



**T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ BÖLÜMÜ**

**NESNELERİN İNTERNETİNİ KULLANARAK ÜÇ
FAZLI ASENKRON MOTORUN GERÇEK
ZAMANLI İZLENMESİ VE DENETİMİ**

ALI HUSEIN BENHUSEIN

DOKTORA TEZİ

DANIŞMAN

DOÇ. DR. M. FATİH KILIÇASLAN

**T. C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**NESNELERİN İNTERNETİNİ KULLANARAK ÜÇ FAZLI
ASENKRON MOTORUN GERÇEK ZAMANLI İZLENMESİ VE
DENETİMİ**

Ali Husein BENHUSEIN

**Danışman
Jüri Üyesi
Jüri Üyesi
Jüri Üyesi
Jüri Üyesi**

**Doç. Dr. M. Fatih KILIÇASLAN
Dr.Öğr. Üyesi Erdal ŞEHİRLİ
Dr.Öğr. Üyesi Aydın TÜRKYILMAZ
Dr. Öğr. Üyesi Cevat RAHEBİ
Dr. Öğr. Üyesi Mohammad M. Gomroki**

**DOKTORA TEZİ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI**

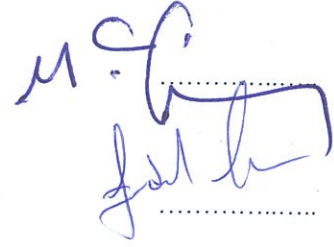
KASTAMONU – 2019

TEZ ONAYI

Ali Husein BENHUSEIN tarafından hazırlanan “Nesnelerin İnternetini Kullanarak Üç Fazlı Asenkron Motorun Gerçek Zamanlı İzlenmesi ve Denetimi” adlı tez çalışması aşağıdaki jüri üyeleri önünde savunulmuş ve oy birliği ile Kastamonu Üniversitesi Fen Bilimleri Enstitüsü Malzeme Bilimi ve Mühendisliği Ana Bilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Danışman

Doç. Dr. M. Fatih KILIÇASLAN
Kastamonu Üniversitesi



Jüri Üyesi

Dr.Öğr. Üyesi Erdal ŞEHİRLİ
Kastamonu Üniversitesi

Jüri Üyesi

Dr.Öğr. Üyesi Aydın TÜRKYILMAZ
Kastamonu Üniversitesi



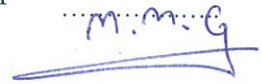
Jüri Üyesi

Dr. Öğr. Üyesi Cevat RAHEBİ
Altınbaş Üniversitesi



Jüri Üyesi

Dr. Öğr. Üyesi Mohammad M. GOMROKI
Türk Hava Kurumu Üniversitesi



8 /11/2019

Enstitü Müdürü

Doç. Dr. Nur BELKAYALI



TAAHHÜTNAME

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildirir ve taahhüt ederim.

Ali Husein BENHUSEIN



ÖZET

Doktora Tezi

NESNELERİN İNTERNETİNİ KULLANARAK ÜÇ FAZLI ASENKRON MOTORUN GERÇEK ZAMANLI İZLENMESİ VE DENETİMİ

Ali Husein BENHUSEIN
Kastamonu Üniversitesi
Fen Bilimleri Enstitüsü
Malzeme Bilimi ve Mühendisliği Bölümü

Danışman: Doç. Dr. M. Fatih KILIÇASLAN

İnternet veya insanların interneti dünya genelinde milyonlarca insanın, bilgisayarın, cep telefonunun birbirine bağlanmasına ve veri alışverişinde bulunmasına imkan sağlar. Son yıllarda Nesnelerin İnterneti (Nİ, *İng.* Internet of Things-IOT) olarak adlandırılan yeni bir teknoloji ortaya çıkmıştır. Nİ, belirli görevlerin yerine getirilmesi amacıyla birbirine internet üzerinden bağlı olan, veri iletimi veya veri alışverişinde bulunan fiziksel nesnelere oluşur. Pek çok endüstriyel uygulamada yer alan önemli unsurlardan biri üç fazlı asenkron motor kullanımınıdır. Daha iyi bir performans elde etmek amacıyla bu motorun kullanım endeksinin hesaplanabilmesi için, yine aynı motora ait parametrelerin analizi, izlenmesi ve kontrolünün sağlanması gereklidir, bu ayrıca olası hataların ve arızaların erken tespitini sağlayacaktır. Her ne kadar üretim süreçlerinde asenkron motorlarının gerçek zamanlı izlenmesi ve kontrolü için kullanılacak pek çok farklı çözüm olsa da, bu çözümlerin çoğu sınırlı veya küçük bir alanda lokal bir izleme sunmaktadır. Önerdiğimiz metotta, dünyanın her yerinden web uygulamaları ve Nİ ile online izleme ve kontrol çözümleri, gelişen teknolojiyle beraber mümkün hale gelmiştir. Nİ'nin ve web uygulamalarının bu şekilde kullanımı, izleme ve kontrol süreçlerinin etkinliğini ve verimliliğini arttıracaktır.

Anahtar kelimeler: Gerçek zamanlı izleme, Üç Fazlı Asenkron Motor, Nesnelerin İnterneti (Nİ)

2019, 98 Sayfa
Bilim kodu: 91

ABSTRACT

Ph.D. Thesis

DESIGN OF A REAL-TIME MONITORING AND CONTROLLING SYSTEM FOR THREE PHASE INDUCTION MOTOR USING INTERNET OF THINGS (IOT)

Ali Husein BENHUSEIN

Kastamonu University

Graduate School of Natural and Applied Sciences

Department of Materials Science and Engineering

Supervisor: Prof. Dr. M. Fatih KILIÇASLAN

Abstract: The normal internet called internet of people which connects millions of people, computers and mobile phones allowing them to exchange data around the world. At recent years, a new technology came out which is the Internet of Things (IOT). IOT consists of physical objects that are connected to each other using internet allowing them to send, receive and exchange data between them to perform some task. One of the important elements in most of industrial applications is the use of three phase induction motors. Analysis, monitoring and controlling of parameters of motors is much essential to find out utilization index of the motor for better performance, and it can help avoid any faults early. Although there are many solutions that are used for real time monitoring and controlling the induction motors in manufacturing process, most of them monitor in local or limited in small area. In proposed method, using IOT and web applications for on-line monitoring and controlling solutions from anywhere around the world have become possible with the developing technology. This utilization of IOT and Web applications will increase the efficiency of the monitoring and controlling process.

Keywords: Real-Time Monitoring, Three Phase Induction Motor, Internet of Things (IoT)

2019, 98 Pages

Science Code: 91

TEŞEKKÜR

Öncelikle, Prof. Dr. M. Fatih Kiliçaslan'a bu araştırma süresince sağladığı rehberliği ve yol göstericiliği için teşekkür ederim. Malzeme Bilimi ve Mühendisliği bölümü'nün hocalarına ve araştırma görevlilerine de, bu araştırmanın pek çok pratik ihtiyaçları için sağladıkları destekler için minnettarım. Kastamonu Üniversitesi'ndeki doktora arkadaşlarıma ve Kastamonu'daki Libyalılar'a destekleri için teşekkür etmek isterim. Son ve son derece önemli olarak; anne-babama ve aileme, her daim bana güven veren ve bu tezin tamamlanması için gerekli çalışmaları tamamlamamı sağlayan manevi destekleri için minnettar olduğumu belirtmek isterim. Bu çalışmanın sonuçlarının ilgili endüstriyel alanlarda çalışanlara yardımcı olmasını ve gelecekte yapılacak çalışmalara katkı sağlamasını ümit ediyorum.

Ali Husein BENHUSEIN
Kastamonu, 2019

İÇİNDEKİLER

	Sayfa
TEZ ONAYI	ii
TAAHHÜTNAME	iii
ÖZET.....	iv
ABSTRACT	v
TEŞEKKÜR	vi
ŞEKİLLERLER DİZİNİ	x
TABLOLARLAR DİZİNİ.....	xi
SİMGELER VE KISALTMALAR DİZİNİ	xii
1. GİRİŞ	1
1.1. Nesnelerin İnterneti.....	1
1.2. Uygulama.....	3
1.3. Tüketici Uygulamaları	3
1.4. Veri Toplamak İçin Uygun Nİ Ortamı	4
2. ASENKRON MOTOR VE LİTERATÜR TARAMASI	7
2.1. Arkaplan.....	7
2.2. Düğüm I Kısmı.....	8
2.3. Düğüm II Kısmı.....	9
2.4. Ana Düğüm Kısmı	9
2.5. Ana Bilgisayar/PC Kısmı.....	10
2.6. Asenkron Motor İzleme Sistemleri Hakkında Literatür Taraması.....	11
2.7. Nesnelerin İnterneti (Nİ) Hakkında Literatür Taraması.....	12
3. MQTT PROTOKOLÜ VE BULUT HESAPLAMA SERVİSLERİ.....	14
3.1. MQTT Protokolü.....	14
3.2. Bulut Hesaplama Servisleri	17
3.2.1. Bulut hesaplama ve utility computing (servislerden faydalanma).....	18
3.2.1.1. Benzerlikler	18
3.2.1.2. Farklılıklar	18
3.2.2. Bulut hesaplama ve dağıtımli hesaplama	18
3.2.2.1. Benzerlikler	19
3.2.2.2. Farklılıklar	19
4. UYGULAMA VE DENEY SONUÇLARI	21
4.1. Donanım Kısmı	21
4.1.1. Ana düğüm.....	22

4.1.1.1. XBee Koordinatörü.....	22
4.1.1.2. Arduino Leonardo kartı.....	23
4.1.1.3. Kişisel Bilgisayar (PC).....	24
4.1.2. Düğüm I.....	24
4.1.2.1. SCT013-000 Split-Core Akım Transformatörü.....	25
4.1.2.2. Arduino Leonardo Kartıyla Akım Sensörüne Arayüz Oluşturma	26
4.1.2.3. ZMPT101B AC tek fazlı voltaj sensör modülü.....	28
4.1.2.3.1. Kontaktör.....	29
4.1.2.3.2. Röle	30
4.1.2.3.3. Arduino Leonardo kartı	30
4.1.2.3.4. XBee uç cihazı.....	31
4.1.2.3.5. 20x4 LCD	31
4.1.3. Düğüm II.....	31
4.1.3.1. Infrared engellerden kaçınma sensörü.....	32
4.1.3.2. LM35 Isı sensörü	34
4.1.3.3. Arduino Leonardo kartı.....	34
4.1.3.4. XBee Uç Cihazı.....	35
4.1.3.5. 16x2 LCD	35
4.1.3.6. XBee modüllerinin yapılandırılması.....	36
4.1.3.7. Ağdaki XBee cihazlarının yapılandırılması için izlenen adımlar	36
4.1.3.8. Parametrelerin Tartışılması.....	40
4.2. Yazılım Kısmı	41
4.2.1. Düşük Seviye Programlama	41
4.2.1.1. Düğüm I kodu.....	41
4.2.1.2. Düğüm II kodu	43
4.2.1.3. Ana Düğüm Kodu	45
4.2.2. Yüksek Seviye Programlama	46
4.3. Bulut hesaplama servisleri.....	49
4.3.1. Software as a Service (SaaS-Hizmet olarak yazılım)	49
4.3.2. Infrastructure as a Service (IaaS-Hizmet olarak altyapı).....	49
4.3.3. Platform as a Service (PaaS-Hizmet olarak platform).....	49
4.3.4. Nİ sistemimizde kullanılan bulut servisi	49

4.4. Üç fazlı asenkron motorun izlenmesi kanalı (Monitoring three phase induction motor channel)	51
4.5. MQTT protokolü (Message Queuing Telemetry Transport- Mesaj Kuyruk Telemetri İletimi).....	54
5. SONUÇ VE GELECEK ÇALIŞMALAR	58
5.1. Sonuç.....	58
5.2. Gelecek Çalışmalar	59
KAYNAKLAR.....	60
EKLER.....	64
EK 1 Sistem Blok Şeması.....	65
EK 2 Düğüm I Düşük Seviye Kodu	66
EK 3 Düğüm II Düşük Seviye Kodu.....	76
EK 5 Ana PC İçin Yüksek Seviye Visual Studio Kodu.....	87
EK 6 İş İstasyonu İçin Yüksek Seviye Visual Studio Kodu.....	95
ÖZGEÇMİŞ	98

ŞEKİLLERLER DİZİNİ

	Sayfa
Şekil 1. 1. Tipik Bir Nİ ortamı örneği.....	1
Şekil 3. 1. Publish-subscribe'a göre MQTT bağlantısı.....	15
Şekil 4. 1. Tasarlanan sistemin tamamı	21
Şekil 4. 2. Ana düğüm için tasarlanan sistem.....	22
Şekil 4. 3. XBee modülü	23
Şekil 4. 4. Arduino Leonardo kartı.....	23
Şekil 4. 5. Düğüm I için tasarlanan sistem	25
Şekil 4. 6. SCT013-00 Akım sensörü	25
Şekil 4. 7. Arduino için akım sensörü uyumlama	27
Şekil 4. 8. ZMPT101B AC tek fazlı voltaj sensor modülü.....	28
Şekil 4. 9. 3 fazlı kontaktör.....	29
Şekil 4. 10. Röle modülü.....	30
Şekil 4. 11. Düğüm I'in yerel olarak izlenmesi.....	31
Şekil 4. 12. Düğüm II için tasarlanan sistem	32
Şekil 4. 13. Infrared engellerden kaçınma sensörü	32
Şekil 4. 14. Infrared engellerden kaçınma sensörünün kurulumu	33
Şekil 4. 15. LM35 Isı sensörünün kurulumu	34
Şekil 4. 16. Düğüm II'nin yerel izlenişi.....	35
Şekil 4. 17. X-CTU programına ait kısayol.....	36
Şekil 4. 18. XBee explorer kartı	37
Şekil 4. 19. XCTU yazılımı	37
Şekil 4. 20. Seri portun seçilmesi ve işaretlenmesi	38
Şekil 4. 21. Bulunan XBee cihazının eklenmesi	38
Şekil 4. 22. XBee'nin yazılım üzerinde çalışması.....	39
Şekil 4. 23. XBee parametreleri	39
Şekil 4. 24. Faz 1 voltaj sensöründen gelen veriyi okumak için yazılan örnek kod ...	43
Şekil 4. 25. Motor RPM'sini hesaplamak için örnek kod.....	45
Şekil 4. 26. Ana Arduino Leonardo kartından PC'ye çıktı olarak yollanan veri	46
Şekil 4. 27. Ana PC için tasarlanan grafik kullanıcı arayüzü (GUI).....	47
Şekil 4. 28. Uzak PC için tasarlanan grafik kullanıcı arayüzü	48
Şekil 4. 29. Microsoft Access'le tasarlanan veritabanı dosyası	48
Şekil 4. 30. Thingspeak.com'da oluşturulan kanallar.	51
Şekil 4. 31. Thingspeak.com'da oluşturulan kanalda sunulan veriler.....	53
Şekil 4. 32. Thingspeak.com'da motor durumu kanalında sunulan veriler	54
Şekil 4. 33. Android işletim sistemi kullanan cihazlar için tasarlanan grafik kullanıcı arayüzü.....	55
Şekil 4. 34. MQTT broker'ının yapılandırılması	56

TABLolarLAR DİZİNİ

	Sayfa
Tablo 4. 1. Ağdaki düğümleri yapılandırmak için kullanılan parametreler	40



SİMGELER VE KISALTMALAR DİZİNİ

Kisaltmalar

AC	Alternatif Akım
ADC	Analog Dijital Çevirici
API	Uygulama Programlama Arayüzü
COM	İletişim Kapısı
CT	Akım Transformatörü
GUI	Grafik Kullanıcı Arayüzü
HTTP	Hyper-Text Transfer Protokolü
IANA	İnternet Tahsisli Sayılar Kurumu
IDE	Tümleşik Geliştirme Ortamı
İNİ	Nesnelerin İnterneti
IP	İnternet Protokolü
IR	Infra-Red
LCD	Sıvı Kristal Görüntü Birimi
MQTT	Message Queuing Telemetry Transport- Mesaj Kuyruk Telemetri İletimi
M2M	Makineden Makineye
OASIS	Yapılandırılmış Bilgi Standartlarını Geliştirme Kuruluşu
PAN ID	Kişisel Ağ Kimlikleyici
PC	Kişisel Bilgisayar
PLC	Programlanabilir Lojik Kontrolör
QOS	Hizmet Kalitesi
RF	Radyo Frekansı
RFID	Radyo Frekansıyla Tanımlama
RMS	Ortalama Karekök
RPM	Dakikadaki Devir Sayısı
RX	Alış
SCL	Seri Saat
SDA	Seri Veri
TCP	İletim Denetim Protokolü
TLS	Taşıma Katmanı Güvenliği
TX	İletim
USB	Evrensel Seri Veriyolu

1. GİRİŞ

1.1. Nesnelerin İnterneti

Nesnelerin İnterneti [1-5], en basit şekilde, günlük hayatta kullanılan, internete bağlanıp veri alışverişi yapabilen nesnelere olarak tanımlanabilir. Bu ortamlarda kullanılan cihazlar düşük maliyetli ve kısıtlı cihazlardır. Bu cihazların doğrudan internete bağlanacak olması ek maliyet getirecektir. Genellikle benimsenen yaklaşım, sensörlerden toplanan verilerin düğümlerde bir araya getirilmesi ve bu verilerin bir geçit cihazına iletilmesidir. Bu geçit cihazı doğrudan İnternet ağına bağlıdır. Dolayısıyla bu cihaz İnternet ve Nİ cihazları arasında bir köprü vazifesi görmektedir. İnternete sunulan veriler incelenebilir, işlenebilir ve saklanabilir. Benzer bir Nİ yapısı Şekil 1.1’de sunulmaktadır [6].



Şekil 1. 1. Tipik Bir Nİ ortamı örneği

Nesnelerin interneti, günlük nesnelerin internetle arasında dijital bir bağlantı olmasını ifade eden bir kavramdır. Diğer bir deyişle, insanlardan ziyade nesnelerin internet bağlantısıdır. Sıklıkla herşeyin interneti veya nesnelere internet olarak da

anılır. Eğer günlük nesnelerin dahili radyo vericileri olsaydı, bu nesneler başka birimler tarafından, tıpkı insanların bu nesneleri tanımladığı gibi tanımlanabilirlerdi. Nesnelerin interneti kavramı 1999 yılında Kevin Ashton tarafından, radyo frekansı ile tanımlama (RFID) ve sensör teknolojileri sahasında araştırmaların yapıldığı MIT'ye ait Auto-ID Center'da geliştirilmiştir [7-11].

Örneğin kitapların, termostatların, buzdolaplarının, kolilerin, lambaların, ilk yardım çantalarının, otomotiv parçalarının ve daha pek çok şeyin herbiri internet bağlantısı sağlayan ve ayrıca kendilerini tanımlayıcı cihazlarla donatılmış olsaydı, teorik olarak, herhangi bir ürünün tükenmiş olması veya son kullanım tarihi geçmiş ilaçların mevcudiyeti söz konusu dahi olmazdı; her bir ürünün veya nesnenin dünya üzerindeki konumunu, nasıl tüketildiğini tam olarak biliyor olurduk; herhangi bir şeyin kayıp olması geçmişte kalmış olurdu ve her hangi bir zamanda neyin açık neyin kapalı olduğunu biliyor olurduk [11].

Nesnelerin interneti teorik olarak, 50 ila 100 milyar civarında nesneyi kodlamak ve hareketlerini takip etmek durumundadır. Her insanın en azından, toplamda 1000 ila 5000 nesne tarafından kuşatılmış olduğu tahmin edilmektedir. Son araştırmalar, bir yıl içinde 30 milyar cihazın kablosuz olarak internete bağlı olacağını göstermektedir. İnternet uygulamalarının bir sonraki nesliyle (IPv6 protokolü), daha önce IPv4 ile mümkün olmayan, tüm nesnelerin tanımlanması mümkün olabilecektir. Bu sistemde bir kod aracılığıyla herhangi bir nesne anında tanımlayabilecektir [12-14].

Nesnelerin interneti girişiminin öncülerinden olan ABD şirketi Cisco, Temmuz 2013'ten 2020'ye kadar internete bağlanan nesnelerin yaklaşık sayısını hesaplamaya imkan veren bir "karşı bağlantılar" dinamiği oluşturmuştur. Nesnelerin internetiyle ilgili yapılan çalışmalarda, düşük güçlü radyo sinyalleriyle cihazları ağa bağlama en aktif araştırma sahasını oluşturmaktadır. Bu gerçek, bu türden sinyallerin wi-fi veya Bluetooth'a neden ihtiyaç duymadığını açıklamaktadır. Ancak Chirp Ağları olarak adlandırılan, daha az enerji gereksinimli ve daha düşük maliyetli alternatifler üzerine araştırmalar devam etmektedir [15, 16]. Kavram olarak nesnelerin interneti, cihazlar arasında gelişmiş bağlantıları, sistemleri ve geleneksel M2M (Makineden Makineye)'nin ötesinde servisleri ve geniş bir yelpazede yer alan protokolleri, etki alanlarını ve uygulamaları kapsamakta ve bunların hepsini temsil etmektedir.

1.2. Uygulama

İnternete baęlı cihazlar üzerindeki uygulamalar çok fazladır. Bu uygulamalar için farklı kategoriler önerilmiştir ancak kullanımına göre başlıca üç kategori üzerinde genel bir mutabakat söz konusudur: tüketicilere dönük uygulamalar, iş amaçlı uygulamalar ve altyapı uygulamaları [17, 18].

Sınırlı bir hafızaya, güce ve CPU'ya sahip gömülü cihazların (*İng.* embedded device) aęa bağlanabilme yeteneęinin olması, Nİ'nin hemen her alanda uygulamalarının söz konusu olabileceęi anlamına gelir [19]. Bu sistemler farklı ortamlarda veri toplamada kullanılabilir: örneęin çevresel gözlemlerde veya şehir planlamada kullanmak amacıyla doğal ekosistemlerden binalara, oradan fabrikalara kadar [20], [21].

Örneęin akıllı alış-veriş sistemleri, belirli bir kullanıcının satın alma alışkanlıklarını cep telefonu aracılığıyla takip edebilirler. Bu kullanıcılara tercih ettikleri ürünlerle alakalı özel tekliflerde bulunulabilir veya satın almaya ihtiyaç duydukları ürünlerin konumuna yönlendirilebilirler. Bu ürünler akıllı buzdolabınız tarafından akıllı telefonunuzdaki listeye otomatik olarak eklenebilir [22, 23]. Isınma, su, elektrik, enerji yönetimi hatta araç sürücüsüne yardım eden akıllı taşıma sistemleri gibi alanlarda daha fazla uygulama sahası bulunabilir [24-26]. Nesnelerin interneti'nin kullanışlı olabileceęi bir dięer uygulama sahası evlere farklı özellikler kazandırma ve daha fazla güvenlik sağlama olabilir [27]. Tanımlanan biyolojik sensör ağlarının bulut hesaplamaya dayalı DNA ve dięer moleküller üzerine analizler yapacağı bir "yaşayan nesnelerin interneti" kavramı da ileri sürülmüştür [28].

1.3. Tüketici Uygulamaları

Tüketicilere yönelik Nİ cihazları her geçen gün artan bir oranda üretilmektedir. Tüketici uygulamalarının bazı örnekleri şunlardır: aęa baęlı arabalar, eğlence, ev otomasyonu, giyilebilir teknolojiler, uzaktan takip için Wi-Fi'yi kullanan aęa baęlı sağlık uygulamaları ve çamaşır makinesi, kurutma makinesi, robotlu elektrik süpürgesi, hava filtresi, fırın, buzdolabı gibi cihazlar [29].

Bazı tüketici uygulamaları gereksizlik ve uyumsuzluk yönüyle fazlaca eleştirilmiştir. Bu eleştiriler “Saçmalıkların İnterneti” (İng. “Internet of Shit”) gibi parodilerin ortaya çıkmasına sebep olmuştur [30]. Bazı şirketler, bir değeri olup olmadığı belirsiz, standart güvenlik gereksinimlerini karşılamayan cihazlar üretmeleri ve buna ek olarak Nİ kavramına aceleci yaklaşımları yönünden ağır eleştirilere maruz kalmaktadır [31], [32].

