

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**AKILLI EV YÖNETİM SİSTEMLERİ (AKYÖN) İÇİN MOBİL
İSTEMCİ PROTOKOL TASARIMI**

MUSTAFA MENTEŞOĞLU

KOCAELİ 2016

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

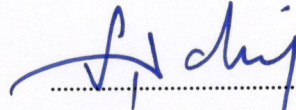
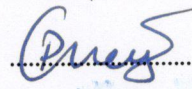

AKILLI EV YÖNETİM SİSTEMLERİ (AKYÖN) İÇİN MOBİL
İSTEMCİ PROTOKOL TASARIMI

MUSTAFA MENTEŞOĞLU

Yrd. Doç. Dr. Suhap ŞAHİN
Danışman, Kocaeli Üniv.

Yrd. Doç. Dr. Pınar ONAY DURDU
Jüri Üyesi, Kocaeli Üniv.

Yrd. Doç. Dr. Ali ÇALHAN
Jüri Üyesi, Düzce Üniv.


.....

.....

.....

Tezin Savunulduğu Tarih: 02.11.2016

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada, akıllı ev sistemine yönelik mobil cihaz-gömülü bilgisayar tabanlı kontrol ünitesi ile sensör/ cihazlar arasında haberleşme protokolü tasarlanıp bu yazılımın gerçekleştirilmesi sağlanmıştır. Bu çalışma kapsamında kullanılan algılayıcı sistem kontrol birimi donanım tasarımı KOU Teknopark bünyesinde faaliyet gösteren MİLETS Ltd. (www.milets.com.tr) firmasının Tübitak Teydeb destekli 113E033 nolu AKYÖN projesi kapsamında yapılmıştır.

Tez çalışmasının planlama, araştırma ve yürütülmesi esnasında ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Yrd. Doç. Dr. Suhap ŞAHİN' e,

Aynı şekilde, birikimlerini benimle paylaşan, gösterdiği sabır, anlayış, verdiği sürekli destek için, saygıdeğer hocam Prof. Dr. Adnan KAVAK' a,

Yüksek öğrenim hayatım boyunca beni her zaman cesaretlendiren, her zaman yanımda olan, manevi desteğini hiç bir zaman esirgemeyen yol arkadaşım Zeynep Pınar YAKINCI' ya

Hayatım boyunca her zaman yanımda olan, maddi ve manevi hiç bir desteğini esirgemeyen Anne ve Babama teşekkürlerimi sunarım.

Kasım – 2016

Mustafa MENTEŞOĞLU

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	v
SİMGELER DİZİNİ VE KISALTMALAR	vi
ÖZET.....	vii
ABSTRACT	viii
GİRİŞ	1
1. AKILLI EV HAKKINDA GENEL BİLGİLER	8
1.1. Akıllı Ev Kavramı	8
1.2. Akıllı Ev Kavramının Tarihçesi	12
1.3. Dünyada Akıllı Ev Kavramı	13
1.4. Türkiye’de Akıllı Ev Kavramı	15
1.5. Akıllı Ev Yönetim Sistemleri	16
1.5.1. Kontrol edilebilir evler	17
1.5.2. Programlanabilir evler.....	18
1.5.3. Yapay zekâya sahip evler	20
1.5.4. Akıllı ve çevreci evler	21
1.5.5. Yaşlı ve engelli insanlar için akıllı evler	22
1.6. Akıllı Evlerin Avantajları	23
1.7. Akıllı Evlerin Dezavantajları.....	26
2. SİSTEM MİMARİSİ.....	28
2.1. Haberleşme	28
2.1.1. Kablosuz haberleşme.....	29
2.1.2. Ethernet	30
2.1.3. TCP/IP	31
2.1.4. UDP	33
2.1.5. Hücresel veri iletişimi (3G/4G).....	33
2.2. Android Yazılım Geliştirme Ortamı.....	34
2.3. Sistemde Yer Alan Sensörler	36
2.3.1. Gaz Sensörleri	37
2.3.2. Yangın ve duman sensörleri	38
2.3.3. Su sensörleri	40
2.3.4. Hareket sensörleri.....	42
2.3.5. Işık sensörleri	43
2.3.6. Sıcaklık sensörleri	45
2.4. Akıllı Ev Yönetim Sistemleri Protokolleri	46
2.5. Beagle bone black	50
3. YAZILIM VE ALGORİTMALAR.....	53
3.1. Sistem Mimarisi	53
3.2. Haberleşme Mimarisi	55
3.2.1. İletişim protokolü	55

3.2.2. Haberleşme paket yapısı.....	57
3.2.3. Sisteme yeni sensör cihaz eklenmesi	59
3.3. Ayarların Saklanması	62
3.4. Haberleşme Performans Testi	63
4. SONUÇLAR VE ÖNERİLER	64
4.1. Öneriler.....	64
KAYNAKLAR	66
EKLER.....	69
KİŞİSEL YAYIN VE ESERLER	98
ÖZGEÇMİŞ	99



ŞEKİLLER DİZİNİ

Şekil 2.1. Gaz Sensörü	38
Şekil 2.2. Yangın ve Duman sensörü	39
Şekil 2.3. Su sensörü	41
Şekil 2.4. Beaglebone Black genel yapısı	51
Şekil 2.5. Beaglebone Black genişleme yuvası GPIO yapısı.....	51
Şekil 3.1. Sistem Mimarisi	53
Şekil 3.2. Sistem Genel Haberleşme Mimarisi	54
Şekil 3.3. AKYÖN Sistemi fiziksel mimarisi	55
Şekil 3.4. El Sıkışma Protokolü	56
Şekil 3.5. Sunucu-Mobil Cihaz İletişim Protokolü	57
Şekil 3.6. Oda Ayarları Arayüzü.....	61
Şekil 3.7. Sensör Ayarları Arayüzü.....	61
Şekil 3.8. Kumanda Panel Arayüzü	62

TABLULAR DİZİNİ

Tablo 3.1. BeagleBone Black sisteminin özellikleri	54
Tablo 3.2. Konfigürasyon Mesaj Paket Yapısı	58
Tablo 3.3. Yönetim Mesaj Paket Yapısı	59
Tablo 3.4. Sisteme Eklenebilen Sensör/Cihaz Listesi.....	60
Tablo 3.5. Temel Yapılandırma Ayarları.....	60
Tablo 3.6. Haberleşme performans test sonuçları.....	63



SİMGELER DİZİNİ VE KISALTMALAR

Kısaltmalar

API	: Application Programming Interface (Uygulama Programı Arabirimi)
AKYÖN	: Akıllı Ev Yönetim Sistemi
CAN	: Controller Area Network (Kontrolör Alan Ağı)
DDR	: Double Data Rate (Çift Veri Oranı)
DTMF	: Dual Tone Multi Frequency (Çift Ton Çoklu Frekans)
EMMC	: Embedded Multimedia Card (Gömülü Multimedya Kart)
GPIO	: General Purpose Input Output (Genel Maksatlı Giriş Çıkış)
GPRS	: General Packet Radio Service (Genel Paket Radyo Hizmeti)
GSM	: Global System for Mobile (Mobil İletişim İçin Küresel Sistem)
HDMI	: High Definition Multimedia Interface (Yüksek Çözünürlüklü Multimedya Arabirimi)
I2C	: Interface to Communicate (İletişim Arabirimi)
LAN	: Local Area Network (Yerel Alan Ağı)
MAC	: Media Access Control (Ortam Erişim Kontrolü)
MAN	: Metropolitan Area Network (Metropol Alan Ağı)
OSI	: Open Systems Interconnection (Açık Sistem Bağlantısı)
PWM	: Pulse Width Modulation (Puls Genişlikli Modülasyon)
RAM	: Random Access Memory (Rastgele Erişimli Bellek)
SMS	: Short Message Service (Kısa Mesaj Servisi)
SPI	: Serial Peripheral Interface (Seri Çevresel Arayüz)
TCP	: Transmission Control Protocol (Gönderim Kontrol Protokolü)
TCP/IP	: Transmission Control Protocol-İnternet Protocol (Gönderim Kontrol Protokolü/İnternet Protokolü)
TEYDEB	: Teknoloji ve Yenilik Destek Programları Başkanlığı
UART	: Universal Asynchronous Receiver Transmitter (Evrensel Asenkron Alıcı ve Gönderici)
UDP	: User Datagram Protocol (Kullanıcı Veribloğu İletişim Kuralları)
WAN	: Wide Area Network (Geniş Alan Ağı)

AKILLI EV YÖNETİM SİSTEMLERİ (AKYÖN) İÇİN MOBİL İSTEMCİ PROTOKOL TASARIMI

ÖZET

Güncel hayatımızı daha emniyetli ve pratik hale getiren akıllı ev yönetim sistemlerinin kullanımı son yıllarda artış göstermiş olup, mobil cihaz ve internet erişilebilirliğinin artması ile bu sistemlerin mobil cihazlar ile uzaktan yönetilmesi ve durumlarının takip edilmesi ihtiyacını ortaya çıkarmıştır.

Bu çalışma kapsamında, MİLETS firması tarafından fiziksel ve donanımsal olarak gerçekleştirilmiş olan Tübitak TEYDEB destekli 113E033 numaralı AKYÖN akıllı ev yönetim sistemi üzerinde bulunan BeagleBone Black gömülü bilgisayar sistemi ile mobil cihazlar arasında yerel ve uzak ağlar aracılığıyla haberleşmenin sağlanması amaçlanmıştır. Gömülü sistem ve mobil cihazlar arasında iletişim TCP/IP ile gerçekleştirilmiştir. Açık kaynak kodlu Android işletim sistemini kullanan mobil cihazlar için Java programlama dili kullanılarak kullanıcı arayüzü ve haberleşme protokolü geliştirilmiştir. Geliştirilen kullanıcı arayüz yazılımı ve haberleşme protokolü ile yerel ağ ve geniş alan ağları üzerinden AKYÖN sisteminde mevcut bulunan çeşitli veri toplama sensörlerinden verilerin alınması, eyleycilerin uzaktan kumanda edilmesi, durumlarının sorgulanması ve sergilenmesi mobil cihazlar kullanılarak başarılı bir şekilde sağlanmıştır.

Anahtar Kelimeler: Akıllı Ev Yönetim Sistemi, Haberleşme Protokolleri, Mobil Cihazlar, Otomasyon, Sensör.

DESIGN AND IMPLEMENTATION OF A COMMUNICATION PROTOCOL FOR MOBILE DEVICE CONTROLLED SMART HOME MANAGEMENT SYSTEM

ABSTRACT

The smart house automation systems have been widespread for the past several years, and with the increasing accessibility of mobile devices and the internet, a need for the remote management and attitude control of these systems emerged.

Within the scope of this study, it was aimed to provide communication via local and remote networks between mobile devices and BeagleBone Black embedded computer system which is present on AKYÖN smart house automation system that is numbered as 113E033, supported by Tübitak TEYDEB, and realized physically and in terms of hardware by MILETS firm. The communication between the mobile devices and the embedded system was provided via TCP/IP. A user interface and communication protocol were developed for the devices which use open-source software, Android. By the use of the developed interface and the communication protocol, gathering data from various data collector sensors present on AKYÖN system through local and wide area networks, the remote controlling of the actuators, the examination and exhibit of their status via mobile devices were successfully enabled.

Key Words: Smart Home Automation System, Communication Protocols, Mobile Devices, Automation, Sensor.

GİRİŞ

Ev; barınma ve fiziki ihtiyaçların giderildiği kişiye özel alandır. Bireyin yaşam tarzına bürünen ev, kişilerin hem bedensel hem ruhsal sağlığını doğrudan etkileyen, hem de huzurunu sağlayan temel etkidir.

Son yirmi yıla baktığımızda yaşanan teknolojik gelişmeler hem birey hayatında hem de yaşanan mekan tasarımında büyük gelişme ve değişmelere sebep olmuştur. Enformasyon teknolojisinde yaşanan hızlı gelişim ve değişimin sonunda, iletişim teknolojilerinin bilgisayar altyapıları ve bileşimleri vasıtasıyla farklı sistemlerin kontrolünü yapabilmesi, dışarıdan gelebilecek verileri algılayıp bu verilere göre tepki ve kararlar verebilen bir sistemin ortaya çıktığı görülerek bu sistemi akıllı ev dediğimiz evlerde kullanmak mümkün olmuştur. Akıllı ev ve akıllı binalar pek çok alt kümesi ile entegre olmuş şekilde senkron konuma getirmiş bir kontrol kumandası ile kontrol edilebilen bir sistem olarak açıklanabilir (Aslan, 2014).

Akıllı evler ve akıllı binalar, içinde yaşayanların rahatlığını maksimize eden, üst düzey bir güvenlik sistemi sağlayan ve konforu ön planda tutan yapılar olarak tanımlanabilir. Var olan kablolar ile aygıtların kontrolü kolaylıkla sağlanır. Akıllı bina ve evlerde mevcut ısı, nem ve pek çok durum akıllıca kontrol edilmelidir. Akıllı evde yaşayanlar evde var olan bütün elektroniklerin kontrolünü elde tutabilir. Böylelikle daha önce el ile kontrol edilen elektronikleri kumanda ile kontrol eden ev sahipleri manuel kontrolden kurtulur.

Gelişen ve gelişmekte olan teknoloji, hem cihaz sayısını hem de cihazların kontrol edildiği ünite sayısını artırmıştır.

Günümüzde yeni çıkan teknolojik ürünler hem yapısal hem de tasarımsal olarak uzaktan kumanda ile kontrol edilebilir niteliktedir. Isınmadan aydınlanmaya ve hatta güvenliğe varana kadar meydana gelen otomatik kontrol, tüm yapıların kontrolünü sağlamak açısından oldukça önemlidir.

Akıllı bina ve evler enerjiden tasarruf, verimlilikte artış ve kazanç olarak büyük avantaj sağlamaktadır. Akıllı yapıların hedefi bireylerin hayatını maksimum konforda geçirmesini sağlamak ve bu bağlamda kolaylaştırmaktır. Fiziksel engelli bireylerin de hayatı akıllı yapılar ile birlikte en konforlu ve rahat şekilde geçecektir.

Teknolojideki hızlı ve dinamik değişimle birlikte bu değişime ayak uydurulmak istenmesi ve gelişen teknolojinin değerlendirilmek istenmesi, akıllı ev teknolojilerinin de büyük bir hızla gelişmesini sağlar. Toplumsal ve bireysel gereksinimlerle paralel olarak akıllı bina ve akıllı evlerin otomasyon sistemleri geliştirilmektedir.

Akıllı evler ve otomasyon sistemleri özelliklerine göre, kontrol edilebilir evler, programlanabilir evler, yapay zekaya sahip evler, akıllı ve çevreci evler ve yaşlı ve engelli bireyler için akıllı evler olarak beş gruba ayrılmaktadır.

Haberleşme, akıllı ev ve bina sistemlerinin önemli bir parçasıdır. Haberleşme sistemi, kullanıcı veya sistemin algılayamadığı uzaktaki noktalara ulaşıp o noktalardan bilgi akışını sağlar. Haberleşme, uzaktan kontrol sisteminin temelini oluşturmaktadır. Ethernet, Internet, GPRS, SMS, WIFI, RF, RS232, RS485 gibi haberleşme protokolleri vasıtası ile kullanıcının sisteme erişmesi ve ana sistemin uzak noktaya erişmesi sağlanır.

Kablosuz bağlantı ağları, genel olarak iş sahasında esnek bir bağlantı alternatifi gibi kabul görmektedir. Frekans teknolojisi kullanılarak havada dağılan manyetik dalgalar ile gerçek zamanlı veri transferi yapılmasına olanak tanır. Kablosuz bağlantı teknolojisi ile kullanıcılar kolay ve rahat bir şekilde kaynaklara erişebilmektedir.

Günlük yaşantımıza büyük bir hızla giren internet, binalarımızda ya da ofislerimizde Ethernet teknolojisi aracılığı ile ulaştığı için var olan bilgisayar ve bilgisayar sistemlerine kolayca uyum sağlar.

Ethernet, Yerel ağ (LAN) için kullanılmakta olan Veri Çerçevesi (Data Frame) tabanlı bir bilgisayar ağı teknolojisidir. Ethernet kelimesinin kökeni “ether” den gelir.

OSI ağ adı verilen modelin Fiziksel katmanı için Veri bağlantısı katmanı/ Ortam erişim kontrolü (MAC) üstünden ağ erişimi aracılığı ile bir dizi kablolama ve sinyalleşme standardı ve ortak bir adresleme formatı tanımlar.

TCP/IP , Bilgisayarlarla internet arasında iletişim sağlayan protokol grubuna verilen isimdir. İki ya da daha fazla bilgisayar arasında veri ve bilgi alışverişi IP adı verilen paketler aracılığı ile yapılır. Mesela A ve B gibi iki bilgisayarımız olsun. A, B'ye göndermek istediği verileri paketler haline getirerek gönderir. Bahsedilen paketin içeriğinde ise A ve B'nin adresleri mevcuttur. A'dan paket gönderimi yapıldığını varsayarsak, A'nın gönderdiği paket IP paketi, A'nın adresi ise IP adresi olarak tanımlanmaktadır. A'nın göndermiş olduğu paket B'nin adresini taşıdığından router yardımıyla doğru bir şekilde doğru bilgisayara iletilir.

UDP katmanı da tıpkı TCP gibi bir taşıma protokolü olarak karşımıza çıkar. TCP protokolü ile arasında bir fark olarak UDP kullanılıyorsa veri tam olarak gidip gitmedi mi bunun doğrulaması yapılamaz. Dolayısıyla bir gönderim garantisi de sağlanmaz.

Akıllı ev sistemlerinde hızın önemli olmasından dolayı karar verilmesi zorunlu olan UDP, teslim garantisi olmamasının yanında TCP protokolünden çok daha süratli ve verimlidir. Akıllı evler için geliştirilmiş mekanizmada, kontrol kumandasına dokunulması ile eş zamanlı olarak komutun uygulanması beklenir. Bu durumda komutun verilme anı ile uygulama anı arasında geçen sürenin uzaması dezavantajdır.

Akıllı ev ve akıllı binaların yönetiliş sistemine dair mobil işletim sistemi tercih edildiğinde, açık kaynak kodlu kütüphane kullanma özgürlüğü, geliştirilip değiştirilen yazılımların dağıtılabilmeye özgürlüğü ve işletim sistemlerinde farklılıklar (Windows vs.) uygulama geliştirme esnekliği var olması nedeniyle Android yazılım önceliklidir. Android yazılım, bilgisayarlarda özel bir donanıma ihtiyaç duymadığı için büyük avantaj ve esneklik sağlamaktadır.

Akıllı ev ve binaların otomasyon sistemleri, kullanıcıları günlük yaşamda etkileyen parametreleri kontrol altına alarak daha az stresli daha çok huzurlu ve mutlu bir hayat yaşamalarını hedefler. Söz konusu parametreler bireyin gündelik hayatındaki iş

hayatı, aile yaşamı gibi alanlarını doğrudan etkileyen sıcaklık, nem, ışık seviyesi gibi parametrelerdir.

Akıllı ev ve binalarda evde yaşanan değişimler hareket sensörü, duman sensörü gibi sensörler vasıtasıyla iletilir. Söz konusu sensörler vasıtasıyla sistem kendi içinde kodladığı farklı bir sistemi ya da sensörü harekete geçirebilmektedir. Mesela duman sensörü duman algıladığı zaman gaz vanalarını kapatır, fiskiyeleri devreye sokar.

İletişim noktaları ve bu noktaların uzantısı olan birimler ev ve bina otomasyonunda en öncelikli ve önemli konulardan biridir. Hepsinin ayrı ayrı fonksiyonları olan, farklı firmalara ait birbirinden bağımsız cihazların bir arada ve uyumla çalışabilmesi tamamen iletişim noktalarının uyum ve yeteneğine bağlıdır.

Akıllı ev ve akıllı bina otomasyon sistemlerinde, beraber çalışılabilme ve genişletilebilirlik sağlanması için ortak bir çalışma yapılmamış olması ve hepsinin farklı üreticiler tarafından yapılıyor olması gelişmeyi biraz yavaşlatmıştır. Bu bağlamda farklı topluluklar ortak bir çalışmayla çözüm üretmeye çalışmış ve geliştirilen protokoller çalışmanın sonucunda incelenmiştir.

Bayram (2006), çalışmasında Akıllı Ev Yönetim Sistemlerinin ev içerisinde yapılan bir takım işlerin sensörler ve anahtarlamalar yardımı ile insan müdahalesi gerektirmeden otomatik olarak yerine getirilmesi, ev güvenliğinin sağlanması vb. işlemlerin periyodik olarak veya duruma göre ayarlanarak, otomatik veya isteğe bağlı şekilde gerçekleştirilmesi amacıyla insan yaşamını kolaylaştırmaya yönelik ev otomasyonuna örnek oluşturmayı amaçlamıştır. AKYÖN sisteminde bu amaçların karşılanması amacıyla kullanıcı tarafından sensörlerin ve eyleyicilerin belirli bir modele göre aktif/pasif yapabilme imkanı sağlanmasının ve sistemin belirli durumlara otomatik tepki vermesinin sağlanması amaçlanmıştır.

Kongaz (2007), çalışmasında Akıllı Ev Yönetim sistemini yönetmek ve kullanıcı ile haberleşmek amacıyla Analog Devices firması tarafından üretilen ADUC845 mikro denetleyicisini kullanmıştır. AKYÖN geliştirme projesi kapsamında ilk geliştirilen prototipte bu çalışmadaki yaklaşıma uygun olarak Microchip firması tarafından geliştirilen PIC 16F877A mikro denetleyicisi kullanılmıştır. Haberleşme protokolü

olarak TCP/IP protokolü kullanılmış fakat mikro denetleyici üzerinde dahili bir ethernet haberleşme modülü olmadığı için UART-Ethernet dönüştürücü kullanılmıştır. Burada haberleşme yığın boyutunun küçük olması (82 byte) sistemin birden çok kullanıcı ile kullanılması esnasında hatalara neden olduğu görülmüştür. Bu sebepten dolayı sistemin daha efektif kullanılabilmesi için son zamanlarda düşük maliyet, düşük enerji tüketimi ve yüksek performan ile ortaya çıkan üzerinde Linux işletim sistemi koştan Beagle Bone Black gömülü bilgisayarının kullanılmasına karar verilmiştir.

Gügöl (2008) gerçekleştirdiği yüksek lisans tez çalışmasında akıllı ev sistemini uzaktan kontrol edebilmek için DTMF kullanmıştır. Bu sistem hem iletişim maliyetinin yüksek olması, bu sistemin efektif olarak kullanılabilmesi ayrı bir telefon hattının gerekmesi sistemin maliyetlerini yükseltmektedir. AKYÖN sisteminde kullanılan TCP/IP protokolünü mobil iletişim hatları üzerinden veri servislerini kullanarak evdeki mevcut internet alt yapısını aracılığıyla ek bir yük getirmeden hızlı ve geri bildirim imkanı sağlayarak sistem ile haberleşebilmekte ve yönetimini sağlamaktadır.

Cincirop (2009) gerçekleştirdiği yüksek lisans tez çalışmasında Microchip firması tarafından üretilen PIC 16F877A mikro kontrolörü ile bir cep telefonunu sistem üzerinde modem olarak kullanmıştır. Sistemi yönetmek ve alarmlar için SMS kullanmıştır.

Tosunoğlu (2009) gerçekleştirdiği yüksek lisans tez çalışmasında bir çamaşır makinesine kendi geliştirdiği elektronik devre aracılığıyla bilgisayar üzerinden bağlanarak çamaşır makinesini kumanda etmiş ve makine üzerindeki verileri bilgisayara aktarmıştır. AKYÖN projesi kapsamında tüm sensör, eyleyici ve cihazların gömülü sisteme bağlanması ve verilerin bu sistem üzerinde depolanması, analiz edilmesi sağlanmıştır.

Algoiare (2014) gerçekleştirdiği yüksek lisans tez çalışmasında Arduino mikro denetleyici kartı üzerine yerleştirilmiş GSM modülü aracılığıyla Akıllı Ev Yönetim sistemine uzaktan komut ve alarmları göndermeyi ve almayı hedeflemiştir. Bu sistemde GSM modülü aracılığıyla sisteme yönetmek amacıyla SMS' ler

gönderilmekte ve sistem tarafından da durum bilgileri yine SMS ile gönderilmektedir. Bu şekilde yapılacak bir haberleşmenin ilk tip cep telefonlarında bile çalışabileceği aşıkarsada yavaş ve maliyetli olması sürekli sistemin uzaktan sergilenmesini zorlaştırmaktadır. Ayrıca mikro denetleyici üzerinde mevcut bulunan sanal seri port aracılığıyla da geliştirmiş olduğu yazılım aracılığıyla PC üzerinden sistemin durumunu sergilemiş ve kontrol etmiştir.

Kardaş (2014) gerçekleştirdiği yüksek lisans tez çalışmasında akıllı ev yönetimini GSM kontrolü ile sağlamıştır. Kullanıcı arayüzü ve sistemin veri alış verişi için haberleşme protokolü olarak UDP kullanmıştır. AKYÖN projesi kapsamında geliştirilen Beagle Bone Black gömülü bilgisayar sistemine sahip ikinci prototipte ilk önce haberleşme protokolü olarak UDP tercih edilmiş olup yapılan testlerde paket kayıplarının yaşandığı ve bu kayıpların denetiminin yapılamadığı, bunu gidermek için kullanılacak yöntemlerin sağlıklı olmadığı ve haberleşme maliyetini arttırdığı için TCP/IP haberleşme protokolünün kullanılmasına karar verilmiştir.

Sevil (2015) gerçekleştirdiği yüksek lisans tez çalışmasında kablosuz sensör ağlarının tasarımına yoğunlaşmış olup, Zigbee, RF ve IR yöntemlerini kullanarak sensör verilerinin toplanması ve kontrolünü sağlamıştır. Ayrıca çalışma kapsamında sensör verilerinin toplanmasının ve yorumlanmasının insan hayatına sağlayacağı katkıları çalışmasında paylaşmıştır. Geliştirdiği sistemdeki verilerin takibi ve yük kontrolü için C# programlama dilini kullanarak arayüz çalışması yapmıştır. Gömülü sistem ile haberleşmek amacıyla seri haberleşme portunu kullanmıştır. Seri port kullanılmasının iletişim paketlerinin güvenli ve kayıpsız bir şekilde gömülü sistem ile yönetim paneli arasında sağlanmasında sıkıntılara sebep olacağı ve fiziksel kısıtlamalar nedeniyle (56 kbits için maksimum 2,6 metre) gerçek ortamda kullanılmasının sistemin takip ve yönetilmesinin zor olacağı değerlendirilmektedir. Bu sebepten dolayı tasarlanacak Akıllı Ev Yönetim Sisteminin haberleşme alt yapısının TCP/IP gibi paket iletim başarımını kontrol eden, ethernet alt yapısı sayesinde yüksek hızlara (1Gbps) ulaşabilen bir haberleşme protokolünün kullanılmasının sistem başarımını arttıracacağı değerlendirilmektedir.

Kargacı (2015) gerçekleştirdiği yüksek lisans tez çalışmasında ev tipi bir klimayı Android işletim sistemi üzerinde koşan bir mobil uygulama ve PHP tabanlı web

arayüz ile kontrol etmesini sağlamıştır. Haberleşme yöntemi olarak kablosuz ağ teknolojilerini kullanmayı seçmiştir.

Küresel çapta yaygınlaşan akıllı ev yönetim sistemlerini yöneten gömülü sistemlerin mobil cihazlar ile efektif bir şekilde yönetilmesi ve veri transferinin sağlanması amacıyla haberleşme protokolünün gerçekleştirilmesi amaçlanmıştır. Bu kapsamda MİLETS firması tarafından donanımsal olarak gerçekleştirilen Akıllı Ev Yönetim Sistemini (AKYÖN), üzerinde açık kaynak kodlu Android işletim sistemi çalışan mobil cihazlar yerel ağ ve geniş alan ağları üzerinden yönetmek amacıyla Java programlama dili kullanılarak yönetim arayüzü ve haberleşme protokolünün gerçekleştirilmesi amaçlanmıştır. Ayrıca geliştirilecek bu yönetici arayüzü aracılığıyla kullanıcının bazı kurallar çerçevesinde sisteme kolay bir şekilde yeni bir sensör veya eyleyiciyi entegre edebilmesi hedeflenmiştir.

