

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**KABLOSUZ ALGILAYICI/EYLEYİCİ AĞLAR İLE WEB
TABANLI ORTAM GÖZLEME VE DENETİM UYGULAMASI**

YÜKSEK LİSANS TEZİ

Önder YAKUT

Anabilim Dalı: Elektronik ve Bilgisayar Eğitimi

Danışman: Doç. Dr. Celal ÇEKEN

KOCAELİ, 2012

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**KABLOSUZ ALGILAYICI/EYLEYİCİ AĞLAR İLE WEB
TABANLI ORTAM GÖZLEME VE DENETİM UYGULAMASI**

YÜKSEK LİSANS TEZİ

Önder YAKUT

Tezin Enstitüye Verildiği Tarih: 13 MART 2012

Tezin Savunulduğu Tarih: 29 MART 2012

**Tez Danışmanı
Doç.Dr. Celal ÇEKEN**


(.....)

**Üye
Doç.Dr. Mehmet YILDIRIM**


(.....)

**Üye
Yrd.Doç.Dr. Ali ÇALHAN**


(.....)

KOCAELİ, 2012

ÖNSÖZ VE TEŞEKKÜR

İletişim teknolojisinin gelişmesi ile birlikte küçük boyutlu ve çok işlevli algılayıcı düğümler (AD) geliştirilmiştir. AD'ler bir araya gelerek oluşturduğu yapıya kablosuz algılayıcı ağlar (KAA) denilmektedir. Bu tez çalışmasında AD'lerden gelen veriler bilgisayar ortamına alınarak veri tabanına kaydedilmiştir. Bu veriler web tabanlı olarak internet üzerinden yayınlanmıştır. Oluşturulan arayüz ile web tabanlı uzaktan denetim uygulaması gerçekleştirilmiştir. Bu çalışma ile farklı uygulamaların geliştirilmesine yönelik bir altyapı oluşturulmuştur.

Yüksek lisans eğitimim süresince değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarına yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygıdeğer hocam Doç. Dr. Celal ÇEKEN'e, tez çalışmasının başından sonuna kadar manevi desteklerini benden esirgemeyen arkadaşlarım; sayın Arş. Gör. Faruk AKTAŞ'a, sayın Uzman Bilal TÜRK'e ve sayın Tekniker Mustafa PEKER'e teşekkürlerimi sunarım.

Bugünlere gelmemi sağlayan anneme, babama ve kardeşlerime saygı, sevgi ve sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER.....	ii
ŞEKİLLER DİZİNİ	iv
SİMGELER VE KISALTMALAR	v
ÖZET	vii
ABSTRACT	viii
1. GİRİŞ	1
1.1. Literatür Taraması.....	2
1.2. Tez Çalışmasının Amacı	3
1.3. Tez Düzeni.....	4
2. KABLOSUZ ALGILAYICI AĞLAR	5
2.1. Giriş	5
2.2. Kablosuz Ortamdan Kaynaklanan Sınırlamalar	6
2.3. Kablosuz Algılayıcı Ağların Avantajları.....	7
2.4. Kablosuz Algılayıcı Ağların Dezavantajları.....	7
2.5. Kablosuz Algılayıcı Ağlar	8
2.6. Kablosuz Algılayıcı Ağ Düğümleri	9
2.7. Kablosuz Algılayıcı ve Eyleyici Ağlar.....	10
2.8. Sonuç	11
3. UYGULAMADA KULLANILAN PROGRAMLAR.....	12
3.1. PHP	12
3.2. jQuery	12
3.3. AJAX.....	13
3.4. WAMP SERVER.....	13
3.5. PostgreSQL Veri Tabanı	14
3.6. PERL	14
3.7. XML Ve XML – RPC.....	15
3.8. TinyOS İşletim Sistemi	16
3.9. nesC Programlama Dili	16
3.10. Sonuç	17
4. KABLOSUZ AĞ MİMARİLERİ VE KULLANILAN CİHAZLAR.....	18
4.1. Örgü (Mesh) Ağının Temelleri.....	18
4.2. Topolojiler	19
4.2.1. Yıldız (Star) Topolojisi.....	20
4.2.2. Örgü (Mesh) Topolojisi.....	21
4.2.3. Yıldız – Örgü (Star - Mesh) Melez Topolojisi	22
4.3. Xmesh.....	23
4.4. Uygulamada Kullanılan Algılayıcı Düğümler ve Erişim Noktası	24
4.5. Xserve.....	25
4.6. Tez Çalışması İçin Kullanılan Xserve Parametreleri	28
4.7. Sonuç	29
5. GELİŞTİRİLEN WEB TABANLI ARAYÜZ TASARIMI.....	30

5.1. Perl Soket Programı	32
5.2. PostgreSQL “wsn” Veritabanı	34
5.3. Geliştirilen Web Tabanlı Arayüzün Tanıtımı	35
5.4. Web Tabanlı Arayüz İle Uzaktan Denetim Uygulaması.....	43
5.5. Sonuç	48
SONUÇLAR VE ÖNERİLER	49
KAYNAKLAR.....	51
EKLER.....	53
ÖZGEÇMİŞ.....	63

ŞEKİLLER DİZİNİ

Şekil 2.1: Kablosuz algılayıcı ağların genel kullanım alanları.....	6
Şekil 2.2: Örnek bir kablosuz algılayıcı ağ uygulaması.....	9
Şekil 2.3 : Bir algılayıcı düğümün yapısı ve mimarisi.....	9
Şekil 2.4 : KAEA fiziksel mimarisi	10
Şekil 2.5 : a) Algılayıcı b) Eyleyici bileşenleri	11
Şekil 4.1 : Yıldız (star) topolojisi	21
Şekil 4.2 : Örgü (mesh) topolojisi.....	22
Şekil 4.3 : Yıldız-örgü (star-mesh) melez topolojisi.....	23
Şekil 4.4 : Xmesh ağı	24
Şekil 4.5 : MicaZ algılayıcı düğümü	25
Şekil 4.6 : MIB520 programlama bordu (erişim noktası) üstten görünüşü	25
Şekil 4.7 : Xserve’de verilerin ham biçimi	26
Şekil 4.8 : Xserve’de verilerin ayrıştırılmış biçimi	27
Şekil 4.9 : Xserve’de verilerin dönüştürülmüş biçimi	27
Şekil 4.10 : Xserve ekran çıktısı.....	29
Şekil 5.1 : Sistemin blok şeması.....	30
Şekil 5.2 : Web tabanlı ortam gözleme uygulaması için kurulan sistem	30
Şekil 5.3 : AD’den gelen XML formatındaki örnek veri	33
Şekil 5.4 : Perl’de oluşturulan soket programının konsol çıktısı.....	34
Şekil 5.5 : “wsn” veri tabanı yapısı.....	35
Şekil 5.6 : Web arayüzünün ana sayfasının görünümü	36
Şekil 5.7 : Web arayüzün grafik çizdirme ekranı	38
Şekil 5.8 : İki AD’ye ait y eksenli ivme grafiğinin zamana göre değişimi (bar)	39
Şekil 5.9 : İki AD’ye ait basınç grafiğinin zamana göre değişimi (line)	39
Şekil 5.10 : İki AD’ye ait batarya grafiğinin zamana göre değişimi (spline)	40
Şekil 5.11 : İki AD’ye ait nem sıcaklığının zaman göre değişimi (scatter)	40
Şekil 5.12 : İki AD’ye ait x eksenli ivme grafiği zamana göre değişimi (area line) ...	41
Şekil 5.13 : İki AD’ye ait ışık grafiğinin zamana göre değişimi (area spline).....	41
Şekil 5.14 : Web arayüzü KAA topoloji ekranı (doğrudan iletişim)	42
Şekil 5.15 : Web arayüzü KAA topoloji ekranı (erişim noktalı iletişim)	42
Şekil 5.16 : Denetleme sistemi için kurulan düzenek	43
Şekil 5.17 : Denetleme sisteminin zaman akış diyagramı	44
Şekil 5.18 : Referans sıcaklığı gönderme ekranı	44
Şekil 5.19 : Xserve’de MDA320 veri edinim bordudan gelen veriler	45
Şekil 5.20 : Perl’de oluşturulan soket programının ekran çıktısı	46
Şekil 5.21 : “mda320result” tablosunun veri tabanı yapısı	46
Şekil 5.22 : Sistemin sıcaklık bilgisi.....	47
Şekil 5.23 : Sistemin denetim bilgisi.....	47

SİMGELER VE KISALTMALAR

Simgeler

KB	:Kilo Bayt
°C	:Santigrat derece
COM	:Seri haberleşme arayüzü
GHz	:Gigahertz
Kbps	:Saniyede iletilen kilobit sayısı
lux	:Lüks
mbar	:Milibar
Mbps	:Saniyede iletilen megabit sayısı
mg	:Miligal, Miligrav
MHz	:Megahertz
mV	:Milivolt

Kısaltmalar

AJAX	:Asynchronous JavaScript And XML (Eş zamanlı Olmayan Javascript ve XML)
ANSI	:American National Standard Institute
AP	:Access Point (Erişim Noktası)
ASP	:Active Server Pages (Etkin Sunucu Sayfaları)
DBMS	:Database Management Systems (Veri tabanı Yönetim Sistemi - VTYS)
EEPROM	:Electrically Erasable Programmable Read Only Memory (Elektriksel Olarak Yazılıp Silinebilen Sadece Okunabilir Bellek)
GPL	:General Public License (Genel Kamu Lisansı)
HTML	:Hypertext Markup Language
HTTP	:Hypertext Transfer Protocol
ID	:Identity (Kimlik Bilgisi)
IEEE	:Institute of Electrical and Electronics Engineers
MD	:Merkezi Düğüm
MEMS	:Micro-Elektro Mechanical Systems (Mikro Elektro Mekanik Sistemleri)
MIT	:Massachusetts Institute of Technology
nesC	:Network Embedded Systems C (Gömülü Ağ Sistemleri İçin C)
OTAP	:Over The Air Programming (Hava Üzerinden Programlama)
PERL	:Practical Extraction and Report Language (Pratik Çıkarım ve Raporlama Dili)
PHP	:Hypertext Preprocessor
RF	:Radio Frequency (Radyo Frekansı)
SN	:Sensor Node (Algılayıcı Düğüm - AD)
SQL	:Structured Query Language (Yapısal Sorgulama Dili)
WSN	:Wireless Sensor Networks (Kablosuz Algılayıcı Ağlar – KAA)

WSAN :Wireless Sensor and Actuator Networks (Kablosuz Algılayıcı ve Eyleyici Ağlar – KAEA)
XML :Extensible Markup Language (Geniřleyebilir İřaretleme Dili)
XML-RPC :Extensible Markup Language-Remote Procedure Call (Geniřleyebilir İřaretleme Dili – Uzaktan Yordam Çaęırma)

KABLOSUZ ALGILAYICI/EYLEYİCİ AĞLAR İLE WEB TABANLI ORTAM GÖZLEME VE DENETİM UYGULAMASI

Önder YAKUT

Anahtar Kelimeler: Kablosuz Algılayıcı/Eyleyici Ağlar, Web Tabanlı Ortam Gözleme, Ağ Tabanlı Denetim

Özet: Kablolı ağ teknolojilerine göre üstün oluşlarından dolayı önemi her geçen gün artan kablosuz algılayıcı ağ (KAA) sistemlerinin kullanımı giderek daha yaygın hale gelmektedir. İlk zamanlarda yoğun olarak ortam gözleme ve izleme amaçlı olarak kullanılan KAA sistemleri günümüzde denetim amaçlı olarak da uygulama alanı bulmaktadır. KAA'lar; askeri, sağlık, tarımsal, endüstriyel ve çevresel alanlarda kullanılmaktadır.

Tez çalışması kapsamında, kablosuz algılayıcı düğümlerden gelen verileri uzaktan görüntüleyebilmek amacıyla web tabanlı bir arayüz geliştirilmiştir. Fiziksel ortama rastgele bırakılmış algılayıcı düğümlerden gelen veriler erişim noktası yardımıyla bilgisayar ortamına aktarılmıştır. XML biçiminde bilgisayara gelen algılanmış veriler, geliştirilen soket uygulaması ile XML formatından ayrıştırılarak PostgreSQL veritabanına kaydedilmiştir. Veri tabanına kaydedilen veriler HTML, Php ve JQuery kullanılarak geliştirilen web tabanlı arayüz ile daha anlaşılır olarak kullanıcıya sunulmuştur. Yapılan çalışma ile, fiziksel ortamın anlık olarak web üzerinden görüntülenmesinin yanı sıra veri tabanına yazılan verilerin belirlenen bir amaca göre değerlendirilip işlenerek farklı istatistiksel bilgilere erişilebilmesi sağlanmıştır. Ayrıca, geliştirilen arayüz, sistemlerin uzaktan denetimini sağlayan bir test düzeneği ile de birleştirilerek, web tabanlı uzaktan denetim uygulaması gerçekleştirilmiştir.

WEB-BASED MONITORING AND CONTROL APPLICATION USING WIRELESS SENSOR/ACTUATOR NETWORKS

Önder YAKUT

Keywords: Wireless Sensor/Actuator Networks, Web-Based Monitoring, Networked Control

Abstract: Since it has a growing importance day to day because it is superior to the wired technologies, the utilization of Wireless Sensor Network (WSN) systems are increasing consistently. WSNs have been used for only monitoring and tracking purposes in the beginning, however; they are employed in control applications recently. WSNs are deployed in such the fields as military, medical, agricultural, industrial, and environmental.

In this study, web-based interface software is developed in order to monitor the data sent by wireless sensor nodes remotely. The data coming from the sensors that are deployed in the environment randomly is delivered to the computer by means of the access point. The sensed data in XML format then inserted into the PostgreSQL database using the socket program developed, after the parsing process. The data stored in the database is presented to end user clearly by the interface program written using the web technologies; HTML, Php, and JQuery. By the study, not only the web based monitoring of the sensed data is provided concurrently, but also the data stored in the database can be accessed for the further statistical information. Moreover, web-based remote control application is implemented by means of integrating the interface developed with a testbed which performs remote control of systems.

1. GİRİŞ

Son yıllarda iletişim teknolojisinin gelişmesiyle birlikte küçük boyutlu, düşük güçlü, algılama, işlem yapma ve haberleşme yeteneklerine sahip çok işlevli cihazlar geliştirilmiştir. Bu cihazlara algılayıcı düğüm (AD) denilmektedir. Kablosuz algılayıcı ağ (KAA), AD'lerin bir araya gelerek oluşturduğu ağ yapısıdır. AD'ler, üzerinde bulundukları algılayıcılara göre fiziksel ortama ait sıcaklık, ses, görüntü, basınç, hız, ivme, yön, hareket, ışık ve nem gibi çeşitli özellikleri algırlar. KAA'lar, fiziksel ortama rastgele dağıtıldıkları için düzensiz bir yapıya sahiptir. KAA'lar, çok sayıda AD'den meydana gelmektedir. AD'ler, bırakıldıkları fiziksel ortamdaki verileri algılayıp, bu verileri kablosuz olarak baz istasyonuna (sink, gateway) iletmek için kullanılırlar [1].

AD'ler, kendi kendilerine organize olarak KAA yapısını kendileri oluştururlar. Dağıtıldıkları fiziksel ortamda, baz istasyonuna olan uzaklıklarına göre baz istasyonu ile doğrudan veya erişim noktalı (access point) kablosuz iletişim kurarlar. Fiziksel ortamdan algıladıkları verileri, kablosuz ortam üzerinden baz istasyonuna iletirler.

KAA'lar, ilk olarak askeri amaçlı kullanılmıştır. Günümüzde, orman yangınlarını sezme, doğal yaşamı izleme, hassas tarım, sağlık, ticari, güvenlik, ev otomasyonu, robot kontrolü, ürün kalite kontrolü gibi günlük hayatımızı etkileyen bir çok alanda kullanılmaktadırlar.

Günlük yaşantımızda, hayatımızı kolaylaştırıcı etkisi giderek aratan kablosuz haberleşme teknolojileri, KAA'ların daha da yaygın kullanılmasını sağlamıştır. Bu nedenle çalışmamızda AD'lerden gelen fiziksel ortama ait veriler web ortamına aktarılmış, verilerin görüntülenmesi ve izlenmesi kolaylaştırılmıştır. Algılanan veriler çok çeşitli kullanım amacıyla web üzerinden yayınlanmış, verilerin grafikleri çizdirilerek görsel olarak kolay yorumlanabilmesi sağlanmıştır. Geliştirilen web tabanlı arayüz ile uzaktan ölü zamanlı bir sistemin denetimi de gerçekleştirilmiştir.

Sistem aynı zamanda KAA'ların denetim amaçlı kullanımına bir örnek teşkil etmektedir.

1.1. Literatür Taraması

Akyıldız ve diğ. 2001'de KAA kavramını tanımlamış ve kablosuz ağ tasarımını etkileyen faktörleri açıklamışlardır [16].

Song ve diğ. (2010), kapalı mekanda, kablosuz algılayıcı ve eyleyici ağlar (KAEA) ile robot çağırma uygulaması tasarlamışlardır. Geliştirilen uygulamada algılayıcı düğümler, çağrı yapan kişinin konum bilgisini robot üzerindeki eyleyici düğümlere göndererek, robotu kişiye hizmet sunması için kullanmışlardır [25].

Yang Peng ve diğ. (2008), aktif bir yanardağı gerçek zamanlı ve doğru olarak gözlemek için akıllı algılama sistemini çözüm olarak sunmuşlardır. Tasarlanan sistem kaynak kullanımını, durum farkındalığını ve sistem başarımını artırmada yararlı olmuştur. Uygulamada eyleyici düğümler fiziksel ortamı gözleme amacı ile kullanılmıştır [27].

Lun-Wu Yeh ve diğ (2010), kullanıcıların üzerinde taşıdığı algılayıcı düğümlerden gelen ışık bilgisine göre otonom ışık kontrol sistemi tasarlamışlardır. Kullanıcı tercihi ve enerji tasarrufuna göre eyleyici düğümler ile ortamdaki aydınlatmayı dinamik olarak değiştiren bir uygulama gerçekleştirmişlerdir [26].

Okçuoğlu ve diğ. 2008'de, doğal gözlem ortamlarındaki değişik problemlere farklı bir yaklaşım ile çözüm getirmişlerdir. Uygulamada kullanılan ideal izleme sistemi, dışarıdan herhangi bir yönlendirme ya da müdahale gerektirmeden çalışan, taşınabilir, esnek ve kullanımı kolay bir yapıda tasarlanmıştır [17].

Türker ve diğ. (2011), sağlık hizmetlerinin daha büyük kitlelere ulaşabilmesi için KAA'nın internet ile bağlantı kurma yöntemlerini açıklamışlardır. KAA'dan alınan verinin ilk olarak merkez sunucuda veritabanında depolanması ve hasta verilerinin

web üzerinden taşınarak aktarılması ile sağlık çalışanlarının hasta verilerini izlemesini kolaylaştırmışlardır [18].

Turhan ve diğ. (2011), Matlab yazılımı kullanarak ortam izleme amaçlı arayüz tasarlamışlardır. Tasarlanan arayüz, süreç denetim sistemi uygulamasında sıcaklık verilerini izlemek amacıyla kullanılmıştır [19].

Bekçibaşı ve diğ. (2010), deniz taşıtlarında ihtiyaç duyulan acil durum tespit ve kurtarma amaçlı KAA tabanlı bir sistem gerçekleştirmişlerdir. K-ATKS uygulaması ile, gerekli veriler gerçek zamanlı olarak bir uzman denetiminde izlenmiş ve insan hayatına mal olabilecek geç veya yanlış müdahale hatasını ortadan kaldırmak amaçlanmıştır [20].

Çelik ve diğ. (2009), kablosuz ağ aracılığıyla uzaktan kontrol edilebilen gezgin bir keşif robotu tasarlamış ve pratik olarak gerçekleştirmişlerdir [21].

Aktaş ve diğ. (2011), KAA altyapısı kullanarak birinci dereceden ölü zamanlı bir sistemin denetimine yönelik deney düzeneği geliştirmişlerdir. Çalışma, KAA'ların geleneksel kullanım alanlarının yanı sıra denetim amaçlı olarak da kullanılabildiğini göstermiştir [7].

1.2. Tez Çalışmasının Amacı

Bu tez çalışması, KAA'larda, AD'den gelen veriler baz istasyonu yardımıyla bilgisayar ortamına alınıp veri tabanına kaydedilerek, web ortamında anlık bir şekilde gözleme gerçekleştirmek ve web arayüzü ile KAA'larda, uzaktan sistem denetimi gerçekleştirmek ve sistemin çalışmasını web tabanlı olarak uzaktan gözlemlemek amacıyla yapılmıştır.

Bu amaca yönelik olarak, AD'lerden gelen uygulama ortamının fiziksel büyüklük verileri baz istasyonu (MD) yardımıyla bilgisayar ortamına XML-RPC yöntemi ile alınmıştır. Alınan veriler XML formatından ayrıştırılarak (XML parsing), PostgreSQL veri tabanına kaydedilmiştir. Böylece çok çeşitli amaçlarda kullanılabilir bir yapı

elde edilmiştir. Hazırlanan bu yapıyı web ortamında geliştirilmiş, uyarlanmış ve uygulanmıştır. Web arayüzü açık kaynaklı PHP web yazılımıyla, kullanıcı etkileşimi ise açık kaynaklı jQuery javascript framework'ü kullanılarak gerçekleştirilmiştir. Algılayıcı düğümlerden gelen uygulama ortamına ait veriler, anlık bir şekilde zaman ve mekan kısıtlaması olmadan gözlemlenmiş ve denetlenmiştir.

1.3. Tez Düzeni

Tez çalışması, beş ana bölümde sunulmaktadır;

Bölüm 2'de KAA'ların özellikleri hakkında genel bilgiler verilmekte, avantajlı/dezavantajlı yönleri ve uygulama alanları sunulmaktadır.

Bölüm 3'te web arayüzünde kullanılan PHP, PostgreSQL, jQuery, Ajax, Wamp Server ve Perl gibi uygulama yazılımı geliştirme programları genel olarak tanıtılmaktadır. KAA'ların altyapısını programlamada kullanılan NesC, TinyOS işletim sistemi gibi programlarda sunulmaktadır.

Bölüm 4'de KAA'larda kullanılan ağ topolojileri açıklanmaktadır ve Xmesh örgü ağı protokolü tanıtılmaktadır. Uygulamada kullanılan kablosuz cihazların özellikleri belirtilmektedir ve AD'lerden gelen verileri XML formatında sunan Xserve ağ geçidi terminal programı açıklanmaktadır.

Bölüm 5'de AD'ler kullanılarak uygulama ortamına ait verilerin alınması, oluşturulan web arayüzü ile verilerin kullanıcıya sunulmasına kadar olan aşamalar anlatılmaktadır.

Yapılan tez çalışmalarının sonuçları ve katkıları son bölümde değerlendirilmektedir.