1.4. Veri Toplamak İçin Uygun Nİ Ortamı

Bireylerden veri toplama, literatürde çok uzun süreden beri çalışılan konulardan biridir. Çalışmaların çoğunda, bireylerin üzerine yerleştirilen sensörler kablo aracılığı ile veri iletimi yapmaktadır. Daha yeni çalışmalarda, veri aktarımı kablosuz ortamlarda gerçekleştirilmektedir. Verinin kablosuz olarak aktarımı için farklı teknolojiler kullanılmıştır. Farklı teknolojiler çözüm gerektiren farklı problemlere vücut vermiştir. Kablosuz ortamda veri aktarımında kullanılacak teknolojilerden biri olan Bluetooth, haberleşme mesafesi bağlamında çok sınırlıdır. Farklı bir yaklaşımla Wi-Fi teknolojisi tercih edildiğinde ise, yüksek enerji gereksinimi ortaya çıkmaktadır. Bu gibi sebeplerle, haberleşme mesafesi büyük ancak düşük enerji gereksinimli teknolojiler üzerine araştırmalar yoğunlaşmıştır. Söz konusu kablosuz seçenekler arasında, en fazla öne çıkan 802.15.4 radyodur. 802.15.4 radyo, alternatiflerine nazaran hem daha düşük enerji gereksinimine sahiptir, hem de haberleşme mesafesi daha uzundur.

Elektronikte ve entegre devrelerde kaydedilen yeni gelişmeler, üç fazlı asenkron motorunda giyilebilir ve takılabilir küçük ve akıllı kontrol cihazlarının gelişmesinde oldukça etkili olmuştur. Bu cihazlar aracılığıyla üç fazlı asenkron motoru üzerindeki kontrol verileri toplanabilir. Bu cihazlar kablosuz iletişim ekipmanı ile donatıldıklarında, ağ ortamında veri toplamayı ve iletişim özelliklerini realize edebilirler. Örneğin, pulse oksimetre ile, üç fazlı asenkron motorundaki motor seviyesinin hızı, tork ölçülebilir. Hız kontrol sensörüyle, bir motorun hız seviyeleri takip edilebilir. Tork kontrol sensörüyle, motorun tork parametreleri takip edilebilir. İzleme sensörüyle, üç fazlı asenkron motorunun pek çok bilgileri takip edilebilir. Bir akselerometre sensörüyle motorun aktiviteleri takip edilebilir. Benzer bir kablosuz

komünikasyon ağı, örneğin, üç fazlı asenkron motorunu takip etmek için bir dispansere yerleştirilebilir. Üç fazlı asenkron motoruna ait veriler izleme merkezine (örneğin bir laboratuvara) internet üzerinden aktarıldığında, motorun hız durumu uzaktan takip edilebilir. Bu sensörlerin bazıları araştırma laboratuvarlarında veya endüstride geliştirilmiştir ve Nesnelerin İnterneti ambiyansı içinde kullanılabilir.

Bu tez çalışmasında Zigbee sensörleriyle donanımlı bireylere ait hayati verilerin, en verimli standart Nesnelerin İnterneti protokol konfigürasyonları ve algoritmalarıyla motor izleme merkezine aktarımı problemi üzerine çalışılmıştır. Yani, bireylere ait hayati veriler MQTT iletişim kuralları yığını ile HTTP üzerinden bir ağ geçidine İnternet yönlendirme protokolü ile, bir Nesnelerin İnterneti ortamına, ağ geçidinden de asenkron motor izleme merkezine aktarılacaktır.

Bu protokoller Nesnelerin İnterneti uygulamalarının geleceğini şekillendirecektir. Uygulamaların başarılı işletiminin ölçümlenebilmesi için performans değerlendirmelerinde enerji, gecikme, güvenilirlik, iş çıkarma yeteneği gibi değerlendirmeler esas alınmaktadır. Kısacası bu çalışma kapsamında, bireye ait hayati verilerin motor izleme merkezine en verimli Nesnelerin İnterneti ağ protokol konfigürasyonları ve algoritmaları ile transfer edilmesi üzerine çalışılmıştır.

Hızla gelişmekte olan Nesnelerin İnterneti kavramı, Daha Akıllı Bir Gezegen vizyonunun realize edilmesinde kritik önemdeki temellerden biridir. Buna ek olarak, bütün cihazları nerede olurlarsa olsunlar, birbirine, internete ve ilgili ticari teşebbüslere bağlamayı mümkün hale getirecek olan telemetriye (uzaktan ölçüm) yeni ve daha gelişmiş yaklaşımlar, aynı zamanda Nesnelerin İnterneti kavramına da destek sağlayacaktır. Bu gelişmelerden biri MQTT mesajlaşma protokolüdür. Bununla ilgili sensörler o kadar hafifi bir ağırlığa sahiptir ki, en küçük ölçümleme ve izleme cihazları tarafından desteklenebilirler ve geniş bir alana yayılmış, bazen gayri nizami olabilen ağlarda dahi veri transferini gerçekleştirebilirler. Bu aynı zamanda açık kaynak bir koda sahiptir, ve böylece farklı mesajlaşma ve komünikasyon/iletişim ihtiyaçlarına göre uyarlanabilmektedir. MQTT'nin detaylarına girmeden önce, geliştiricilerin MQTT'yi kullanarak birbirine bağlamaya çalıştığı, gelişmekte olan dünyaya kısaca göz atmak akıllıca olacaktır. Herhangi bir internet tarayıcısına erişimi olan, sosyal medya veya arama motoru kullanan herkes,

internetin insanları bilgiye ulařtırmadaki veya başka insanlarla iletiřime geirmedi-
gücünün farkındadır. Bununla birlikte, farklı akıllı cihazların daha da ön plana
ıkmasıyla birlikte, internet, Nesnelerin İnterneti denen şeyi de kapsayacak şekilde
evrimleřecektir. Bunun kaçınılmaz neticesi ise günlük yařamın içinde yer alan tüm
veri paralarını ölçen, hareket ettiren ve bunlara göre bazen bağımsız bir şekilde
konum alan milyarlarca birbirine bağı akıllı cihaz olacaktır.



2. ASENKRON MOTOR VE LİTERATÜR TARAMASI

2.1. Arkaplan

Asenkron Motorları istikrarlı olmaları, dayanıklılıkları ve hız kontrolünde sağladıkları esneklik nedeniyle endüstride tercih edilmektedir. Bu bağlamda, asenkron motorlarının korunması konusu aktif bir araştıma sahasıdır ve pek çok araştırmacıyı cezbetmektedir. Bu tez çalışmasıyla, üç fazlı asenkron motorun korunması ve izlenmesi alanında yapılan çalışmalara katkı sunulması amaçlanmaktadır.

Asenkron motorlarda hata tespiti ve bu motorların korunması amacıyla kullanılan pek çok teknik geliştirilmiştir. Hata tespiti amacıyla kullanılan bazı teknikler; mikrodnetleyici tabanlı koruma sistemleri, Yapay Sinir Ağları, Programlanabilir Mantıksal Denetleyici (PLC) tabanlı koruma sistemleri ve stator hata kontrol stratejileridir [33].

Bu çalışmamızda kullanacağımız teknik mikrodnetleyici tabanlı kontrol sistemidir, izleme amacıyla ise nesnelere interneti (Nİ) kullanılacaktır. Devrenin motor üzerinde tam kontrolü söz konusu olacaktır ve böylelikle devre yüksek/düşük voltaj gibi bazı hatalardan korunacaktır, ayrıca devre motorun sadece güvenli sayılan şartlarda çalışmasına izin verecektir.

Buna ek olarak, önerilen sistem asenkron motoru bir diğer başlıca hata olan tek fazlı olma durumundan da koruyacaktır. Üç fazlı voltaj ve akımın sürekli olarak izlenmesi amacıyla motor, mikrodnetleyici tarafından tam olarak kontrol edilmektedir ve voltaj veya akım anormal bir değere sahip olduğunda, bu değerler normale dönüşüne kadar mikrodnetleyici motoru kapatmaktadır. Motorun dönüş sayısını ve ısısını denetleyen yakınlık algılayıcı ve ısı sensörleri aracılığıyla, bu her iki ölçümden herhangi biri belirli seviyeleri aştığında, üst seviyeli kodlara göre veya sistem yöneticisinin manuel olarak motorun otomatik olarak korunması amacıyla doğru kararı vermesini sağlamak için, mikrodnetleyici ana düğüme ZigBee ağı üzerinden kablosuz olarak sinyal gönderir.

Günümüzde Nesnelere İnternetinin (Nİ) gösterdiği gelişim neticesinde, makinelerin izlenmesi ve kontrolünün sağlanması çok daha kolay hale gelmiştir ve söz konusu

işlemler dünyanın herhangi bir noktasından yapılabilir. Önerdiğimiz metot, üç fazlı asenkron motorun izlenmesi ve kontrolünde ZigBee ve Nİ kullanımına odaklanmaktadır.

Sistemimiz dört kısma ayrılmaktadır (Düğüm I, Düğüm II, Ana Düğüm ve Ana Bilgisayar veya Ana PC), her bir kısmın yapması gereken bir grup görev bulunmaktadır.

Geliştirdiğimiz sistemin Ana PC dışında kalan kısımları yerel ana birim olarak, ZigBee kablosuz iletişim sistemi tabanlı Xbee modülü ve AVR destekli RISC mimarisine bağlı ATmega32u4 düşük-güçlü CMOS 8-bit mikrodenetleyicisi tabanlı Arduino Leonardo kart kullanmaktadır.

Üç fazlı asenkron motorun Arduino Leonardo kartıyla korunuyor olması, gerektiği durumlarda motorun kapatılması ve ayrıca motorun üç fazlı voltajını, akımını, ısısını, devir sayısını (RPM) her zaman için izleme imkanı vermektedir. Dahası, motorun her türlü faaliyeti simültane olarak hem iki adet likit kristal ekrandan (LCD) gösterilir, hem de ölçülen değerler "thingspeak.com" adresinden kullandığımız bulut hesaplama sistemine yüklenir.

2.2. Düğüm I Kısmı

Bu kısım şu görevleri yerine getirmesi amacıyla tasarlanmıştır: Algılama, Biçim Verme, Yerel Sunum, Gönderme, Bekleme, Alma ve Yerine Getirme.

- **Algılama:** Motora uygulanan üç fazlı voltajı ve akımı ölçmek için Bazı sensörler kullanılır (voltaj ve akım sensörleri).
- **Biçim Verme:** Sensörlerden elde edilen verilere kullanışlı ve kontrol edilebilir bir biçim verilir.
- **Yerel Sunum:** Biçim verilmiş olan veriler, yerel izleme amacıyla, 20x4 boyutlardaki LCD ekranında gösterilir.
- **Gönderme:** Biçim verilmiş olan veriler XBee modülü üzerinden kablosuz olarak ana düğüme gönderilir.

- **Bekleme:** Ana düğümden XBee modülü üzerinden kablosuz olarak gelen bir komut olup olmadığını kontrol eder.
- **Alma:** Gönderilen komutları alır.
- **Yerine Getirme:** Ana düğümden XBee modülü üzerinden kablosuz olarak alınan komutlara göre, röleler için atanmış olan portları aktif hale getirir veya devre dışı bırakır.

2.3. Düğüm II Kısmı

Bu düğüme ait görevler (algılama, biçim verme, yerel sunum ve gönderme) şeklinde belirlenmiştir.

- **Algılama:** Bu düğümde, motorun devir sayısını ve ısını hesaplamak amacıyla iki adet sensör (yakınlık ve ısı sensörleri) kullanılmaktadır.
- **Biçim Verme:** Sensörlerden elde edilen verilere kullanışlı ve kontrol edilebilir bir biçim verilir.
- **Yerel Sunum:** Motorun devir sayısı (RPM) ve ısıyla ilgili biçim verilmiş olan veriler, yerel izleme amacıyla, 16x2 boyutlardaki LCD ekranında gösterilir.
- **Gönderme:** Biçim verilmiş olan veriler XBee modülü üzerinden kablosuz olarak ana düğüme gönderilir.

2.4. Ana Düğüm Kısmı

Bu düğüm Xbee koordinatörüne bağlıdır, Xbee koordinatörü ana bilgisayardan ve iki Xbee düğümünden kablosuz olarak komut veya veri alır, aynı birimlere komut veya veri gönderir. Ana Düğüm: Bağlantı, Alma, İletim, Okuma ve Gönderme görevleri için tasarlanmıştır.

- **Bağlantı:** Bu düğüm XBee kablosuz ağının tesisi ve işletiminden sorumludur, diğer iki düğümün tesis edilmiş ağla bağlantısını sağlar ve düğümler arası (ana düğüm ve diğer iki düğüm) bağlantıyı garanti eder.
- **Alma:** Kablosuz olarak XBee ağı üzerinden diğer iki düğümünden gelen biçimlendirilmiş verileri alır.

- **İletim:** Alınan verileri, USB COM portundan doğrudan bağlı olan ana bilgisayara iletir.
- **Okuma:** USB COM portundan doğrudan bağlı olan ana bilgisayardan gelen sistem veya kullanıcı komutlarını okur.
- **Gönderme:** USB COM portundan doğrudan bağlı olan ana bilgisayardan gelen sistem veya kullanıcı komutlarını (AÇMA/KAPAMA komutları), motoru çalıştırmaktan ve durdurmaktan sorumlu olan Düğüm I'e gönderir.

2.5. Ana Bilgisayar/PC Kısmı

Ana Bilgisayar, sistemimizin merkezi işlem birimi konumundadır. Visual Studio 2017 Community sürümünü kullanarak bir grafik kullanıcı arayüzü (GUI) tasarladık ve ana düğüme doğrudan bağlı olan Ana Bilgisayara yükledik.

Grafik kullanıcı arayüzü olarak tasarlanan yazılım, ana düğümden elde edilen motor hakkındaki veriler için (Alma, Çıkarma, Kaydetme, Yayınlama, İndirme ve Gönderme) şeklinde bir dizi talimat içermektedir.

- **Alma:** Ana düğümün diğer düğümlerden elde ettiği verileri string (karakter dizisi) formatında USB COM portu üzerinden alır.
- **Çıkarma:** Düğümlerin topladığı string formatında alınan verileri bölümlere ayırır ve bunlardan motor parametrelerini çıkarır/ayıklar.
- **Kaydetme:** Çıkarımı yapılan verileri, motor parametreleri hakkındaki verileri mihafaza etmek amacıyla tasarlanmış olan Microsoft Access veritabanı dosyasına ileride kullanmak amacıyla kaydeder.
- **Yayınlama:** Motor parametrelerini Nİ uzaktan izleme cihazları için sistemimizin bulutu olarak işlev gören Thingspeak bulut adresinde bir kanaldan yayınlam.
- **İndirme:** Nİ uzaktan izleme ve kontrol cihazlarının buluttan motoru kontrol edebilmesi için hazır olması gereken motor parametrelerini, sistemimizin bulutu olarak atanmış olan Thingspeak bulut adresinde bir kanaldan indirir.
- **Gönderme:** AÇMA/KAPAMA komutlarını motoru başlatmak/durdurmak için gönderir, bu komutlar motora manuel olarak, otomatik olarak veya uzakta bulunan izleme/kontrol cihazlarından gönderilebilir.

2.6. Asenkron Motor İzleme Sistemleri Hakkında Literatür Taraması

Geçtiğimiz yıllar boyunca asenkron motorların verimliliğinin izlenmesi amacıyla önemli çabalar harcandı ve bu amaca yönelik olarak pek çok teknik önerildi. Dolayısıyla bu bölümde literatürde sunulan teknikler kısaca karakterize edilecek ve söz konusu teknikler avantajları ve dezavantajları açısından değerlendirilecektir.

M. P. Bodkhe ve K. N. Pawar [34] Üç Fazlı Asenkron Motor İzleme ve Kontrol Sistemi (Monitoring and Control System for Three Phase Induction Motor) tasarladılar. Çok Fazlı Çok Fonksiyonlu Enerji Ölçümü için IC ADE7758 kullandılar ve ZigBee Protokolünü uyguladılar. ZigBee ağ topolojisi noktadan noktaya bir topolojidir, yani, ZigBee ağı sadece iki düğümle sınırlıdır. Bir ZigBee düğümü bütün motor parametrelerinin okunması içindir (voltaj, akım, devir sayısı-RPM ve stator sargısının ısısı), diğer düğüm ise baz istasyonu düğümüdür ve sistemin izlenmesi ve kontrolü için kullanılır. Sistem sınırlı bir alan dahilinde kullanılmak amacıyla tasarlanmıştır ve "sadece yerel izleme içindir".

Not: GUI'de gösterilen kontrol seçenekleri için herhangi bir kontrol tuşu bulunmamaktadır.

Mehmet Fatih Işık, Mustafa Reşit Haboğlu, ve Büşra Yartaşı [35] akıllı telefona dayalı bir Üç Fazlı Asenkron Motor için Enerji İzleme Sistemi önerisinde bulundular. Önerilen sistem motorun kontrolü için PLC kullanmaktadır. CMT-SVR olarak adlandırılan bilgisayar ethernet bağlantısı üzerinden PLC'ye bağlanmaktadır, CMT-SVR'de toplanan ve işlenen bütün motor parametrelerine ait verilere, IOS/Android işletim sistemine sahip mobil cihazlardan ulaşılabilir. CMT-SVR ve mobil cihazlar arası bağlantı, bu cihazları bir erişim noktasından kablosuz iletişime (Wi-Fi) bağlayarak sağlanmaktadır. Yazarlar Omron'un 3G3RX motor hız kontrolü serisinden A4022EE frekans dönüştürücüsüyle motorun hız kontrolünü yapmaktadır. Geliştirilen sistemde izlenen parametreler voltaj, akım, motor hızı ve motoru çalıştırmak için kullanılan frekanstır. Ayrıca tasarlanan sistem, motora uygulanan gücün frekansını değiştirerek kullanıcının motoru çalıştırmasına/durdurmasına ve motorun hızını kontrol etmesine izin vermektedir.

Yine, sistem sınırlı bir alan dahilinde kullanılmak amacıyla tasarlanmıştır ve "sadece yerel izleme içindir".

Prof. Mahendra P. Bodkhe ve Prof. K. N. Pawar [36], üç fazlı asenkron motora ait parametreleri izlemek için ZigBee protokolünü kullanan bir sistem önerisinde bulundular. Geliştirilen sistem motora ait voltaj, akım, sargı ısısı devir sayısı gibi parametreleri izleyebilmektedir, acil durumlarda motoru açma/kapama imkanı vermektedir. Geliştirilen sistem iki kısımdan oluşur, ilk kısım izleme yazılımının yüklü olduğu bilgisayardır, ikinci kısım ise asenkron motor kontrol devresi olarak adlandırılır ve AT Mega mikrodenetçilerden ve sensörlerden oluşur, bu sensörler motorun hızını ve diğer parametreleri kontrol ederler. Bu iki kısım birbirine ZigBee protokolüne uygun olarak kablosuz iletişimle bağlıdır. Aynı şekilde, bu sistem de sınırlı bir alan dahilinde kullanılmak amacıyla tasarlanmıştır ve "sadece yerel izleme içindir".

P. Geetha ve V. Saravanan [37] da farklı bir teknoloji kullanarak aynı işi yapmışlardır; bu araştırmacılar sistemlerini inşa ederken mikro çip çözümlerini tercih ettiler, veri toplama sistemi olarak da PLC mikrodenetleyici kullandılar.

V.S.D Rekha ve K.Srinivasa Ravi [38] Nİ kullanarak tek fazlı asenkron motoru izlemek ve kontrol etmek için bir sistem geliştirdiler. Geliştirilen sistemde veri toplamak için ve bulut ağ geçidi için Raspberry Pi kullanılmaktadır. Raspberry Pi buluta kablolu ethernet bağlantısı üzerinden bağlanmaktadır. Bu sistemde kullanılan bütün sensörler doğrudan Raspberry Pi'nin girişlerine bağlıdır. Geliştirilen sistemde akım, voltaj, ısı, titreşim, nem ve hız sensörleri kullanılmaktadır.

2.7. Nesnelerin İnterneti (Nİ) Hakkında Literatür Taraması

Nesnelerin İnterneti teriminin İngilizcesi olan "Internet of Things" tabiri, gerçek hayatta sensörleri olan bütün cihazların birbiriyle bağlantılı olduğu bir sistemi tanımlamak amacıyla, ilk olarak teknoloji öncülerinden İngiliz Kevin Ashton tarafından 1999 yılında kullanılmıştır. Aynı yıl, Kevin Ashton, David Brock ve Sanjay Sarma beraber Auto-ID Labs'ı kurdular, bu lab MIT'deki araştırma odaklı

Auto-ID Center'ın devamı gibidir. Aynı kişiler dünya genelinde UPC barkodlarının yerini alması amaçlanan, RFID-tabanlı ürün tanımlama sistemi Elektronik Ürün Kodu'nun (Electronic Product Code) veya EPC'nin geliştirilmesine yardımcı oldular [39].

Cisco Internet Business Solutions Group (IBSG - Cisco İnternet Şirketleri Çözümleri Grubu)'na göre, Nesnelerin İnterneti 2008 ile 2009 arasında, insanlardan daha fazla sayıda “nesne veya eşya” internete bağlandığı anda doğmuştur.

19xx'dan-Günümüze: Bir Dizi Nİ platformu (Pachube, Thingspeak vs.), protokolleri (6LoWPAN, Dash7 vs.) donanım ve yazılımı (Contiki, TinyOS, Arduino vs) geliştirildi.

2011: IPV6 İnternet protokolü genel kullanıma girdi—Bu protokolün kullanımıyla 2^{128} farklı cihaza farklı adres ataması yapmak mümkün hale gelmiştir, veya Steven Leibson'un dediği gibi, “yeryüzündeki her bir atoma bir IPV6 adresi atayabiliriz, ve yine de başka 100 dünyaya yetecek kadar elimizde adres kalmış olur,” [40] bu protokol sayesinde etrafımızdaki bütün cihazların adreslendirilmesi mümkündür.

Günümüzde Nİ cihazları üzerinde yoğun çalışmalar yürütülmektedir ve her geçen gün bu cihazlar üzerine yapılan çalışma sayısı artmaktadır.

3. MQTT PROTOKOLÜ VE BULUT HESAPLAMA SERVİSLERİ

3.1. MQTT Protokolü

Mesaj Kuyruk Telemetri İletimi (*İng.* Message Queuing Telemetry Transport-MQTT), yüksek gecikmeli veya kısıtlı ağlarda cihazlar arası telemetri verilerinin iletimine olanak sağlayan, açık mesajlaşma protokolüne sahip bir makineden makineye (M2M) iletişimdir [41]. İlgili cihazlar sensörlerden aktüatörlere, cep telefonlarına, araçlardaki veya laptoplardaki ve çok gelişmiş bilgisayarlardaki dahili sistemlere kadar çeşitlilik gösterir.

İnternet Tahsisli Sayılar Kurumu (*İng.* The Internet Assigned Numbers Authority-IANA) 1883 ve 8883 portlarını MQTT için ayırmıştır. MQTT mesajları, TLS protokolü kullanılarak şifrelenebilir.

