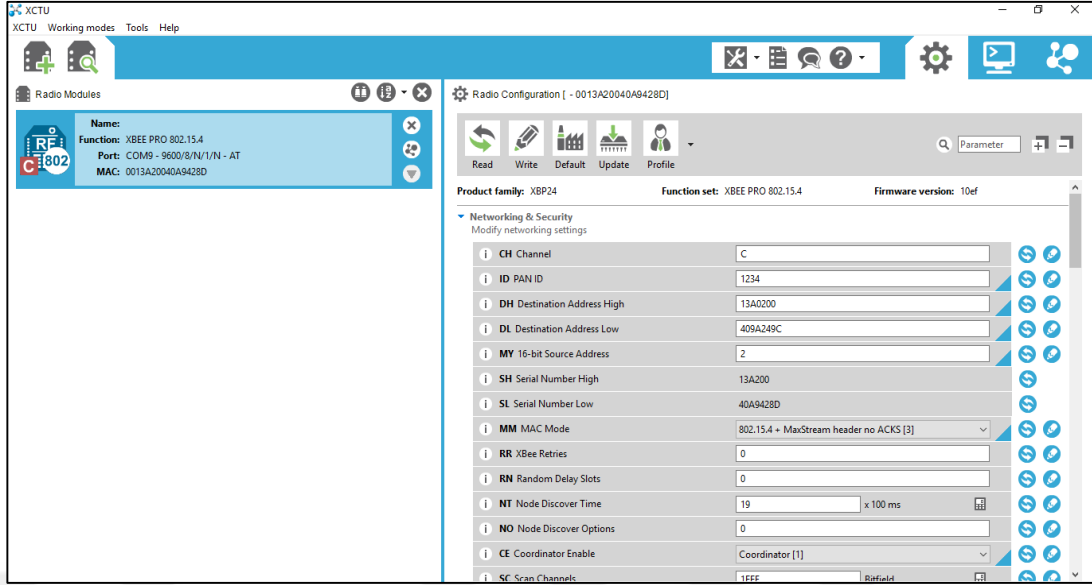
XBee modülünün 'PAN ID' ayarı '1234' olarak yapılmıştır. Bu ayar erişim noktası tarafında kullanılan XBee modülünde de aynı ayarlanmıştır. Bir ağ içerisindeki 'PAN ID' değerlerinin aynı olması gerekmektedir. Aksi takdirde modüller birbirleriyle haberleşmemektedir. 'DH' (Destination Address High) ve 'DL' (Destination Address Low) ayarları AT moda uygun olarak yapılmıştır. Bu kısma yukarıda anlatıldığı gibi

erişim noktası tarafındaki XBee modülün adres bilgileri doğru şekilde yazılmıştır. Bu şekilde cihazlar birbirleriyle direkt olarak haberleşebilmektedir. XBee modülün kendi adres ayarları; 'SH' (Serial Number High) değeri '13A200', 'SL' (Serial Number Low) değeri ise '409A249C' şeklinde yapılmıştır. Algılayıcı düğüm sadece veri gönderen yani son cihaz (End Device) olduğu için 'CE' (Coordinator Enable) seçeneği 'End Device' olarak ayarlanmıştır. Şekilde görülmeyen kısımda AT modunda iletişim kurulacağı için 'AP - API Enable' ayarı 'Disable' olarak seçilmiştir. 'Baud Rate' ayarı Arduino yazılımında belirlenen şekilde '9600' olarak ayarlanmıştır. Farklı baud hızı ayarlarında iletişim gerçekleşmemektedir.

4.2.2. Koordinatör düğüm

Tez çalışması çerçevesinde belirlenen senaryoya göre geliştirilen gerçek zamanlı sistemin ikinci kısmı koordinatör düğümdür. Koordinatör düğüm, fizyolojik verileri toplamak için kurulan KVAA sisteminin erişim noktası ile RFID sistemin okuyucu kısmından oluşmaktadır. Benzetim senaryosunda olduğu gibi KVAA erişim noktası ve RFID okuyucu tek cihaz içerisinde birleştirilmiştir. KVAA ve RFID sistemlerin çalışma şekilleri birbirlerinden farklıdır. Normal şartlarda ayrı ayrı bilgisayar ortamına bağlanan ve gelen verileri farklı şekillerde aktaran bu sistemler IoT tabanlı olarak gerçekleştirilen sistemde bir araya getirilmiştir. Bu birleştirilme ile farklı çalışma frekansları ve farklı çalışma şekilleriyle elde edilen fizyolojik ve kişisel veriler bir merkez tarafından toplanmış ve bilgisayar ortamına aktarılmıştır.

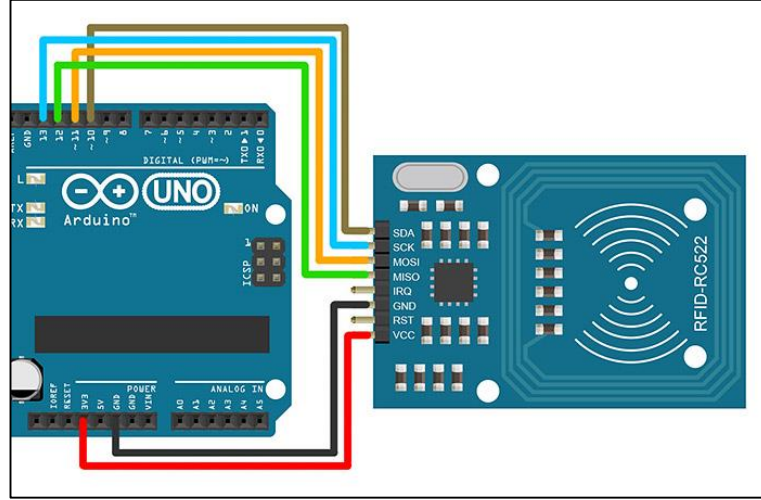
Algılayıcı düğümden gelen verileri kablosuz ortamdan alabilmek için Xbee modül kullanılmıştır. Erişim noktası tarafında erişim noktası olarak kullanılabilen Xbee Pro S1 modülü tercih edilmiştir. Algılayıcı düğüm tarafında olduğu gibi Xbee Pro S1 için de X-CTU programında haberleşme için ayarlamalar yapılmıştır. Şekil 4.25'de X-CTU programında Xbee Pro S1 için yapılan ayarlar görülmektedir.



Şekil 4.25. Erişim noktası tarafındaki XBee Pro S1 ayar ekranı

Şekil 4.25’de görüldüğü gibi, Xbee PRO S1 modülünün ‘PAN ID’ ayarı algılayıcı tarafındaki gibi ‘1234’ olarak yapılmıştır. ‘DH’ ve ‘DL’ kısmına algılayıcı düğüm tarafındaki Xbee S1 modülünün adres bilgileri yazılmıştır. Xbee PRO S1 modülün kendi adres ayarları; ‘SH’ değeri ‘13A200’, ‘SL’ değeri ise ‘40A9428D’ şeklinde yapılmıştır. ‘CE’ (Coordinator Enable) seçeneği ‘Coordinator’ olarak ayarlanmıştır. Benzetim çalışmasında süper çerçeve kılavuzlu modda sürekli aktif periyotta (no-sleep mode) çalışacak şekilde ayarlanmıştır. Gerçek zamanlı uygulamayla benzetim çalışmasının uyumlu olması için, şekilde görülmeyen kısımda ‘No Sleep Mode’ seçeneği ayarlanmıştır. Yani cihazlar uyku moduna geçmeden sürekli veri alıp göndermektedirler.

Koordinatör düğümün RFID okuyucu kısmı için Arduino uyumlu MF-RC522 okuyucu kartı kullanılmıştır. Şekil 4.26’da MFRC522 kartı ile Arduino arasındaki kablo bağlantıları görülmektedir.

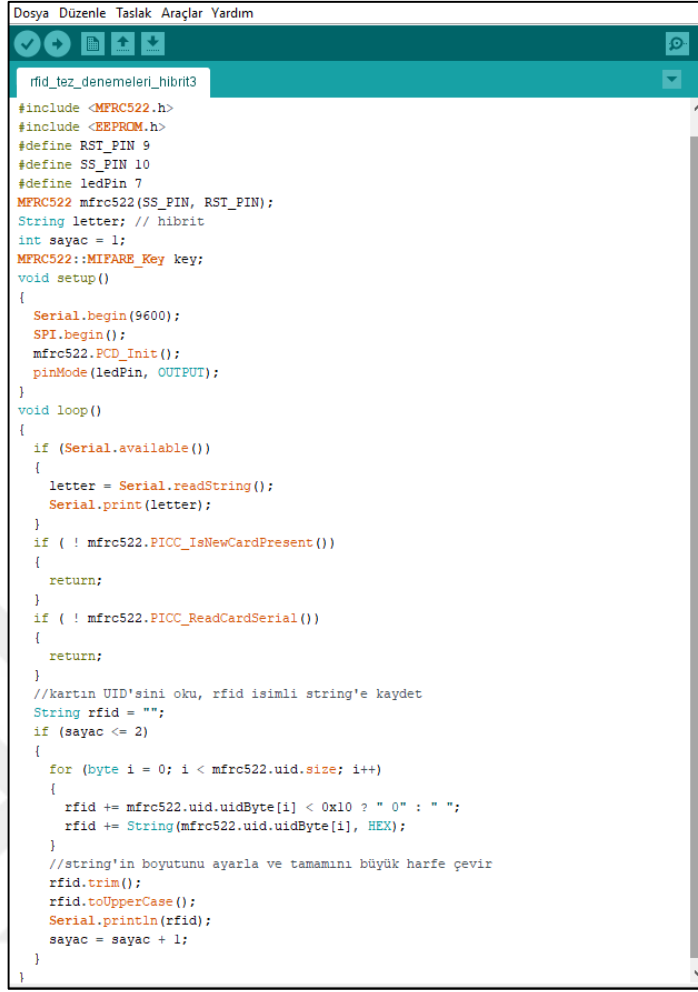


Şekil 4.26. MFRC522 okuyucu kartı ile Arduino bağlantısı

MF-RC522 okuyucu kartı 13,56 MHz’de çalışmaktadır. Bu yüzden etiket okuma mesafesi NFC kartlar gibi oldukça kısadır. Etiketler belirli bir mesafeye geldiğinde okuyucu ile iletişime geçerek veri aktarımı gerçekleştirmektedir. Sistemde kullanılan etiketler pasif etiketlerdir, yani dahili enerji kaynakları bulunmamaktadır. Bellekleri aktif etiketlere göre çok düşük olduğundan veri saklama yetenekleri sınırlıdır. Geliştirilen sistemde hastaların her birine ait RFID etiketler bulunmaktadır. Bu kartlar kişilerin kişisel bilgilerini temsil eden bir sayı dizisi (ID) saklamaktadır. Etiket RFID okuyucuya okutulduğu anda uzaktaki sunucu kısmında ilgili hastanın gerçek kişisel bilgileri ile ID bilgisi eşleştirilmektedir. Uygulama yazılımı kısmında eşleştirme işlemi daha detaylı bir şekilde anlatılmaktadır.

Koordinatör düğümde, KVAA erişim noktası ve RFID okuyucunun beraber çalışabilmesi için Arduino IDE programında kod yazılmıştır. KVAA ile fizyolojik veriler, RFID ile kişisel veriler yazılan kod ile sorunsuz bir şekilde toplanarak değerlendirmek üzere uygulama yazılımına gönderilmiştir. Şekil 4.27’de Arduino IDE programında yazılan kod parçası görülmektedir.