1. AKILLI EV HAKKINDA GENEL BİLGİLER

1.1. Akıllı Ev Kavramı

Ev; insanın barındığı, içerisinde fiziksel ve sosyal ihtiyaçlarını giderdiği alandır. Kişinin hayatı ve yaşam tarzı ile özdeşleşen evler, bireylerin bedensel ve ruhsal sağlığını, huzurlarını ve sosyal hayatlarını etkileyen temel etkenlerin başındadır (Akdur, 1998).

Günümüz evlerinde tek kontrol kumandası üzerinden kontrole olanak sağlayan ve eve özgü yazılımlar aracılığıyla bütün kontrolleri yerine getiren işlevsel otomasyonlar ev sahibinin hizmetindedir. Bu sistemleri geliştirme sürecinde, sistemin bilgisayar uyumlu olmasına özen gösterilmiştir. Günümüzde ofislerde kullanılan bilgisayarlar, ortak bir ağ bağlantısı ile birbiriyle iletişim içindedir. Bu iletişim sonucu ofis içinde ya da dışında otomasyon sağlama imkânı sağlar (Küçükbakırcı, 2006).

Otomasyon; sistemi hazır ve belli bir senaryoya bağlı olan ve bir operatör ihtiyacı bulunmadan istenilen eylemi yapan sistem olarak tanımlanabilir. Ev teknolojileri sanayisinde kullanılan otomasyonlar, gündelik hayatımızda da ev otomasyonu sistemi olarak karşımıza çıkar ve tamamen kişiselleştirilmiş istek ve ihtiyaçlara uyumludur. Akıllı ev ise bu teknolojilerin tümünün sağlayacağı, ev sakinlerinin ihtiyaçlarına cevap verip hayatlarını kolaylaştıran ve güvenlik, konfor ve yüksek tasarruf imkanı sunan yapılar olarak tanımlayabiliriz. Akıllı evler dış dünyadaki değişim ve gelişmelere tepki vererek ev sakinlerinin davranış ve alışkanlıklarından hareketle neyi ne zaman yapacaklarını kestiren bir yapı tipidir (Göktaş, 2006).

Otomasyon, uzun yıllardır birbirinden farklı alanlarda kullanılır ancak son yıllarda pratik çözümleriyle konut projelerinin göz bebeği olmuştur. Rutin işleri otomatik hale getiren akıllı evlerin avantajlarını gören firmalar, konu üzerine çalışmalar başlatmış ve çalışmalar sonucunda olumlu noktalara ulaşmışlardır (Bayram, 2006). Ev teknolojileri bağlamında, otomasyon ve kontrol sistemleri kayda değer bir

ilerleme kaydederek günlük hayatın vazgeçilmezi olmuşlardır. Otomasyon ve kontrol sistemleri üç aşamadan geçerek son halini alır. Bu üç aşamayı şöyle sıralamak mümkündür:

- Birinci aşama: İhtiyaçlar göz önüne alınarak bu ihtiyaçları karşılayacak cihaz ve donanımların geliştirilmesi ve günlük hayatta kullanılabilir hale getirilmesi şeklindedir.
- İkinci aşama: Geliştirilen cihazların bir kumanda ile kontrolleri ve zamanlayıcı kullanılması, kullanıcılara alternatif birçok seçenek sunması şeklindedir.
- Üçüncü aşama: İhtiyaçlar dahilinde tüm cihazların tek kumandanın ya da uzaktan iletişim ile kontrol ve programlanması şeklindedir.

Son yirmi yılı ele aldığımızda gördüğümüz teknolojik gelişmeler, insan hayatında da mekan tasarımı alanında büyük değişimlere neden olmuştur. Enformasyon teknolojisinde yaşanan hızlı gelişim ve değişimin sonunda, iletişim teknolojilerinin bilgisayar altyapıları ve bileşimleri vasıtasıyla farklı sistemlerin kontrolünü yapabilmesi, dışarıdan gelebilecek verileri algılayıp bu verilere göre tepki ve kararlar verebilen bir sistemin ortaya çıktığı görülerek bu sistemi akıllı ev dediğimiz evlerde kullanmak mümkün olmuştur. Akıllı ev bir çok alt bileşeniyle entegre biçimde senkronize duruma getiren bir kumanda ile kontrol edilen kompleks bir sistem şeklinde açıklanabilir (Seçer, 2006).

Farklı bir açıdan bakıldığında, gelişmiş şekliyle genel geçer olarak yapının tümünü bütün işlevsel fonksiyonları ile kontrolü mümkün halde olan ve kontrol sistemlerini bünyesinde barındıran evler olarak akıllı evi tanımlayabiliriz. The Intelligent Building Institution(TIBI)'nin akıllı ev tanımı ise; “bazı sistemlerin birbirine koordineli halde kullanılarak yüksek performans alma, düşük maliyet ve esneklik sağlamayı azami seviyelere getiren yapılar” şeklindedir (Kongaz, 2007).

Bir yapıyı akıllı ev statüsüne koymak için, standart parçalar ve bazı hazır sistemlerin var olması gerekmektedir. Bunun yanında bu ürünlere ihtiyaç ve gereklilikler dahilinde yeni parçalar eklenilip çıkarılabilmelidir. Hazır sistemlerde genelde; kontrol kutusu, kontrol paneli ve algılayıcılar, denetleyici parçalar, kumandalar ve mobil modüller mevcuttur. Her biri bir kitap büyüklüğünde olan kontrol panelleri

akıllı evlerin girişinde bulunur. Kontrol kutusu ile algılayıcılar ve cihazın denetleyicileri, tadilata ihtiyaç duymayacak şekilde kablosuz ya da şebeke üzerinden haberleşme imkanı veren çeşitlerden biri olduğu gibi enerji kesintisinden etkilenmemesi için kablolu modellerin de seçilebilirliği mümkündür (Gerhart, 1999).

Harrison ve diğerleri (1998) 1980’li yıllarda akıllı binaların özelliklerini şöyle anlatır:

“1980’lerin başında kurumsal sistemlerin çekirdeğini merkezi bir bilgisayar oluşturuyordu. Tarih ilerledikçe minimal bilgisayarlar kurumsal olarak daha yaygın tercih edilmeye başlandı. Yapıların otomasyon sistemleri bu değişimden ve yeni sistemden faydalanmaya başladı. Bu zamanın akıllı bina ve akıllı ev tanımı teknolojik gelişmelerle öne çıkan yapı otomasyonu ve iletişim ve ofis otomasyonu üzerine kuruluydu.”

Akıllı evlerin sistem gelişimi sırasında dikkat edilen temel etkenlerden biri, sistemin bilgisayarlara birebir uyum sağlaması olmuştur. Akıllı bina teknolojisinde yeni adım, kişisel bilgisayarların ev yaşantısını da değiştirmesi oluşturur. Bu değişim ev içindeki teknolojik sistemlerin bilgisayar ile kontrolü olarak tanımlanır (Gül, 2010).

GSM şebekelerinin kurulması ve mobil teknolojik cihazların kullanımının yaygınlaşması, ev otomasyon sistemlerinin de bu şebekelere eklenmesi fikrini ortaya çıkarmıştır. GSM bünyesinde yapılacak kontrolü iki türlü yapmak mümkündür. Bunların birincisi mobil cihazlar aracılığıyla kullanılan bir sistem edinip telefonun tuşları aracılığıyla sistemi kontrol etmek, ikinci yolu ise GSM üzerinden mobil cihaz ile kısa mesaj göndererek evdeki cihazları kontrol etmek olarak tanımlanır. Sıklıkla kullanılan ilk yöntem olduğu söylenebilir. Pek çok uygulamayı birlikte sunan çeşitli sistemlerin de olduğu bilinir (2005; Adao ve ark., 2008; Khiyal ve ark., 2009; Cincirop, 2009).

Akıllı ev denince akla gelen ve kullanımı yaygın olan cihazlar şunlardır:

- Klimalar: Ev sıcaklığını istediğiniz uygun bir değere getirerek sabitler.
- Çamaşır Makineleri: Çamaşırlarınız için uygun su seviyesi, uygun olan yıkama programı ve başlatıp bitirme seçenekleri ile dikkat çeker.
- Buzdolapları: Gıda maddelerinin uygun şartlarda muhafaza edilmesi, kontrollü bir şekilde tüketiminin sağlanması.

- Bulaşık Makineleri: Bulaşıkların fazlalığına göre uygun program ve su miktarının ayarlanması, deterjan ve parlatıcı oranının ayarlanması.
- Kurutucular: Yıkanan çamaşır miktarı ve çamaşırın ıslaklık derecesine göre ısı belirlenmesi.
- Fırınlara: Pişirilecek gıda maddesine göre uygun süre ve sıcaklık ayarlaması.
- Elektrikli Süpürgeler: Ev içindeki toz oranına göre yeterli sürede çalıştırılma (Tosunoğlu, 2009).

Artık akıllı ev kullanımı bir lüks değil, bir ihtiyaç halini almıştır. Harris ve Wigginton (2006), akıllı binaların ihtiyaç açılığını şöyle anlatır “Akıllı bina kavramı, giderek artan ve karmaşıklaşan, beraberinde karmaşık bina formlarını ve içeriklerini getiren konfor ihtiyaçlarının giderilmesi alanında var olmaktadır. Geleneksel anlamdaki bir bina, içine entegre edilmiş fiziksel çevre kontrol sistemleri olmadan durağan ve cansız bir yapıya sahiptir. Yapı; yapısal güçlerin ve ısının yarattığı gerilimler sonucu çok az bir miktarda hareket edebilir. Durağan ve tepkisiz yapının gerekli konfor koşullarını yaratmadaki yetersizliği, fiziksel çevre kontrol sistemlerinin bu eksikliğin üstesinden gelmek için geliştirilmesine sebebiyet vermiştir.”

Akıllı evlerden beklentiler beş ana maddede sıralanabilir:

- Yüksek güvenli ve emniyetli olması,
- Konforlu bir yaşam alanı olması,
- Hem basit hem kolay kullanılabilir olması,
- Kullanılan sistemlerin enerji tasarrufu sağlıyor olması,
- Fiziki engelleri olan bireylere uygun şartlar sağlanması (Şahinoğlu, 2006; Aslan, 2014).

Ev içi çalışmaya uygun ortam, İngilizce ya da günümüzde kullanılan “Home Office”, akıllı evlerde olmazsa olmaz denebilecek bir özelliktir. Özellikle büyük şehirlerde ulaşımın büyük bir sıkıntı olmasından kaynaklı, iş yerine gidip eve dönmek için yollarda uzun saatler geçmektedir. Home Office çalışma durumunda hem çalışan enerjisini koruyacak hem de zamandan büyük tasarruf sağlanacaktır. Zamanın ve enerjinin ne kadar önemli olduğu düşünüldüğünde Home Office çalışma ortamı,

akıllı konutların olmazsa olmaz ana özelliklerinden biridir. Evden çalışma da akıllı evler sayesinde sağlanabilir (Geçim, 2002).

Akıllı evlerin tasarımında kişi ve nesne grubu arasındaki ilişki ve iletişimin belirlenmesi çok önemlidir. Akıllı ev tasarımcıları, sosyal gözlem yetenekleri ile kişiler ve nesnelere arasındaki ilişkileri ya da daha sade bir anlatımla, kişi grupları arası sosyal ilişkileri, kişiler arası iletişim türlerini ve birbirlerine davranışları üzerine çalışmalarına devam etmektedir. Bunun amacı sistemin yeniden tanımlanması ve değerlendirilmesidir. Tasarım ve teknolojinin ortak öznesi olan kişi, nesne ve sistem arası etkileşim ve iletişimin yeniden tanımlanması akıllı evlerde iletişim sistemlerinin geliştirilmesi açısından oldukça önemlidir (Türkcan, 2007).

1.2. Akıllı Ev Kavramının Tarihçesi

Akıllı bina ya da akıllı ev kavramı 1980'li yıllarda ilk olarak Amerika'da ortaya çıktı. Washington'da bulunan Akıllı Bina Enstitüsü, akıllı binayı şöyle tanımlamıştır: "Kullanıcıların performansını, ilk yatırım ve işletme maliyetlerinde tasarrufu ve esnekliği en üst düzeye çıkarmak için çeşitli sistemleri entegre eden binadır" (Avinçan, 1999; Seçer, 2006).

1990'lı yıllar başladığında akıllı ev kavramı birden unutuldu, ancak 90'ların sonunda yeniden popüler olmaya başladı. İlk yıllarında henüz gelişmemiş olan teknolojiye bağlı olarak akıllı evlere olan talep yetersizliği girişimi başarısız kıldı. Ancak günümüz olan 2010'larda akıllı evler bilgisayar tamamlayıcısı olarak kabul edilmekte. 1996'dan 2000'e kadar küçülen dünya ekonomisi, 2001 ekonomik krizi sonrası doruğa çıktı ve bu durum akıllı ev ve ev teknolojilerinin ilerlemesini yavaşlattı. Akıllı ev ve akıllı bina pazarı 2002'de yeniden doğrulamaya başladı ancak akıllı ev pazarında en büyük sıçrama 2005'te meydana geldi (Tosunoğlu, 2009).

Akıllı bina ve akıllı ev bağlamında yaşanan gelişmelerin devamında; Ekim 2002'de, Honeywell, Intel, Compaq, Microsoft, Mitsubishi ve Philips gibi köklü ve güçlü firmalar ev aletlerinin kullanımını basite indirgemek için HOME API (Uygulama Programlama Arabirimleri) isimli bir standart oluşturdu. Bu projenin amacı ev otomasyon araçlarını kontrol etmek olarak tanımlanabilir. Özetle diyebiliriz ki,

evdeki diğer elektronikler tıpkı bir bilgisayar gibi “tak ve kullan” özelliğine bürünecektir. Yani uygulama ve kullanılan otomasyon sayesinde herhangi bir ev aleti tanımlanacak ve sistem içinde kullanımı sağlanarak kolaylaştırılacaktır (Aslan, 2014).

1.3. Dünyada Akıllı Ev Kavramı

Akıllı Ev ve akıllı binalar, yaşayanların hayatını maksimum rahatlatan, güvenlik bakımından en üst düzeyde olan ve konforlu olarak tanımlanabilir. Zaten var olan elektrik kabloları ile elektronik cihazların kontrolü akıllı evlerde kolayca sağlanır. Akıllı evlerde mevcut ısı ve nem durumu, ayrıca evin diğer durumları zekice kontrol altına alınmalıdır. Akıllı evlerde ve akıllı binalarda, bina sakinlerine bütün elektrikli aletlerin kontrolünü elde tutabilecekleri olanaklar sağlanır. Akıllı ev sakinleri daha önce manuel olarak kontrol ettikleri işlerden kurtulur (Douligeris, 1993; Güğül, 2008).

Dünya devi Microsoft, akıllı ev piyasasındaki büyümeyi görerek ve gelecek vaat ettiğine ikna olarak 2003 yılının ortalarına doğru proje için 7 milyar dolarlık bir bütçe belirlemiştir. İlk olarak Microsoft’un tasarladığı akıllı evler, örnek bir proje olmasına karşın, 5 yıl gibi bir süre içinde hayata geçirilen bir proje özelliğine sahiptir. Akıllı evlerin giriş kısmında bir ekran sizi karşılar. Bu ekran kullandığımız bilgisayar ekranlarına benzer; kişiselleştirilebilir, farklı resimler tercih edilebilir ve dokunmatik ekrana dokunarak ev sahiplerine sesli mesaj bırakılabilir. Ev sahibi tercihine göre eve döndüğünde bu mesajı dinler ya da mesaj bırakıldığı anda elektronik posta olarak kendisine iletilir. Akıllı evlerin kapılarında kapı kolu yoktur; kapıyı açmak için akıllı kart kullanılır ya da retina taraması ile otomatik açılan kapıdan içeri girilir (Küçükbakırcı, 2006; Tosunoğlu, 2009).

Akıllı Ev, yepyeni bir sistem olduğu gibi büyük bir hızla gelişip ilerlemektedir. Öyle ki akıllı ev olgusuyla ilgili yeni kablosuz sistemler geliştirilip piyasadaki yerini almaya başlamıştır. Bir diğer teknoloji devi IBM, Windows altında yeni bir ev otomasyon sistemi olan “Home Director”ı başlattı. Bir başka teknoloji şirketi olan EIB tarafından Avrupa’da piyasaya sunulan ‘Home Assistant’ı, akıllı ev teknolojileri için önemli bir yere koyabiliriz. Türkiye’deki gelişmelere baktığımızda bazı

güvenlik, aydınlatma ve ısı tesisat firmaları kendi otomatik kontrollü cihazlarını Akıllı Ev için otomasyon sistemi olarak adlandırır. Fakat, bütün bunlar birbiriyle irtibatlı olmayan eski birer kontrol noktasından oluşmaktadır.

Akıllı Ev sistemi kesinlikle yapılması kolay olmayan karmaşık bir sistem olup, kontrolün tek kumandadan sağlandığı göz önüne alınarak elektroniklerin çalıştırılmasından güvenliğe kadar her detayın hem en ekonomik hem de en konforlu haliyle yapılması gereken bir projedir (Aslan, 2014).

Japonların geliştirdiği akıllı ev sistemlerinde, ev sahibinin yokluğu neredeyse hiç hissedilmiyordu. Japon işi akıllı evlerde ev sahibi yokken de yapılması gereken her şey tıkr tıkr işleyen bir şekilde yapılıyor. Japonların yaptığı akıllı evler Tokyo’da bir elektronik fuarında meraklı gözler karşısına çıktı. Japonların geliştirdiği bu evlerde her cihaz cep telefonu vasıtasıyla çalışıyor. Örneğin telefonda verilen bir talimat ile bahçe sulaması ya da köpeğimizin karnının doyması hayal değil. Bunun yanında eve gelmiş olan postaları alan otomatik bir kutu “alındı” belgesini de otomatik olarak veriyor. Ev sahibi yokken eve gelenlerin fotoğraflarını çeken bir kamera ise girişteki yerini alıyor (Göktaş, 2006).

Akıllı ev projesi için adım atan Ericsson ve Sun Microsystems, akıllı evlerde bulunan elektronik cihazlarınıza dışarıdayken de ulaşabileceğiniz ileri teknoloji ürünü çalışmalarına hız verdi. Bunun ilk örneği de Ericsson ve Elektrolux arasında gelişen işbirliği oldu. Buna göre evde değilken buzdolabınızı arayıp domates kalıp kalmadığı sorabiliyorsunuz. Bu tasarıma baktığımızda dolap kapağına yerleştirilecek bir dokunmatik ekran, internet bağlantılı bir bilgisayar ve “e-box” adı verilen bir cihaz bulunuyor. Ericsson’un yapmış olduğu çeşitli marka ortaklıkları sonucu ortaya çıkan e-hizmet, gündelik yaşamın bir parçası olan davranış ve faaliyetlere yardımcı oluyor. E-hizmet ile temizlik, çamaşır yıkama gibi gündelik işlerin yapılmasını, yaşlı veya hasta bakımında daha özenli olunmasını ve hatta çocukların okula bırakılıp okuldan alınmasını sağlıyor (Aslan, 2014).

1.4. Türkiye’de Akıllı Ev Kavramı

Her geçen gün gelişen teknoloji ile hem kullanılan teknolojik cihaz sayısı hem de kontrol ünitelerinin sayısı artmıştır. Artık çıkarılan yeni ürünler yapı ve tasarım gereği uzaktan kontrol edilebilir olarak üretilip uygun cihazlarla kontrolleri sağlanan cihazlardır. Aydınlatmadan ısıtmaya, acil duruma kadar otomatikleşme tüm binaların kontrolünün sağlanması açısından büyük önem arz etmektedir. Gelecekte kullanımı artacak ve daha da gelişecek olan akıllı binalar ile birlikte önemi daha da artacaktır (Bekçibaşı ve Tenruh, 2011).

Ses teknolojileri alanında hizmet veren bir şirket olan GVZ, CeBIT Fuarı’nda, Arçelik ve CoreNet iş birliği ile birlikte GVZ’nin konuşma tanıma - konuşma sentezleme teknolojilerinden faydalanılarak geliştirilen Akıllı Evini sergileyip Türkiye’de bir ilki gerçekleştirdi. Söz konusu akıllı ev, kendi sahibinin sesini tanıyor ve sahibinin sesinden verilen komutları uygulayabiliyor. Akıllı Evler ana sistemde bulunan bir sunucu aracılığıyla yönetiliyor. Kablosuz teknolojili bir mikrofon ile sesli komut evin içinde herhangi bir yerden çeşitli noktalara yerleştirilen alıcılar aracılığıyla sisteme iletiliyor. Sisteme müzik dinleme, film izleme, ışık ayarlama gibi bazı modlar yükleniyor ve ev sahibi ışıkları istediği noktadan ayarlayabiliyor. Akıllı evin sadece tanıdığı seslere kapı açma özelliği sayesinde hırsızların girişimini de engelliyor (Küçükbakırcı, 2006).

Türkiye’nin bilinen ilk akıllı evi, CompeX fuarında sergilenmiş olup, dünya üzerindeki örneklerinden büyük farkları bulunmaktadır. Öyle ki bu farklar diğer akıllı evlerde kullanılan elektroniklerden ziyade piyasada bulunması daha kolay olan cihazlarla kurulmuş olmasıdır. Söz konusu akıllı ev, IBM Global Services “Smart Home” ağ altyapısı ile oluşturulmuştur. Bu akıllı evin fiyatıysa 119 milyon olarak saptandı (Göktaş, 2006).

Akıllı evin altyapısı sayesinde, evde çıkabilecek yangın, eve giren hırsız gibi güvenlik gerektirecek ihtiyaçları kontrol edilebilir durumdadır. Eve bulunan akıllı prizler, telefon ve internet erişimi gibi sistemler vasıtasıyla eve yeni eklenen dijital cihazların sisteme dahil olması birkaç dakika sürmektedir.

Akıllı evin yaşayan ev olmasını sağlayan en önemli özellikleri; otomatik olarak evin ışıklarını yakan, televizyonu açan, kahve makinesindeki kahveyi hazırlamaya başlayan bir sistem olarak sıralayabiliriz. Akıllı evin temel özelliğinde olduğu gibi bu sistemin de uzaktan kontrol ağırlıklı olduğunu söyleyebiliriz (Tosunoğlu, 2009)

Rezidans olarak yapılan binaların otomasyonda bir adım önde olduğu görülüyor. Bunun yanında müstakil tarzda yapılan yapıların da son yıllarda enerji tasarrufu ve güvenlik için akıllı ev ve akıllı bina tarzı sistemlere ilgi duyduğu görülmeye başlanıyor. Türkiye ve Avrupa karşılaştırıldığında Avrupa'nın enerji tasarrufunu kavramış olduğunu ve Türkiye'de ise durumun daha çok konfor öncelikli olduğunu söyleyebiliriz. Avrupa'nın akıllı ev teknolojileri ile Türkiye'yi kıyasladığımızda ise Türkiye'nin Avrupa'nın yüzde 5'i olduğu söylenebilir (Şahinoğlu, 2006).

1.5. Akıllı Ev Yönetim Sistemleri

Akıllı evlerde bulunan sistemler teknolojinin nimetlerini kullanarak ev sakinlerinin yaşam kalitesini artırmak ve bunu yaparken tasarrufu sağlamak amacıyla kullanılır. Akıllı evden önce inşa edilen evlerde mevcut olan sistemler; ısınma, aydınlatma gibi işlevleri sağlayan cihazları kontrol etmeyi içerir. Ancak gelişen ve değişen teknoloji ile bir ev için temel olan bu işlevlerin yanında çok daha zengin bir hizmet skalası beklenmektedir. (örneğin telefon çaldığında müziğin sesini azaltmak gibi) (Menteşoğlu ve ark., 2015).

Akıllı olarak dizayn edilen evler ve binalar hem üretkenlik hem de enerji tasarrufu hem de kazanç bakımından ciddi bir artış sağlamaktadır. Akıllı ev ve binaların amacı insan hayatını en konforlu şekilde geçirmesini sağlayarak kolaylaştırmaktır. Üstelik akıllı sistemlerle engelli bireylerin de hayatı en konforlu ve en rahat şekilde geçecektir (Işılak ve Baydere, 2011).

Teknolojinin her geçen gün gelişmesi ve gelişen teknolojinin kullanılmak istenmesi akıllı ev teknolojilerinin de hızla gelişmesini sağlamaktadır. İhtiyaç ve gereklilikler doğrultusunda akıllı evler ve otomasyon sistemleri şekillenmektedir. Akıllı ev teknolojisinde sistem olarak yapılabilecekleri şöyle sıralayabiliriz;

- Sızıntı oluşturabilecek su ve gazın sızıntı oluşturması durumunda vana kapatılıp kullanıcı ile iletişime geçilerek haberdar edilebilir,
- Uzaktayken eve kameralar aracılığı ile bağlanılarak evin güvenlik durumu kontrol altında tutulabilir,
- Panjurların otomatik sistemi sayesinde evde yokken dahi ev aydınlanabilir. Uzaktayken evdeki ışıklar kontrol edilebilir, kapalıysa açılabilir, açıksa şiddeti ayarlanabilir.
- Perdeler otomatik olarak kapatılabilir ya da kişiler kendi kontrollerinde bunu sağlayabilir,
- Klima veya kombi ısıyı kontrol edilebilir, oda sıcaklığı uzaktan kontrol edilebilir. Herhangi olumsuz bir durumda kullanıcı haberdar edilebilir,
- Bahçe sulaması için topraktaki nem durumu kontrol edilerek yapılabilir, nem durumu ev sahibine aktarılabilir,
- Prizde unutulmuş eşyaların prizlerinden elektrik kesilebilir, çocukların ulaşabileceği prizlerin elektriği kesilerek güvenlik sağlanabilir (Koçak ve Kırbaş, 2016).

Akıllı ev yönetim sistemleri karakteristik özelliklerine göre, kontrol edilebilir evler, programlanabilir evler, yapay zekaya sahip evler, akıllı ve çevreci evler ve yaşlı ve engelli bireyler için akıllı evler olarak beş gruba ayrılmaktadır.

1.5.1. Kontrol edilebilir evler

Bu tip evler, evde bulunan cihazların kumanda gibi çeşitli sistemler ile kolayca kontrol edildiği evlerdir. Programlama yapmadan ev ile iletişim söz konusu değildir(Güçül, 2008).

Cihaz sistemlerinin el çırpma, ses ya da uzaktan kumanda işlevi gören bir cihazla kontrol edilebildiği evlerdir (Erbey, 2016).

Kontrol edilebilmesi çeşitli cihaz ya da beşeri yöntemler kullanılan bu evler büyük bir kolaylıkla kontrol edilebilirlik sağlamaktadır. Kontrol edilebilir evlerde muhakkak bir programlama yapılmalıdır, programlamadan etkileşime girmek mümkün değildir. Kontrol edilebilir evler yalnız o an verilen komutlarla kontrol

edilebilmektedir. Örneğin odaya girince ışıkların yanması ya da ses ile perdelerin açılması gibi (Aslan, 2014).

Kontrol edilebilir evler, kendi içinde üç ana grupta toplanabilir:

- Evdeki tüm elektroniklerin uzaktan kumanda ile kontrol edilebildiği evler: Tüm elektronik cihazların bir uzaktan kumanda ile idare edildiği evlerdir. Kumanda bir yüzük ya da bir küpe bile olabilirken fonksiyonel bir kumanda da olabilir.
- Farklı cihazların bağlantılı olduğu evler: Evde bulunan elektronikler tek bir ana kumandadan kontrol edilmese de birbirine mümkün olduğunca bağlanmıştır. Böylece tüm cihazlar birbiriyle uyumlu çalışabilir ve cihazlar arası veri aktarımı sağlanabilir.
- Görünmez bir ünite ile kontrolü sağlanan evler: Seslere ya da hareketlere duyarlı sensörler vasıtasıyla kurulmuş bir kontrol mekanizmasının olduğu evlerdir.