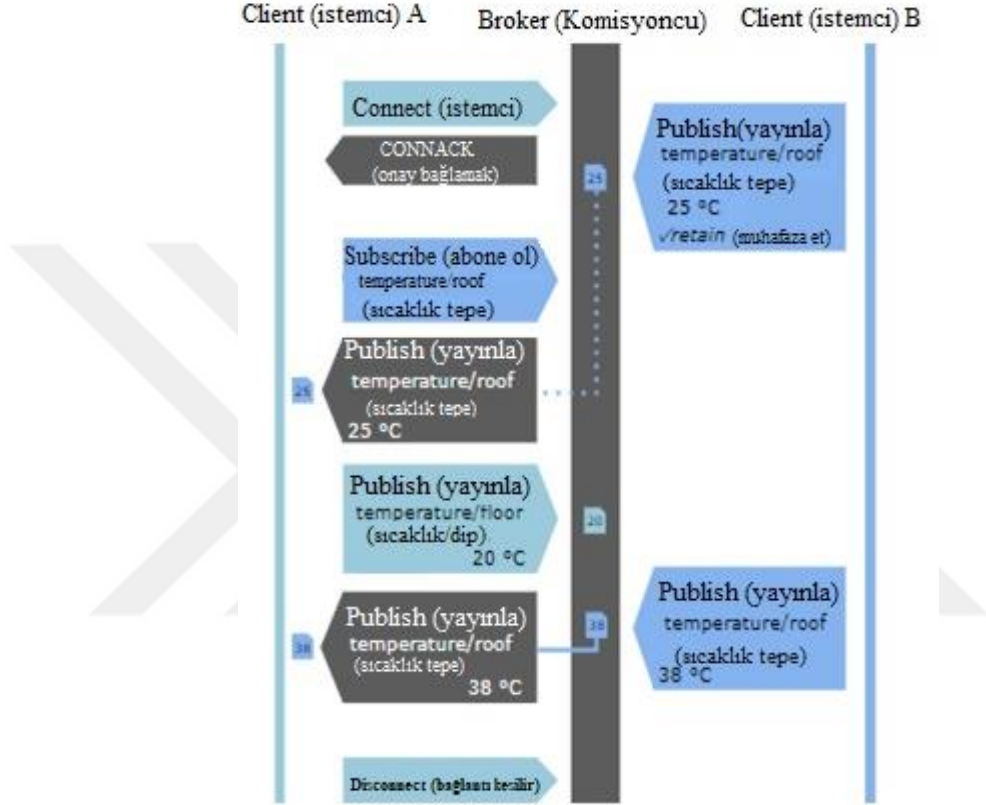
Bir MQTT sunucusunun ("broker") bütün iletişim ortaklarının durumlarıyla ilgili verileri tutması ve bu sayede bir veritabanı olarak kullanılabilmesi ilginçtir. Dolayısıyla küçük, çok fazla işlevsel olmayan MQTT cihazlarının bir MQTT broker'ına bağlanması, böylece cihazların veri toplaması ve/veya komut almaları, bu arada olası karmaşık durumların sadece MQTT broker'ında ortaya çıkması ve bu brokerda veya verimli bir iletişim ortamında bu durumun üstesinden gelmesi mümkündür; bu şekilde, ciddi aksaklıklar MQTT broker'ına iletilebilir ve oradan bireysel cihazlara dağıtılabilir. Bu uygulama MQTT'yi otomasyon çözümleri için çok ideal bir hale getirir ve buna ek olarak kullanım kolaylığı nedeniyle Nİ'de çok yaygın olarak tercih edilmektedir.

Bu protokol IBM'den Andy Stanford-Clark ve Cirrus Link Solutions'tan Arlen Nipper tarafından geliştirilmiştir. 2013'ten beri Yapılandırılmış Bilgi Standartlarını Geliştirme Kuruluşu (Organization for the Advancement of Structured Information Standards-OASIS) aracılığıyla MQTT, Nesnelerin İnterneti ile ilgili standart protokol haline gelmiştir [11, 42].

Aşağıda publish-subscribe (yayınla-abone ol) kalıbına göre bir MQTT bağlantı örneği

görülmektedir. İstemci B'den gelen ilk mesaj, önceden belirlenmiş saklama kriteri nedeniyle muhafaza edilmektedir.

Publish-subscribe'a göre MQTT bağlantısı 3.1'de gösterilmektedir.



Şekil 3. 1. Publish-Subscribe'a Göre MQTT Bağlantısı

MQTT bir istemci-sunucu protokolüdür. İstemciler bağlandıktan sonra, sunucuya ("broker") mesajı hiyerarşik olarak sınıflandıracak konuya sahip bir mesaj gönderirler; istemciler bu konulara abone olabilirler, sunucu da gelen mesajları ilgili abonelere yönlendirir.

Mesajlar her zaman bir konudan ve mesaj içeriğinden oluşurlar. Mesajlar tanımlanabilir bir Servis Kalitesine (Quality of Service-QOS) göre gönderilirler: en fazla bir kez (mesaj sadece bir kez gönderilir ve bağlantının kopması durumunda yerine ulaşamayabilir), en az bir kez (mesaj, alındığı teyit edilene kadar

gönderilmeye devam eder, ve bu teyit gelene kadar tekrarlanabilir) ve tam olarak bir kez (bu şekilde, bağlantıda kesinti olsa bile mesajın tam olarak bir kez iletilmesi garanti edilmiş olur) şeklinde gönderilebilirler. Buna ek olarak, sunucunun ilgili konu hakkındaki mesajları önbelleğe alması için bir saklama kriteri belirlenebilir. Bu konuya abone olan istemciler ilk olarak bu önbelleğe alınmış mesajı görürler [43].

İstemciler, bağlantı kurarken mesaj şeklinde bir "son istek" tanımlayabilirler. Eğer istemciyle bağlantı kaybedilirse, bu mesaj yayınlanır ve ilgili abonelere ulaştırılır.

MQTT genellikle TCP üzerinden kullanılır ve 2-baytlık bir üst bilgiye sahiptir. İlk bayt mesaj türü bilgisini (4 bit), servis kalitesi bilgisini (2 bit), saklama kriteri bilgisini içerir.

Mesaj türleri şu şekildedir:

- CONNECT
- CONNACK
- PUBLISH
- PUBACK
- PUBREC
- Pubrel
- PUBCOMP
- SUBSCRIBE
- SUBACK
- UNSUBSCRIBE
- UNSUBACK
- PINGREQ
- PINGRESP
- DISCONNECT

İkinci bayt ise MQTT paketinin geri kalan kısmının uzunluğu bilgisini muhafaza eder.

Bu kısmı MQTT konusunu içeren değişken kısım takip eder, yani konu. Son olarak ise veriyükü gelir, yani konu altında yayınlanacak olan veri içeriği.

Konular hiyerarşik olarak organize edilir [44]. Örneğin:

```
New Year's speech / 1984 / audio / ogg
New Year's speech / 1984 / audio / mp3
New Year's speech / 1984 / video
New Year's speech / 1984 / text / ascii
New Year's speech / 1984 / text / odt
```

karakteri, klasör içindeki ve hiyerarşik olarak o klasörün altındaki her şeyin getirilmesi için kullanılabilir.

+ karakteri ise bir hiyerarşi seviyesini genel arama karakteri olarak atamak için kullanılabilir.

Örneğin:

New Year's speech / 1984 / #	New Year's speech/1984/ altındaki herşey
New Year's speech / + / text / ascii	New Year's speech içindeki bütün ASCII text (yazı) dosyaları
New Year's speech / + / audio / #	New Year's speech içindeki bütün audio (ses) formatları

3.2. Bulut Hesaplama Servisleri

Bulut hesaplama veya kısaca bulut servisi, internet aracılığıyla kullanım imkanı kazanmış olan bir bilgi teknolojisi altyapısıdır. Bu servislere saklama alanı, güçlü işlemci imkanı ve yazılım uygulamaları dahildir. Teknik olarak ifade edecek olursak, bulut hesaplama bir bilgisayar ağı üzerinden, bazı yazılım veya donanımların o

bilgisayarda yüklenmiş olmasına gerek kalmaksızın bazı bilgi teknolojileri altyapılarına ulaşım imkanı veren yaklaşımları tanımlar. Bu servislerin kullanımı tamamen teknik arayüzler ve protokoller üzerinden gerçekleşir, örneğin bir web tarayıcısı kullanımıyla. Bulut hesaplamasının sunduğu servisler altyapı, platform ve yazılım da dahil olmak üzere bilişim teknolojilerinin tamamını kapsamaktadır.

3.2.1. Bulut Hesaplama ve Utility Computing (Servislerden Faydalanma)

Utility computing'te, donanım ve yazılım kaynakları devasa veri merkezlerinde bulunur ve kullanıcılar faydalandıkları servislere ve saklama alanına göre faturalandırılırlar.

3.2.1.1. Benzerlikler

Bulut hesaplama ve utility computing arasında bir çok benzerlik görülebilir. Utility computing için genellikle bulut-türü bir altyapı gereklidir; ancak bu uygulamanın cazibesi hesaplama servisleri için ortaya koyduğu iş modelindedir. Utility computing kullanıcılara imkanları doğrudan kiralama hizmeti sunar (bunun anlamı kullanıcıların bu imkanları doğrudan kontrol edebileceğidir).

3.2.1.2. Farklılıklar

Buna karşılık, bulut hesaplamada kullandıkları her bir servis için faturalandırılırlar; ancak, servisin kullanımı yazılımı ve altyapısı açısından daha karmaşıktır. Hem bulut sahipleri hem de yöneticiler imkanların kontrolünü elinde bulundurmaktadır.

3.2.2. Bulut Hesaplama ve Dağıtımli Hesaplama

Dağıtımli hesaplama yazılım ve donanım imkanlarını bölen ve bunları pek çok müşteriye kiralayan, tahsis edilmiş kaynakları içeren bir ağ olarak tanımlanabilir; sadece bir sahibi olan buluttan farklı olarak birden fazla organizasyon bu ağa sahip

olabilir. Bu türden bir ađın kullanıcıları dađıtımlı hesaplama yöneticilerinin kontrolünde, belli bir takvim ve plan dahilinde, ellerinde bulundurdıkları yazılım ve donanım imkanlarını diđer kullanıcılarla paylaşmak durumundadır.

3.2.2.1. Benzerlikler

Bulut hesaplamanın ve dađıtımlı hesaplamanın belirli benzer özellikleri bulunmaktadır. Teorik olarak, her iki model de, tek bir kaynak kullanımıyla yapılması çok zor veya imkansız olan tek bir görevi veya bir dizi karmaşık görevi tamamlamak için farklı bilişim/bilgisayar kaynaklarını bir araya getirerek daha gelişmiş bir hesaplama imkanı ortaya çıkaracak şekilde yeteneklerini tek elde toplama düşüncesine dayanır. Dađıtımlı hesaplama, bilişim/hesaplama kaynakları işlem döngüleri, hafıza alanları, ađlar, yazıcılar, tarayıcılar, yazılım lisansları, uzak cihazlar vb gibi pek çok unsuru içerebilir. Hesaplama/bilişim anlamında gereksinimi fazlaca olan durumlar için hızlı ve düşük maliyetli çözümler getirdiđi için, dađıtımlı hesaplama daha ziyade akademik veya bilimsel amaçlar çerçevesinde kullanılır [45].

Dađıtımlı hesaplardan faydalanmak isteyen herhangi bir kullanıcı, dađıtımlı hesaplama olanađı sađlayan kiři veya kuruluşlarla görüşmeden önce, araştırma konusunun kapsamı ve yapılması düşünölen çalışmanın ne kadar hesaplama kaynađına ihtiyaç duyacađı gibi konularda kapsamlı bir tasarı hazırlamalıdır [46]. Dađıtımlı hesaplama geliştirilirken hedeflenen başlıca amaçlardan biri, yakın bölgede yer alan hesap merkezinden herhangi bir sebeple faydalanılamayacađı veya aynı hesap merkezinin altından kalkamayacađı bir işin söz konusu olduđu durumlarda, uzaktaki hesaplama merkezlerine yakın durumda olan ve atıl durumdaki hesaplama gücünden kullanıcıların yararlanmasına olanak vermektir.

3.2.2.2. Farklılıklar

Dađıtımlı hesaplama kaynakları genellikle ücretsiz olarak kullanıma sunulmaktadır; ancak bu uygulamadan faydalanan kiři elindeki hesaplama imkanlarının gerektiđi durumlarda başkalarının kullanılmasına onay vermek zorundadır [47].

Bulut hesaplama servisleri ise aksine, kendi hesaplama/bilişim çözümünü üretmesi mümkün olmayan veya bundan kaçınan (genellikle ekonomik faktörlerden dolayı) organizasyon veya kuruluşlar için, ticari amaçlarla kamuya arz edilmektedir [48].

Teknik bir bakış açısıyla ele alındığında, dağıtımli hesaplamanın fonksiyonu, farklı organizasyonların kaynaklarını bir araya getirerek bir kaynak havuzu oluşturmak ve böylece tek bir hesaplama merkezinin elde edemeyeceği bir hesaplama imkanı sunmaktır [46]. Bu organizasyonlar coğrafi olarak dağınık durumdadır ve kendilerine sunulacak kaynakları kimin tedarik edeceğine karar verme hakkına sahiptirler. Bulut hesaplamanın amacı ise, kaynakları küçük parçalara ayırmak ve bunları taleplerine uygun olarak istemci veya müşterilerine arz etmektir.

4. UYGULAMA VE DENEY SONUÇLARI

Uygulama için, bir mikrodeneççi tarafından kontrol edilen herhangi bir sistemde olduğu gibi, donanımın amaçlanan görevleri yerine getirmesi amacıyla, donanım bileşenleriyle yazılım kodlarından oluşan bir sistem inşa edilmelidir. Bu doğrultuda, sistemimiz iki ana parçaya bölünecektir (Yazılım ve Donanım).

4.1. Donanım Kısmı

Şekil 4.1’te bir bütün halinde tasarlanan sistem görülmektedir.



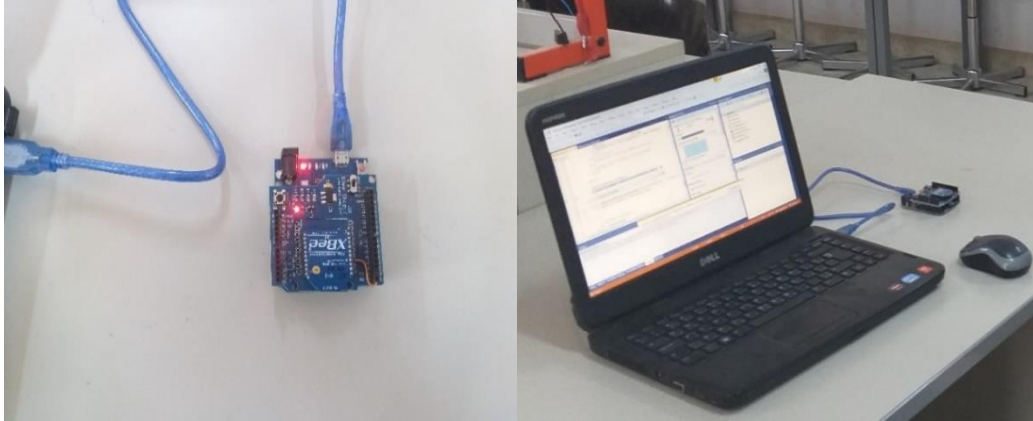
Şekil 4. 1. Tasarlanan Sistemin Tamamı

“Bu deney, Trablus/Libya’da bulunan Trablus Elektronik Teknoloji Üniversitesi’nin Güç Elektroniği laboratuvarında gerçekleştirilmiştir.”

Önerilen modelin şeması EK-A'da gösterilmiştir. EK-A'da ve Şekil 4.1'de görüldüğü gibi, sistemimiz üç parçadan oluşmaktadır: Ana Düğüm, Düğüm I ve Düğüm II.

4.1.1. Ana Düğüm

Bu düğüm diğer düğümlerden gelen verileri toplamakla, sistemin bulutla bağlantısını sağlamakla ve motorun işleyişinde ortaya çıkan hatalı durumlarda anında doğru kararlar almakla yükümlüdür. Şekil 4.2'de ana düğüm için tasarlanan sistem görülmektedir.



Şekil 4. 2. Ana Düğüm İçin Tasarlanan Sistem

Ana düğüm şu birimlerden oluşmaktadır: Xbee Koordinatörü, Arduino Leonardo kartı ve kişisel bilgisayar (PC)

4.1.1.1. Xbee Koordinatörü

Xbee koordinatörü ZigBee ağını başlatmak ve diğer Xbee düğümlerinin birbirleriyle ve Xbee koordinatörüyle veri alış-verişinde bulunmak üzere bu ağa bağlanması için kullanılır. Şekil 4.3'te bir Xbee modülü örneği verilmektedir.



Şekil 4. 3. XBee Modülü

Diğer düğümlerden kablosuz olarak gelen veriler PC'ye seri olarak bağlı olan Arduino Leonardo kartına gelir (tersi de benzer şekilde ilerler) ve koordinatör bu verileri işler.

XBee modülü X-CTU kullanılarak uç cihaz veya koordinatör olarak çalışmak üzere yapılandırılabilir. Bu konu ilerleyen sayfalarda ele alınmıştır.

4.1.1.2. Arduino Leonardo Kartı

Arduino Leonardo ATmega32u4 tabanlı bir mikrodenetleyici karttır. Dijital girdi/çıkışı için 20 pimi bulunur (bu pimlerin 7'si PWM çıkışı olarak ve 12'si analog girdi olarak kullanılabilir), 16 MHz'lik bir kristal osilatörü, bir micro USB bağlantısı, bir power jack'i, bir ICSP başlığı ve reset tuşu bulunur. Mikrodenetleyiciyi desteklemek için gerekli herşey bu kartta bulunmaktadır [49]. Bir Arduino Leonardo kart örneği Şekil 4.4'te gösterilmektedir.



Şekil 4. 4. Arduino Leonardo Kartı

Bu kart sistemdeki ana işlem birimi olarak kullanılmaktadır, XBee koordinatöründen gelen veriyi alacak, bu veriden gerekli çıkarımı yapacak ve kullanışlı bir formata sokacak şekilde uygun kodla programlanır.

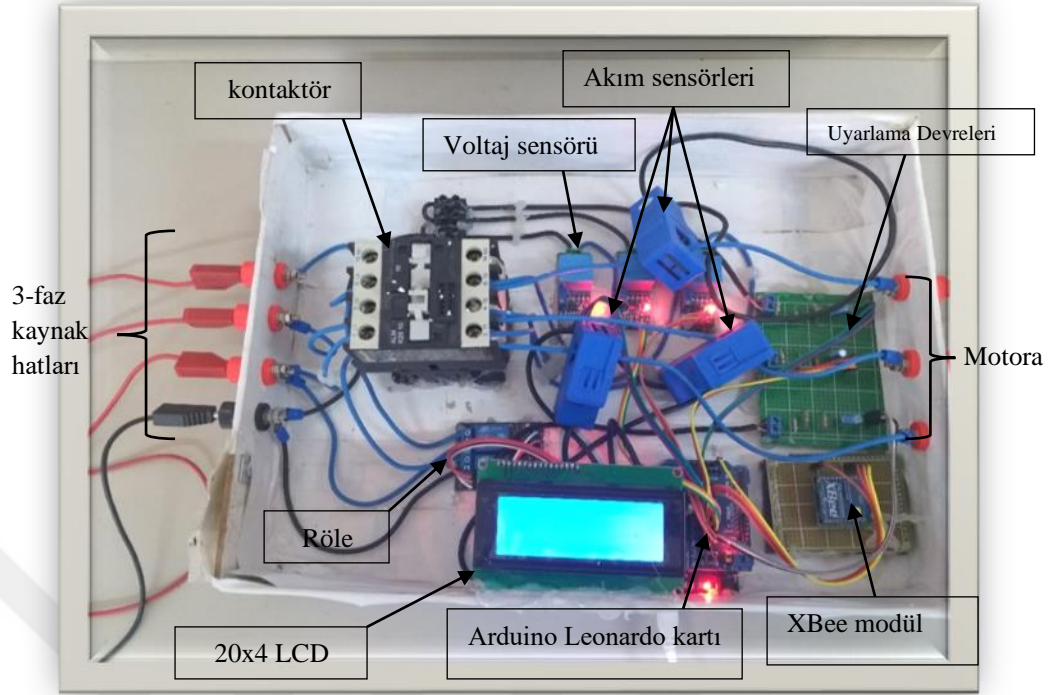
Ayrıca, bu karttaki kodun kullanıcıdan XBee koordinatörüne, oradan da hedeflenen düğüme ulaşması gereken gelen komutları alabilmesi için kullanılması gerekir.

4.1.1.3. Kişisel Bilgisayar (PC)

Kişisel bilgisayar veya kısaca PC gerçek zamanlı olarak sistemi izlemek ve kontrol etmek için, çıkarılan verileri veritabanı dosyasına kaydetmek için, bütün bu dosyaların internet erişiminde olması amacıyla sistemi buluta bağlamakta ağ geçidi olarak kullanılmaktadır.

4.1.2. Düğüm I

Bu düğüm üç fazlı voltajı, motora uygulanan akımı izlemekten ve ana düğümden kablosuz olarak iletilen komutlara bağlı olarak motoru AÇMA ve KAPAMA'dan sorumludur. Şekil 4.5'te Düğüm I için tasarlanan sistem görülmektedir.



Şekil 4. 5. Düğüm I İçin Tasarlanan System

Bu düğüm aşağıdaki parçalardan oluşmaktadır:

4.1.2.1. SCT013-000 Split-Core Akım Transformatörü

Bu transformatör AC motorlarında, ışıklandırma ekipmanlarında, hava kompresörlerinde... vs akımı ölçmek, akımı izlemek ve bu cihazları akım dalgalanmalarına karşı korumak amacıyla kullanılmaktadır. Şekil 4.6'da bir SCT013-00 akım sensörü görülmektedir.



Şekil 4. 6. SCT013-00 Akım Sensörü

Bu sensör devre kesiciye gelen gücü kesmeksizin ana şebekeye bağlanabilir. Akım transformatörünün çıktısı, sensörün halkası içinde yer alan, voltaj çıktı tipi dahili örnekleme direnci sonucu olan AC akımıyla orantılı bir AC voltajdır.

Bu transformatör 0'dan 100A'a kadar AC akımını ölçebilir, AC voltaj çıktısı 0 olduğunda "0A" ve 50mv AC olduğunda ise "100A" değerini verecektir.

Üç adet akım sensörü kullanılmıştır; devre kesiciye giden her bir "sıcak" kabloya bir adet sensör takılmıştır.

4.1.2.2. Arduino Leonardo Kartıyla Akım Sensörüne Arayüz Oluşturma

Arduino'ya bir akım sensörü bağlamak için, Arduino analog girdiler için gerekli olan şartı—yani 0V ile ADC referans voltaj değeri arasında pozitif bir voltaj—sağlayabilmesi için akım sensöründen gelen çıktı sinyaline uyumlama yapılması gerekir.

Bilindiği gibi akım sensörünün çıktı sinyali bir akım çıktısıdır, dolayısıyla "yük direnci" (burden resistor) denen ve akım sensörünün iki çıktı kablosuna paralel olarak bağlanan bir dirençle voltaj sinyaline dönüştürülmesi gerekir.

Bu direncin değeri şu formülle bulunabilir:

100A'de akım sensörünün maksimum çıktı değeri 50mA RMS değerini verecektir [50].

RMS akım değerinden, tepeden tepeye akım değerini hesaplayabiliriz, bunun için şu formül gereklidir:

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

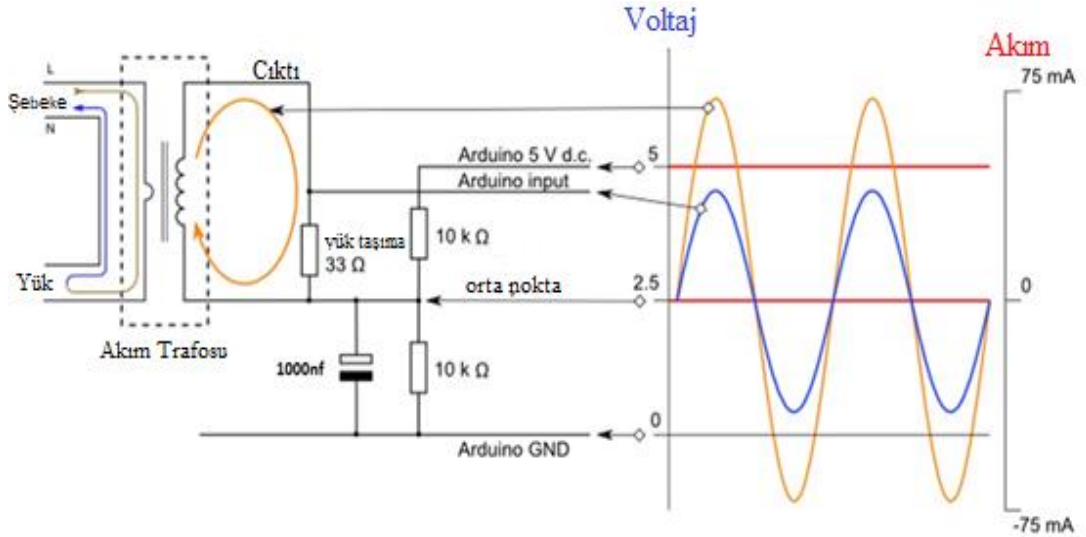
Arduino kartının maksimum voltajı 5V'dur, dolayısıyla yük direncini ohm kanunundan hesaplayabiliriz:

$$R = \frac{V}{I} = \frac{5}{141,42 \cdot 10^{-3}} = 35,35 \Omega \quad (4.2)$$

35 Ω yaygın bir direnç değeri değildir. 35 Ω 'a en yakın değerler 39 ve 33 Ω 'dur. Her zaman daha küçük olan değer seçilir, dolayısıyla 33 $\Omega \pm 1\%$ yük değerini tavsiye edilir.