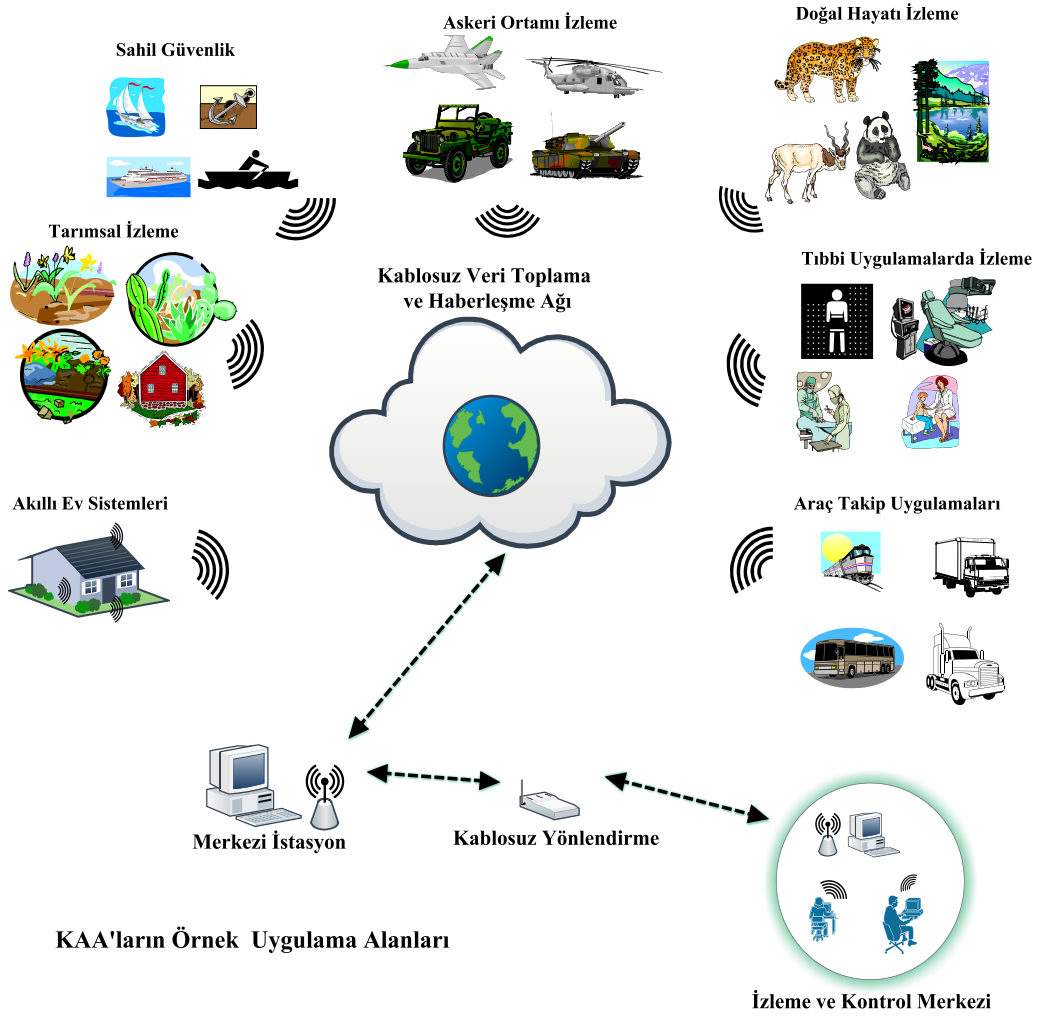
2. KABLOSUZ ALGILAYICI AĞLAR

2.1. Giriş

Son bir asır içerisinde iletişim teknolojilerinin gelişmesiyle birlikte çeşitli teknolojik haberleşme sistemleri ortaya çıkmıştır. Bu sistemler, insanların günlük hayatını etkileyerek gelişmesine devam etmektedir. Son yıllarda haberleşme teknolojisinin gelişimi ile kablosuz haberleşme sistemlerinin kullanım alanı ve sağladığı kolaylıklar önemli derecede artmıştır. Günümüzde kablosuz haberleşme alt yapısı kullanılarak dünya çapında çeşitli karakteristik özelliklere sahip veri haberleşmesi yapılabilmektedir.

Kablosuz algılayıcı ağ (KAA) kavramı, 1980'li yıllarda kullanılmaya başlanmıştır [2]. Gelişen mikroelektronik ve kablosuz haberleşme teknolojisi ile birlikte küçük boyutlu ve çok işlevli AD'ler ortaya çıkmıştır. Şekil 2.1'de KAA'ların genel kullanım alanları gösterilmektedir.

Bu bölümde Kablosuz Algılayıcı Ağ (KAA) teknolojisinin avantaj ve dezavantajlarından bahsedilecektir. Daha sonra KAA'lar ve KAEA'lar detaylı bir şekilde açıklanacak ve uygulama alanları anlatılacaktır.



Şekil 2.1: Kablosuz algılayıcı ağların genel kullanım alanları

2.2. Kablosuz Ortamdan Kaynaklanan Sınırlamalar

Kablosuz ortamda, kablosuz ağ bileşenleri birbiriyle haberleşmek için radyo frekansı (RF) teknolojisini kullanmaktadır. Kablosuz iletişim ortamının pek çok avantajlı yönü bulunmasına rağmen kullanılan radyo frekans ortamının bazı bozucu etkileride bulunmaktadır. Kablosuz haberleşmede, gönderilen sinyal; çok yönlü yayılım, yansıma, kırılma, dağılma gibi kablosuz ortamdan kaynaklanan etkiler nedeniyle alıcıya farklı sinyallerin gitmesine ve sinyalin bozulmasına neden olur. Bu durum güç tasarrufu yapmak zorunda olan AD'ler için bir sınırlama oluşturmaktadır.

2.3. Kablosuz Algılayıcı Ağların Avantajları

- KAA'lar kablo kurulumunun zor ve maliyetli olduđu uygulama ortamlarında kolay ve düşük maliyetli iletişim imkanı sağlarlar.
- KAA'larda AD'ler, var olan sisteme kolaylıkla eklenebilirler ya da çıkarılabilirler.
- AD'ler, veri süzme ve birleştirme algoritmalarını kullanarak, diđer AD'lerden gelen verileri toplayabilir, bir araya getirebilir ve bilgiye dönüştürebilirler.
- AD'lerin doğru ve hassas bir şekilde ölçüm yapabilme oranı yüksektir. Çünkü AD'ler uygulama ortamının içerisinde ve fiziksel olaylara daha yakındırlar.
- AD'ler kendi kendilerine organize olarak haberleşme alt yapısını kolay ve hızlı bir şekilde oluşturabilirler. AD'ler birbirleri üzerinden en kısa mesafeli haberleşme yolunu bulup merkezi düğüme (MD) ulaşarak, yoldan ve batarya güçlerinden tasarruf elde ederler.
- Batarya ömürleri veya dışsal ve içsel bir olaydan dolayı birden fazla AD devre dışı kalsa bile oluşturulan KAA sistemi çalışmasını sürdürebilir.
- AD'ler boyutları küçük olduđu için kullanım kolaylığı sağlarlar.

2.4. Kablosuz Algılayıcı Ağların Dezavantajları

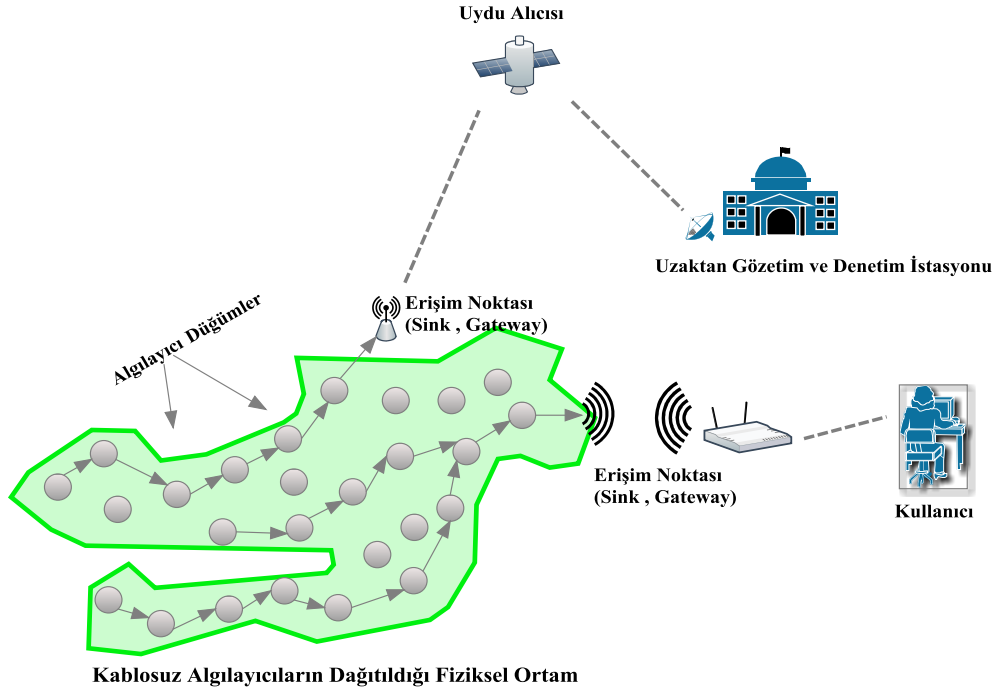
- AD'lerin yaşam süreleri, kullandıkları güç kaynağının ömrü ile sınırlıdır.
- AD'ler dış ortamın zararlı etkilerine açık olduđu için güvenlik zaafı oluşmaktadır.
- Kablosuz ortamdan kaynaklanan sınırlamalar, KAA'larda alıcı ve verici arasındaki iletişimi sağlayan radyo sinyallerini, ortamdan kaynaklanan bazı bozucu etkilere maruz bırakmaktadır. Bu nedenle AD'ler için olumsuz bir durum oluşmaktadır.

2.5. Kablosuz Algılayıcı Ağlar

Son yıllarda iletişim teknolojisinin gelişmesiyle birlikte küçük boyutlu, düşük güçlü, algılama, işlem yapma ve haberleşme yeteneklerine sahip çok işlevli cihazlar geliştirilmiştir. Bu cihazlara algılayıcı düğüm (AD) denilmektedir. Kablosuz algılayıcı ağ (KAA), AD'lerin bir araya gelerek oluşturduğu ağ yapısıdır. AD'ler, üzerinde buldukları algılayıcılara göre fiziksel ortama ait sıcaklık, ses, görüntü, basınç, hız, ivme, yön, hareket, ışık ve nem gibi çeşitli özellikleri algırlar [1].

AD'ler, uygun bir şekilde programlanıp fiziksel ortama rastgele bir şekilde dağıtılmaktadırlar. Dağıtıldıkları ortamda kendi kendine organize olarak bir birleriyle ve baz istasyonu ile iletişime geçerek KAA yapısını oluştururlar. Baz istasyonu ile ya doğrudan ya da erişim noktalı bir haberleşme kanalı oluştururlar. AD'ler aynı zamanda bir yönlendirici gibi davranırlar ve baz istasyonu ile doğrudan iletişimi olmayan diğer AD'lerin veri paketlerini, enerji tasarrufunu göz önüne alarak yönlendirme yaparlar. Yönlendirme yapılırken; mevcut enerjinin yüksek olduğu yola, yönlendirilecek yolun en düşük enerji tüketecek şekilde olmasına ve en az hop sayısı olan yola bakılarak yönlendirme işlemini yerine getirirler. AD'ler, zamanlarının çoğunu uyku modunda geçirerek enerji tasarrufu sağlamaya çalışırlar. AD'ler sınırlı pil ömürleri nedeniyle istenilen görevleri yerine getirmek için kesmeler ve zamanlayıcılar ile tetiklenirler. Bir AD'nin yaşam ömrü, AD'yi besleyen bataryanın ömrü ile çok yakından ilişkilidir.

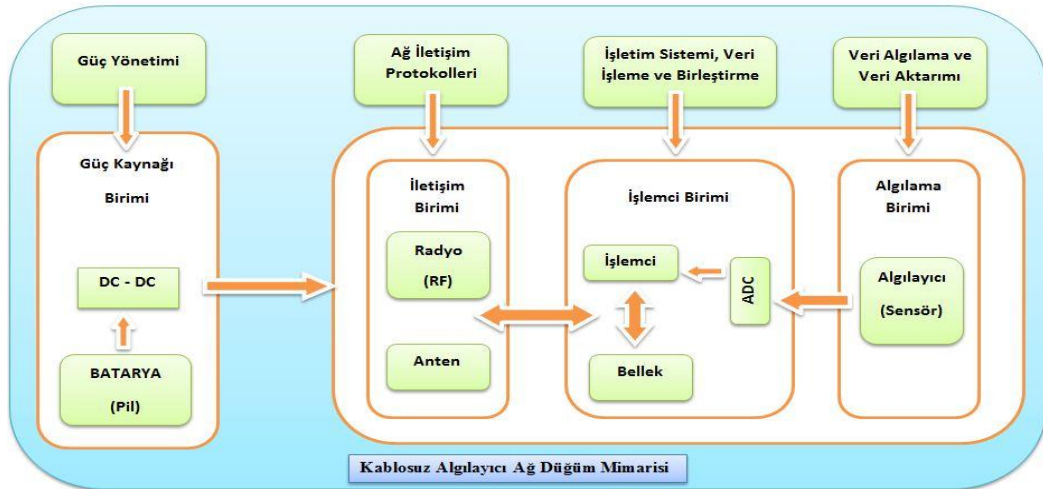
Tipik bir KAA uygulaması çok sayıda AD'den meydana gelir. Şekil 2.2'de görüldüğü gibi AD'ler fiziksel ortama rastgele dağıtılmıştır. AD'ler bu fiziksel ortamda kendi kendilerine organize olarak fiziksel ortama ait verileri toplarlar, işlerler ve bu verileri yayarlar. Bu sayede KAA'daki bilgiye her an her yerden kolay bir şekilde ulaşılması sağlanmış olur. Fiziksel ortamdaki verileri gözleme ve denetleme işlemleri uydular ve yerel erişim noktaları aracılığı ile yapılabilmektedir.



Şekil 2.2: Örnek bir kablosuz algılayıcı ağ uygulaması

2.6. Kablosuz Algılayıcı Ağ Düğümleri

AD'ler bir çok işlevsel birimden meydana gelir. AD'lerin yapısında işlemci, iletişim, algılama ve güç kaynağı birimi bulunmaktadır. Şekil 2.3'de AD yapısı ve görevleri gösterilmiştir.

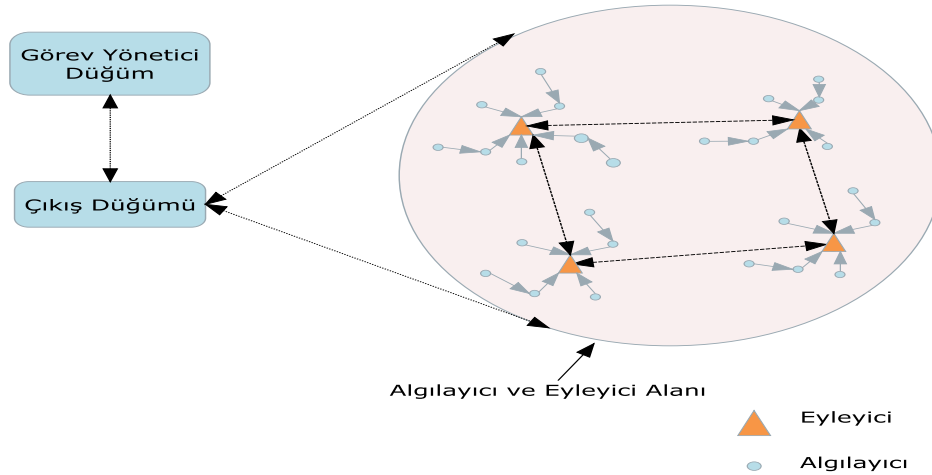


Şekil 2.3 : Bir algılayıcı düğümün yapısı ve mimarisi

Güç kaynağı birimi, AD'nin diğer birimlerinin enerji ihtiyacını karşılamaktadır. AD'nin yaşam ömrü güç kaynağına bağlıdır. İletişim birimi, Ağ iletişim protokolleri çerçevesinde AD ile KAA arasındaki veri haberleşmesini sağlamaktadır. İşlemci birimi, AD'nin çalışmasını devam ettirmesini, algılanan verilerin işlenmesini ve KAA ortamına gönderilecek yapıda birleştirilmesini sağlamaktadır. Algılama birimi, fiziksel ortamdaki analog verileri algılayıp, işlemci biriminin ADC kısmına aktarılmasını sağlamaktadır.

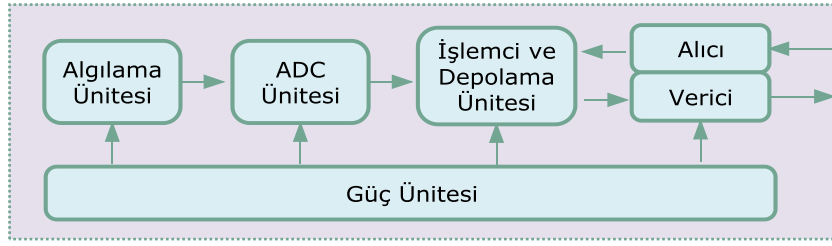
2.7. Kablosuz Algılayıcı ve Eyleyici Ağlar

Kablosuz algılayıcı ve eyleyici düğümlerin bir araya gelerek oluşturduğu yapıya kablosuz algılayıcı ve eyleyici ağlar (KAEA) denilmektedir. Eyleyici düğümler, ortamdan toplanan verilere göre uygun denetim yapmak amacı ile kullanılmaktadırlar [28]. KAEA'lar ile ortamdan gelen verileri gözleme işlemi yapmanın yanı sıra toplanan verileri işlerler ve işlenmiş verilere göre eylem komutları oluşturup uygun denetimi gerçekleştirirler.

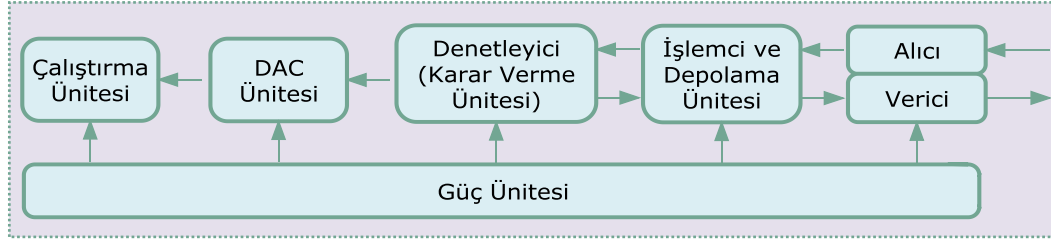


Şekil 2.4 : KAEA fiziksel mimarisi [28]

KAEA'larda, algılayıcı ve eyleyici düğümlerin görevleri sırasıyla, fiziksel ortamdan gelen verileri toplamak ve toplanan verilere göre uygun eylemleri gerçekleştirmektir. Şekil 2.4'de çıkış düğümü uçtan uca tüm ağı gözlemlemekte ve aynı zamanda görev yönetici düğümü ve algılayıcı/eyleyici düğümleri ile iletişim kurmaktadır [28].



a



b

Şekil 2.5 : a) Algılayıcı b) Eyleyici bileşenleri [28]

Şekil 2.5’de KAEA’larda kullanılan algılayıcı ve eyleyici düğümlerin bileşen yapıları gösterilmektedir.

Şekil 2.5 (a)’da algılayıcı düğüm, güç ünitesi, alıcı-verici ünitesi, işlemci ve depolama ünitesi, analog dijital çevirici ünitesi ve algılama ünitesinden meydana gelmektedir. Algılama ünitesi, fiziksel olayları gözlemler. Toplanan analog veriler, analog-dijital çevirici ünitesi tarafından dijital veriye dönüştürülür. Dijital veri, işlemci ünitesi tarafından analiz edilir ve daha sonra alıcı-verici ünitesi tarafından yakında bulunan bir eyleyici düğüme iletilir. Şekil 2.5 (b)’de eyleyici düğüm, algılayıcı düğümlerden gelen verileri alır ve karar verme ünitesi yardımıyla verileri işleyerek eylem komutları oluşturur. Oluşturduğu eylem komutlarını dijital-analog çevirici vasıtası ile analog sinyale dönüştürür ve çalıştırma ünitesine gönderir [28].

2.8. Sonuç

Bu bölümde KAA’lar, KAEA’lar ve algılayıcı ve eyleyici düğümler ile ilgili açıklamalar yapılmış, KAA’ların avantajlı ve dez avantajlı yönlerinden bahsedilmiştir. KAA’lar ve KAEA’lar resimler ile açıklanmıştır.

3. UYGULAMADA KULLANILAN PROGRAMLAR

Bu bölümde web arayüzünü hazırlarken kullandığımız uygulama programları açıklanmaktadır.

3.1. PHP

PHP, web tabanlı uygulamalar için tasarlanmış, sunucu tarafında (server side) çalışan bir script dilidir. 1994 yılında Ramus Lerdorf tarafından oluşturulmuştur. PHP, açık kaynak (open source) tabanlı bir yazılım türüdür. Kaynak koduna erişilip ücretsiz bir şekilde kullanılabilir, kaynak kodunda değişiklik yapılabilir ve yeniden dağıtılabilir.

Kelime anlamı olarak ilk dönemlerde Personal Home Page (Kişisel Ana Sayfa) olarak adlandırılmıştır. GNU isimlendirme standartıyla uyumlu hale getirilmesi ile birlikte PHP, Hypertext Preprocessor olarak yeniden adlandırılmıştır. PHP'nin resmi web sitesi <http://www.php.net> internet adresidir. Bu adresten PHP kaynak kodu indirilip kullanılabilir ve PHP hakkında kullanıcı klavuzundan yararlanılabilir.

Statik HTML (Hypertext Markup Language) sayfaları yerine, PHP kullanarak dinamik olarak kullanılabilen, yönetilebilen ve kullanıcılar ile daha çok etkileşim kurulabilen web sayfaları oluşturulabilmektedir.

3.2. jQuery

jQuery; javascript, ajax ve efekt kütüphanesi olan bir çerçevedir (framework). jQuery 2006 yılında MIT ve GPL lisansı ile açık kaynaklı olarak geliştirilmiştir. jQuery'nin resmi web sitesi, <http://www.jquery.com> [10] internet adresidir. Bu web sitesinden jQuery kaynak koduna, jQuery ile ilgili dökümantasyona, eğitsellere, eklentilere ve soruların yazılabildiği forumlarda ulaşılabilir.

JQuery ile web sitelerinin bir kısmında ya da tamamında görsel ve yapısal değişiklikler yapılabilmektedir. Web sitelerinde jQuery'i kullanarak animasyonlar kolay bir şekilde oluşturulabilmekte ve bu sayede web arayüzü ile kullanıcıların etkileşimi üst düzeye çıkabilmektedir. JQuery ile PHP, ASP gibi sunucu taraflı çalışan dillere veri gönderilip alınabilmektedir. JQuery'nin önemli bir özelliği ise eklentileridir. Bu eklentiler belli görevleri gerçekleştiren kodlardır. Bu eklentiler sayesinde web sitesine işlevsellik kazandırmak çok daha kolay olmaktadır. Geliştirilen web arayüzünde, "Highcharts JS" [23] ve "jsPlumb" [22] eklentileri kullanılmıştır. Bu eklentiler kaynak olarak jQuery çerçevesini kullanmaktadırlar ve ticari amaçlı olmayan kullanımlar için açık kaynaklı GPL lisansı ile lisanslandırılmışlardır.