Algılayıcı düğüm tarafından gönderilen fizyolojik veriler Arduino’nun seri port girişlerine bağlı olan XBee modül tarafından okunmaktadır. Bu işlem için seri port ile ilgili ayarlamalar kod içerisinde yapılmaktadır. ‘string’ formatında gelen veriler Serial.read() fonksiyonu yerine Serial.readString() fonksiyonuyla okunmakta ve bir değişkene aktarılmaktadır. ‘string’ formatındaki veri hiç dokunulmadan uygulama yazılımında değerlendirmek üzere seri port üzerinden bilgisayara aktarılmaktadır.



```
Dosya Dzenle Taslak Araçlar Yardım
rfid_tez_denemeleri_hibrit3
#include <MFRC522.h>
#include <EEPROM.h>
#define RST_PIN 9
#define SS_PIN 10
#define ledPin 7
MFRC522 mfrc522(SS_PIN, RST_PIN);
String letter; // hibrit
int sayac = 1;
MFRC522::MIFARE_Key key;
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  if (Serial.available())
  {
    letter = Serial.readString();
    Serial.print(letter);
  }
  if (!mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  if (!mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //kartın UID'sini oku, rfid isimli string'e kaydet
  String rfid = "";
  if (sayac <= 2)
  {
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
      rfid += mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ";
      rfid += String(mfrc522.uid.uidByte[i], HEX);
    }
    //string'in boyutunu ayarla ve tamamını büyük harfe çevir
    rfid.trim();
    rfid.toUpperCase();
    Serial.println(rfid);
    sayac = sayac + 1;
  }
}
```

Şekil 4.27. Koordinatör düğümü IDE kodu

RFID sistemde, RFID kartın çalışması için MF-RC522 kartı ile ilgili tanımlamaları ve çalışma fonksiyonlarını içeren kütüphane dosyası (MFRC522.h) kodun giriş kısmına eklenmiştir. Okuyucu kartlar ile ilgili tanımlamalar 'setup()' fonksiyonunun üstünde yapılmaktadır. 'setup()' fonksiyonu içerisinde 'mfrc522.PCD_Init()' fonksiyonu ile okuyucu kart okuma işlemine hazır hale gelmektedir. Sürekli çalışan 'loop' fonksiyonunun içerisinde okuma mesafesindeki etiketin seri numarasının okunması ve öncekinden farklı etiket okutulup okutulmadığının kontrolü ile ilgili fonksiyonlar bulunmaktadır. Hastaya ait okunacak RFID etiketinin seri numarası da 'string' formatında kaydedilmektedir.

KVAA ve RFID sistemle ilgili IDE programında yazılan koda göre koordinatör düğümün çalışması şu şekildedir; Koordinatör düğüm çalışmaya başladığında, KVAA erişim noktasına bağlı XBee düğüm üzerinden kablosuz olarak veri gelip gelmediği sürekli kontrol edilmektedir. Serial.available() fonksiyonu kontrol işlemini

koordinatör düğüm tarafından seri porta gönderilen tüm değerler Node.js platformunda socket programlama ile okunarak veri tabanında ilgili yerlere yazılmaktadır. Algılayıcı düğüm tarafında ölçülüp gönderilen fizyolojik verileri izlemek için uygulama yazılımı kısmında web tabanlı uygulama geliştirilmiştir. Web tabanlı uygulamaya bir 'login' sayfası üzerinden ulaşılmaktadır. Sisteme tanımlı kullanıcılar şifreleri ile sisteme girip fizyolojik verilerin grafiksel olarak takibini yapabilmektedir. Fizyolojik verilerin izlendiği herhangi bir anda RFID etiket okuyucuya okutulduğunda etiketin seri numarasına göre tanımlı olan kişinin ismi fizyolojik verilerle eşleştirilmekte ve web tabanlı uygulamada gösterilmektedir. Web tabanlı uygulama içerisinde yüksek vücut sıcaklığı ve nabız değerleri için alarm durumları tanımlanmıştır. Alarm durumları oluştuğunda hem grafikler üzerinde uyarı meydana gelmektedir hem de sistemdeki kayıtlı kullanıcı uyarılmaktadır. Ayrıca benzetim senaryosunda olduğu gibi gerçek zamanlı uygulama kısmında da QoS kriterlerinin sağlanıp sağlanmadığı incelenmiştir. Fizyolojik verilerin algılayıcı düğüm tarafında ölçülüp seri porta gönderildiği andan web tabanlı uygulamada gösterildiği (veri tabanına yazılış zamanı) ana kadar olan süreler takip edilmiştir. Tablo 1.1'de görülen vücut sıcaklığı ve nabız referans bilgilerinin doğru değerlendirilebilmesi için, geçerli olan gecikme sürelerinin altında sürelerde verilerin iletildiği görülmüştür. Uygulama yazılımı ile ilgili tüm ayrıntılar sırasıyla anlatılacaktır.

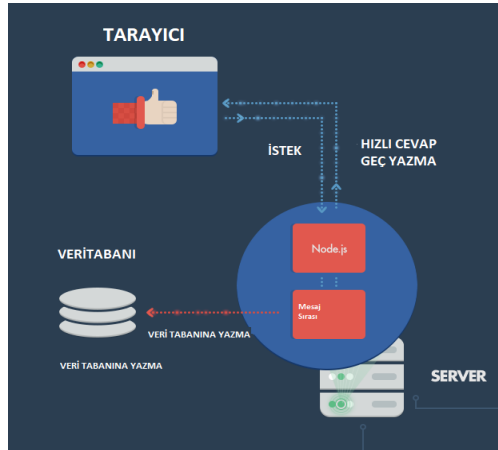
4.2.3.1. Koordinatör düğümden gelen verilerin okunması

Tez çalışmasının bu kısmında Node.js platformu ile gelen verilerin nasıl okunduğu ve veri tabanına nasıl yazıldığı anlatılmaktadır.

Node.js, sunucu tarafında JavaScript kodlama dilini kullanarak web uygulamaları geliştirmek için kullanılan açık kaynaklı bir yürütme çevresi platformudur [66]. Geliştirici ekibin sitesinde Node.js şöyle tanımlanmaktadır; Node.js platformu, Chrome V8 JavaScript motoru üzerine inşa edilmiş bir JavaScript yürütme platformudur. Node.js platformunun olay-güdümlü, bloklanmayan bir I/O (giriş/çıkış) modeli kullanması, platformu hafif ve verimli bir hale getirmektedir. Node.js sisteminin paket sistemi olan NPM, dünyadaki en geniş açık kaynaklı sistemdir [67]. Javascript, zengin metin işaretleme dili (Hyper Text Markup Language - HTML) kodlarının arasında yazılıp tarayıcıya gönderilmektedir. Tarayıcı, bu kodları alıp

yorumlayarak ekrandaki HTML nesnelere yer deęiřtirmesi, özellik deęiřtirmesi gibi işlemleri yapmaktadır. Javascript kodları, sunucuda yorumlanmak yerine, bilgisayar üzerinde çalışırlar. Node.js platformu Javascript kodlarının sunucu tarafında da yorumlanmasını sağlamaktadır. Bu durum, sunucu ve istemci için kodlama farklılıklarını ortadan kaldırarak tüm ihtiyaçların tek programlama dili ile karşılanmasını sağlamıştır [66].

Sunucu ile kullanıcı arasında sürekli bir bağlantı gerektiren ve bağlantı üzerinde yoğun olarak gerçek zamanlı veri transferi gerçekleřtiren bir uygulama tasarımı için Node.js uygun bir platform olarak karşımıza çıkmaktadır. Node.js ve Socket.io kombinasyonu ile kullanıcıya gerçek zamanlı olarak güncelleme bilgileri veren bir uygulama geliştirilebilmektedir. Node.js platformunda, sunucu ve istemci tarafında yapılan hemen hemen tüm işlemlerde (veri transferi, form doęrulama, MongoDB veri tabanı işlemleri, dosya nesne modeli (Document Object Model - DOM) öğelerinin etkileşimli özellikleri) Javascript kodlama kullanılması geliştiriciler için büyük kolaylıklar sağlamaktadır [66]. Node.js platformunda sunucu yapılandırması oldukça kolaydır. Birkaç satırlık kod ile tam anlamıyla fonksiyonel bir HTTP sunucusu yapılabilmektedir. Node.js platformunun genel çalışma şekli Şekil 4.29'daki gibidir.

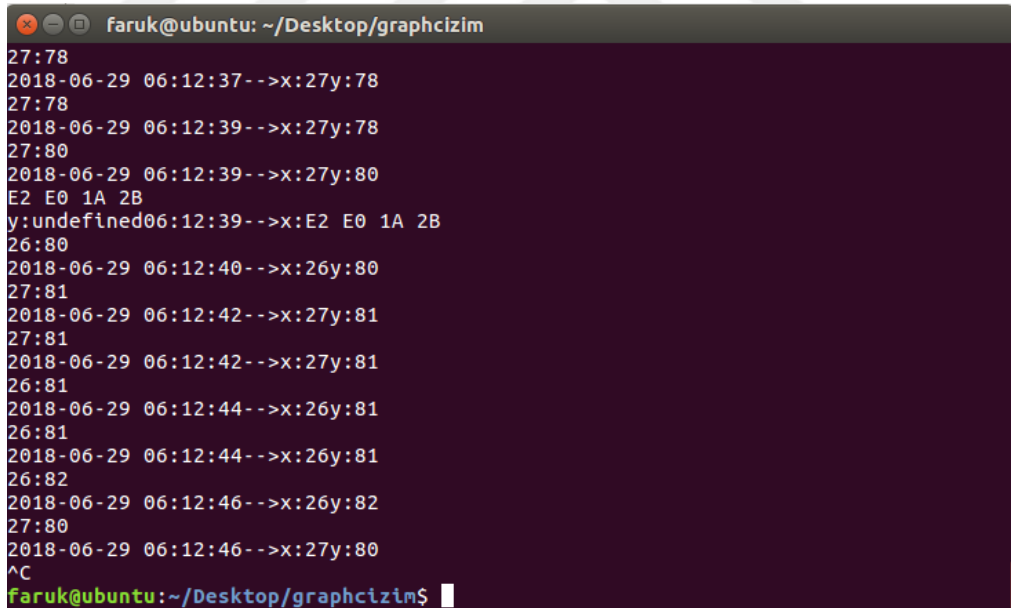


Şekil 4.29. Node.js çalışma şekli

Node.js platformu çeşitli işletim sistemleri (Windows, Ubuntu, MACOS) üzerinde kurulabilmektedir. Kurulum dosyalarına platformun internet sitesi üzerinden ücretsiz olarak erişilebilmektedir. Tez çalışmasında seri port üzerinden gelen verileri okumak için Node.js platformu Ubuntu işletim sistemi üzerine kurulmuştur. Gerçek zamanlı sistemin üçüncü kısmı olan web tabanlı uygulama yazılımının tamamının çalıştırılması

için açık kaynak kodlu ve ücretsiz olması dolayısıyla Ubuntu işletim sistemi seçilmiştir. İşletim sistemi içerisinde açılacak bir terminalde tek satır kod yazılarak Node.js platformu kolaylıkla kurulabilmektedir. Node.js platformunda kodlar '.js' uzantısı ile kaydedilmektedir. Kodlar için ayrıca bir derleyici kurmaya gerek yoktur. Javascript dilinde bütün değişkenler 'var' ön tanımlamasıyla kaydedilmektedir. Dosyalar terminal üzerinde dosya ismi çağrılarak çalıştırılmaktadır. Node.js kodlarını çağırmak için dosya ismi öncesine 'node' ifadesi yazmak gerekmektedir. Örneğin 'server.js' adlı dosyayı çalıştırmak için 'node server.js' ifadesi yazmak gereklidir.

Node.js programında seri porttan verileri okumak için gerekli olan kodlar 'server.js' adlı dosyaya yazılmıştır. Kod içerisinde seri porttan gelen 'string' tipindeki değerler aralarındaki ':' işareti görüldüğünde birbirlerinden ayrıştırılmışlardır (split). Ayrıştırılan bu değerler gerçek zamanlı grafikler için bir dizinin farklı elemanlarına atanmışlardır. Grafikler bu dizilerdeki veriler çekilerek çizdirilmektedir. 'server.js' dosyası çalıştırıldıktan sonraki terminal ekranı Şekil 4.30'da görülmektedir.