Bu tipte olan evlerde cihazlar kumanda ile kontrol edilebildiği gibi el çırpınca perdelerin açılması gibi komutlarla da kontrol edilebilir. Diğer bir deyişle ev o an verilen komutu uygular (Gül, 2010)

1.5.2. Programlanabilir evler

Programlanabilir evler; ev sahibinin anlık değil ileri bir saatte ya da ileri bir tarihte olan isteklerini yerine getiren ve koşullara tepki veren evlerdir. Kendi içinde ikiye ayrılan programlanabilir evler, zamanlama ayarıyla ya da koşul ve duruma göre eyleme geçen ve duruma tepki verebilen evlerdir. Örneğin sabah olunca perdeleri açan, akşam saatlerinde ışıkları yakan evler gibi (Erbey, 2016).

Programlanabilir evler, kontrol edilebilir evlere göre bir adım daha ileride olan evlerdir. Ev sakinlerinin daha önce yapmış olduğu eylem zincirine dayalı tepkiler veren bu evlerde kullanılan yazılım muhakkak güvenli olmalıdır. Tek handikabı farklı bir tepki istendiğinde sisteme girilmesi gerekliliğidir. Ev sakinlerinin memnun olmadığı bir tepkiyi değiştirmek için evde yaşayanların bunu sisteme kaydetmesi zorunludur. Bu evlerde kullanılan yazılımlara mühendisler kolaylıkla müdahale edebilirken ev sahipleri için bu durum oldukça zordur (Gül, 2010).

Programlanmaya uygun olan bir otomasyon sistemi, bahsedilen kontrollerin tamamını, bir kısmını ya da daha fazlasını gerçekleştirebilir.

- Aydınlanma gereçleri hareket olan bir noktada otomatik yanar, hareketsiz ve kimsenin olmadığı yerlerde otomatik söndürülür.
- Evin veya suyun ısı derecesi ev sahibi uyanmadan, işten eve dönmeden istenilen seviyeye getirilir.
- Bahçe varsa sulaması belirlenmiş aralıklarla ve yağış olmadığı zamanlar gerçekleştirilir.
- Ev sahibi yatarken ışık ve elektronik cihazlar kapatılır, uyunan odanın ışık seviyesi düşürülür, ısınma cihazı ekonomik moda çalışmaya başlar, güvenlik bölgelerinde alarm devreye sokulur ve müzik bir saat sonra kapatılır.
- Sadece bir komutla perdeler kapatılır, ışıklar kısılır, mısır patlatma cihazı çalıştırılır, telefonlar sessiz duruma getirilip DVD player çalışır ve bir ev sineması keyfi yaşanır.
- Olası bir yangın durumunda, ev sahibi evdeyse ev içi siren eğer evde değilse telefon ile uyarılır. Akıllı ev itfaiyeyi arayıp gaz akışını keser ve havalandırmayı kapatır. Bazı ışıklar yanar ve çıkış kolaylaştırılır.
- Evden çıkarken tek tuş ile bütün cihazlar kapatılır, ısınma cihazı ekonomik moda geçirilir, kişi evden çıktıktan bir süre sonra alarm sistemi devreye girer.
- Ev içinde gerçekleştirilen komutlar uzaktayken telefonla da gerçekleştirilebilir.
- Çocuğunuz okuldan eve geldiğinde telefon ile bilgi verilebilir.
- Evde bulunan elektronik cihaz ve ışık sistemi bir uzaktan kumanda aracılığıyla kontrol edilebilir.
- Eve bağlı olan ışıkların yanma ve söndürülme saatleri gün batımı ve gün doğumuna göre ayarlanır.
- Evde bulunan bütün lambaların parlaklık seviyesi ayarlanabilir.
- Perde ve güneşliklerin kontrolü uzaktan kumanda aracılığıyla sağlanabilir.
- Sabahları belirlenen saatte ya da günün herhangi bir saatinde kişiyi uyandırır.
- Çocuk odalarındaki televizyonlar her gün belirlenen saatte kapatılabilir.
- Sabah uyanıldığında ya da günün istenilen bir saatinde taze kahvenin hazır olması sağlanabilir.
- Telefon geldiğinde televizyon açıksa ya da müzik çalıyorsa ses kısılabilir.

- Telsiz telefonlar tüm evi kontrol edecek bir kumandaya dönüştürülebilir.
- Yatağa bağlı ya da kısıtlı hareketi olan kişiler uzaktan kumanda ile çevresini çok daha rahat kontrol edip uyum sağlayabilir. Gerekiyorsa yardım çağırabilir.
- Ev sahibi tatildeyken ya da evde yokken eve yaklaşan şüpheli biri olduğunda ev doluymuş izlenimi verilebilir.
- Yine şüpheli bir durumla karşılaşıldığında evin iç ve dış ışıkları tek bir düğme yardımıyla açılabilir.
- Belirlenen bölgelerde alarm açılabilir ya da devre dışı bırakılabilir.
- Elektrik tüketimi fazla olan çamaşır makinesi gibi cihazların sayaçlarla uyumlu olarak indirimli saatlere denk getirilip çalışması sağlanabilir.
- Lamba ve ışıkların parlaklığı yüzde 90'a getirilerek ampul ömrü iki kat uzatılabilir ve enerji tüketimi yüzde 30 azaltılabilir.
- Uzun süre evde olunmadığında panjurlar ve perdeler açılıp ev güneşlendirilebilir ya da havalandırılabilir (Bayram, 2006).

1.5.3. Yapay zekâya sahip evler

Yapay zekâ kavramı “zekâ” kavramından yola çıkarak zekâ yapısı teorisi üzerinden geliştirilmiş algoritmaların, bilgisayar yazılım ve donanımlarına uygulanıp düşünce üretme ve sistem oluşturma çalışmalarının bütünüdür. Daha basit anlatımla programlanan bir bilgisayarın düşünme eylemidir. Farklı bir tanım yapay zeka makinelerin henüz yapamadıkları olarak yapay zekâyı nitelemektedir(Göktaş, 2006).

Yapay zekâyı sahip evler, ev sakinlerinin yaşam standartlarını ve yaşam stilini taklit ederek kendi sistemini oluşturur. Ev sahibinin sürekli yaptığı işleri, hangi durumda hangi işi yaptığını kendi zekâsına kodlayarak duruma uygun tepkiyi verir. Ancak günümüzde uygulamaya geçememe sebebi olarak insanoğlunun değişken ruh hali ve karmaşık duygu dünyası olarak tanımlanabilir (Erbey, 2016).

Programlanabilir evler ile benzeşse de programlanabilir evlerin bir tık daha gelişmiştir. Programlanabilir evlerde insanların kurguladığı senaryo sistemleşirken yapay zekâyı sahip evlerde senaryo kendiliğinden oluşur. Yapay zekâyı sahip evlerin en büyük yeteneği öğrenmesidir. Ev sakinini inceler ve ayarlarını kendisi yaratır. Bu evler için gereken şey yapay zekâdır bu da öğrenme yeteneğine sahip

yazılımlarla mümkün olur. Ev sakinlerinin gün içindeki davranışlarını izleyerek sık tekrarlayan hareketleri kodlar ve bu durumda yapması gerekeni belirler. Aynı durum ile karşılaştığında ise aynı hareketi tekrar eder (Aslan, 2014).

Yapay zekâya sahip evlerin bazı dezavantajları vardır. Örneğin insan karmaşık bir varlıktır ve yapay zekâ kendini insan davranışına göre oluşturur. Karmaşık ruh halinin ve insanın her zaman aynı davranamayacağına göz ardı edilmesi yapay zekaya sahip evlerin bir handikabıdır (Güçül, 2008).

Akıllı evlerde kullanılan algoritmaların oluşumunda yapay zekâdan ve yapay sinir ağlarından faydalanılır. Yazılım oldukça zor olan bu algoritmaların sadece akademik düzeyde kalmasına şaşmamak gerekir. Dolayısıyla ticari amaçla yapılan çalışmalar yapay zekâya sahip evlere değil programlanabilir evlere yönelik olmalıdır (Gül, 2010).

Yapay zekâya sahip evler günümüzde hala hayal gücünde kalmaya devam etmektedir. Ne Türkiye’de ne de dünyada bu kapsama uyan bir akıllı ev bulunmamaktadır (Yumurtacı ve Keçebaş, 2009).

1.5.4. Akıllı ve çevreci evler

Enerji verimliliği günümüzde tüm sektörlerin üstünde durduğu önemli bir konudur. Akıllı ev ya da ofislerde enerji tasarrufu önemli bir konudur, hem verimlilik hem maliyet hem de çevre açısından bu durum oldukça önemsenir. Elektrik enerjisinin oldukça pahalı olduğu bugünlerde, akıllı konutlar enerjinin verimliliğini sağlamaktadır (Geçim, 2002).

Yenilenebilirliği en kolay enerji kaynağı olan güneşin zararlı etkilerinin nasıl minimize edileceği yıllar boyu tartışılırken ve bunun için çözüm aranırken ortaya çıkan sonuç güneş panoları vasıtasıyla aktif ya da kumandalı gölgeleme cihazları yoluyla pasif olarak kullanılabilmesi bilinmektedir. Pasif koruma dediğimiz gölgeleme cihazlarının devreye girdiği sistem, akıllı ve çevreci evler için enerji tasarrufunda bir numaralı etkidir (Pehlivan ve ark., 2008).

Akıllı çevreci evlerin akıllı evlerden farkı; hem ekonomik tasarruf hem de çevreye verilen önemin birinci sırada tutulmasıdır. Akıllı ve çevreci evlerde kullanılan fotovoltaik güneş pilleri vasıtasıyla kullanacağı enerjiyi ve fazlasını üretebilir. Güneş pili ya da diğer bir adıyla fotovoltaik pil, üzerine doğrudan gelen güneş enerjisini direkt elektrik enerjisine dönüştürme etkisine sahip olan maddelerdir. Açık ve güneşli bir havada ve uygun şartlarda 1 dm çapında bir güneş pili, aşağı yukarı 1 W elektrik üretebilir. Güneş pilleri; bugün elektrik elde etmek amacıyla alternatif bir enerji kaynağı olarak düşünülüp uygulama alanı bu noktaya kaydırılmıştır (Gül, 2010).

Hızla kirlenen çevre, yırtılan ve gün geçtikçe büyüyen ozon tabakası, atmosferin artık zararlı ışınları süzememesi ve küresel ısınma sonucu meydana gelen iklim değişiklikleri, tüm bireyleri ve insanlığı ilgilendiren önemli bir konudur. Dünya üzerinde konut sayısı diğer bütün binaların ve yapıların sayısından fazladır. Bu da bize konutların inşaat sektörü için önemli ve çevresel değerler açısından değerli olduğunu göstermektedir (Geçim, 2002).

1.5.5. Yaşlı ve engelli insanlar için akıllı evler

Engelli bireyler gündelik hayatta herkesten daha fazla problemle baş etmek zorunda kalmaktadır. İşitme engelliler olası bir hırsızlığı duyamazken, bunama hastalığı olanlar ise gazı açık unutabilmektedir. Engelli insanların evde yalnız kaldığında karşılaşılabilecek olan bu problemlerin WNS teknolojisi ile çözümü mümkün hale gelmiştir (Işılak ve Baydere, 2011).

Yaşlı ve engelli bireylerin hayatını kolaylaştıran akıllı evler; fiziki olarak yardıma ihtiyacı olan ya da yatağa bağımlı ve dışarıdan gelecek yardıma ihtiyacı olan bireylerin karşılaştıkları problemlerle kolayca baş edebilmelerini sağlayan ve bu bağlamda eve yerleştirilen birçok akıllı cihazla engelli bireyin sağlık kontrolünde güvenliği sağlamayı hedefleyen evlerdir (Erbey, 2016).

Akıllı ev fikri özel ihtiyaç sahibi insanlar için uygulansa ve bunun için pek çok akıllı ev türü geliştirilse de günümüzde oldukça yaygındır. Artık ihtiyaç sahibi olsun olmasın her insanın özel ihtiyaçlarını karşılamak ve fiziksel engeli varsa bu engeli

gidermek için tasarlanmıştır. Yaşlı ve engelli insanlar için kurulan akıllı evler, cihazların tür ve düzenlenmesine göre farklılık gösterir. Fiziksel engelli, yaşlı ve hatta görme engelliler için, duyma kaybı yaşayanlar için akıllı evler mevcuttur (Güğü, 2008).

Akıllı evlerde bulunan yaşam destek üniteleri, bireylerin her daim sağlıklı yaşamayacağı ve olumsuz durumlara düşebileceği düşünülerek tasarlanmıştır. Yaşam destek üniteleri, akıllı evlerin muhakkak içermesi gereken bir özellik olarak sıralandırılmıştır. Engelli bireylerin kimi zaman eli, kimi zaman da gözü kulağı olmak için tasarlanan akıllı evler, genel olarak konfor ve lüks ihtiyacı çerçevesinde değerlendirilir. Bu yaklaşım yanlış da olsa akıllı evler engelli olsun olmasın kullanıcıların yaşayabileceği en olumsuz ve kötü durumlara cevap verebilir nitelikte olmalıdır. Aksi halde akıllı ev kavramından söz etmek pek de doğru bir yaklaşım olmayacaktır (Geçim, 2002).

1.6. Akıllı Evlerin Avantajları

Akıllı ev sistemi, ev sakinlerinin gündelik hayatını etkileyen detayları kontrol ederek stresini azaltıp mutlu ve huzurlu bir hayat yaşamasını amaçlar. Bu etkenler ısı, ışıklandırma, nem, konfor ve buna benzer şeylerdir (Kongaz, 2007).

Akıllı evin ana prensibi ev içinde verimliliği artırarak enerji tasarrufuna gidilmesidir. Her evde gereksiz yere tüketilen enerjinin nedenlerinin başında uzun süre ve gereksizce açık kalan ışıklar, klima gibi araçların kısa sürede etki vermesi için yüksek seviyelerde çalıştırılması, etkin bir gün ışığından faydalanamama durumu, evin kullanılmayan odalarının da gereksizce ısıtılması, açık unutulmuş elektronikler gelir. Oysa akıllı ev sistemiyle kontrolü sağlanan bir evin ışıkları gereksiz yere yanmaz ve cihazların kullanımı günün indirimli saatlerine ayarlanarak yüzde 30 ve ısıtmanın programlanması ile yüzde 10 oranlarında bir tasarruf söz konusu olur (Küçükbakırcı, 2006).

Akıllı ev ve otomasyon sistemlerinin güvenlik avantajlarının başında alarm istasyonlarının yanı sıra daha önce tanımlanmış telefon numaralarının da aranabilmesi gelir. Bunların yanında bazı akıllı evlerin astrolojik saat ve seyahat

programlaması da mevcuttur. Bu sayede ev sakinlerinin astrolojik saatine uygun olarak daha konforlu ve güvenli bir şekilde seyahat etmesine olanak tanır. Mevsime göre değişiklik gösteren güneşin doğuş ve batış saatlerine göre akıllı evlerdeki panjurların ve ısınmanın kontrolü özellikle kontrol altına alınmaktadır (Gül, 2010).

Hareket sensörlü algılayıcı cihazlar, kapı ve pencere gibi alanlara yerleştirilmiş olan sensörlerle tüm ev gözetim altında tutulabilmektedir. Akıllı evlerin sağlamayı amaçladığı en büyük güvenlik sistemi ise hırsızlık ve yangın gibi bir durumda alarm sisteminin aksine olayı gerçekleşmeden önleyebilmektir. Siz evde yokken evinize yaklaşan biri olduğunda ev otomatik olarak devreye girerek ışıkları ve televizyonu devreye sokar ve evde biri varmış izlenimiyle yaşanacak olumsuzluğun önüne geçer (Aslan, 2014).

Akıllı Evlerin bu özelliği "Aktif Caydırıcı Etki" olarak adlandırılır ve bu etkiye sahip olmaları, buna bağlı olarak tehlikeyi sezdiği an uzaklaştırmaya çalışmaları, olası bir tehlike anında geleneksel alarm sistemlerinden eksik kalacakları anlamına gelmez. Tüm bu önlemler sonucu en sağlam ve en güvenilir sistemler yine bu sistemlerdir. Bir alarm sisteminin yapmış olduğu ve yapacağı alarm merkezini arama gibi temel işlevleri de vardır (Güçül, 2008).

Bunların yanı sıra, ev otomasyonunun hayatımıza kattığı ve bize sağladığı bazı faydalar şöyledir:

- Işık sensörü sayesinde güneşin batış ve doğuş saatleri algılanarak perdeler açılır veya kapanır. İsteğe göre hangi ışıkların açılacağı belirlenir.
- Gece saatlerinde lambaların insan olan ortamda yanması, olmayan ortamlarda söndürülmesi sağlanır.
- Uyuma vakti geldiğinde evde açık olan tüm ışık ve cihazların otomatik olarak kapatılması, yatılacak odanın ışığının minimuma indirilmesi, ısıtıcının kendini ekonomik moda alması ve istenilen bölgelerde alarmın devreye sokulması,
- Alarmın devreye girmesi durumunda hareket sensörü ile ortamdaki hareketlerin algılanması ve güvenlik ışıklarının yanması,

- Garajda hareket olduğunda ve garaj kapısı açıldığında hareket sensörünün bunu algılayarak evin geçişini aydınlatması,
- Uzun süre uzakta olduğunda evin perdelerinin ve pencerelerinin açılarak evin havalandırılması ve güneşlendirilmesi,
- Duman sensörü yardımıyla yangın anında alarmin çalıştırılması,
- Yangın gibi bir durumda eğer ev sakinleri evdeyse iç siren, evde değilse telefon vasıtasıyla uyarılması, itfaiyeye ulaşılarak gaz ve havalandırma vanalarının kapatılması, aydınlatmaların devreye girmesi ve çıkışa yardımcı olması,
- Evden çıkılırken tek tuş yardımıyla kapatılacak her şeyin kapatılması, ısıtıcının ekonomi modunda çalışmasının sağlanması, aranacak yerin aranması ve evden çıkılınca alarmin devreye sokulması,
- Evde yapılabilecek tüm işlerin ve verilecek tüm komutların telefonla uzaktan da verilebilmesi,
- Çocuklarınız varsa okuldan geldiklerinde telefon ya da mail ile haber verilmesi,
- Evdeki elektronik ve akıllı cihazların ve lambaların uzaktan kontrol edilebilmesi,
- Eve ait tüm ışıklandırmaların gün batımı ve gün doğumu saatlerine göre ayarlanılarak yakılması veya söndürülmesi,
- Evdeki tüm aydınlatmaların yüksek ya da düşüklüğünün ayarlanması,
- Evin ya da suyun ısısının uyanılmadan ya da istenilen bir saatte istenilen seviyede ayarlanması,
- İstenildiği saatte akıllı ev sakinlerinin uyandırılmasının sağlanması,
- Televizyon karşısında uyumaya alışkın olanlar için belirlenen saatlerde televizyonun kapatılmasının sağlanması,
- Sabah uyanıldığında ya da her hangi bir saatte taze kahvenin hazır olmasının sağlanması,
- Telefon ya da kapı çaldığında televizyon veya müzik sesinin kısılması,
- Hareketi kısıtlı ya da yatağa bağlı kimselerin bir kumanda ile hem evi hem de çevrelerini rahat bir şekilde kontrol edebilmelerinin sağlanması ve gerekirse yardım çağırabilmeleri,
- Evden uzakta olduğunda, tatile gidildiğinde eve yaklaşan bir tehlike olduğunda evin doluymuş gibi gösteriminin sağlanması,
- Herhangi şüphe içeren bir durumda tek bir tuşla iç ve dış ışıkların açılması,

- Alarmin istenilen bölgede açık, istenilen bölgede kapalı hale getirilmesinin sağlanması,
- Çamaşır, bulaşık makineleri gibi yoğun elektrik tüketimine sahip cihazların akıllı sayaçlarla entegre biçimde, elektriğin indirimli olduğu zamanlarda çalışmasının sağlanması,
- Ampulleri tam parlak yerine yüzde 90 oranında parlak yakarak hem ampul ömrünü uzatması hem de enerji tüketiminin yüzde 30 oranında azaltılmasının sağlanması şeklinde özetlenebilmektedir (Şahinoğlu, 2006)

Son yıllarını istatistikî değerlerine bakıldığında, fiziki engeli olan ve dışarıdan gelecek yardıma muhtaç olarak hayatını sürdüren birey sayısında artış olduğunu söyleyebiliriz. Bu insanlara yardım edecek birilerini bulmanın problemin temel çözümü olmadığı bilinirken, buna dayalı olarak ortaya çıkan “yaşlılar için teknoloji” terimi Graafman tarafından oluşturuldu. Fiziksel olarak engeli ya da yaşlı kimselerin akıllı ev ihtiyacını anlamak için ortaya iki temel yaklaşım çıkıyor:

- Fiziksel olarak engeli bulunan ya da hareket etmesi zor olan insanların temel ihtiyaçlarına uygun mimari çözümler uygulanması.
- Ev sakinlerinin hayatını kolaylaştıracak özel teknolojiler. Örneğin hareket engelini ortadan kaldırmaya yönelik yardımcı cihazlar (Güçül, 2008).

1.7. Akıllı Evlerin Dezavantajları

Akıllı evlerin genellikle avantajları fazlayken dezavantajları olduğunu da söylemeden geçmemek gerekir. Akıllı evlerde karşılaşılabilecek olumsuzluklar ve dezavantajlar şöyledir:

- Uzaktan erişilebilen sistemler neticesinde evinizin kontrolü başkasının eline geçebilir.
- Sisteminin kontrolü sırasında oluşabilecek aksaklık ve eksiklikler sonucunda çeşitli olumsuz sonuçlar ortaya çıkabilir.
- Nem sensöründe meydana gelebilecek bir hasar sonucu bahçeniz gereğinden fazla sulanabilir, dolayısıyla tasarruf hedeflenirken aslında israf ortaya çıkabilir.
- Ev üzerinde sorumlulukları azalttığı için ev sakinlerini tembelliğe itebilir, bu sayede fazlasıyla sıkıcı bir hayata neden olabilir.

- Ses ile verilen komutlarda sesi algılayamadığı için ortaya bazı aksaklıklar çıkabilir.
- Bir süre sonra otomatige alışan insan mekanikleşebilir (Güğöl, 2008; Erbey, 2016).



2. SİSTEM MİMARİSİ

2.1. Haberleşme

Akıllı bina ve akıllı ev sistemlerinin oluşum sürecinin başlangıcı 1980'lerin başına denk gelir. 80'li yılların başında, bilgi teknolojilerinde ilk adımlar atılmış ve bilgisayarlar insan yaşamında yerini almaya başlamıştı. Bunun yanı sıra otomasyon sistemleri de gün geçtikçe yaygınlaşıyordu. Binalarda ve evlerde yer alan sistemler kullanıcı kontrolünden çıkarak bilgi alışverişi yaparak bunu anlamlandıran ve bunu paylaşan bir yapı haline gelmiştir. Gelişen bu sistem içerisinde iletişim, başlangıcından günümüze gelişimini ve değişimini sürdürmüş ve hala da değişim içinde yer alan bir sistem halini almıştır (Türkcan, 2007).

Telefon hattının uzaktan kumanda olarak kullanıldığı ve hat üzerinden gerçekleştirilen kontrol sistemleri akıllı evlerde, yapıların güvenliğinde, tarım ve hayvancılıkta sulama alanında oldukça sıklıkla kullanılmaktadır. Söz konusu olan sistem, bir telefon hattının kullanımı ile uzaktan kontrol işleminin yapılmasını kolaylaştırır. Bu sistemin en büyük avantajı, sistemi kullanmak, sinyal göndermek ve sinyal almak için yeni bir hat ya da sisteme ihtiyaç duyulmaksızın mevcut telefon hattı üzerinden bütün bunları gerçekleştirebilmektir. Bu durumda haberleşme yapılarından tasarruf sağlanacak ve mevcut telefon alt yapısı sayesinde esnek bir çalışma ortamı oluşturulacaktır (Bekçibaşı ve Tenruh, 2011).

Haberleşme, akıllı yapılar için sistemin kilit parçalarından biridir. Sistemin farklı kolları gibi davranan haberleşme, sistemin ya da kullanıcının göremediği uzak noktalara ulaşır ve bu noktalarda var olan sistemler hakkında bilgi akışını sağlar. Uzaktan kontrolün temelinde haberleşme yatar. Ethernet, İnternet, GPRS, SMS, WIFI, RF, RS232, RS485 gibi haberleşme protokolleri vasıtası ile kullanıcının sisteme ulaşabilmesi ya da ana sistemin uzak noktaya erişimini sağlar (Kongaz, 2007).

Akıllı ev haberleşme ve iletişim sisteminin alt kollarını şöyle sıralayabiliriz:

- Telefon, santral ve yönlendirme sistemleri
- Uzaktan çalıştırma sistemleri
- Uzaktan izleme sistemleri
- Mobil- kablosuz haberleşme sistemleri
- İletişim ağı sistemleri
- Veri işleme sistemleri
- OT/VT sistemleri.

2.1.1. Kablosuz haberleşme

Kablosuz ağlar iş dünyasında genel ve esnek bir alternatif bağlantı olarak kabul görür. Radyo Frekans Teknolojisi aracılığıyla elektromanyetik dalgalarla noktalar arası geçiş sağlanarak eş zamanlı veri transferi yapılmasına olanak sağlar. Kablosuz ağ ile kullanıcılar hızlı ve kolay bir şekilde kaynaklara erişebilir(Kılıç, 2007).

Veri aktarımı ve haberleşme kablolu ya da kablosuz olmak üzere iki farklı şekilde gerçekleştirilebilir. Kablosuz denince akıllara ilk olarak Radyo Frekans haberleşme gelir. Uzun mesafede kablosuz iletişim Radyo Frekans haberleşme aracılığıyla yapılmaktadır (Kardaş, 2014).

Kablosuz ağ ve iletişimin sağladığı avantajları şu şekilde sıralayabiliriz:

- Tüm çalışanların hareket özgürlüğü sağlanarak kabloya bağımlı olmadan istenilen kaynağa kolayca ulaşmanın mümkün olması.
- Kablolama imkanı olmayan durumlarda sürekli olarak iletişim ağının korunması ve iletişimin sağlanması.
- Toplu yaşam alanlarında her noktadan mobil olarak internet erişiminin sağlanabilmesi.
- 11 Mbps veya 54 Mbps gücünde yüksek hızlı link'lerle karasal bağlantıların alternatifini olması (Kılıç, 2007).

Kablosuz ağ iletişinden Radyo Frekansı kullanıldığı için kullanıcı ve ağ arasında geçen iletişim dinlenebilmektedir. İletişim güvenliğini sağlamak için WEP, WPA ve WPA2 gibi bazı şifreleme yöntemlerinin kullanılır ancak yine de bu yöntemler

yeterli güvelliği sağlayamayabilir. Kafi miktarda şifreli paket toplandığında pek çok şifreli kablosuz iletişim çözülebilir. Kurumsal firmalar bu sebeple VPN gibi üst düzey çalışan şifreleme yöntemine yönelir. Bu yöntem için kullanılan protokol 802.1x'dir (Kargacı, 2015).

İletişim ağları boyutuna, kapsama alanına ya da yapısına göre sınıflandırılabilir. Kapsadığı alana göre ağlar şu şekilde sınıflandırılır:

- Yerel ağ (LAN (LocalArea Network)): Bir binada, bir kurumda, bir odada ya da binalar arasında kablolar ile oluşturulmuş ağ sistemleridir.
- Orta ölçekli ağ (MAN (Metropolitan /MiddleArea Network)) : İçinde pek çok LAN sistemini barındıran ve bunları birbirine bağlayan ağ sistemidir.
- Geniş Ağ (WAN (WideArea Network)) : Birden çok MAN sistemini birbirine bağlayıp modemler yardımıyla bağlanılarak ağa bağlanmaya olanak veren sistemdir. İnternet bu ağ sisteminin en güzel örneğidir.