Bu, iki ana kısımdan oluşan aşağıdaki devreyle elde edilebilir (Şekil 4.7):

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



Şekil 4. 7. Arduino İçin Akım Sensörü Uyumlama

Üç uyumlama devresi için üç akım sensörü kullanılmıştır. Arduino Leonardo analog portlarına bağlı üç sensörün akım okumaları A0'dan A2'ye kadardır.

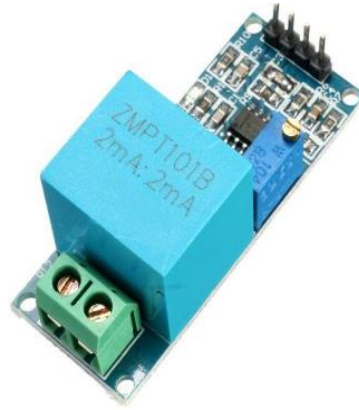
4.1.2.3. ZMPT101B AC tek fazlı voltaj sensör modülü

ZMPT101B Tek Fazlı Voltaj Sensör Modülü Arduino mikrodenetçi kartlarıyla rahatlıkla kullanılmak üzere tasarlanmıştır.

Bu modülün ürün özellikleri şu şekildedir [51]:

1. Dahili mikro hassasiyette voltaj transformatörü.
2. Yüksek hassasiyette dahili amplifikatör devresi, hassas örnekleme ve uygun sinyal kompanzasyon fonksiyonu.
3. 250V AC voltaja kadar ölçüm yapmada kullanılabilir, buna karşılık gelen analog çıktı ayarlanabilir.
4. PCB kart boyutları: 49.5 (mm) x19.4 (mm).

ZMPT101B AC tek fazlı voltaj sensor modülü Şekil 4.8'de görülmektedir.



Şekil 4. 8. ZMPT101B AC Tek Fazlı Voltaj Sensor Modülü

Motora gelen her bir voltaj kaynağını ölçmek için her bir faza bir adet voltaj sensörü kullanılmıştır.

Arduino Leonardo analog portlarına bağılı üç sensörün okuduğı deęerler A3 ve A5 arsında yer almaktadır.

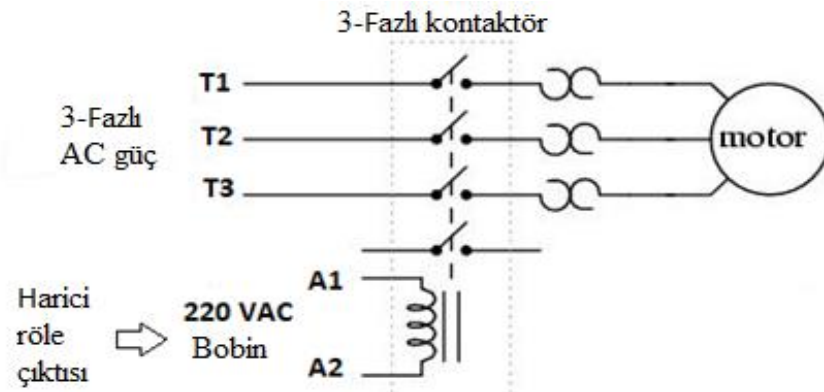
4.1.2.3.1. Kontaktör

Kontaktör, bir elektrik güç devresini açıp kapatabilen ve elektriksel olarak kontrol edilen bir anahtardır. Daha yüksek akıma haiz olması dışında röleye çok benzer. Kontaktör, açıp kapadığı devreden çok daha düşük düzeyde güç seviyesine sahip bir devre tarafından kontrol edilir.

Yüksek akım yükü olan cihazlara doğrudan bağlanır ve onları kontrol eder. Bu cihazlara örnek motorlar, aydınlatma, ısıtma, hava kompresörleri, termal buharlaştırıcılar vb cihazlar olabilir.

Kontaktörlerin tipik olarak birden fazla elektrik temas noktası olabilir, ve bu temas noktaları (her zaman olmasa da) genellikle açıktır, bu sayede bobinin elektriğı kesildiğinde yüke giden güç de kesilir [52].

Tasarımımızda kontaktörün elektriğini açıp kapamak için kontaktör fişleri A1 ve A2'ye bağılı olan harici röle kullandık. 3 fazlı bir kontaktör örneğı Şekil 4.9'da görölmektedir.



Şekil 4. 9. 3 Fazlı Kontaktör

4.1.2.3.2. Rôle

Rôle elektriksel olarak kontrol edilen ve açılıp kapatılabilen bir devre anahtarıdır, böylece devreden akım geçip geçmemesine izin verebilir ve Arduino pimlerinin sağladığı 5V gibi düşük voltajla kontrol edilebilir. Şekil 4.10'da bir rôle modülü örneği görülmektedir.



Şekil 4. 10. Rôle Modülü

Bu rôle modülünün iki kanalı vardır ve Arduino için uygun olan 5V'luk bir voltaja ihtiyaç duyar.

Bu röleyi A1 ve A2 girişlerine 200VAC uygulayarak kontaktörü aktifleştirmek veya devre dışı bırakmak için kullandık, Arduino Leonardo kartında D5 piminden 1 mantık değeri yollandığında rôle aktif hale gelmektedir.

4.1.2.3.3. Arduino Leonardo kartı

Bu kart A0'dan A5'e kadar olan portlardan gelen verileri okumak üzere kod yazılarak programlanmıştır, XBee Düğüm I üzerinden ana düğüme yollamadan önce, gerekli mantık dizilerini oluşturur ve uyumlu bir format için gereken kalibrasyonları yapar.

Kod ayrıca, XBee Düğüm I ve XBee koordinatörü arasında bağlantının sağlandığını garanti edecek prosedürleri içerir ve buralardan gelen veya buralara giden verilerin işlenmesinden sorumludur.

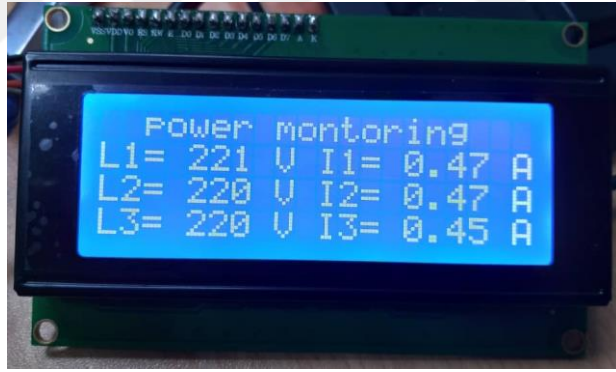
4.1.2.3.4. XBee Uç Cihazı

Bu cihaz Arduino Leonardo kartı ve ana düğüm arasında kablosuz olarak gerçekleşen veri alış-verişini gerçekleştirmekle sorumludur.

4.1.2.3.5. 20x4 LCD

Bu birim sensörler tarafından ölçülen voltajın ve akımın yerel olarak izlenmesi için kullanılmıştır.

Şekil 4.11’de düğüm I’in çıktılarının yerel olarak izlenişi görülmektedir. LCD ekranı motora uygulanan L1, L2 ve L3 voltaj ve akım değerlerini göstermektedir.

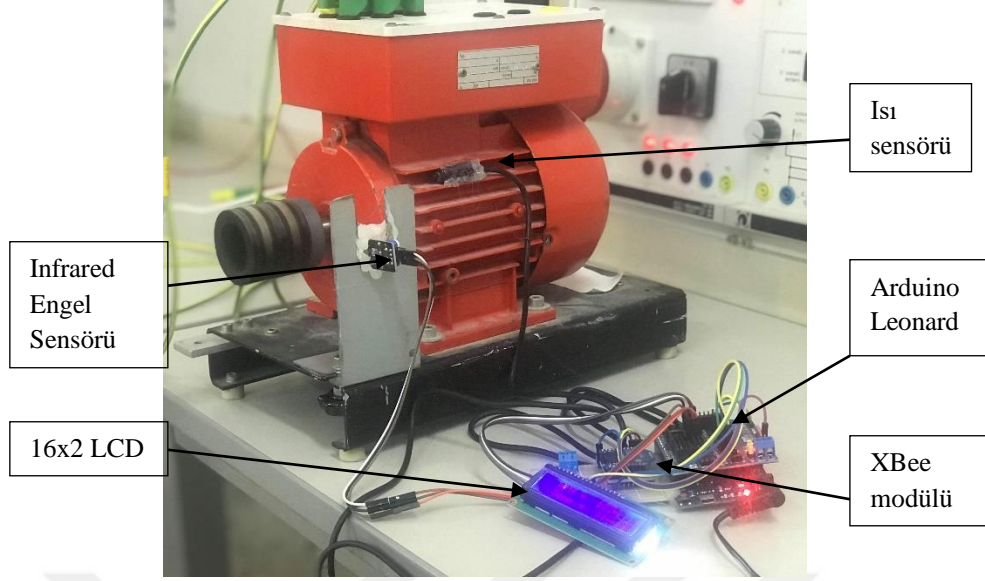


Şekil 4. 11. Düğüm I’in yerel olarak izlenmesi

Bu LCD, Arduino Leonardo kartına pim 2 ve pim 3’ten bağlıdır.

4.1.3. Düğüm II

Bu düğüm sadece motorun ısını ve hızını izlemek için kullanılmaktadır. Şekil 4.12’de Düğüm II için tasarlanan sistem görülmektedir.

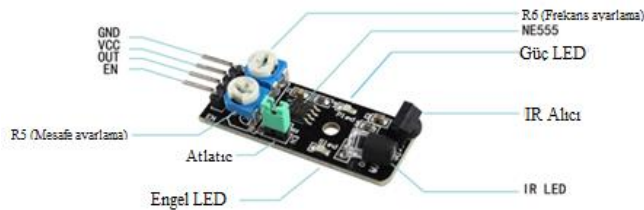


Şekil 4. 12. Düğüm II için tasarlanan sistem

Bu düğüm şu kısımlardan oluşmaktadır:

4.1.3.1. Infrared Engellerden Kaçınma Sensörü

Infrared engellerden kaçınma sensörü'nün dahili bir IR alıcısı ve vericisi vardır, etrafa IR enerjisi yayar ve yansıyan IR enerjilerini inceleyerek sensör modülünün önünde olası bir engelin varlığını algılamaya çalışır. Modülün kart üzerinde, kullanıcının algılama mesafesini belirlemesine imkan veren potansiyometresi vardır. Çok az çevresel ışığın varlığında hatta tamamen karanlık ortamlarda dahi sensörün oldukça iyi ve sağlam bir etkinliği vardır. Infrared engellerden kaçınma sensörü Şekil 4.13'te görülmektedir.



Şekil 4. 13. Infrared Engellerden Kaçınma Sensörü

Şekil 4.13'te görüldüğü gibi, Infrared engellerden kaçınma sensörünün Güç, Topraklama, Sinyal, ve Devreye alma pimleri bulunmaktadır. Ayrıca kart üzerinde 2 adet potansiyometre ve bir adet atlatici bulunmaktadır.

Şekildeki R5 potansiyometresi sensörün hassasiyetini ayarlamak için kullanılmaktadır. Ve sensörün istenilen mesafedeki bir nesneyi algılayacağı şekilde ayarlanabilir.

Bu sensör motorun RPM'sini, yani "Dakikadaki Devir Sayısı" nı hesaplayarak motorun hızını ölçmek için kullanılır.

Şekil 4.14'te devir sayısını algılamak üzere bu sensörün tasarım içine nasıl kurulduğu görülmektedir.



Şekil 4. 14. Infrared Engellerden Kaçınma Sensörünün Kurulumu

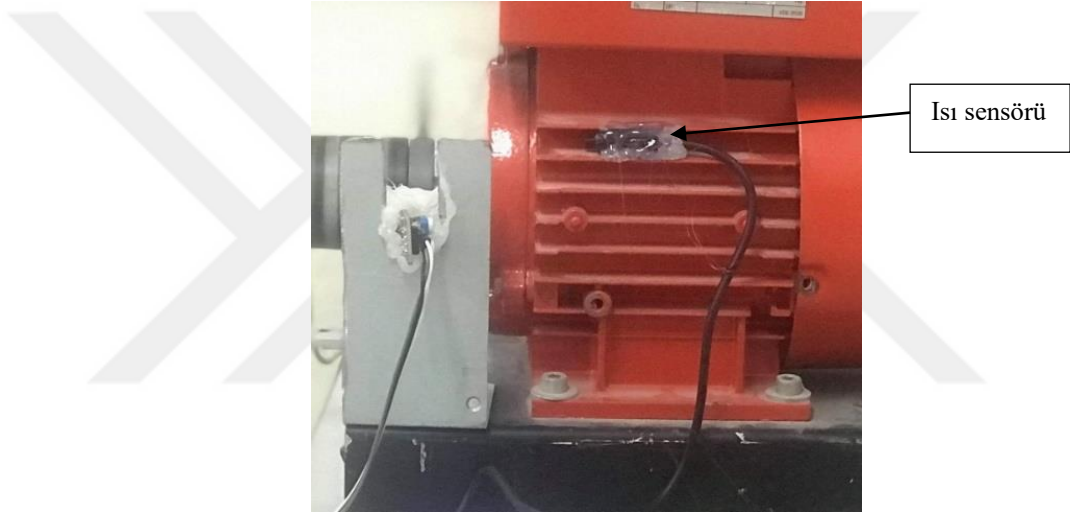
Arduino Leonardo kartı üzerine programlanan kod, sensörün dakikada oluşturduğu sıfır sinyallerinin sayısını kullanarak motorun dakikadaki devir sayısını hesaplamaktadır.

Bu sensörün okuma çıktısı Arduino Leonardo'nun pim 7'sine bağlanmaktadır.

4.1.3.2. LM35 Isı Sensörü

Bu sensör motorun ısını izlemek için kullanılmaktadır, bu sensörün okuma çıktısı Arduino Leonardo kartında A0'a bağlıdır.

Şekil 4.15'te bu sensörün motorun ısını algılamak amacıyla nasıl yerleştirildiği görülmektedir.



Şekil 4. 15. LM35 Isı Sensörünün Kurulumu

4.1.3.3. Arduino Leonardo Kartı

Bu kart analog A0 pim portundan ve dijital pim 7 portundan gelen verileri okumak üzere kod yazılarak programlanmıştır. XBee Düğüm II üzerinden ana düğüme yollamadan önce, gerekli mantık dizilerini oluşturur ve uyumlu bir format için gereken kalibrasyonları yapar.

Kod ayrıca, XBee Düğüm II ve XBee koordinatörü arasında bağlantının sağlandığını garanti edecek prosedürleri içerir ve buralardan gelen veya buralara giden verilerin işlenmesinden sorumludur.

4.1.3.4. XBee Uç Cihazı

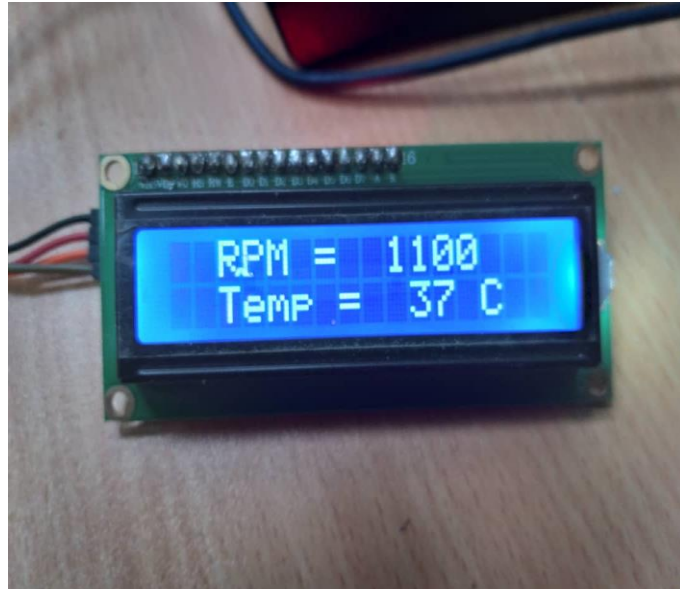
Bu cihaz Arduino Leonardo kartı ve ana düğüm arasında kablosuz olarak gerçekleşen veri alış-verişini gerçekleştirmekle sorumludur.

4.1.3.5. 16x2 LCD

Bu birim sensörler tarafından ölçülen motora ait ısı ve RPM-dakikadaki devir sayısını izlemek için kullanılmıştır.

Arduino Leonardo kartına pim 2 ve pim 3'ten bağlıdır.

Şekil 4.16'da düğüm II'in çıktılarının yerel olarak izlenişi görülmektedir. Burada motorun RPM ve ısı değerleri verilmektedir.



Şekil 4. 16. Düğüm II'nin Yerel İzlenişi

4.1.3.6. XBee Modüllerinin Yapılandırılması

Bu bölümde ZigBee ağında çalışmak üzere XBee modüllerine uygulanan yapılandırma parametrelerinden ve izlenen adımlardan bahsedeceğiz.

Burada XBee parametrelerini yapılandırmak için uluslararası Digi firmasının sağladığı Digi X-CTU yazılımının "Version 6.4.0" sürümü kullanılmıştır, bu yazılım ücretsiz olarak kullanılabilir.

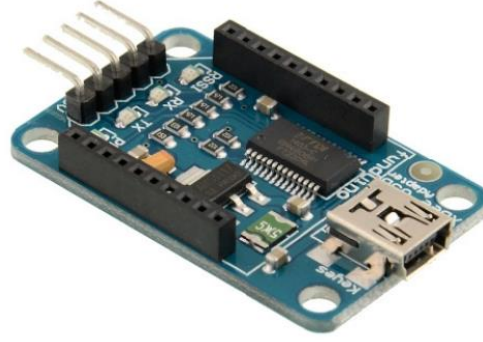
X-CTU bedava bir çoklu-platform uygulamasıdır ve Digi radyo frekansını (RF) geliştiricilerin basit bir grafik arayüzden yönetmelerine olanak verir. Bu uygulama Digi RF modüllerinin kurulumunu, yapılandırılmasını ve test edilmesini kolaylaştıran dahili araçlar ihtiva etmektedir [53]. X-CTU programının kısayolu Şekil 4.17'de görülmektedir.



Şekil 4. 17. X-CTU programına ait kısayol

4.1.3.7. Ağdaki Xbee Cihazlarının Yapılandırılması İçin İzlenen Adımlar

X-CTU kullanarak XBee'yi programlamadan önce, XBee explorer kartına ihtiyaç duyarız. Bu kart XBee parametrelerini yapılandırmak için kullanılır ve XBee ayarları formatını destekler. Ayrıca USB-TTL adaptörü olarak kullanılabilir. Şekil 4.18'de bir XBee explorer kartı örneği görülmektedir.

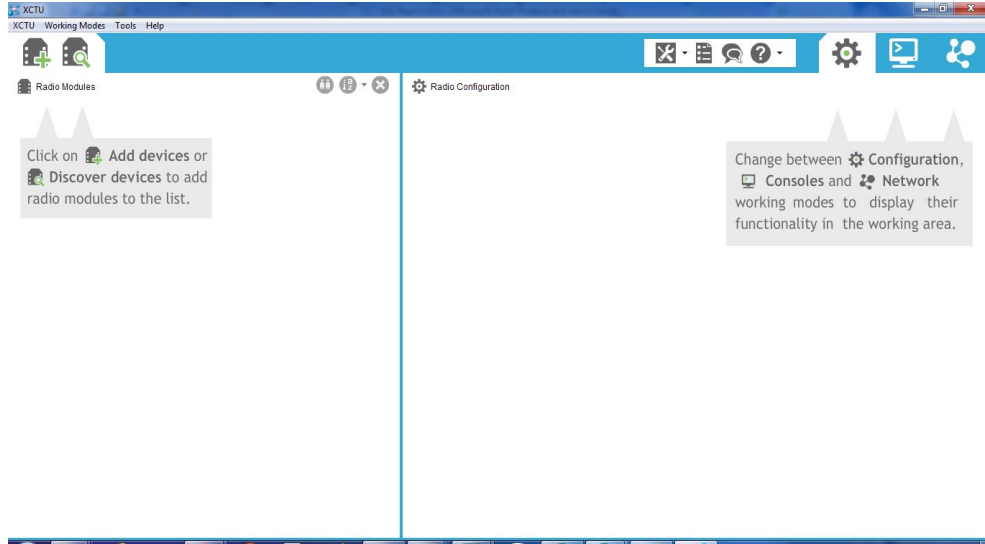


Şekil 4. 18. XBee Explorer Kartı

Bütün XBee modülleri bu karta kurulabilir veya eklenebilir.

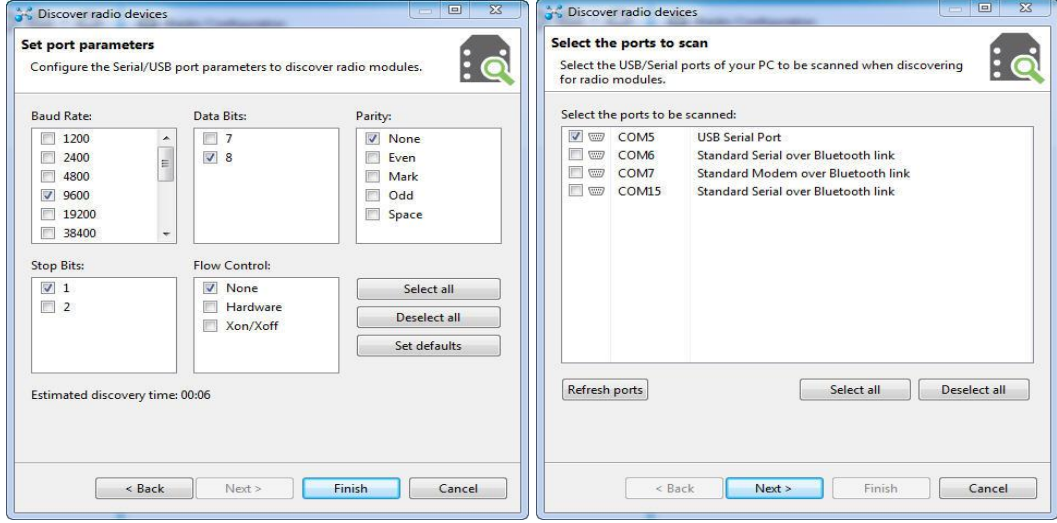
- 1- XBee'yi XBee explorer karta yerleştirdikten ve kartı PC'nin bir USB portuna taktıktan sonra X-CTU'yu çalıştırın.
- 2- Karta takılı olan XBee'yi eklemek için Discover Device sekmesini tıklayın

XCTU yazılımını Şekil 4.19'da görülmektedir.



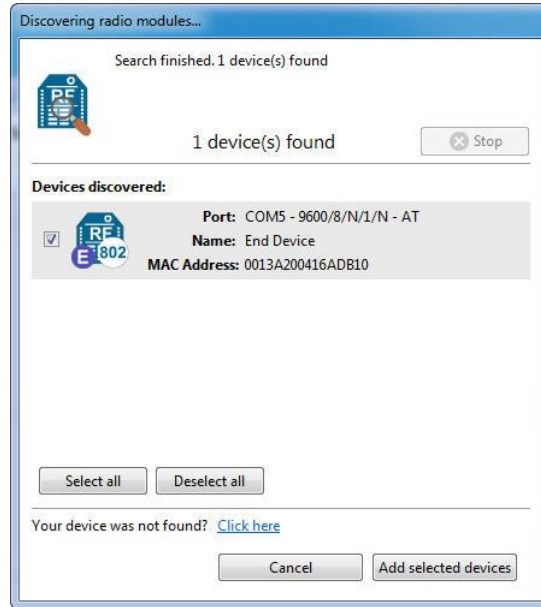
Şekil 4. 19. XCTU Yazılımı

- 3- Kartın bağlı olduğu seri portu seçin ve port parametrelerini Şekil 4.20'de görüldüğü gibi ayarlayın.