3.3. AJAX

AJAX (Asynchronous JavaScript And XML – Eş zamanlı olmayan JavaScript ve XML), İstemci tarafından sunucuya gönderilen bir isteğin sonucunun web sayfasının bütünüyle yeniden yüklenmesine (post back) gerek kalmadan, sunucudan istemciye ulaştırılmasıdır. Ajax ile web sayfasında yapılan işlemler sunucuya XMLHttpRequest isteği olarak gönderilir. XMLHttpRequest web tarayıcıları üzerinde çalışan bir bileşendir.

Ajax ile sağlanan en büyük özellik ise kısmi güncellemedir. Bütün halinde bir web sayfasının sunucuya gönderilmesi yerine, ilgili kısım sunucuya gönderilir, yorumlanır, işlenir ve sonucu web sayfasında ilgili yere yazılarak kısmi güncelleme sağlanmış olur. Bu sayede istemci ve sunucu arasında kullanılan bant genişliği azaltılmış olmaktadır.

3.4. WAMP SERVER

WAMP server; Apache, MySQL ve PHP gibi sunucu özellikli yazılımları windows işletim sistemi üzerinde çalışmasını sağlayan açık kaynaklı bir sunucu yazılımıdır. Wamp server'ın resmi web sitesi <http://www.wampserver.com> [11] internet adresidir. Bu adresten GPL lisanslı wamp server'ı indirip bilgisayara kurulabilmekte ve wamp server ile ilgili dokümantasyona erişilebilmektedir.

3.5. PostgreSQL Veri Tabanı

PostgreSQL, veri tabanları için geliştirilmiş olan ilişkisel modeli kullanan ve standart sql dilini yapısında bulunduran bir veri tabanı yönetim sistemidir (DBMS) [3].

PostgreSQL'in resmi web sitesi <http://www.postgresql.org> internet adresidir. Bu web sitesinden PostgreSQL veri tabanına ait tüm sürümlerine erişilebilmekte, veri tabanına ait dökümantasyonu elde edilebilmekte ve forumlardaki sorulara ve cevaplara ulaşılabilir.

PostgreSQL veri tabanı sunucusu bilgisayara indirilip, kurulduktan sonra bu sunucuya erişebilmek için PgAdmin3 arayüzü kullanılabilir. PgAdmin3, PostgreSQL için en popüler ve zengin içerikli açık kaynak yönetim ve geliştirme platformudur. PgAdmin3; Linux, FreeBSD, Solaris, Mac OSX ve Windows platformlarında PostgreSQL veri tabanı sunucusunu yönetmek için kullanılabilir. PgAdmin3, temel SQL sorgularından karmaşık veri tabanları geliştirmeye kadar kullanıcıların tüm ihtiyaçlarını karşılamak için tasarlanmıştır. PgAdmin3'ün grafiksel arayüzü tüm PostgreSQL'in özelliklerini destekler ve yönetimini kolaylaştırır. Uygulama ayrıca sözdizimini vurgulayan SQL editörünü içerir. Veritabanı sunucusu ile haberleşmek için ek sürücüler gerektirmez [4]. PgAdmin3, yönetim ve geliştirme aracı resmi web sitesinden <http://www.pgadmin.org> indirilip, bilgisayara kurulabilir.

3.6. PERL

Perl, Larry Wall tarafından geliştirilmiş açık kaynak kodlu bir programlama dilidir. Perl veri taramada, veri işleme ve veri dönüştürmede rapor üretmek için kullanılan başarımlı yüksek bir yorumlama dilidir. Perl'de kod yazarken fazladan fonksiyon bildirimine, veri türü ve değişken tanımları yapmaya gerek kalmadan hızlıca kod yazılabilir.

Perl yazılım programının resmi web sitesi <http://www.perl.org> [12] internet adresidir. Bu adresten Perl ile ilgili dökümantasyon ve forumlara ulaşılabilmektedir. Perl'ün windows işletim sisteminde çalışan sürümü ActiveState Perl'dür. ActiveState Perl'e ait resmi web sitesi <http://www.activestate.com> [15] internet adresidir. Bu adresten ActiveState Perl ile ilgili dökümantasyona, foruma ve ActiveState Perl kurulum paketine erişilebilmektedir.

3.7. XML Ve XML – RPC

XML, Extensible Markup Language (Genişleyebilir İşaretleme Dili) ifadesinin kısaltılmış biçimidir. XML, etiketlerden oluşan bir işaretleme dilidir. XML'de işaretlenen her bir veriye element adı verilmektedir. XML belgeleri, belli kurallara göre oluşmuş elementlerin bir araya gelmesi ile oluşmaktadır. XML belgesi içerisinde bulunan elementler ile anlamlı veriler taşınmaktadır. Geliştirilen bir çok program ve sistem, kendisi için gerekli ayar dosyalarını XML'i kullanarak yapmaktadır. Özellikle bankacılık ve finans işlemlerinde ve web tabanlı portallarda veriler XML belgeleri ile taşınmaktadır. XML belgelerindeki elementler ile taşınan anlamlı verileri elde edebilmek için XML ayrıştırıcı (XML parsing) kullanılmaktadır. Ayrıştırma işlemi yapmak için yazılım programları (C, C++, C#, Perl, Java gibi) kullanılması gerekmektedir.

XML-RPC, Extensible Markup Language-Remote Procedure Call (Genişleyebilir İşaretleme Dili-Uzaktan Yordam Çağırma) ifadesinin kısaltılmış biçimidir. XML-RPC, her hangi bir yazılım programının kurulu bir ağ üzerinden uzaktan yordam çağırısı yapabilmesi için farklı işletim sistemleri üzerinde çalışan ve farklı ortamlarda bulunan bir web hizmetidir.

XML-RPC genellikle, soket programlama teknikleriyle kullanılmaktadır. Bir soket oluşturulur ve oluşturulan soket ile uzaktaki bir sistemden gönderilen parametrelere göre bir yordam çağırısı isteğinde bulunulur. Uzaktaki sistem, uzaktan yordam çağırısı isteğini XML-RPC dinleyicisi vasıtasıyla alarak ilgili yordama parametreleri gönderir ve bir sonuç üretmesini sağlar. Üretilen sonuç cevap olarak XML-RPC istemcisine geri gönderilir ve istemci yordam çağırısının cevabını alarak işlemine

devam eder. XML-RPC ile uzak bir sistemden yordam çağırısı yapıldığı için isteğin cevabı bir gecikme ile gelmektedir. Oluşturulan socket programının bu gecikmeyi tolere edecek yapıda geliştirilmesi gerekmektedir.

3.8. TinyOS İşletim Sistemi

TinyOS, kablosuz gömülü algılayıcı ağlar için tasarlanmış açık kaynaklı bir işletim sistemidir [5]. TinyOS işletim sistemi, nesC programlama dili kullanılarak geliştirilmiştir. TinyOS işletim sistemini <http://www.tinyos.net> [13] web sitesinden bilgisayarınıza indirebilir ve TinyOS ile ilgili kaynaklara erişebilirsiniz.

TinyOS, açık kaynaklı, bileşen tabanlı bir mimariye sahiptir. Geliştirilen uygulamaların ihtiyaçlarına göre bileşenler, işletim sistemini yapılandırmaktadır. Bileşenler kendi aralarında haberleşebilmek için bir birilerine arayüzler kullanarak bağlanmışlardır. TinyOS işletim sistemi, algılayıcı düğümlere; veriyi algılamak, algılanan veriyi işlemek ve depolamak, eylemde bulunmak ve veri haberleşmesi yapmak için arayüzler ve bileşenler sağlamaktadır.

TinyOS işletim sistemi, kablosuz algılayıcı ağların güç tüketimini azaltmak için tasarlanmıştır. TinyOS, işlemcinin olabildiğince kısa süre içerisinde işlemlerini yapmasını sağlar ve işlemciyi az güç harcadığı uyku modunda bekletir. TinyOS, olay güdümlü kesmeler ve zamanlayıcılar ile uygulamaların görevlerini yerine getirmesini sağlar. TinyOS işletim sistemi, kablosuz algılayıcı ağların kısıtlı olan güç kaynaklarını olabildiğince verimli bir şekilde kullanıp, çalışma ömürlerini uzatmalarını sağlamaktadır.

3.9. nesC Programlama Dili

NesC (network embedded systems C – Gömülü ağ sistemleri için C), TinyOS işletim sistemi modelinin yapısal kavramlarını somutlaştırmak ve yürütmek için tasarlanmış C dilinin bir uzantısıdır [5]. NesC açık kaynaklı bir programlama dilidir ve <http://nesc.sourceforge.net> [14] internet adresinden nesC'nin kaynak koduna

ulařabilir, nesC uygulaması geliřtirmek iin dökümantasyon ve destek saęlayabilirsiniz.

NesC programlama dili ile bileřen tabanlı ve olay güdümlü TinyOS iřletim sistemi geliřtirilmiřtir. Kısıtlı bellek ve güç kaynaęı ile alıřması gereken algılayıcı düęümler, nesC programlama dili ile olay güdümlü alıřan, esnek bir eř zamanlılık sunan ve bileřen tabanlı uygulamalar geliřtirilerek programlanmaktadırlar.

3.10. Sonu

Bu bölümde, oluřturduęumuz web tabanlı arayüzde, KAA merkezi düęümünde ve AD'lerde kullanılan programlama dilleri tanıtılmıřtır.

Web tabanlı arayüzde, Php, jQuery, Ajax, Wamp Server, PostgreSQL veri tabanı ve Perl yazılımları kullanılmıřtır. Merkezi düęüm ve AD'lerde ise TinyOS, nesC ve XML-RPC yapıları kullanılmıřtır.

4. KABLOSUZ AĞ MİMARİLERİ VE KULLANILAN CİHAZLAR

4.1. Örgü (Mesh) Ağının Temelleri

Kısa mesafeli, kablosuz örgü (mesh) ağı mimarileri, bilgisayar dışındaki cihazları yönetmek için güç tasarrufu sağlamaktadır. Kendi kendine organize olan örgü ağı mimarileri, savaş alanında kullanılan hareket algılama algılayıcıları, gıda ürünlerinin sıcaklığını ölçen termometreler ve gezgin ilaç ve tıbbi cihazlar sayesinde hastanın önemli bulgularının izlenmesi gibi işlemlerde kablosuz olarak cihazdan cihaza iletişim kurulmasına katkı sağlamaktadırlar. KAA'lar, uygulamaların gereksimine bağlı olarak uygun teknolojilerden oluşturulabilirler ve farklı önceliklere hitap eden çeşitli şekillerde tasarlanabilirler. Kablosuz örgü ağı sistemleri bir dizi ortak gereksinimleri paylaşmaktadırlar. Bunlar;

- **Düşük Güç Tüketimi:** Uzun vadeli çalışmayı desteklemek için radyo bağlantısının güç tüketimini azaltır ve böylece cihazın batarya ömrünü artırmış olur.
- **Kullanım Kolaylığı:** Ağ protokolü, algılayıcı ağın öz-organize bir biçimde kendi kendine organize olarak iletişime başlamasını sağlar.
- **Ölçülebilirlik:** Ağ çok sayıda algılayıcı düğümün gereksinimlerini desteklemelidir ve üstsel büyümeye neden olmadan gelecekteki büyümeyle baş edebilmelidir.
- **Cevap verilebilirlik:** Topoloji, özellikle hareketli mimariler, ekipmanlar veya giyilebilir (insanların veya hayvanların üzerinde taşınabilir) algılayıcılar gibi hareketli algılayıcı düğümlerin olduğu uygulamalarda keşif ve yeniden keşif için etkili olmalıdır.
- **Mesafe:** Kısa mesafeli hareketlilikte düşük güçlü RF sinyallerini yaymak için çok etkilidir. Ancak çoğu zaman yüksek güçlü sinyalleri daha uzun mesafeye

aktarıırken o kadar etkili değildir. Bu yüzden, taşınabilir RF terminallerinin baz istasyonundan çok uzak olduğu durumlarda bir tekrarlayıcıdan diğerine veri paketlerini aktarabilmek için, çoklu-atlamalı (multi-hopping) yönlendirmeyi destekleyen bir ağ protokol biçimini kullanan yönlendiricilere sahiptir.

- İki Yönlü İletişim: Baz istasyonu, algılayıcı düğümlerden gelen veri iletim sinyallerini alabilir, belirli işletim parametrelerini ayarlamak için algılayıcı düğümlere sinyal gönderebilir. Ağ geçidi ve AD arasındaki bu haberleşme iki yönlü olmaktadır.
- Güvenilirlik: Kablosuz ortamda veriler çok çeşitli türlerde saldırılara ve tehditlere açıktır. Veri güvenirliliği her zaman önemlidir ve özellikle tıbbi izleme uygulamalarında hayati önem taşımaktadır.
- Küçük Modül Biçim Faktörü: Ağ modülleri için çok küçük biçim (şekil) faktörüne ihtiyaç duyulur. Uç noktalar içine sığabilecek veya var olan bir cihaza kolay bir şekilde eklenebilecek yapıda olmalıdırlar.

Güçlü bir ağ protokolü, belirli örgü ağı tasarımının yanı sıra yukarıda bahsedilen gereksinimleri yerine getirme ihtiyacı duyar. Ağ protokolü, ağın topolojisini destekler ve ağ boyunca verinin yönlendirilmesini sağlar [6].

4.2. Topolojiler

KAA uygulamalarını hayata geçirmek için birkaç mimari kullanılabilir. Bunlar yıldız (star), örgü (mesh) ve yıldız-örgü (star-mesh) melez topolojileridir. Her bir topolojinin kendine özgü zorlukları, avantajları ve dezavantajları bulunmaktadır. Bu topolojileri anlayabilmek için KAA bileşenleri hakkında bilgi sahibi olmak gerekmektedir. Bu bileşenler;

- Bitiş Noktaları (Endpoint): Algılayıcılar ve eyleyiciler algılanan veriyi yakalamak için entegre edilmişlerdir. AD'ler yukarı yönde ve aşağı yönde ağ mesajlaşmasını bitiş noktasıyla yapmaktadırlar.
- Yönlendiriciler (Routers): Radyo dalgalarının engellendiği durumlarda bu engellerin etrafından yönlendirme yapmak ve ağ sıkışması veya aygıt hataları

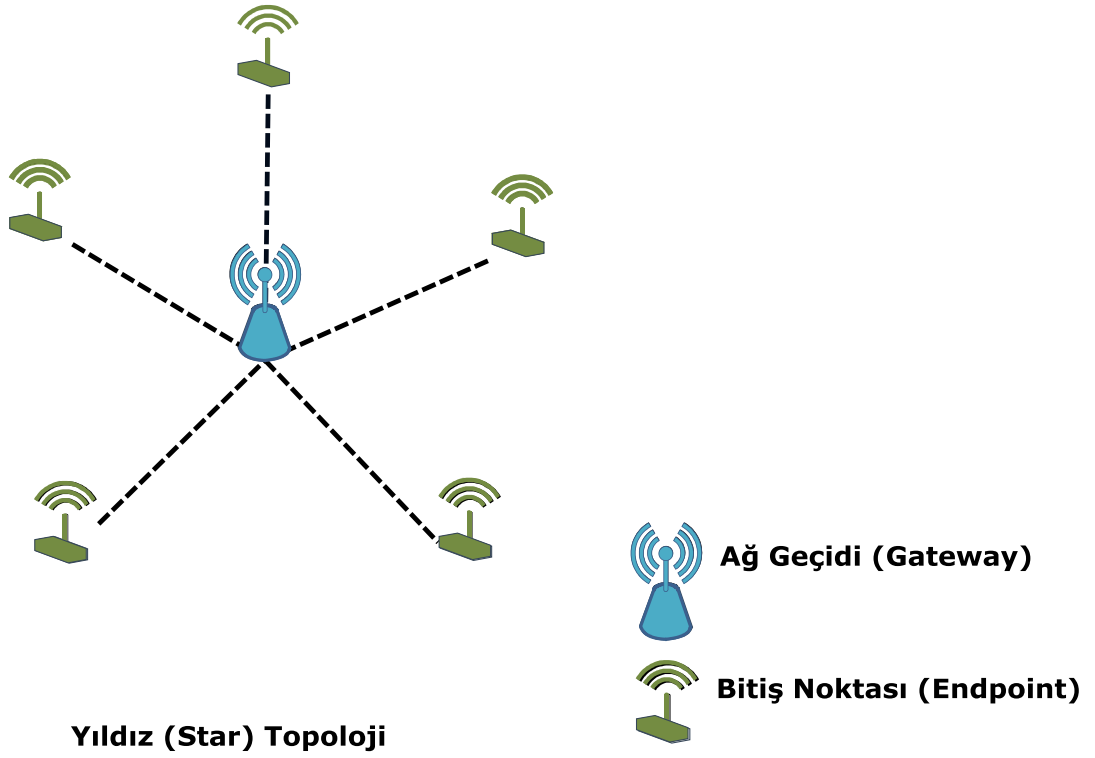
durumunda yedek yönlendirmeyi sağlamak için ağ kapsama alanını genişletmektedirler. Bazı durumlarda yönlendiriciler, bitiş noktası gibi davranabilmektedirler.

- Ağ Geçitleri (Gateways): Ağdan, sunucu arayüzünden, yerel alan ağından veya internette gelen veriyi bir araya getirmektedirler. Ağ parametrelerini yapılandırmak ve ağ başarımını izlemek için bir portal olarak kullanılmaktadırlar.
- Sistem Yazılımı: Ağ protokolünün, kendi kendini yapılandırmasına ve iyileştirmesine olanak sağlayan ad-hoc ağlar sağlamaktadırlar.

Topoloji, donanım bileşenlerinin yapılandırılmasını ve yapılandırmaya göre verinin nasıl iletileceğini belirlemektedir. Çok yaygın üç topoloji türü vardır. Bunlar; yıldız, örgü, yıldız-örgü melez topolojileridir. Her bir topoloji bazı durumlarda kendisi için uygun olan fakat diğer topolojiler için uygun olmayan özelliklere sahiptir [6].

4.2.1. Yıldız (Star) Topolojisi

Yıldız topolojisi, tüm kablolu algılayıcı düğümlerinin bir geçit düğümüyle doğrudan haberleşme mesafesi içinde (genellikle 30 metreden 100 metreye kadar) olan tek-atlamalı (single-hop) bir ağ sistemidir. Tüm AD'ler özdeştir ve her birinin bitiş noktaları bulunmaktadır. Ağ geçidi (gateway), algılayıcı bitiş noktalarına veri ve komut haberleşmesi hizmetini sunmaktadır. Ağ geçidi ayrıca daha yüksek seviyeli kontrol veya izleme sistemlerine veri iletmek için kullanılmaktadır. Şekil 4.1'de görüldüğü gibi bitiş noktaları birbirilerine veri ve komut iletmezler, sadece ağ geçidini bir koordinasyon noktası olarak kullanırlar.

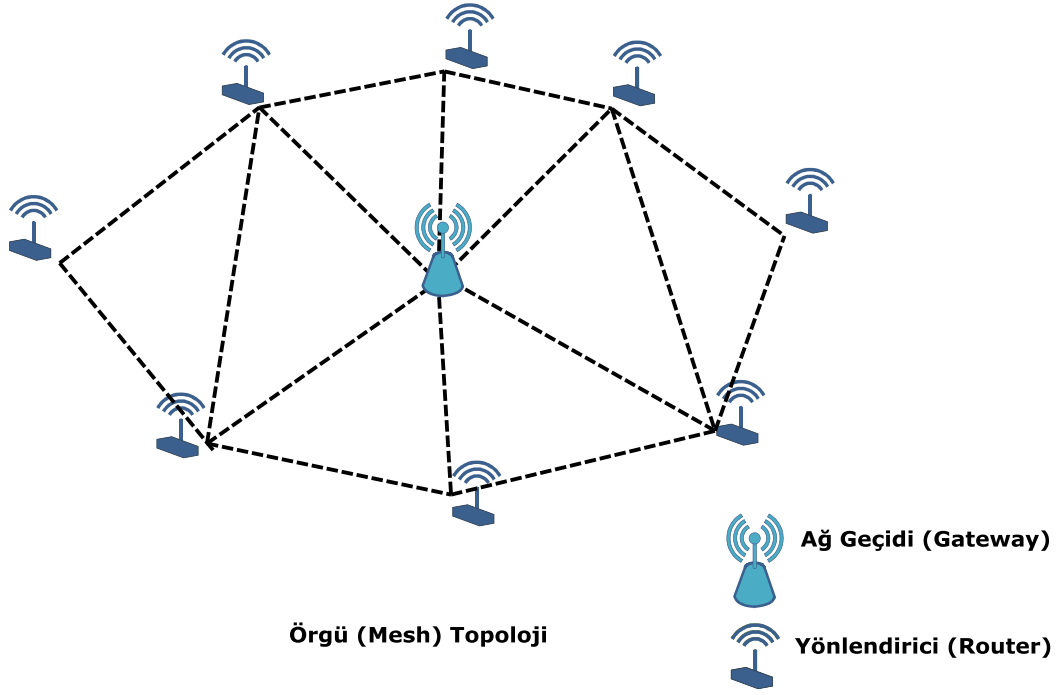


Şekil 4.1 : Yıldız (star) topolojisi [6]

Yıldız topolojisi, en düşük toplam güç tüketimini sağlar fakat her bir bitiş noktasının ağ geçidine olan radyo mesafesi iletim alanı sınırlı olmaktadır. Ayrıca bitiş noktalarının alternatif iletişim yolları bulunmamaktadır. Bir yol tıkanırsa, ilgili bitiş noktasıyla iletişim kaybolmaktadır [6].

4.2.2. Örgü (Mesh) Topolojisi

Örgü topolojisi, tüm kablosuz algılayıcı ağ düğümlerinin bütün yönlendirme işlemlerini yerine getirmektedir. AD'lerden ve ağ geçidinden gelen verilerin birbiri üzerinden atlatılarak iletişim kurulmasını sağlayan çoklu-atlamalı (multi-hopping) bir ağ sistemidir. Bu Xmesh yapılandırması için bir standarttır. Yıldız topolojisinin aksine, örgü topolojisindeki düğümler diğer yönlendirici düğümler arasındaki mesajları Şekil 4.2'de görüldüğü gibi iletebilmektedirler.