İletişim ağları tarafından sunulan ortak avantajları şöyle sıralayabiliriz:

- Ortak kaynak kullanımı (disk, yazıcı, cd-rom, modem vb.) ile donanım maliyetleri düşer.
- Ortak çalışma olanağı artırılır ve takım çalışmasında hız artar.
- Çalışanlardan yüksek verim ve performans alınır.
- Doküman kullanımı minimuma iner, dökümantasyon trafiği azalarak karışıklık önlenir. Bu durumda yer ve zamandan tasarruf edilir.
- Dosyaların CD ya da bir okuyucuya taşınmasındansa ağ üzerinden aktarımı zaman kazandırır.
- Mail ve mesajlaşma servisleri sayesinde firma içi haberleşme daha kolay sağlanır, iletişimde kopukluk yaşanmaz.
- Önemli dosya ve bilgilerin depolanıp yedeklenmesi çok daha basit bir hale gelir (Kılıç, 2007).

2.1.2. Ethernet

Ethernet Yerel ağlar (Local Area Network (LAN)) için kullanılan Veri Çerçevesi (Data Frame) tabanlı bir bilgisayarlar arası ağ teknolojileri ailesidir. Ethernet kelimesinin kökeni “ether” sözcüğünden gelir. OSI ağ modelinin fiziki katmanı için

Veri bağlantı katmanı/ Ortam erişim kontrolü (İngilizce: Media Access Control (MAC)) aracılığı ile ağ erişimi üstünden bir dizi kablolar ve sinyalleşmeler standardı ve ortak bir adresleme formatı tanımlar.

Uluslararası Standartlar Organizasyonu'na belirlenen ve birbirinden farklı bazı sistemlerin birbiriyle haberleşebilmesini sağlayan OSI (Sistemler arası Bağlantı) katmanlarından, kablo standartları ile Fiziksel katmanda ve farklı haberleşme ortamlarının tek adres yapısı ile Veri Bağlantı katmanında çalışır. IEEE 802.3 standardı olan Ethernet, son noktaları aynı ağda birbirine bağlamakta kullanılan ve en çok kullanılan kablolu yerel ağ teknolojisidir. İlk etapta tek taraflı yayın ile bilgisayarlar arası haberleşmeyi sağlamak amacıyla geliştirilen bu teknoloji, daha sonra tam ve bütün bir ağ teknolojisi halini almış ve birçok yerel ağda kullanılmaya başlanmıştır (Tosunoğlu, 2009).

Ethernet kullanmak yanında şu faydaları getirir:

- Dünyanın en geniş kapsamlı ticari standartlarında ve pratikte kullanım
- Kolay bir algoritma, ucuz maliyet
- İstenilirse o an yeni istasyonların kurulumu
- Bozulma ya da herhangi bir kopma yaşanmadan istasyonların devre dışı bırakılması ya da sökülebilmesi
- Pek çok elektronikte ucuz ve standart ara yüz
- İstenildiği kadar genişletilebilmesi (Tosunoğlu, 2009).

2.1.3. TCP/IP

TCP-IP (Transmission Control Protocol - Internet Protocol), İnternet ve bilgisayarlar arası iletişimi sağlayan protokol grubunun adıdır. Bilgisayarlar arasında veri ve bilgi alışverişleri IP paketleri vasıtasıyla yapılır. Örneğin A bilgisayarı, B bilgisayarına göndermek istediği veri veya bilgiyi paket halinde gönderir. Bu paket içeriğinde A bilgisayarının ve B bilgisayarının adresleri mevcuttur. A bilgisayarının gönderdiği paket IP-Paketi, A bilgisayarının adresi IP-Adresi olarak tanımlanır. Gönderilen paket, B bilgisayarının adresini taşıyor olduğu için bilgisayarlar arası kullanılan ağ yönlendiriciler (Router) yardımıyla doğru bilgisayara.

TCP protokolü, A ve B bilgisayarları arasında veri ve bilgi alışverişi yapılmadan önce bağlantının kurulması ve iletişimin garantili olarak yapıldığı bağlantı tabanlı bir protokoldür. TCP iletişiminde veri paketleri kullanımı vardır. Ayrıca gönderici ve alıcı uygulamalarda TCP paket başlıklarına port bilgisi eklenir. Paketlerde bulunan portlar, kaynaklar ve hedef IP bilgileri ile uygulamanın iletişimini sağlar.

TCP protokolünde verinin gönderim hedefi olan bilgisayarın port numarasını “destination port” terimi belirtir. Datanın segment segment bölünüp gönderiminin ardından, yeniden birleştirilmesi için gereksinim duyulan numara “sequence number” olarak tanımlanır. Bir diğer TCP socketinin beklendiğini “acknowledgment number” kuralı belirler. Başlık bilgisi içerisinde yer alan 32 bitlik kelimeleri “header length” belirler. “Reserved” kuralı daima “0” değerini alır. Bir oturum kurulup bitirilebilmesi için gereken fonksiyonun adı “code bits” fonksiyonudur. “Window”, octet cinsinden gönderen bilgisayarın kabul ettiği pencere büyüklüğü olarak ifade edilmektedir. “Checksum”, CRC header ve data alanlarını kontrol eden iletişim kuralına verilen isimdir. Acil veri sorununu “urgent pointer” belirler. “Options” kuralı ise maksimum TCP segment büyüklüğüne “0” ile “32” arası bir değer atar (Ünlü, 2007).

IP, iki bilgisayar paketinin de yönlendirilmesini sağlayan bağlantısız bir protokoldür. Bağlantısız (connectionless) olma nedeninin oturumun iletişimden önce açılmaması ile ilgili olarak açıklanır. Bununla beraber veri iletimindeki başarı da garantilidir demek yanlıştır. Veri ve bilgi iletiminin garantisi daha üst düzey bir protokol olan TCP ile sağlanır. IP adresini bir ağa bağlı cihazların ağ üzerinden birbirleri ile iletişime geçmek ve birbirlerine veri göndermek için kullandıkları adres olarak tanımlamak mümkündür. İnternet bağlantısı olan her bilgisayarın bir IP adresi vardır. Bu nedenle aynı ağ üzerinde olmayan iki bilgisayar birbirlerine IP adresi sayesinde veri ve bilgi gönderebilir (Kardaş, 2014).

IP paketi bağımsız olarak gönderilir. B bilgisayarının görevi tüm IP paketlerini alıp, A bilgisayarının göndermiş olduğu veriyi kazanmaktır. Gönderilen kimi IP paketleri B bilgisayarına ulaşamayabilir. Her bir paketin içinde gönderimi yapılacak paket sayısı ile ilgili bilgi var olduğundan B bilgisayar alması gereken IP paketinin sayısını bilir. Eksik IP paketleri varsa A bilgisayarından istenir. Bu noktada devreye giren TCP'nin görevi, gönderilen IP paketlerinin denetimini sağlamaktır. TCP eksik

paket varsa bu paketleri tespit eder ve A bilgisayarından yeniden temin edilmesi için harekete geçip gerekli işlemleri yapar (Bir, 2005).

2.1.4. UDP

UDP (User Datagram Protocol - Kullanıcı Veribloğu İletişim Kuralları), TCP/IP protokolü takımlarının çift aktarım katman protokolünden biri olarak bilinir. Veri ve bilgileri bağlantı ihtiyacı duymadan gönderir. UDP, mümkün mertebe az sayıda mesaj alışverişi odaklı ulaşım katmanlı faaliyette bulunan bir protokoldür. Veri iletiminin garantisi ya da doğruluğu belli olmaz.

UDP, hepsi 16 bit uzunluğuna ulaşan 4 farklı bölümden oluşur. IPv4'te Kaynak Port Numarası ve Kontrol Sayısı (Checksum) seçmeli bölümleridir. IPv6'da ise sadece Kaynak Port Numarası seçmelidir.

UDP de tıpkı TCP gibi bir taşıma katmanı protokolü olarak adlandırılır. TCP'den farklı olarak UDP kullanımında verinin tam olarak alınıp alınmadığıyla ilgili bir doğrulama (authentication) yapılmadığı için gönderim garantisi olmaz.

Gelişmiş bilgisayar ağlarında paket anahtarlama bilgisayar etkileşiminde bir datagram modu oluşturabilmek amacıyla UDP protokolü yazılmıştır. UDP protokolü bir programdan diğerine mesaj gönderimi için kendi içinde bir prosedür içermektedir. Bu protokol 'transaction' yönlendirmelidir. Veri veya bilginin teslim garantisini isteyen uygulamalar TCP protokolünü kullanmaktadır (Ünlü, 2007).

Akıllı ev sisteminde karar verilmesi gereken protokollerden olan UDP, teslim garantisi olmamasının yanında TCP'den çok daha hızlıdır. Geliştirilen mekanizmaya göre, kontrol kumandasına dokunulduğu an komutun uygulanması beklenir. Dolayısıyla komutun verilmesi ve ağ ile cihaza gitmesi arasındaki sürenin uzun olması dezavantajdır (Menteşoğlu ve ark., 2015).

2.1.5. Hücresel veri iletişimi (3G/4G)

Global Systemfor Mobile Communications veya kısaca GSM (Mobil İletişim İçin Küresel Sistem), cep telefonları arası iletişim protokolüdür. Önceleri Avrupa Telekomünikasyon Standartlar Komitesi'nin GroupeSpéciale Mobile (Mobil İletişim

Özel Grubu) isimli alt kuruluşunun adını taşıyan GSM, sistemin küreselleşmesi ile birlikte yeni adını almaya başlamıştır. GSM, en bilinen ve yaygın olan mobil standart olan ve iki milyardan fazla insanın 212 ülkede kullandığı bir sistemdir. En cazip edici özelliklerinin başında aynı hat ile farklı ülkelerde görüşme sağlayabilme olanağıdır. Tüm GSM standartları, hücrenel bir ağ kullanarak dolaşım sırasında hücreler arasında geçiş yapabilme yeteneğine sahiptir. Teorikte kapsama alanı içinde kalındığında tüm dünyayı telefon görüşmesini kesmeksizin dolaşmak mümkündür (Kargacı, 2015).

Mobil telefon teknolojilerinde, haberleşme alanı, hücre denilen küçük coğrafi alanlara bölünmüştür. Bu hücrelerin merkezinde birer baz istasyonu bulunur. Mobil iletişim baz istasyonları üzerinden yapılmaktadır. Baz istasyonları, Mobil Anahtarlama Merkezlerine bağlıdır. Mobil Anahtarlama Merkezleri de farklı merkezler ile birbirlerine bağlıdır. Bu bağlantılar kablo ya da yönlü radyolinklerle sağlanmıştır. Mobil telefonlarla baz istasyonları arasındaki iletişim, elektromanyetik dalgalar yoluyla gerçekleştirilmektedir. Hücrenel yapı sayesinde aynı anda daha çok kullanıcı haberleşebilmekte ve mobil cihazını kullanabilmektedir (Cincirop, 2009).

GSM sistemlerinin nesilleri aşağıdaki gibidir:

- 0G olarak adlandırılan birinci nesil sistemlerde, analog veri akışı kullanılır.
- 1G olarak adlandırılan birinci nesil sistemlerde, analog veri akışı kullanılır.
- 2G olarak adlandırılan ikinci nesil sistemlerde sayısal veri akışı kullanılır.
- 2,5G olarak adlandırılan ikinci nesil sistemlerde devre anahtarlama sistemlere ilave olarak paket bazlı veri iletişimin eklenmesini (GPRS, EDGE) içeren şebekeleri ifade eder. GSM 2G ve 2,5G kategorisine giren ikinci nesil bir sistemdir
- 3G olarak adlandırılan üçüncü nesil sistem ile daha hızlı veri ve bilgi transferi ve bant genişliğinin daha verimli kullanımı mümkün olmuştur.
- 4G olarak adlandırılan dördüncü nesil sistem ile kapsama alanı başta olmak üzere 3G ile çözülememiş olan sorunların çözülmesi beklenmektedir.

2.2. Android Yazılım Geliştirme Ortamı

Android, günümüz teknolojisinde bazı telefonlar için geliştirilmiş bir sistemdir. Linux tabanlı mobil telefonlar (PDA ve cep telefonları) için geliştirilen, Open

Handset Alliance ve Google tarafından kodlanan, açık kaynak kodlu olan bir işletim sistemidir. Android işletim sistemi, mobil cihazların kullanış ve verimliliğini artıran uygulamalar oluşturan oldukça geniş bir geliştirici ve yükseltici kitlesine sahiptir. Android telefon ya da tabletler için birbirinden kullanışlı 250,000 kadar uygulama mevcuttur. Android Market diğer adıyla Google Play Store, arama motoru Google tarafından işletilen bir kurumsal uygulama mağazasıdır ve Android işletim sistemi için geliştirilen uygulamalarının farklı sitelerden download edilmesini sağlamaktadır (Koçak ve Kırbaş, 2016).

Open Handset Alliance, 5 Kasım 2007 tarihinde Android'i kurduğunu duyurmuş ve hemen akabinde 35 kadar bilişim şirketi mobil cihazlar için geliştirilen ve telif hakkına ihtiyaç duymayan bir işletim sisteminin teknolojik gelişim için faydalı olduğu konusunda mutabakat sağlamışlardır (Kardaş, 2014).

Android işletim sistemi, Linux çekirdeği üzerine kurulan bir mobil sistemdir. Android işletim sistemi ara katman yazılımı ve kütüphaneler API C diliyle yazılmıştır. Android uygulamalarının yazılımlarına bakıldığında Apache harmony üzerine kurulmuş ve Java uyumlu kütüphaneleri içeren bir uygulama sistemi üzerinden çalışmasını gerçekleştirir. Android işletim sistemi derlenmiş olan Java kodlarını çalıştırmak için dinamik çevirmeli (JIT) santral makinesini kullanır. Cihazlar için oldukça fonksiyonel olan ve uygulamaların geliştirilmesi amacıyla çalışan oldukça geniş bir geliştirici çevresine sahiptir (Ünlü, 2007).

Akıllı ev ve binaların yönetim sistemleri için tercih edilen mobil işletim sistemleri açık kaynak kodlu kütüphane kullanma ve geliştirilen yazılımları dağıtabilme özgürlüğüne sahip olması ve işletim sistemlerinde uygulama geliştiriciliği esnekliğine sahip olan Android işletim sistemi her zaman daha önceliklidir. Android işletim sistemi, günümüz bilgisayarlarında özel bir yazılım ya da donanım gerektirmediği için kullanıcıya büyük avantaj ve esneklik sağlar (Goadrich and Rogers 2011).

Akıllı ev ve binalarda uygulamaya sunulan Android sistemin taşıması gereken özellikler şöyle sıralanabilmektedir:

- Eş zamanlı çalışabilme.

- Esneklik.
- Hızlı ve pratik yazılım geliştirebilme.
- Hızlı ve kolay bir biçimde yeni özellikler ekleme, çıkarma ve var olan özellikleri değiştirebilme. Yapay zeka algoritmalarını kolaylıkla değiştirebilme.
- Donanımdan bağımsız olarak çalışabilme.
- Pek çok sensör ve dedektör çeşidi ile çalışıyor olabilme (Göktaş, 2006).

Menteşoğlu vd. (2015) çalışmasında mobil aygıtlarla akıllı ev ve akıllı bina sisteminin gözlenip kontrol edilebilmesi için Android'i kullanmış ve mobil cihazın kontrol birimi ve algılayıcı birimler arasında var olan uygulamaya özgün bir haberleşme protokol tasarımı uygulanmasına dikkat verilmiştir.

2.3. Sistemde Yer Alan Sensörler

Akıllı ev ve akıllı bina otomasyonları günlük hayatımızda karşımıza çıkan ve hayatımızı direkt olarak etkileyen etkenleri kontrol edip daha az stresli daha çok mutlu ve huzurlu bir yaşam hedefler. Bu etkenler hem aile hayatımızı hem de iş hayatımızı direkt etkileyen ısı, ışık, nem gibi etkenlerdir (Kongaz, 2007).

Akıllı ev ve akıllı bina otomasyonlarında kullanılan sensör ve dedektörlerin diğer sistemlerde kullanılanlardan bir farkı bulunmamaktadır. Fakat iletişim birimlerine uygun olan sinyali üretebilmesi açısından çeşitli modifiye işlemlerine tabi olabilir veya da ek bir cihaz yardımı ile uyumluluğu sağlanabilir.

Akıllı ev ve binalarda evde var olan uyarıcılar elektronik uygulamalara dedektör, sensör gibi cihazlar aracılığıyla iletilmektedir. Söz konusu dedektör ve sensörler vasıtası ile sistem var olan bir kontrolü aktifleştirebilmektedir. Mesela su baskını ihtimali olduğunda akıllı ev su vanalarını kapatabilir (Erbey, 2016).

Akıllı evlerde var olan duman, su baskını, güvenlik gibi dedektör ve sensörler evde kimse olmasa bile bir problem anında gerekli yerlere telefon ya da e mail aracılığı ile ulaşır ve bilgilendirme gönderir. Bu dedektörlerin su veya duman seviyesi daha önce belirlenen sınırı aştığı anda gereken işlem yapılır yani su ya da doğalgaz mekanizması kapanır ve ihtiyaç varsa alarm devreye girer (Jamalabad, 2014).

Akıllı ev ve binalarda mevcut olan sensör ve dedektörlerin yapabileceklerini şöyle sıralayabiliriz:

- Isı algılayıcı sensör ile mekânın ısı algılanır, klima açılır ya da kapanır. Isı sistemlerine internet üzerinden online olarak erişim sağlanabilir.
- Işık sensörleri ile güneşin batış saati ayarlanır, perdeler duruma göre kapatılır ve eğer istenirse bahçe ışıkları ve ev ışıkları açılabilir.
- Alarm aktif duruma getirildiğinde hareket algılama sensörü sayesinde ortamda hareket meydana geldiğinde alarm tetiklenebilir, güvenlik ışıkları yanar pozisyona getirilip cep telefonuna mesaj atılabilir.
- Garaj kapısının açıldığı algılanır ve eve geçişte kullanılacak yol aydınlanarak klima çalıştırılır.
- Uzaktan (internet üzerinden) erişim sayesinde uzun süreli evden uzak kaldığında panjurlar ve perdeler açılır, ev güneşlendirilir ve duruma göre ev havalandırılır ve bahçe sulanır.
- Duman sensörleri yangını algılayarak ev sahibine SMS gönderir, duruma göre alarmı çalıştırır.
- Sismik sensörü ile deprem önceden algılanır, sular ve gaz kesilebilir ve alarm çalıştırılabilir.
- Evdeki var olan kamera sistemleri ile dünyanın herhangi bir yerinde evin arka bahçesinde oynayan ya da odasına uyuyan bir çocuk canlı olarak izlenebilir, bakıcıya emanet edilen çocuklar dakika dakika izlenebilir.
- Dış kapıya ve hatta sitenin güvenlik kapısına bir ziyaretçi gelmişse evde bulunan TV ya da bilgisayar üzerinden canlı olarak izlenme, filme alınması veya fotoğrafının çekilebilmesi sağlanır (Seçer, 2006).

2.3.1.Gaz Sensörleri

Akıllı ya da akıllı olmayan evlerde gaz sistemi, gündelik hayatın vazgeçilmez ihtiyaçlarının başında gelmektedir. Mutfakta pişirme, evi ısıtma ya da sıcak su ihtiyacını karşılamak için kullanılan gazlar, önlem alınmaya çalışsa da kaçak ve sızıntı sırasında ölümcül sonuçlara neden olabilmektedir. Böyle olumsuz durumlarda olayı kısa sürede algılayan otomasyon sistemleri gaz dağıtımını devre dışı bırakarak

mekana gaz girişini keser ve gaz firmasına acil koduyla bir uyarı gönderir. Ev sahipleri evde yokken ev sahibine de telefon ile haber verilebilir (Aslan, 2014).



Şekil 2.1. Gaz Sensörü

Akıllı ev ve akıllı bina sistemlerine dahil edilmiş gaz sensörleri, temel olarak ortama yayılan gaz kokusu yoğunluğunu algılayarak gaz kaçağı gibi durumlarda sistemi haberdar eder. Alarm sistemi kurulmamış olsa bile 24 saat çalışır (Bayram, 2006).

Gaz alarm cihazı, çeşitli gazların kaçak vermesi durumunda ev, ofis gibi kapalı alanlarda kullanılmak üzere tasarlanmıştır. Sabit olarak kurulan ve daima çalışan bir alarm cihazıdır. Röle çıkışı olması dolayısıyla eğer gaz alarmı verilirse aspiratörler, sirenler ve gaz kesme valfi gibi birimlere ulaşarak otomatik olarak kilitleme işlevini gerçekleştirir (Görel ve Küçüktepe, 2012).

Patlayıcı gazların üretim süreci içinde ortaya çıktığı ve depolanıp nakledildiği ortamlarda kaçakların patlayıcı karışım oluşmadan algılanıp önlemlerin alınması yaşamsal olarak büyük önem taşır. Gündelik hayatımızda kullandığımız doğalgaz, LPG ve havagazı gibi gazların yanı sıra endüstriyel gazlara da duyarlı olan sensörler, gaz alarm sistemi kullanım bölgesi ve kullanım amacına uygun olarak üretilmektedir (Aslan, 2014).

2.3.2. Yangın ve duman sensörleri

Yangın müdahalesinde zaman en önemli unsurdur. Evin içinde insanlar varken yangının fark edilmesi uzun sürebilirken evde kimse yokken bu süre çok daha uzayabilir. Akıllı evlerde var olan yangın ve duman sensörleri yangına müdahalenin zamanını en aza indirir. Duman sensörleri bir duman algıladığı zaman fiskiyeler otomatik olarak devreye girer ve akıllı evlerde daha önce yapılan ayarlama sonucu itfaiyeye haber verilmektedir (Kongaz, 2007).



Şekil 2.2. Yangın ve Duman sensörü

Belirlenen seviye üzerinde duman algılandığında ya da yangın meydana geldiğinde herhangi bir yangın ya da duman tespit edildiği sırada gaz vanaları ve havalandırmalar otomatik olarak kapatılır ve böylelikle yangının daha büyük boyutlara ulaşması engellenir. Yine bir yangın durumunda itfaiye aranır ya da e mail gönderilir ve ev sahibinin kişisel telefonuna SMS ile mesaj gönderilerek bilgi verilir (Seçer, 2006).

Bununla birlikte akıllı ev içinde tehlikeli olabilecek biçimde zararlı gaz, duman ya da karbondioksit oranı artarsa otomasyon sistemindeki veri akışına göre gerekli noktalarda pencerelerin açılıp ortamın havalanması sağlanabilir (Şahinoğlu, 2006).

Duman sensörü ya da detektörü bir tür yangın dedektörüdür. Yangın sensörleri yangınların kendine has özelliklerine göre dizayn edilmiş cihazlardır. Isı, alev, duman gibi alt kategorilere ayrılan dedektörler normal çevre şartları ve yangın arasındaki farkları kolaylıkla algılayabilme özelliğine sahiptir.

Sensörler, yangın algıladıkları zaman eğer ev içindeyseniz iç sirenle eğer evde değilseniz sizi telefonla arayarak uyarma özelliğine sahiptir. Size haber verdiği esnada itfaiyeyi de arayarak uyarır ve bu sırada gaz vanaları ile havalandırma pencerelerini de kapatır. Evin içindeyseniz çıkışınızı kolaylaştırmak için çıkış yolunu aydınlatarak işinizi kolaylaştırır ve evi kolayca tahliye etmenize olanak sağlar (Küçükbakırcı, 2006).

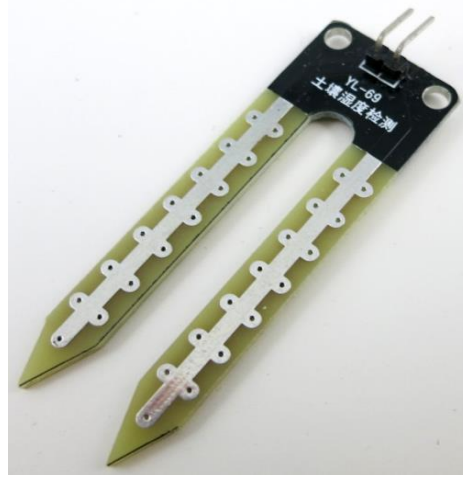
Günümüzde hemen hemen bütün binalarda çeşitli önlemler vardır ve akıllı ev ve akıllı binalarda teknolojinin desteği ile bu önlemler maksimize edilmeye çalışılır. Mesela bir yangın esnasında yangın ve duman yayılmasını diye klima kapatılır, fiskiyeler devreye girer. Güvenli çıkışa yardımcı olmak için bazı ışıklar açık kalır, alarm ışıkları ise komşulara ve çevredekilere haber verir. Açılan ışıklar ise ev içinde kalan bireylerin güvenli bir şekilde çıkışına yardımcı olur. Telefon ile ise yangın istasyonuna ya da daha önce belirlenen numaralara haber verilebilir (Özçekiç, 2005).

2.3.3. Su sensörleri

Akıllı binalarda var olan akıllı su sistemleri, akıllı evde yer alan suyu sürekli kontrol altında tutmaya yarar. Şebeke suyu fark edilir derecede kirliyse ya da hastalık bulaştırabilecek derecede tehlikeliyse bağlı olunan ağa bu bilgi geçilir. Yakın konumda yer alan konutlar aynı sorunu yaşadığı zaman bağlı olduğu ağı kontrol eden birimler geri bildirim ihtiyacı duymadan müdahale edip gerekli tedbirleri almaktadır (Türkcan, 2007).

Su sensörü vasıtasıyla, suyun seviyesi elektronik olarak izlenip ekstra müdahaleye ihtiyaç duymaksızın otomasyon bilgisayarınca dolun ve kontrolü yapılmaktadır. Akıllı bina içinde var olan bütün su dolaşımı aynı yöntemle elektronik olarak izlenir. Otomasyon sistemi herhangi bir sorun ile karşılaştığı zaman bunu otomatik olarak algılar ve servis anında harekete geçer (Küçükbakırcı, 2006)

Su sızıntısında en büyük etken ek ve bağlantı yerlerinin sayıca çok olması olarak açıklanabilir. Üstelik salt ek ve bağlantı değil dağıtım hattında oluşması muhtemel olan basınç artışı ya da kaza sonucu borularda meydana gelen delinmeler suyu zararlı bir duruma getirebilir. Akıllı evlerde otomasyon ile kontrolü sağlanan sistemler ev içinde su kullanılan bölgelerde su sızıntı ya da kaçaklarını tespit eden sensör algılayıcı cihazlar kullanılabilir.



Şekil 2.3. Su sensörü

Bunları şöyle sınıflayabiliriz:

- Su sızıntısı uyarısı verildiğinde su ilk bağlantı noktasındaki valfin kapalı konuma getirilmesiyle kesilir. Böylelikle su ne mekana ne evde bulunan malzemelere zarar verebilir.
- Ortaya çıkabilecek elektrik teması durumunda canlıları elektrik çarpmasına karşı korur.
- Sızan suyun boşa akan su olduğunu algılar ve su maliyetinden tasarruf ettirir.
- Seyahate çıktığında su giriş valfleri devre dışı bırakılarak olası bir su baskını riski minimize edilmiş olur (Şahinoğlu, 2006).

Bunların yanı sıra, ayarlanan yıllık bir programla bahçede bulunan bitkiler yılın değişik zamanlarına göre farklı miktarlarda sulanabilir. Meteorolojik otomasyon ve toprağın nemini ölçümleyen sistem ne zaman ne kadar sulama yapılacağını belirler (Türkcan, 2007).