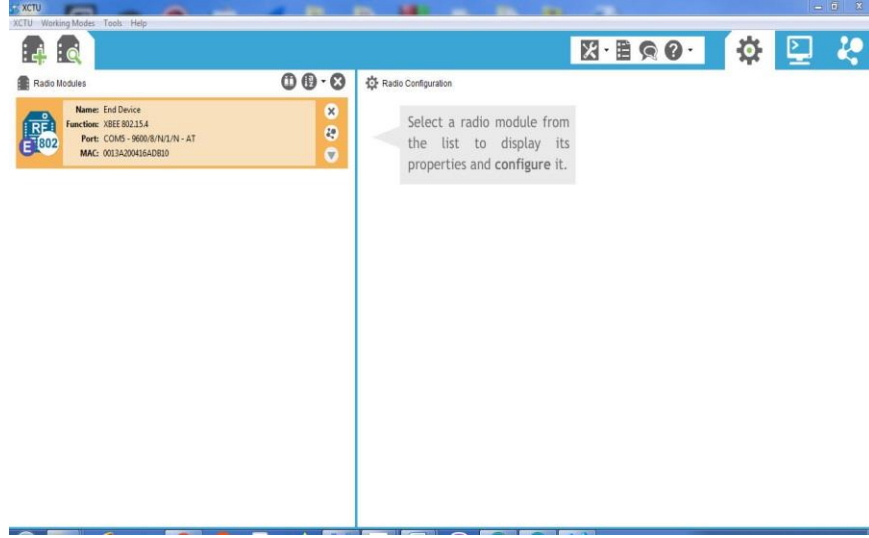
Şekil 4. 20. Seri Portun Seçilmesi Ve İşaretlenmesi

- 4- X-CTU bulana kadar XBee'yi aramaya devam edecektir. Bulunan XBee cihazını Şekil 4.21'de görüldüğü gibi Add selected devices sekmesine tıklayarak ekleyin.



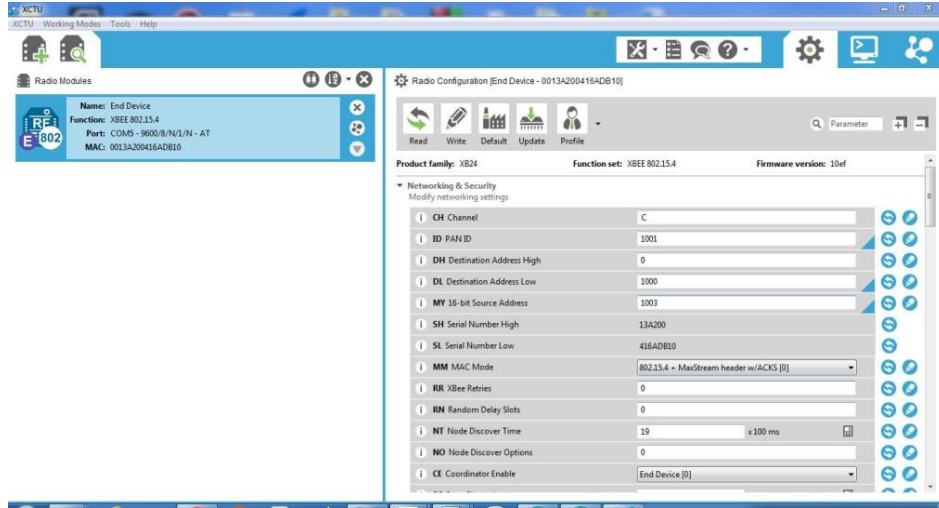
Şekil 4. 21. Bulunan Xbee Cihazının Eklenmesi

- 5- X-CTU, Şekil 4.22'de görülmektedir.



Şekil 4. 22. XBee'nin yazılım üzerinde çalışması

- 6- Program üzerinde bulunan XBee cihazına tıkladığında XBee parametreleri görülecektir, bu değerler Şekil 4.23'te görüldüğü gibi write sekmesine tıklanarak ve XBee'ye gerekli parametreleri yazarak istenen şekilde ayarlanabilir.



Şekil 4. 23. XBee Parametreleri

Ağımızdaki düğümleri yapılandırmak için kullandığımız parametreler Tablo 4.1'de verilmektedir.

Tablo 4. 1. Ağdaki düğümleri yapılandırmak için kullanılan parametreler

parametre	Koordinatör	Düğüm I	Düğüm II
Kullanılan Kanal	C	C	C
PAN ID	1001	1001	1001
Hedef Adres Üst (HD)	0	0	0
Hedef Adres Alt (LD)	FFFF	1000	1000
Cihaz Türü	Koordinator	Uç Cihaz	Uç Cihaz
16-bit Adres	1000	1002	1003
Mod	AT	AT	AT

4.1.3.8. Parametrelerin Tartışılması

Kullanılan Kanal: Birbirlerine bağlanabilmek için bütün düğümlerin aynı kanal ve PAN ID'ye sahip olması gerekir. PAN ID 16-bit'lik (4 hex sayısı) bir adrestir, PAN ID'nin İngilizce açılımı Personal Area Network Identification Number, yani Kişisel Ağ Kimlikleyici'dir, her bir ağ için benzersiz ve tektir.

Hedef adres üst (HD): bu 48 bit'lik XBee modülünün hedef düğüme ait uzun adresin 24 bitlik üst kısmıdır, yani "adresin üst kısmı".

Hedef adres alt (LD): bu 48 bit'lik XBee modülünün hedef düğüme ait uzun adresin 24 bitlik alt kısmıdır, yani "adresin alt kısmı".

Düğümün hedef adres alt kısmı, koordinatöre ait 16-bit'lik adres olmalıdır çünkü, tek nokta iletim modunda (unicast transmission mode) düğümler sadece koordinatörle bağlantı kurmaktadır, koordinatörün hedef adresi ise FFFF'tir, bunun anlamı koordinatörün ağdaki bütün düğümlerle çok nokta modunda (multicast mode) bağlantı kurduğudur.

Cihaz türü: XBee cihazının ağda nasıl çalıştığını tanımlar, yani "uç cihaz veya koordinatör" olarak.

Not: Her XBee ağının sadece bir tane PAN koordinatörü ve en az 1 en fazla 216 tane uç cihazı olmalıdır.

16-bit Adres: 4 hexadecimal ile temsil edilen XBee modülüne ait kısa adrestir, Kişisel Ağda benzersiz olması gerekir.

4.2. Yazılım Kısmı

Yazılım kısmı önerilen sistemin uygulanması için kullanılmaktadır. Sistemin uygulanması için kullanılan yazılım iki parçadan oluşur: Düşük seviye programlama ve yüksek seviye programlama.

4.2.1. Düşük Seviye Programlama

Düşük seviye programlama, kendisine bağlı birimlerden, yine kendi üzerindeki pimler aracılığıyla alınan veya iletilen sinyalleri işlemek üzere mikrodenetleyiciye yüklenen bir grup talimatı temsil eder. Bu program ayrıca, Arduino kartıyla düğümler arasında veri ve komut alışverişine olanak sağlayacak, ayrıca bu verilerden çıkarım yapacak, ve bu verileri kullanışlı bir veri formatına sokacak kodu da içermektedir. Arduino Mikrodenetleyiciyi açık kaynak Arduino Yazılım (IDE) derleyicisini "Arduino.cc" kullanarak programladık [54].

Geliştirdiğimiz sistemdeki her bir Arduino kartı sensörlerinden, yani “kendisine bağlı sensörlerden” gelen verileri işlemek üzere programlanmıştır. Her bir düğümün kodunu şu şekilde açıklayabiliriz:

4.2.1.1. Düğüm I kodu

Bildiğimiz gibi, düğüm I asenkron motora gelen gücü izlemek ve kontrol etmek üzere tasarlanmıştır.

İzleme süreci, aşağıdaki güç sensörleri tarafından gerçekleştirilir:

– Arduino Leonardo analog portlarına bağlı üç adet tek fazlı AC voltaj sensörü, A0 faz 1 voltajı için, A2 faz 2 voltajı için ve A4 faz 3 voltajı için.

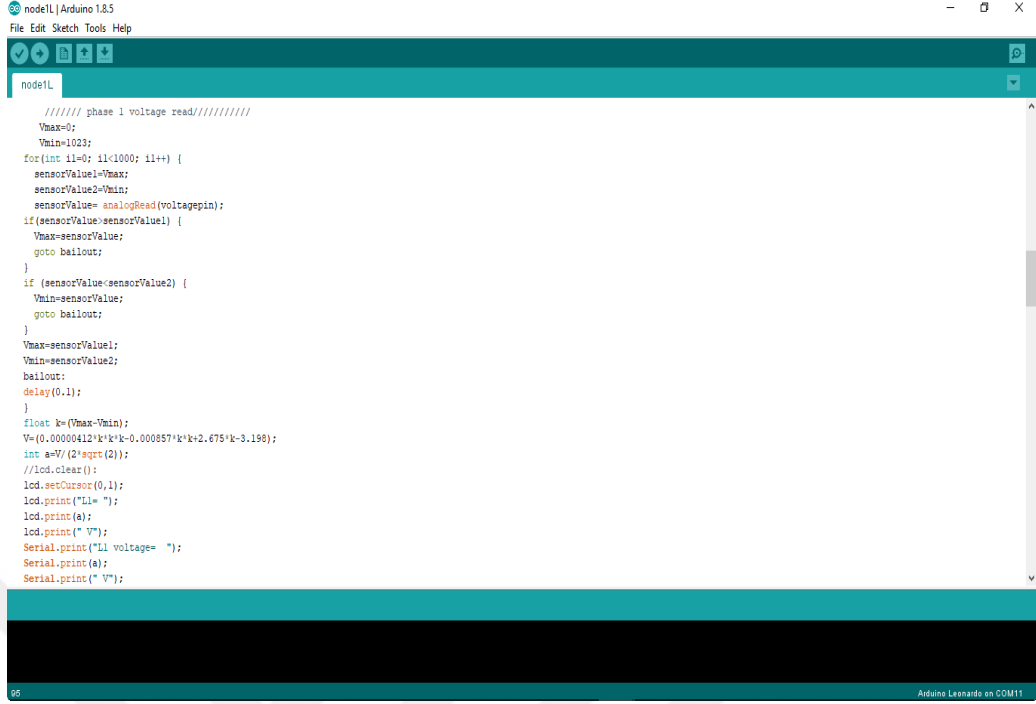
– Arduino Leonardo analog portlarına bağlı üç adet AC akım sensörü A1 faz 1 akımı için, A3 faz 2 akımı için ve A5 faz 3 akımı için.

Bütün bu sensörlerden gelen okumalar Arduino Leonardo dijital pim D0 "RX pin" ve pim D1 "TX pin"e bağlı olan XBee modülü tarafından işlenir. Aynı zamanda bütün bu okumalar, Arduino Leonardo dijital port D2 SDA LCD pimi ve Arduino Leonardo digital port D3 SCL LCD pimine bağlı 20x4 LCD ekrana yansıtılır.

Kontrol etme işlemi üç fazlı kontaktörü kontrol eden iki röle tarafından gerçekleştirilir.

Arduino Leonardo'daki "Relay in pins" D5 ve D6 portlarını dijital çıktı portları olarak işlev görecektir şekilde yapılandırdıktan sonra, veri girişi portlarına 0 mantık değeri olan "0 volt" uygulanarak iki röle aktif hale getirilebilir. Ayrıca, Arduino Leonardo kartındaki aynı dijital portlara, yani veri girişi portlarına 1 mantık değeri olan "5 volt" uygulanarak bu röleler devre dışı bırakılabilir.

Geliştirilen ve Arduino Leonardo kartına yüklenen kod, voltaj ve akım sensörleri tarafından algılanan, analog girdilerden (A0'dan A5'e kadar) gelen elektrik sinyallerini fiziksel okuma değerlerine dönüştürmek ve daha sonra bu verileri XBee modülünü kullanarak RF sinyalleriyle ana düğüme string formatında aktarmak için gerekli talimatları da içermektedir. String formatındaki veri şu bilgileri içermektedir: (faz 1 voltaj "3 tam sayı değeri" /faz 1 akım "3 reel sayı değeri" /faz 2 voltaj "3 tam sayı değeri" /faz 2 akım "3 reel sayı değeri" /faz 3 voltaj "3 tam sayı değeri" /faz 3 akım "3 reel sayı değeri"). Şekil 4.24'de faz 1 voltaj sensöründen gelen veriyi okumak için yazılan örnek kod görülmektedir.



```
nodeL | Arduino 1.8.5
File Edit Sketch Tools Help

nodeL

//////// phase 1 voltage read////////
Vmax=0;
Vmin=1023;
for(int i=0; i<1000; i++) {
  sensorValue1=Vmax;
  sensorValue2=Vmin;
  sensorValue= analogRead(voltagepin);
  if(sensorValue>sensorValue1) {
    Vmax=sensorValue;
    goto bailout;
  }
  if (sensorValue<sensorValue2) {
    Vmin=sensorValue;
    goto bailout;
  }
  Vmax=sensorValue1;
  Vmin=sensorValue2;
  bailout:
  delay(0.1);
}
float k=(Vmax-Vmin);
V=(0.00000412*k*k-k-0.000857*k*k+2.675*k-3.198);
int a=V/(2*sqrt(2));
//lcd.clear();
lcd.setCursor(0,1);
lcd.print("L1= ");
lcd.print(a);
lcd.print(" V");
Serial.print("L1 voltage= ");
Serial.print(a);
Serial.print(" V");
```

Şekil 4. 24. Faz 1 Voltaj Sensöründen Gelen Veriyi Okumak İçin Yazılan Örnek Kod

Ayrıca, geliştirilen kod, ana düğümden XBee modülüyle RF sinyalleriyle gönderilen özel komutlara göre röleleri açıp kapayacak bir grup talimatı da içermektedir.

4.2.1.2. Düşüm II Kodu

Düşüm II, üç fazlı asenkron motorda oluşacak hasarı erken öngörmede en sık kullanılan metotlardan biri olan motorun dakikadaki devir sayısını ve stator ısısını ölçümlemek üzere tasarlanmıştır.

Motorun dakikadaki devir sayısını (RPM) ölçümlemek üzere Infrared engellerden kaçınma sensörünü kullandık. Arduino Leonardo kartındaki D7 pimini dijital girdi olarak yapılandırdıktan sonra, Infrared engellerden kaçınma sensörü üzerindeki "in" pimini Arduino Leonardo kartında D7 pimiyle bağladık.

Ayrıca Arduino Leonardo kartındaki A0 analog pimini Lm35 ısı sensörünün "in" pimiyle bağladık.

Motorun dakikadaki devir sayısı ve stator ısısı, Arduino Leonardo kartında D2 ve D3 pimlerine baęlı olan 16x4 LCD ekranına yansıtılmaktadır. Farklı portlardan toplanan veriler D0 ve D1 pimlerinden seri baęlı XBee modülü kullanılarak RF ile iletilecektir.

Arduino Leonardo kartının D0 "RX" pimi XBee modülünün Dout pimine, ve Arduino Leonardo kartının D1 "TX" pimi XBee modülünün Din pimine baęlanmaktadır.

Geliştirilen ve Arduino Leonardo kartına yüklenen kod, motor RPM'sini hesaplamak için gerekli talimatları içermektedir, dakikadaki devir sayısını hesaplamak için kesme hizmet yordamı (Interrupt Service Routine) kullanılmıştır.

Her bir devir bir kesinti oluşturmaktadır, Arduino Leonardo kartında D7 pim giriři, her dönen kenarda bir sinyal vermektedir, bu pim girişini yapılandırmak için 4 nolu kesinti protokolünü kullandık, yani "pim D7'nin mantık durum 1'den mantık durum 0'a her deęişikliği kesinti başlangıcıdır". İki kesinti arasında kalan zamanı milisaniye cinsinden ölçmek için timer kullandık, ancak nihai sonuç dakika türünden ifade edilmektedir. Son olarak sunulan deęer, daha kararlı bir deęer elde edilmesi amacıyla, son 10 okumanın ortalaması olacaktır.

Ayrıca LM35 ısı sensörü tarafından algılanan ve analog girdi A0'dan gelen elektrik sinyalleri kod tarafından gerçek fiziksel okumalara dönüştürülmektedir, ve nihayetinde XBee modülü tarafından RF sinyalleriyle string formatında ana düğüme iletilmektedir. String dizisi řu bilgileri içermektedir: (ısı "4 reel sayı deęeri" / RPM "4 tam sayı deęeri). řekil 4.25'te motor RPM'sini hesaplamak için kullanılan örnek kod görülmektedir.

```
File Edit Sketch Tools Help
node2L6
//////////////////////////////////// Reading Proximity Sensor //////////////////////////////////////

hhh: if (millis() - lastmillis >= 1000){ /*Update every one second, this will be equal to reading frequency (Hz).*/
detachInterrupt(0); //Disable interrupt when calculating
total = 0;
samples[index] = rpmcount * 60; /* Convert frequency to RPM, note: this works for one interruption per full rotation.*/
for (int i=0; i<=9; i++){
total = total + samples[i];
}
average = total / numsamples;
rpm = average;

rpmcount = 0; // Restart the RPM counter
index++;
if(index >= numsamples){
index=0;
goto hhh;
}
if (millis() > 11000){ // wait for RPMs average to get stable
Serial.print(" RPM = ");
Serial.println(rpm);
lcd.clear();
lcd.setCursor(2,0);
lcd.print("RPM = ");
lcd.print(rpm);
a=rpm;
}

lastmillis = millis(); // Update lastmillis
attachInterrupt(4, motor, FALLING); //enable interrupt
}

83 Arduino Leonardo on COM11
```

Şekil 4. 25. Motor RPM'sini Hesaplamak İçin Örnek Kod

4.2.1.3. Ana Düğüm Kodu

Arduino IDE derleyicisini kullanarak Arduino Leonardo kartını bu düğüm için, bağlı olan sensörlerinden XBee düğümlerine gelen verileri alması, ve bunları seri port üzerinden string formatında ana PC'ye aktarması için programladık, iletilecek olan string dizisinde şu şekilde veriler yazılı olacaktır (faz 1 voltaj "3 tam sayı değeri" /faz 1 akım "3 reel sayı değeri" /faz 2 voltaj "3 tam sayı değeri" /faz 2 akım "3 reel sayı değeri" /faz 3 voltaj "3 tam sayı değeri" /faz 3 akım "3 reel sayı değeri" / ısı "4 reel sayı değeri" /RPM "4 tam sayı değeri").

Ayrıca yazılan kod, her bir düğümle ana XBee düğümü arasında bir bağlantıyı garanti altına alacak bağlantı mekanizmasıyla da donatılmıştır. Şekil 4.26'da ana Arduino Leonardo kartından PC'ye çıktı olarak yollanan veri görülmektedir.

```
221/0.45/220/0.60/219/1.23/1501/35.53/000/000/000/0000/
222/0.22/219/1.22/221/0.25/1250/25.00/000/000/000/0000/
220/1.14/224/1.22/219/0.25/1500/29.23/000/000/000/0000/
219/1.12/222/1.20/220/1.20/1460/30.00/000/000/000/0000/
221/0.45/220/0.60/219/1.23/1501/35.53/000/000/000/0000/
222/0.22/219/1.22/221/0.25/1250/25.00/000/000/000/0000/
220/1.14/224/1.22/219/0.25/1500/29.23/000/000/000/0000/
219/1.12/222/1.20/220/1.20/1460/30.00/000/000/000/0000/
221/0.45/220/0.60/219/1.23/1501/35.53/000/000/000/0000/
222/0.22/219/1.22/221/0.25/1250/25.00/000/000/000/0000/
220/1.14/224/1.22/219/0.25/1500/29.23/000/000/000/0000/
219/1.12/222/1.20/220/1.20/1460/30.00/000/000/000/0000/
221/0.45/220/0.60/219/1.23/1501/35.53/000/000/000/0000/
222/0.22/219/1.22/221/0.25/1250/25.00/000/000/000/0000/
220/1.14/224/1.22/219/0.25/1500/29.23/000/000/000/0000/
219/1.12/222/1.20/220/1.20/1460/30.00/000/000/000/0000/
221/0.45/220/0.60/219/1.23/1501/35.53/000/000/000/0000/
222/0.22/219/1.22/221/0.25/1250/25.00/000/000/000/0000/
220/1.14/224/1.22/219/0.25/1500/29.23/000/000/000/0000/
219/1.12/222/1.20/220/1.20/1460/30.00/000/000/000/0000/
```

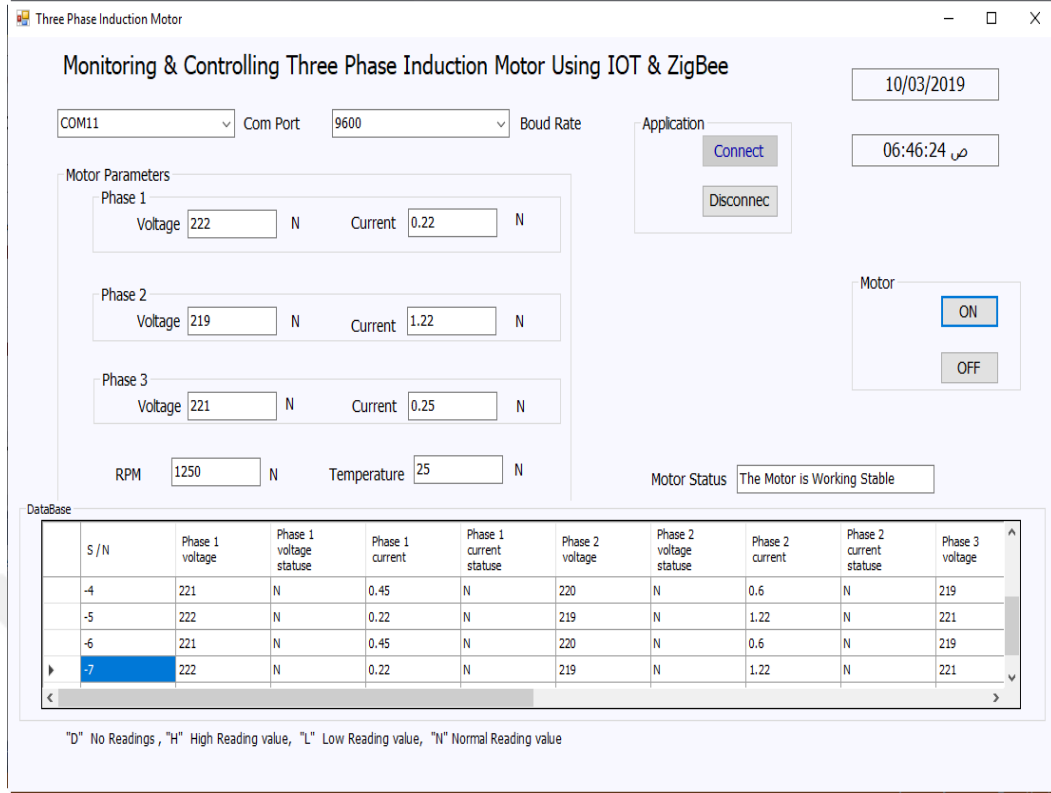
Şekil 4. 26. Ana Arduino Leonardo Kartından PC'ye Çıktı Olarak Yollanan Veri

4.2.2. Yüksek Seviye Programlama

Yüksek Seviye Programlama, sistemin farklı cihazlar üzerinden gerçek zamanlı olarak bir kullanıcı arayüzü ile izlenilebildiği ve kontrol edilebildiği anlamına gelir.

Daha önce belirttiğimiz gibi, bütün düğümlerden toplanan veriler COM portu üzerinden, string formatında ana PC'ye gönderilecektir. Grafik kullanıcı arayüzü ücretsiz ve açık kaynak kodlu bir yazılım olan Visual Studio 2017 Community sürümü ("Visual Basic. Net") kullanılarak hazırlanmıştır [55].

İki adet grafik kullanıcı arayüz sistemi tasarlanmıştır; bunların ilkinin kurulumu, doğrudan ana Arduino Leonardo kartına bağlı olan, sistemde ihtiyaç duyulan bütün servisleri sağlamaktan sorumlu bir sunucu olarak işlev görmek üzere yapılandırılan yerel ana PC'ye yapılmıştır. İkincisi ise, uzaktan erişim sağlayacak olan PC'lere yüklenmek üzere geliştirilmiştir, düşük seviyeli kullanıcılarına gerçek zamanlı olarak izleme imkanı verir, sistem yöneticilerine ise izlemenin yanısıra "NI'yi kullanma" imkanı verir. Şekil 4.27'de ana PC için tasarlanan grafik kullanıcı arayüzü görülmektedir.