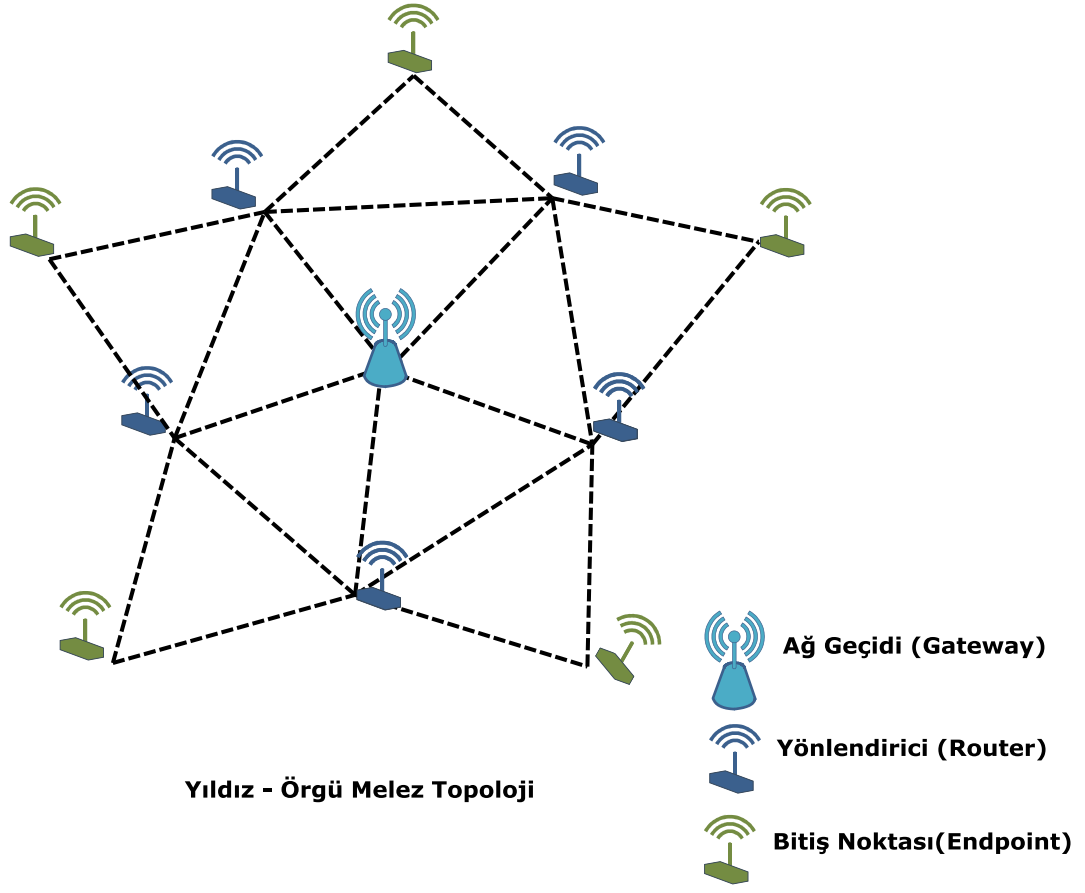


Şekil 4.2 : Örgü (mesh) topolojisi [6]

Örgü yapısı ile algılanan verinin yayılımı, KAA'ların teorik olarak sınırsız bir mesafede genişlemesine olanak sağlamaktadır. Örgü ağı ayrıca diğer AD'lere ve ağ geçidine çoklu yollarla bağlı olan her bir AD'nin yüksek dereceli hatalarına karşı tolerans göstermektedir. Bir AD başarısız olursa, örgü ağı kendini yeniden yapılandırabilmektedir [6].

4.2.3. Yıldız – Örgü (Star - Mesh) Melez Topolojisi

Yıldız-örgü melez topolojisi, yıldız topolojisinin basitliği ve düşük güçlü yönünün yanı sıra örgü topolojisinin kendini iyileştirmesini ve genişletilmiş mesafesini bir arada sunmaktadır. Yıldız-örgü melez topolojisi, örgü ağı içerisindeki yönlendiriciler, etrafındaki AD'leri yıldız topolojisine göre düzenlerler. Şekil 4.3'te görüldüğü gibi yönlendiriciler hem ağın iletişim mesafesini genişletmek için hem de hata toleransını artırmak için hizmet sunmaktadırlar.



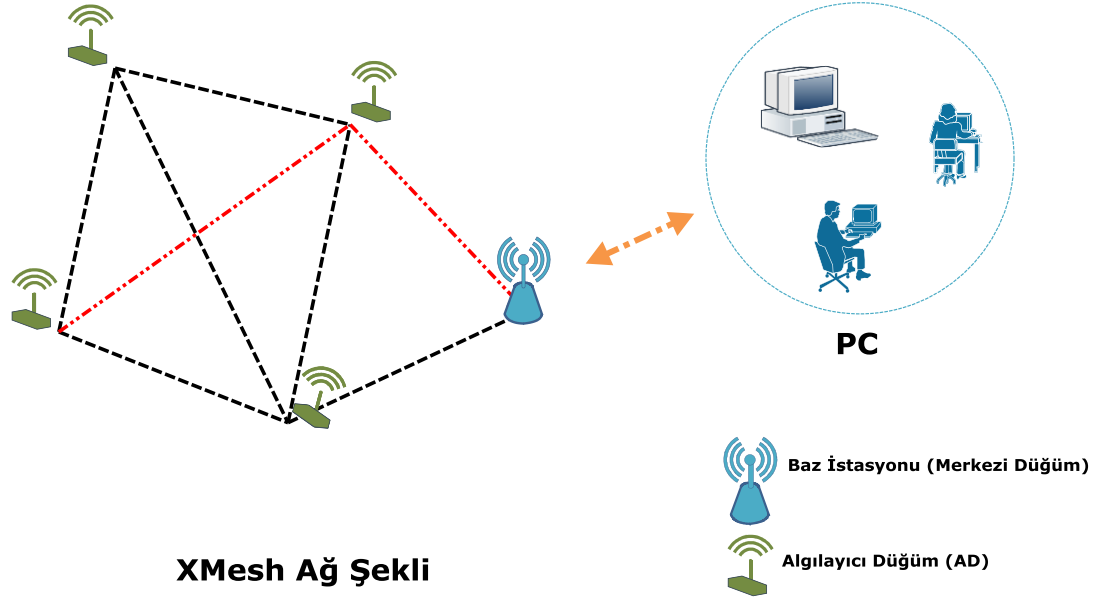
Şekil 4.3 : Yıldız-örgü (star-mesh) melez topolojisi [6]

Kablosuz AD'ler çoklu yönlendiriciler ile haberleşebilmektedirler. Başarısız olan AD'ler olursa ağ kendini yeniden yapılandırabilmektedir [6].

4.3. Xmesh

Xmesh, kablosuz ağlar için Crossbow firması tarafından geliştirilen çoklu-atlamalı ve ad-hoc özellikli bir örgü ağı protokolüdür. Xmesh ağı, birbiri ile kablosuz olarak haberleşebilen ve bir baz istasyonuna radyo mesajı gönderme yeteneğine sahip olan AD'lerden meydana gelmektedir. Atlamalı yapı, radyo haberleşme mesafesini etkin bir biçimde genişletir ve mesajları iletmek için gerekli gücü azaltmaya çalışır. Bu yapı sayesinde Xmesh, iki önemli faydayı sağlar; radyo kapsamını ve güvenilirliğini artırır ve iki düğümün birbiriyle haberleşmek için doğrudan radyo mesafesinde olması gerekliliğini ortadan kaldırmış olur. Bir mesajı, bir ya da daha fazla düğüme,

düğüm arasında veriyi yönlendirmek için ulaştırabilir. Şekil 4.4'te görüldüğü gibi, iki düğüm arasında iletişim kurulamazsa, bu iki düğümün etrafındaki diğer düğümler yeniden yönlendirme yaparak iletişimi sağlarlar.



Şekil 4.4 : Xmesh ağı [6]

Genel olarak, AD'ler düşük güç modunda çalışırlar, zamanlarının çoğunu uyku modunda harcarlar, bu sayede batarya ömürlerini artırmaya çalışırlar [6].

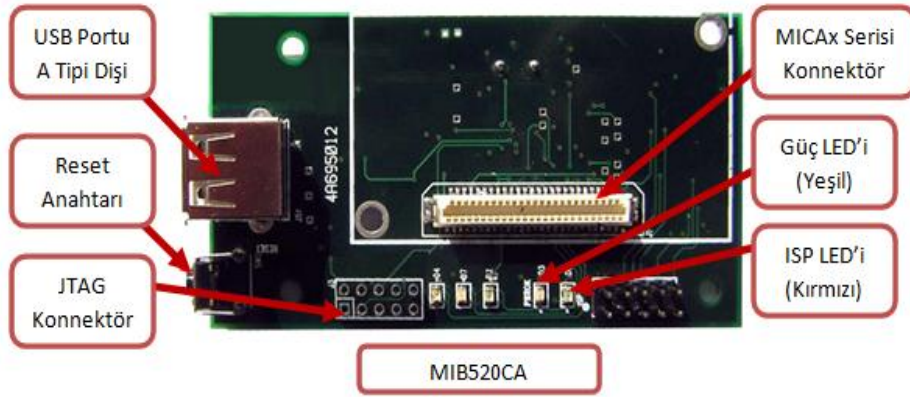
4.4. Uygulamada Kullanılan Algılayıcı Düğümler ve Erişim Noktası

KAA uygulamasında kullanılan MicaZ AD'leri 16 MHz atmega128L işlemcisine sahiptir ve yapısında 128 KB kod belleği, 4 KB veri belleği ve 4 KB EEPROM bellek bulunmaktadır. Kablosuz haberleşme için IEEE 802.15.4 standardı ile uyumlu Chipcon CC2420 MPR 2600 haberleşme yongasını kullanarak 2.4 GHz frekans bandında 250 Kbps hızında iletişim yapabilmektedir. AD'lerin üzerine monte edilen MTS400CA algılayıcı kartı ile fiziksel ortamdan sıcaklık, ışık şiddeti, nem, ivme ve basınç bilgilerini ölçmektedir. AD'leri programlamak ve erişim noktası (baz istasyonu) olarak onlar ile iletişime geçmek amacıyla MIB520 programlama boardu kullanılmaktadır. MIB520, bilgisayar ile USB seri portu üzerinden haberleşmektedir.

Şekil 4.5’de MicaZ algılayıcı düğümü ve Şekil 4.6’da MIB520 programlama bordu (erişim noktası) görülmektedir.



Şekil 4.5 : MicaZ algılayıcı düğümü [6]



Şekil 4.6 : MIB520 programlama bordu (erişim noktası) üstten görünüşü [6]

4.5. Xserve

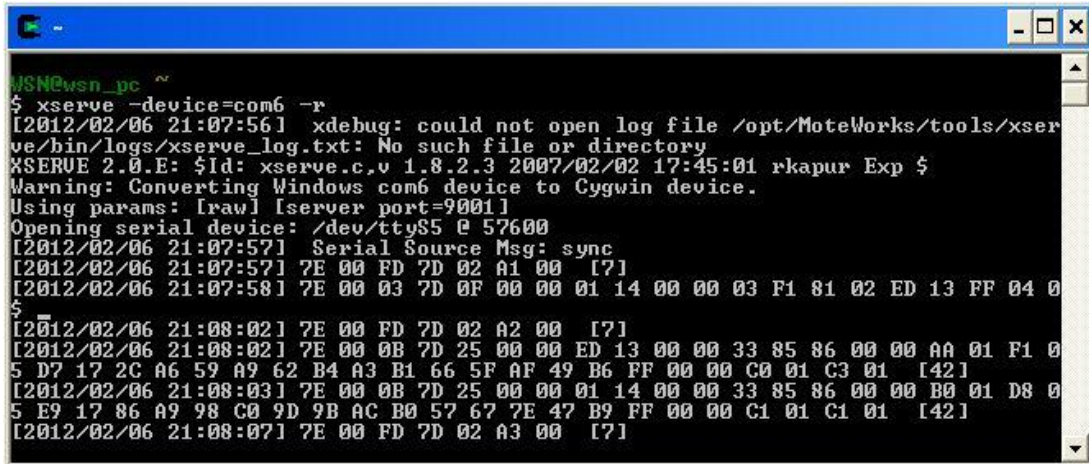
Xserve, kablosuz örgü ağı ile Xmesh mimarisiyle etkileşimde bulunan kurumsal uygulamalar arasında birincil ağ geçidi olarak hizmet sunar. Xserve, örgü ağı ile dış dünya uygulamaları arasındaki verileri yönlendirmek, ayrıştırmak, dönüştürmek ve işlemek için hizmetler sunmaktadır. Sunulan hizmetler, XML tabanlı yapılandırma dosyaları ve yüklenebilir eklenti modülleri kullanılarak özelleştirilebilmektedir.

Xserve, kendisiyle veya örgü ağı ile etkileşim kurmak isteyen uygulamalar için çoklu iletişim imkanı sağlar.

Kullanıcılar, uygulamalara doğrudan ya da güçlü bir XML-RPC komut arayüzü üzerinden erişebilir ve bir terminal arayüzü aracılığıyla (Cygwin) Xserve ile etkileşime geçebilirler. Xserve, çalışma anında gelen verileri ayrıştırabilir, dönüştürebilir ve işleyebilir. Xserve, gelen verileri terminal ekranına yazdırabilir, bir dosyaya aktarabilir, bir veri tabanına depolayabilir ve XML formatında yayımlayabilir [8].

Xserve, verileri üç biçimde gösterebilir;

- Ham Biçimi (Raw Format): Veriler bir dizi şeklinde 16'lık sayı sisteminde gösterilir. Verinin, hangi özelliğe ait olduğu yanında yazılı olmadığı için verinin neyi ifade ettiği anlaşılabilir. Şekil 4.7'de Xserve'de verilerin ham biçiminin ekran çıktısı görüntülenmektedir.



```
MSN@msn_pc ~
$ xserve -device=com6 -r
[2012/02/06 21:07:56] xdebug: could not open log file /opt/MoteWorks/tools/xserve/bin/logs/xserve_log.txt: No such file or directory
XSERVE 2.0.E: $Id: xserve.c,v 1.8.2.3 2007/02/02 17:45:01 rkapur Exp $
Warning: Converting Windows com6 device to Cygwin device.
Using params: [raw] [server port=9001]
Opening serial device: /dev/ttyS5 @ 57600
[2012/02/06 21:07:57] Serial Source Msg: sync
[2012/02/06 21:07:57] 7E 00 FD 7D 02 A1 00 [7]
[2012/02/06 21:07:58] 7E 00 03 7D 0F 00 00 01 14 00 00 03 F1 81 02 ED 13 FF 04 0
$
[2012/02/06 21:08:02] 7E 00 FD 7D 02 A2 00 [7]
[2012/02/06 21:08:02] 7E 00 0B 7D 25 00 00 ED 13 00 00 33 85 86 00 00 AA 01 F1 0
5 D7 17 2C A6 59 A9 62 B4 A3 B1 66 5F AF 49 B6 FF 00 00 C0 01 C3 01 [42]
[2012/02/06 21:08:03] 7E 00 0B 7D 25 00 00 01 14 00 00 33 85 86 00 00 B0 01 D8 0
5 E9 17 86 A9 98 C0 9D 9B AC B0 57 67 7E 47 B9 FF 00 00 C1 01 C1 01 [42]
[2012/02/06 21:08:07] 7E 00 FD 7D 02 A3 00 [7]
```

Şekil 4.7 : Xserve'de verilerin ham biçimi

- Ayrıştırılmış Biçimi (Parsed Format): Veriler 16'lık sayı sisteminde gösterilir. Verilerin hangi özelliğe ait olduğu yanlarında belirtilmiştir. Şekil 4.8'de Xserve'de verilerin ayrıştırılmış biçimi görüntülenmektedir.

```
WSN@wsn_pc ~
$ xserve -device=com6 -p
[2012/02/06 21:10:18] xdebug: could not open log file /opt/MoteWorks/tools/xserve/bin/logs/xserve_log.txt: No such file or directory
XSERVE 2.0.E: $Id: xserve.c,v 1.8.2.3 2007/02/02 17:45:01 rkapur Exp $
Warning: Converting Windows com6 device to Cygwin device.
Using params: [parsed] [server port=9001]
Opening serial device: /dev/ttyS5 @ 57600
[2012/02/06 21:10:19] Serial Source Msg: sync
[2012/02/06 21:10:19] amtype=0xfd,
[2012/02/06 21:10:20] MTS400 [sensor data converted to engineering units]:
health:         node id = 0x1401
battery:        = 0x1b0 mv
humid:          = 0x5ca%
Temperature:    = 0x17ee degC
IntersemaTemperature: = 0x675f degC
IntersemaPressure: = 0x4779 mbar
Light :         = 0xffb9 lux
X-axis Accel:   = 0x1c1 mg
Y-axis Accel:   = 0x1c2 mg
$
```

Şekil 4.8 : Xserve’de verilerin ayrıştırılmış biçimi

- Dönüştürülmüş Biçimi (Converted Format): Verilerin ham değerlerinden, mühendislik değerlerine uygun bir şekilde dönüştürülmüş halini göstermektedir. Verilerin hangi özelliğe ait olduğu yanlarında belirtilmiştir. Şekil 4.9’da Xserve’de verinin dönüştürülmüş biçimi görüntülenmektedir.

```
WSN@wsn_pc ~
$ xserve -device=com6 -c
[2012/02/06 21:11:27] xdebug: could not open log file /opt/MoteWorks/tools/xserve/bin/logs/xserve_log.txt: No such file or directory
XSERVE 2.0.E: $Id: xserve.c,v 1.8.2.3 2007/02/02 17:45:01 rkapur Exp $
Warning: Converting Windows com6 device to Cygwin device.
Using params: [converted] [server port=9001]
Opening serial device: /dev/ttyS5 @ 57600
[2012/02/06 21:11:32] Serial Source Msg: sync
[2012/02/06 21:11:32] amtype=253,
[2012/02/06 21:11:32] MTS400 [sensor data converted to engineering units]:
health:         node id = 5101
battery:        = 2932 mv
humid:          = 50%
Temperature:    = 21 degC
IntersemaTemperature: = 11.137695 degC
IntersemaPressure: = 984.602661 mbar
Light :         = 72.449997 lux
X-axis Accel:   = -40.000000 mg
Y-axis Accel:   = 20.000000 mg
$
```

Şekil 4.9 : Xserve’de verilerin dönüştürülmüş biçimi

Xserve, XML-RPC web hizmetini kullanarak AD’lerden gelen veri paketlerini varsayılan olarak 9002 nolu portundan XML formatında sunmaktadır. XML formatlı

veri akışını sağlamak için Xserve’de kullanılan parametreler ve anlamları şu şekildedir;

- -xmlr : Ham biçimli (raw format) XML veri çıkışı verir.
- -xmlp : Ayrıştırılmış biçimli (parsed format) XML veri çıkışı verir.
- -xmlc : Dönüştürülmüş biçimli (converted format) veri çıkışı verir.

4.6. Tez Çalışması İçin Kullanılan Xserve Parametreleri

Tez çalışmasında AD’lerden gelen verileri XML formatında uygun bir şekilde alabilmek için Xserve arayüz terminaline yazılan parametreler ve parametrelerin yerine getirdiği görevler aşağıda belirtilmiştir.

Xserve terminaline ulaşmak için Cygwin programı kullanılmaktadır. Cygwin, windows tabanlı işletim sistemlerinde Linux tabanlı kod yazabilmek için geliştirilmiş bir programdır.

Cygwin çalışma ekranına aşağıdaki parametreler yazılarak, bu parametrelerin ekran çıktısı Şekil 4.10’daki gibi görüntülenir.

```
xserve -device=com6 -h -c -xmlc -xmlport=9002
```

Bu parametrelerin yerine getirdiği görevler;

xserve: Xserve terminalini çalıştırır.

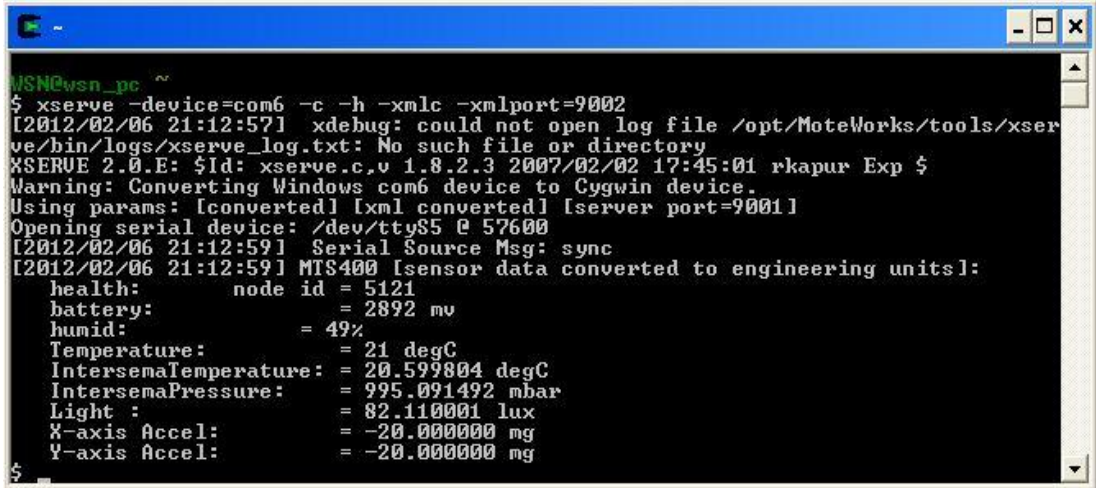
-device=com6: Bilgisayarın hangi sanal portunu kullanarak erişim noktası ile haberleşmesi gerektiğini belirtmektedir. KAA baz istasyonu hem AD’leri programlamak için hem de erişim noktası olarak kullanıldığı için bilgisayara ilk kurulduğunda kendine iki tane (Com_n, Com_{n+1}) seri sanal port istihdam etmektedir. Oluşturulan bu iki seri sanal porttan büyük numaralı olan ile (Com_{n+1}) AD’lere erişim noktası (baz istasyonu) haberleşmesi yapmaktadır, bu portlardan küçük numaralı olan ile de (Com_n) AD’leri programlamak için haberleşmektedir. Kullanılan bilgisayarda, erişim noktası Com6 portundan AD’ler ile haberleşme yapabilmektedir.

-h: Verilerin web sunucusu yoluyla görüntülenmesini sağlar.

-c: Gelen verilerin ekranda dönüştürülmüş biçimde gösterilmesini sağlar.

-xmlc: Dönüştürülmüş biçimde olan veri paketlerinin XML formatında bilgi çıkışını sağlar.

-xmlport=9002: Belirtilen numaralı portta XML sunucusunu başlatmak için kullanılır. Xserve'de varsayılan olarak XML sunucusu port numarası 9002'dir.



```
WSNC@wsn_pc ~
$ xserve -device=com6 -c -h -xmlc -xmlport=9002
[2012/02/06 21:12:57] xdebug: could not open log file /opt/MoteWorks/tools/xserve/bin/logs/xserve_log.txt: No such file or directory
XSERVE 2.0.E: $Id: xserve.c,v 1.8.2.3 2007/02/02 17:45:01 rkapur Exp $
Warning: Converting Windows com6 device to Cygwin device.
Using params: [converted] [xml converted] [server port=9001]
Opening serial device: /dev/ttyS5 @ 57600
[2012/02/06 21:12:59] Serial Source Msg: sync
[2012/02/06 21:12:59] MTS400 [sensor data converted to engineering units]:
  health:      node id = 5121
  battery:    = 2892 mv
  humid:      = 49%
  Temperature: = 21 degC
  IntersemaTemperature: = 20.599804 degC
  IntersemaPressure: = 995.091492 mbar
  Light :     = 82.110001 lux
  X-axis Accel: = -20.000000 mg
  Y-axis Accel: = -20.000000 mg
$
```

Şekil 4.10 : Xserve ekran çıktısı

4.7. Sonuç

Bu bölümde KAA'larda kullanılan ağ topolojilerini verilerek Crossbow firması tarafından KAA'lar için geliştirilen Xmesh örgü ağı protokolü tanıtılmıştır.