Gelişen ve her geçen gün yenilenen teknoloji ile beraber üst kullanıcıya hitap eden evlerde duş, jakuzi ve banyolarda suyun kontrolü dokunmatik sistemli paneller aracılığı ile yapılabilmektedir. Dokunmatik panele bağlı sistem suyun ısısı, akışının şiddeti ve akış süresi gibi etkenler panelden yapılacağı gibi evin herhangi bir noktasından da uzaktan kumanda aracılığı ile yapılabilmektedir (Honeywell, 2006).

2.3.4. Hareket sensörleri

Güvenlik sisteminin en önemli ve en temel gereksinimi olan sensörler, kullanım amacına göre ses, ışık, titreşim gibi etkileri algılayan elektronik cihazlardır. Kapı, pencere gibi noktalara konuşlandırılan algılayıcılar tüm evi gözetim altında tutabilmektedir. Sensör ve algılayıcıların ortak kontrolünü sağlayan bir ana kumanda bulunur ve bu kumanda o bölgenin güvenlik birimleri ile bağlantı halindedir. Bir tehlike durumunda bu kuruluşlara haber verilir (Seçer, 2006).

Akıllı ev ortamını daha yakından incelediğimizde ev ortamında günlük hareketlerin takibine de daha yakından baktığımızda özellikle sağlık destekli yaşam formları ortaya çıkmaktadır. Örneğin ev sahibi gece kaç kez uyandı, kaç saat uyudu ve kaç kez tuvalete çıktı gibi hareketler sonucu Alzheimer gibi bir hastalığa yakalanma riski, teşhisi veya hastalığın gidişatı hakkında fikir sahibi edebilir. Diğer taraftan ortam destekleyici yaşam uygulaması için insan hareketinin uzun dönemde takibi ve günlük davranışların belirlenmesi akabinde bu davranışlardan sapmanın anlaşılmasına yardımcı olur (Ertan ve ark, 2012; İncel ve Ersoy, 2016).

Hareket sensörü ile mekânların güvenliğinin sağlanması akıllı ev içi ya da ev dışı mekânlarda sıklıkla kullanılan bir sistemdir. Ultrasonik ve mikro dalgalar fotosel ışınları ile sürekli bir tarama halindedir. Kızılötesi ışın ile yapılan tarama ise son zamanların en çok kullanılan sistemlerinin başında gelmektedir (Gül, 2010).

Hareket dedektörleri ev etrafında siz uykudayken dolaşan davetsiz misafirleri bütün ışıkları yakarak uzağa iter. Kapı ya da pencere zorlanır ya da kırılırsa alarm verir ve cep telefonunuzdan size ulaşır. Evden çıkmadan bir tuş ile evde yokum modunu aktif edebilir, açık kalmış kapı ya da pencereleri fark edebilirsiniz. Eve dönüş sırasında aracınız ise garaj tarafından fark edilerek otomatik olarak garaj kapısı açılır (Türkcan, 2007).

Hareket dedektörlerinin en bilinen ve en önemli özelliği caydırıcı sistem olarak kullanılmalarıdır. Söz konusu sistemin en önemli özelliği ve temel amacı ise tehlike durumunun oluşmamasını sağlamak ya da olası bir tehlike anında tehlikeyi uzaklaştırmaktır. Örnek vakalar göstermektedir ki olası bir hırsızlık vakasında hırsız

eve hapsedildiğinde yakalanma psikolojisi ile evdeki değerli eşyalara ve hatta eve dahi zarar verebilmektedir.

Aktif denilen akıllı evleri tüm imkânları ile kontrolü altında tutabilen sistemler aynı zamanda yaşayan sistemlerdir. Aktif sistemin amacı sizin sezmediğiniz tehlikeleri önceden sezmek ve evden uzaklaştırmaktır. Tehlike izlenimi veren herhangi bir durumda mevcut bütün senaryoları kullanarak evi ve çevresini savunur. Örneğin evde kimse yokken kapıyı çalan birini fotoğrafını çekerek isterseniz eve döndüğünüzde isterseniz gerçek zamanlı olarak sizinle paylaşır.

Bilinen alarmların tehlike sırasında alarm çalması ve gerekli yerlere ulaşması dışında akıllı evlerde ek olarak SMS ile haber verilmesi ve gerektiğinde evi ışıkları yakıp panjurları kapatarak korumaya alması da göze çarpar (Küçükbakırcı, 2006).

2.3.5. Işık sensörleri

Akıllı evlerde kontrol edilecek ikinci önemli unsur aydınlatmadır. Işık şiddeti görme duyumuzu etkileyen önemli bir parametredir. Işık yeterli seviyenin altında yanıyorsa göz sağlığına zarar vermekte ve performans düşüklüğü meydana getirmektedir. Işığın kontrolü için çeşitli yöntemler kullanılırken sensörler ortamın ışık şiddetini ölçer ve kontrol yapabilir. Bir diğer yöntem ise zamana bağlı olarak yapılan ışık şiddeti kontrolüdür. Bu tip kontrolde daha önce ayarlanmış bir zaman diliminde ışıklar yanabilir veya sönebilir. Akıllı evler ışık kontrolü ile enerjiden tasarruf etmeyi amaçlar (Kongaz, 2007).

Aydınlatma sensörleri, oda ya da mekân içerisinde farklı bölgelerde oluşacak aydınlık seviyelerini ölçmek ve her bir sensörü bağlı olduğu aydınlatma ve ışık grubuna uygun şekilde ayarlama amaçlı olarakta kullanılabilir.

Aydınlatma, sadece karanlığın giderilmesi demek değildir. Evde yaşayanların istek ve psikolojik durumlarına göre aydınlık seviyesinin ayarlanmasıdır. Aydınlatma sistemi sayesinde aydınlık derecesinin programlanmış bir seviyeye ve senaryoya göre otomatik ayarlanması veya da kumanda üzerinden arzuya göre değişimi mümkündür (Seçer, 2006).

Aydınlatma sistemi ve alt birimleri su şekilde sıralanır:

- Fotoselli ve otomatik aydınlatma sistemi
- Dimmerleme, anahtarlama ve gölgelendirme sistemi
- Senaryo kontrol ve aydınlatma sistemi
- Panjur, jaluzi ve diğer güneş kontrol sistemi
- Dış aydınlatma sistemi
- İç aydınlatma sistemi
- Işık kaynakları
- Aksesuarlar

Bahçe aydınlatmalarının veya havuz aydınlatmalarının evin herhangi bir veya birden fazla yerlerinden kontrol edilebilmesi de mümkün olmaktadır. Bina iç ya da dışında koridor ve geniş alanlara, merdiven gibi noktalara yerleştirilecek sensörler ve hareket dedektörleri ile aydınlatmanın zamanı ve konumlandırılması mümkündür ve bu da enerji tasarrufunu sağlar. Örneğin gece vakti eve yaklaşan biri varsa önce hareket bölgesi ve daha sonra tüm güvenlik lambaları aktif bir hale gelmektedir. Söz konusu sistem ışın çiti ya da farklı algılayıcılar ile yapılabilmektedir. Dış aydınlatma olarak adlandırılan bahçe ve kapı önü aydınlatmaları gündüz ve gece sensörü denilen sensör türü ile programlanabilir. Yani gün ışığı seviyesine göre aydınlatmalara aç ya da kapat komutu verilebilmektedir (Şahinoğlu, 2006)

Bilindiği gibi ev otomasyonunun bir tarafında maliyet ve bir tarafında ise sınırsız konfor bulunur. Aydınlatmanın kendisini gün ışığı saatine göre ayarlaması için programlama da kişisel opsiyon içinde bulundurulabilir. Yatma zamanı gelince ev içindeki tüm ışıklar tek bir düğmeyle aracılığıyla kapatılabilmektedir (Özçekiç, 2005).

Akıllı evlerde aydınlatma sistemi ilk olarak enerjiden tasarruf sağlamalıdır. Enerji ücretinin fazla olması sebebiyle akıllı konut tüm sistemler gibi aydınlatma sisteminde de tasarrufa yönelmelidir. Bunun için ilk olarak aydınlatmada kullanılacak ekipmanın az enerji yüksek verim prensibine uyması gerekmektedir. İlk etapta yüksek bir yatırım maliyeti olsa da bu sistemler uzun dönemde kar olarak geri dönüş

sağlar. Öyle ki bu ekipmanlar uzun vadede bakıma ve değişime ihtiyaç duymamaktadır (Geçim, 2002).

2.3.6. Sıcaklık sensörleri

Asırlar boyu insanlar, vücut ısısına uygun ısıdaki alanlarda yaşamayı tercih etmiş ve buldukları ortamları kendi vücut ısısına uygun hale getirmeye çalışmışlardır. Akıllı evlerde bunu baz alarak sıcaklık kontrolünü önemli bir parametre haline getirmiştir. Sıcaklıkla birlikte nem, havalandırma gibi parametreler de kontrol edilebilmektedir. Bu üçünün aynı anda kontrolü iklimlendirme olarak adlandırılır. Akıllı evler ve akıllı binalarda iklimlendirmenin kontrolü her zaman ön planda tutulmuştur. Kontrol sayesinde hem sıcaklık hem nem hem de havalandırma değerleri vücut ısısına en uygun hale getirilmiş olur. Bunlarla birlikte ekonomik olarak da tasarruf sağlanır (Kongaz, 2007).

Binalarda var olan iklimlendirme ve havalandırma fonksiyonlarını karşılayan HVAC sistemlerini incelediğimizde büyük yapılar ve küçük yapılarıdaki sistemlerin karmaşıklık ve amaç açısından farklılık gösterdiğini söylemek mümkündür. Ev içinde yaşayan birey sayısı ve elektronik cihaz sayısı gibi detaylar küçük yapılarda var olan ısı kazanımını büyük yapılara göre daha az seviyede tutar. Küçük yapılarda kullanıcı sıcaklık için istenen seviyeye ulaşmak ve HVAC sistemi aracılığıyla havalandırma işlemini gerçekleştirmeden daha öncelikli hedeftir. (Heerwagen, 2004).

İklimlendirme sisteminin alt kolları şöyle sıralanabilir:

- Havalandırma sistemi
- Isıtma sistemi
- Soğutma sistemi
- Senaryo kontrolü ve iklimlendirme sistemi
- Aksesuarlar (Seçer, 2006).

Ayrıca “pasif solar sistem” adı verilen ve güneş ışığının yapı içerisine girişini denetleyen mekanizmalarla ısının dengelenmeye çalışıldığı sistemler kullanılıp hem konfor koşulları ayarlanır hem de enerji tasarrufu artırılır. Pasif solar ısıtma, harici

mekanik pompalama sistemleri olmadan güneşten toplanıp dağıtılmasıyla gerçekleştirilir (Gül, 2010).

Müstakil konutların çok katlı olması ve oda olarak sayıca fazla olması ısı kontrolünün iyi yapılmasını gerekli kılar. Isınan hava yükselir, soğuyan hava alçalır ve yer değiştirir. Kullanım alanı olmayan ya da efektif olarak kullanılmayan odalarda yaşanan ısı kayıpları konutlarda çoklu bölgesel ısınma ve soğutma bölümlerine ayrışması ile kontrol edilebilmektedir (Aslan, 2014).

Akıllı ev ve binaların hayatımıza kattığı bunca konfor ve rahatlığa rağmen bazı durumlarda olumsuz sonuçlara da neden olabilmektedir. Bilhassa iklimlendirme cihazlarının sağlıklı düzenlenmesi neticesinde baş dönmesi, mide bulantısı ya da adaptasyon sorunları baş gösteren en önemli sorunlardır (Gül, 2010).

2.4. Akıllı Ev Yönetim Sistemleri Protokolleri

İletişim birimleri ve iletişim birimlerinin uzantıları otomasyonda ve bunun sonucu olarak da ev-bina otomasyonunda en önemli konularda biri haline gelmiştir. Her biri farklı işlev gören ve farklı firmalar tarafından üretilen farklı cihazların bir arada çalışabilmesi tamamen iletişim yeteneğine bağlıdır. Bugüne dek bu yeteneğin gelişmesi ve geliştirilmesi için pek çok adım atılmış sayısız uygulama hayata geçirilmiştir. İletişim biriminin en önemli iki faktörü şunlardır:

- a) Mesajın yapısı (Message Structure)
- b) Mesajın nasıl gönderileceği

Bu iki özelliği kısaca iletişimin dili olarak tanımlayabiliriz. Ağın nasıl kurulacağı ve ağın yapısı da bunların yanında önemli birer faktördür fakat kritik önemi iletişim dili taşır. Basit bir iletişim protokolü ağ ve cihazın kolay uyum sağlaması demektir (Özçekiç, 2005).

Kısacası iletişim protokolü, iki cihazın bilgi aktarımında kullandığı lisansa verilen isimdir. Nasıl ki iki bireyin anlaşması için aynı dilden konuşup aynı yerden bakması gerekiyorsa iki cihazın da anlaşması aynı protokole bağlı olmasından geçer (Kılıç, 2007).

Bilişim sistemleri açısından, bir linkteki veri akışının kontrolü için, gereken sinyallerin biri de saat sinylidir. Gönderici ve alıcı cihazda bir bit ne zaman gönderilecek ya da ne zaman alınacak kararı verilirken saat sinyali kullanılır. Veriyi gönderen ve alan noktaların belirli kurallar çerçevesinde haberleşmesi gerekmektedir. Verinin paketlenme metodu, bir karakterde var olan bit sayısı ve verinin başlama ve bitme anları gibi bilgileri bu kurallar belirler. Bu kuralların adına protokol adı verilir (Bayram, 2006).

Modeller, bilgisayar ağı protokollerinin sağlaması gereken servisleri tasarlamaktadır. Modeller arasında en popüler olan model ise ISO tarafından tasarlanan yedi katmanlı OSI modelidir. OSI modelinden önce tanımlanan bir model, Birleşik Devletler Savunma Bakanlığı, (Department of Defence) tarafından geliştirilen olan (DoD) adı verilen modeldir. Bu model, 1970'lerin ortasında geliştirilen TCP/IP protokol suite modelidir. Yazılım ve donanım mühendisleri, protokolleri gerçekleştirecek ürünler oluşturur. Belli bir protokolü birbirinden farklı firmalar geliştirmiş olsa da aralarında bazı farklar meydana gelecektir.

Fakat protokol doğru gerçekleştirilirse farklı ürünleri aynı protokol üzerinden iletişimde hiçbir sıkıntı yaşanmayacaktır (Bir, 2005).

Bir akıllı ev ya da akıllı bina yaratmak için evde bulunan cihazlarla alakalı bilgi toplamak ve protokolleri öğrenmek gerekir. Cihazlarda kullanılan protokolleri şöyle sıralayabiliriz:

- Cihaz Elektronik Kartları: Kontrol edilecek cihazda var olan ve haberleşme aracılığıyla bağlanacak olan kartın adıdır. Cihazın işlevine göre kullanıcı tarafından seçilecek fonksiyonu gerçekleştirir.
- Seri Haberleşme: Cihaz elektronik Kartı ile seri-Ethernet dönüştürücü arasında haberleşme aracılığında kullanılan protokolün ismidir.
- Seri-Ethernet Dönüştürücü: Cihazdan seri olarak aktarılan veri ve bilgileri Ethernet protokolüne çevirerek yerel ağ üzerinden iletimini sağlar. Ve aynı yöntemle yerel ağdan gönderilen bilgileri seri protokole çevirip elektronik karta ulaştırmayı sağlar.

- Ethernet: Dönüştürücü ile son kullanıcı arasındaki iletişim ve haberleşme Ethernet protokolü ve yerel ağ aracılığı ile sağlanır.
- Modem: Sisteme eklenecek modem cihazı internete erişmeye ve buna bağlı olarak ev dışından sistem müdahalesine yarar (Tosunoğlu, 2009).

Akıllı ev ve bina sistemlerinde uyumlu bir birlikte çalışma ve genişletilebilirliği sağlama için ortak bir protokol standardının belirleniyor olması ve sistemlerin farklı firmalar aracılığıyla yapılması gelişimi biraz yavaşlatmıştır. Fakat bu durum için farklı toplulukları üretmeye çalıştığı çözümler mevcuttur. Bunun sonucunda Profibus, Modbus, Echelon ve Bacnet gibi bazı protokoller geliştirilmiştir (Kılıç, 2007).

- Profibus: Profibus haberleşme sistemi Siemens’inde içinde bulunduğu birçok PLC üretici firma tarafından geliştirilmiş ve standart olarak sınıflandırılmış bir ağ sistemidir. Kendine özel bir ara birime ihtiyaç duymaksızın farklı üreticilerden çıkmış cihazlar arasında haberleşme işlevini görür. Profibus, üreticilerinden bağımsız olan bir açık hava protokolüdür. Hızlı bir veri alışverişi sağlar. Buna bağlı olarak da yüksek hıza ihtiyaç duyan uygulamalar ve karmaşık sistemli haberleşme işlerinde Geniş ölçekli üretim ve proses otomasyonu için tasarlanmıştır. Söz konusu protokolde cihaz değişimi mümkündür. Profibus’ta her bir bölüme 32, toplamda 126 katılımcı bağlanabilir. Çevre birimleri slave’ler ve saha elemanları çalışma esnasında takip çıkarılabilir özelliindedir. 2 damarlı blendajlı kablo veya optik iletkenler yardımıyla veri transferi gerçekleştirilir.
- Modbus: PLC’ler arası seri haberleşme protokolü olarak MODICON firması tarafından tasarlanarak kullanıma sunulan Modbus, basit kullanımı ve güçlü yapısı sayesinde endüstriyel alanda en sık kullanılan protokollerin başında gelir. Farklı cihazlar arası haberleşmeyi sağlayan sunucu ve istemci bir taban protokolü olarak anılır.
- Canbus: CANBus (Controller Area Network Bus/Kontrol Alan Ağı Veriyolu) protokolü, Robert Bosch tarafından geliştirilen, birden çok kablo yerine tek bir kablo ile veri transferi amaçlayan bir protokoldür. Otomotiv sektöründe oldukça yaygın bir kullanımı vardır. İstatistiklere bakıldığında 100 yıl içinde 1 mesaj hatası yaptığı söylenebilir. Yüksek güvenilirliği sayesinde de endüstriyel alanlarda büyük bir yer edinmiştir.

- DeviceNet: Akıllı sensör ve dedektörler için Allen-Bradley tarafından geliştirilmiş endüstriyel iletişim protokolü olan DeviceNet ile düşük seviyeli aygıtlara bağlanılabilir ve PC veya PLC gibi daha üst seviyeli aygıtlarla haberleşme sağlanabilir. CAN mimarisi üzerine kurulan DeviceNet açık ve düşük maliyetlidir. Kararlı bir yapıya sahip olan Device Net, güvenlik ve performans ihtiyaçlarını karşılamak için ortaya çıkarılmıştır. Her cihaz ağın içinde bir düğümdür. Mesaj sıralaması önceliği olan üretici-tüketici bir protokoldür.
- AS-i: AS-i (Aktüatör Sensör-Arayüzü), PLC, DCS ve PC tabanlı otomasyon sistemlerinde kullanılan bir endüstriyel iletişim protokolüdür. Paralel kablolanmanın en basit alternatifidir. Tam olarak standartlaşma gösteremese de son derece basit ve ucuz bir sistemdir. Bunun yanında güvenilirliği de en üst seviyelerdedir.
- Interbus: Kontrol sistemleri arasında (PC, PLC, otomasyon kontrolörleri vb.) veri aktarımı sağlayan seri bus sistemidir. Phoenix Contact firması tarafından geliştirilip piyasaya sunulmuştur. Çift taraflı veri iletişimi gerçekleştiren sistem kapalı alan topolojisi ile haberleşmeye imkân verir.
- Hart: HART (Highway Addressable Remote Transducer), akıllı sistem ve izleme sistemi arasında analog kablo kullanarak dijital bilgileri gönderip almamıza yarayan bir protokoldür. Ana sistem ve akıllı cihaz arasında veri iletişimin iki yönlü olarak yapar. En önemli özelliği 4-20 mA üzerinden kablolanma gereçleriyle iletişim kurabilir.
- Lonworks: 80'li yılların ortalarında Mike Markula'nın fikirlerinin geliştirmesinden sonra temelleri atılan Echelon firması 1988 yılında kuruldu. 1990 yılında da Lonworks açık protokolünü geliştirerek büyük bir atılım yaptı. Echelon'un yönetimine günümüzde de görevine devam etmekte olan, PBX teknolojisinin yaratıcısı Dr. Kenneth Oshman getirildi. Echelon günümüzde de, gündelik kullanımda elektronik cihazlar arası iletişimi sağlayan teknolojileri üretip geliştiren firma imajını korumaktadır. Lonworks'ün teknolojik altyapısını oluşturmuş olan Nöron çipin lisansını elinde tutan Echelon, kontrol sektöründeki firmaların bu iletişim standardına kolaylıkla ve hızlıca adaptasyonunu sağlayacak ürünler/sistemler geliştirmeye devam etmektedir. Echelon farklı firmalar ile

(Toshiba, Motorola, Cypress Semiconductor) lisans anlaşmaları imzalayarak hem fiyat hem teknoloji rekabetini dengede tutacak bir politika izlemiştir (Kılıç, 2007).

- Bacnet: BACnet (Building Automation and Control Network) Amerikan Isıtma-Soğutma ve iklimlendirme Mühendisleri Derneği (ASHRAE -American Society of Heating, Refrigerating & Air Conditioning Engineers) tarafından oluşturulmuş, örgütler ve firmalar üzeri bir data haberleşme protokolüdür.

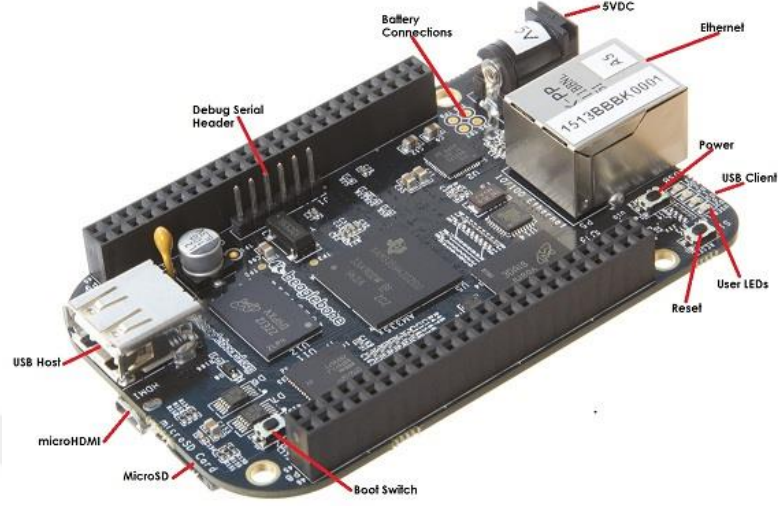
Herhangi bir üretici ya da sistemin protokol özellikleri baz alınarak oluşturulmamıştır. 1987 yılında binalarda farklı amaç ve hedefler için kurulmuş olan ve her bir farklı protokollerde haberleşen ve başka başka üreticilerin sistemlerinin birbirine entegre edilebilmesi mecburiyetinin, yüksek fiyat politikasını ortadan kaldırılması için standart bir iletişim protokolü oluşturma hedefiyle başlatılan çalışma 95 yılında üçüncü kere ve son kez revize edilerek onaylanmış ve bir ASHRAE standardı olarak basılmıştır. Yine aynı yıl BACnet standardı Amerikan Ulusal Standartları Enstitüsü (ANSI) tarafından ulusal bir standart olarak kabul edilmiştir (Bushby, 1997).

2.5. Beagle bone black

AKYÖN akıllı ev yönetim sisteminde mevcut sensör ve hareket eyleyiciler ile kullanıcılar arasındaki bağlantıyı sağlamak amacıyla boyutları küçük, enerji tüketimi az, maliyeti düşük beagle bone black gömülü sistem bilgisayarı tercih edilmiştir. Bu kısımda bu cihazın özelliklerinden bahsedilecektir.

Beaglebone Black, düşük fiyatlı olmakla beraber genişletilebilir odaklı üretilmiş, üzerinde Texas Instruments'ın geliştirdiği ARM Cortex A8 işlemci bulunduran bir mini bilgisayardır ve Beagleboard ailesinin en yeni üyesidir. Beaglebone Black cihazı üzerinde, 1 GHZ ve 2000 MIPS gibi özelliklere sahip ARM Cortex A8 işlemci, 800 MHZ frekansında çalışan ve 512 MB kapasiteli DDR3 RAM, 2 GB eMMC, 1280x1024 çıkış verebilen HDMI bağlantı noktası, RJ45 Ethernet girişi gibi gelişmiş özellikler barındırmaktadır. Cihaz 3,4" x 2,1" boyutundadır ve bu bir kredi kartı boyutuna tekabül etmektedir. Çevresel birim olarak 7 adet analog giriş, 4 adet

zamanlayıcı, seri haberleşme için I2C, CAN, SPI, UART, dijital giriş çıkış için 69 adet GPIO ve 8 adet PWM çıkışı vardır (URL-1).



Şekil 2.4. Beaglebone Black genel yapısı (URL-1)

65 possible digital I/Os

P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	GPIO_38	3	4	GPIO_39
VDD_5V	5	6	VDD_5V	GPIO_34	5	6	GPIO_35
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BTN	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
GPIO_30	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
GPIO_31	13	14	GPIO_50	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
GPIO_5	17	18	GPIO_4	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	GPIO_22	19	20	GPIO_63
GPIO_3	21	22	GPIO_2	GPIO_62	21	22	GPIO_37
GPIO_49	23	24	GPIO_15	GPIO_36	23	24	GPIO_33
GPIO_117	25	26	GPIO_14	GPIO_32	25	26	GPIO_61
GPIO_115	27	28	GPIO_113	GPIO_86	27	28	GPIO_88
GPIO_111	29	30	GPIO_112	GPIO_87	29	30	GPIO_89
GPIO_110	31	32	VDD_ADC	GPIO_10	31	32	GPIO_11
AIN4	33	34	GND_ADC	GPIO_9	33	34	GPIO_81
AIN6	35	36	AIN5	GPIO_8	35	36	GPIO_80
AIN2	37	38	AIN3	GPIO_78	37	38	GPIO_79
AIN0	39	40	AIN1	GPIO_76	39	40	GPIO_77
GPIO_20	41	42	GPIO_7	GPIO_74	41	42	GPIO_75
DGND	43	44	DGND	GPIO_72	43	44	GPIO_73
DGND	45	46	DGND	GPIO_70	45	46	GPIO_71

Şekil 2.5. Beaglebone Black genişleme yuvası GPIO yapısı (URL-2)

Beaglebone Black cihazının giriş ve çıkış pinlerinin sayıları diğer gömülü sistemlerde olduğu sınırlı sayıdadır. Bir pin üzerinden sadece bir tip işlem yapmak yerine, pinler seçimli yapıda tasarlanmış olup, bir adet pinin sekiz tipte kullanılması

sağlanmıştır. Şekil 2.5. te Beagle Bone Black sayısal giriş/çıkış pin şeması verilmiştir. GPIO' ların etkin bir şekilde kullanılması amacıyla Yiğit YÜCE tarafından Beagle Bone Black için blackLib kütüphanesi kullanılmıştır. BlackLib, Beaglebone Black cihazının analog girişlerinin okunması, ilgili çıkışlarından PWM üretilmesi ve GPIO pinlerinin kullanılabilmesi için tasarlanmış bir kütüphanedir (URL-3). Bunlara ek olarak, hata ayıklama üzerine yoğunlaşmış ve genel veya detaylı hata algılama özelliğine sahiptir.