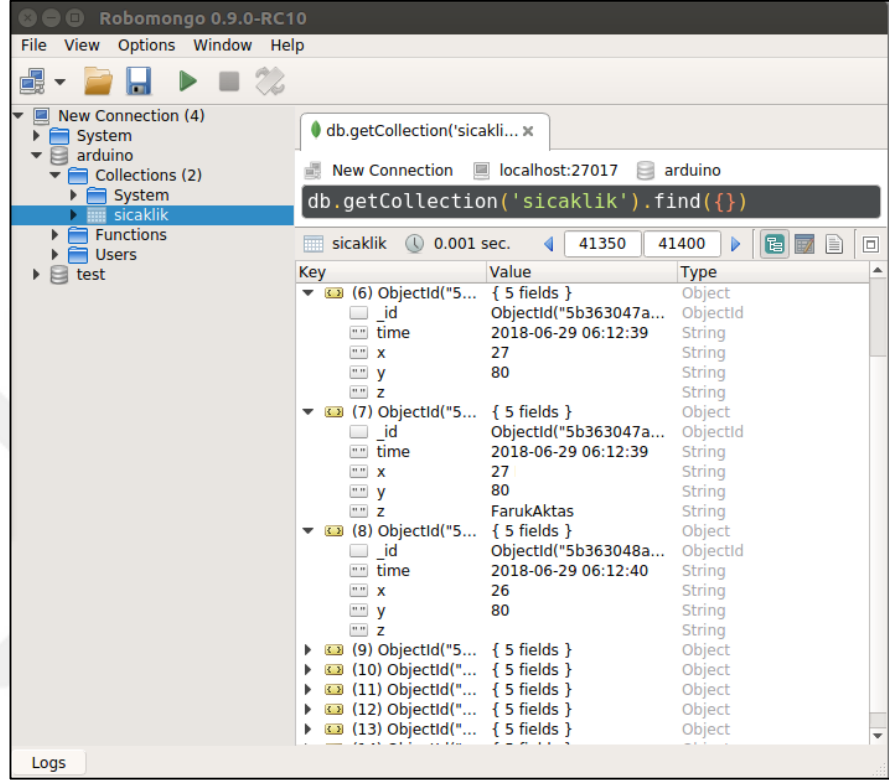


```
faruk@ubuntu: ~/Desktop/graphcizim
27:78
2018-06-29 06:12:37-->x:27y:78
27:78
2018-06-29 06:12:39-->x:27y:78
27:80
2018-06-29 06:12:39-->x:27y:80
E2 E0 1A 2B
y:undefined06:12:39-->x:E2 E0 1A 2B
26:80
2018-06-29 06:12:40-->x:26y:80
27:81
2018-06-29 06:12:42-->x:27y:81
27:81
2018-06-29 06:12:42-->x:27y:81
26:81
2018-06-29 06:12:44-->x:26y:81
26:81
2018-06-29 06:12:44-->x:26y:81
26:82
2018-06-29 06:12:46-->x:26y:82
27:80
2018-06-29 06:12:46-->x:27y:80
^C
faruk@ubuntu:~/Desktop/graphcizim$
```

Şekil 4.30. Node.js platformunda seri porttan gelen verilerin okunması

Seri porttan gelen veriler tez çalışmasındaki senaryoya uygun olarak bir veri tabanına kaydedilmektedir. Veri tabanı olarak MongoDB tercih edilmiştir. MongoDB veri tabanına Robomongo programı kullanılarak erişilebilmektedir. Veri tabanında veriler 5 ayrı alana yazılmaktadır. Birinci alanda verinin geldiği cihaza verilen 'ObjectID' numarası bulunmaktadır. İkinci alanda verinin veri tabanına yazıldığı gün ve tarih

bilgisi bulunmaktadır. Üçüncü alanda vücut sıcaklığı (x), dördüncü alanda nabız verisi (y) ve beşinci alanda RFID etiketin seri numarasının kime ait olduğu bilgisi (z) bulunmaktadır. Şekil 4.31’de seri porttan veri okunurken verilerin MongoDB veri tabanına nasıl yazıldığı görülmektedir.



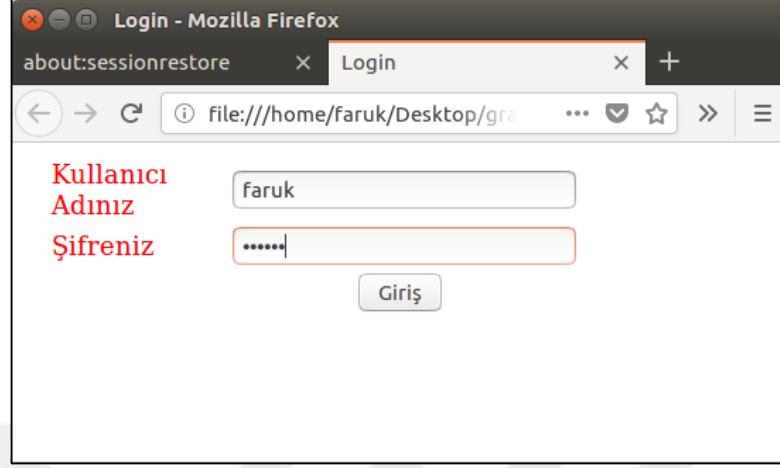
Şekil 4.31. MongoDB veri tabanı ekranı

4.2.3.2. Web tabanlı gerçek zamanlı izleme uygulaması

Gerçek zamanlı olarak gerçekleştirilen sistemin son kısmı izleme sistemidir. Socket.io() kütüphanesiyle port programlama yapılarak seri porttan okunan veriler Node.js ile ayrıştırılarak (pars edilerek) işlenmektedir. İşlenen bu veriler hem veri tabanına yazılmakta hem de HTML içine gerçek zamanlı olarak gömülmektedir. HTML kod içerisinde yerleştirilen Javascript, basamaklı stil şablonları (Cascading Style Sheets - Css) ve bootstrap kütüphaneleri ile dinamik grafik ekranlarında veriler kolaylıkla izlenebilmektedir. Grafik ekranları dinamik olup ekranın büyüklüğüne göre ilgili grafiklerin boyutları değişebilmektedir.

Gerçek zamanlı izleme sistemine ‘Kullanıcı adı’ ve ‘Şifre’ bilgilerinin sorulduğu bir ‘login’ sayfası üzerinden erişilebilmektedir. Sistemde kayıtlı kullanıcılar kullanıcı adı

ve şifrelerini girerek kişilerin verilerine erişebilmektedir. Şekil 4.32’de giriş sayfası ekranı görülmektedir.



Şekil 4.32. Gerçek zamanlı izleme sistemi giriş ekranı

Gerçek zamanlı izleme sistemi ekranında fizyolojik verilerle ilgili üç adet grafik ekran bulunmaktadır. Bunlardan nabız bilgisi çizgi grafik olarak tasarlanmıştır. Çizgi grafiğin yatay ekseninde verinin geldiği zaman bilgisi, dikey ekseninde o anki nabız bilgisi görülmektedir. Sıcaklık bilgisi için bir yüzdelik dilim grafiği bir de kadran şeklinde grafik çizdirilmiştir. Nabız bilgisi ve kadran şeklindeki sıcaklık grafiklerini hemen altında gelen verilerin yazıldığı bir bilgilendirme kutucuğu bulunmaktadır. Bu kutucukta hem anlık gelen veri yazılmakta hem de gelen verinin normal değerlerde mi yoksa yüksek veya düşük değerlerde mi bilgisi bulunmaktadır. Grafik ekranı gelen bilgilere göre dinamik olarak sürekli değişmektedir.

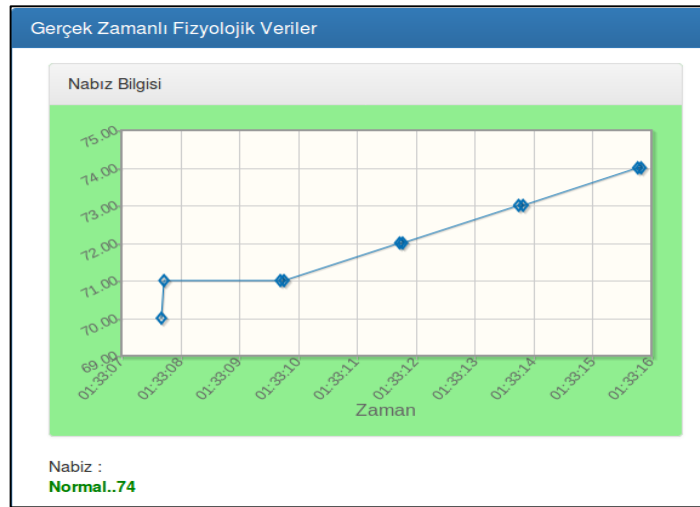
Gerçek zamanlı izleme sisteminde gelen verilerin belirli değerleri aştığı durumlar için alarm durumları oluşturulmuştur. Normal kalp atım oranı (nabız) ve vücut sıcaklığı değerleri kişilerin yaşlarına göre değişmektedir. Özellikle bebekler için bu verilerin takibi çok önemlidir. Bebeklerde normal kalp atım oranları 100 civarında iken erişkin bir birey için bu değer 60 ila 80 arasında değişmektedir. Yaşlara göre normal kalp atım oranları ve vücut sıcaklığı değerleri Tablo 4.6’da görülmektedir [68].

Tablo 4.6. Çocuk ve erişkinler için normal vücut sıcaklığı ve kalp atım oranı değerleri

Yaş	Vücut Sıcaklığı	Kalp Atım Hızı (Uyanık)	Kalp Atım Hızı (Uyku)
Prematüre	36,0–37,2	110–180	–
Yenidoğan	36,0–37,2	100–180	80–160
1–12 Ay	36,0–37,2	100–160	75–160
1–3 Yaş	36,4–37,0	80–110	60–90
3–6 Yaş	36,4–37,0	70–110	60–90
6–12 Yaş	36,4–37,0	65–110	60–90
12–18 Yaş	36,4–37,0	60–100	50–90
18 Yaş Üstü	36,4–37,0	60–100	50–90

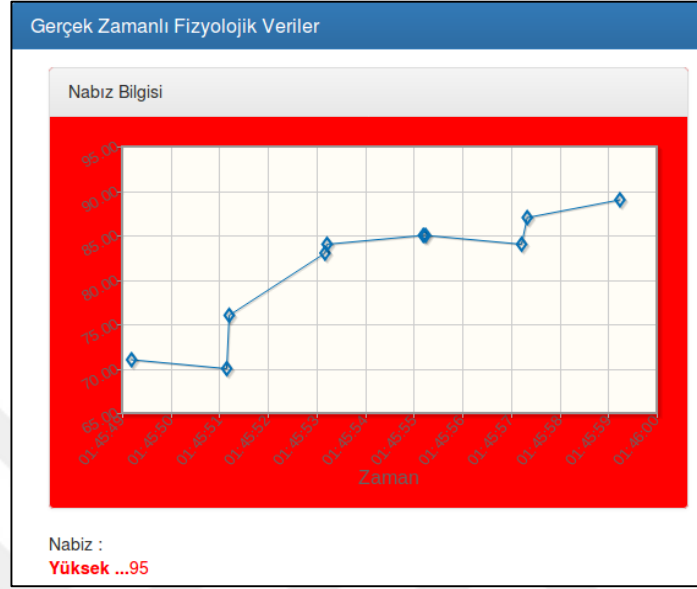
Gerçek zamanlı izleme sisteminde hastadan gelen normal değerlerin üzerinde çıkan veriler için değerlendirme yapılarak alarm durumları oluşturulmuştur. Alarm durumları değerleri yaş gruplarına göre yazılım ile esnek biçimde ayarlanabilmektedir.

Nabız bilgisi grafiği ekrana çizdirilirken alt kısmında bulunan kutucukta o anda gelen veri ve gelen verinin normal değer mi yoksa yüksek veya düşük değer mi olduğu bilgisi yazmaktadır. Grafik ekranlarının gösterimi için 90 üstü nabız değerleri yüksek, 73 altındaki nabız değerleri ise düşük nabız değeri olarak kabul edilmiştir. Normal değerlerde grafik ekranı ve kutucuk içindeki yazı açık yeşil renk tonlarında görülmektedir. Şekil 4.33’de gelen normal değerlerde gelen veriler için grafik ekranı görülmektedir.