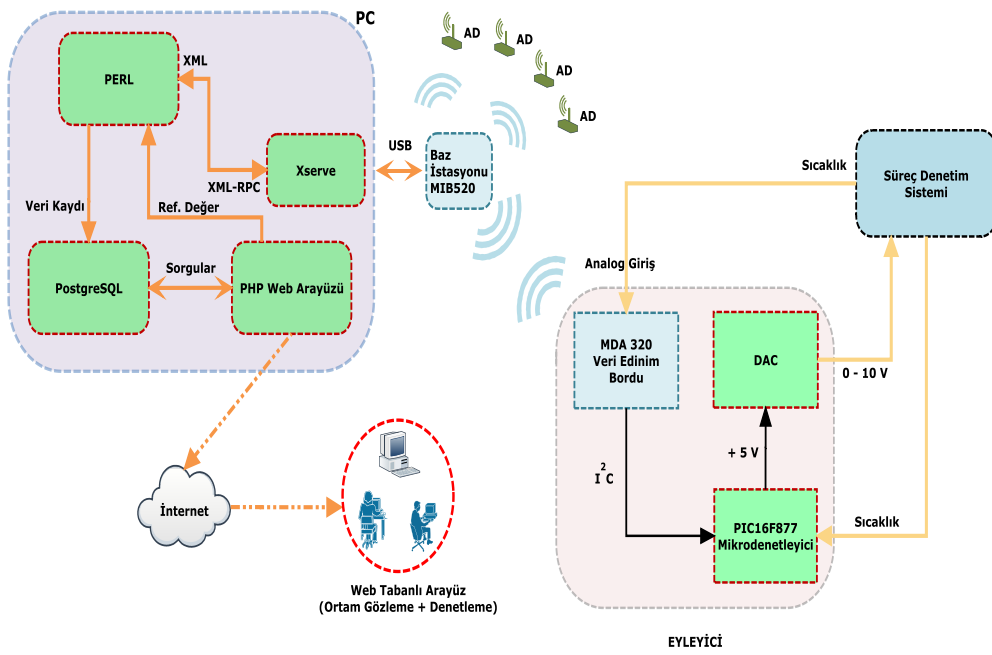
Uygulamada kullanılan AD'ler ve erişim noktasının özellikleri belirtilerek kullanılan cihazların resimleri sunulmuştur.

AD'lerden gelen verileri erişim noktasından bilgisayar ortamına alınmasını sağlayan Xserve ağ geçidi terminal programı tanıtılmıştır. Xserver'de verilerin hangi komutlar kullanılarak XML formatında veri akışı sağlanabildiği örnekler ile gösterilmiştir.

Son kısımda, tez çalışmasında kullanılan Xserve parametreleri ve görevleri açıklanmıştır.

5. GELİŞTİRİLEN WEB TABANLI ARAYÜZ TASARIMI

Tez çalışmasında oluşturulan web tabanlı ortam gözleme arayüzünün blok şeması Şekil 5.1’de gösterilmektedir. Şekil 5.2’de ise kablosuz algılayıcı ağ için web tabanlı ortam gözleme uygulamasının çalışması için kurulan sistem gösterilmektedir.



Şekil 5.1 : Sistemin blok şeması



Şekil 5.2 : Web tabanlı ortam gözleme uygulaması için kurulan sistem

Şekil 5.1’de AD’ler fiziksel ortama ait sıcaklık, nem, basınç, ışık şiddeti ve ivme bilgilerini ölçmektedir. Ölçtükları bu verileri kablosuz ortamı kullanarak MIB520 baz istasyonuna iletmektedirler. MIB520 baz istasyonu, bilgisayar ile USB seri portu üzerinden haberleşmektedir. MIB520 baz istasyonu AD’lere ait verileri Xserve terminaline ulaştırmaktadır. Xserve terminali, AD’lere ait veri paketlerini XML-RPC web hizmeti üzerinden XML belgesi formatında sunmaktadır. Xserve’ün sunduğu AD’lere ait verileri alabilmek için Perl programlama dili kullanılarak bir soket programı oluşturulmuştur. Oluşturulan soket programı ile web hizmeti (XML-RPC) haberleştirilerek gelen veriler XML belgesi formatında alınmıştır. XML formatındaki veriler, elementlerinden ayrıştırılarak (XML parsing) AD’lere ait kimlik bilgisi, fiziksel ortama ait sıcaklık, ışık şiddeti, nem, basınç, ivme ve kalibrasyon bilgileri mühendislik verilerine dönüştürülmüş bir şekilde alınmaktadır. Alınan veriler, PostgreSQL “wsn” veri tabanındaki “mts400result” tablosuna kaydedilmektedir. Php web arayüzü ile veri tabanından sorgulama yapılarak veri tabanına kaydedilen bu veriler web sayfalarına aktarılmaktadır. PostgreSQL veri tabanından veriler alınırken Ajax teknolojisi kullanılmaktadır. Web sayfasında, alınan veriler anlık olarak gösterilmekte, AD’lerin çevrim durumları belirtilmekte, verilere ait grafikler çizdirilmekte ve AD’lerin Xmesh mimarisindeki bir birilerine göre olan konumları grafiksel olarak sunulmaktadır. Wamp server aracılığı ile geliştirilen web tabanlı arayüz internet ortamına sunularak, AD’lerden gelen verilere uzaktan erişilip, gözlenmesi sağlanmaktadır.

Geliştirilen soket programıyla, ayrıştırılmış veriler, kullanılan bilgisayarın yerel veri tabanına kaydedilmesinin yanı sıra internet üzerinden uzaktaki bir sunucuda oluşturulan veri tabanına da anlık olarak kaydedilmektedir. AD’lerden gelen veriler ayrıştırıldıktan sonra uzaktaki bir veri tabanı sunucusunda tutulabileceği de gösterilmiştir. Bu durum, verilerin saklanması ve veri güvenliğini artırıcı bir önlem olarak sunulmuştur.

Geliştirilen web tabanlı arayüze [24]’daki çalışmada kullanılan süreç denetim sistemi modül olarak eklenmiştir. Eklenen modül ile geliştirilen web arayüzü uzaktan sistem denetleme ve denetlenen sistemin gözlenmesini sağlamaktadır.

Geliştirilen web arayüzden, web tabanlı uzaktan denetim gerçekleştirilmektedir. Perl ile soket programlama oluşturulmakta, oluşturulan bu soket programına, web arayüzünden referans sıcaklık değeri parametre olarak gönderilmektedir. Soket programı, parametreleri XML formatına dönüştürüp, XML-RPC web hizmetini kullanarak Xserve terminaline ulaştırmaktadır. Xserve terminali, MIB520 baz istasyonu üzerinden MDA320 veri edinim borduna referans sıcaklık değerini göndermektedir. MDA320 veri edinim bordu, aldığı referans sıcaklık bilgisini I²C çıkışları üzerinden haberleştiği mikrodenetleyiciye göndermektedir. Mikrodenetleyici, süreç denetim sisteminden gelen sıcaklık bilgisi ile referans sıcaklık bilgisini değerlendirerek bir denetim bilgisi üretmektedir. Mikrodenetleyici ürettiği denetim bilgisini dijital-analog çeviriciye göndermektedir. Dijital-analog çevirici gelen denetim bilgisini analog sinyale dönüştürerek süreç denetim sistemine iletmektedir. Süreç denetim sistemi, gelen denetim sinyaline göre çalışmasını değiştirmektedir. MDA320 veri edinim bordu, süreç denetim sisteminin sıcaklık bilgisini alarak kablosuz ortam üzerinden MIB520 baz istasyonuna iletmektedir. MDA320 veri edinim bordundan gelen sıcaklık verisi, Xserve terminali üzerinden XML formatında sunulan verileri Perl ile oluşturulmuş soket programı tarafından ayrıştırarak, PostgreSQL “wsn” veri tabanındaki “mda320result” tablosuna kaydetmektedir. Web arayüzünden veri tabanındaki verilere ulaşılarak süreç denetim sistemine ait sıcaklık grafiği ve sistemin kontrol bilgisi grafiği anlık olarak uzaktan gözlemlenebilmektedir. Web arayüzü aracılığıyla uzaktan bir sistemin denetimi yapılmış ve sistemin çalışması uzaktan gözlemlenebilmiştir.

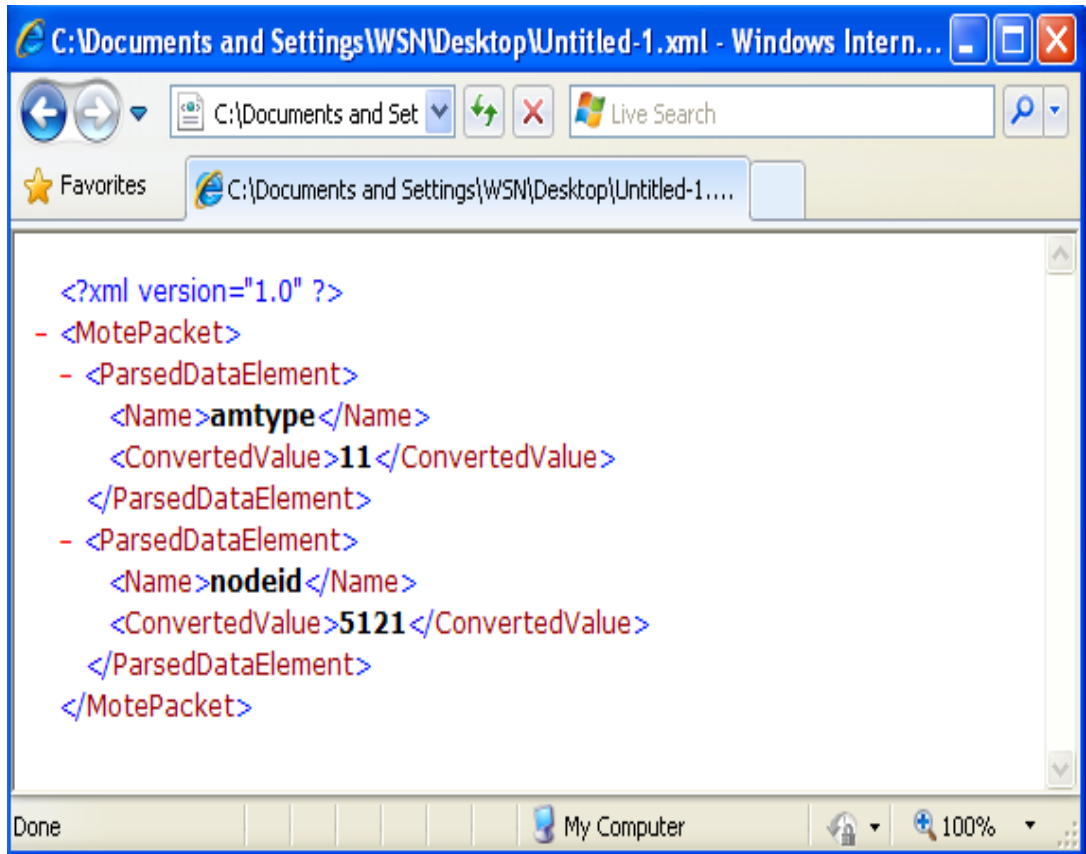
Veri tabanına kaydedilen gözleme ve denetleme bilgileri istatistiksel amaçlı olarak ileriye dönük çalışmalarda kullanılabilir bir yapıda tutulmaktadır. Veri tabanında tutulan bu verileri kullanarak sistemi geliştirmek için analiz yapabilme imkanı sunmaktadır.

5.1. Perl Soket Programı

Soketler, bilgisayar ortamında farklı yapıların birbiri ile haberleşmesini sağlayan yöntemlerdir. Soketler, bilgisayar sistemindeki dosyalar gibi davranırlar. Bilgisayar sisteminde bir dosyaya erişmek, okumak ve yazmak için yapılan işlemler, soket

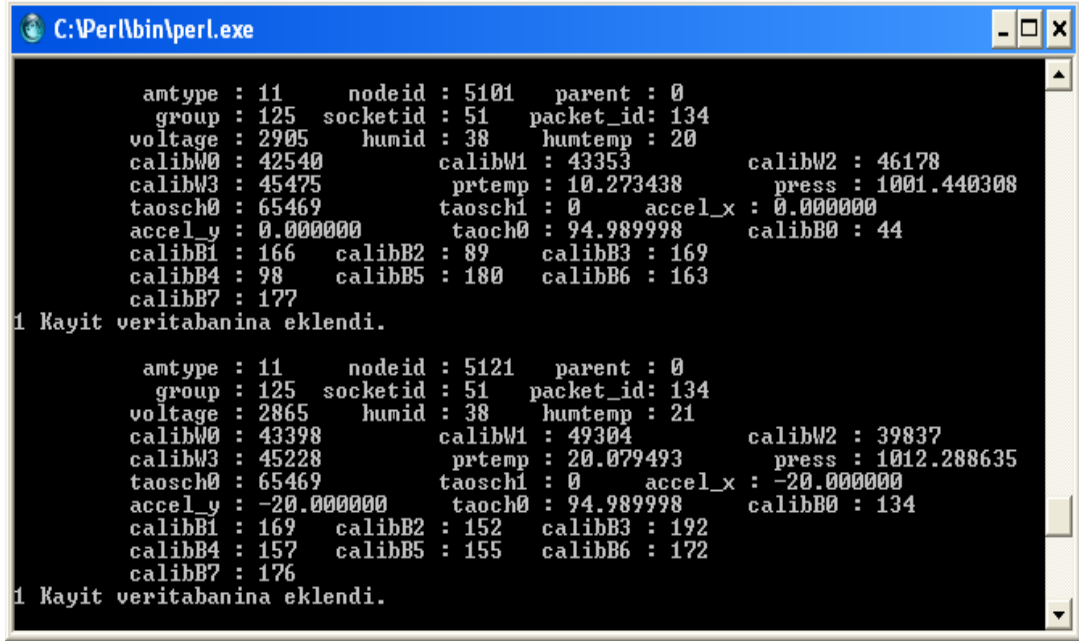
programlama içinde yapılabilmektedir. Soket programlama, istemci-sunucu yapısından oluşmaktadır. İstemci, bir sunucuya bağlanır ve bu sunucudan bilgi talebinde bulunur. Sunucu ise istemcinin bilgi talebini alarak, gerekli işlemleri yerine getirir ve işlemin sonucunu istemciye gönderir. İstemcinin talebi ne zaman gerçekleşeceği belli olmadığından, sunucu bilgisayar sistemindeki bir portu sürekli dinler ve gelen taleplere göre bu port üzerinden hizmet verir.

Oluşturulan soket programıyla, bir XML sunucusundan gelen veriler 9002 nolu port dinlenerek elde edilmektedir. XML formatındaki verilere (Şekil 5.3) soket programlamayla ulaşılarak elementlerinden ayrıştırılmıştır. Ayrıştırılmış veriler “mts400result” tablosuna insert sorgusu ile eklenmiştir. Şekil 5.4’de Perl dilinde oluşturulmuş soket programının konsol çıktısı görülmektedir.



```
<?xml version="1.0" ?>
- <MotePacket>
- <ParsedDataElement>
  <Name>amtype</Name>
  <ConvertedValue>11</ConvertedValue>
</ParsedDataElement>
- <ParsedDataElement>
  <Name>nodeid</Name>
  <ConvertedValue>5121</ConvertedValue>
</ParsedDataElement>
</MotePacket>
```

Şekil 5.3 : AD'den gelen XML formatındaki örnek veri



```
C:\Perl\bin\perl.exe

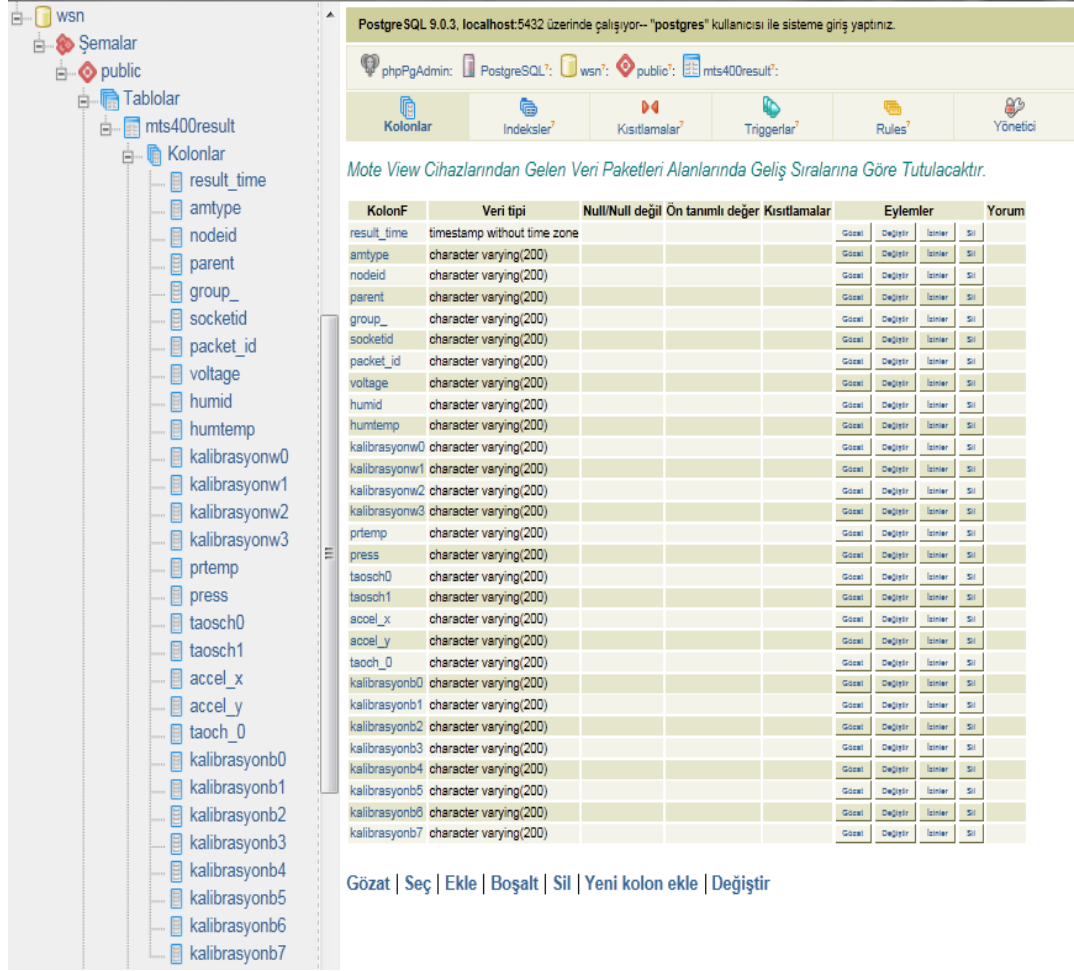
  amtype : 11      nodeid : 5101   parent : 0
  group  : 125    socketid : 51   packet_id: 134
  voltage : 2905   humid  : 38    humtemp : 20
  calibW0 : 42540  calibW1 : 43353   calibW2 : 46178
  calibW3 : 45475  prtemp : 10.273438   press : 1001.440308
  taosch0 : 65469  taosch1 : 0      accel_x : 0.000000
  accel_y : 0.000000  taoch0 : 94.989998   calibB0 : 44
  calibB1 : 166   calibB2 : 89   calibB3 : 169
  calibB4 : 98   calibB5 : 180  calibB6 : 163
  calibB7 : 177
1 Kayit veritabanina eklendi.

  amtype : 11      nodeid : 5121   parent : 0
  group  : 125    socketid : 51   packet_id: 134
  voltage : 2865   humid  : 38    humtemp : 21
  calibW0 : 43398  calibW1 : 49304   calibW2 : 39837
  calibW3 : 45228  prtemp : 20.079493   press : 1012.288635
  taosch0 : 65469  taosch1 : 0      accel_x : -20.000000
  accel_y : -20.000000  taoch0 : 94.989998   calibB0 : 134
  calibB1 : 169   calibB2 : 152  calibB3 : 192
  calibB4 : 157   calibB5 : 155  calibB6 : 172
  calibB7 : 176
1 Kayit veritabanina eklendi.
```

Şekil 5.4 : Perl’de oluşturulan socket programının konsol çıktısı

5.2. PostgreSQL “wsn” Veritabanı

PostgreSQL veri tabanı sunucusunda AD’lerden gelen veriler “wsn” veri tabanındaki “mts400result” tablosuna kaydedilmektedir. “mts400result” tablosuna ait alanlar, AD’lerden gelen verinin, geliş sırasına uygun olarak oluşturulmuştur. Veri tabanında tutulan veriler aynı anda başka bir amaç için farklı bir uygulama arayüzü tarafından kullanılabilir. Şekil 5.5’de “wsn” veri tabanı yapısının resmi gösterilmektedir.



Şekil 5.5 : “wsn” veri tabanı yapısı

5.3. Geliştirilen Web Tabanlı Arayüzün Tanıtımı

Şekil 5.6’da KAA web tabanlı arayüzün ana sayfasına ait görüntüsü gösterilmektedir. Şekil üzerindeki numaralandırılan alanların görevleri şu şekildedir;

1. Bu kısımda, web arayüzünün menüsü bulunmaktadır. Bu menü sayesinde web arayüzündeki diğer sayfalara bağlantılar yapılarak web arayüzünün sunduğu diğer özelliklere kolay ve hızlı bir şekilde ulaşılmasını sağlamaktadır.
2. Bu kısımda, veri tabanında bulunan AD’lerin düğüm numarası (ID) ve düğümün durumu (çevrim içi – çevrim dışı) gösterilmektedir. Veri tabanına bir AD veri göndermiş ise ilgili AD’nin düğüm numarası ve durumu bu kısımda gösterilir. Düğüm, son bir dakika içinde veri göndermiş ise yeşil, göndermemiş ise gri

renkli olarak gösterilmektedir. Ayrıca bu alan her beş saniyede bir kendini dinamik olarak güncellemektedir. Mavi renkli yenileme butonu ise istenildiği zaman bu alanı el ile (tıklayarak) güncellememizi sağlamaktadır.

3. Bu kısımda, AD numaralarının üstüne fare ile gelindiğinde ilgili AD'nin en son gönderdiği veriye ait bilgiler, açılır bir pencerede dinamik olarak gösterilmektedir.
4. Bu kısım, 2 numaralı alandaki AD durumlarını gösteren sembollerin ne anlama geldiğini açıklayan bir gösterge tablosu olarak tasarlanmıştır.
5. Bu kısım, veri tabanına kayıtlı olan AD'lerden gelen ortama ait fiziksel büyüklük verilerinin bir tablo şeklinde gösterildiği kısımdır. En üstte, en son gelen verinin bilgileri gösterilmektedir. Her bir satırdaki verinin hangi AD'e ait olduğu belirtilmiştir.