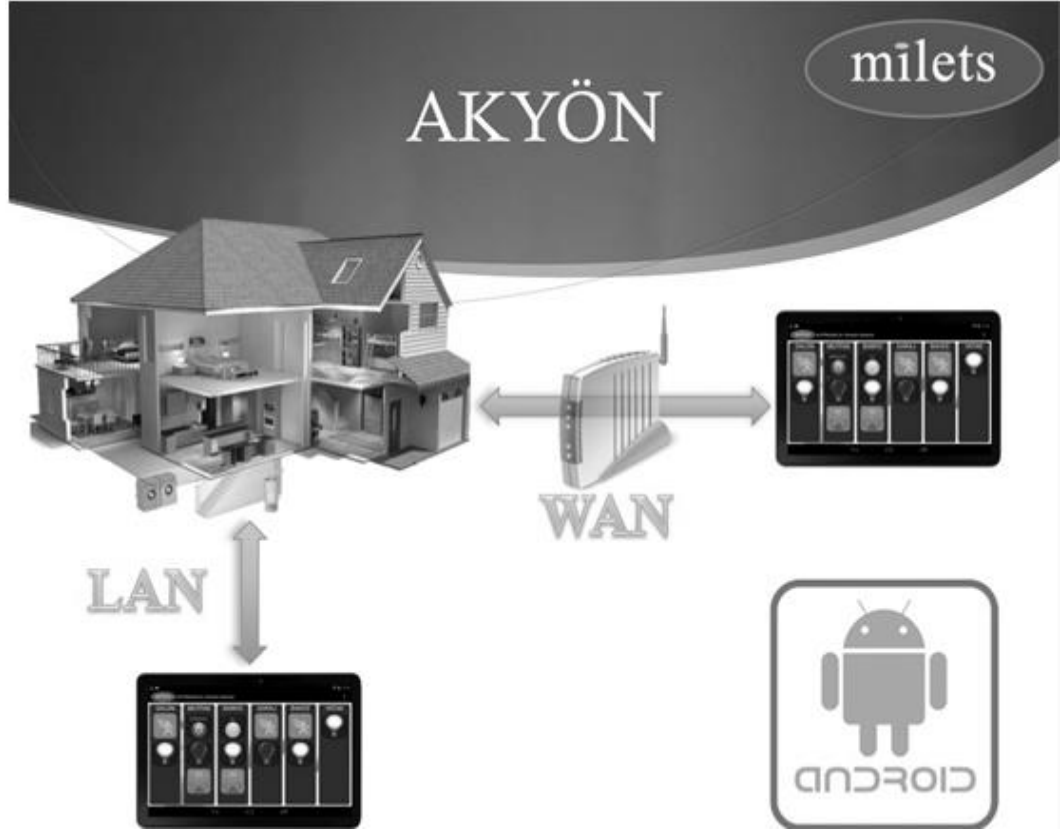


3. YAZILIM VE ALGORİTMALAR

Mobil cihazlar ile kontrol edilebilen akıllı ev yönetim sistemleri için haberleşme protokolü tasarımı ve gerçekleşmesi amaçlanan bu tez çalışmasında tasarlanan sistem yazılım ve algoritmaları bu bölümde anlatılmıştır.

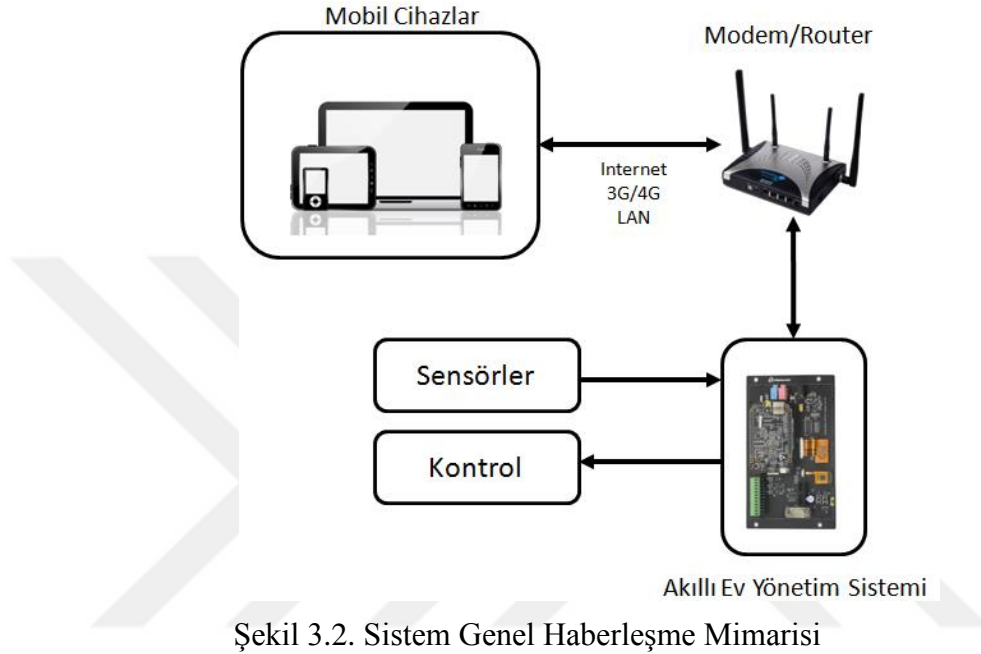
3.1. Sistem Mimarisi

Tez kapsamında hazırlanan yazılım ile akıllı ev yönetim sisteminin hem yerel ağ üzerinden hem de geniş alan ağları aracılığıyla kontrol edilmesi sağlanmıştır. Kullanılacak mobil cihazın yeteneklerine ek olarak 3G/4G sistemler üzerinden de sisteme erişilebilmekte ve yönetilmesine imkan verilmektedir. Sistemin genel mimarisi Şekil 3.1. de gösterildiği gibidir.



Şekil 3.1. Sistem Mimarisi

Yazılım ve algoritmaların gerçekleştirilmesi amacıyla Google firması tarafından geliştirilen ve açık kaynak kodlu Android mobil işletim sistemi kullanılmıştır. Yazılım ve algoritmalar MİLETS firması tarafından 113E033 nolu AKYÖN projesi kapsamında hazırlanan donanım ile birlikte kullanılmıştır. Şekil 3.2. de sistemin genel haberleşme mimarisi gösterilmiştir.

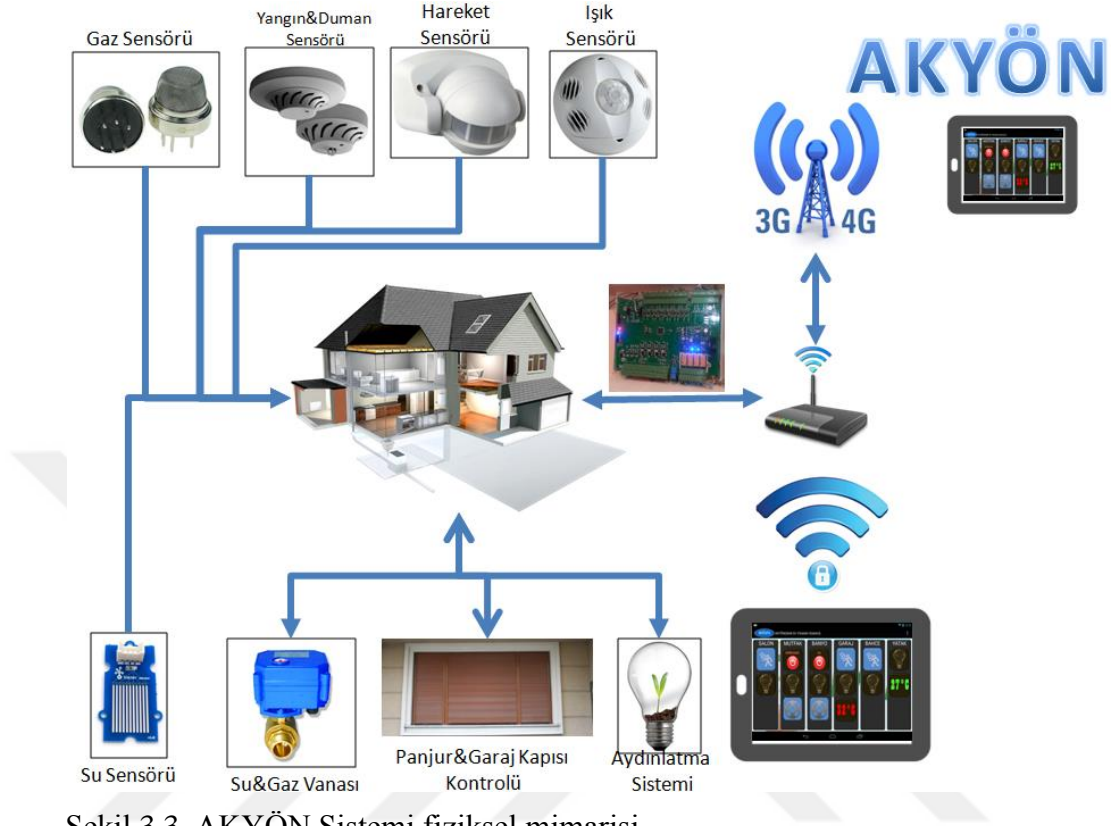


Tablo 3.1. BeagleBone Black sisteminin özellikleri

Özellik	Açıklama
İşlemci	Sitara AM3359AZCZ100 1Ghz, 2000 MIPS
Grafik	SGX530 3D
SDRAM	512MB DDR3L 606MHz
Flaş Bellek	4Gb, 8bit gömülü eMMC
Ölçüleri	88,98mm x 54,63mm x 18,84 mm
Çevre Birimleri	SPI, GPIOx65, LCD,MMC1,MMC2, 7AIN, 4xTimer, 3xUART,Ethernet
Ağırlık	39,8gr

Akıllı ev yönetim sistemi kapsamında ev içinde mevcut bulunan sensör ve cihazların yönetilmesi için hazırlanan elektronik sistemde sunucu olarak BeagleBone Black gömülü sistemi kullanılmıştır. Tablo 3.1. de BeagleBone Black özellikleri gösterilmiştir.

Şekil 3.3. te AKYÖN sisteminin fiziksel alt yapı mimarisi gösterilmiştir.



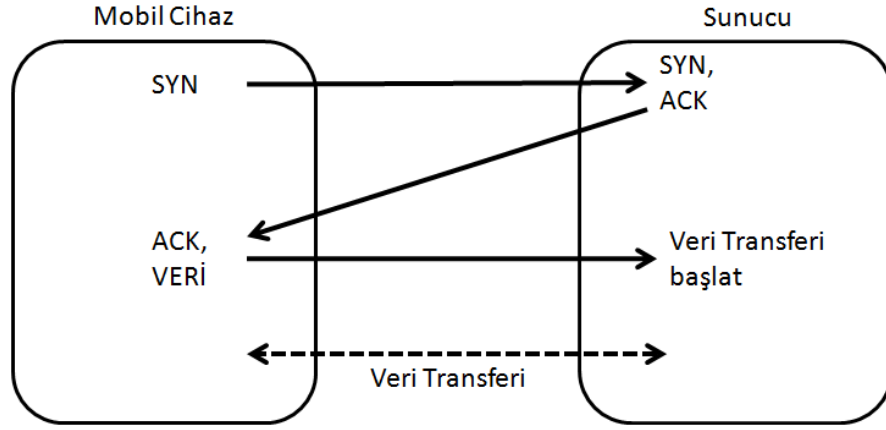
Şekil 3.3. AKYÖN Sistemi fiziksel mimarisi

3.2. Haberleşme Mimarisi

Sunucu ile mobil cihazlar arasında yapılan haberleşmeler TCP/IP protokolü kullanılarak gerçekleştirilmiştir.

3.2.1. İletişim protokolü

Sunucu ve mobil cihaz arasında güvenli haberleşmenin sağlanması amacıyla iki sistem arasında bir el sıkışma protokolü gerçekleştirilmiştir. El sıkışma yönteminin başarılı olması durumunda sistemler birbirleri arasında veri transferi işlemini gerçekleştirebilmektedir. İşlemin başarısız olması durumunda sunucu tarafından bağlantıya son verilmektedir. Şekil 3.4. te el sıkışma protokolü gösterilmiştir.

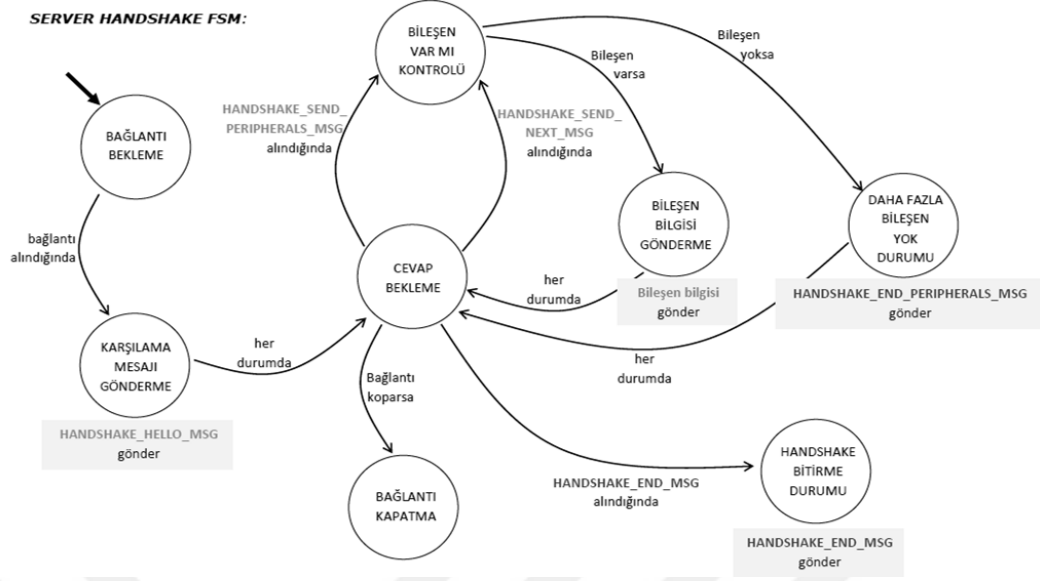


Şekil 3.4. El Sıkışma Protokolü

Sunucu bağlantı isteği gönderen mobil cihaza "HANDSHAKE_HELLO_MSG" paketini gönderir. Bu mesaj paketini alan mobil cihaz kullanıcı adı ve şifre bilgisini sunucu ile paylaşır. Sunucu veritabanından kullanıcı adı ve şifresini kontrol ettikten sonra bilgilerin doğru olması durumunda iletişime devam eder. Şekil 3.5. te Sunucu-Mobil cihaz arasındaki iletişim protokolü sonlu durum makinası olarak gösterilmiştir.

Mobil cihaz ve sunucu arasında başarılı bir şekilde bağlantı kurulması durumunda öncelikle sunucu ve mobil cihaz arasında sistemde mevcut bulunan bileşenlerin kontrolü yapılır, eğer sisteme yeni tanıtılmış olan bileşen(sensör vs.) varsa sunucu tarafından bunun bilgisi mobil cihaz ile paylaşılarak mobil cihaz tarafına eklenmesi sağlanır. Mobil cihaz bağlantıyı sonlandırana kadar belirlenen zaman aralığında sensör bilgileri yenilenerek kullanıcıya ekranda sergilenir.

Mobil cihaza ait IP bilgileri sunucu tarafından kullanıcı oturumunu sonlandırana kadar saklanır. Sensörlerde oluşabilecek bir alarm durumu gerçek zamanlı olarak sunucu tarafından otomatik olarak sisteme bağlı mobil cihazlara iletilir. Mobil cihazlar uyku durumunda dahi olsa arka tarafta sürekli olarak sistemi dinleyen ayrı bir kanal tarafından uyandırılarak hem sesli emde görsel olarak kullanıcının uyarılması sağlanır.



Şekil 3.5. Sunucu-Mobil Cihaz İletişim Protokolü

3.2.2. Haberleşme paket yapısı

Akıllı ev yönetim sisteminde mevcut bulunan sensör ve cihazların yönetilmesi ve toplanan verilerin aktarılması amacıyla iki tip mesaj paketi yapısı oluşturulmuştur. Birinci mesaj paket yapısı "Konfigürasyon mesaj paketi" olup, veri yapısı ve özellikleri Tablo 3.2. de verilmiştir.

Konfigürasyon mesaj paketi ile sisteme bağlı olan veya yeni bağlanan sensör/cihazların yapılandırma ayarları mobil cihaz aracılığıyla sunucuya iletilir. Sunucu gelen bilgilere göre sensör/cihazı sisteme ekler, ekli olan sensör/cihazda ise gerekli güncellemeleri yapar. Aynı mesaj paketi eğer sunucu tarafından mobil cihaza geliyorsa mobil cihaz tarafından tanımlı olan sensör ve cihazların konfigürasyon durumları güncellenir.

Konfigürasyon mesaj paketinde yer alan ilk byte olan "MESAJ_ID" alanı mesajın gelenmi, gidenmi mesaj paketi olduğunu en düşük değerli bit(LSB)' te saklar. LSB bitinin 0(sıfır) olması durumunda bu paketin mobil cihaz tarafından sunucuya, 1(bir) olması durumunda sunucu tarafından mobil cihaza gönderilen bir mesaj paketi olduğunu ifade eder.

Tablo 3.2. Konfigürasyon Mesaj Paket Yapısı

Tanım	Boyut	Açıklama
MESAJ_ID	1 Byte	0x10000000b : Gelen Konfigürasyon Paketi 0x10000001b : Giden Konfigürasyon Paketi
KULLANICI_ID	8 Byte	Kullanıcı Adı ve Zaman Pulunun bir araya gelmesi ile güvenlik amaçlı oluşturulmuş MD5 hash değeri
SENSOR_ID	2 Byte	Sensör için atanmış ID numarası
MODULE_ID	1 Byte	0 : BBB kart ID 1..N : wireless sensor ID
IO_ID	1 Byte	Sensor I/O numarası
DEVICE_TYPE	1 Byte	0x1: Yangın/Algılayıcı 0x2: Duman/Algılayıcı 0x3: PIR/Algılayıcı 0x4: Gaz/Algılayıcı 0x5: Işık/Algılayıcı 0x6: Nem/Algılayıcı 0x7: Sıcaklık/Algılayıcı 0x8: Oksijen/Algılayıcı 0x9: Lamba/Kontrol 0x10: Su Valfi/Kontrol 0x11: Gaz Valfi/Kontrol 0x12 : Alarm/Alarm
FLAG	1 Byte	0. Bit: Takılı/Takılı Değil 1. Bit: Aktif/Pasif 2. Bit: Çalışma Seviyesi 3. Bit: Geçici Devre Dışı 4. Bit: Açık/ Kaplı 5. Bit : wireless sensor
TIME_STAMP	8 Byte	Mesaj Gönderim Zamanı

Haberleşme protokolünde kullanılan ikinci mesaj paketi yönetim mesaj paketidir. Bu mesaj paketinin amacı gönderildiği yere göre değişmektedir. Eğer bu mesaj paketini gönderen taraf sunucu tarafı ise, sensöre ait durum bilgisi ve değerleri barındırır. Mobil cihaz tarafından gönderiliyorsa eğer sensör veya cihazın hangi zaman aralığında ve hangi günlerde açık/kapalı olacağını bildirir. Yönetim mesaj paket yapısı Tablo 3.3. de olduğu gibidir.

Tablo 3.3. Yönetim Mesaj Paket Yapısı

Tanım	Boyut	Açıklama
Mesaj ID	1 Byte	Paket tanımlayıcısıdır. Gelen ve giden yönetim paketine göre değeri değişir.
Kullanıcı ID	8 Byte	Güvenlik amaçlı oluşturulmuş, hash'i alınmış kullanıcı adı.
Bileşen ID	2 Byte	Her bir bileşen için atanmış eşsiz (unique) tanımlayıcıdır.
Veri	2 Byte	Herhangi bir veri gönderilmek istendiğinde, paketin bu bitlerine veri yerleştirilmektedir.
Bayraklar	1 Byte	İlgili bileşene ait bazı konfigürasyon bilgilerini tutar.
Başlangıç Zamanı	4 Byte	Planlanmış (scheduled) işlerin başlangıç zamanını gösterir.
Bitiş Zamanı	4 Byte	Planlanmış (scheduled) işlerin bitiş zamanını gösterir.
Günler	1 Byte	Planlanmış (scheduled) işlerin hangi günler yapılacağını gösterir.
Zaman Bilgisi	8 Byte	Paketin gönderim zamanını gösterir.

3.2.3. Sisteme yeni sensör cihaz eklenmesi

Hazırlanan akıllı ev yönetim sisteminde yazılım mimarisi sensör ve cihazların dinamik olarak eklenip çıkartılmasına izin verecek şekilde yapılmasını sağlayacak şekilde tasarlanmıştır. Sisteme yeni bir sensör veya cihaz modülü eklemek için "Konfigürasyon mesaj paketi" kullanılır. Sisteme bağlanabilecek sensör/cihaz tipleri ve numaraları Tablo 3.4. de verilmiştir. Kullanıcı tarafından mobil cihaz üzerinde yapılması gereken 3 temel yapılandırma ayarı Tablo 3.5' te verilmiştir.

Kullanıcı bir sensör veya cihaz modülünü sistem tanıtabilmesi için eklenecek oda/mekanın tanımlanmış olması gerekmektedir. Eğer oda/mekan daha önce tanımlanmamış ise "Ayarlar/Oda Ayarları" sekmesine giderek Şekil 3.6. da gösterilen "Oda Ayarları" arayüzünü kullanarak yeni oda/mekan tanımlamasını yapması gerekmektedir.

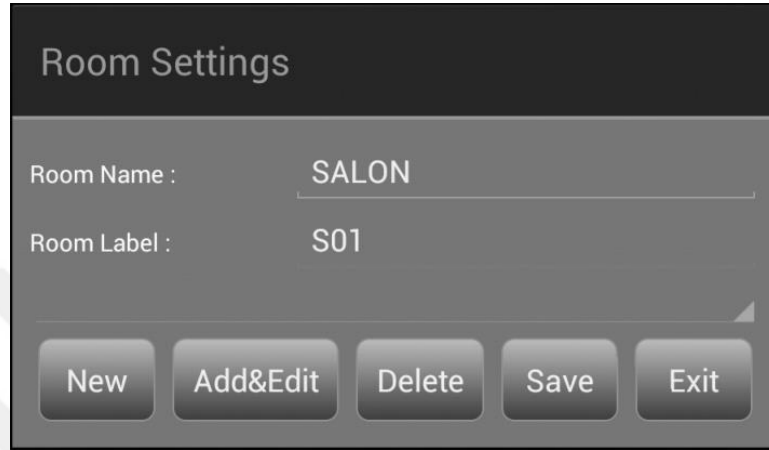
Tablo 3.4. Sisteme Eklenebilen Sensör/Cihaz Listesi

Tanım	Tipi	Değer
Bilinmiyor		0
Yangın	Sensor	1
Duman	Sensor	2
Pır	Sensor	3
Gaz	Sensor	4
Işık	Sensor	5
Nem	Sensor	6
Sıcaklık	Sensor	7
Oksijen	Sensor	8
Lamba	Çıkış	9
Su Valfi	Çıkış	10
Gaz Valfi	Çıkış	11
Alarm Butonu	Giriş	12
Deprem	Sensor	13
Su Baskını	Sensor	14
Klima	Çıkış	15
Fırın	Çıkış	16
Havalandırma	Çıkış	17
Sulama Sistemi	Çıkış	18
Kapı Otomatığı	Çıkış	19
Priz	Çıkış	20
Sis Üretici	Çıkış	21

Tablo 3.5. Temel Yapılandırma Ayarları

Sıra No.	Açıklama
1.	Kullanıcı tanımlama
2.	Oda/Mekan tanımlama
3.	Sensör/Cihaz tanımlama/ekleme

Oda/Mekan tanımlama işlemi tamamlandıktan sonra kullanıcı Şekil 3.7. de gösterilen "Sensör Ayarları" arayüzünü kullanarak eklemek istediği sensör/cihazı ve eklemek istediği odayı seçerek ekleme işlemini yapar. Mobil cihaz kullanıcı tarafından sensör/cihazın eklenmesine müteakip sunucuya yeni yapılandırma mesajını "Yapılandırma Konfigürasyon" mesaj paketini kullanarak iletir.



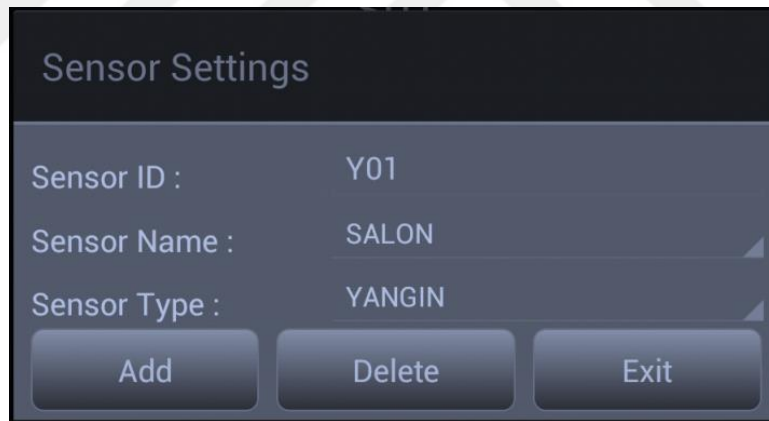
Room Settings

Room Name : SALON

Room Label : S01

New Add&Edit Delete Save Exit

Şekil 3.6. Oda Ayarları Arayüzü



Sensor Settings

Sensor ID : Y01

Sensor Name : SALON

Sensor Type : YANGIN

Add Delete Exit

Şekil 3.7. Sensör Ayarları Arayüzü

Kullanıcının sistemi etkili olarak yönetebilmesi amacıyla Şekil 3.8. de gösterilen kumanda panel arayüzü hazırlanmıştır. Oda/Mekana ve sensör/cihaz ayarlarında yapılan tüm değişiklikler kumanda panel arayüzünde eş zamanlı olarak yansıtılmaktadır.



Şekil 3.8. Kumanda Panel Arayüzü

3.3. Ayarların Saklanması

Sisteme bağlı mobil cihazın kullanıcı adı/şifrelerini ve ayarları saklamak için Android işletim sistemi tarafından sağlanan veri depolama yöntemlerinden biri olan, anahtar-değer ikilisini saklamak için hazırlanmış "shared preference" özelliği kullanılmıştır. Android işletim sistemi tarafından sağlanan Güvenliğin sağlanması amacıyla bu özelliğin gizli niteliği kullanılmış sadece uygulama tarafından erişilmesi sağlanmıştır.

Hash tablosu yapısındaki gibi anahtar-değer prensibi ile çalışan shared preferences yapısı sadece temel veri tipleri ile kullanılabilir. Saklayabildiği veri tipleri sırasıyla aşağıda belirtilmiştir;

- boolean
- float
- int
- long
- string

3.4. Haberleşme Performans Testi

Bu kısımda mobil kullanıcı arayüz ve beagle bone black arasında gerçekleşen haberleşme paketlerinin yerel ağ ve geniş alan ağlarındaki performans değerlendirilmesi yapılmıştır. Bu test kapsamında 23 byte'lık konfigürasyon paketi ve 31 byte'lık yönetim mesaj paketi kullanılmıştır. Test ev içi ortamda kablosuz ağ üzerinden farklı sinyal güçlerinde gerçekleştirilmiştir. Geniş alan ağlarında ise 3G ve 4G üzerinden mevcut iki farklı operatör aracılığıyla gerçekleştirilmiştir. Kablosuz ağlar ile ev ortamında yapılan işlemlerde mobil cihaz ve gömülü sistemler arasında erişim sürelerinin çok kısa olduğu gözlemlenmiştir. Paket kayıplarının ise ağın yoğunluğuna göre değişmekle beraber çok düşük olduğu gözlemlenmiştir. Tablo 3.6. da test sonuçları gösterilmiştir. Performans testleri sonucunda yerel alan ağında haberleşme esnasında problem yaşama oranının çok düşük olduğu gözlemlenmiştir. Fakat mobil ağlar üzerinden yapılan haberleşme esnasında paket kayıpları ve iletişim kurma problemlerinin %12,02 olduğu gözlemlenmiştir. Buda önemli uyarı paketlerinin yaşanacak bir problem dolayısıyla kaybedilebileceğini ve zafiyet yaratabileceğini göstermiştir.