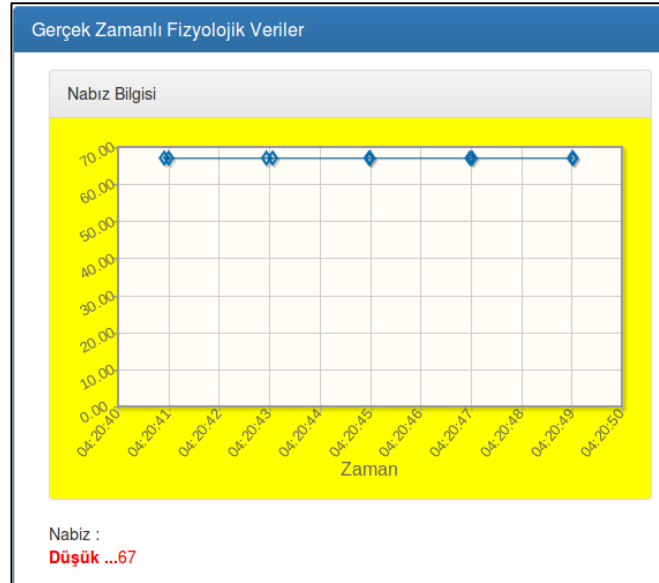
Şekil 4.33. Normal değerler için nabız grafiği ekranı

Yüksek değerli kabul edilen bir nabız değeri geldiğinde grafik alanının rengi kırmızı renge dönmekte ve kutucuk ekranındaki yazıda kırmızı renkte değişerek kullanıcıyı uyarılmaktadır. Şekil 4.34’de yüksek nabız verisi için grafik ekranı görülmektedir.



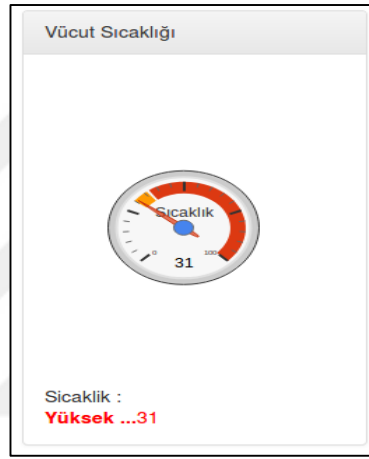
Şekil 4.34. Yüksek değerler için nabız grafiği ekranı

Düşük değerli kabul edilen bir nabız değeri geldiğinde grafik alanının rengi sarı renge dönmekte ve kutucuk ekranındaki yazıda kırmızı renkte değişerek kullanıcıyı uyarılmaktadır. Şekil 4.35’de düşük nabız verisi için grafik ekranı görülmektedir.



Şekil 4.35. Düşük değerler için nabız grafiği ekranı

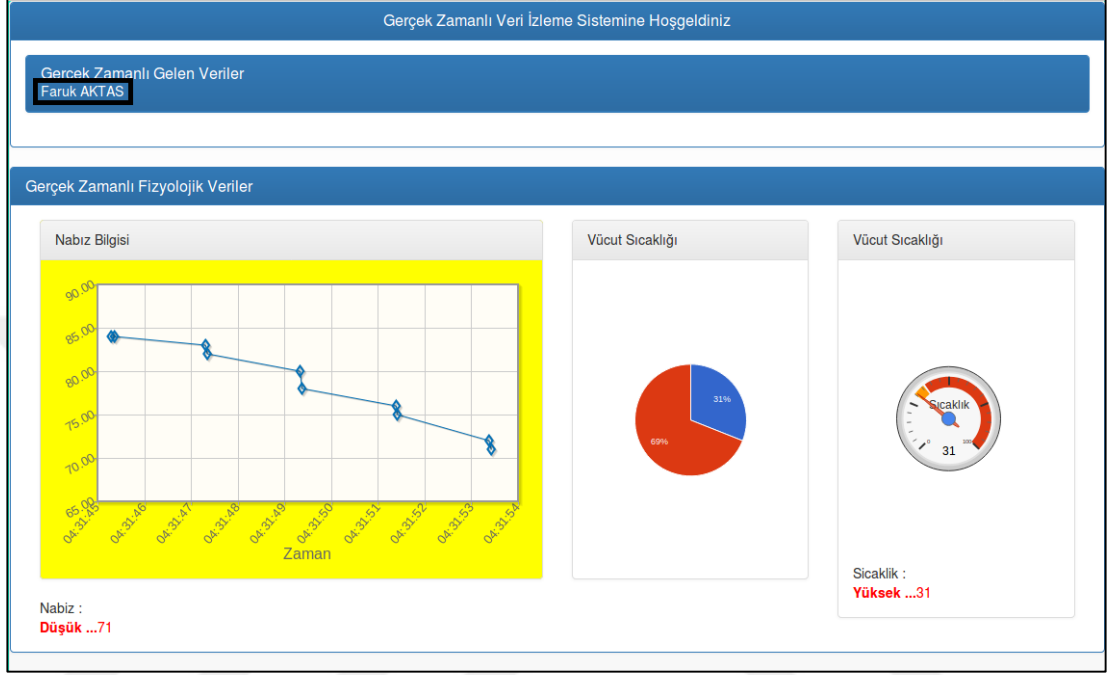
Sıcaklık bilgisi kadran şeklindeki grafikte gösterilmektedir. Grafiğin hemen altında hem anlık gelen sıcaklık verilerini gösteren hem de değerin normal mi yoksa yüksek mi olduğunu belirten bir kutucuk bulunmaktadır. Vücut sıcaklığı için oluşturulan alarm değeri yazılım ile esnek biçimde ayarlanabilmektedir. Grafik gösterimi için 30 derece üzerindeki vücut sıcaklıkları yüksek değer olarak kabul edilmiştir. Yüksek sıcaklık değeri geldiğinde kadran grafik üzerindeki ibre, sıcaklığın yüksek olduğunu belirten kısma gelmektedir ve aynı anda grafik altındaki bilgilendirme kutucuğundaki yazı kırmızı renge dönüşerek ‘Yüksek’ şekline kullanıcıyı uyarılmaktadır. Şekil 4.36’da yüksek sıcaklık değeri için grafik ekranı görülmektedir.



Şekil 4.36. Sıcaklık ekranı

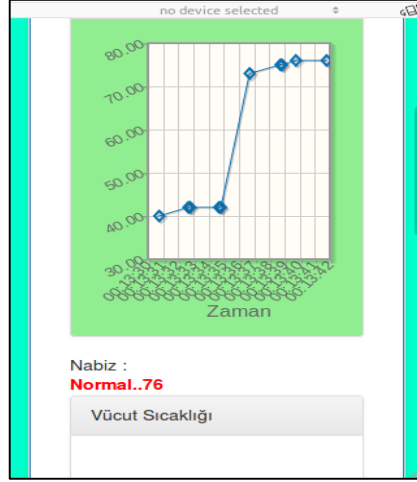
Hastaya ait fizyolojik veriler grafikler aracılığıyla izlenebilirken kişisel verilerin eşleştirilmesi için RFID etiketin okuyucuya iletişim mesafesinde yakınlaştırılması gerekmektedir. Fizyolojik veriler KVAA ile toplanıp birer saniye aralıklarla bilgisayar ortamına gönderilirken, RFID etiket istenilen zamanda sisteme okutulabilmektedir. Yani fizyolojik veriler senkron çalışırken RFID sistem asenkron şekilde çalışabilmektedir. Kişiden fizyolojik veriler alınırken istenilen bir zamanda RFID etiket okuyucuya yaklaştırılabilmektedir. İletim mesafesindeki RFID etiket kişiye ait kişisel bilgileri temsil eden 64 bitlik seri numarasını okuyucuya göndermektedir. Okuyucuda bu bilgiyi seri port üzerinden bilgisayar ortamına aktarmaktadır. Seri port üzerinden okunan bu değeri Node.js platformunda ‘data.includes()’ yapısı ile fizyolojik verilerden ayrıştırılıp veri tabanında ilgili yere yazılmaktadır. Şekil 4.31’de veri tabanına kişisel verinin nasıl yazıldığı görülmektedir. Node.js ortamında ayrıştırılan bu veri gerçek zamanlı izleme sistemine de gömülmüştür. İzleme ekranında ‘Hasta

Adı' yazan bölüme RFID etiket okutulduğu anda kişinin ismi yazdırılmaktadır. Böylece sunucu tarafında kişiye ait fizyolojik veriler ile kişisel bilgiler eşleştirilmiş olmaktadır. Şekil 4.37'de 'Gerçek Zamanlı Gelen Veriler' yazısının altında yazan hasta ismi ve ona ait fizyolojik veriler görülmektedir.

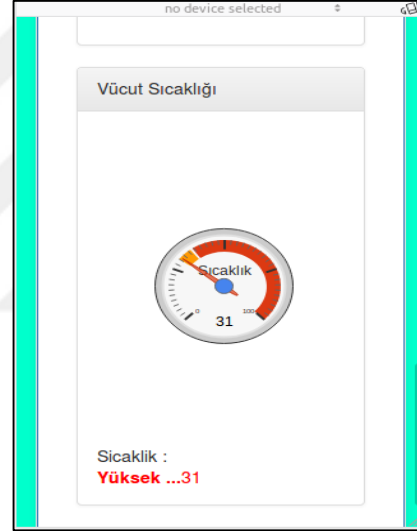


Şekil 4.37. Gerçek zamanlı sistemin çalışma ekranı (hasta ismi ve fizyolojik veriler ile birlikte)

Gerçek zamanlı izleme sisteminin HTML dosyasında kullanılan bootstrap ve Css kütüphaneleri sayesinde tarayıcı ekranı dinamik hale getirilmiştir. İzleme sistemi mobil ekran üzerinde izlenmek istenirse grafikler bu ekrana göre adaptif olarak değişmekte ve ekrana sığacak hale gelmektedir. Aynı durum tablet ekranlar içinde geçerlidir. Ayrıca bilgisayar ortamında da tarayıcı ekranın boyutunun değiştirilmesiyle birlikte grafikler adaptif olarak ekranda görülen kısma göre değişmektedir. Şekil 4.38 ve Şekil 4.39'da sırasıyla nabız ve sıcaklık değerlerinin mobil ekran görüntüleri verilmektedir.



Şekil 4.38. İzleme sistemi nabız grafiği mobil ekran görüntüsü



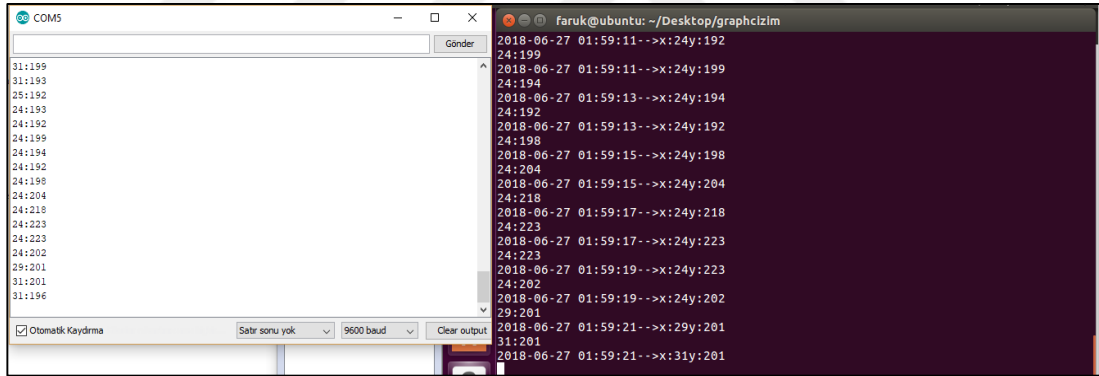
Şekil 4.39. İzleme sistemi vücut sıcaklığı grafiği mobil ekran görüntüsü

4.3. Gerçek Zamanlı Sistemin QoS Değerlendirmesi

Doktora tez çalışmasına yönelik gerçekleştirilen gerçek zamanlı IoT tabanlı sistem içinde servis kalitesi (QoS) gereksinimleri için inceleme yapılmıştır. Gerçek zamanlı çalışan sistemde benzetimden farklı olarak EKG verileri yerine kalp atım oranı (nabız) değerlerinin ve vücut sıcaklığı değerlerinin uçtan uca gecikme değerleri incelenmiştir. Bu değer incelenirken algılayıcı düğüm tarafında algılanan verinin seri portuna gönderildiği andan, bilgisayarda bulunan uygulama yazılımında veri tabanına yazıldığı ana kadar olan zaman ölçüt olarak alınmıştır. Tablo 1.1’de nabız değeri için uçtan uca

gecikme değerinin 300 ms'nin altında olması gerektiği görülmektedir. Ancak bu durum nabız bilgisinin ham değerin alınıp iletdikten sonra son kısımda hesaplanması durumunda geçerli olmaktadır. Tez çalışmasında nabız bilgileri Arduino'da toplanıp, hesaplandıktan sonra iletiği için, sıcaklık veya SpO₂ gibi 3 saniyenin altında gecikmelerin uçtan uca gecikme gereksinimini sağladığı değerlendirilmiştir.