Şekil 5.6 : Web arayüzünün ana sayfasının görünümü

Şekil 5.7’de KAA web tabanlı arayüzün grafik çizdirme ekranı gösterilmektedir. Grafik seçim ekranı, şekil üzerinde 4 numaralı olarak gösterilmiştir. “Veri Türü” açılır kutusunda, AD’lerden gelen ortama ait fiziksel büyüklükler listelenmektedir. Bunlar;

- Işık: Ortama ait ışık şiddetini gösterir. Birimi lüks (lux)'tür.
- Basınç: Ortama ait basınç değerini gösterir. Birimi milibar (mbar)'dır.
- Batarya: AD'ye ait pilin gerilim değerini gösterir. Birimi milivolt (mV)'tur.
- Nem: Ortama ait nem oranını gösterir. Yüzde olarak (%) ifade edilir.
- Nem Sıcaklığı: Ortama ait nem sıcaklığı değerini gösterir. Birimi °C'dir.
- Sıcaklık: Ortama ait sıcaklık değerini gösterir. Birimi °C'dir.
- Accel x: AD'nin x eksenine göre ivmesini gösterir. Birimi mg'dir.
- Accel y: AD'nin y eksenine göre ivmesini gösterir. Birimi mg'dir.

Şekil 5.7'de, grafik seçim ekranının “Grafik Türü” açılır kutusunda çizdirilmek istenilen grafik çeşitleri bulunmaktadır. Bunlar; Bar Grafik, Line Grafik, Spline Grafik, Scatter, Area Grafik ve Area Spline Grafik'tir.

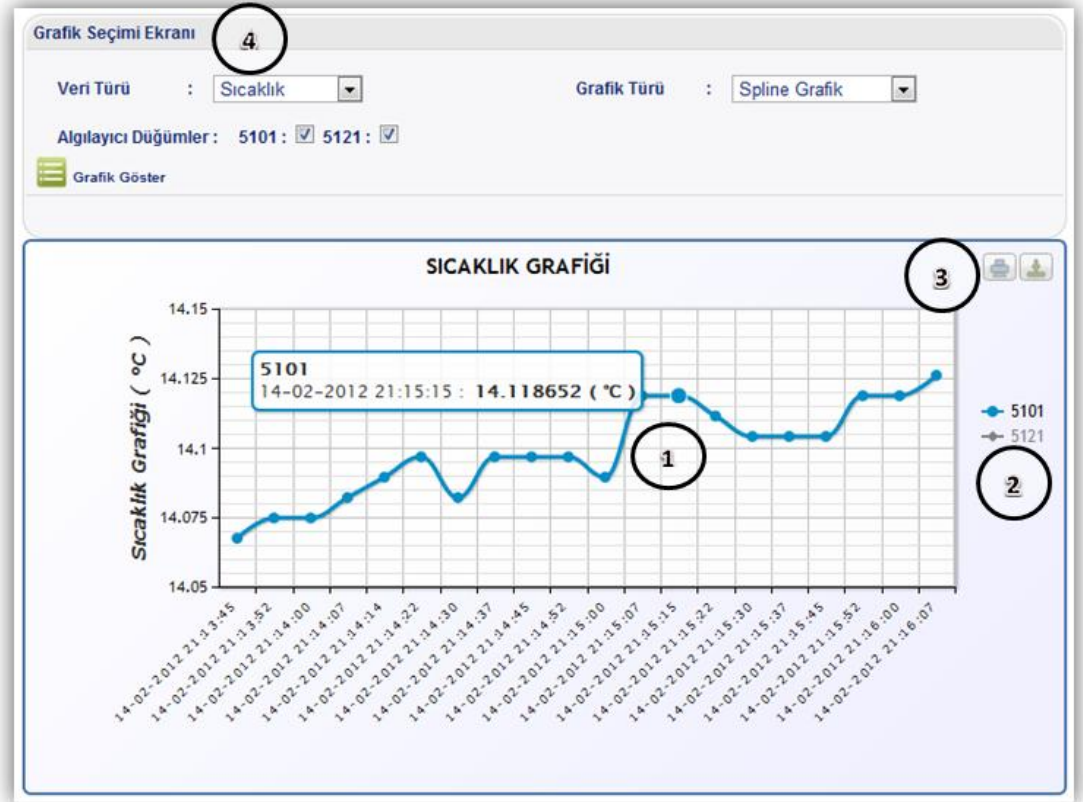
Şekil 5.7'de, grafik seçim ekranının “Algılayıcı Düğümler” kısmında seçim kutuları bulunmaktadır. Veri tabanında verileri bulunan AD'lerin kimlik numaraları (ID) yanlarında yazılıdır. Grafiği çizdirilmek istenen düğüm veya düğümler seçilip, “Grafik Göster” butonuna tıklanarak grafik çizdirilmektedir.

Şekil 5.7'de görülen numaralı alanlar kullanıcıların, çizdirilen grafik üzerinde bazı işlemler yapabileceğini göstermektedir. Bunlar;

1. Bu kısımda, çizdirilen grafik üzerindeki her hangi bir noktanın üzerine fare ile gelindiğinde bir pencere açılmakta ve bu pencere içinde verinin hangi AD'ye ait olduğu, noktanın x ve y eksenini değerleri ve fiziksel olarak birimi gösterilmektedir. Ayrıca çizdirilen grafik üzerinde her hangi bir alana fare sürüklenip seçme işlemi yapılarak grafiğin odaklanma (zoom) özelliği kullanılmış olmaktadır.

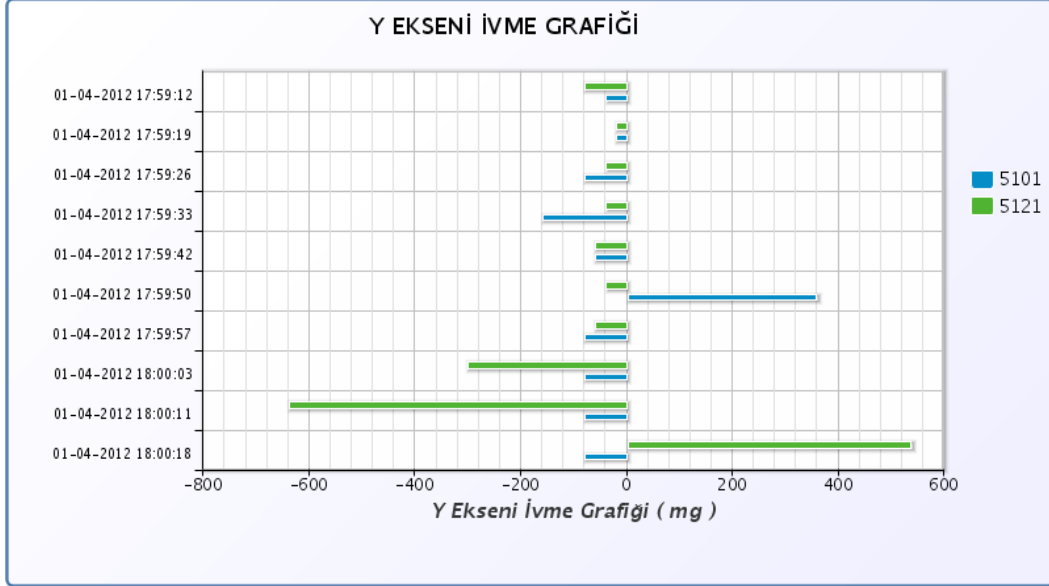
2. Bu kısımda, grafikte birden fazla AD'ye ait veriler çizdirilmektedir. Bu yüzden buradaki verilerin hangi AD'ye ait olduğunu belirtmek için çizdirilen grafiklerde renklendirme yapılmıştır. Grafiği düzgün görebilmek ve doğru yorumlamak için renklerine göre AD'lerin numaraları (ID) gösterilmektedir. AD numaraları (ID) üzerine tıklanarak ilgili AD'ye ait grafik kapatılıp, açılabilir. Bu sayede kullanıcıların çizdirdiği grafiği kullanma esnekliği artırılmış olmaktadır.
3. Bu kısımda, çizdirilen grafiğin çeşitli biçimlerde dışarı alınması gerekebilir. Buradaki simgelere tıklanarak yazıcıdan çıktı alınabilmekte veya bilgisayar ortamında grafik; png, jpeg, pdf ve svg formatında kaydedilebilmektedir.

Grafik menüsünün altında, "Anlık Grafikler" ekranı bulunmaktadır. Bu ekranda yukarıdaki özelliklere ek olarak, grafiğe gelen AD verileri anlık olarak çizdirilmektedir. Bu ekranda grafik, anlık bir şekilde yeniden çizdirilerek, AD verilerinin canlı olarak grafik analizi yapılabilmesi sağlanmaktadır.

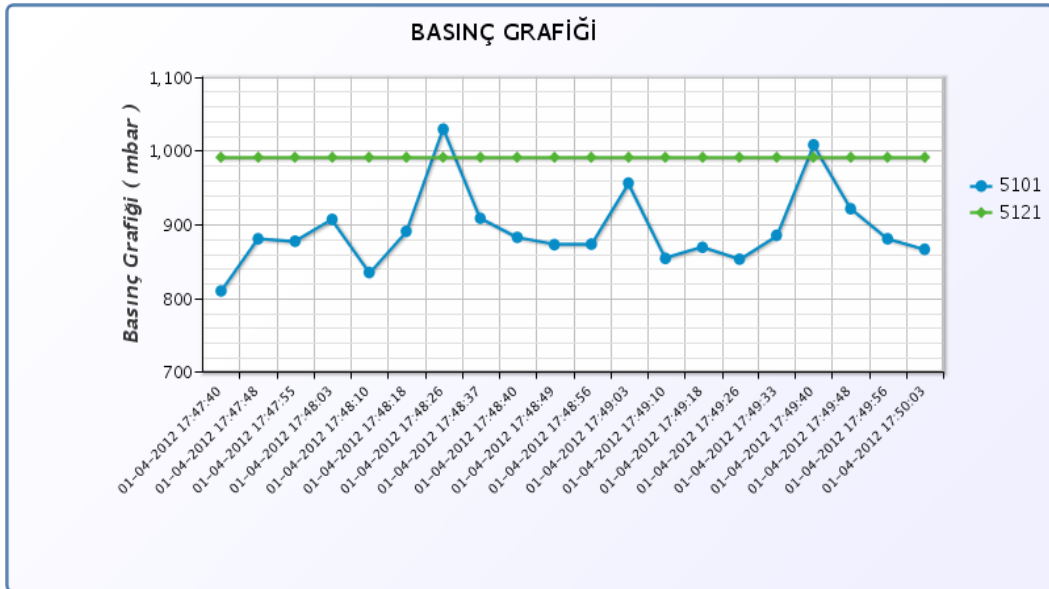


Şekil 5.7 : Web arayüzün grafik çizdirme ekranı

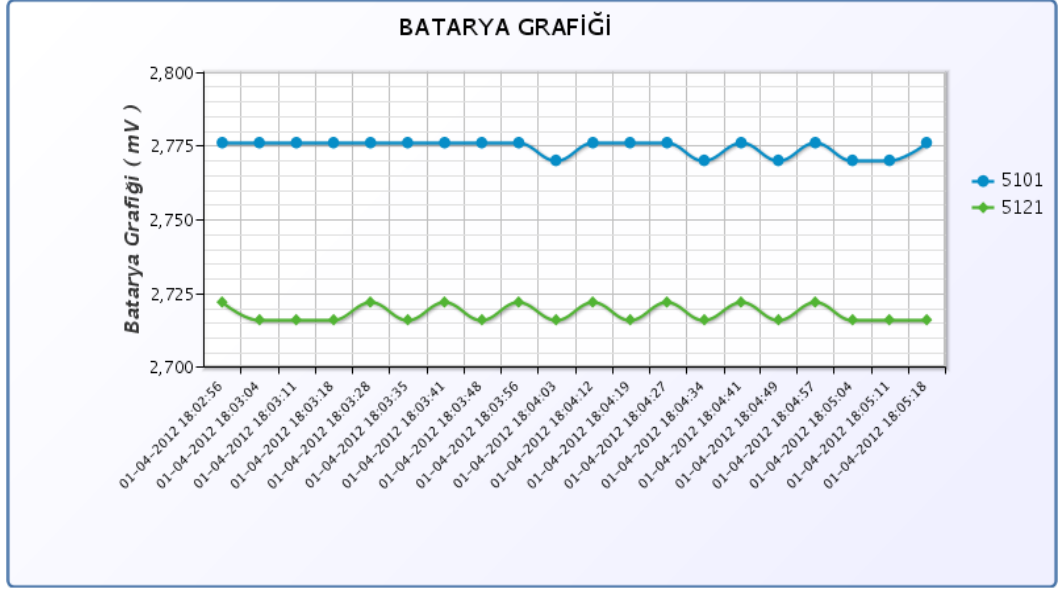
Aşağıdaki grafikler, web arayüzünde grafik çizdirme ekranında çizdirilebilen grafik çeşitlerini göstermektedir.



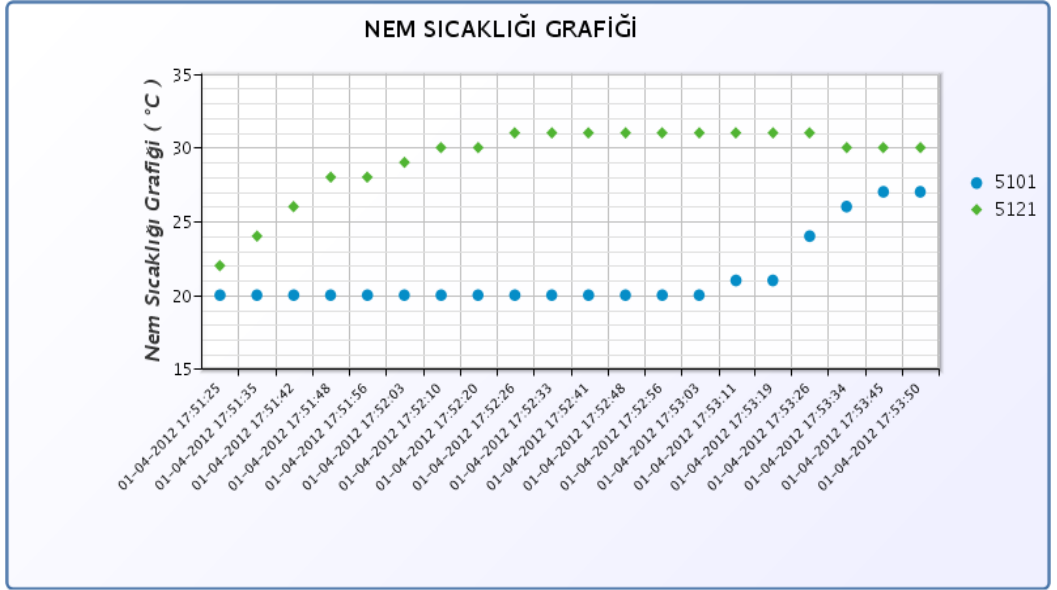
Şekil 5.8 : İki AD'ye ait y eksenî ivme grafiğinin zamana göre değişimi (bar)



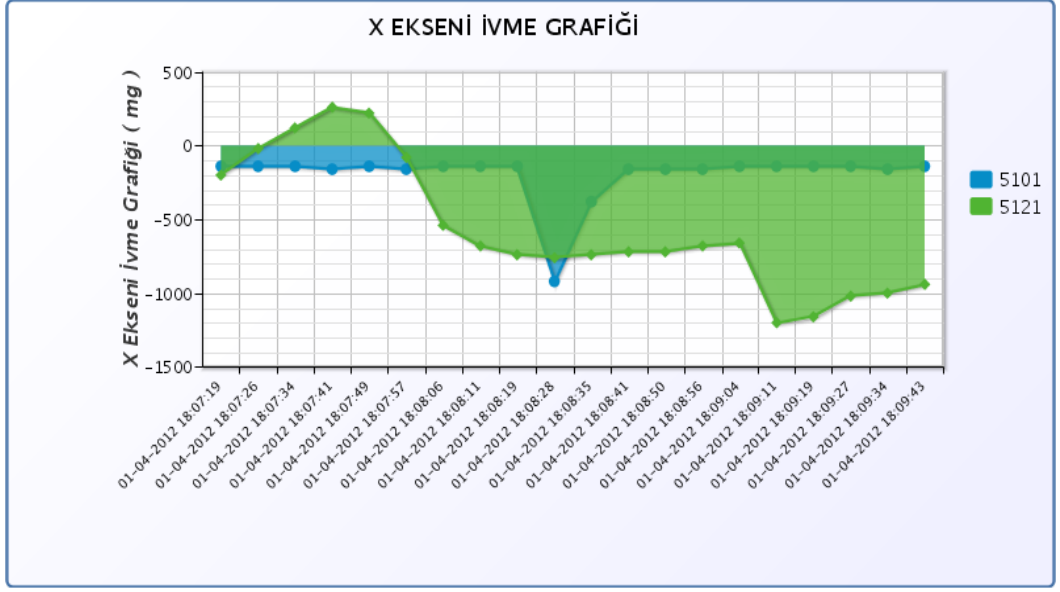
Şekil 5.9 : İki AD'ye ait basınç grafiğinin zamana göre değişimi (line)



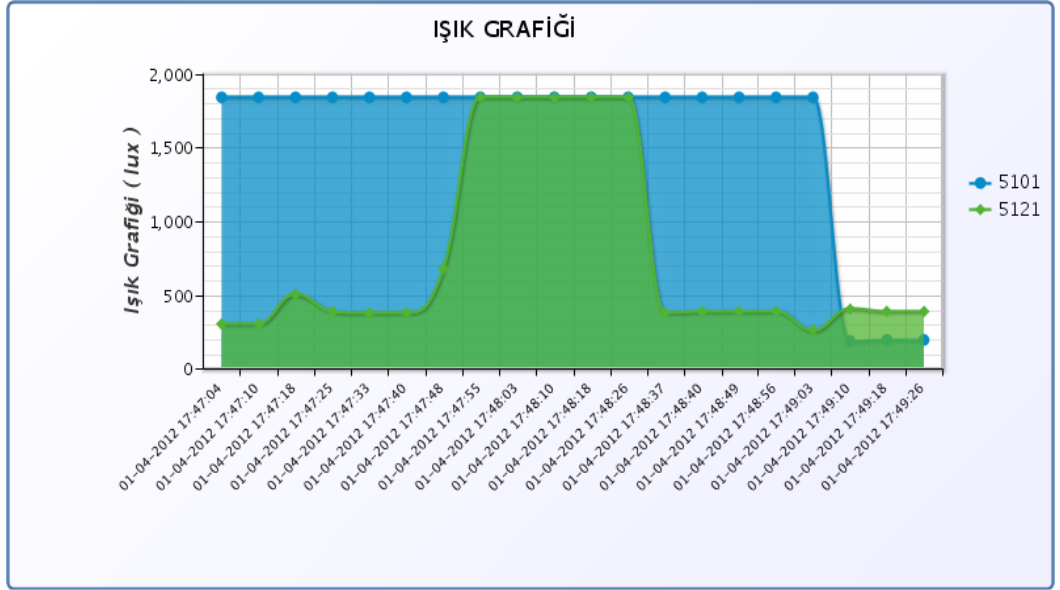
Şekil 5.10 : İki AD'ye ait batarya grafiğinin zamana göre değişimi (spline)



Şekil 5.11 : İki AD'ye ait nem sıcaklığının zaman göre değişimi (scatter)



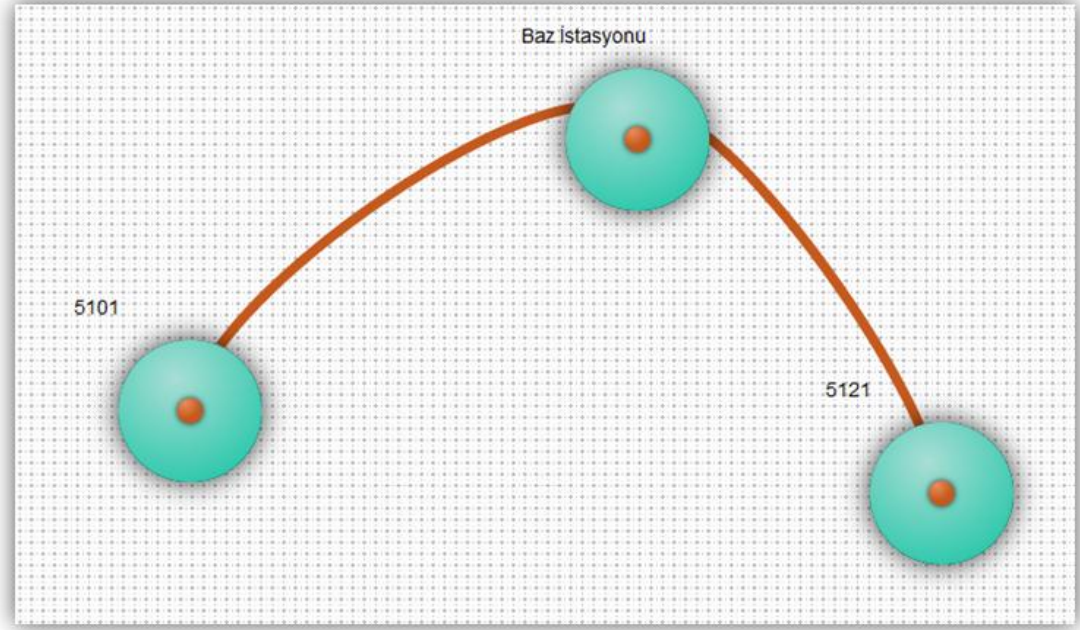
Şekil 5.12 : İki AD'ye ait x eksenİ İvme grafiđi zamana göre deđiřimi (area line)



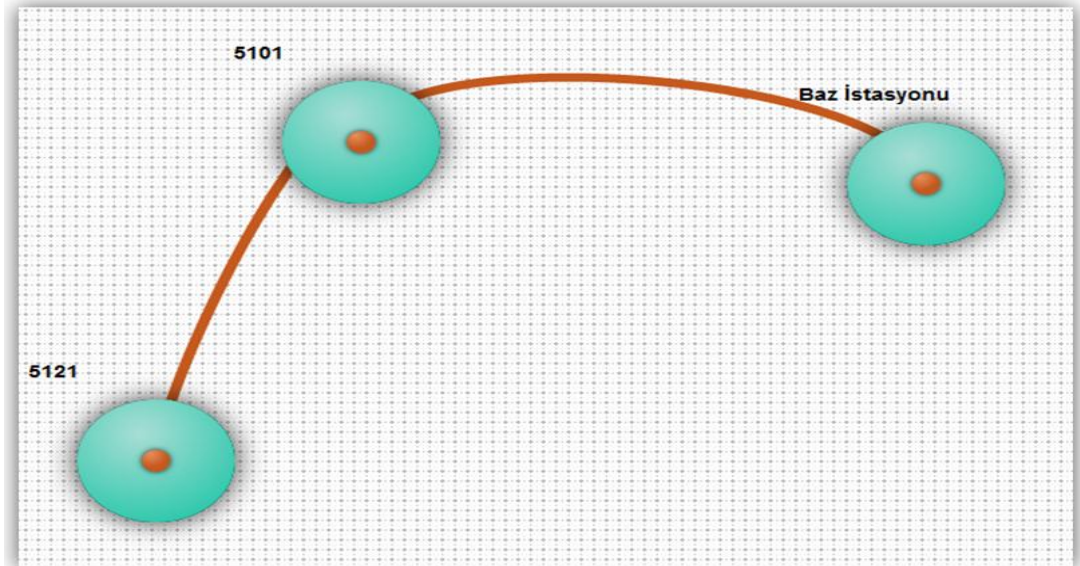
Şekil 5.13 : İki AD'ye ait ıřık grafiđinin zamana göre deđiřimi (area spline)

Şekil 5.14'de ve Şekil 5.15'da tasarlanan web tabanlı arayüzün ađ topolojisi ekranları gösterilmektedir. Bu ekranlar AD'lerin birbiriyle ve baz istasyonu ile nasıl bađlı olduklarını göstermek amacıyla tasarlanmıřlardır. İlgili AD'nin numarası (ID), ilgili yuvarlak řeklin üzerine yazılmaktadır. Bu ekranlar, Xmesh mimarisine göre KAA'nın ađ topolojisini göstermektedirler.