Paketlerin bütünlüğü açısından değerlendirildiği zaman, teslim edilen tüm paketlerin hatasız olarak teslim edildiği görülmüştür. Aynı anda iki mobil cihaz ile yapılan bağlantılarda herhangi bir performans kaybı yaşanmamış olup hataya neden olmadığı gözlenmiştir.

Tablo 3.6. Haberleşme performans test sonuçları

	Kablosuz Yerel Ağ	Mobil Ağ
Gönderilen Paket Sayısı	30.000	30.000
Sinyal Şiddeti	%90	%90
En Hızlı Erişim Süresi	1 msn	38 msn
En Yavaş Erişim Süresi	1831 msn	11948 msn
Ortalama Erişim Süresi	4,97 msn	67,38 msn
Hata (Kayıp Paket Sayısı)	0	3608
Hata Oranı	%0,0	%12,02

4. SONUÇLAR VE ÖNERİLER

Bu çalışmada TCP/IP haberleşme protokolünü kullanarak Akıllı ev yönetim sistemi ile mobil cihazların haberleşme protokolü tanımlanmıştır. Oluşturulan bu protokolü kullanarak istemci/sunucu haberleşmesini kullanan Android işletim sistemine uygun mobil uygulama gerçekleştirilerek istemci-sunucu haberleşmesi kablosuz ağları kullanarak gerçekleştirilmesi sağlanmıştır. Kablosuz ağları kullanarak, sistemin yerel ağ bağımlılığı ortadan kaldırılarak kullanıcının sadece ev içinde değil uzak konumda da mobil iletişim şebekesini (GPRS/3G) kullanarak sisteme bağlanabilmesi sağlanmıştır.

Akıllı ev yönetim sistemine yeni bir cihaz/sensörün eklendikten sonra yapılandırılması işleminin kullanıcı tarafından yapılabilmesi, sistemin kurulum sonrasında da dinamik olarak yapılandırılabilmesine imkan sağlamıştır.

Güçül (2008) yüksek lisans tez çalışmasında akıllı ev sistemini uzaktan kontrol edebilmek için analog telefon hatlarında DTMF yönetimini kullanmıştır. Cincirop (2009) ve Algoiare (2014) yüksek lisans tez çalışmasında akıllı ev yönetim sistemini GSM üzerinden SMS alt yapısını kullanarak sağlamıştır. Sevil (2015) yüksek lisans tez çalışmasında, geliştirdiği sistemde haberleşme alt yapısı olarak seri haberleşme protokolünü kullanmıştır.

Bu haberleşme yöntemlerinin iletişim hızının yavaş, veri güvenliğinin düşük ve özellikle SMS yönteminin maliyetinin yüksek olmasından dolayı, daha hızlı ve güvenli olan ethernet ve kablosuz haberleşme alt yapıları üzerinden veri hatalarını en aza indirmek amacıyla TCP/IP haberleşme protokolü kullanılmıştır.

4.1. Öneriler

AKYÖN projesi kapsamında geliştirilen haberleşme altyapısı doğrudan gömülü sistem-mobil cihaz arasında iletişime dayanmaktadır. Mevcut sistemde yerel alan ağı

üzerinden haberleşme esnasında iletişim başarılı bir şekilde sağlanırken, mobil iletişim ağları üzerinden yapılan haberleşme esnasında paket gecikmesi ve kayıplarının yaşandığı tespit edilmiştir. Bu kayıpların engellenmesi ve sistemin merkezi bir şekilde yönetilebilmesi amacıyla sistemin merkezi sunucu mimarisine taşınarak mobil cihaz ve gömülü sistemin bir sunucu üzerinden haberleşmesinin sağlanması paket kayıplarını azaltacak ve oluşturulacak bir model ile tehlikeli durumlarda 3. kişilerin/kurumların uyarılmasını sağlayarak daha efektif olarak kullanılabilceği değerlendirilmektedir.

Ayrıca mevcut durumda haberleşme alt yapısı mesaj paketlerini şifreli bir şekilde taşıyacak şekilde tasarlanmamıştır. Bu durum gerçek ortamda kullanılması durumunda güvenlik zafiyetine sebep olacaktır. Bu sebepten dolayı haberleşme alt yapısında simetrik veya asimetrik şifreleme yöntemlerinden birini kullanılmasının sonradan oluşabilecek bu tür güvenlik zafiyetlerinin önüne geçebileceği değerlendirilmektedir.

KAYNAKLAR

Adao H., Antunes R., Grilo F., Web Based Control & Notification for Home Automation Alarm Systems, *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2008, 2(1), 74-78.

Algoiare O. T., Design and Implementation of Intelligent Home Using GSM Network, Yüksek Lisans Tezi, Çankaya Üniversitesi, Ankara, 2014, 372504.

Aslan A., Akıllı Ev Kavramı ve Otomasyon Sistemleri, Yüksek Lisans Tezi, Haliç Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2014, 357283.

Avincan G., Akıllı Bina Otomasyon Sistemleri ve Türkiye'deki Uygulamaları, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 1999, 83019.

Bayram U., Akıllı Ev Otomasyonu, Yüksek Lisans Tezi, Çanakkale Onsekiz Mart Üniversitesi, Fen Bilimleri Enstitüsü, Çanakkale, 2006, 185918.

Bekçibaşı U., Tenruh M., Telefon Şebekesi Üzerinden Şifre Güvenlikli Akıllı Ev Kontrol Sistemi, *Akademik Bilişim'11 - XIII. Akademik Bilişim Konferansı*, İnönü Üniversitesi, Malatya, Türkiye, 2-4 Şubat 2011.

Bir T., Endüstriyel Veri İletişiminde Tcp/Ip Uygulaması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2005, 166172.

Bushby S. T., BACnetTM: a standard communication infrastructure for intelligent buildings, *Automation in Construction*, 1997, 6(5), 529-540.

Cincirop B., GSM Kontrollü Akıllı Ev Otomasyonu, Yüksek Lisans Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, 2009, 245337.

Douligeris C., Intelligent Home Systems, *IEEE Communications Magazine*, 2002, 31(10), 52-61.

Erbey A., Akıllı Evler için Mobil Uygulama Geliştirilmesi, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü, Afyon, 2016, 420430.

Geçim S., Akıllı Konut Değerlendirme Modeli, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2002, 127166.

Gerhart J., *Home automation and wiring*, 1st, ed., McGraw Hill Professional, New York, 1999.

Goadrich M. H., Rogers M. P., Smart smartphone development: iOS versus Android, *SIGCSE-11*, Texas, USA, 9-12 Mart 2011.

Göktaş İ., Akıllı Ev Teknolojisi, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2006, 180389.

Güğü G. N., Akıllı ev sistemleri ev uygulaması, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2008, 212922.

Gül F., Akıllı Ev Sistemleri, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara, 2010, 269942.

Harris J., Wigginton M., *Intelligent Skins*, 1st ed., Architectural Press, New York, 2002.

Harrison A., Loe E., Read J., *Intelligent Buildings in South East Asia*, 1st ed., Taylor & Francis, Avusturalya, 1998.

Heerwagen D., *Passive and Active Environmental Controls – Informing the Schematic Design of Buildings*, 1st ed., McGraw-Hill, New York, 2003.

Işlak A. H., Baydere Ş., Kablosuz Duyarga Ağları ile Engelli İnsanlar İçin Akıllı Ev Uygulamaları, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 2011, 4(1).

Jamalabad M. S., Enerjisini İklımden Üreten Akıllı Evler, Yüksek Lisans Tezi, T.C. İstanbul Aydın Üniversitesi, Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, İstanbul, 2014, 394421.

Kardaş Y. Y., GSM Kontrollü Akıllı Ev Uygulaması, Yüksek Lisans Tezi, Dicle Üniversitesi, Fen Bilimleri Enstitüsü, Diyarbakır, 2014, 377381.

Kargacı M., Ev Tipi Klimalara Kablosuz Ağ Üzerinden Uzaktan Erişim Ve Kontrol İle Akıllı Ev Teknolojilerinin Geliştirilmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2015, 409758.

Khiyal M. S. H., Khan A., Shehzadi E., SMS Based Wireless Home Appliance Control System (HACS) for Automating Appliances and Security, *Issues in Informing Science and Information Technology*, 2009, 6, 887-894.

Kılıç H., Akıllı Binalar, Kurulmaları ve İşletilmeleri, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007, 222851.

Koçak Ç., Kırbas İ., Arduino Tabanlı Prototip Akıllı Ev Sistemi Tasarımı, *Akademik Bilişim '16 - XVIII. Akademik Bilişim Konferansı*, Aydın, Türkiye, 30 Ocak-5 Şubat 2016.

Kongaz H., Akıllı Ev Otomasyonunun Mikrodenetleyici ile Gerçekleştirilmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007, 213302.

Kovatsch M., Weiss M., Guinard D., Embedding internet technology for home automation, *Emerging Technologies and Factory Automation (ETFA)*, Berlin, Almanya, 13-16 Eylül 2010.

Küçükbakırcı C., Bir Mikro Denetleyici Tabanlı Akıllı Ev Sıcaklık Ve Aydınlatma Otomasyonu Uygulaması, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü, Fen Bilimleri Enstitüsü, Kocaeli, 2006, 182536.

Menteşoğlu M., Kavak A., Yakut M., Tangel A., Sahin S., Ozcan H., Mobil Cihazlar ile Kontrol Edilebilen Akıllı Ev Yönetim Sistemleri için Haberleşme Protokolü Tasarımı ve Gerçeklenmesi, *SİU 2015 IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Malatya, Türkiye, 16-19 Mayıs 2015.

Özçekiç E., Akıllı Evler, Yüksek Lisans Tezi, Beykent Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2005, 201957.

Pehlivan N., Alemdar H., Tunç C., Ersoy, C., Akıllı Evde İnsan Eylemi Tanıma ve Yorumlama: Bir Etiket Toplama ve Veri Görselleştirme Uygulaması, *Akademik Bilişim '15 - XVII. Akademik Bilişim Konferansı*, Eskişehir, Türkiye, 4-6 Şubat 2015.

Seçer F., Teknolojik Gelişmelerin Konut İç Mekan Tasarımına Etkisi ve Akıllı Evler, Sanatta Yeterlilik Tezi, Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2006, 184163.

Sevil M., Akıllı Binalar İçin Özgün Kablosuz Sensör Ağları Tasarımı İle Gerçek Zamanlı Enerji İzleme Ve Kontrolü, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2015, 406367.

Şahinoğlu G., Akıllı Evlerde Otomasyon, Yüksek Lisans Tezi, Marmara Üniversitesi Güzel Sanatlar Enstitüsü , İstanbul, 2006, 190626.

Tosunoğlu S. O., Akıllı Ev Sistemlerinde Merkezi Veri Toplama ve Cihaz Yönetimi, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü, Mühendislik ve Fen Bilimleri Enstitüsü, Kocaeli, 2009, 238769.

Türkcan Ü. S., Akıllı Konut Binalarında Etkileşim Kavramı ve Kentsel Ağ Bağlamında Bir Model, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007, 223292.

URL-1: <http://www.elinux.org/Beagleboard:BeagleBoneBlack>, (Ziyaret Tarihi : 6 Haziran 2016)

URL-2: <http://beagleboard.org/support/bone101>, (Ziyaret Tarihi : 6 Haziran 2016)

URL-3 : <http://blacklib.yigityuce.com/index.html>, (Ziyaret Tarihi : 3 Nisan 2016)

Yumurtacı M., Keçebaş A., Akıllı Ev Teknolojileri Ve Otomasyon Sistemleri, 5. *Uluslararası İleri Teknolojiler Sempozyumu (IATS'09)*, Karabük, Türkiye, 13-15 Mayıs 2009.



EKLER

EK-A GELİŞTİRİLEN SİSTEM İÇİN YAZILAN KAYNAK KODLAR

AkyonMain.Java

```
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.example.akyonv07.R;

public class AkyonMain extends Activity {
    static AkyonMain thisClass;
    private volatile boolean userOk = false;
    private static myUser userCurrent = null;

    // TCP Server Socket haberleşmesini sağlamak için kullanılan
    // Runnable arabirimini kullanan TCP Sınıfı
    public static class ThreadReceiveTCP implements Runnable {
        int i = 0;

        InetAddress ipadress;

        // TCP Socket bağlantısını gerçekleyen metot
        public void baglan() {
            try {

                socket.connect(new InetSocketAddress(DestIP, DestPort), 2000);
                socket.setSoTimeout(10);
                toServer = new DataOutputStream(socket.getOutputStream());
                fromServer = new DataInputStream(socket.getInputStream());
            } catch (IOException e) {
                Log.w("TCP Comm", "Connection is unsuccessfull " + DestIP + "/" +
                    DestPort);
            }

            handler.post(new Runnable() {
                public void run() {
                    setStateChange(socket.isConnected());
                }
            });
        }

        //TCP Socket listener için thread başlatma fonksiyonu
        @Override
        public void run() {
            socket = new Socket();

            baglan();
            if (!socket.isConnected())
                return;

            while (true) {
                final byte[] inMsg;
                try {
                    if (fromServer.available() > 0) {
```

```

inMsg = new byte[fromServer.available()];
if (msgWait)
refreshMsgWait(false);

fromServer.read(inMsg);

handler.post(new Runnable() {
public void run() {
ParseMessage(inMsg);
}
});
}

} catch (IOException e) {
Log.w("SERVER Thread Error", "Çok Pis Hata");
}

}

}

// UDP

static synchronized boolean getStateChange() throws
SocketException {
Log.w("Baglanti Durumu : ", ((Boolean)
socket.getKeepAlive()).toString());
return connectionStateBool;
}

static synchronized void refreshMsgWaitTime() {
msgWaitTime--;
}

static synchronized void refreshMsgWait(boolean bl) {
msgWait = bl;
}

static synchronized boolean getMsgWait() {
return msgWait;
}

static synchronized int getMsgWaitTime() {
return msgWaitTime;
}

static int msgWaitTime = 3000;
static boolean msgWait = true;

//TCP Socket kullanarak metoda gönderilen mesajı ilgili server' a
gönderen metot.
public static void SendMessageTCP(byte[] Message) {

Log.w("Giden Mesaj ", Arrays.toString(Message));
try {

toServer.write(Message);

```

```

toServer.flush();
// msgWaitTime = 1000;
// refreshMsgWait(true);
// while (msgWait)
// ;

} catch (IOException e) {
Log.w("SENDMESSAGE Hata TCP", e.toString());
}
}

// TCP Socket bağlantı durumunu ayarlayan ve buna uygun kullanıcı
arayüzünde değişikliği yapan metot
static synchronized void setStateChange(boolean in) {
connectionStateBool = in;
if (connectionStateBool == true) {
stateTW.setText("Connection is Successfull");
stateTW.setTextColor(Color.GREEN);
} else {
stateTW.setText("Connection is not Successfull");
stateTW.setTextColor(Color.RED);
}
}

// Variables
static Handler handler = new Handler();
// Sensor List
public static ArrayList<myImage> SensorList = new
ArrayList<myImage>();
// Room List
public static ArrayList<myLayout> RoomList = new
ArrayList<myLayout>();
// User List
public static ArrayList<myUser> UserList = new
ArrayList<myUser>();
// Network Settings
public static final String NTSet = "NetworkSettings";
public static int DestPort = 0;

public static int RcvPort = 0;
public static String DestIP = "";
static Thread Listener;
boolean listenerBool = true;

boolean sensorAdd = false;

private boolean sifreAyniFarkli = false;

public static TextView stateTW;

// 0 : TCP
// 1 : UDP
private static int CommunicationType = 0;

private static int tpTCP = 0;

private static int tpUDP = 1;

final public Context context = this;

```

```

static DataOutputStream toServer;

static DataInputStream fromServer;

static Socket socket;

static boolean connectionStateBool = false;

private void closeProject() {
    android.os.Process.killProcess(android.os.Process.myPid());
}

private void closeProject(String Tmp) {
    Log.w("EXIT", Tmp);
    android.os.Process.killProcess(android.os.Process.myPid());
}

public void CreateRooms(String RoomName, String RoomLabel) {
    // From string resources find
    LinearLayout rootLay = (LinearLayout)
    findViewById(getResources().getIdentifier("LinearAltRoot", "id",
    getPackageName()));

    // LinearLayout rootLay =
    (LinearLayout) findViewById(R.id.LayoutRoot);

    Log.w("RoomList", RoomName + " " + RoomLabel);
    final myLayout A = new myLayout(getApplicationContext(), null);
    A.setOrientation(LinearLayout.VERTICAL);
    A.RoomName = RoomName;
    A.RoomLabel = RoomLabel;
    TextView t = new TextView(getApplicationContext());
    LinearLayout.LayoutParams TextlayoutParams = new
    LinearLayout.LayoutParams(LayoutParams.MATCH_PARENT,
    LayoutParams.WRAP_CONTENT, 1.0f);
    TextlayoutParams.setMargins(5, 5, 5, 5);
    // t.setLayoutParams(TextlayoutParams);
    t.setText(A.RoomLabel);
    t.setTextColor(Color.rgb(255, 255, 255));
    t.setTextSize(28);
    t.setGravity(Gravity.TOP | Gravity.CENTER);
    A.addView(t);
    A.setBackground(getResources().getDrawable(R.drawable.layout_bg));
    LinearLayout.LayoutParams layoutParams = new
    LinearLayout.LayoutParams(LayoutParams.MATCH_PARENT,
    LayoutParams.MATCH_PARENT, 1.0f);
    layoutParams.setMargins(1, 1, 1, 1);
    layoutParams.weight = 1;
    A.setLayoutParams(layoutParams);

    rootLay.addView(A);
    A.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            if (!sensorAdd)
                return;
            Log.w("Layout", A.RoomLabel);
            final Dialog dialog = new Dialog(context);

```

```

dialog.setContentView(R.layout.sensorsetting);
dialog.setTitle("Sensor Settings");
dialog.setCancelable(false);
final EditText EditSensorID = (EditText)
dialog.findViewById(R.id.editsensorid);

ArrayList<String> TmpAry = new ArrayList<String>();
for (myLayout x : RoomList) {
TmpAry.add(x.RoomName);
}

final Spinner spinSensorNames = (Spinner)
dialog.findViewById(R.id.spinnersensorname);

ArrayAdapter<String> adapter = new ArrayAdapter<String>(context,
android.R.layout.simple_spinner_item, TmpAry);
spinSensorNames.setAdapter(new ArrayAdapter<String>(context,
android.R.layout.simple_spinner_dropdown_item, TmpAry));
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dr
opdown_item);
spinSensorNames.setAdapter(adapter);

final Spinner spinSensorType = (Spinner)
dialog.findViewById(R.id.spinnersensortype);

ArrayAdapter<CharSequence> adapterSType =
ArrayAdapter.createFromResource(context, R.array.SensorTypes,
android.R.layout.simple_spinner_item);
adapterSType.setDropDownViewResource(android.R.layout.simple_spinn
er_dropdown_item);
spinSensorType.setAdapter(adapterSType);

Button SensorAdd = (Button)
dialog.findViewById(R.id.buttonsensoradd);
SensorAdd.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
String SnsID = EditSensorID.getText().toString();
String SnsNm = spinSensorNames.getSelectedItem().toString();
String SnsTp = spinSensorType.getSelectedItem().toString();
if (SnsID.equals("")) {
Log.w("Button", "SensorNotAdd");
dialog.dismiss();
return;
}
Log.w("Button", "SensorAdded");
for (myLayout x : RoomList) {
if (x.RoomName.equals(SnsNm)) {
x.addView(createSensor(SnsID, SnsNm, SnsTp));
break;
}
}
onSaveSensor();
dialog.dismiss();
}
});

Button SensorDelete = (Button)

```

```

dialog.findViewById(R.id.buttoncancel);
SensorDelete.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
String SnsID = EditSensorID.getText().toString();
String SnsNm = spinSensorNames.getSelectedItem().toString();
for (myImage x : SensorList) {
if (x.IO_ID.equals(SnsID)) {
((ViewManager) x.getParent()).removeView(x);
SensorList.remove(x);
break;
}
}
onSaveSensor();
dialog.dismiss();
}
});

Button SensorExit = (Button)
dialog.findViewById(R.id.buttonexit);
SensorExit.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
String SnsID = EditSensorID.getText().toString();
String SnsNm = spinSensorNames.getSelectedItem().toString();
onSaveSensor();
dialog.dismiss();
}
});

dialog.show();
}
});
RoomList.add(A);

}

private myImage createSensor(String SensorID, String SensorName,
String SensorType) {
myImage tmpNew = new myImage(context, null);
tmpNew.IO_ID = SensorID;
tmpNew.Name = SensorName;
tmpNew.Type = SensorType;
tmpNew.setSensorImage();
SensorList.add(tmpNew);

return SensorList.get(SensorList.size() - 1);

}

private void onClearRoomList() {
LinearLayout Tlout = (LinearLayout)
findViewById(R.id.LinearAltRoot);
for (myLayout x : RoomList) {
Tlout.removeView(x);
}
}
Log.w("OnSaveRoom", "Start");

```

```

SharedPreferences rooms = getSharedPreferences("RoomList",
MODE_PRIVATE);
String TempRoom = "";
SharedPreferences.Editor editor = rooms.edit();
editor.putString("RoomLists", TempRoom);
editor.commit();
}

private void onClearSensor() {
LinearLayout TLout = (LinearLayout)
findViewById(R.id.LinearAltRoot);
for (myImage x : SensorList) {
TLout.removeView(x);
}
SharedPreferences rooms = getSharedPreferences("SensorList",
MODE_PRIVATE);
String TempRoom = "";
SharedPreferences.Editor editor = rooms.edit();
editor.putString("SensorLists", TempRoom);
editor.commit();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
Log.w("AkyonMain", "Program is started");
SensorList.clear();
super.onCreate(savedInstanceState);
thisClass = this;
if (Build.VERSION.SDK_INT >= 9) {
StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
StrictMode.setThreadPolicy(policy);
}
setContentView(R.layout.activity_akyon_main);
getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_O
N);
// Kullanici Listesi Olusturulmusmu

if (!onLoadUser()) {
openNewUserLoginWindow();
} else {
openUserLoginWindow();
}
}

private void openUserLoginWindow() {
// Kullanıcı Ekranı Sifre Dialog Penceresi
final Dialog dialog = new Dialog(context);
dialog.setContentView(R.layout.userlogin);
dialog.setTitle("Kullanici Girisi");
dialog.setCancelable(false);
final EditText userName = (EditText)
dialog.findViewById(R.id.EditUserName);
final EditText userPass = (EditText)
dialog.findViewById(R.id.EditUserPass);
Button okButton = (Button)
dialog.findViewById(R.id.buttonUserLogin);
Button cancelButton = (Button)

```

```

dialog.findViewById(R.id.buttonUserCancel);

okButton.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
// Check User Name and Password
String tempUserName = userName.getText().toString();
String tempUserPass = userPass.getText().toString();

for (myUser x : userList) {
if (x.isSame(tempUserName, tempUserPass)) {
Toast.makeText(context, "ŞİFRE DOGRU", Toast.LENGTH_LONG).show();
userCurrent = x;

stateTW = (TextView) findViewById(R.id.comState);
RoomList.clear();
onLoadNT();
// Listener Baslat
if (CommunicationType == tpTCP) {
Listener = new Thread(new ThreadReceiveTCP());
Listener.start();

}

onLoadRoom();
onLoadSensor();
dialog.cancel();
return;
}
}
Toast.makeText(context, "KULLANICI ADI/ŞİFRE YANLIŞ",
Toast.LENGTH_LONG).show();

}
});

cancelButton.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
closeProject();
finish();
}
});

dialog.show();

}

private void openNewUserLoginWindow() {
Toast.makeText(context, "Kullanıcı Olusturulmamış",
Toast.LENGTH_LONG).show();
final Dialog dialog = new Dialog(context);
dialog.setContentView(R.layout.newuserlogin);
dialog.setTitle("Yeni Kullanıcı Girişi");
dialog.setCancelable(false);

final EditText newUserPassOk = (EditText)

```

```

dialog.findViewById(R.id.EditNewUserPassOk);
final EditText newUserPass = (EditText)
dialog.findViewById(R.id.EditNewUserPass);
final EditText newUserName = (EditText)
dialog.findViewById(R.id.EditNewUserName);
final TextView labelSifre = (TextView)
dialog.findViewById(R.id.labelSifreOnay);
final RadioGroup radioGroupUserType = (RadioGroup)
dialog.findViewById(R.id.commType);

// Yeni kullanıcı Ekranı ekleme
newUserPassOk.addTextChangedListener(new TextWatcher() {

@Override
public void onTextChanged(CharSequence s, int start, int before,
int count) {
}

@Override
public void beforeTextChanged(CharSequence s, int start, int
count, int after) {
}

@Override
public void afterTextChanged(Editable s) {

if
(!newUserPassOk.getText().toString().equals(newUserPass.getText().
toString())) {
labelSifre.setTextColor(Color.RED);
sifreAyniFarkli = false;
} else {
labelSifre.setTextColor(Color.GREEN);
sifreAyniFarkli = true;
}

}

});

Button btnOk = (Button) dialog.findViewById(R.id.buttonUserLogin);
btnOk.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
String userName, userPass;
String userType = "";
int selectedId = radioGroupUserType.getCheckedRadioButtonId();
RadioButton radioUserType = (RadioButton)
dialog.findViewById(selectedId);
userType = radioUserType.getText().toString();

if (sifreAyniFarkli) {
UserList.add(new
myUser(myUser.SHA1Converter(newUserName.getText().toString()),
myUser.SHA1Converter(newUserPassOk.getText()
.toString()), userType));
dialog.cancel();
onSaveUser();
}
}
}

```

```

} else {
Toast.makeText(context, "Girdiginiz sifreler farkli",
Toast.LENGTH_LONG).show();
}

}

});

// Kullanici olusturmayı iptal et
Button btnCancel = (Button)
dialog.findViewById(R.id.buttonUserCancel);
btnCancel.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
closeProject();
finish();
}
});

dialog.show();

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is
present.
getMenuInflater().inflate(R.menu.newmenu, menu);
return true;
}

public void onDestroy() {
super.onDestroy();
}

private void onLoadNT() {
SharedPreferences NTSettings = getSharedPreferences(NTSet,
MODE_PRIVATE);
DestIP = NTSettings.getString("DestinationIP", "127.0.0.1");
DestPort = NTSettings.getInt("DestinationPort", 8001);
RcvPort = NTSettings.getInt("ReceivePort", 5001);
CommunicationType = NTSettings.getInt("CommType", tpTCP);
}

private void onLoadRoom() {
RoomList.clear();
SharedPreferences Room = getSharedPreferences("RoomList",
MODE_PRIVATE);
String TempRoom = Room.getString("RoomLists", "");
Log.w("OnLoad", TempRoom);
StringTokenizer st = new StringTokenizer(TempRoom, "|");
while (st.hasMoreElements()) {
StringTokenizer stn = new StringTokenizer(st.nextToken(), ":");
CreateRooms(stn.nextToken(), stn.nextToken());
}
}

private void onLoadSensor() {

```

```

SensorList.clear();
SharedPreferences Room = getSharedPreferences("SensorList",
MODE_PRIVATE);
String TempRoom = Room.getString("SensorLists", "");
Log.w("OnLoad", TempRoom);
StringTokenizer st = new StringTokenizer(TempRoom, "|");
while (st.hasMoreElements()) {
StringTokenizer stn = new StringTokenizer(st.nextToken(), ":");
String SnsId = stn.nextToken();
String SnsName = stn.nextToken();
String SnsTyp = stn.nextToken();
for (myLayout x : RoomList) {
if (x.RoomName.equals(SnsName)) {
x.addView(createSensor(SnsId, SnsName, SnsTyp));
break;
}
}
}

private boolean onLoadUser() {
UserList.clear();

SharedPreferences userList = getSharedPreferences("UserList",
MODE_PRIVATE);
if (!userList.contains("UserList")) {
Log.w("Kullanıcılar ", "Kullanıcı Yok");
return false;
}
Log.w("Kullanıcılar ", "Kullanıcı Var");
String TempUser = userList.getString("UserList", "");
StringTokenizer st = new StringTokenizer(TempUser, "|");
while (st.hasMoreElements()) {
StringTokenizer stn = new StringTokenizer(st.nextToken(), ":");
UserList.add(new myUser(stn.nextToken(), stn.nextToken(),
stn.nextToken()));
}

return true;
}

private void onAddUser() {
}

// Menu Click
public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {

case R.id.armMode:
MenuItem mi = item;
if(mi.isCheckable() && mi.isChecked()) {
mi.setChecked(false);
//protection = false;

}else{
mi.setChecked(true);
//protection = true;
}
}
}