Algılayıcı kısmında algılanan vücut sıcaklığı ve nabız bilgileri, XBee modüllerinin ayarlarından dolayı saniyede bir seri porta gönderilmektedir. Koordinatör düğümden alınan bu bilgiler bekletilmeden bilgisayar ortamına aktarılmaktadır. Node.js platformunda okunan bu veriler ayrıştırıldıktan sonra ortalama 1,5 saniye civarında veri tabanına yazılmaktadır. QoS gereksinimlerine göre vücut sıcaklığı ve nabız bilgileri 3 sn altında gerçekleşen gecikme değerlerinde doğru kabul edilmektedir. Şekil 4.40'da algılayıcı düğüm tarafında seri porta bırakılan değerler ile bu değerlerin veri tabanına yazılma zamanları görülmektedir.



```
COM5
Gönder
31:199
31:193
25:192
24:190
24:192
24:199
24:194
24:192
24:198
24:204
24:218
24:223
24:202
29:201
31:201
31:196

Faruk@ubuntu: ~/Desktop/graphcizim
2018-06-27 01:59:11 -->x:24y:192
24:199
2018-06-27 01:59:11 -->x:24y:199
24:194
2018-06-27 01:59:13 -->x:24y:194
24:192
2018-06-27 01:59:13 -->x:24y:192
24:193
2018-06-27 01:59:15 -->x:24y:198
24:204
2018-06-27 01:59:15 -->x:24y:204
24:218
2018-06-27 01:59:17 -->x:24y:218
24:223
2018-06-27 01:59:17 -->x:24y:223
24:223
2018-06-27 01:59:19 -->x:24y:223
24:202
2018-06-27 01:59:19 -->x:24y:202
29:201
2018-06-27 01:59:21 -->x:29y:201
31:201
2018-06-27 01:59:21 -->x:31y:201
```

Şekil 4.40. Algılayıcı düğümden gönderilen değerlerin veri tabanına yazılma zamanları

Şekil 4.40'da görüldüğü gibi, algılayıcı düğüm tarafından algılanıp seri porta gönderilen son dört değer referans alındığında meydana gelen değişiklikler iki saniye gecikme ile veri tabanına yazılmaktadır. Algılayıcı düğüm tarafından saniyede bir gönderilen veriler koordinatör düğüm tarafında RFID kart okuyucunun çalışma fonksiyonlarından dolayı 1–2 sn arasında verileri alıp bilgisayara aktarabilmektedir. QoS gereksinimlerine göre nabız ve vücut sıcaklığı değerleri için üç saniyenin altındaki gecikmeler makul görülmektedir. Elde edilen değerlere göre gerçek zamanlı

sistemin QoS gereksinimlerinden uçtan uca gecikme kriterlerini sağladığı görülmektedir.

4.4. Sonuç

Bu bölümde doktora tez çalışması çerçevesinde önerilen IoT tabanlı olarak gerçekleştirilen gerçek zamanlı sistemin hangi yazılım ve donanım bileşenlerini içerdiği ve sistemin nasıl çalıştığı anlatılmıştır. Gerçek zamanlı sistemde, vücut sıcaklığı ile benzetim ortamından farklı olarak EKG bilgileri yerine nabız bilgisi ölçülmüştür. Algılayıcı düğümden algılanan fizyolojik veriler koordinatör düğümdeki KVAA erişim noktasına başarıyla iletilmiştir. Koordinatör düğüm üzerinde birleştirilen KVAA erişim noktası ve RFID sistemi sorunsuz bir şekilde çalışmıştır. KVAA ile algılanan fizyolojik veriler ile RFID sistemden elde edilen kişisel bilgiler sunucu üzerinde eşleştirilmiş gelen veriler izleme sistemiyle grafikler üzerinden başarılı bir şekilde takip edilmiştir. Elde edilen sonuçlara göre gerçekleştirilen sistemin benzetim ortamındaki gibi servis kalitesi (QoS) gereksinimlerinden uçtan uca gecikme kriterlerini sağladığı görülmüştür.

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, biyomedikal uygulamalar için IoT tabanlı gerçek zamanlı veri toplama ve izleme platformu geliştirilmiştir. Geliştirilen platformun belirli bir senaryo için benzetimi ve gerçek zamanlı uygulaması gerçekleştirilmiştir.

Platform için yapılan benzetim çalışmasında, içinde KVAA, RFID sistemi, hücresel ağ, veritabanı ve bir web sunucusu bulunan yeni bir IoT tabanlı sağlık hizmetleri sistemi anlatılmaktadır. Benzetim çalışmasında, QoS desteği ile fizyolojik sinyallerin iletimini gerçekleştirmek üzere RFID-GSM arayüzünün kullanıldığı yeni bir PANc cihazı geliştirilmiştir. Tez çalışmasındaki sistem Riverbed Modeler kullanılarak modellenmiş ve benzetimi gerçekleştirilmiştir. Benzetim sistemin performansı üç farklı senaryo altında değerlendirilmiştir. Sistemin bileşenlerini doğrulamak için, algılamayı (düğümler üzerinden), kimlik doğrulamayı (sunucu aracılığıyla) ve EKG'nin ve vücut sıcaklığı sinyallerinin (PANc) aktarımlarını içeren, bunların yanı sıra kişisel bilgileri de içeren gerçekçi veriler ile benzetim çalışması gerçekleştirilmiştir. Benzetim sonuçlarına göre, ISO / IEEE 11073 standardı tarafından belirlenen QoS gereksinimlerindeki veri hızı ve gecikme sürelerinin, süper çerçeve yapısındaki dilimlerin (CAP veya GTS) etkin biçimde kullanılmasıyla enerji verimli bir şekilde sağlandığı gözlenmiştir. Düşük trafik yoğunluğuna sahip fizyolojik verilerin iletiminde CAP dilimlerinin kullanımı uygun olsada özellikle zaman-kritik verilerin iletimi durumunda (yüksek trafik yoğunluğu) mutlaka GTS tahsisi yapılarak verilerin CFP kanalları üzerinden iletilmesi sonucuna varılmıştır.

Doktora tez çalışmasında önerilen platformun gerçek zamanlı uygulaması çeşitli donanım ve yazılım bileşenleri kullanılarak gerçekleştirilmiştir. Gerçek zamanlı uygulama kısmında benzetim çalışmasına benzer olarak KVAA, RFID ve bilgisayar üzerinde bulunan sunucu bileşenleri yer almaktadır. Gerçek zamanlı geliştirilen sistemde, veri toplama kolaylığı sebebiyle nabız verileri (benzetim çalışmasından farklı olarak EKG verileri yerine) KVAA ile temin edilip izleme sistemine aktarılmıştır.

Benzetim çalışmasında olduğu gibi KVAA erişim noktası ile RFID okuyucu koordinatör düğüm (PANc) üzerinde birleştirilmiştir. KVAA aracılığıyla toplanan fizyolojik veriler ile RFID sistem tarafından elde edilen kişisel bilgiler bilgisayar üzerinde çalışan ağ tabanlı izleme sistemine başarıyla aktarılmıştır. Gerçek zamanlı veriler, kullanılan yazılım bileşenleri sayesinde, sisteme kayıtlı kullanıcılar tarafından kolaylıkla takip edilebilmektedir. İzleme sistemine gelen gerçek zamanlı veriler, belirlenen anormal durumlara göre işlenerek sistemde kayıtlı olan kullanıcılar uyarılmaktadır. Örneğin; hasta nabızının düşmesi veya yükselmesi durumunda sistem veriyi gerçek zamanlı işleyerek grafikler üzerinden kullanıcıyı ‘Yüksek’ veya ‘Düşük’ nabız şeklinde uyarabilmektedir. Ayrıca elde edilen sonuçlara göre, geliştirilen gerçek zamanlı sistemin QoS gereksinimlerinden uçtan uca gecikme kriterini başarıyla sağladığı görülmüştür.

Doktora tezinde önerilen platformun tüm bileşenleriyle hem benzetim ortamında hem de gerçek zamanlı uygulamada başarıyla çalıştığı görülmüştür. Ayrıca, ilgili fizyolojik sinyaller, hem benzetim ortamında hem de gerçek zamanlı uygulamada QoS gereksinimlerinde belirlenen kriterlere göre başarıyla iletilmiştir. Hastalara ait kişisel bilgiler her iki ortamda da RFID sistem ile elde edilerek uzak sunucuda fizyolojik veriler ile başarıyla eşleştirilmiştir.

Tez çalışmasında gerçekleştirilen sistem, Apache Kafka, Apache Spark vb. gerçek zamanlı veri analitiği platformları ile birleştirilerek daha kapsamlı bir sağlık yönetim sistemine dönüştürülebilir.

Gelecek çalışmalarda, hastalardan elde edilen gerçek zamanlı verilerin analizi yapılarak doktorlara teşhis ve tedavi konusunda yardımcı olacak bir sistem tasarlanması düşünülmektedir. Örneğin; hastanın nabız bilgilerinin gün içindeki durumu gözlemlenerek, geliştirilecek sistem ile hasta için kısa veya orta vadede tedavi önerisinde bulunulabilecektir.

KAYNAKLAR

- [1] Kırbaş İ., Dinamik Lider Düğüm Ve Çakışma Önleme Temelli Kablosuz Vücut Alan Ağı Oek Protokolü Tasarımı Ve Gerçeklemesi, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya, 2013, 418559.
- [2] Atzori L., Iera A., Morabito G., The Internet of Things: A survey, *Computer Networks*, 2010, **54**(15), 2787–2805.
- [3] Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash, M., Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys and Tutorials*, 2015, **17**(4), 2347–2376.
- [4] Pang Z., Tian J., Ecosystem Analysis in the Design of Open Platform Based In-Home Healthcare Terminals Towards the Internet-of-Things, *15th International Conference on Advanced Communication Technology*, PyeongChang, South Korea, 27-30 January 2013.
- [5] Yang G., Xie L., Mäntysalo M., Zhou X., Pang Z., Xu L. D., Zheng L. R., A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor, and Intelligent Medicine Box, *IEEE Transactions on Industrial Informatics*, 2014, **10**(4), 2180–2191.
- [6] Luo J., Chen Y., Tang K., Luo J., Remote Monitoring Information System and Its Applications Based on the Internet of Things, *International Conference on Future Biomedical Information Engineering*, Sanya, China, 13-14 December 2010.
- [7] Cubo J., Nieto A., Pimentel E., A Cloud-Based Internet of Things Platform for Ambient Assisted Living, *Sensors*, 2014, **14**, 14070-14105.
- [8] Rohokale V. M., Prasad N. R., Prasad, R., A Cooperative Internet of Things (IoT) for Rural Healthcare Monitoring and Control, *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology (Wireless VITAE)*, Chennai, India, 28 February-3 March 2011.
- [9] Movassaghi S., Abolhasan M., Lipman J., Smith D., Jamalipour, A., Wireless Body Area Networks: A Survey, *Ieee Communications Surveys and Tutorials*, 2014, **16**(3), 1658–1686.
- [10] Yuce M. R., Implementation of Wireless Body Area Networks for Healthcare Systems, *Sensors and Actuators, A: Physical*, 2010, **162**(1), 116–12.