Şekil 5.14’de AD’ler, yıldız topolojisine göre haberleşmektedir. AD’ler, baz istasyonunun iletişim mesafesi içinde bulduklarından dolayı, baz istasyonuna doğrudan bağlı oldukları şekilde gösterilmektedir.



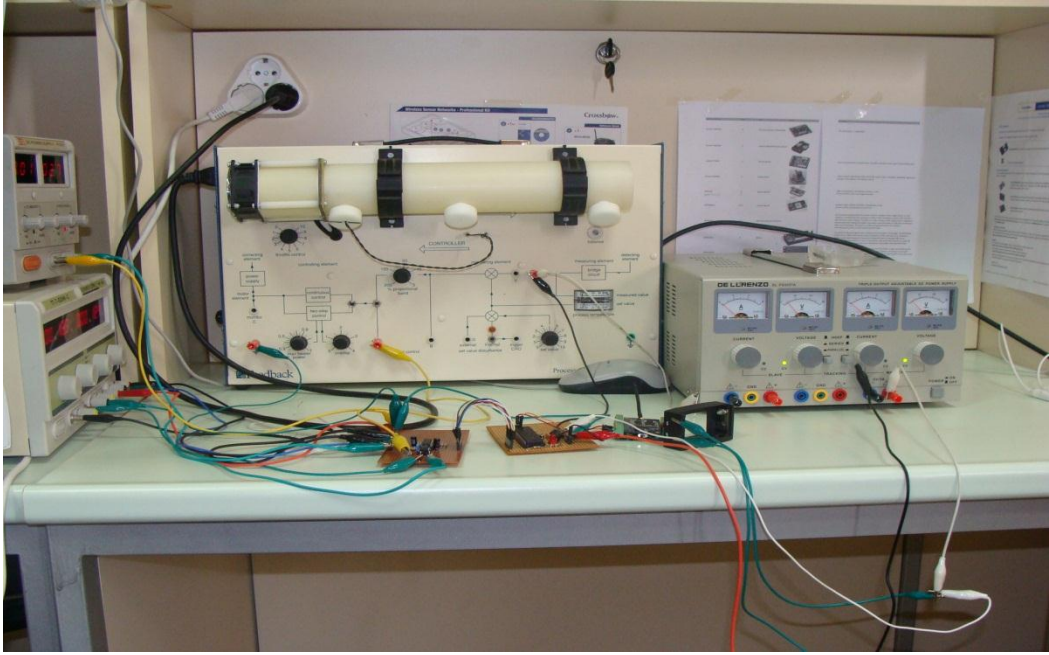
Şekil 5.14 : Web arayüzü KAA topoloji ekranı (doğrudan iletişim)



Şekil 5.15 : Web arayüzü KAA topoloji ekranı (erişim noktalı iletişim)

Şekil 5.15’da AD’ler, örgü topolojisine göre haberleşme yapmaktadırlar. Şekildeki 5121 nolu AD, baz istasyonunun iletişim mesafesi içinde bulunmamaktadır. 5121 nolu AD, baz istasyonu ile 5101 nolu AD üzerinden haberleşme yapmaktadır. 5101 nolu AD, baz istasyonunun iletişim mesafesi içinde bulunmaktadır. Böylece, hem kendi verilerini hem de 5121 nolu AD’nin verilerini baz istasyonuna iletebilmektedir.

5.4. Web Tabanlı Arayüz İle Uzaktan Denetim Uygulaması

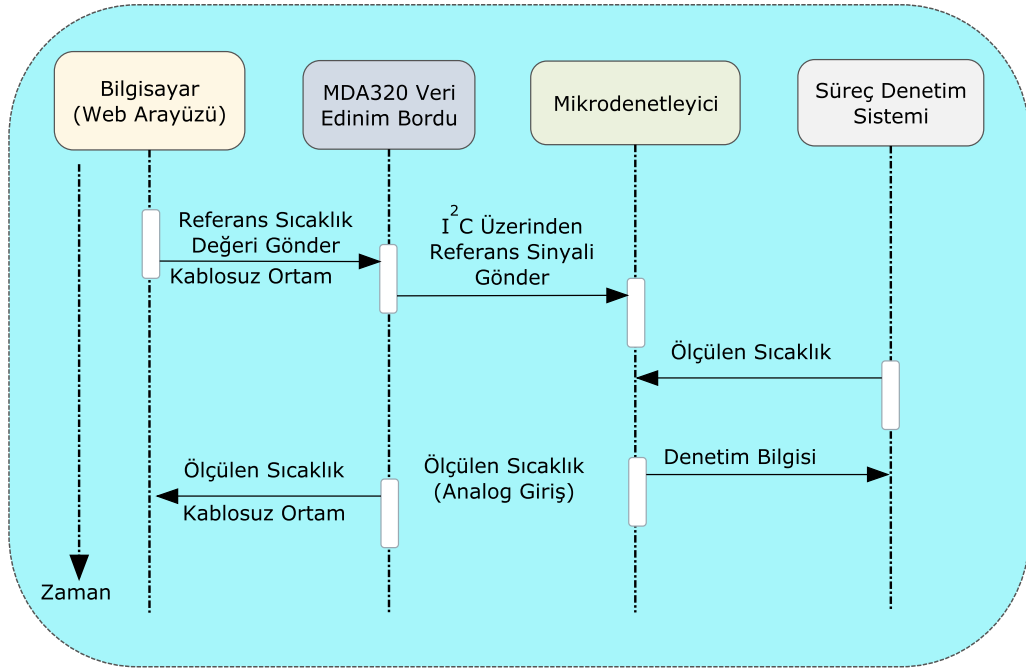


Şekil 5.16 : Denetleme sistemi için kurulan düzenek

Şekil 5.16’da web arayüzü kullanılarak uzaktan denetlenen süreç denetim sistemi görülmektedir. Deney düzeneği; süreç denetim sistemi, MDA320 veri edinim bordu, mikrodenetleyici ve dijital-analog çeviriciden oluşmaktadır.

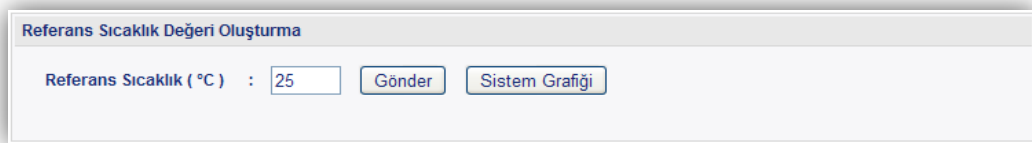
Şekil 5.17’de web arayüzün ile uzaktan denetlenen sistemin zaman akış diyagramı gösterilmektedir. Web tabanlı arayüzden sıcaklık referans değeri kablosuz ortam kullanılarak MDA320 veri edinim borduna gönderilmektedir. MDA320 veri edinim bordu I²C çıkışları üzerinden mikrodenetleyiciye referans sıcaklık değerini göndermektedir. Mikrodenetleyici süreç denetim sisteminden ölçülen sıcaklık bilgisini alır ve referans sıcaklık değeri ile değerlendirme işlemine tabi tutarak bir

denetim bilgisi üretir. Mikrodenetleyici çıkışlarına bağlı olan süreç denetim sistemine denetim sinyalini gönderir ve süreç denetim sistemi mikrodenetleyiciden gelen denetim sinyalini alarak çalışmasını bu sinyale göre değiştirir. Süreç denetim sisteminin sıcaklık bilgisi çıkışı MDA320 veri edinim bordunun analog girişlerine bağlıdır ve bu yol ile MDA320 süreç denetim sisteminin sıcaklık bilgisini alarak kablosuz ortam üzerinden baz istasyona iletmektedir. Baz istasyonu gelen sıcaklık bilgisini bilgisayar ortamına aktarır ve bilgisayar ortamında, gelen sıcaklık bilgisi işlenerek süreç denetim sisteminin sıcaklık ve durum bilgisi grafiği oluşturulur. Geliştirilen web tabanlı arayüz ile sistemin hem uzaktan denetimi hem de uzaktan gözlenmesi web ortamı üzerinden sağlanmıştır.



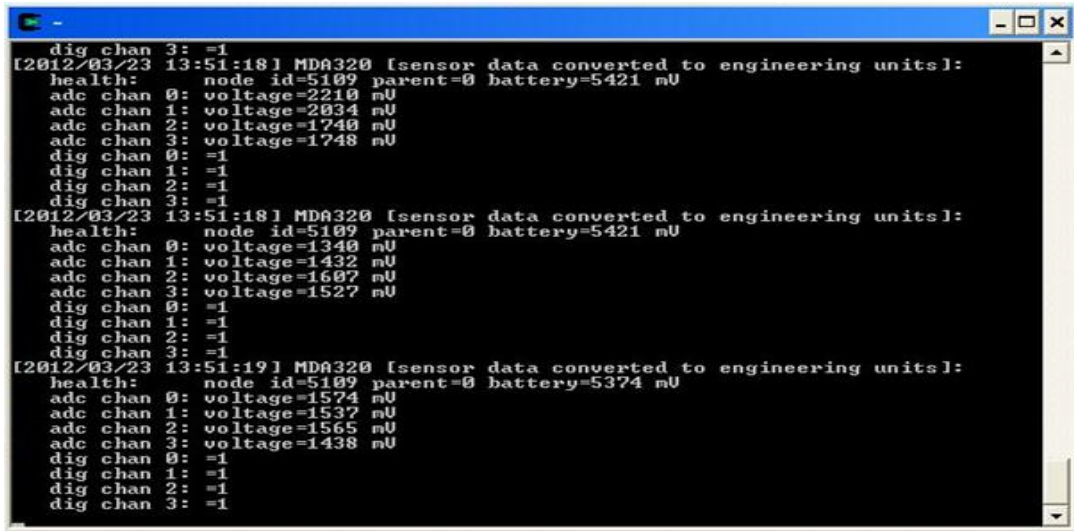
Şekil 5.17 : Denetleme sisteminin zaman akış diyagramı [24]

Şekil 5.18’de KAA web tabanlı arayüzün uzaktan denetim ekranı gösterilmektedir.



Şekil 5.18 : Referans sıcaklığı gönderme ekranı

Bu ekran; sistemin, denetlenmek istendiği referans sıcaklık değerinin MDA320 veri edinim borduna gönderilmesini sağlamaktadır. Sistem 25 °C’de denetlenmek istenirse, gönderilmesi gereken referans sıcaklık değeri Şekil 5.18’deki gibi ayarlanmalıdır. Gönder butonuna basıldıktan sonra bu referans sıcaklık değeri Php web sayfasından, Perl dilinde yazılmış olan soket programına parametre olarak gönderilmektedir. Oluşturulan soket programı, XML-RPC yöntemi ile MDA320 veri edinim borduna baz istasyonu üzerinden parametre göndermektedir.



```
dig chan 3: =1
[2012/03/23 13:51:18] MDA320 [sensor data converted to engineering units]:
health: node id=5109 parent=0 battery=5421 mU
adc chan 0: voltage=2210 mU
adc chan 1: voltage=2034 mU
adc chan 2: voltage=1740 mU
adc chan 3: voltage=1748 mU
dig chan 0: =1
dig chan 1: =1
dig chan 2: =1
dig chan 3: =1
[2012/03/23 13:51:18] MDA320 [sensor data converted to engineering units]:
health: node id=5109 parent=0 battery=5421 mU
adc chan 0: voltage=1340 mU
adc chan 1: voltage=1432 mU
adc chan 2: voltage=1607 mU
adc chan 3: voltage=1527 mU
dig chan 0: =1
dig chan 1: =1
dig chan 2: =1
dig chan 3: =1
[2012/03/23 13:51:19] MDA320 [sensor data converted to engineering units]:
health: node id=5109 parent=0 battery=5374 mU
adc chan 0: voltage=1574 mU
adc chan 1: voltage=1537 mU
adc chan 2: voltage=1565 mU
adc chan 3: voltage=1438 mU
dig chan 0: =1
dig chan 1: =1
dig chan 2: =1
dig chan 3: =1
```

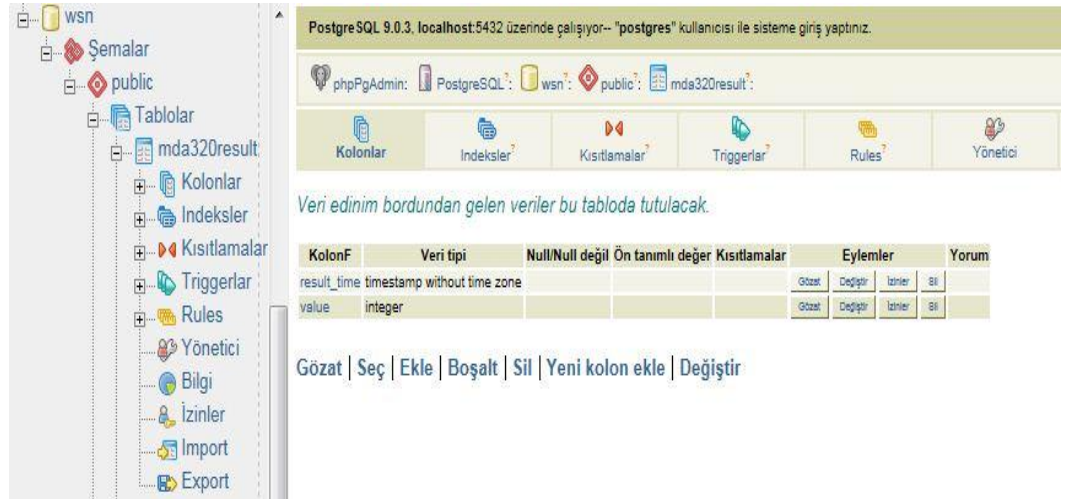
Şekil 5.19 : Xserve’de MDA320 veri edinim bordudan gelen veriler

Şekil 5.19’de veri edinim borduyla baz istasyonunun iletişime geçtikten sonra veri edinim bordundan gelen veri akışı gösterilmektedir.

MDA320 veri edinim bordundan sisteme ait veriler kablosuz ortam aracılığıyla baz istasyonuna gönderilmektedir. Baz istasyonuna gelen bu veriler XML formatında yayımlanmaktadır. PERL dili kullanılarak oluşturulan soket programıyla gelen veriler XML formatından ayrıştırılarak, PostgreSQL veri tabanındaki “mda320result” tablosuna kaydedilmektedir. Şekil 5.20’de veri edinim bordundan gelen verilerin veri tabanına kaydedilmesi amacıyla oluşturulan soket programının ekran çıktısı gösterilmektedir. Şekil 5.21’de “mda320result” tablosunun veri tabanı yapısı gösterilmektedir.

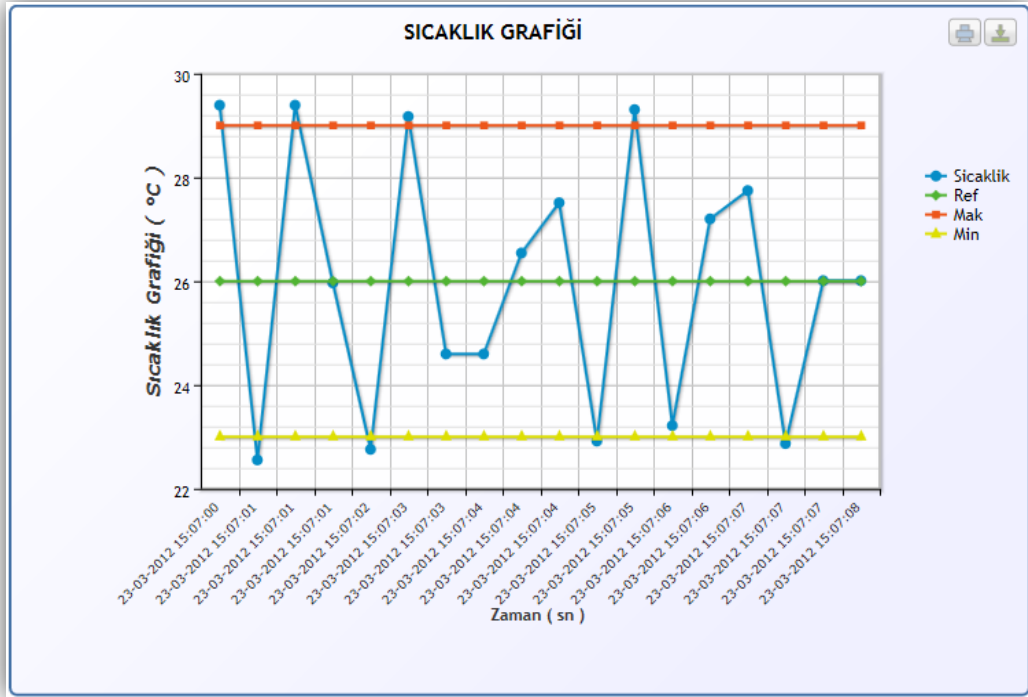
```
C:\Perl\bin\perl.exe
1 row affected
1017
1 row affected
1811
1 row affected
1932
1 row affected
1113
1 row affected
1709
1 row affected
2296
1 row affected
1027
1 row affected
1027
```

Şekil 5.20 : Perl’de oluşturulan soket programının ekran çıktısı

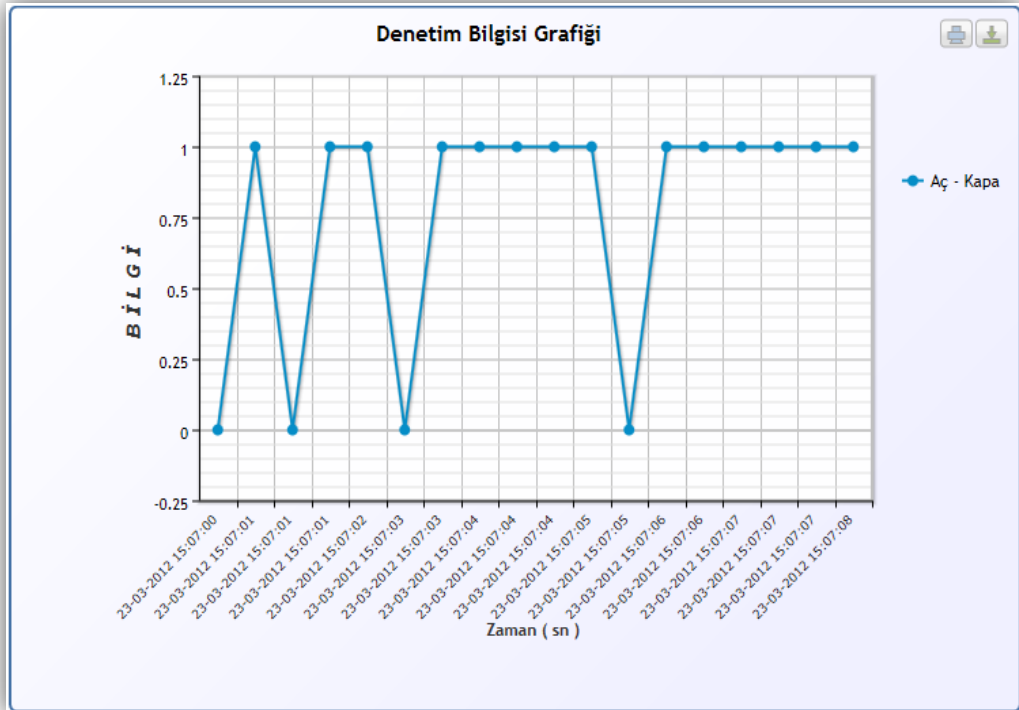


Şekil 5.21 : “mda320result” tablosunun veri tabanı yapısı

Şekil 5.22’de ve Şekil 5.23’de sistemin çalışmasını gösteren grafikler görülmektedir. Şekil 5.22’de, sistemin sıcaklığının, referans sıcaklık değerine göre maksimum ve minimum sıcaklık değerleri aralıklarında nasıl değiştiğini göstermektedir. Şekil 5.23’de, süreç denetim sisteminin sıcaklık değişimine göre ısıtıcısının açık-kapalı durumunu göstermektedir. Denetim bilgisi grafiğinden faydalanılarak denetlenen sistemin hangi zaman aralıklarında açma-kapama denetimini yaptığı anlaşılmaktadır.



Şekil 5.22 : Sistemin sıcaklık bilgisi



Şekil 5.23 : Sistemin denetim bilgisi

Sistemin ürettiđi ilk sıcaklık deęeri, üst sınır olan 29 °C'nin üstündedir. İlk durumda sıcaklık deęerine göre denetlenen sisteme “ısıtıcıyı kapa” denetim bilgisi gitmektedir. Sıcaklık deęeri alt sınır olan 23 °C'nin altına indiğnde ise denetlenen sisteme “ısıtıcıyı aç” denetim bilgisi gönderilmektedir. “Isıtıcıyı aç” denetim bilgisi ile sistem üzerindeki rezistans açılmaktadır. Sıcaklık artışı histeresiz aralıđın üst sınırı olan 29 °C'ye kadar devam etmektedir. Sıcaklık deęeri üst sınır olan 29 °C ölçüldüğünde ise denetlenen sisteme tekrar “ısıtıcıyı kapa” denetim bilgisi gönderilmektedir. Sistemin denetimi bir müdahale olmadan devam etmektedir. Grafikler, denetlenen sistemden gelen gerçek zamanlı verilere göre dinamik olarak çizdirilmektedir ve web tabanlı olarak uzaktan gözlemlenebilmektedir.

5.5. Sonuç

Bu bölümde, KAEA'lar için tasarlanmış web arayüzünün blok şeması verilerek çalışma mantığı anlatılmıştır. AD'lerden gelen verileri soket programlama kullanarak bilgisayar ortamına nasıl alındığı ve veri tabanına nasıl kaydedildiğı gösterilmiştir. Web arayüzüne eklenen süreç denetim sisteminin web arayüzü kullanılarak nasıl denetlendiğı ve gözlemlendiğı anlatılmıştır.

Son olarak, geliştirilen web arayüzünün tanıtımı yapılarak, kullanıcılara sağladığı işlevler açıklanarak anlatılmıştır.