```

```

SendMessageTCP( createConfigurationPackage(0, (byte)0, (byte)0,
(byte)0, protection ? (byte)0 : (byte)255) );
return true;
case R.id.settings:
openConnectionSettingWindow();
return true;
case R.id.roomsetting:
if(userCurrent.getUserType()!=0){
Toast.makeText(context, "Bu işlemi yapabilmek için yetkili
kullanıcı değilsiniz", Toast.LENGTH_LONG).show();
return true;
}
openRoomSettingWindow();
return true;
case R.id.SensorSettings:
Log.w("MENU", "SENSORS");
if(userCurrent.getUserType()!=0){
Toast.makeText(context, "Bu işlemi yapabilmek için yetkili
kullanıcı değilsiniz", Toast.LENGTH_LONG).show();
return true;
}
MenuItem ms = item;
if(item.isCheckable() && item.isChecked()){
item.setChecked(false);
sensorAdd = false;
}else{
item.setChecked(true);
sensorAdd = true;
}
return true;
case R.id.reset:
if(userCurrent.getUserType()!=0){
Toast.makeText(context, "Bu işlemi yapabilmek için yetkili
kullanıcı değilsiniz", Toast.LENGTH_LONG).show();
return true;
}
menuReset();

return true;
case R.id.NewUser:
if(userCurrent.getUserType()!=0){
Toast.makeText(context, "Bu işlemi yapabilmek için yetkili
kullanıcı değilsiniz", Toast.LENGTH_LONG).show();
return true;
}
openNewUserLoginWindow();
return true;
case R.id.alarm:
Log.w("MENU", "ALARM");
SendMessageTCP( createConfigurationPackage(1, (byte)1, (byte)1,
(byte)1, protection ? (byte)0 : (byte)255) );

// SendMessage(Message.toString());
return true;
case R.id.help:
Log.w("MENU", "HELP");
return true;
case R.id.exit:
Log.w("MENU", "EXIT");

```

```

closeProject();
finish();
return true;
default:
return super.onOptionsItemSelected(item);
}
}

private void openRoomSettingWindow() {
Log.w("MENU", "ROOM SETTINGS");
try {
final Dialog dialog = new Dialog(context);
dialog.setContentView(R.layout.roomsettings);
dialog.setTitle("Room Settings");
dialog.setCancelable(false);
final Spinner spin = (Spinner)
dialog.findViewById(R.id.spinnerUser);
Button NewButton = (Button) dialog.findViewById(R.id.ButtonNew);
Button AddButton = (Button)
dialog.findViewById(R.id.ButtonAddEdit);
Button SaveButton = (Button) dialog.findViewById(R.id.ButtonSave);
Button DeleteButton = (Button)
dialog.findViewById(R.id.ButtonDelete);
final EditText RoomName = (EditText)
dialog.findViewById(R.id.EditRoomName);
final EditText RoomLabel = (EditText)
dialog.findViewById(R.id.EditRoomLabel);

final ArrayList<String> TmpAry = new ArrayList<String>();
for (myLayout x : RoomList) {
TmpAry.add(x.RoomLabel);
}
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, TmpAry);
spin.setAdapter(new
ArrayAdapter<String>(this.getApplicationContext(),
android.R.layout.simple_spinner_dropdown_item, TmpAry));
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dr
opdown_item);
spin.setAdapter(adapter);

NewButton.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
RoomName.setText("");
RoomLabel.setText("");
}
});

AddButton.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
boolean Durum = false;
if (RoomName.getText().toString().equals("") ||
RoomLabel.getText().toString().equals("")) {
Log.w("ADD ROOM", "NOT NULL");
return;
}
}
for (myLayout x : RoomList) {

```

```

if (x.RoomName.equals(RoomName.getText().toString())) {
x.RoomLabel = RoomLabel.getText().toString();
Durum = true;
break;
}
}

if (!Durum) {
myLayout tmp = new myLayout(context, null);
Log.w("Room", RoomName.getText().toString() + " " +
RoomLabel.getText().toString());
CreateRooms(RoomName.getText().toString(),
RoomLabel.getText().toString());
}
onSaveRoom();
}
});

SaveButton.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
Log.w("Click", "Kayıt");
onSaveRoom();
}
});

DeleteButton.setOnClickListener(new OnClickListener() {

@Override
public void onClick(View v) {
int ind = spin.getSelectedItemId();
LinearLayout roomlay = (LinearLayout)
findViewById(R.id.LinearAltRoot);
roomlay.removeView(RoomList.get(ind));
RoomList.remove(ind);
onSaveRoom();
}
});

Button exitButton = (Button) dialog.findViewById(R.id.ButtonExit);
// if button is clicked, close the custom dialog
exitButton.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
dialog.dismiss();
}
});

spin.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

@Override
public void onItemClick(AdapterView<?> arg0, View arg1, int
arg2, long arg3) {
int ind = spin.getSelectedItemId();
RoomName.setText(RoomList.get(ind).RoomName);
RoomLabel.setText(RoomList.get(ind).RoomLabel);
}
}
}

```

```

@Override
public void onNothingSelected(AdapterView<?> arg0) {
    return;
}
});
dialog.show();
} catch (Exception e) {
    Log.w("Dialogs", e.toString());
}

}

private void menuReset() {
    Log.w("MENU", "RESET");

    AlertDialog.Builder alt_bld = new AlertDialog.Builder(context);
    alt_bld.setMessage("Veritabanını Silmek istediğimize
eminmisiniz").setCancelable(false)
    .setPositiveButton("YES", new DialogInterface.OnClickListener() {

@Override
public void onClick(DialogInterface dialog, int which) {

    Log.w("RESET", "Tüm Veritabanı Silindi");
    onClearSensor();
    onClearRoomList();
    onLoadRoom();
    onLoadSensor();
}
}).setNegativeButton("NO", new DialogInterface.OnClickListener() {

@Override
public void onClick(DialogInterface dialog, int which) {

    Log.w("RESET", "Veritabanı Silme işlemi iptal Edildi");
}
});

AlertDialog alert = alt_bld.create();
alert.setTitle("DataBase Reset");
alert.setIcon(R.drawable.miletslogomini);
alert.show();

}

private void openConnectionSettingWindow() {
    if (userCurrent.getUserType() != 0) {
        Toast.makeText(context, "Bu ayarlar için yetkili değilsiniz",
        Toast.LENGTH_LONG).show();
        return;
    }
    Log.w("MENU", "SETTINGS");
    try {
        final Dialog dialog = new Dialog(context);
        dialog setContentView(R.layout.settingslayout);
        dialog.setTitle("Settings");
        dialog.setCancelable(false);
        // İletisinin TCP/UDP üzerinden sağlanacağı

```

```

final RadioGroup commTypeRadioGroup = (RadioGroup)
dialog.findViewById(R.id.commType);

final EditText DstIP_ = (EditText)
dialog.findViewById(R.id.EditUserName);
final EditText DstPrt_ = (EditText)
dialog.findViewById(R.id.EditUserPass);
final EditText RcvPrt_ = (EditText)
dialog.findViewById(R.id.editText3);
DstIP_.setText(DestIP);
DstPrt_.setText(String.valueOf(DestPort));
RcvPrt_.setText(String.valueOf(RcvPort));
if (CommunicationType == 0) {
commTypeRadioGroup.check(R.id.radioTCP);
} else if (CommunicationType == 1) {
commTypeRadioGroup.check(R.id.radioUDP);
}

// Button Save
Button saveButton = (Button)
dialog.findViewById(R.id.buttonUserLogin);
// if button is clicked, close the custom dialog
saveButton.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
RadioButton choosenRadiobutton;
int selectedId = commTypeRadioGroup.getCheckedRadioButtonId();
choosenRadiobutton = (RadioButton)
dialog.findViewById(selectedId);
String tercih = choosenRadiobutton.getText().toString();
int commType = 0;
if (terjih.equals("TCP")) {
commType = tpTCP;
} else if (terjih.equals("UDP")) {
commType = tpUDP;
}

String TmpDstIP = DstIP_.getText().toString();
int TmpDstPort = Integer.parseInt(DstPrt_.getText().toString());
int TmpRcvPort = Integer.parseInt(RcvPrt_.getText().toString());
onSaveNT(TmpDstIP, TmpDstPort, TmpRcvPort, commType);
dialog.dismiss();
}
});
// Button Exit
Button exitButton = (Button)
dialog.findViewById(R.id.buttonUserCancel);
// if button is clicked, close the custom dialog
exitButton.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
dialog.dismiss();
}
});

dialog.show();

} catch (Exception e) {
Log.w("Dialogs", e.toString());
}

```

```

}

}

// commType
// iletişim türünü belli eder 0:TCP 1:UDP
private void onSaveNT(String DstIP, int DstPrt, int RcvPrt, int
commType) {
SharedPreferences settings = getSharedPreferences("NTSet",
MODE_PRIVATE);
SharedPreferences.Editor editor = settings.edit();
editor.putString("DestinationIP", DstIP);
editor.putInt("DestinationPort", DstPrt);
editor.putInt("ReceivePort", RcvPrt);
editor.putInt("CommType", commType);

// Commit the edits!
editor.commit();
super.recreate();
}

private void onSaveUser() {
Log.w("OnUserSave", "Start");
SharedPreferences rooms = getSharedPreferences("UserList",
MODE_PRIVATE);
String TempUser = "";
for (myUser x : UserList) {
TempUser += (x.UserName + ":" + x.UserPass + ":" + x.userType +
"|");
}
SharedPreferences.Editor editor = rooms.edit();
editor.putString("UserList", TempUser);
editor.commit();

// Commit the edits!
editor.commit();
super.recreate();
}

private void onSaveRoom() {
Log.w("OnSaveRoom", "Start");
SharedPreferences rooms = getSharedPreferences("RoomList",
MODE_PRIVATE);
String TempRoom = "";
for (myLayout x : RoomList) {
TempRoom += (x.RoomName + ":" + x.RoomLabel + "|");
}
SharedPreferences.Editor editor = rooms.edit();
editor.putString("RoomLists", TempRoom);
editor.commit();
}

private void onSaveSensor() {
Log.w("OnSaveSensor", "Start");
SharedPreferences rooms = getSharedPreferences("SensorList",
MODE_PRIVATE);
String TempRoom = "";
for (myImage x : SensorList) {
TempRoom += (x.IO ID + ":" + x.Name + ":" + x.Type + "|");
}
}

```

```

}
SharedPreferences.Editor editor = rooms.edit();
editor.putString("SensorLists", TempRoom);
editor.commit();
}

final static String HANDSHAKE_HELLO_MSG = "hello";
final static String HANDSHAKE_SEND_PERIPHERAL_MSG =
"sendMePeripherals";
final static String HANDSHAKE_SEND_NEXT_MSG =
"okIGotThis,SendMeNext";
final static String HANDSHAKE_END_PERIPHERALS_MSG =
"endPeripherals";
final static String HANDSHAKE_END_HANDSHAKE_MSG = "bye";

static boolean HandShake = false;

public static void ParseMessage(byte[] msg) {
Log.w("Gelen Mesaj : ", Arrays.toString(msg));
String inMsg = new String(msg);

if (inMsg.equals(HANDSHAKE_HELLO_MSG)) {
HandShake = true;
SendMessageTCP(HANDSHAKE_SEND_PERIPHERAL_MSG.getBytes());
return;
}

if (HandShake) {
if (inMsg.equals(HANDSHAKE_END_PERIPHERALS_MSG)) {
HandShake = false;
SendMessageTCP(HANDSHAKE_END_HANDSHAKE_MSG.getBytes());
return;
}
SendMessageTCP(HANDSHAKE_SEND_NEXT_MSG.getBytes());
return;
}

if (msg[0] == 65) {

int peripheralId = (int) (((msg[9] & 0xFF) << 8) | (msg[10] &
0xFF));
int data = (int) (((msg[11] & 0xFF) << 8) | (msg[12] & 0xFF));

if( peripheralId == 0 )
{
if( data == 0 )
{
protection = false;
}
else
{
protection = true;
}
}

if (msg[10] >= 30 && msg[10] <= 47) {
short io_id = ByteBuffer.wrap(Arrays.copyOfRange(msg, 9,
11)).order(ByteOrder.BIG_ENDIAN).getShort();
Log.w("IOID",String.valueOf(io_id));
}
}

```

```

int state = msg[12];

for (myImage st : SensorList) {
if (st.IO_ID.equals(String.valueOf(msg[10]))) {
st.setImage(state);
return;
}
}

if(msg[0]==49)
{
int peripheralId = (int) ((msg[9] & 0xFF) << 8) | (msg[10] &
0xFF));
byte flags = msg[14];

if( peripheralId == 0 )
{
if( flags == CR_PROTECTION_DEACTIVATED )
{
protection = false;
}
else
if( flags == CR_PROTECTION_ACTIVATED )
{
protection = true;
}
return;
}

if( flags == CR_ALARM_TRIGGERED )
{
for (myImage st : SensorList)
{
if (st.IO_ID.equals(String.valueOf(peripheralId)))
{
st.setImage(2); // 2 is alarm
return;
}
}
}

public void StateAutoUpdate() {
final Thread tempThre = new Thread() {
public void run() {
while (true) {

try {
Thread.sleep(10000);
} catch (InterruptedException e) {
e.printStackTrace();
}
}
};
};
};

```

```

};
tempThre.start();
}

static int CONFIGURATION_PACKAGE_SIZE = 23;
static int MANAGEMENT_PACKAGE_SIZE = 31;

static byte RECEIVED_CONFIGURATION_PACKAGE = 49; //
0x31
static byte SENT_CONFIGURATION_PACKAGE = 48; //
0x30
static byte RECEIVED_MANAGEMENT_PACKAGE = 65; //
0x41
static byte SENT_MANAGEMENT_PACKAGE = 64; // 0x40

static byte CR_RECEIVED_PACKAGE_SIZE_ERROR = 1; // 0b
0000|0001 ==> 0x01
static byte CR_RECEIVED_PACKAGE_ID_ERROR = 2; // 0b
0000|0010 ==> 0x02
static byte CR_PERIPHERAL_DELETED = 3; // 0b
0000|0011 ==> 0x03
static byte CR_PERIPHERAL_ID_NOT_FOUND_ERROR = 4; // 0b
0000|0100 ==> 0x04
static byte CR_PERIPHERAL_ADDED = 5; // 0b
0000|0101 ==> 0x05
static byte CR_PERIPHERAL_ID_RESERVED_ERROR = 6; // 0b
0000|0110 ==> 0x06
static byte CR_PERIPHERAL_ONBOARD_ADD_ERROR = 7; // 0b
0000|0111 ==> 0x07
static byte CR_PERIPHERAL_CHANGED = 8; // 0b
0000|1000 ==> 0x08
static byte CR_PERIPHERAL_NOTHING_CHANGED = 9; // 0b
0000|1001 ==> 0x09
static byte CR_PERIPHERAL_ONBOARD_CHANGE_ERROR = 10; // 0b
0000|1010 ==> 0x0A
static byte CR_PROTECTION_ACTIVATED = 11; // 0b
0000|1011 ==> 0x0B
static byte CR_PROTECTION_DEACTIVATED = 12; // 0b
0000|1100 ==> 0x0C
static byte CR_PROTECTION_NOT_CHANGED = 13; // 0b
0000|1101 ==> 0x0D
static byte CR_ALARM_TRIGGERED = 14; // 0b
0000|1110 ==> 0x0E
static byte CR_PERIPHERAL_RESETEED = 15; // 0b
0000|1111 ==> 0x0F

static boolean protection = false;

public static byte[] createManagementPackage(int PeripheralId, int
Data, byte flags)
{
byte[] msgArr = new byte[MANAGEMENT_PACKAGE_SIZE];
String KullaniciID = "DENEME12";
String StartTime = "strt";
String StopTime = "stop";
byte day = 0;
String timestamp = "timetime";

```

```

msgArr[0] = SENT_MANAGEMENT_PACKAGE;
System.arraycopy(KullaniciID.getBytes(), 0, msgArr, 1,
KullaniciID.length()); // 1, 2, 3, 4, 5, 6, 7, 8. byte'lar

msgArr[9] = (byte)((PeripheralId >> 8) & 0x000000FF);
msgArr[10] = (byte)(PeripheralId & 0x00FF);

msgArr[11] = (byte)((Data >> 8) & 0x000000FF);
msgArr[12] = (byte)(Data & 0x00FF);

msgArr[13] = flags;

System.arraycopy(StartTime.getBytes(), 0, msgArr, 14,
StartTime.length()); // 14, 15, 16, 17. byte'lar
System.arraycopy(StopTime.getBytes(), 0, msgArr, 18,
StopTime.length()); // 18, 19, 20, 21. byte'lar

msgArr[22] = day;

System.arraycopy(timestamp.getBytes(), 0, msgArr, 23,
timestamp.length()); // 23, 24, 25, 26, 27, 28, 29, 30. byte'lar

return msgArr;
}

public static byte[] createConfigurationPackage(int PeripheralId,
byte moduleId, byte ioID, byte deviceType, byte flags)
{
byte[] msgArr = new byte[CONFIGURATION_PACKAGE_SIZE];
String KullaniciID = "DENEME12";
String timestamp = "timetime";

msgArr[0] = SENT_CONFIGURATION_PACKAGE;
System.arraycopy(KullaniciID.getBytes(), 0, msgArr, 1,
KullaniciID.length()); // 1, 2, 3, 4, 5, 6, 7, 8. byte'lar

msgArr[9] = (byte)((PeripheralId >> 8) & 0x000000FF);
msgArr[10] = (byte)(PeripheralId & 0x00FF);

msgArr[11] = moduleId;
msgArr[12] = ioID;
msgArr[13] = deviceType;
msgArr[14] = flags;

System.arraycopy(timestamp.getBytes(), 0, msgArr, 15,
timestamp.length()); // 15, 16, 17, 18, 19, 20, 21, 22. byte'lar

return msgArr;
}
}

```

```

package com.example.akyonv07;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketException;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.StringTokenizer;

import android.R.menu;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.StrictMode;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;

```

myImage.java

```

package com.example.akyonv07;
import android.content.Context;
import android.media.MediaPlayer;
import android.util.AttributeSet;
import android.util.Log;
import android.view.Gravity;

```

```

import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;

public class myImage extends ImageView {
public static int msgLen = 31;
String IO_ID;
String Name;
String Type;
boolean Alarm = false;
boolean State = false;

// Images ID
Integer Positive = 0;
Integer Negative = 0;
Integer AlarmSt = 0;

public myImage(Context context, AttributeSet attrs) {
super(context, attrs);
this.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View v) {
OnClickListener();
}
});
}

public void setSensorImage() {
LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(150, 150);
layoutParams.gravity = Gravity.CENTER | Gravity.TOP;
this.setPadding(2, 2, 2, 2);

this.setLayoutParams(layoutParams);
if (Type.equals("YANGIN")) {
Positive = R.drawable.blue_fire;
Negative = R.drawable.grey_fire;
AlarmSt = R.drawable.red_fire;
}

if (Type.equals("LAMBA")) {
Positive = R.drawable.blue_lamb;
Negative = R.drawable.red_lamb;
}

if (Type.equals("DOGALGAZ")) {
Positive = R.drawable.blue_ngas;
Negative = R.drawable.red_ngas;
}

if (Type.equals("SU")) {
Positive = R.drawable.red_water;
Negative = R.drawable.blue_water;
}

if (Type.equals("DUMAN")) {
Positive = R.drawable.blue_smoke;
Negative = R.drawable.grey_smoke;
}
}

```

```

AlarmSt = R.drawable.red_smoke;
}

if (Type.equals("HAREKET")) {
Positive = R.drawable.blue_pir;
Negative = R.drawable.grey_pir;
AlarmSt = R.drawable.red_pir;
}

if (Type.equals("SUSENSOR")) {
Positive = R.drawable.blue_waters;
Negative = R.drawable.grey_waters;
AlarmSt = R.drawable.red_waters;
}

setImage(0);
}

public void setImage(int State) {
switch (State) {
case 0:
this.setImageResource(Negative);
this.State = false;
break;
case 1:
this.setImageResource(Positive);
break;
case 2:
this.setImageResource(AlarmSt);
if (Type.equals("HAREKET") || Type.equals("DUMAN") ||
Type.equals("YANGIN") || Type.equals("SUSENSOR")) {

PlayAlarm();
Alarm = true;
}
this.State = true;
this.State = true;
default:
break;
}
}

public void OnClickListener() {
// Toast.makeText(getApplicationContext(), IO_ID + " " + Name + " " + Type,
// Toast.LENGTH_SHORT).show();
Log.w("ImageOnClick", IO_ID + " " + Name + " " + Type);
if (Type.equals("LAMBA")) {
if (State) {
State = false;
} else {
State = true;
}
}

}

if (Type.equals("DOGALGAZ") || Type.equals("SU")) {
if (State) {
State = false;
}
}
}

```

```

} else {
State = true;
}
}

if ((Type.equals("HAREKET") || Type.equals("DUMAN") ||
Type.equals("SUSENSOR") || Type.equals("YANGIN")) && (Alarm)) {

if (State) {
State = false;
// Sireni susturacak paketi gonder.
Alarm = false;
AkyonMain.SendMessageTCP(AkyonMain.createManagementPackage(45, 0,
(byte) 0));
this.setImage(0);
} else {
State = true;
}
}

SendMessage(State);

}

public void ControlFlagAlarm(int FormatID, int AlarmSt) {
if (AlarmSt == 1) {
if (Type.equals("HAREKET") || Type.equals("DUMAN") ||
Type.equals("YANGIN") || Type.equals("SUSENSOR")) {
if (FormatID == 6) {
PlayAlarm();
Alarm = true;
setImage(2);
} else if ((FormatID == 4) && (!Alarm)) {
setImage(1);
}
}
}
if (Type.equals("LAMBA")) {
setImage(AlarmSt);
State = true;
}
}
if (Type.equals("SU") || Type.equals("DOĞALGAZ")) {
setImage(0);
State = false;
}
}
else if (AlarmSt == 0) {
if (Type.equals("HAREKET") || Type.equals("DUMAN") ||
Type.equals("YANGIN") || Type.equals("SUSENSOR")) {
if (FormatID == 6) {
Alarm = false;
setImage(1);
} else {
setImage(0);}
}
}
if (Type.equals("LAMBA")) {
setImage(AlarmSt);
State = false;
}
}
}

```

```

if (Type.equals("SU") || Type.equals("DOGALGAZ")) {
    setImage(1);
    State = true;
}
}
}

public void SendMessage(boolean onOff) {
    AkyonMain.SendMessageTCP(preMessage(onOff));
}

/**
 * Bu fonksiyonun amacı BBB(Beagle Bone Black) için mesaj
 paketinin
 * hazırlanmasıdır. 0x00 Hexadecimal gösterim şeklindedir.
 *
 * @param onOff
 *          Cihazın açık kapalı olma durumunu temsil eder
 * @return {@link Byte} Bir byte Array döndürür
 */
public byte[] preMessage(boolean onOff) {
    String KullaniciID = "DENEME";
    byte[] msgArr = new byte[msgLen];
    msgArr[0] = 0x40; // Mesaj ID Giden Mesaj Paketi
    KullaniciID.getBytes(0, KullaniciID.length(), msgArr, 1); //
    Kullanıcı
    // ID to
    // Byte
    // Array
    msgArr[10] = Byte.valueOf(IO_ID); // SensorID

    // Cihaz açık mı kapalı mı durumu 2. bit ile temsil edilir.
    // 0: Kapalı 1: Açık
    if (onOff) {
        msgArr[12] = 1;
    } else {
        msgArr[12] = 0;
    }
    return msgArr;
}

public void PlayAlarm() {

    final Thread tempThre = new Thread() {
        public void run() {
            MediaPlayer mp = MediaPlayer.create(getContext(),
            R.raw.soundfire);
            mp.start();
            while (true) {
                if (!Alarm) {
                    mp.stop();
                    return;}
            }
        };
    };
    tempThre.start();
}
}
}

```

myLayout.java

```
package com.example.akyonv07;

import android.content.Context;
import android.util.AttributeSet;
import android.widget.LinearLayout;

public class myLayout extends LinearLayout{
    String RoomName="";
    String RoomLabel="";
    Context context;
    public myLayout(Context context, AttributeSet attrs) {
        super(context, attrs);
        this.context = context;
    }
}
```

myUser.Java

```
package com.example.akyonv07;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import android.util.Log;

public class myUser {
    String UserName;
    String UserPass;
    String userType;

    // 0 Administrator
    // 1 Ebeveyn
    // 2 Cocuk

    // UserName Ascii Karakter kodlari ile
    // UserPass SHA-1 olarak saklanmakta
    myUser(String userName, String userPass,String userType) {
        this.UserName = userName;
        this.UserPass = userPass;
        this.userType = userType;
        Log.w("HASH Name : ",this.UserName.toString());
        Log.w("HASH Pass : ",this.UserPass.toString());
        Log.w("User Type :", this.userType.toString());
    }

    public static String SHA1Converter(String st) {
        try {
            final MessageDigest digest = MessageDigest.getInstance("SHA-256");
            digest.update(st.getBytes());
            byte byteData[] = digest.digest();

            // Convert the byte to hex format method
            StringBuffer sb = new StringBuffer();
```

```

for (int i = 0; i < byteData.length; i++) {
sb.append(Integer.toString((byteData[i] & 0xff) + 0x100,
16).substring(1));
}

return sb.toString();
} catch (NoSuchAlgorithmException e) {
e.printStackTrace();
return null;
}
}

boolean isSame(String userName, String userPass) {
userName = SHA1Converter(userName);
userPass = SHA1Converter(userPass);
Log.w("Kullanıcı Adı",userName);
Log.w("Kullanıcı Sifresi",userPass);
if(userName.equals(this.userName)&&userPass.equals(this.userPass)) {
Log.w("Durum","Burada");
return true;
}
return false;
}

/**
 * 0: Yönetici
 * 1: Ebeveyn
 * 2: Çocuk
 * @return
 */
public int getUserType(){
if(userType.equals("Yönetici")) return 0;
if(userType.equals("Ebeveyn")) return 1;
if(userType.equals("Çocuk")) return 2;
return -1;
}
}

```

KİŞİSEL YAYIN VE ESERLER

Menteşođlu M., Kavak A., Yakut M., Tangel A., Sahin S., Ozcan H., Mobil Cihazlar ile Kontrol Edilebilen Akıllı Ev Yönetim Sistemleri için Haberleşme Protokolü Tasarımı ve Gerçeklenmesi, *SİU 2015 IEEE 23. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Malatya, Türkiye, 16-19 Mayıs 2015.



ÖZGEÇMİŞ

1982 yılında İstanbul'da doğdu. İlk, orta ve lise öğrenimini İstanbul'da tamamladı. 2010 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünden 2013 yılında mezun oldu. 2013-2016 yılları arasında, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı'nda Yüksek Lisans öğrenimini tamamladı. 2000 yılından 2013 yılına kadar Deniz Kuvvetleri Komutanlığının deniz ve kara birliklerinde DSA Astsb.' ı olarak görev yapmıştır. 2013 yılından itibaren özel sektörde Yazılım Uzmanı olarak çalışmaya başlamıştır. 2016 Mart ayından beri Cardtek grup bünyesinde Yazılım Uzmanı olarak çalışmaktadır.