- [11] Lee D. S., Bhardwaj S., Alasaarela E., Chung W. Y., An ECG Analysis on Sensor Node for Reducing Traffic Overload in U-Healthcare with Wireless Sensor Network, *Proceedings of IEEE Sensors*, Atlanta, USA, 28-31 October 2007.
- [12] Hu F., Xiao Y., Hao Q., Congestion-Aware, Loss-Resilient Bio-Monitoring Sensor Networking for Mobile Health Applications, *IEEE Journal on Selected Areas in Communications*, 2009, **27**(4), 450–465.
- [13] Rahim M. R. A., Rashid R. A., Ariffin S. H. S., Faisal N., Sarijari M. A., Abdul Hamid A. H. F., Testbed Design for Wireless Biomedical Sensor Network (WBSN) Application, *IEEE Conference on Computer Applications and Industrial Electronics ICCAIE 2011*, Penang, Malaysia, 4-7 December 2011.
- [14] Gündoğdu K., Çalhan, A., An Implementation of Wireless Body Area Networks for Improving Priority Data Transmission Delay, *Journal of medical systems*, 2016, **40**(3), 75.
- [15] Zhang Z., Hu X., ZigBee Based Wireless Sensor Networks and Their Use in Medical and Health Care Domain, *Seventh International Conference on Sensing Technology (ICST)*, Wellington, New Zealand, 3-5 December 2013.
- [16] Zigbee Alliance, <http://www.zigbee.org/>, (Ziyaret Tarihi: 12 Haziran 2016)
- [17] Rouse M., Internet Protocol, TechTarget Network, <http://searchunifiedcommunications.techtarget.com/definition/Internet-Protocol>, (Ziyaret Tarihi: 12 Haziran 2016)
- [18] Sevin A., Bayilmis C., Kirbas I., Design and Implementation of a New Quality of Service-Aware Cross-Layer Medium Access Protocol for Wireless Body Area Networks, *Computers & Electrical Engineering*, 2016, **56**, 145-156.
- [19] Tachtatzis C., Di Franco F., Tracey D. C., Timmons N. F., Morrison J., An Energy Analysis of IEEE 802.15.6 Scheduled Access Modes for Medical Applications, *Ad Hoc Networks*, 2012, **89**, 209–222.
- [20] He D., Zeadally, S., An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography, *IEEE Internet of Things Journal*, 2015, **2**(1), 72–83.
- [21] Jara A. J., Belchi F. J., Alcolea A. F., Santa J., Zamora-Izquierdo M. A., Gomez-Skarmeta A. F., A Pharmaceutical Intelligent Information System to Detect Allergies and Adverse Drugs Reactions Based on Internet of Things, *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, Mannheim, Germany, 29 March-2 April 2010.
- [22] Celentano L. J., RFID-Assisted Wireless Sensor Networks for Cardiac Tele-Healthcare, Yüksek Lisans Tezi, Rochester Teknoloji Enstitüsü, Rochester, 2007.

- [23] Çetin H. E., Kablosuz Sensör Ağlarının MicaZ Tabanlı Biyomedikal Uygulaması, Yüksek Lisans Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, 2009, 256481.
- [24] Yang L., Vyas R., Rida A., Pan J., Tentzeris M. M., Wearable RFID-Enabled Sensor Nodes for Biomedical Applications, *IEEE Electronic Components and Technology Conference (ECTC)*, Lake Buena Vista, USA, 27-30 May 2008.
- [25] Aktaş F., Çeken C., Erdemli Y. E., IoT-Based Healthcare Framework for Biomedical Applications, *Journal of Medical and Biological Engineering*, DOI:10.1007/s40846-017-0349-7
- [26] N. Kutup, Nesnelerin İnterneti; 4H Her Yerden, Herkesle, Her Zaman, Her Nesne ile Bağlantı, *16. Türkiye'de İnternet Konferansı inet-tr'11*, İzmir, Türkiye, 30 Kasım-2 Aralık 2011.
- [27] Bozkurt A., M2M Uygulamaları ve Türkiye, Bilişim, Türkiye Bilişim Derneği, <http://www.bilisimdergisi.org/s159>, (Ziyaret Tarihi: 12 Haziran 2018)
- [28] Dilek S., Nesnelerin İnterneti Tabanlı Uzaktan Sağlık İzleme Uygulaması, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2013, 460930.
- [29] Aktaş F., Kavuş E., Kavuş Y., A Real-Time Infant Health Monitoring System for Hard of Hearing Parents by Using Android-Based Mobil Devices, *Istanbul University Journal of Electrical and Electronics Engineering, IUJEEE*, 2017, **17**(1), 3107-3112.
- [30] Aktaş F., Çeken C., Erdemli Y. E., Nesnelerin İnterneti Teknolojisinin Biyomedikal Alanındaki Uygulamaları, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2016, **4**, 37-54.
- [31] Ateş M. R., Nesnelerin İnterneti, Gelecek & Trendler, Siemens, http://Siemens.e-dergi.com/pubs/gelecek_ve_trendler/2013/2013_02.html, (Ziyaret tarihi: 08.06.2014).
- [32] Gubbi J., Buyya R., Marusic S., Palaniswami M., Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions, *Future Generation Computer Systems*, **29**(7), 2013, 1645-1660.
- [33] Saleem S., Ullah S., Kwak K. S., A Study of IEEE 802.15.4 Security Framework for Wireless Body Area Networks, *Sensors*, 2011, **11**(2), 1383-1395.
- [34] Farahani S., *Zigbee Wireless Networks and Transceivers*, Elsevier, USA, 2009.
- [35] Jurcik P., Koubaa A., The IEEE 802.15.4 OPNET Simulation Model: Reference Guide v2.0, *IPP HURRAY Research Group*, Tech. Rep. TR-070509, 1-13, 2009.

- [36] Jurcik P., Koubâa A., Alves M., Tovar E., A Simulation Model for the IEEE 802.15.4 Protocol: Delay / Throughput Evaluation of the GTS Mechanism, *Proceedings in 15th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2007)*, Istanbul, Turkey, 24-26 October 2007.
- [37] Pala Z., RFID Teknolojisinin Acil Müdahalede Kullanımı, *Akademik Bilişim '09*, Şanlıurfa, Türkiye, 11-13 Şubat 2009.
- [38] Pala Z., RFID Teknolojisi İle Otomasyon Bir Uygulama Olarak: Otopark Takibi, Yüksek Lisans Tezi, Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü, Van, 2007, 199670.
- [39] Tuğaç B., Radyo Frekansı ile Kimlik Tanıma-RFID, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007, 201519.
- [40] R. Vaddi, R. P. Agarwal, S. Dasgupta, T. T. Kim, Design and Analysis of Double-Gate Mosfets for Ultra-Low Power Radio Frequency Identification (Rfid): Device and Circuit Co-Design, *Journal of Low Power Electronics and Applications*, 2011, **1**(2), 277-302.
- [41] Rao S. S., Adaptive Slot Adaptive Frame (Asaf) Aloha Protocol - An Introduction, Shodhganga, http://shodhganga.inflibnet.ac.in/bitstream/10603/19096/11/11_chapter%203.pdf, (Ziyaret tarihi: 04.01.2017).
- [42] Riverbed Technology, OPNET Technologies – Network Simulator, Riverbed, <http://www.riverbed.com/gb/products/steelcentral/opnet.html?redirect=opnet>, (Ziyaret tarihi: 28.03.2017).
- [43] Irmak E., Kose A., Gocmen G., Simulation and Zigbee Based Wireless Monitoring of the Amount of Consumed Energy at Smart Homes, *5th International Conference on Renewable Energy Research and Applications (ICRERA 2016)*, Paris, France, 14-17 October 2016.
- [44] Asaduzzaman A., Chidella K. K., Mridha M. F., A Time and Energy Efficient Parking System Using Zigbee Communication Protocol, *Proceedings of the IEEE SoutheastCon 2015*, Fort Lauderdale, USA, 9-12 April 2015.
- [45] MicaZ, Crossbow, http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf, (Ziyaret tarihi: 04.01.2017).
- [46] Zheng F., Kaiser T., (2016). Adaptive Aloha Anti-Collision Algorithms for RFID Systems, *EURASIP Journal on Embedded Systems*, 2016, **2016**(7), 1-14.
- [47] Dobkin D. M., The Rf in RFID: Physical Layer Operation of Passive UHF Tags and Readers, Enigmatics, http://www.enigmatic-consulting.com/Communications_articles/RFID/RFID_protocols.html, (Ziyaret tarihi: 28.03.2017).
- [48] Marino F., Massei G., Paura L., Modeling and Performance Simulation of EPC Gen2 RFID on OPNET, *IEEE International Workshop on Measurements and Networking Proceedings (M&N)*, Naples, Italy, 7-8 October 2013.

- [49] Robotik Sistem, ARDUINO/GENUINO UNO, http://www.robotiksistem.com/arduino_uno_ozellikleri.html, (Ziyaret tarihi: 10.06.2018).
- [50] Store Home, Arduino Uno Rev3, <https://store.arduino.cc/usa/arduino-uno-rev3>, (Ziyaret tarihi: 10.06.2018).
- [51] Digi, XBee-PRO S1 802.15.4 extended-range module w/U.fl connector, <https://www.digi.com/products/models/xbp24-aui-001>, (Ziyaret tarihi: 11.06.2018).
- [52] Thomarmax, Example Zigbee nodes, http://thomarmax.github.io/QtXBee/doc/pre_alpha/XBee_zigbee_basic_concepts.html, (Ziyaret tarihi: 11.06.2018).
- [53] Lezzetli Robot Tarifleri, MLX90614 - Kızılötesi Temassız Sıcaklık Sensörü, <http://lezzetlirobottarifleri.com/lrtsosyal/konu-mlx90614-kizilotesi-temassiz-sicaklik-sensoru.html>, (Ziyaret tarihi: 03.06.2018).
- [54] Özgören M., Şakar M., Öniz A., Vücut Sıcaklığının Gözlenmesi İçin Temaslı/Temassız Sensör Ağı, *15. Biyomedikal Mühendisliği Ulusal Toplantısı BİYOMUT 2010*, Antalya, Türkiye, 21-24 Nisan 2010.
- [55] Aktaş F., Kavuş E., Kavuş Y., Duyma Engelli Ebeveynler için Gerçek Zamanlı Bebek Sağlık İzleme Sistemi, *Tıp Teknolojileri Kongresi TIPTEKNO'16*, Antalya, Türkiye, 27-29 Ekim 2016.
- [56] Pulse Sensor, PulseSensor.com, <https://pulsesensor.com/products/pulse-sensor-amped>, (Ziyaret tarihi: 03.06.2018).
- [57] Jojo, Pulse Sensor and Arduino-Inteerfacing, <http://www.circuitstoday.com/pulse-sensor-arduino>, (Ziyaret tarihi: 07.06.2018).
- [58] RFID Okuyucu ve Etiketler, Rc522 RFID Okuyucu 13,56 MHz, <https://www.direnc.net/rc522-rfid-okuyucu-1356mhz>, (Ziyaret tarihi: 08.06.2018).
- [59] RF, XBee, Wifi, RFID Kartları, MRFC-522 RFID Modülü, <http://www.robishop.com/mfrc-522-rfid-modulu>, (Ziyaret tarihi: 07.06.2018).
- [60] Aktaş F., Kablosuz Algılayıcı/Eyleyici Ağlarla Denetim Sistemi Tasarımı, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli, 2012, 301635.
- [61] Arduino Projects, Pulse Sensor with Arduino, <http://www.theorycircuit.com/pulse-sensor-arduino/>, (Ziyaret tarihi: 07.06.2018).
- [62] Yalçın M., Arduino IDE Nedir?, <http://www.dudigan.com/2015/12/arduino-ide-nedir/>, (Ziyaret tarihi: 10.06.2018).
- [63] Support/Product Detail, XCTU Software, <https://www.digi.com/support/productdetail?pid=3352>, (Ziyaret tarihi: 11.06.2018).

- [64] RF Kits Common, XBee transparent mode, [http://docs.digi.com/display/RF KitsCommon/XBee+transparent+mode](http://docs.digi.com/display/RF+KitsCommon/XBee+transparent+mode), (Ziyaret tarihi: 12.06.2018).
- [65] Özkaya Ö., XBee Kullanımı, ozenozkaya.com/blog/?p=346, (Ziyaret tarihi: 12.06.2018).
- [66] Kasal Ö., *Node.js*, 1. Baskı, Dikeyksen, İstanbul, 2016.
- [67] Node.js Foundation, About Node.js, <http://nodejs.org/en>, (Ziyaret tarihi: 13.06.2018).
- [68] Savaşer S., *Hemşireler için Çocuk Sağlığı ve Hastalıkları Öğrenim Rehberi*, 1. Baskı, İstanbul Tıp Kitabevi, İstanbul, 2000.