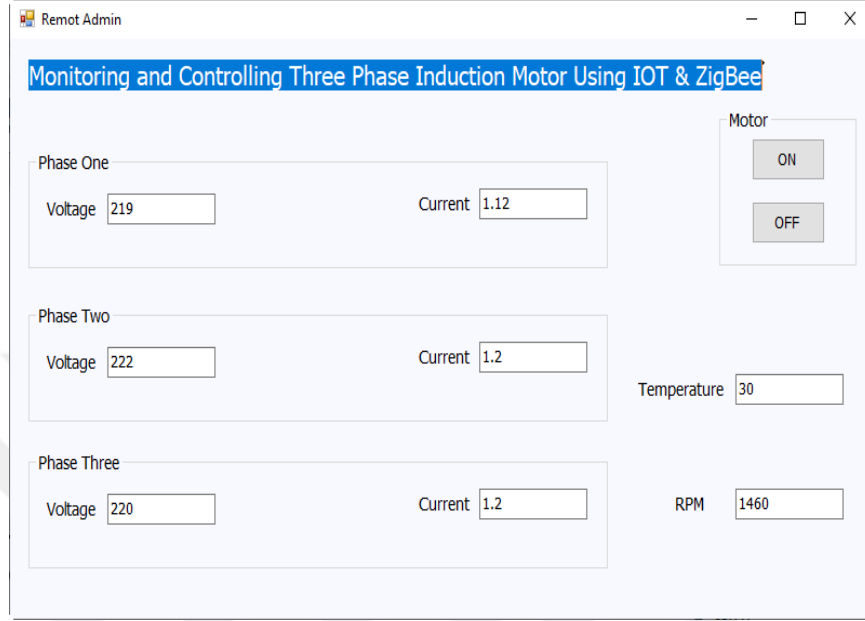


Şekil 4. 27. Ana PC İçin Tasarlanan Grafik Kullanıcı Arayüzü (GUI)

Ana PC için tasarlanan grafik kullanıcı arayüzü aşağıdaki görevleri yerine getirebilmektedir:

- 1- Seri portu Arduino Leonardo'dan gönderilen string veriyi alacak şekilde yapılandırma.
- 2- String veriden çıkarım yapma ve orijinal haliyle bu veriyi sistemde kullanma.
- 3- Çıkarımı yapılan veriyi, motorla ilgili bütün parametreleri ayrı ayrı ve net bir şekilde gösterecek şekilde bir formata sokma.
- 4- Çıkarımı yapılmış veriye dayanarak herhangi bir faaliyette bulunmadan önce, elde edilen değerlerin normal olup olmadığını anlamak üzere çeşitli testler yapma.
- 5- Kullanıcının motoru çalıştırmasına veya durdurmasına izin verme.
- 6- İlerde kullanmak amacıyla motora ait parametreleri kaydetmek üzere sistemin bir veritabanıyla bağlantısını sağlama.
- 7- Sistemin seçili bir bulut servisiyle bağlantısını sağlayarak Nİ uygulamalarına çalışabilecek şekilde sistemi yapılandırma, böylece buluttan veri indirebilme

veya bulutta verileri yayınlama. Şekil 4.28’de uzak PC için tasarlanan grafik kullanıcı arayüzü görülmektedir.



Şekil 4. 28. Uzak PC İçin Tasarlanan Grafik Kullanıcı Arayüzü

Farklı motor parametrelerinin kaydedileceği bir veritabanı geliştirmek amacıyla Microsoft Access kullanılmıştır. Şekil 4.29’da Microsoft Access kullanılarak tasarlanan veritabanı dosyası görülmektedir.

S/	Phase 1 vo	Phase 1 vo	Phase 1 cu	Phase 1 cu	Phase 2 vo	Phase 2 vo	Phase 2 cu	Phase 2 cu	Phase 3 vo	Phase 3 vo	Phase 3 cu	Phase 3 cu
0	0	D	0.00	D	0	D	0.00	D	0	D	0.00	D
3	219	N	1.12	N	222	N	1.20	N	220	N	1.20	N
4	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
5	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
6	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
7	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
8	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
9	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
10	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
11	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
12	219	N	1.12	N	222	N	1.20	N	220	N	1.20	N
13	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
14	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
15	219	N	1.12	N	222	N	1.20	N	220	N	1.20	N
16	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
17	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
18	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
19	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
20	219	N	1.12	N	222	N	1.20	N	220	N	1.20	N
21	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
22	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
23	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
24	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
25	219	N	1.12	N	222	N	1.20	N	220	N	1.20	N
26	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
27	220	N	1.14	N	224	N	1.22	N	219	N	0.25	N
28	221	N	0.45	N	220	N	0.60	N	219	N	1.23	N
29	222	N	0.22	N	219	N	1.22	N	221	N	0.25	N
30	219	N	1.12	N	222	N	1.20	N	220	N	1.20	N

Şekil 4. 29. Microsoft Access’le Tasarlanan Veritabanı Dosyası

4.3. Bulut Hesaplama Servisleri

Dahili olarak veya üçüncü bir taraf tarafından sunulan pek çok bulut servisi arasında en yaygın olanları şunlardır:

4.3.1. Software as a Service (SaaS-Hizmet Olarak Yazılım)

İlgili yazılım kullanıcının bilgisayarına yüklenmez, SaaS hizmetini veren şirketin sahip olduğu ve yönettiği bilgisayarlarda yüklü olur. Yazılıma ulaşım internet üzerinden sağlanır, ve genellikle kullanıcı bu hizmet için aylık olarak faturalandırılır.

4.3.2. Infrastructure as a Service (IaaS-Hizmet Olarak Altyapı)

Hesaplama, depolama, ağ ve diğer hizmetler (güvenlik, farklı araçlar) IaaS şirketi tarafından internet, VPN veya kullanıcıya özel ağ üzerinden sağlanır. Kullanıcılar ilgili sistemin, uygulamanın veya altyapı üzerindeki bilginin sahibidirler ve sağlanan hizmetleri kullandıkları ölçüde faturalandırılırlar.

4.3.3. Platform as a Service (PaaS-Hizmet Olarak Platform)

Bulut tabanlı uygulama için gerekli hem donanım hem de yazılım internet, VPN veya kullanıcıya özel ağ üzerinden PaaS şirketi tarafından sağlanır. Kullanıcılar platformu kullandıkları kadar faturalandırılırlar ve dolaşımında olduğu süre boyunca uygulamanın ne şekilde kullanılacağı kullanıcının kontrolündedir.

4.3.4. Nİ Sistemimizde Kullanılan Bulut Servisi

Bulut servisi olarak ücretsiz olan, SaaS usulü çalışan ve web üzerinden hizmet veren ThingSpeak'i kullandık; bu servis tasarımcılara sensör verilerini bulutta toplama ve

Nesnelerin İnterneti uygulamaları geliştirme imkanı sunmaktadır. Dahası kullanıcılara web üzerinden elde edilen verileri MATLAB® yazılımıyla analiz etme ve görselleştirme, ve bu verilere bağlı olarak hareket etme imkanı da sunulmaktadır. ThingSpeak bilgisayar, Arduino®, Raspberry Pi™, Beagle Bone Black ve daha pek çok donanım ile yollanan sensör verilerini işleyebilmektedir [56].

ThingSpeak'te kanal oluşturduğunuzda Nİ kullanarak elde ettiğiniz veriler üzerine çalışabilir ve daha fazlasını da yapabilirsiniz. Her bir kanal en fazla sekiz sensörden gelen veri üzerine çalışma ve yayınlama imkanı sunmaktadır, bu veri kümelerinin her biri "field" (alan) olarak isimlendirilmektedir. Her bir field tek bir sensörden gelen verilere tahsis edilmektedir.

ThingSpeak'te ücretsiz bir hesabınız olduğunda bulut sistemine toplam 3 milyon mesaj yükleme hakkınız olmaktadır, bu sınırı aşma durumunda bulut servisinden yararlanmaya devam edebilmek için belirli bir ücret ödemeniz gerekmektedir.

ThingSpeak'in bulut servisinden yararlanmak için öncelikle mathworks web sitesi adresinden bir hesap açmanız, daha sonra ise ThingSpeak'e girip bir kanal oluşturmanız gerekmektedir. Kanalınızı oluşturduktan sonra, field'lardan gelen verileri okumak, yüklemek veya yayınlamak için şu bilgilere ihtiyaç duyulacaktır:

- **Channel ID:** (Kanal kimliği) Thinkspeak.com'un size özel atadığı ThingSpeak kanal field'larında kullanacağınız tek bir numaradır.

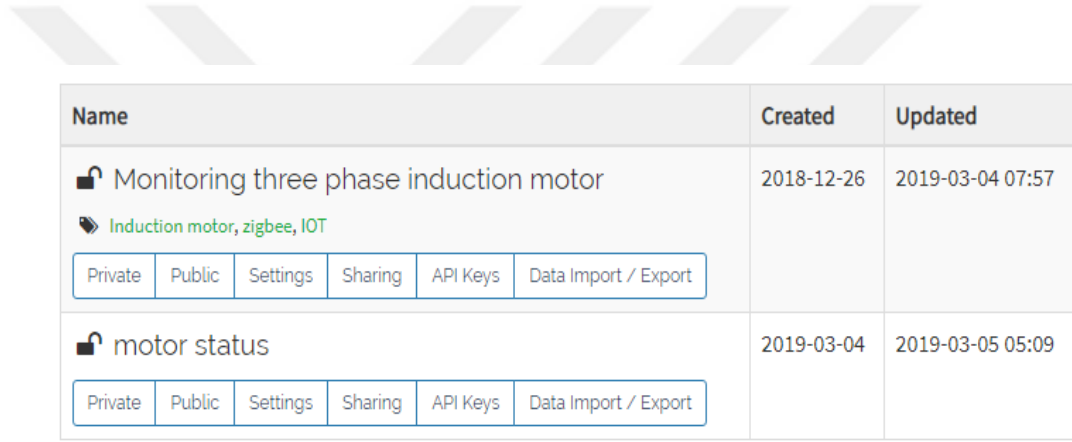
- **Read API key:** (Okuma API anahtarı) Bu anahtar, Channel ID'den sonra, field numarasından ise önce gelir ve ThingSpeak kanalınızın field'larında yer alan verileri okumak için kullanılır.



- **Write API key:** (Yazma API anahtarı) Bu anahtar, Channel ID'den sonra, field numarasından ise önce gelir ve ThingSpeak kanalınızın field'larına sensörden gelen verileri yazmak için kullanılır.

- **MQTT API key:** (MQTT API anahtarı) Bu anahtar, Channel ID'den sonra, field numarasından ise önce gelir ve ThingSpeak kanalınızın field'larındaki verileri yayınlamak veya bu verilere abone olmak için kullanılır.

Ayrıca kanalınızın erişimini herkese veya bazı seçilmiş kullanıcılara açabilir, veya kimseye erişim hakkı vermeyip sadece kendinize özel hale getirebilirsiniz ("private view").

ThingSpeak'te geliştirdiğimiz Nİ sistemi için iki adet ücretsiz kanal oluşturduk, bu kanalların Channel ID numaraları ise şöyledir: **662590** (sekiz adet field bulunmaktadır ve kanalın ismi "Monitoring three phase induction motor—Üç fazlı asenkron motorun izlenmesi" şeklindedir) ve **718912** (iki adet field bulunmaktadır ve kanalın ismi "motor status—motor durumu" şeklindedir). Şekil 4.30'da Thingspeak.com'da oluşturulan kanallar görülmektedir.



Name	Created	Updated
 Monitoring three phase induction motor <small>Induction motor, zigbee, IOT</small> Private Public Settings Sharing API Keys Data Import / Export	2018-12-26	2019-03-04 07:57
 motor status Private Public Settings Sharing API Keys Data Import / Export	2019-03-04	2019-03-05 05:09

Şekil 4. 30. Thingspeak.Com'da Oluşturulan Kanallar.

4.4. Üç Fazlı Asenkron Motorun İzlenmesi Kanalı (Monitoring three phase induction motor channel)

Bu kanal motor parametrelerini gerçek zamanlı olarak yayınlamak, kaydetmek ve izlemek için oluşturulmuştur.

Bu kanalda sekiz field oluşturulmuştur. Her bir field tek bir sensöre şu şekilde atanmıştır:

- Field 1 asenkron motora gelen faz 1 voltajına atanmıştır.
- Field 2 asenkron motora gelen faz 1 akımına atanmıştır.
- Field 3 asenkron motora gelen faz 2 voltajına atanmıştır.

- Field 4 asenkron motora gelen faz 2 akımına atanmıştır..
- Field 5 asenkron motora gelen faz 3 voltajına atanmıştır.
- Field 6 asenkron motora gelen faz 3 akımına atanmıştır..
- Field 7 asenkron motorun stator ısısına atanmıştır.
- Field 8 asenkron motorun dakikadaki devir sayısına (RPM) atanmıştır.

Thingspeak.com'da oluşturulan kanalda sunulan veriler Şekil 4.31'de görülmektedir.

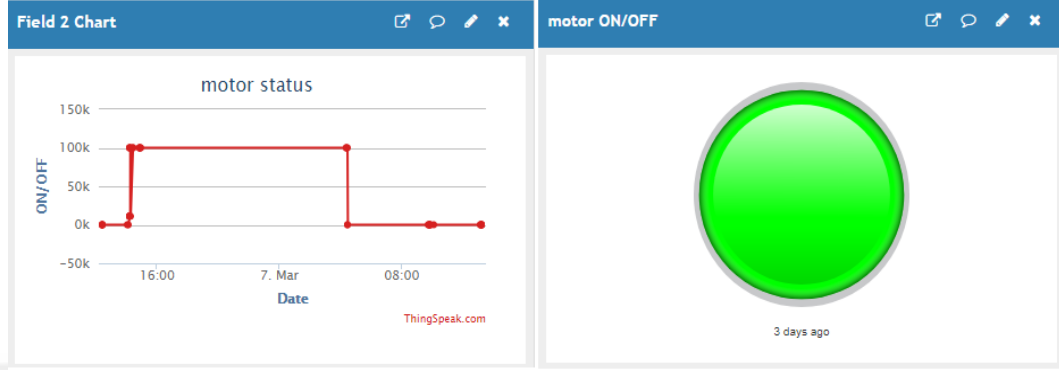




Şekil 4. 31. Thingspeak.Com'da Oluşturulan Kanalda Sunulan Veriler

Ve ikinci kanalda ise iki field oluşturduk, bu veriler motorun AÇIK veya KAPALI olmasına göre atanmıştır.

Thingspeak.com'da motor durumuyla alakalı kanalda sunulan veriler Şekil 4.32'de görülmektedir.



Şekil 4. 32. Thingspeak.Com'da Motor Durumu Kanalında Sunulan Veriler

Sistemden gelen sensör verilerini buluta yüklemek için ThingSpeak bulutun verdiği Write API key'i (yazma API anahtarını) kullandık. Bulutta yüklü olan kayıtlı verileri sisteme indirmek içinse Read API key'i (okuma API anahtarını) kullandık.

4.5. MQTT Protokolü (Message Queuing Telemetry Transport- Mesaj Kuyruk Telemetri İletimi)

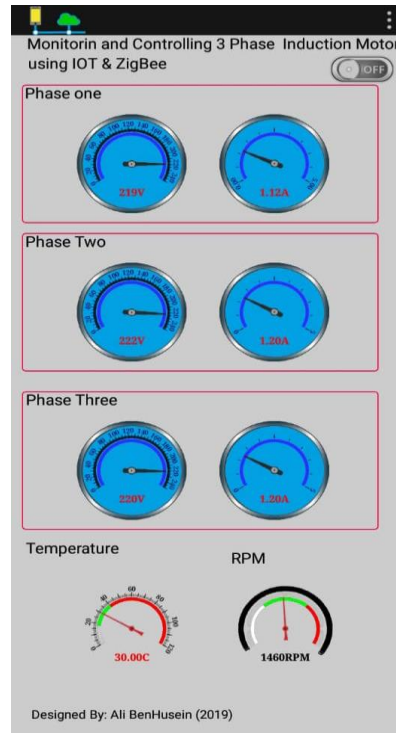
Mesaj Kuyruk Telemetri İletimi (Message Queuing Telemetry Transport) protokolü, MQTT sunucusu içinde, istemci ve sunucu arasında veya istemci ve istemci arasında veri alışverişi için yayımla/abone ol işlemlerini kullanan hafif bir mesajlaşma protokolüdür [57].

Kullanımında ve yazılım uygulamalarındaki kolaylığından dolayı Nİ projelerinde yaygın bir şekilde kullanılmaktadır.

Android işletim sistemine sahip cihazlarda sistemi uzaktan izlemek ve kontrol edebilmek için grafik kullanıcı arayüzü olarak bir Android uygulaması tasarladık. Bu grafik kullanıcı arayüzü VirtuinoMQTT Application Basic edition sürümü

kullanılarak tasarlanmıştır, ve bu program Nİ uygulamaları için kullanılan bir yardımcı yazılımdır [58].

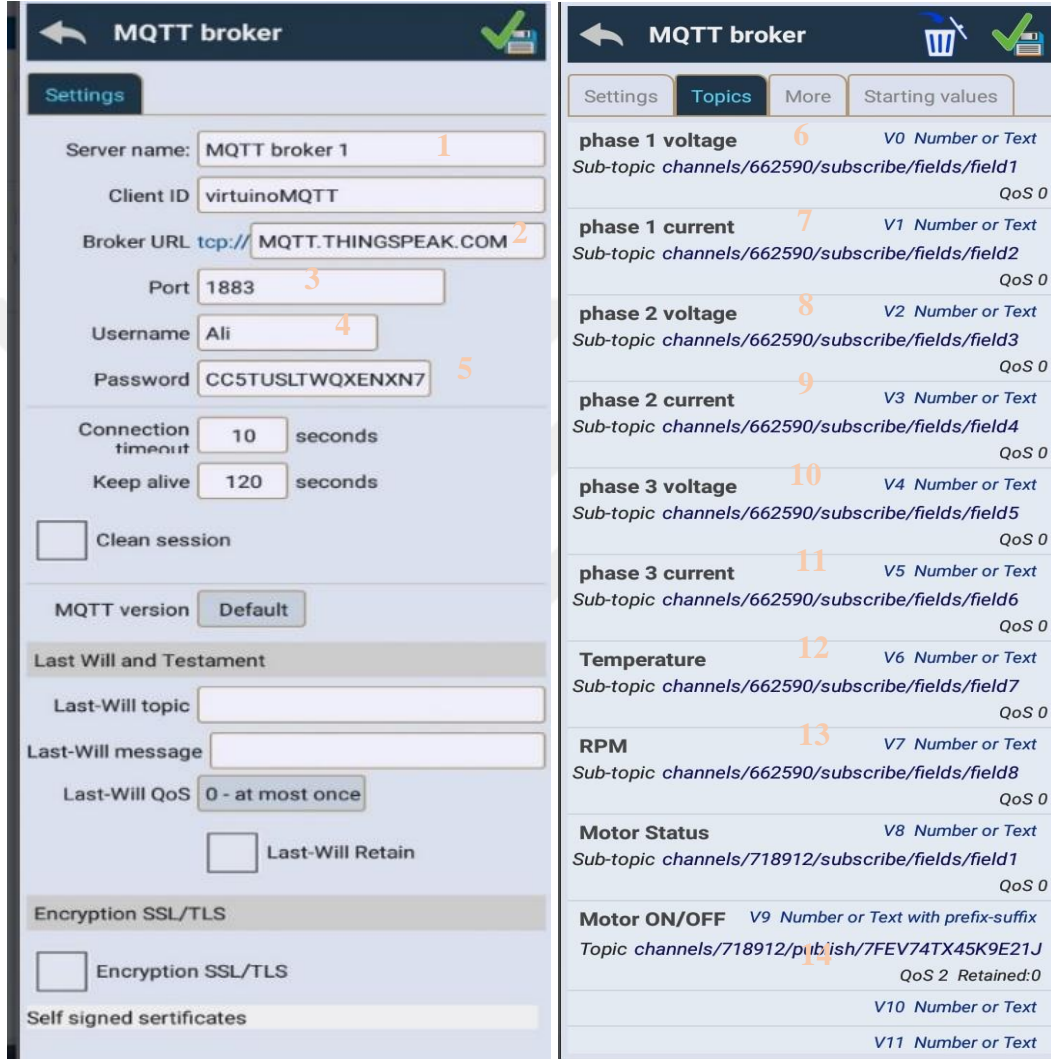
Bu uygulama motor verilerini thingspeak.com'da bulunan bulut kanalına yazmak (yayınlamak)—veya bulut kanalından almak için—veya okumak (abone olmak) için MQTT protokolünü, thingspeak.com'dan sağlanan MQTT broker'ını (kontrol eden yönetici) ve MQTT anahtarıyla beraber kullanır. Thingspeak.com'dan verilere MQTT broker'ıyla abone olmak içinse mesajın “sub-topic channels/channel number/subscribe/fields/field number” (kanal alt konusu/kanal numarası/subscribe/fieldlar/field numarası) formatında olması gerekmektedir. Thingspeak.com kanalında MQTT broker'ıyla verileri yayınlamak içinse mesajın “topic channels/channel number/Publish/channel password” (kanal konusu/kanal numarası/Publish/kanal şifresi) formatında olması gerekmektedir. Android işletim sistemi kullanan cihazlar için tasarlanan grafik kullanıcı arayüzü Şekil 4.33'te görülmektedir.



Şekil 4. 33. Android İşletim Sistemi Kullanan Cihazlar İçin Tasarlanan Grafik Kullanıcı Arayüzü

- VirtuinoMQTT Uygulamasını thingspeak.com MQTT broker'ından gelen verileri işleyebilecek şekilde yapılandırmak

MQTT broker'ının yapılandırılması Şekil 4.34'te görülmektedir.



Şekil 4. 34. MQTT Broker'ının Yapılandırılması

Şekil 4.34 üzerinden numaralandırılmış alanları şu şekilde açıklayabiliriz:

- (1) numaralı alan VirtuinoMQTT'de yapılandırılan sunucu adını içermektedir.
- (2) numaralı alan broker'ın tcp://mqtt.thingspeak.com olan URL adresini içermektedir.

- (3) numaralı alan MQTT protokolü için kullanılan port numarasını içermektedir.
- (4) ve (5) numaralı alanlar broker kullanıcı adını ve şifresini içerir.
- (6), (7), (8), (9), (10), (11), (12) ve (13) numaralı alanlar sekiz field içeren 662590 numaralı kanalına abonelik için gerekli yapılandırma ayarlarını içerir, her bir field tek bir konuya ayrılmaktadır. Örneğin, field1 faz 1 voltajı ve field2 faz 2 akımı için ayrılmıştır.
- (14) numaralı alan bulut kullanarak uzaktan motoru başlatan/durduran komutları yayınlamak için gerekli yapılandırma bilgilerini içerir.



5. SONUÇ VE GELECEK ÇALIŞMALAR

5.1. Sonuç

Nesnelerin İnterneti, fiziksel nesnelerin birbirine veya daha büyük bir sisteme bağlı olduğu bir ağıdır. Emsalsiz bir tanımlayıcı ile işaretlenen nesnelerin internet altyapısı üzerinden birlikte çalışabileceği ve böylece küçük parçaların toplamından daha büyük değerlerin elde edileceği öngörülmektedir. Bu tez çalışmasında, Nesnelerin İnterneti ve Komünikasyon Teknolojileri kullanılarak, şu an kullanılmakta olan ve sadece yerel olarak izleme imkanı veren sistemin yerine, bir web uygulaması üzerinden üç fazlı asenkron motoru gerçek zamanlı olarak kontrol eden, ölçümleyen ve izleyen düşük maliyetli bir sistem tasarlanmıştır; bu amaçla ayrıca Arduino ve Zigbee de kullanılarak kablosuz bir sensör ağı da tasarlanmıştır.

Önerilen sistem ayrıca fazla akım, sargılarda aşırı ısınma ve düşük/yüksek voltaj gibi bazı arızalara karşı motoru koruma amaçlı da kullanılabilir.