SONUÇLAR VE ÖNERİLER

Son yıllarda iletişim teknolojisinin gelişmesiyle birlikte küçük boyutlu, düşük güçlü, algılama, işlem yapma ve haberleşme yeteneklerine sahip çok işlevli algılayıcı düğümler geliştirilmiştir. KAA, AD'lerin bir araya gelerek oluşturduğu ağ yapısıdır. KAA'lar; askeri, sağlık, tarım, endüstriyel ve çevresel alanlarda ortam gözleme amaçlı olarak kullanılmaktadır. KAA'ların kullanım alanları ve kullanılma oranları her geçen gün gittikçe artmaktadır. KAA'ların bu artan kullanım talebine göre donanımsal ve yazılımsal mimarilerinin geliştirilmesi gerekmektedir.

Tez çalışması ile AD'lerden gelen fiziksel ortama ait veriler, bir baz istasyonu yardımıyla bilgisayar ortamına alınıp, bilgisayar ortamında veri tabanına kaydedilerek, bu verilerin anlık olarak internet üzerinden gözlenmesi sağlanmıştır. Bu sayede algılamanın yapıldığı fiziksel ortamdaki bağımsız olarak, AD verilerine web üzerinden erişilebilmesine imkan sağlanmıştır. Ayrıca veri tabanına kaydedilen veriler farklı amaçlar için tasarlanabilecek uygulamaların geliştirilmesine esnek bir alt yapı hazırlamaktadır.

Geliştirilen web tabanlı arayüz ile [24]'deki çalışmada kullanılan süreç denetim sistemi için geliştirilen deney seti düzeneği haberleştirilerek web tabanlı arayüzün kullanım alanı ve olanakları artırılmıştır. Web arayüzüne eklenen bu modül ile KAEA'larda uzaktan denetim gerçekleştirilmiş ve sistemin çalışması web tabanlı olarak izlenebilmiştir.

Tez çalışmasının katkıları aşağıdaki gibi maddeler halinde sıralanabilir;

- Fiziksel ortama ait verilerin, ortama bağlı kalmadan anlık bir şekilde web tabanlı olarak izlenilmesi ve kaydedilmesi sağlanmıştır.
- Veri tabanına kaydedilen verilerin, belirlenen bir amaca yönelik değerlendirilip, işlenerek farklı uygulamaların geliştirilmesine olanak sağlanmıştır.

- Oluşturulan sistemin aşamaları anlatılarak, KAA mimarisinin yazılımsal ve donanımsal bileşenleri ile ilgili bilgiler sunulmuştur.
- Web arayüzü sayesinde kullanıcılar, AD'lerin bilgilerine kolayca erişebilmekte, AD'lerin ağ topolojisini görebilmekte, AD'lerin anlık verilerini görebilmekte ve fiziksel ortama ait verilerin çeşitli şekillerde grafiklerini çizdirebilmektedir.
- Hazırlanan web arayüzü geliştirilmeye açık bir yapıda olup, çok çeşitli uygulama alanlarına hitap edebilen özellikleri içeren bir yaklaşım sunmaktadır.
- Web arayüzü ile KAA'larda uzaktan sistem denetimi yapılmıştır ve denetlenen sistemin web tabanlı olarak uzaktan gözlenmesi sağlanmıştır.

Tez çalışmasının ileride geliştirilmesi gereken yönleri;

- Tez çalışmasında oluşturulan web arayüzü üzerinden AD'leri hava üzerinden programlıyarak (OTAP) sistemin dinamikliği ve kullanılabilirliği artırılabilir.
- Gelişen teknolojiye ve artan talebe göre web arayüzüne eklenmesi gereken bazı özel durumlar ve gereksinimler ortaya çıkabilir. Sonraki çalışmalarda bu özel durumlar ve gereksinimler göz önüne alınarak web arayüzü ileri seviyede genişletilebilir ve geliştirilebilir.

KAYNAKLAR

- [1] Solak, S., Çeken, C., “Kablosuz Algılayıcı Ağlarda Enerji ve Gecikme Duyarlı MAC Protokollerinin Karşılaştırılmalı Başarım Analizi”, **HABTEKUS**, İstanbul, Türkiye, (2009).
- [2] Çakıcı, S., “Kablosuz Algılayıcı Ağların Analitik Modellenmesi”, Yüksek Lisans Tezi, **Kocaeli Üniversitesi Fen Bilimleri Enstitüsü**, (2009).
- [3] Online, http://www.gunduz.org/seminer/pg/PostgreSQL-8.0_DevrimGunduz_29112004.pdf, (**Ziyaret Tarihi: 2012**).
- [4] Online, <http://www.pgadmin.org>, (**Ziyaret Tarihi: 2012**).
- [5] Crossbow, “MoteWorks Getting Started Guide”, **Crossbow Technology**, Revision D, (2007).
- [6] Crossbow, “Xmesh Users Manual”, **Crossbow Technology**, Revision D, (2007).
- [7] Aktaş, F., Çeken, C., Erkan, K., Yıldırım, M., “Kablosuz Algılayıcı Ağlar Kullanılarak Birinci Dereceden Ölü Zamanlı Bir Sistemin Denetimi”, **6th International Advanced Technologies Symposium (IATS'11)**, Elazığ, Türkiye, (2011).
- [8] Crossbow, “XServe User’s Manual”, **Crossbow Technology**, Revision D, (2007).
- [9] Online, <http://www.php.net>, (**Ziyaret Tarihi: 2012**).
- [10] Online, <http://jquery.com>, (**Ziyaret Tarihi: 2012**).
- [11] Online, <http://www.wampserver.com/en>, (**Ziyaret Tarihi: 2012**).
- [12] Online, <http://www.perl.org>, (**Ziyaret Tarihi: 2012**).
- [13] Online, <http://www.tinyos.net>, (**Ziyaret Tarihi: 2012**).
- [14] Online, <http://nescs.sourceforge.net>, (**Ziyaret Tarihi: 2012**).
- [15] Online, <http://www.activestate.com>, (**Ziyaret Tarihi: 2012**).
- [16] Akyıldız, I. F., Su, W., Sankasubramaniam, Y., Çayırıcı, E., “Wireless Sensor Networks: A Survey”, **Computer Networks**, 393 – 422, (2002).

- [17] Okçuoğlu, Z., Ertürk, İ., Karahan, A., “Kablosuz Algılayıcı Ağ Uygulaması: İdeal İzleme”, *EMO*, (2008).
- [18] Türker, G. F., Kutlu, A., “Medikal Bilişimde Kablosuz Algılayıcı Ağlar ve İnternet”, *XVI. Türkiye'de İnternet Konferansı*, Ege Üniversitesi (2011).
- [19] Turhan, E., Erkan, K., Aktaş, F., Çeken, C., “Matlab ile Kablosuz Algılayıcı Ağlar İçin Ortam İzleme Amaçlı Kullanıcı Arayüz Tasarım”, *Elektrik-Elektronik ve Bilgisayar Sempozyumu*, Fırat Üniversitesi (2011).
- [20] Bekçibaşı, U., Sevin, A., Ekiz, H., Bayılmış, C., Ertürk, İ., Atmaca, S., Bandırmalı, N., Seyhun, A., Akgül, M. M., “K-ATKS: Kablosuz Algılayıcı Ağ Kullanarak Gemiler İçin Geliştirilen Bir Acil Durum Tespit ve Kurtarma Sistemi”, *Akademik Bilişim 2010*, Muğla, Türkiye, (2010).
- [21] Çelik, O., Yiğiter, E., Sedef, H., “Kablosuz Ağ Tabanlı Gezgin Keşif Robotu: Kaşif”, *Elektrik-Elektronik-Bilgisayar ve Biyomedikal Mühendisliği 13. Ulusal Kongresi ve Fuarı*, Ankara, Türkiye (2009).
- [22] Online, <http://jsplumb.org/doc/usage.html>, (**Ziyaret Tarihi: 2012**).
- [23] Online, <http://www.highcharts.com>, (**Ziyaret Tarihi: 2012**).
- [24] Aktaş, F., “Kablosuz Algılayıcı/Eyleyici Ağlarla Denetim Sistemi Tasarımı”, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, (2012).
- [25] Song, G., Zhang, J., Ye, X., Niu, Y., Song, A., “Wireless Sensor and Actuator Network System for Calling Home Robots”, *Proceedings of the IEEE International Conference on Information and Automation*, Harbin, China, (2010).
- [26] Yeh, L. W., Lu, C. Y., Kou, C. W., Tseng, Y. C., “Autonomous Light Control by Wireless Sensor and Actuator Networks”, *Sensors Journal, IEEE*, Vol. 10, 1029-1041, (2010).
- [27] Peng, Y., Lahusen, R., Shirazi, B., Song, W., “Design Of Smart Sensing Component For Volcano Monitoring”, *Intelligent Environments, 2008 IET 4th International Conference on*, Seattle, WA, (2008).
- [28] Akyıldız, I. F., Kasımoğlu, İ. H., “Wireless Sensor And Actor Networks: Research Challenges”, *Ad Hoc Networks 2*, 351 – 367, (2004).

EKLER

EK-A WEB TABANLI ORTAM GÖZLEME UYGULAMASI VERİ TABANI SQL KODLARI

“wsn” Veri Tabanı:

```
-- Database: wsn  
  
-- DROP DATABASE wsn;  
  
CREATE DATABASE wsn  
WITH OWNER = tele  
ENCODING = 'UNICODE';
```

“mts400result” Tablosu:

```
-- Table: mts400result  
  
-- DROP TABLE mts400result;  
  
CREATE TABLE mts400result  
(  
    result_time timestamp without time zone,  
    amtype character varying(200),  
    nodeid character varying(200),  
    parent character varying(200),  
    group_ character varying(200),  
    socketid character varying(200),  
    packet_id character varying(200),  
    voltage character varying(200),  
    humid character varying(200),  
    humtemp character varying(200),  
    kalibrasyonw0 character varying(200),  
    kalibrasyonw1 character varying(200),  
    kalibrasyonw2 character varying(200),  
    kalibrasyonw3 character varying(200),  
    prtemp character varying(200),  
    press character varying(200),  
    taosch0 character varying(200),  
    taosch1 character varying(200),  
    accel_x character varying(200),  
    accel_y character varying(200),
```

```

taoch_0 character varying(200),
kalibrasyonb0 character varying(200),
kalibrasyonb1 character varying(200),
kalibrasyonb2 character varying(200),
kalibrasyonb3 character varying(200),
kalibrasyonb4 character varying(200),
kalibrasyonb5 character varying(200),
kalibrasyonb6 character varying(200),
kalibrasyonb7 character varying(200)
)
WITHOUT OIDS;
ALTER TABLE mts400result OWNER TO tele;
COMMENT ON TABLE mts400result IS 'Mote View Cihazlarından Gelen Veri
Paketleri, Tablo Alanlarında Verilerin Geliş Sıralarına Göre Tutulacaktır.';

```

“mda320result” Tablosu:

```

SET statement_timeout = 0;
SET client_encoding = 'SQL_ASCII';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;
SET search_path = public, pg_catalog;
SET default_tablespace = '';
SET default_with_oids = false;
--
-- Name: mda320result; Type: TABLE; Schema: public; Owner: postgres;
Tablespace:
--
CREATE TABLE mda320result (
    result_time timestamp without time zone,
    value integer
);
ALTER TABLE public.mda320result OWNER TO postgres;
--
-- Name: TABLE mda320result; Type: COMMENT; Schema: public; Owner:
postgres
--
COMMENT ON TABLE mda320result IS 'Veri edinim bordundan gelen veriler bu
tabloda tutulacak.';

```


EK-B WEB TABANLI ORTAM GÖZLEME UYGULAMASI WEB ARAYÜZÜ PROGRAM KODLARI

“node.php” Web sayfası kodu:

```
<?php
include "baglan.php";

$sorgu_nodeid = "SELECT DISTINCT(nodeid) AS dugum FROM mts400result
ORDER BY nodeid ASC;";
$sorgu_nodeid_sonuc = pg_query($baglanti,$sorgu_nodeid);
$node_id=array();
$sayac = 0;
while($sorgu_nodeid_veriler = pg_fetch_assoc($sorgu_nodeid_sonuc))
{
$node_id[$sayac] = $sorgu_nodeid_veriler['dugum'];
$sayac++;
}
?>

<div style="width:210;vertical-align:top;-moz-box-shadow:      0px 1px 2px 2px
#CCC;
  -webkit-box-shadow: 0px 1px 2px 2px #CCC;
  box-shadow:      0px 1px 2px 2px #CCC;" onmouseout="tooltipGizle();" >
<form id="dugumlerform" name="dugumlerform" onmouseout="tooltipGizle();" >
<table      cellpadding="7"      cellspacing="0"      width="210px"
onmouseout="tooltipGizle();">
  <thead>
    <tr>
      <th      colspan="3"      align="left"><div      class="ustegel"
onclick="NodeGetir();"></div></th>
    </tr>
    <tr>
      <th width="160px" align="left">Düğüm ID</th>
      <th width="30px" align="left">Durumu</th>
    </tr>
  </thead>
  <tbody>
    <tr onmouseout="tooltipGizle();">
      <td width="150px" onmouseover="Getir(' <?php echo $node_id[$i]; ?>');"
onmouseout="tooltipGizle();" > <?php echo $node_id[$i]; ?></td>
      <td width="30px"><div id="<?php echo $node_id[$i]; ?>"
style="width:29px;"></div></td>
```

```

        </tr>
        <tr><td colspan="3"><hr /></td></tr>
<?php
}
?>
</table>
<input type="hidden" id="dugumidleri" name="dugumidleri" value="<?php echo
$dugumidleri; ?>" />
</form>
</div>
<br />
<div id="gosterge" style=" -moz-box-shadow: 0px 1px 2px 2px #CCC;
 -webkit-box-shadow: 0px 1px 2px 2px #CCC;
 box-shadow: 0px 1px 2px 2px #CCC;" >
<table cellpadding="7" cellspacing="0" width="210px"
onmouseout="tooltipGizle();">
<tr>
<td><img src='images/yesil.png' width='20px' /></td>
<td >:</td>
<td><font size="-1"> İlgili Düğüm Çevrim İçi</font></td>
</tr>
<tr><td colspan="3"> <hr /></td></tr>
<tr>
<td><img src='images/gri.png' width='20px' /></td>
<td >:</td>
<td><font size="-1">İlgili Düğüm Çevrim Dışı</font></td>
</tr>
</table>
</div>

<?php
echo "<script>NodeDurum();</script>";
?>

```

“nodeanlikveri.php” Web sayfası kodu:

```

<link rel="stylesheet" type="text/css" href="css/table.css"/>
<?php
include "baglan.php";
$query = "SELECT * FROM mts400result ORDER BY result_time DESC LIMIT
10";
$result = pg_query($baglanti,$query) ;
?>
<div>
<table id="Tablo" summary="Anlık Düğüm Verileri">
<thead>
<tr>
<th scope="col" width="20px">Düğüm ID</th>
<th scope="col" width="150px">Zaman</th>

```

```

<th width="30px" rowspan="2" scope="col">Batarya (mV)</th>
<th width="30px" rowspan="2" scope="col">Nem (%)</th>
<th width="30px" rowspan="2" scope="col">Nem Sıcaklığı (&deg;C)</th>
<th width="30px" rowspan="2" scope="col">Basınç Sıcaklığı (&deg;C)</th>
<th width="110px" rowspan="2" scope="col">Basınç (mbar)</th>
<th width="110px" rowspan="2" scope="col">İvme-X (mg)</th>
<th width="110px" rowspan="2" scope="col">İvme-Y (mg)</th>
<th width="110px" rowspan="2" scope="col">Işık Şiddeti (lux)</th>
</tr>
<tr>
<th scope="col" width="30px"></th>
<th scope="col" width="150px"></th>
</tr>
</thead>
<tbody>
<?php
while($veri= pg_fetch_assoc($result))
{ ?>
<tr>
<td width="20px"><?php echo $veri["nodeid"]; ?></td>
<td width="150px"><?php $zaman = explode(' ', $veri["result_time"]);
$zaman[1] = substr($zaman[1],0,8); echo $zaman[0]. " " . $zaman[1]; ?></td>
<td width="30px"><?php echo $veri["voltage"]; ?></td>
<td width="30px"><?php echo $veri["humid"]; ?></td>
<td width="30px"><?php echo $veri["humtemp"]; ?></td>
<td width="30px"><?php echo $veri["prtemp"]; ?></td>
<td width="110px"><?php echo $veri["press"]; ?></td>
<td width="110px"><?php echo $veri["accel_x"]; ?></td>
<td width="110px"><?php echo $veri["accel_y"]; ?></td>
<td width="110px"><?php echo $veri["taoch_0"]; ?></td>
</tr>
<?php } ?>
</tbody>
</table>
</div>

```

“nodedurum.php” Web sayfası kodu:

```

<?php
$node_id = pg_escape_string($_POST["node_id"]);
date_default_timezone_set('Europe/Istanbul');
include "baglan.php";

function saatFarki($saat1,$saat2)
{
return abs(strtotime(date('Y-m-d '.$saat1)) - strtotime(date('Y-m-d '.$saat2)))/60;
}

$zaman = getdate();

```

```
$saat = $zaman['hours'].":".($zaman['minutes']-1).":".$zaman['seconds'];
```

```
$veritabani_saatkismi = array();
```

```
$gecicidizi = array();
```

```
$sorgu_node_kontrol = "SELECT * FROM mts400result WHERE  
nodeid='".$node_id.'" ORDER BY result_time DESC LIMIT 1";  
$sorgu_node_kontrol_sonucu = pg_query($baglanti,$sorgu_node_kontrol) ;  
$sorgu_node_kontrol_veriler = pg_fetch_assoc($sorgu_node_kontrol_sonucu);  
$veritabani_saatkismi = explode(' ', $sorgu_node_kontrol_veriler['result_time']);
```

```
if(intval(saatFarki($saat,$veritabani_saatkismi[1])) < 1)
```

```
{  
echo " " . "<script>document.getElementById('".$node_id  
.").style.background=#00FF00;</script>";  
echo "<img src='images/yesil.png' width='20px' />";  
}  
else  
{  
echo " " . "<script>document.getElementById('".$node_id  
.").style.background=#CCCCCC;</script>";  
echo "<img src='images/gri.png' width='20px' />";  
}  
?>
```

“tooltipdoldur.php” Web sayfası kodu:

```
<?php  
$id = pg_escape_string($_POST["id"]);  
include "baglan.php";  
$query = "SELECT * FROM mts400result WHERE nodeid='".$trim($id)."' ORDER  
BY result_time DESC LIMIT 1";  
$result = pg_query($baglanti,$query) ;  
$kayitsayisi = pg_num_rows($result);  
$veri = pg_fetch_assoc($result);
```

```
if($kayitsayisi>0)
```

```
{  
?>
```

```
<table align="left">
```

```
<tr>
```

```
<td width="200px"><b>Düğüm Id :</b><?php echo $veri["nodeid"]; ?></td>
```

```
<td width="200px"><b>Zaman :</b><?php $zaman = explode(''  
, $veri["result_time"]); $zaman[1] = substr($zaman[1],0,8); echo $zaman[0]."  
".$zaman[1]; ?></td>
```

```
</tr>
```

```
<tr>
```

```

<td width="200px"><b>Batarya :</b><?php echo $veri["voltage"]." mV";
?></td>
<td width="200px"><b>Nem :</b><?php echo $veri["humid"]." %"; ?></td>
</tr>
<tr>
<td width="200px"><b>Nem Sıcaklığı :</b><?php echo
ceil(floatval($veri["humtemp"]))." &deg;C"; ?></td>
<td width="200px"><b>Basınç Sıcaklığı :</b><?php echo
ceil(floatval($veri["prtemp"]))." &deg;C"; ?></td>
</tr>
<tr>
<td width="200px"><b>Basınç :</b><?php echo ceil(floatval($veri["press"]))."
mbar"; ?></td>
<td width="200px"><b>Işık Şiddeti :</b><?php echo
ceil(floatval($veri["taoch_0"]))." lux"; ?></td>
</tr>
<tr>
<td width="200px"><b>İvme X :</b><?php echo
ceil(floatval($veri["accel_x"]))." mg"; ?></td>
<td width="200px"><b>İvme Y :</b><?php echo
ceil(floatval($veri["accel_y"]))." mg"; ?></td>
</tr>
</table>

<?php
}
else
{
echo "<center><p>Düğüme Ait Kayıt Bulunamadı...</p></center>";
}
?>

```

“grafik.php” Web sayfası kodu:

```

<?php
include "baglan.php";
?>
<div id="dersbolme" style="width:745px">
<form id="GrafikForm" name="GrafikForm">
<table class="arama" width="745px">
<thead>
<tr>
<th colspan="3">
Grafik Seçimi Ekranı
</th>
</tr>
</thead>
<tr width="50%">
<td>

```



```

$i=0;
while($sorgu_nodeid_veriler = pg_fetch_assoc($sorgu_nodeid_sonuc))
{
$i++;
?>
        <?php echo $sorgu_nodeid_veriler['dugum']; ?> : <input type="checkbox"
id="dugum-<?php echo $i; ?>" name="dugum-<?php echo $i; ?>" value="<?php
echo $sorgu_nodeid_veriler['dugum']; ?>" />
<?php
}
?>
        </td>
</tr>
<tr>
        <th align="center" colspan="3">
        
        <sup onclick="GrafikGetir();"> Grafik Göster </sup>
        </th>
</tr>
<tr><td></td></tr>
</table>
</form>
<div id="grafiksonuc">
<br />
</div>
</div>

```

“sistemkontrolu.php” Web sayfası kodu:

```

<?php
include "baglan.php";
?>
<div id="sistemkontrol" name="sistemkontrol" style="width:745px">
<table class="arama" width="745px">
<thead>
<tr>
        <th colspan="5">
                Referans Sıcaklık Değeri Oluşturma
        </th>
</tr>
</thead>
<tr width="50%">
        <td>
                <table class="">
                <tr>
                        <td width="135">Referans Sıcaklık ( &deg;C )</td>
                        <td width="3">:</td>
                        <td>

```

```

        <input      type="text"      id="refsinyal"      name="refsinyal"
style="width:45px"/>
    </td>
    <td>
        <input type="button" value="Gönder" onclick="SistemKontrolu();" />
    </td>
    <td>
        <input      type="button"      value="Sistem      Grafiği"
onclick="SistemKontroluGrafik();" />
    </td>
</tr>
</table>

</td>
</tr>
<tr><td></td></tr>
</table>
</div>
<br />
<div id="referanssinyali" name="referanssinyali" style="width:745px;"></div>
<br />
<div id="sistemgrafik" name="sistemgrafik"></div>
<br />
<div id="sistemgrafikackapa" name="sistemgrafikackapa"></div>

```


ÖZGEÇMİŞ

1982 yılında Kocaeli'nin İzmit ilçesinde doğdu. Lise eğitimini Körfez Anadolu Teknik Lisesi Elektrik Bölümü'nde tamamladı. 2004 yılında girdiği Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü Bilgisayar Öğretmenliği Programı'ndan 2008 yılında mezun oldu. 2008 yılında başladığı Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalı'ndaki Yüksek Lisans Eğitimine devam etmektedir. 2011 yılından itibaren Namık Kemal Üniversitesi Bilgi İşlem Daire Başkanlığı'nda Uzman olarak çalışmaktadır.