EKLER

Ek-A

Web tabanlı uygulama yazılımını Node.js platformunda verilerin seri porttan okunması için yazılan 'server.js' dosyası içeriği:

```
var express = require('express');
var app= express();
var path=require('path');
var http = require('http').Server(app);
var io = require('socket.io')(http);
app.use('/public', express.static(path.join(__dirname, 'public')));
var monk = require("monk");
var should = require("should");
var SerialPort = require("serialport");
var dateFormat = require('dateformat');
var DevicePort = "/dev/ttyUSB0";
var serialPort = new SerialPort(DevicePort, {
baudrate: 9600,
parser: require("serialport").parsers.readline("\n")});
var db = monk('localhost/arduino');
should.exists(db);
var collection = db.get("sicaklik");
should.exists(collection);
io.on('connection', function(socket) {
socket.on('disconnect', function(){
});});
serialPort.on("open", function ()
{serialPort.on('data', function(data) {
console.log(data);
var date = new Date();
var dataArray = data.split(':');
var z="";
console.log(dateFormat(date.getTime(),"yyyy-mm-dd HH:MM:ss")+ '--
>x:'+dataArray[0]+'y:'+dataArray[1]);
```

```

var temp=dateFormat(date.getTime(),"yyyy-mm-dd HH:MM:ss")+ '--
>x:'+dataArray[0]+'y:'+dataArray[1];
var x=dataArray[0]
var y=dataArray[1];
if(data.includes("E2 E0 1A 2B"))
{z="FarukAktas";}
else if(data.includes("67 41 B0 AB"))
{z="GulayAktas";}
io.emit('alldata', data);
collection.insert({"time":dateFormat(date.getTime(),"yyyy-mm-dd
HH:MM:ss"),"x":dataArray[0],"y":dataArray[1],"z": z}, function(err,doc)
{if(err){
console.log("HATA");
} });});});
app.get('/', function(req, res){
res.sendFile(__dirname + '/index.html'); });
var port=8000;
http.listen(port, function(){
console.log('Listening ' + port);
});

```

Ek-B

Gerçek zamanlı hasta izleme sistemi için yazılan HTML tabanlı kod parçası:

```
<!doctype html>
<html><head>
<title>IoT Veri İzleme Sistemi</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script type="text/javascript" src="/public/libs/jquery/jquery-3.2.1.min.js"></script>
<script type="text/javascript"
src="/public/libs/bootstrap/js/bootstrap.min.js"></script>
<link rel="stylesheet" href="/public/libs/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="/public/libs/bootstrap/css/bootstrap-theme.min.css">
<link href="/public/libs/font-awesome/css/font-awesome.min.css" rel="stylesheet">
<style>
#path {
position:absolute;width:746px;height:79px;z-index:1;left: 174px;top: 51px;
text-align: center;
background-color: #F90;}
#messages {position:absolute;width:721px;height:74px;z-index:1;left: 279px;
top: 138px;text-align: center;background-color: #0F0;}
</style><style>
.ui-helper-center {
text-align: center;}
</style>
<script type="text/javascript"
src="/public/libs/jquery/dist/jquery.jqplot.min.js"></script>
<script type="text/javascript"
src="/public/libs/jquery/dist/plugins/jqplot.canvasTextRenderer.js"></script>
<script type="text/javascript"
src="/public/libs/jquery/dist/plugins/jqplot.canvasAxisLabelRenderer.js"></script>
<script type="text/javascript"
src="/public/libs/jquery/dist/plugins/jqplot.dateAxisRenderer.js"></script>
```

```

<script type="text/javascript"
src="/public/libs/jquery/dist/plugins/jqplot.logAxisRenderer.js"></script>
<script type="text/javascript"
src="/public/libs/jquery/dist/plugins/jqplot.canvasAxisTickRenderer.js"></script>
<script type="text/javascript"
src="/public/libs/jquery/dist/plugins/jqplot.highlighter.js"></script>
<script src="https://cdn.socket.io/socket.io-1.2.0.js"></script>
<link rel="stylesheet" type="text/css"
href="/public/libs/jquery/dist/jquery.jqplot.min.css" />
<link rel="stylesheet" type="text/css" href="/jquery-ui.css" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<style type="text/css">
body {background-color: #0FC;}
</style></head><body>
<div class="container">
<script type="text/javascript">
function toggle(button) {
var buttonid = "#" + button.id;
switch (button.value) {
case "ON":
button.value = "OFF";
$(buttonid).removeClass("btn-success").addClass("btn-danger");
break;
case "OFF":
$(buttonid).removeClass("btn-danger").addClass("btn-success");
button.value = "ON";
break;
}socket.emit('digitalIO', button.id, button.value == "OFF" ? 0 : 1);}
</script>
<nav class="navbar navbar-default">
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">

```

```

$(document).ready(function () {
google.charts.load('current', {'packages': ['gauge', 'corechart']});
var temp = 50;
google.charts.setOnLoadCallback(drawGauge);
function drawGauge() {
var data = google.visualization.arrayToDataTable([
['Label', 'Value'], ['Sıcaklık', temp],]);
var options = {
width: 400, height: 120, redFrom: 37, redTo: 100, yellowFrom: 30, yellowTo: 36,
minorTicks: 10};
var chart = new google.visualization.Gauge(document.getElementById('chart_div'));
chart.draw(data, options);}
var temperaturelevel = 90;
google.charts.setOnLoadCallback(drawChart); //otomatik çizdirme
function drawChart() {
var data = google.visualization.arrayToDataTable([
['Task', 'Hours per Day'], ['Normal', temperaturelevel], ['Ortalama', 100 -
temperaturelevel],]);
var options = {legend: 'none',};
var chart = new
google.visualization.PieChart(document.getElementById('piechart'));
chart.draw(data, options);}
var socket = io();
function getData(data, sensorName) {
return data.sensor[sensorName];};
var myDataax = [[0, 0]];var myDatay = [[0, 0]];var myDataaz = [[0, 0]];
var z="Hasta Adı";
socket.on('alldata', function (msg)
if(msg.includes("E2 E0 1A 2B")){
z="Faruk AKTAS";}
else if(msg.includes("67 41 B0 AB")){
z="Gulay AKTAS";}
var dataArray = msg.split(':');

```

```

$('#messages').html($('

# 


```

```

lineWidth: 1.0,color: '#0571B6',markerOptions: {style: 'diamond'}
},{lineWidth: 2,markerOptions: {style: 'circle'}},{
showLine: false,
markerOptions: {size: 7, style: "x"}},
axes: {xaxis: {renderer: $.jqplot.DateAxisRenderer,
tickRenderer: $.jqplot.CanvasAxisTickRenderer,
tickOptions: {
formatString: '%H:%M:%S', angle: -50 },
numberTicks: 10,label: 'Zaman',labelOptions: { fontFamily: 'Arial', fontSize: '12pt' },
labelRenderer: $.jqplot.CanvasAxisLabelRenderer},
yaxis: { min: -100,max: 250,tickOptions: {formatString: '%.2f',angle: -30},
tickRenderer: $.jqplot.CanvasAxisTickRenderer,
labelRenderer: $.jqplot.CanvasAxisLabelRenderer,
numberTicks: 15,labelOptions: { fontFamily: 'Arial',fontSize: '14pt'},
labelRenderer: $.jqplot.CanvasAxisLabelRenderer}},
cursor: {zoom: false,showTooltip: false,show: false},
highlighter: {useAxesFormatters: true,showMarker: true,show: false},
grid: {drawGridLines: true,gridLineColor: '#cccccc', background: '#fffd6',
borderColor: '#999999', borderWidth: 2.0,shadow: true,shadowAngle: 45,
shadowOffset: 1.5,shadowWidth: 3,shadowDepth: 3,shadowAlpha: 0.07,
renderer: $.jqplot.CanvasGridRenderer }});
plot1.series[0].data = myDataax;
plot1.resetAxesScale();
plot1.axes.xaxis.numberTicks = 10;
plot1.axes.y2axis.numberTicks = 15;
plot1.replot();
function updateSeries() {
myDataax.splice(0, myDataax.length - 10);
plot1.series[0].data = myDataax;
plot1.resetAxesScale();
plot1.axes.xaxis.numberTicks = 10;
plot1.axes.y2axis.numberTicks = 15;
plot1.replot();}

```

```

window.setInterval(updateSeries, 1000;});
</script><div class="row"><div class="col-md-12">
<div class="panel panel-primary"><div class="panel-heading">
<h3 class="panel-title" align="center">Gerçek Zamanlı Veri İzleme Sistemine
Hoşgeldiniz </h3>
</div><div class="panel-body"> <div class="row"><div class="col-md-12">
<div class="panel panel-primary"><div class="panel-heading">
<h3 class="panel-title">Gerçek Zamanlı Gelen Veriler</h3>
<div id="kullanicibilgi">
“”
</div></div></div></div></div></div></div></div></div>
<div class="row"><div class="col-md-12"><div class="panel panel-primary ">
<div class="panel-heading "><h3 class="panel-title">Gerçek Zamanlı Fizyolojik
Veriler</h3>
</div><div class="panel-body"><div class="col-md-6 ">
<div class="panel panel-default" id="icerik">
<div class="panel-heading">Nabız Bilgisi</div>
<div class="panel-body"><div class="plot" style="height: 300px;" id="chart1">
</div></div></div>
Nabız :
<div id="nabiz"></div></div><div class="col-md-3">
<div class="panel panel-default" id="icerik1">
<div class="panel-heading">Vücut Sıcaklığı
</div><div class="panel-body"><div id="piechart"
style="margin-top:55px; margin-bottom:55px; height:190px; "></div>
</div></div></div><div class="col-md-3 "><div class="panel panel-default"
id="icerik2">
<div class="panel-heading"> Vücut Sıcaklığı</div><div class="center-block">
<div class=" panel-body"><div id="chart_div"
style="margin-top:90px; margin-bottom:10px; margin-left:40px; height:
200px;"></div>
Sicaklik :

```


Ek-C

Riverbed Modeller yazılımındaki RFID sistemine ait kodlar aşağıda verilmiştir.

RFID Okuyucu Init Kodu:

```
read_Mod=OPC_TRUE;
readCompletedTagCount=0;
TotalTagCount=2;
Query_rep_count=0;
kill_status=OPC_FALSE;
readTotalTagCount=OPC_FALSE;
my_id = op_id_self();
my_node_id = op_topo_parent (my_id);
op_ima_obj_attr_get (my_node_id, "TerminalAddress", &terminalAddress);
tx_id = op_topo_assoc (my_id, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_RATX, 0);
comp_id = op_topo_child (tx_id, OPC_OBJTYPE_COMP, 0);
tx_ch_id = op_topo_child (comp_id, OPC_OBJTYPE_RATXCH, 0);
```

RFID Okuyucu Query Kodu:

```
query_pkt= op_pk_create_fmt("fa_query_packet_format");
op_pk_nfd_set (query_pkt, "Q",3);
op_pk_nfd_set (query_pkt, "Sel", 10);
op_pk_send (query_pkt, TX_OUT_STRM);
```

RFID Okuyucu Query_Rep Kodu:

```
if (Query_rep_count>5)
{query_adj_pkt= op_pk_create_fmt("fa_queryAdjust_packet_format");
op_pk_nfd_set (query_adj_pkt, "UpDn",2);
op_pk_send (query_adj_pkt, TX_OUT_STRM);
Query_rep_count=0;}
else {query_rep_pkt= op_pk_create_fmt("fa_queryRep_packet_format");
op_pk_send (query_rep_pkt, TX_OUT_STRM);
Query_rep_count++;}
```