Sinyal uyumlaması, veriyi ZigBee alıcısına ileten Zigbee verici modülüne bağlı olan Arduino'ya aktarılır. Bunun neticesinde, alınan veri dünyanın herhangi bir yerinden ulaşılabilir hale gelmiş olur.

Bu sistemin başlıca fonksiyonu yük voltajına, yük akımına, motorun hızına ve ısısına dair verileri izlemektir, ve sistemin motoru kontrol etme ve anormal şartlar altında motoru kapatma yeteneği bulunmaktadır. Sistemin bir diğer yeteneği gerçek zamanlı olarak verileri elde etme ve ileride kullanılabilir şekilde muhafaza etmesidir. Asenkron motorun aktüatör olarak kullanıldığı Endüstri ve Ev Otomasyonu uygulamalarında önerilen projenin büyük yardımı olacaktır.

Tamamlanan çalışma yazılım ve donanım şeklinde iki kısımdan oluşmaktadır. Donanım kısmına, önerilen sistemin yürütülebilmesi için gerekli olan motorun ısısını ve hızını okumak için bir dizi sensör (hız sensörü ve ısı sensörü), akım ve voltaj ölçümü ve kontrolü için akım ve güç transformatörü, ve motoru açıp kapamak için

kontaktör ve röleler dahildir. Sensörlerden ve transformatörlerden elde edilen bütün veriler dahili donanıma, Arduino Leonardo'ya iletilmiştir.

Yazılım kısmında ise; farklı sensörlerden Xbee düğümüne gelen verilerin işlenmesi için alt düzey bir kod, thingspeak.com bulut sistemine/sisteminden motor parametrelerini çıkarmak, işlemek, kaydetmek, almak, istemek ve yayınlamak içinse üst düzey bir kod tasarlanmıştır. Ana düğümden buluta Nesnelerin İnterneti (Nİ) verilerini iletmek için ThingSpeak'e ait API ("Application Programming Interface") ve web servisi kullanılmıştır. Ayrıca android uygulamasını tasarlamak için Virtuino MQTT platformu kullanılmıştır. Motorun izlenmesi ve kontrol edilmesi için tasarlanan android uygulaması test edilmiştir.

5.2. Gelecek Çalışmalar

Gelecek çalışmalar kapsamında, önerilen metodu bir fabrika ortamında, üretim hattındaki makineleri, aktüatörleri ve diğer ekipmanları kontrol etmede ve nihayetinde herhangi bir fabrikada çıkabilecek sorunları kontrol altında tutma amaçlı olarak kullanıp test edebiliriz. Ayrıca, ana birim olarak bir PC yerine Raspberry Pi 4-Model B'yi kullanabiliriz, bu cihazın Arduino Leonardo kartının bağlanmasına imkan veren USB portu, ve buluta bağlanabileceği bir WiFi'ı bulunmaktadır. Dahası, yazılım hatası gibi durumlarda uzaktan sorun giderebilmek amacıyla dünyanın herhangi bir yerinden ana birime (PC veya Raspberry Pi) erişim için statik bir IP Adresi kullanarak sistemi daha da geliştirebiliriz.

KAYNAKLAR

- [1] L. Atzori, A. Iera, & G. Morabito, (2010), "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787-2805, 2010.
- [2] D. Bandyopadhyay & J. Sen, (2011), "Internet of things: Applications and challenges in technology & standardization," *Wireless personal communications*, vol. 58, no. 1, pp. 49-69, 2011.
- [3] G. Kortuem, F. Kawsar, V. Sundramoorthy, & D. Fitton, (2009), "Smart objects as building blocks for the internet of things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44-51, 2009.
- [4] S. Li, L. Da Xu, & S. Zhao, (2015), "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243-259, 2015.
- [5] L. Mainetti, L. Patrono, & A. Vilei, (2011), "Evolution of wireless sensor networks towards the internet of things: A survey," in *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*, 2011: IEEE, pp. 1-6.
- [6] P. Sethi & S. R. Sarangi, (2017), "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.
- [7] M. M. Zanjireh & H. Larijani, (2015), "A survey on centralised and distributed clustering routing algorithms for WSNs," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015: IEEE, pp. 1-6.
- [8] M. Conner, (2010), "Sensors empower the " Internet of Things", " *EDN (Electrical Design News)*, vol. 55, no. 10, p. 32, 2010.
- [9] P. Magrassi, A. Panarella, N. Deighton, & G. Johnson, (2001), "Computers to acquire control of the physical world," *Gartner research report T-14-0301*, 2001.
- [10] R. H. Weber, (2009), "Internet of things–Need for a new legal environment?," *Computer law & security review*, vol. 25, no. 6, pp. 522-527, 2009.
- [11] K. Ashton, (2009), "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97-114, 2009.
- [12] J. Rivera & R. van der Meulen, (2013), "Gartner says the internet of things installed base will grow to 26 billion units by 2020," *Stamford, conn., Aralık*, vol. 12, 2013.
- [13] M. Than, (2013), "Billion devices will wirelessly connect to the internet of everything in 2020," *ABI Research*, vol. 30, Mayıs 2013.
- [14] J.-B. Waldner, (2013), *Nanocomputers and swarm intelligence*. John Wiley & Sons, 2013.
- [15] S. Wang, (2009), *Intelligent buildings and building automation*. Routledge, 2009.
- [16] B. Henderson, (2014), *Rethinking the Internet of Things: a scalable approach to connecting everything*. Apress, 2014.

- [17] I. Lee & K. Lee, (2015), "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431-440, 2015.
- [18] C. Perera, C. H. Liu, & S. Jayawardena, (2015), "The emerging internet of things marketplace from an industrial perspective: A survey," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 585-598, 2015.
- [19] S. Vongsingthong & S. Smachat, (2014), "Internet of things: a review of applications and technologies," *Suranaree Journal of Science and Technology*, vol. 21, no. 4, pp. 359-374, 2014.
- [20] C. Bormann, M. Ersue, & A. Keranen, (2014), "Terminology for constrained-node networks," *Internet Engineering Task Force (IETF): Fremont, CA, USA*, pp. 2070-1721, 2014.
- [21] S. Mitchell, N. Villa, M. Stewart-Weeks, & A. Lange, (2013), "The Internet of everything for cities: connecting people, process, data and things to improve the livability of cities and communities," *San Jose: Cisco*, 2013.
- [22] P. Shankara, P. Mahanta, E. Arora, & G. Srinivasamurthy, (2015), "Impact of internet of things in the retail industry," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2015: kemer ayađı, pp. 61-65.
- [23] A. Andrushevich, B. Copigneaux, R. Kistler, A. Kurbatski, F. Le Gall, & A. Klapproth, (2013), "Leveraging multi-domain links via the internet of things," in *Internet of Things, Smart Spaces, and Next Generation Networking: kemer ayađı*, 2013, pp. 13-24.
- [24] D. Kyriazis, T. Varvarigou, D. White, A. Rossi, & J. Cooper, (2013), "Sustainable smart city IoT applications: Heat and electricity management & Eco-conscious cruise control for public transportation," in *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, 2013: IEEE, pp. 1-5.
- [25] S. Eggimann *et al.*, (2017), "The potential of knowing more: A review of data-driven urban water management," *Environmental science & technology*, vol. 51, no. 5, pp. 2538-2553, 2017.
- [26] X.-F. Xie & Z.-J. Wang, (2017), "Integrated in-vehicle decision support system for driving at signalized intersections: A prototype of smart IoT in transportation," 2017.
- [27] A. Witkovski, A. Santin, V. Abreu, & J. Marynowski, (2017), "An IdM and key-based authentication method for providing single sign-on in IoT," in *IEEE Global Communications Conference (GLOBECOM)*, 2015: IEEE, pp. 1-6.
- [28] L. Clark, (2017), "Oxford Nanopore: we want to create the internet of living things," ed, 2017.
- [29] R. Guhathakurta, (2016), "How IoT's Are Changing the Fundamentals of'Retailing'," *Trak. in-Business of Tech, Mobile & Startups in India*, 2016.

- [30] D. Rose, (2014), *Enchanted objects: Design, human desire, and the Internet of things*. Simon & Schuster, 2014.
- [31] D. J. Cappelleri, J. F. Keller, T. Kientz, P. Szczesniak, & V. Kumar, (2009), "Saast robotics: An intensive three week robotics program for high school students," in *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2009: American Society of Mechanical Engineers Digital Collection, pp. 587-595.
- [32] J. Jorup, (2016), "internet of things" security is hilariously broken and getting worse," ed: Haziran, 2016.
- [33] M. M. B. z. M. V. vithal, (2015), "Microcontroller Based Protection and control of Three-Phase Induction Motor," *International Journal on Recent and Innovation Trends in computing and communication(IJRITCC)*, vol. 3, no. 11, pp. 6287-6292, Aralık 16, 2015.
- [34] M. Bodkhe & K. Pawar, (2017), "Monitoring and Control System for Three Phase Induction Motor Using Poly Phase Multifunction Energy Metering IC ADE7758 and Zigbee Protocol."
- [35] M. F. Işık, M. R. Haboğlu, & B. Yartaşı, (2017), "Smart Phone Based Energy Monitoring System for 3 Phase Induction Motors," 2017.
- [36] M. P. Bodkhe & K. Pawar, (2014), "Parameter Monitoring Using Zigbee Protocol for Three Phase Induction Motor," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 1, pp. 73-77, 2014.
- [37] P. Geetha & V. Saravanan, (2012), "Online Parameter Monitoring Of Induction Motor Using Wireless Network," *International Journal on Advanced Computer Theory and Engineering (IJACTE)*, vol. 1, no. 2, pp. 1-8, 2012.
- [38] V. Rekha & K. S. Ravi, (2015), "Induction Motor Condition Monitoring and Controlling Based on IoT," *International Journal of Electronics, Electrical and Computational System*, vol. 6, no. 9, pp. 74-89, 2015.
- [39] R. Buyya & A. V. Dastjerdi, (2016), *Internet of Things: Principles and paradigms*. Elsevier, 2016.
- [40] (2018), "Internet of Things (IoT) History," *Postscapes, Edited Aug*, vol. <https://www.postscapes.com/internet-of-things-history>, Aralık 2018.
- [41] (2018) <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.
- [42] C. W. Group, (2014), "Constrained Application Protocol (CoAP) RFC 7252," *Retrieved*, vol. 20, pp. 1-9, 2014.
- [43] D. Soni & A. Makwana, (2017), "A survey on mqtt: a protocol of internet of things (IoT)," in *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*, 2017.
- [44] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, & R. Al-Hatmi, (2017), "Internet of Things: Survey and open issues of MQTT protocol,"

- International Conference on Engineering & MIS (ICEMIS)*, 2017: IEEE, pp. 1-6.
- [45] Q. Zhang, L. Cheng, & R. Boutaba, (2010), "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications*, vol. 1, no. 1, pp. 7-18, 2010.
- [46] F. Berman, G. Fox, & T. Hey, (2003), "Overview of the book: grid computing—making the global infrastructure a reality," *Making the Global Infrastructure a Reality*, p. 3, 2003.
- [47] J. W. Rittinghouse & J. F. Ransome, (2017), *Cloud computing: implementation, management, and security*. CRC press, 2017.
- [48] B. Furht & A. Escalante, (2018), *Handbook of cloud computing*. kemer ayağı, 2010.
- [49] (2018), "Arduino Leonardo With Headers," <https://store.arduino.cc/usa/arduino-leonardo-with-headers>, Haziran 2018.
- [50] (2018), A. SCT-013-000 Split core current transformer, "<https://www.poweruc.pl/collections/split-core-current-transformers2/products/split-core-current-transformer-sct013-rated-input-5a-100a>," Eylül 2018.
- [51] (2018), A. ZMPT101B AC single phase voltage sensor module specifications and features, "https://www.aliexpress.com/item/Active-Single-Phase-Voltage-Transformer-Module-AC-Output-Voltage-Sensor-for-Arduino-Mega/32347399136.html?src=google&albch=search&acnt=479-062-3723&isdl=y&aff_short_key=UneMJZVf&albcpl=266121556&albag=7593673036&slnk=&trgt=aud-30186416528:dsa-42862830006&plac=&crea=64152518476&netw=g&device=c&mtctp=b&emo1=1t1&albbt=Google_7_search&aff_platform=google&gclid=EAIaIQobChMItdncztPn3QIVwR0YCh3pag1AEAAAYASAAEgLfX_D_BwE" Ağustos 2018.
- [52] (2018), A. Electromechanical Relays, Internet sayfası "<https://www.allaboutcircuits.com/textbook/digital/chpt-5/contactors/>," Haziran 2018.
- [53] (2018), A. XCTU User Guide, "<https://www.digi.com/resources/documentation/digidocs/90001458-13/default.htm>," Mayıs 2018.
- [54] (2018), A. ARDUINO 1.xx The open-source Arduino Software (IDE), "<https://www.arduino.cc/en/Main/Software>," Haziran 2018.
- [55] (2018), "Visual Studio 2017 Community free download, Internet sayfası," <https://visualstudio.microsoft.com/downloads/>, Eylül 2018.
- [56] (2019), A. get started with Thing speak, Internet sayfası, "<https://www.mathworks.com/help/thingspeak/>," Kasım 2018.
- [57] (2019), A. MQTT Protokol - Nasıl Çalışır, "<https://1sheeld.com/mqtt-protocol/>" Mart 2019.
- [58] (2019), A. Virtuino MQTT internet sayfası, <https://virtuino.com/index.php/virtuino-mqtt>, Şubat. 2019.

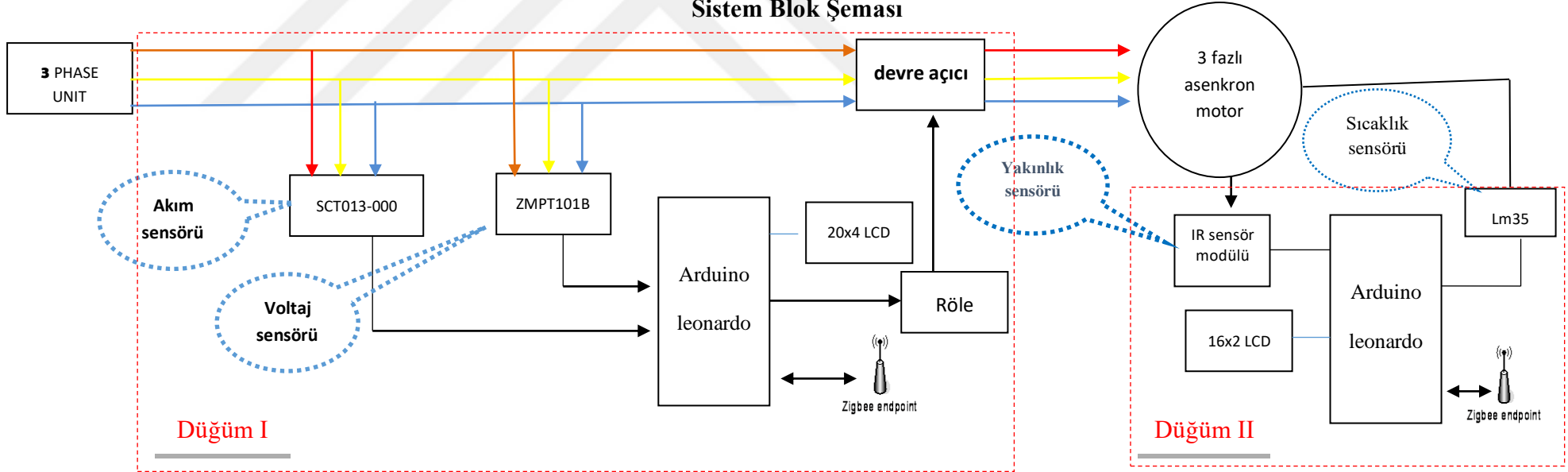
EKLER

EK 1	Sistem Blok Şeması
EK 2	Düğüm I Düşük Seviye Kodu
EK 3	Düğüm II Düşük Seviye Kodu
EK 4	Ana Düğüm Düşük Seviye Kodu
EK 5	Ana PC İçin Yüksek Seviye Visual Studio Kodu
EK 6	İş İstasyonu İçin Yüksek Seviye Visual Studio Kodu



EK 1

Sistem Blok Şeması



EK 2

Düğüm 1 Düşük Seviye Kodu

```
String address = "ADDRESS1";

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include "EmonLib.h"          // Include Emon Library

EnergyMonitor emon1,emon2,emon3;

LiquidCrystal_I2C lcd(0x27,20,4);

const float Ref=55.3;

const int voltagepin=A0, voltage1pin=A2, voltage2pin=A4 ;

double sensorValue=0, sensor1Value=0, sensor2Value=0;

double sensorValue1=0, sensor1Value1=0, sensor2Value1=0;

double sensorValue2=0, sensor1Value2=0, sensor2Value2=0;

float V, Vrms, V1, Vrms1, V2, Vrms2;

int crosscount=0;

int climbhill=0;

double Vmax, Vmax1, Vmax2 ;

double Vmin, Vmin1, Vmin2;

double Irms, Irms1, Irms2;

String z,q="A";

String r="/" ,s="u",o="u",t="u";

char re_address[20],v;

String receive_address="";

String data_address="";

boolean ok = false;
```

EK 2'in Devamı

```
//int temp,hum;

void tranceiver_data()

{

    int i = 0;

    receive_address = "";

    data_address="";

    if (Serial1.available() > 0)

    {

        while (Serial1.available() > 0)

        {

            re_address[i] = Serial1.read();

            delay(1);

            if(i<=7) receive_address += String(re_address[i]);

                else if(i<=10) data_address+=String(re_address[i]);

            i++;

        }

        Serial1.flush();

    }

    if(receive_address==address)

        ok=true;

}

void setup()

{

    lcd.init();           // initialize the lcd
```


EK 2'in Devamı

```
lcd.backlight();

lcd.setCursor(2,0);

lcd.print("power monitoring");

//pinMode(4,OUTPUT);

pinMode(5,OUTPUT); // line 1 relay

pinMode(6,OUTPUT); // line 2 relay

//pinMode(7,OUTPUT);

emon1.current(1, Ref); // Current: input pin, calibration.

emon2.current(3, Ref);

emon3.current(5, Ref);

pinMode(13,OUTPUT);

digitalWrite(6,HIGH);

digitalWrite(5,HIGH);

Serial.begin(9600);

Serial1.begin(9600);

}

void loop()

{

if (!ok)

trancever_data();

else

{

//////// phase 1 voltage read//////////
```

EK 2'in Devamı

```
Vmax=0;
Vmin=1023;
For (int i1=0; i1<1000; i1++) {
    sensorValue1=Vmax;
    sensorValue2=Vmin;
    sensorValue= analogRead(voltagepin);
    if(sensorValue>sensorValue1) {
        Vmax=sensorValue;
        goto bailout;
    }
    if (sensorValue<sensorValue2) {
        Vmin=sensorValue;
        goto bailout;
    }
    Vmax=sensorValue1;
    Vmin=sensorValue2;
    bailout:
    delay(0.1);
}
float k=(Vmax-Vmin);
V=(0.00000412*k*k*k-0.000857*k*k+2.675*k-3.198);
int a=V/(2*sqrt(2));
//lcd.clear();
lcd.setCursor(0,1);
```

EK 2'in Devamı

```
lcd.print("L1= ");
lcd.print(a);
lcd.print(" V");
Serial.print("L1 voltage= ");
Serial.print(a);
Serial.print(" V");
delay(2);
        //////////////// line 1 current read //////////////////////////////////////
Irms = emon1.calcIrms(1480); // Calculate Irms only
// Irms -=0.18;
float b =Irms;
lcd.print(" I1= ");
lcd.print(b);
lcd.print(" A");
Serial.print(" I1 = ");
Serial.print(b);
Serial.println(" A");
        //////////////// // phase 2 voltage read //////////////////////////////////////
Vmax1=0;
Vmin1=1023;
For (int i2=0; i2<1000; i2++) {
    sensor1Value1=Vmax1;
    sensor1Value2=Vmin1;
    sensor1Value= analogRead(voltage1pin);
```

EK 2'in Devamı

```
if(sensor1Value>sensor1Value1) {
    Vmax1=sensor1Value;
    goto bailout1;
}
if (sensor1Value<sensor1Value2) {
    Vmin1=sensor1Value;
    goto bailout1;
}
Vmax1=sensor1Value1;
Vmin1=sensor1Value2;
bailout1:
delay(0.1);
}
float k1=(Vmax1-Vmin1);
V1=(0.00000412*k1*k1*k1-0.000857*k1*k1+2.675*k1-3.198);
int w=V1/(2*sqrt(2));
Serial.print("L2 voltage= ");
lcd.setCursor(0,2);
lcd.print("L2= ");
lcd.print(w);
lcd.print(" V");
Serial.print(w);
Serial.print(" V");
delay(2);
```

EK 2'in Devamı

```
////////// line 2 current read ////////////

Irms1 = emon2.calcIrms(1480); // Calculate Irms only

Irms1 -=0.18;

float d = Irms1;

Serial.print(" I2 = ");

Serial.print(d);

lcd.print(" I2= ");

lcd.print(d);

lcd.print(" A");

Serial.println(" A");

////////// phase 3 voltage read ////////////

Vmax2=0;

Vmin2=1023;

For (int i3=0; i3<1000; i3++) {

  sensor2Value1=Vmax2;

  sensor2Value2=Vmin2;

  sensor2Value= analogRead(voltage2pin);

  if(sensor2Value>sensor2Value1) {

    Vmax2=sensor2Value;

    goto bailout2;

  }

  if (sensor2Value<sensor2Value2) {

    Vmin2=sensor2Value;
```

EK 2'in Devamı

```
goto bailout2;
}
Vmax2=sensor2Value1;
Vmin2=sensor2Value2;
bailout2:
delay(0.1);
}
float k2=(Vmax2-Vmin2);
V2=(0.00000412*k2*k2*k2-0.000857*k2*k2+2.675*k2-3.198);
int e=V2/(2*sqrt(2));
lcd.setCursor(0,3);
lcd.print("L3= ");
lcd.print(e);
lcd.print(" V");
Serial.print("L3 voltage= ");
Serial.print(e);
Serial.print(" V");
delay(2);
//////// // line 3 current read////////
Irms2 = emon3.calcIrms(1480); // Calculate Irms only
Irms2 -=0.18;
float f = Irms2;
Serial.print(" I3= ");
```

EK 2'in Devamı

```
Serial.print(f);
Serial.println(" A ");
lcd.print(" I3= ");
lcd.print(f);
lcd.print(" A");
// Serial.println(q);
String x= data_address;
char l= x.charAt(0);
Serial.println(l);
If (l=='c')
{
digitalWrite(5,HIGH);
digitalWrite(6,HIGH);
o="u";
}
else if(l=='d')
{
digitalWrite(5,LOW);
digitalWrite(6,LOW);
o="M";
}
delay(15);
ok=false;
String
j=String(a)+r+String(b)+r+String(w)+r+String(d)+r+String(e)+r+String(f)+r ;
```

EK 2'in Devamı

```
String c = address+String(j);  
char send_to_master[c.length() + 1];  
c.toCharArray(send_to_master, c.length() + 1);  
Serial1.write(send_to_master, sizeof(send_to_master)); //sending the data via  
xbee  
}  
}
```



EK 3

Düğüm II Düşük Seviye Kodu

```
String address = "ADDRESS2";

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);

////////// for RPM //////////////////////////////////////

const int numsamples = 10;

int samples[numsamples];

unsigned long average = 0;

int index = 0;

unsigned long total;

volatile int rpmcount = 0;//see http://arduino.cc/en/Reference/Volatile

unsigned long rpm = 0 , a=0 ;

float b=0;

unsigned long lastmillis = 0;

////////// for XBee //////////////////////////////////////

String z,q="A";

String r="/" ,s="u" ,o="u" ,t="u";

char re_address[20],v;

String receive_address="";

String data_address="";

boolean ok = false;

void motor()
```

EK 3'in Devamı

```
{ /* this code will be executed every time the interrupt 4 (pin7) gets low.*/  
  rpmcount++;  
}  
  
void tranceiver_data()  
{  
  int i = 0;  
  receive_address = "";  
  data_address="";  
  if (Serial1.available() > 0)  
  {  
    while (Serial1.available() > 0)  
    {  
      re_address[i] = Serial1.read();  
      delay(1);  
      if(i<=7) receive_address += String(re_address[i]);  
      else if(i<=10) data_address+=String(re_address[i]);  
      i++;  
    }  
    Serial1.flush();  
  }  
  if(receive_address==address)  
  ok=true;  
}
```

EK 3'in Devamı

```
void setup()
{
  lcd.init(); // initialize the lcd
  lcd.backlight();
  lcd.setCursor(2,0);
  lcd.print("RPM & TEMP. ");
  lcd.setCursor(3,1);
  lcd.print("MONITORING ");
  pinMode(6,OUTPUT);
  digitalWrite(6,HIGH);
  Serial.begin(9600);
  Serial1.begin(9600);
  attachInterrupt(4, motor, FALLING);
}

void loop()
{
  if (!ok)
    trancever_data();
  else
  {
    //////////////////////////////////// Reading Proximity Sensor ////////////////////////////////////

    hhh: if (millis() - lastmillis >= 1000){ /*Update every one second, this will be
equal to reading frequency (Hz).*/
```

EK 3'in Devamı

```
detachInterrupt(0); //Disable interrupt when calculating total = 0;

samples[index] = rpmcount * 60; /* Convert frequency to RPM, note: this works for
one interruption per full rotation. For two interrupts per full rotation use rpmcount *
30.*/

for (int i=0; i<=9; i++){

    total = total + samples[i];

}

average = total / numsamples;

rpm = average;

rpmcount = 0; // Restart the RPM counter

index++;

if(index >= numsamples){

    index=0;

    goto hhh;

}

if (millis() > 11000){ // wait for RPMs average to get stable

Serial.print(" RPM = ");

Serial.println(rpm);

lcd.clear();

lcd.setCursor(2,0);

lcd.print("RPM = ");

lcd.print(rpm);

a=rpm;

}

lastmillis = millis(); // Update lastmillis
```

EK 3'in Devamı

```
attachInterrupt(4, motor, FALLING); //enable interrupt

}

//////////////////////////////////// Reading temperature sensor////////////////////////////////////

int Sin = analogRead(A0);

float Vm = ( Sin/1024.0)*5000;

float b = Vm/10;

lcd.setCursor(2,1);

lcd.print("Temp = ");

lcd.print(b);

lcd.print(" C");

// Serial.println(q);

String x= data_address;

char l= x.charAt(0);

Serial.println(l);

if(l=='c')

{

digitalWrite(6,HIGH);

o="u";

}

else if(l=='d')

{

digitalWrite(6,LOW);

o="M";

}
```

EK 3'in Devamı

```
delay(15);
```

```
ok=false;
```

```
String  
j=String(a)+r+String(b)+r+String(0)+r+String(0.00)+r+String(0)+r+String(0.00)+r ;
```

```
String c = address+String(j);
```

```
char send_to_master[c.length() + 1];
```

```
c.toCharArray(send_to_master, c.length() + 1) ;
```

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

```
}}
```

EK 4

Ana Düğüm Düşük Seviye Kodu

```
String address = "ADDRESS";

char re_address[28];

#define number 3

String address_receive[number],data_receive[number];

String a="0/0/0/d/u/u/u/", b="0/0/0/d/d/u/u/u/", x="0/0/0/d/d/u/u/u/",
y="0/0/0/d/d/u/u/u/", w="0/0/0/d/d/u/u/u/0/0/0/d/d/u/u/u/B/B";

//String r="?",r1="!",r2="&";

boolean slave_state[number];

int selecktor,
slave_counter,slaveError[number],slaveError2[number],slaveError3[number];

char d='r',v;

char s1='A', s2='B';

int m=0;

int maximum=30;

void trancever_data(int time_to_delay, int number_of_slave)
{
    int i = 0;

    volatile long delay_data_time = millis();

    address_receive[slave_counter] = "";

    data_receive[slave_counter] = "";

    while (millis() - delay_data_time <= time_to_delay)
    {
```

EK 4'in Devamı

```
if (Serial1.available() > 0)
{
  while (Serial1.available() > 0 && i <= 28)
  {
    re_address[i] = Serial1.read();
    delay(1);
    if (i<=7)address_receive[slave_counter] += String(re_address[i]);
    else if (i<=28)data_receive[slave_counter] += String(re_address[i]);
    i++;
  }
}
Serial1.flush();
}
}

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

void loop()
{
  if (d=='+')
  maximum++;
  else if(d=='-')
```


EK 4'in Devamı

```
maximum--;
for (slave_counter = 0; slave_counter < number; slave_counter++)
{
    if(Serial.available()>0)
    d=Serial.read();
    if(d!='r')
    m++;
    if(m>=6)
    {d='r';
    m=0;}
    String c = address + String(slave_counter + 1)+String(d);
    char send_to_slave[c.length() + 1];
    c.toCharArray(send_to_slave, c.length() + 1) ;
    Serial1.write(send_to_slave, sizeof(send_to_slave));
    delay(50);
    trancever_data(400, slave_counter);
    if(address_receive[slave_counter]==address+String(slave_counter+1))
    {
        slaveError[slave_counter]=0;
    }
    if(address_receive[slave_counter]== "")
    {
        slaveError[slave_counter]+=1;
        slaveError[slave_counter]=constrain(slaveError[slave_counter],0,6);
    }
}
```

EK 4'in Devamı

```
if (slaveError[0]<=1)
s1='A';
else
s1='B';
if (slaveError[1]<=1)
s2='A';
else
s2='B';
// Serial.println(s1,s2);
//      Serial.println(address_receive[slave_counter]+slaveError[slave_counter]);
//          if (address_receive[slave_counter] == address + String(1))
//              {
x=data_receive[slave_counter];
delay(500);
}
//          else if (address_receive[slave_counter] == address + String(2))
//              {
y=data_receive[slave_counter];
//Serial.println(y);
}
//if (s1=='B')
//a="0/0/0/d/d/d/d/";
//else
a=x;
```

EK 4'in Devamı

```
//if (s2=='B')  
//b="0/0/0/d/d/d/d/";  
//else  
b=y;  
String r="";  
w=a+b+s1+r+s2+r;  
Serial.println(w+maximum+r);  
//Serial.println(w);  
}  
}
```

EK 5

Ana PC İçin Yüksek Seviye Visual Studio Kodu

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

Public Class Motor
    Dim s As String
    Dim mydata As String = ""
    Dim stat As String = ""
    Dim value1 As String
    Dim value2 As String
    Dim value3 As String
    Dim value4 As String
    Dim value5 As String
    Dim value6 As String
    Dim value7 As String
    Dim value8 As String
    Dim counter As Integer
    Dim a, b, c, d, f, j, w, r As Decimal
    Dim myPort As Array 'COM Ports detected on the system will be stored here
    Dim jlast As Array

    Private Sub btnConnect_Click(sender As Object, e As EventArgs) Handles
        btnConnect.Click
            SerialPort1.PortName = cmbPort.Text 'Set SerialPort1 to the
            selected COM port at startup
            SerialPort1.BaudRate = cmbBaud.Text 'Set Baud rate to the
            selected value on

            'Other Serial Port Property
            SerialPort1.Parity = IO.Ports.Parity.None
            SerialPort1.StopBits = IO.Ports.StopBits.One
            SerialPort1.DataBits = 8 'Open our serial port
            SerialPort1.Open()

            btnConnect.Enabled = False 'Disable Connect button
            btnDisconnect.Enabled = True 'and Enable Disconnect button

    End Sub

    Private Sub Motor_Load(sender As Object, e As EventArgs) Handles
        MyBase.Load
            'TODO: This line of code loads data into the 'MotordbDataSet.Table1'
            table. You can move, or remove it, as needed.
            Me.Table1TableAdapter.Fill(Me.MotordbDataSet.Table1)
```

EK 5'in Devamı

```
        myPort = IO.Ports.SerialPort.GetPortNames() 'Get all com ports
available
        cmbBaud.Items.Add(9600)      'Populate the cmbBaud Combo box to common
baud rates used
        cmbBaud.Items.Add(19200)
        cmbBaud.Items.Add(38400)
        cmbBaud.Items.Add(57600)
        cmbBaud.Items.Add(115200)

        For i = 0 To UBound(myPort)
            cmbPort.Items.Add(myPort(i))
        Next
        cmbPort.Text = cmbPort.Items.Item(0) 'Set cmbPort text to the first
COM port detected
        cmbBaud.Text = cmbBaud.Items.Item(0) 'Set cmbBaud text to the first
Baud rate on the list

        btnDisconnect.Enabled = False      'Initially Disconnect Button
is Disabled
    End Sub

    Private Sub cmbPort_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cmbPort.SelectedIndexChanged
        If SerialPort1.IsOpen = False Then
            SerialPort1.PortName = cmbPort.Text      'pop a message box to
user if he is changing ports
        Else
            'without disconnecting first.
            MsgBox("Valid only if port is Closed", vbCritical)
        End If
    End Sub

    Private Sub cmbBaud_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cmbBaud.SelectedIndexChanged
        If SerialPort1.IsOpen = False Then
            SerialPort1.BaudRate = cmbBaud.Text      'pop a message box to user
if he is changing baud rate

        Else
            'without disconnecting first.
            MsgBox("Valid only if port is Closed", vbCritical)
        End If
    End Sub

    Private Sub btnDisconnect_Click(sender As Object, e As EventArgs) Handles
btnDisconnect.Click
        Try
            SerialPort1.DiscardInBuffer()
            SerialPort1.DiscardOutBuffer()
            SerialPort1.Close()
            btnConnect.Enabled = True
            btnDisconnect.Enabled = False

        Catch ex As Exception

        End Try
    End Sub
```

EK 5'in Devamı

```
Private Sub datareceived(ByVal sender As Object, ByVal e As
SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

    Try

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

        End If

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

End Sub

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

    TextBox3.Text = Today
    clk.Text = TimeOfDay

    If (SerialPort1.IsOpen()) Then
        'If (mtron.Enabled = False) And (mtroff.Enabled = True) Then
        s = mydata + "/" + "/" + "/" + "/" + "/" + "/" + "/"
        Table1BindingSource.AddNew()
        Dim somestring() As String
        Dim Remote As Integer
        'Dim jso3 As String
        Dim VH As Integer = 0
        Dim VL As Integer = 0
        Dim IH As Integer = 0
        Dim IL As Integer = 0
        Dim TH As Integer = 0
        Dim TL As Integer = 0
        Dim ML As Integer = 0
        Dim Temp0 As Decimal
        somestring = s.Split(New Char())
        value1 = somestring(0)
        Temp0 = Val(value1)
        ph1v.Text = Temp0
        a = ph1v.Text
        If ph1v.Text > 225 Then
            Label11.Text = "H"
            VH = VH + 1
        ElseIf ph1v.Text < 217 And ph1v.Text > 0 Then
            Label11.Text = "L"
            VL = VL + 1
        ElseIf ph1v.Text = 0 Then
            Label11.Text = "D"
        Else
            Label11.Text = "N"
        End If
        value2 = somestring(1)
```

EK 5'in Devamı

```
ph1C.Text = Val(value2)
b = ph1C.Text
If ph1C.Text > 2.0 Then
    Label12.Text = "H"
    IH = IH + 1
ElseIf ph1C.Text < 0.2 And ph1C.Text > 0.0 Then
    Label12.Text = "L"
    IL = IL + 1
ElseIf ph1C.Text = 0.0 Then
    Label12.Text = "D"
Else
    Label12.Text = "N"
End If
value3 = somestring(2)
ph2v.Text = Val(value3)
c = ph2v.Text
If ph2v.Text > 225 Then
    Label13.Text = "H"
    VH = VH + 1
ElseIf ph2v.Text < 217 And ph2v.Text > 0 Then
    Label13.Text = "L"
    VL = VL + 1
ElseIf ph2v.Text = 0 Then
    Label13.Text = "D"
Else
    Label13.Text = "N"
End If
value4 = somestring(3)
ph2C.Text = Val(value4)
d = ph2C.Text
If ph2C.Text > 2.0 Then
    Label14.Text = "H"
    IH = IH + 1
ElseIf ph2C.Text < 0.2 And ph2C.Text > 0 Then
    Label14.Text = "L"
    IL = IL + 1
ElseIf ph2C.Text = 0.0 Then
    Label14.Text = "D"
Else
    Label14.Text = "N"
End If
value5 = somestring(4)
ph3v.Text = Val(value5)
f = ph3v.Text
If ph3v.Text > 225 Then
    Label15.Text = "H"
    VH = VH + 1
ElseIf ph3v.Text < 215 And ph3v.Text > 0 Then
    Label15.Text = "L"
    VL = VL + 1
ElseIf ph3v.Text = 0 Then
    Label15.Text = "D"
Else
    Label15.Text = "N"
End If
value6 = somestring(5)
ph3C.Text = Val(value6)
j = ph3C.Text
If ph3C.Text > 2.0 Then
```

EK 5'in Devamı

```
Label16.Text = "H"
    IH = IH + 1
ElseIf ph3C.Text < 0.2 And ph3C.Text > 0 Then
    Label16.Text = "L"
    IL = IL + 1
ElseIf ph3C.Text = 0.0 Then
    Label16.Text = "D"
Else
    Label16.Text = "N"
End If
value8 = somestring(6)
RPM.Text = Val(value8)
w = RPM.Text
If RPM.Text < 1000 And RPM.Text > 0 Then
    Label19.Text = "L"
    ML = ML + 1
ElseIf RPM.Text = 0 Then
    Label19.Text = "D"
Else
    Label19.Text = "N"
End If
value7 = somestring(7)
Temp.Text = Val(value7)
r = Temp.Text
If Temp.Text < 38.0 Then
    Label17.Text = "N"
ElseIf Temp.Text > 40 Then
    Label17.Text = "H"
    TH = TH + 1
ElseIf Temp.Text = 0 Then
    Label17.Text = "D"
End If

'=====
===== '////////// Publishing Data to thingspeak.com
//////////

'creat a Request using a URL that can Receive a post.
Dim Request As WebRequest =
WebRequest.Create("https://api.thingspeak.com/update?api_key=U5A39EG4SP4MTB46&
field1=" & a & "&field2=" & b & "&field3=" & c & "&field4=" & d & "&field5=" &
f & "&field6=" & j & "&field7=" & r & "&field8=" & w & "")
'set the method property of the request to post
Request.Method = "POST"
'creat post data and convert it to byte Array
Dim postdata As String = "55"
Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)
'set the content type property of the webrequest
Request.ContentType = "application/x-www-form-urlencoded"
'set the content length property of webrequest
Request.ContentLength = byteArray.Length
'get the request stream
Dim datastream As Stream = Request.GetRequestStream()
' write the data to the request stream
datastream.Write(byteArray, 0, byteArray.Length)
'close the stream object
datastream.Close()
```


EK 5'in Devamı

```
'get the response
Dim response As WebResponse = Request.GetResponse()
'display the stastuse
Console.WriteLine(CType(response,
HttpWebResponse).StatusDescription)
'get the stream containing content returned by the server
datastream = response.GetResponseStream()
' open the stream using stream read for easy access
Dim reader As New StreamReader(datastream)
'read the content
Dim responsefromserver As String = reader.ReadToEnd()
'display the content

Console.WriteLine(responsefromserver)
'clean up the streams
reader.Close()
datastream.Close()
response.Close()

'=====
=====

'////////////////////// Readding Remote Comands from Cloud
////////////////////////////////////

Dim webClient As WebClient = New WebClient()
'Read only the last data Entered to filed 2 of the motor status
channel
Dim data =
webClient.DownloadString("https://api.thingspeak.com/channels/718912/fields/fi
eld2/last")
'convert string to number
Remote = Val(data)
If Remote = 44 Then
    If mtron.Enabled = False And mtroff.Enabled = True Then
        SerialPort1.Write("c")
        mtron.Enabled = True           'enable motor ON button
        mtroff.Enabled = False        ' disable motor OFF button
        MessageBox.Show("Motor is stopped from Remote Device",
"Attention")

    End If
ElseIf Remote = 99 Then
    If mtron.Enabled = True And mtroff.Enabled = False Then
        SerialPort1.Write("d")
        mtron.Enabled = False         'disable motor ON button
        mtroff.Enabled = True         ' enable motor OFF button
        MessageBox.Show("The Motor is startting to Work from remot
device", "Attention")
    End If
End If

'=====
=====
'////////////////////// Testing the motor Parameters
////////////////////////////////////
If VH > 0 Then
    If mtron.Enabled = False And mtroff.Enabled = True Then
' if motor ON button is disabled and motor OFF is enabled
        SerialPort1.Write("c")
```

EK 5'in Devamı

```
        mtron.Enabled = True           'enable Connect button
        mtroff.Enabled = False
        mtrstats.Text = "Motor stopped due to High Voltage"
    End If
    ElseIf VL > 0 Then
        If mtron.Enabled = False And mtroff.Enabled = True Then
'enable Connect button
            SerialPort1.Write("c")
            mtron.Enabled = True       'enable Connect button
            mtroff.Enabled = False
            mtrstats.Text = "Motor stopped due to Low Voltage"
        End If
    ElseIf IH > 0 Then
        If mtron.Enabled = False And mtroff.Enabled = True Then
'enable Connect button
            SerialPort1.Write("c")
            mtron.Enabled = True       'enable Connect button
            mtroff.Enabled = False
            mtrstats.Text = "Motor stopped due to High Current"
        End If
    ElseIf IL > 0 Then
        If mtron.Enabled = False And mtroff.Enabled = True Then
'enable Connect button
            SerialPort1.Write("c")
            mtron.Enabled = True       'enable Connect button
            mtroff.Enabled = False
            mtrstats.Text = "Motor stopped due to Low Current"
        End If
    ElseIf TH > 0 Then
        If mtron.Enabled = False And mtroff.Enabled = True Then
'enable Connect button
            SerialPort1.Write("c")
            mtron.Enabled = True       'enable Connect button
            mtroff.Enabled = False
            mtrstats.Text = "Motor stopped due to High Temperature"
        End If
    ElseIf ML > 0 Then
        mtrstats.Text = "The Motor working in low speed"
    Else
        mtrstats.Text = "The Motor is Working Stable"
    End If

End If

Table1BindingSource.EndEdit()
Me.Table1TableAdapter.Update(Me.MotordbDataSet.Table1)

End Sub

Private Sub mtron_Click(sender As Object, e As EventArgs) Handles
mtron.Click
    If (SerialPort1.IsOpen()) Then
        SerialPort1.Write("d")
        mtron.Enabled = False         'enable Connect button
        mtroff.Enabled = True
        mtrstats.Text = "The Motor is startting to Work manually"
    End If
End Sub
```

EK 5'in Devamı

```
Private Sub mtroff_Click(sender As Object, e As EventArgs) Handles
mtroff.Click

    If (SerialPort1.IsOpen()) Then
        SerialPort1.Write("c")
        mtron.Enabled = True           'enable Connect button
        mtroff.Enabled = False
        mtrstats.Text = "The Motor stopped Manually"

    End If
End Sub

Private Sub Label14_Click(sender As Object, e As EventArgs) Handles
Label14.Click

    End Sub

Private Sub DataGridView1_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs)

    End Sub

Private Sub DataGridView1_CellContentClick_1(sender As Object, e As
DataGridViewCellEventArgs)

    End Sub

Private Sub DataGridView1_CellContentClick_2(sender As Object, e As
DataGridViewCellEventArgs)

    End Sub

End Class
```

EK 6

İş İstasyonu İçin Yüksek Seviye Visual Studio Kodu

```
Imports System.IO.Ports
Imports System.Text
Imports System.Net
Imports System.IO
Public Class Form1
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

        End Sub

        Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
            Dim webClient As WebClient = New WebClient()
            Dim data =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld1/last")
            Ph1V.Text = data
            Dim data1 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld2/last")
            Ph1C.Text = data1
            Dim data2 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld3/last")
            Ph2V.Text = data2
            Dim data3 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld4/last")
            Ph2C.Text = data3
            Dim data4 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld5/last")
            Ph3V.Text = data4
            Dim data5 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld6/last")
            Ph3C.Text = data5
            Dim data6 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld7/last")
            Temp.Text = data6
            Dim data7 =
webClient.DownloadString("https://api.thingspeak.com/channels/662590/fields/fi
eld8/last")
            RPM.Text = data7
            Dim data8 =
webClient.DownloadString("https://api.thingspeak.com/channels/718912/fields/fi
eld2/last")
            Dim Remote As Decimal = Val(data8)
            'If Remote = 0 Then
            'mtron.Enabled = True           'enable Connect button
            'mtroff.Enabled = False
            ' ElseIf Remote = 1 Then
```

EK 6'in Devamı

```
'mtron.Enabled = False
'mtroff.Enabled = True
' End If

End Sub

Private Sub mtron_Click(sender As Object, e As EventArgs) Handles
mtron.Click
    Dim Request As WebRequest =
WebRequest.Create("https://api.thingspeak.com/update?api_key=7FEV74TX45K9E21J&
field2=1")
    'set the method property of the request to post
Request.Method = "POST"
    'creat post data And convert it to byte Array
Dim postdata As String = "55"
Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)
    'set the content type property of the webrequest
Request.ContentType = "application/x-www-form-urlencoded"
    'set the content length property of webrequest
Request.ContentLength = byteArray.Length
    'get the request stream
Dim datastream As Stream = Request.GetRequestStream()
    ' write the data to the request stream
datastream.Write(byteArray, 0, byteArray.Length)
    'close the stream object
datastream.Close()
    'get the response
Dim response As WebResponse = Request.GetResponse()
    'display the status
Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
    'get the stream containing content returned by the server
datastream = response.GetResponseStream()
    ' open the stream using stream read for easy access
Dim reader As New StreamReader(datastream)
    'read the content
Dim responsefromserver As String = reader.ReadToEnd()
    'display the content

Console.WriteLine(responsefromserver)
    'clean up the streams
reader.Close()
datastream.Close()
response.Close()

End Sub

Private Sub mtroff_Click(sender As Object, e As EventArgs) Handles
mtroff.Click
    Dim Request As WebRequest =
WebRequest.Create("https://api.thingspeak.com/update?api_key=7FEV74TX45K9E21J&
field2=0")
    'set the method property of the request to post
Request.Method = "POST"
    'creat post data And convert it to byte Array
Dim postdata As String = "55"
Dim byteArray As Byte() = Encoding.UTF8.GetBytes(postdata)
    'set the content type property of the webrequest
Request.ContentType = "application/x-www-form-urlencoded"
    'set the content length property of webrequest
Request.ContentLength = byteArray.Length
```

EK 6'in Devamı

```
'get the request stream

Dim datastream As Stream = Request.GetRequestStream()
' write the data to the request stream
datastream.Write(byteArray, 0, byteArray.Length)
'close the stream object
datastream.Close()
'get the response
Dim response As WebResponse = Request.GetResponse()
'display the status
Console.WriteLine(CType(response, HttpWebResponse).StatusDescription)
'get the stream containing content returned by the server
datastream = response.GetResponseStream()
' open the stream using stream read for easy access
Dim reader As New StreamReader(datastream)
'read the content
Dim responsefromserver As String = reader.ReadToEnd()
'display the content

Console.WriteLine(responsefromserver)
'clean up the streams
reader.Close()
datastream.Close()
response.Close()

End Sub
End Class
```

ÖZGEÇMİŞ

Adı Soyadı : Ali Husein BENHUSEIN
Doğum Yeri ve Yılı : 04.12.1971, Trablus, Libya
Medeni hal : Evli
Yabancı Dili : İngilizce
E-posta : ALI_HUSSAIN_M@YAHOO.COM



Eğitim Durumu

Lise : Hay-demashek – Trablus /Libya (1987-1990)
Lisans : Bilgisayar Teknolojisinin Kolajı - Trablus / Libya
(1991-1995)
Yüksek Lisans : Trablus Üniversitesi- Trablus / Libya (2006 – 2009)

Mesleki Deneyim

Elektronik Teknolojinin Trablus Koleji 1997- Şimdiki

Yayımları

[1] Ali Husein BENHUSEIN, M. Fatih KILIÇASLAN, (2019), “Hardware Implementation of a Real Time Monitoring and Controlling System for Three-Phase Induction Motor Using ZigBee and IOT”, *International Journal of Science and Research (IJSR)*, 2019.

[2] Ali Husein BENHUSEIN, M. Fatih KILIÇASLAN, (2019), “Software Implementation of A Real-Time Monitoring and Controlling system for Three-Phase Induction Motor Using ZigBee and IOT”, *International Journal of Research in Engineering & Technology*, 2019.