RFID Okuyucu Paket Alma Kodu:

```
if((packet_came = op_pk_get (RX_IN_STRM))!=OPC_NIL)
{op_pk_format(packet_came,pk_format);
if (strcmp("fa_rfid_RN16_tag_packet_format",pk_format)==0){
op_pk_nfd_get(packet_came, "RN", &rn_value);
ack_pkt= op_pk_create_fmt("fa_rfid_ack_packet_format");
op_pk_nfd_set (ack_pkt, "RN16",rn_value);
op_pk_send (ack_pkt, TX_OUT_STRM);}
else if (strcmp("rfid_epc_send_packet",pk_format)==0){
op_pk_nfd_get_pkt(packet_came, "EPC_number", &receive_epc);
op_pk_nfd_get (receive_epc, "Serial_number", &came_serial_number);
read_req_packet= op_pk_create_fmt("fa_read_packet_format");
op_pk_nfd_set (read_req_packet, "RN",rn_value);
op_pk_send (read_req_packet, TX_OUT_STRM);}
else if (strcmp("fa_read_reply_tag_packet_format",pk_format)==0){
op_pk_nfd_get(packet_came, "personID", &personel_id);
if(personel_id==1){op_pk_nfd_get(packet_came, "Memory_Words",
&read_value1);
op_pk_nfd_get(packet_came, "Yas", &read_value_yas1);
read_value_pkt1= op_pk_create_fmt("read_value_packet");
op_pk_nfd_set (read_value_pkt1, "Read_value",read_value1);
op_pk_nfd_set (read_value_pkt1, "Yas",read_value_yas1);
op_pk_nfd_set (read_value_pkt1, "IDperson",personel_id);
copy_read_value_pkt1=op_pk_copy(read_value_pkt1);
op_pk_send (read_value_pkt1, SINK_OUT_STRM);
op_pk_send (copy_read_value_pkt1, STRM_FROM_READER_TO_BASE);}
else if (personel_id==2){
op_pk_nfd_get(packet_came, "Memory_Words", &read_value2);
op_pk_nfd_get(packet_came, "Yas", &read_value_yas2);
read_value_pkt2= op_pk_create_fmt("read_value_packet");
op_pk_nfd_set (read_value_pkt2, "Read_value",read_value2);
op_pk_nfd_set (read_value_pkt2, "Yas",read_value_yas2);
op_pk_nfd_set (read_value_pkt2, "IDperson",personel_id);
```

```

copy_read_value_pkt2=op_pk_copy(read_value_pkt2);
op_pk_send (read_value_pkt2, SINK_OUT_STRM);
op_pk_send (copy_read_value_pkt2, STRM_FROM_READER_TO_BASE);}
readComplatedTagCount++;
if(readComplatedTagCount==TotalTagCount)
readTotalTagCount=OPC_TRUE;
else
timeOutEndEvh=op_intrpt_schedule_self((op_sim_time()+.01),tIMEoUTINTcODE;
}}

```

RFID Etiket Init Kodu:

```

Qvalue=0;Qrange=0;came_select=OPC_FALSE;
came_query=OPC_FALSE;reply_state=OPC_FALSE;
reply_mod=OPC_FALSE;came_ack=OPC_FALSE;
came_read_req=OPC_FALSE;came_query_rep=OPC_FALSE;
arbitrate_state=OPC_FALSE;rtrn_rcv_pkt=OPC_FALSE;
read_Complated=OPC_FALSE;came_query_adj=OPC_FALSE;
came_kill=OPC_FALSE;header=0x10;
man_number=0x01F1C07;object_class=0x0101B3;
reply_was_send=OPC_FALSE;
ack_mod=OPC_FALSE;
newreply_mod=OPC_FALSE;
my_id = op_id_self();
my_node_id = op_topo_parent (my_id);
op_ima_obj_attr_get (my_node_id, "TerminalAddress", &TermID);
op_ima_obj_attr_get (my_node_id, "SerialNumber", &Seri_no);
op_ima_obj_attr_get (my_node_id, "IDno", &IDnumber);
op_ima_obj_attr_get (my_node_id, "personAge", &personAge);
tx_id = op_topo_assoc (my_id, OPC_TOPO_ASSOC_OUT,
OPC_OBJTYPE_RATX, 0);
comp_id = op_topo_child (tx_id, OPC_OBJTYPE_COMP, 0);
tx_ch_id = op_topo_child (comp_id, OPC_OBJTYPE_RATXCH, 0);
if((came_packet = op_pk_get (RX_IN_STRM))!=OPC_NIL){

```

```

op_pk_format(came_packet,pk_format);
if (strcmp("fa_select_packet_format",pk_format)==0){
came_select=1;
op_pk_nfd_get (came_packet, "Target", &Target);
if(Target==100)
SL=1;}}

```

RFID Etiket ACK Kısmı:

```

if (came_ack==OPC_TRUE){
op_pk_nfd_get(came_packet, "RN16", &came_RN16);
came_ack=OPC_FALSE;
if (random_RN16==came_RN16){
epc_pkt = op_pk_create_fmt ("rfid_epc_packet");
op_pk_nfd_set (epc_pkt, "Header", header);
op_pk_nfd_set (epc_pkt, "Man_number",man_number);
op_pk_nfd_set (epc_pkt, "Obj_class", object_class);
op_pk_nfd_set (epc_pkt, "Serial_number", Seri_no);
send_epc_pkt = op_pk_create_fmt ("rfid_epc_send_packet");
op_pk_nfd_set_pkt (send_epc_pkt, "EPC_number", epc_pkt);
op_pk_send (send_epc_pkt, TX_OUT_STRM);}}
else if(came_read_req==OPC_TRUE){
read_tag_reply= op_pk_create_fmt ("fa_read_reply_tag_packet_format");
op_pk_nfd_set (read_tag_reply, "Memory_Words", IDnumber);
op_pk_nfd_set (read_tag_reply, "Yas", personAge);
op_pk_nfd_set (read_tag_reply, "personID", TermID);
op_pk_send (read_tag_reply, TX_OUT_STRM);
came_read_req=OPC_FALSE;
read_Complated=OPC_TRUE;
ack_mod=OPC_FALSE;}}

```

RFID Etiket Query Kodu:

```

if(came_query==OPC_TRUE)
{op_pk_nfd_get(came_packet, "Q", &Qvalue);
temp_Qvalue=Qvalue;

```

```

Qrange=pow(2,Qvalue)-1;
srand(time(NULL)/TermID);
random_Qvalue=rand()%Qrange;
printf("%d. etiket Uretilen deger: %d**\n",TermID,random_Qvalue);
came_query=OPC_FALSE;}
else if (came_query_adj==OPC_TRUE){
op_pk_nfd_get(came_packet, "UpDn", &UpDn_value);
if (UpDn_value==2){
Qvalue--;
if (Qvalue==0){
Qvalue=temp_Qvalue;}
Qrange=pow(2,Qvalue)-1;
srand(time(NULL)/TermID);
random_Qvalue=rand()%Qrange;
printf("%d. etiket Yeni Uretilen deger: %d**\n",TermID,random_Qvalue);
came_query_adj=OPC_FALSE;}
else if(came_query_rep==OPC_TRUE){
random_Qvalue=random_Qvalue-1;
printf("%d. etiket Deger dusur: %d**\n",TermID,random_Qvalue);
came_query_rep=OPC_FALSE;}
if(random_Qvalue==0)
reply_state=OPC_TRUE;
else
arbitrate_state=OPC_TRUE;

```

RFID Etiket Reply Kodu:

```

RN16_pkt= op_pk_create_fmt("fa_rfid_RN16_tag_packet_format");
srand(time(NULL)/TermID);
random_RN16=rand();
op_pk_nfd_set (RN16_pkt, "RN",random_RN16);
printf("%d. etiket RN16 degeri: %d**\n",TermID,random_RN16);
op_pk_send (RN16_pkt, TX_OUT_STRM);
reply_state=OPC_FALSE;

```

```
reply_mod=OPC_TRUE;
ack_mod=OPC_TRUE;
newreply_mod=OPC_FALSE;
```

RFID Etiket Paket Alma Foksiyonu:

```
static void Receive_packet(void) {
char pk_format[50];
FIN(Receive_packet ());
if((came_packet = op_pk_get (RX_IN_STRM))!=OPC_NIL){
op_pk_format(came_packet,pk_format);
if (strcmp("fa_query_packet_format",pk_format)==0)
came_query=OPC_TRUE;
else if (strcmp("fa_rfid_ack_packet_format",pk_format)==0)
came_ack=OPC_TRUE;
else if (strcmp("fa_queryRep_packet_format",pk_format)==0)
came_query_rep=OPC_TRUE;
else if (strcmp("fa_read_packet_format",pk_format)==0)
came_read_req=OPC_TRUE;
else if (strcmp("fa_queryAdjust_packet_format",pk_format)==0)
came_query_adj=OPC_TRUE;
else if (strcmp("kill_packet",pk_format)==0)
came_kill=OPC_TRUE;}
if(ack_mod==OPC_TRUE){
if (came_ack==OPC_FALSE && came_read_req==OPC_FALSE){
newRN16evh=op_intrpt_schedule_self ((op_sim_time()+0.05),newRN16INTcODE);
if (op_ev_valid(newRN16evh)){
newreply_mod=OPC_TRUE;
op_ev_cancel(newRN16evh);}}}
FOUT;}
```

KİŞİSEL YAYIN VE ESERLER

Aktaş F., Çeken C., Erkan K., WNCsbed: A Wireless Sensor and Actuator Networks Based Testbed for Networked Control Systems, *Computer Standards & Interfaces*, 2014, **36**, 585-591.

Aktaş F., Çeken C., Erdemli Y. E., Biyomedikal Uygulamaları için Nesnelerin İnterneti Tabanlı Veri Toplama ve Analiz Sistemi, *Tıp Teknolojileri Ulusal Kongresi TIPTEKNO'14*, Nevşehir, 25-27 Eylül 2014.

Aktaş F., Çeken C., Erdemli Y. E., Kablosuz Vücut Alan Ağları Kullanılarak Fizyolojik Sinyallerin Servis Kalitesi Desteğiyle Aktarılması, *Tıp Teknolojileri Ulusal Kongresi TIPTEKNO'15*, Muğla, 15-18 Ekim 2015.

Aktaş F., Çeken C., Erdemli Y. E., Nesnelerin İnterneti Teknolojisinin Biyomedikal Alanındaki Uygulamaları, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2016, **4**, 37-54.

Aktaş F., Kavuş E., Kavuş Y., Real Time Infant Health Monitoring System for Hard of Hearing Parents, *Tıp Teknolojileri Ulusal Kongresi TIPTEKNO'16*, Antalya, 27-29 Ekim 2016.

Aktaş F., Çeken C., Erdemli Y. E., RFID Teknolojisi Kullanılarak Nesnelerin İnterneti Tabanlı Hasta Bilgi Takip Sistemi, *XX. Biyomedikal Mühendisliği Ulusal Toplantısı BİYOMUT 2016*, İzmir, 3-5 Kasım 2015.

Aktaş F., Kavuş E., Kavuş Y., A Real Time Infant Health Monitoring System for Hard of Hearing Parents by using Android-based Mobil Devices, *Istanbul University - Journal of Electrical & Electronics Engineering*, 2017, **17**(1), 3107-3112.

Aktaş F., Çeken C., Erdemli Y. E., IoT-Based Healthcare Framework for Biomedical Applications, *Journal of Medical and Biological Engineering*, 2018, DOI:10.1007/s40846-017-0349-7.

ÖZGEÇMİŞ

Faruk Aktaş 1983’de yılında Bursa’da doğdu. Lise öğrenimini Bursa’da Ali Osman Sönmez Endüstri Meslek Lisesi’nde tamamladı. Önlisans Eğitimini Marmara Üniversitesi Elektronik bölümünde tamamladı. 2004 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik Öğretmenliği bölümünden 2008 yılında mezun oldu. 2009 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik-Bilgisayar Eğitimi Anabilim Dalında yüksek lisansa başladı. “Kablosuz Algılayıcı/Eyleyici Ağlarla Denetim Sistemi Tasarımı” başlıklı yüksek lisans tezini 2012 yılında tamamladı. 2012 yılında başlamış olduğu Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Biyomedikal Mühendisliği Anabilim Dalında doktora çalışmasını “Biyomedikal Uygulamaları için Nesnelerin İnterneti Tabanlı Gerçek Zamanlı Veri Toplama ve İzleme Platformunun Geliştirilmesi” başlığı altında sürdürmektedir. Ayrıca, 2010 yılında Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümünde Araştırma Görevlisi olarak çalışmaya başladı. Halen Kocaeli Üniversitesi Teknoloji Fakültesi Biyomedikal Mühendisliği Bölümünde Araştırma Görevlisi olarak görev yapmaktadır.